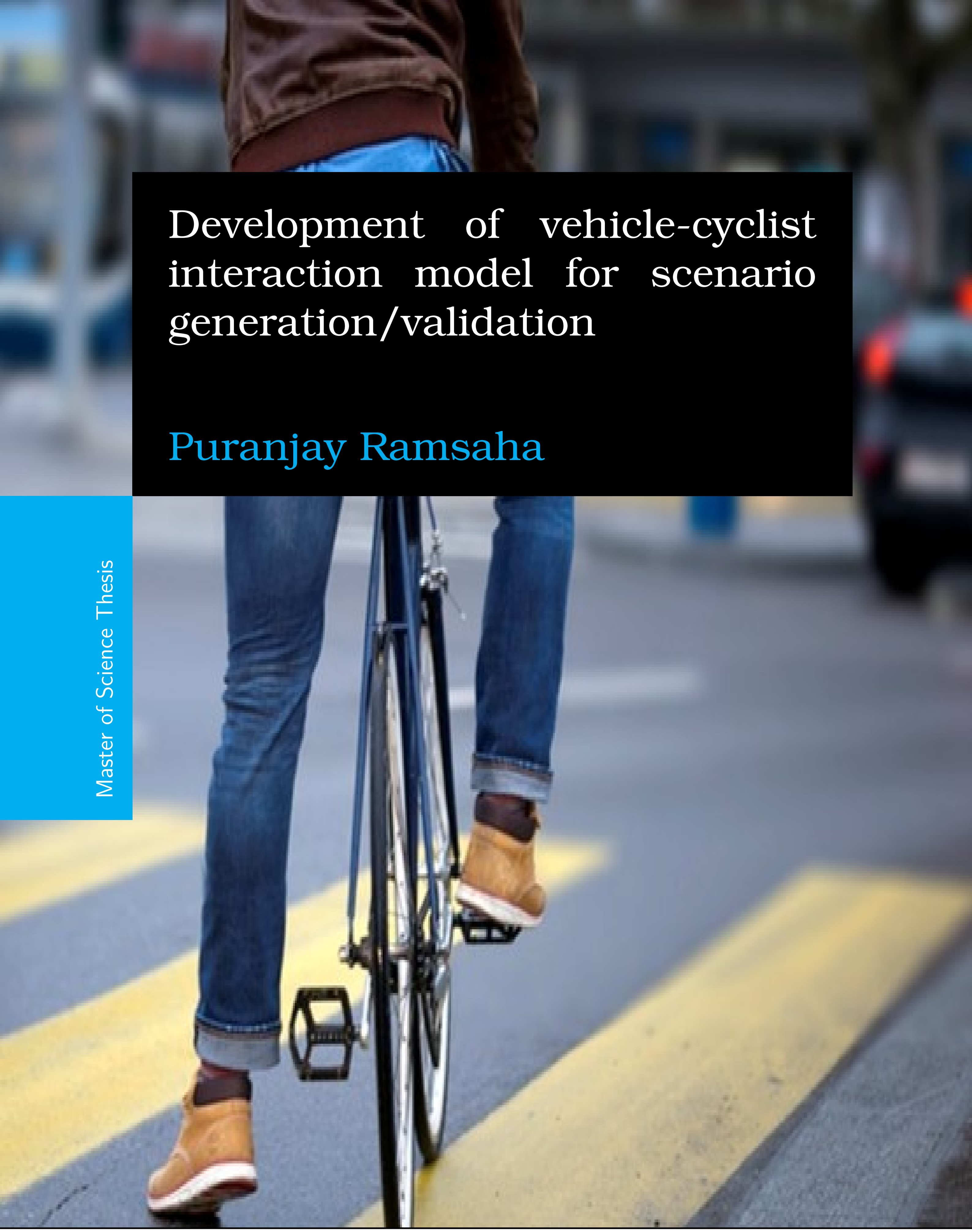


Development of vehicle-cyclist interaction model for scenario generation/validation

Puranjay Ramsaha

Master of Science Thesis

Delft University of Technology

Cognitive Robotics

Development of vehicle-cyclist interaction model for scenario generation/validation

Master of Science Thesis

For the degree of Master of Science in Mechanical Engineering at Delft University of Technology

Puranjay Ramsaha

November 23, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by TNO Automotive. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Cognitive Robotics (CoR)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Development of vehicle-cyclist interaction model for scenario generation/validation

by

Puranjay Ramsaha, 4626591

in partial fulfillment of the requirements for the degree of

Master of Science Mechanical Engineering
Track: Vehicle Engineering

Dated: November 23, 2018

Supervisor(s):

Dr. ir. Riender Happee (TU Delft)

Ir. Riske Meijer (TNO)

Reader(s):

Dr. ir. Arend L. Schwab

Dr. Julian F.P. Kooij

Abstract

The number of road accidents is increasing all over the world. When Vulnerable Road Users (VRU) are involved in an accident, they are prone to more serious injuries. Half of the fatalities around the world involve VRUs. One way of mitigating the severity and the number of VRU accidents is the introduction of Advanced Driver Assistance Systems (ADAS) functionalities. Such systems must be validated before being introduced on the market. Real-life tests for validating the system would imply driving around for millions of operational hours which would be very time consuming and therefore expensive. For this purpose, TNO has developed a scenario generation method and is currently extending it. The objective of this study was to develop a vehicle-cyclist interaction model which would then be used along the existing scenario generation method or for validating generated scenarios. The preceding literature study to this Master Thesis, from which the architecture of the models was developed and potential input parameters were defined, was used as a starting point. Based on the architecture, the interaction model was first developed with the help of a Hidden Markov Model (HMM). First, a sensitivity analysis was performed from which it was concluded that speed, acceleration, and steering angle are important parameters for a cyclist, whereas speed acceleration and yaw rate are important for a vehicle's intent recognition. Furthermore, this analysis also indicated that distance to the center of the crossing is a very good interaction parameter. After selection of the parameters, the model was trained and cross-validated using data earlier recorded at TNO. The cross-validation results of the interaction model showed that the intent recognition is more constant for the whole observation sequence and an improved performance is seen for a prediction time of more than $2s$ compared to a HMM without interaction parameters. From this study, the kinematic input parameters for the cyclist and the vehicle, as well as the interaction parameters, were defined that need to be included in the vehicle-cyclist interaction model. However, the model developed in this study needs to be further validated using naturalistic vehicle-cyclist kinematic data, and for all type of vehicle-cyclist scenarios.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor Dr ir. Riender Happee, from Delft University of Technology (TU Delft) for his valuable suggestions, brainstorming, and assistance during the writing of this literature report.

I would also like to thank Ir. Riske Meijer who is my daily supervisor at TNO. I highly appreciate her suggestions, time and effort invested in helping me make this project a success. I would also thank my other colleagues at TNO, namely Dr. ir. Hala Elrofai, Ir. Erwin de Gelder and Dr. ir. Jan-Pieter Paardekooper for their constant support and motivation.

Finally, I would like to thank my parents, Mr. Mookess and Mrs. Poonam for their unconditional love and support throughout my life. My elder sister, Srishti has constantly guided me in the right direction and always with the right words to help me get back on track. And not to forget my younger sister, Shruti, who has always helped me to be a more imaginative and creative person.

Delft, University of Technology
November 23, 2018

Puranjay Ramsaha

Chapter 1

Introduction

The number of road accidents and fatalities is increasing all over the world [1]. Each year more than 1.2 million people are killed and around 50 million people get injured worldwide in road accidents [2]. According to a study by World Health Organization (WHO), road accidents were ranked 9th for the causes of the death among people of all ages in 2015, and it is predicted that this rank will change to the 7th position by 2030 [3]. Hence, there is an urgent need to stop the trend of increasing number of accidents and to take measures to reduce the loss of life and properties.

Vulnerable Road Users (VRU), such as pedestrians and cyclist are exposed to the highest level of risk if an accident occurs. Nearly 50% of the people who die in accidents worldwide are VRU [4]. In The Netherlands, around 600 fatalities in road traffic occurred in 2013 as well as in 2014, out of which 190 cases are cyclists, which is almost 32%. The number of road accidents kept on increasing in 2015 and 2016. This makes the Netherlands ranked first in cyclist fatalities followed by Hungary and Denmark [5],[6]. However, per km ride The Netherlands is the safest country to ride a bike [6]. The fact that cycling is very popular in The Netherlands increases the chances of a cyclist having an accident even though cycle paths are present almost everywhere. In 2015, passenger cars, vans, trucks or buses were involved in 70% of the road deaths and 20% of serious road injuries among cyclists [7]. Approximately 90% of road accidents occur due to human error or negligence [8] and Advanced Driver Assistance Systems (ADAS) are being introduced to mitigate this [9]. ADAS functionalities are developed to not only help to increase the comfort and safety of the driver but also increase the safety of vulnerable road users [10]. Also, autonomous vehicles are thought to be the future solution for increasing safety and comfort as well as reducing traffic jams and carbon dioxide emission [11].

1-1 Problem Definition

The development of more ADAS functionalities are very promising steps towards autonomous driving however, their introduction brings along a lot of responsibilities. One of the main challenges is to test the functionality and give the approval for it to be introduced on the market. TNO works on the development and testing of ADAS functionalities to ensure the safety of the driver and the VRU. There are several ways of testing of an ADAS, namely testing in prescribed conditions in a test laboratory (e.g. Euro NCAP Cyclist-AEB tests), driving around in a car equipped with the ADAS, Simulation in the Loop (SIL) and Hardware in the Loop (HIL). Driving in a car for real-life and human-in-the-loop testing is very time consuming and expensive as a very large number of scenarios should be tested. Also, it is predicted that at least one billion operational hours would be required for the verification of an ADAS, which is not feasible. [12],[13].

By means of simulations, the interaction of the car with the other road users could be created virtually, and the performance of the specific ADAS could be assessed for many scenarios [14]. The design of the ADAS consists of creating a controller which stabilizes the vehicle in critical situations. Hence, virtual testing is preferred for assessing the stability of the controller at an initial design phase. Virtual testing offers the possibility of simulating many normal and critical scenarios in a short time period where real life test would have more serious implications. Critical situations such as a near miss or an accident could involve serious injuries which are not acceptable during the test phase. Hence, safety considerations make these scenarios untestable in HIL and real-life testing.

For the HIL, a vehicle equipped with a specific ADAS is coupled to a simulation environment. The road interaction for the car under test is simulated using a dynamometer and robot vehicles simulate the surrounding traffic [15]. HIL offers the benefit of testing the ADAS functionality in a real car under controlled conditions. The performance of the controller also depends on the sensors present in the vehicle. The best way to assess the vehicle's perception of the surroundings is to use physical sensors. In this way, the potential of the algorithms developed for extracting the useful information from the real sensors can be assessed. The sensors impose limitations such as noise, lack of accuracy, and clarity, etc, which the algorithms have to overcome to provide a robust perception of the environment for the controller to act in critical situations. In case of a bicycle, HIL can still be used, but the system is much simpler compared to a vehicle. The bicycle is also coupled to a simulation environment which acts as an input to the cyclist. Hence, HIL could be used to assess the cycling behavior or to check the working of controllers and actuators, such as roll stabilizers.

A combination of simulations and HIL in the development process will reduce the time and costs for prototyping thereby resulting in a more robust design [15]. The conventional methods of validation of ADAS will still need to be performed in order to test the system as a whole (the car including sensors, controllers, etc.) However, the scenarios to be tested might change when adding virtual testing. With virtual testing, standard scenarios (most common ones) might be replaced by more challenging ones by selecting scenarios based on the virtual testing results. To validate for the safety and performance of ADAS, a scenario-based method is thought of being a good option [16],[17],[18]. In a scenario-based method, the generated scenarios should be realistic, hence physically possible and close to accidents happening in real-life.

1-2 Objective

TNO is developing a scenario database to be used for the virtual assessment of ADAS (SIL and HIL). The aim of TNO is to collect a limited amount of data and to define scenarios that were encountered as well as create realistic variations of these scenarios which were not recorded. TNO developed the parametrization method for scenario generation where the important parameters are extracted and fitted to a distribution using Kernel Density Estimation (KDE). The objective of this MSc project is to develop a vehicle-cyclist interaction model for the generation of realistic variations on vehicle-cyclist crossing scenarios (see Figure 1-1). Like in the current physical testing of ADAS, the scenarios to be generated will be at an unregulated intersection since most fatal cyclist accidents take place there. The interaction model should capture on what parameters a driver and a cyclist base their decision to proceed or to stop, and the influence of the parameter variations. In other words, the model will capture the behavior of the driver and a cyclist and can be adapted with ADAS features for the development of ADAS functionalities.

This model can be further used in a framework where scenarios are generated from a database, or this model could be used for checking the validity of a scenario which has been created by TNO's method of scenario generation. As a vehicle-cyclist interaction model will be developed, this will help TNO in testing ADAS and Automated Driving Systems (ADS). As a scenario-based method is used, the effort and investment will be greatly reduced. Furthermore, as the ADAS will be tested rigorously, the number of vehicle-cyclist accidents can further decrease. As the models can be extended to validate ADS, the transition to autonomous driving can be smooth as acceptability can be increased with other road users. Hence, this will be in line with the efforts of the Ministry of Transport, Public Works and Water Management in improving the safety of the roads of the Netherlands for all users [19].

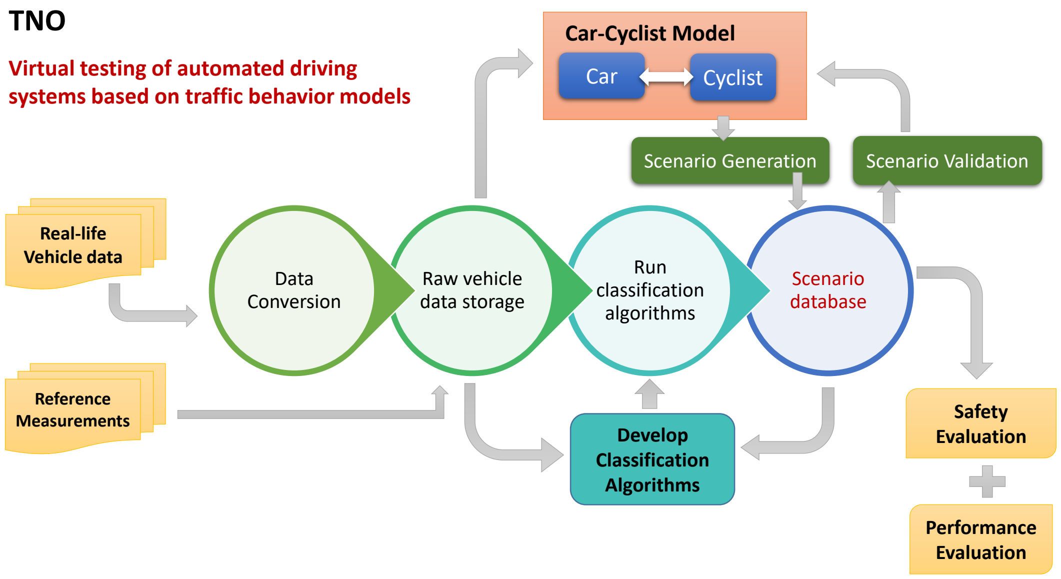


Figure 1-1: Testing of ADAS using scenarios

1-3 Thesis Outline

The thesis is structured as follows: Chapter 2 provides the summary of the Literature survey as described in [20], which is about other car-cyclist interaction models as well as on modeling techniques that might be appropriate for the development of a car-cyclist interaction model for the application of scenario generation. Chapter 3 explains the development of the interaction model. Chapter 4 describes the source and the characteristics of the data that is used for training the model as well as testing its performance. Next, Chapter 5 shows and discusses the results of the car-cyclist interaction model. Chapter 6 describes the conclusions drawn from this study as well as recommendations for future work.

Chapter 2

Literature Survey

The selection of the appropriate parameters and modeling techniques is prime for the development of the interaction model. Considering the fact that different parameters and models are required for predicting the driver and cyclist intentions, the interaction model is subdivided into two parts, namely the driver and the cyclist. Hence, the literature study was focused on models for intention recognition for each of these types of road users. A lot of research has been carried out to recognize driver intentions while limited research has been performed for recognition of cyclist intentions. Vehicle-cyclist interaction models for scenarios generation do not exist yet. Vehicle-cyclist models were mainly developed for traffic-flow models, however these have another time-scale, as such cannot be used for this study.

2-1 Selection of parameters and modeling techniques

After an extensive review of the literature on the methods for recognition of driver intentions, it was concluded that probabilistic models cannot be considered as they are helpful only in simple situations. The same is observed with the model based on fuzzy logic. These models cannot be easily extended to recognize the intentions of a driver at intersections compared to machine learning algorithms. The two most popular and commonly used machine learning models are Support Vector Machine (SVM) and Hidden Markov Model (HMM). SVM are useful in the classification of behavior in a relatively simple situation, where the decisions are already defined. Nevertheless, existing training algorithm, like Forward and Viterbi algorithm, can be easily applied to HMM, making it a more suitable method for modeling driver behavior. Hence, two tasks, namely segmenting and classifying the behavior, can be combined in one step. The models based on HMM, can be understood by a person and they make use of history which is not the case for the other models. The decisions taken by drivers are usually not defined and can be considered as internal states. HMM can be trained with data to facilitate the estimation of these states and better predict the behavior. In this way, HMM is selected for continuously estimating the driver intention. Table 2-1 [20] shows a summary of the models used previously in predicting driver intention. Most of the studies have used data collected from the CAN-Bus, as it is a cheap source and easily available. The combination of vehicle speed, acceleration and yaw rate gives the best indication of the vehicle states. Hence, this combination is selected for training and testing purposes for the driver intention.

Methods (No. of studies)	Type of Model		Type of Vehicle Detected	Type of data used				
	Intent	Interaction		CAN	Camera	GPS/ Map	Lidar/ Radar	Simulations
Hidden Markov Model (12)	✓		Ego Cars,Trucks*	9	2	-	1	-
Support Vector Machine (4)	✓		Ego Cars, Trucks*	2	-	-	2	-
Fuzzy Model (4)	✓		Ego Cars, Trucks*	2	2	-	-	-
Gaussian Mixture Model (2)	✓		Ego	2	-	-	-	-
Cellular Automata Model (1)	✓		Ego	1	-	-	-	-
Hybrid Dynamical System (2)	✓		Ego	2	-	-	-	-
Intelligent Driver Model (2)		✓	All vehicles	-	-	1	-	1
Binary Probit Model (2)	✓		Ego	2	-	-	-	-
Probabilistic Models (7)	✓		Ego Cars, Trucks*	5		1		1

* These vehicles were detected with Camera/GPS/Map/Lidar/Radar/Simulations

Table 2-1: Comparison of driver intention models

Limited research has been carried out on cyclists and the studies found were mainly focused on assessing their influence on traffic flow. Some studies were carried out for recognizing violating behavior in simple situations, however these models cannot be used or adapted for predicting intentions at intersections. Table 2-2 [20] presents a summary of the study on cyclist intention models. As HMM is good at determining the internal hidden states required in decision making, it is selected with the same motivation. For the cyclist intention, the parameters were selected based on the work carried out by TNO in the European project, Prospect [21]. Considering these factors, the velocity and steering angle were selected. For both cases, the distance to the intersection was considered the best way to describe the interaction as in real life, a driver or cyclist base their decision on the perceived distance. The conclusion of the Literature study is summarized in Table 2-3 [20]. Studies carried out for recognizing driver intentions are briefly summarized in the next section.

Methods (No. of studies)	Type of Model		Type of Bicycles Detected	Type of data used				
	Effect on traffic	Intention (violating)		CAN	Camera	GPS/ Map	Lidar/ Radar	Simulations
Data Analysis (3)	✓		All bicycles	1	2	-	-	-
Duration Model (2)		✓	Ego bicycles All bicycles*	-	2	-	-	-

* These bicycles were detected with Camera/GPS/Map/Lidar/Radar/Simulations

Table 2-2: Comparison of cyclist intention models

Type of Study	Research Done	Selected Parameters	Selected Model	Interaction Parameters
Driver Intention Recognition	Extensive	Velocity, Acceleration, Yaw rate	HMM	Distance to the intersection
Cyclist Intention Recognition	Limited	Velocity, Steering angle	HMM	Distance to the intersection

Table 2-3: Summary of the Literature Study

2-2 Previous Work on Driver Modeling using Hidden Markov Model

Amsalu and Homaifar [22] selected the scenario of a road intersection with traffic lights and described the driver behavior modeling near it using HMM. The model had to estimate the driver's intention (ego vehicle) when approaching the intersection. Hence, the aim of this work was to develop a driver model to be able to recognize the possible actions of the driver (to stop, go straight, take a left or right).

A combination of Genetic Algorithm (GA) and Baum-Welch Algorithm [22] was used for training the HMM on real-world data. Usually, HMM are trained using the Baum-Welch Algorithm with a guess of the initial values for the parameters and these are updated after each iteration until the algorithm converges. Nonetheless, the drawback of using only the Baum-Welch is that the algorithm does not always converge to the global maximum (required solution) but keeps oscillating around a local maxima (close to the initial guess) [23]. Hence, Amsalu and Homaifar [22] chose this combination as the GA extends the local search to the entire solution space. In this way, the GA ensures that the optimum parameters are found for the training of the best HMM. For faster convergence, the Baum-Welch is used in between each iteration with the GA.

With the help of the Hybrid-State System (HSS) [24], it is possible to describe the vehicle dynamics as a continuous-state system (CSS) and to describe the decisions of the driver as a high level discrete-state system (DSS) (see Figure 2-1) [22]. By using HMMs for the modeling, changes in the CSS result in a modification in the DSS, which were not previously defined. The changes in the driver states were found by using the observation data to train the model. The observations were made based on sensors installed on the ego-vehicle. The HMM used in this case can be described using a set of three parameters, namely the prior probability distribution π, the transition probability matrix A and observation probability matrix B. Using the Baum-Welch algorithm, the driver state at a given time t can be expressed as

$$S(t) = \arg\max_{i} P(o_1, o_2, \ldots, o_t|\lambda_i) \quad i = 1, \ldots, n. \tag{2-1}$$

where $o_1, o_2, \ldots, o_t$ are the observations data for the chosen signals (velocity, acceleration and yaw-rate), λ_i are the models that are to be trained and i are the estimated driver states.

Real world data was collected using a car equipped with sensors (GPS unit, CAN bus and cameras). Some specific paths were selected to represent urban conditions. The road intersections were most relevant to this study and 6 left turns, 18 straight, 5 right turn and 21 stop maneuvers were recorded. Three observations from the recorded data, namely velocity $(0 - 15m/s)$, yaw rate $(-35$ to $25°/s)$ and acceleration $(-3$ to $4m/s^2)$ were selected as the dataset to train and test the model. The confusion matrix was generated using the testing set, which accounts for 30% of the data set. The ground truth was found by watching the recorded videos and manually registering the drivers' decisions.

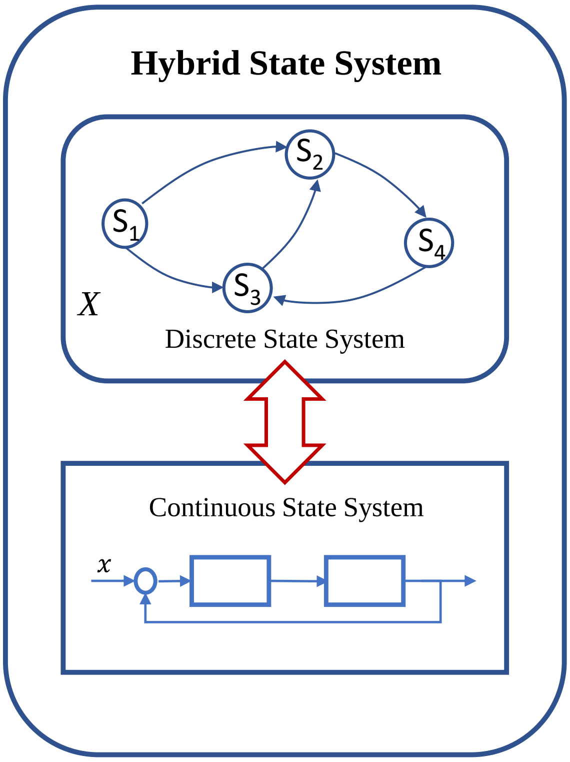


Figure 2-1: Hybrid State System
[22]

Amsalu and Homaifar [22] compared the HMM trained with the combination of GA and Baum-Welch algorithm to the HMM trained with just the Baum-Welch algorithm. A confusion matrix consists of rows and columns, where each row shows the number of instances in an actual class and each column represents the number of instances in a predicted class [22]. The model gives a prediction of the driver's intention every $0.1s$. The GA had an overall accuracy of 76.33% at predicting straight, stop, left and right turn maneuvers while the Baum-Welch algorithm achieved an overall accuracy of only 62.03% at recognizing the same intents. It was concluded from the results that recognizing the final intention occurred quite near to the intersection (approx 1 second before). Hence, a combination of GA and Baum-Welch algorithm resulted in an improvement of 14.3% over the training done only with Baum-Welch algorithm. Furthermore, this work could be extended to other driving scenarios such as lane change, merging, entering ramps and so on.

Gadepally et al. [25] also presented a framework based on both HSS and HMM for recognizing the driver's intentions at intersections. Gadepally et al. [25] selected the same scenario as Amsalu and Homaifar [22] for data collection. GPS, CAN bus and video data were recorded from the inbuilt and other installed sensors in the ego vehicle. This was done to find the good combination of signals for training and testing the model and the ground truth was found out manually by analyzing the data. 10 straight, 10 right, 10 left, and 10 stop maneuvers were successfully recorded. Out of all the signals recorded, the combination of velocity, acceleration and yaw rate was the best fit to carry out the intent recognition of the ego vehicle. The model was trained using the Baum-Welch and Viterbi algorithm to find the optimal HMM parameters and the future state respectively. An estimate of the intent is given at a regular interval of 0.1-0.2 s (depending on the sequence) for each one of 40 recorded maneuvers. The proposed model (HMM+HSS) was compared to the traditional method (K-nearest neighbor (KNN)) and the manual marking. Out of the HMM+HSS and KNN, the proposed model was more superior and outperformed the traditional model as HMM made use of history. When the HMM+HSS and the human marking were compared, it was seen that the HMM was able to detect the behavior much faster(1-2 s ahead) as it had access to the kinematics of the vehicle. In general, the proposed framework was proved to be quite promising.

In the same context, Amsalu et al. [26] carried out a study where discrete HMM and HSS were used for modeling the driver behavior. Compared to [22] and [27], it was seen that acceleration was not very useful to find the difference between the different driving maneuvers. The ground truth was found out manually from the recorded videos and the driver's actions were noted for the highway merging, entering/exiting a ramp and for road intersections (straight, stop, left and right turn). A total of 9 straight, 6 left turns, 5 right turns, and 7 stop maneuvers were recorded. Velocity ($0 - 13\ m/s$) and yaw rate ($-20\ to\ 25°/s$) were the observations mostly used for predicting behavior at road intersections. The real world data was divided into six categories based on different combinations of speed and the driver's action. The continuous data was discretized (0.1 s time step) to train (using the traditional Baum-Welch algorithm) and validate the model. Similar to Amsalu and Homaifar [22], Amsalu et al. [26] used a confusion matrix to assess the results. Using 30% of the data for testing, it was seen that this method attained an overall accuracy of 89.45% at estimating the behavior at the road intersections. The model had an updated estimation every 0.1 s and was able to carry out the intent recognition 1-2 s before arriving at the intersections. Hence, it was concluded that the discrete HMM were more accurate than continuous HMM.

HMM can also be used with two different approaches, namely top-to-bottom (starting with the whole route and identifying some maneuvers) and bottom-to-top (starting with the maneuver recognition to get an essence of the whole picture). In a study by Sathyanarayana et al. [28], an improved method of Pentland and Liu [29] was used to formulate a framework to be used for recognizing maneuvers of the driver and routes. Two routes (residential and commercial regions) each containing right turn, left turn, lane change, cruising and car following conditions are selected for data collection. The chosen routes allowed the total collection of 112 right turns, 29 left turns, and 70 lane change. The signal channels of the ego vehicle included cameras (to monitor the driver and road), microphone, GPS, CAN, pedal actuation. The CAN-Bus data was preferred over the other signals as it is a low-cost solution for modeling. Hence, the vehicle speed, steering wheel angle and brake/acceleration pedal actuation were selected to train and validate the model.

For the bottom-to-down approach, 8, 16, 32, and 64 Gaussian mixtures were used for training the continuous HMM. The best results were obtained with 64 mixtures and 4 states. For the right turns, left turns and the lane change an accuracy of 100%, 93% and 81% were obtained respectively which proved to be very encouraging to also test the continuous model on distracted driving. For the distracted driving, the model achieved an accuracy of 100% for left turn and lane change but the performance decreased for right turns. This could have been due to the distracted data being set too close to the neutral rather than distracted driving. The same data was used for training the discrete HMM and 4, 8, and 12 sized code-books were used for the symbol representation. On increasing the size, the performance of the discrete model improved. With a code-book size of 12, an accuracy of 100%, 87% and 75% was observed for neutral right turn, lane change and left turn respectively. With an accuracy of 95% for the distracted driving, Sathyanarayana et al. [28] deduced that if a maximum-likelihood algorithm was used, maneuvers can be detected properly. By comparing the continuous and discrete HMM, detection of maneuvers was better performed using the continuous model.

Wu et al. [30] did a study to predict the possible steering action of a driver. This study was closely aligned with driver behavior modeling as the authors wanted to develop a model to estimate the steering action at a certain time. Wu et al. [30] collected video data and extracted two main parameters namely the vehicle speed and direction. The selected scenario was at a four-way road junction with traffic lights. The HMM were trained using the traditional method (Baum-Welch algorithm) to find the HMM parameters. The study showed that the HMM was able to predict the steering action quite well. Some maneuvers were detected 10 frames ahead while some were detected with a delay of 10 frames. Hence, better results could be obtained if the number of states was increased and if there was less interference.

Mitrovic [31] presented a discrete HMM model to recognize and predict driving events. The concerned vehicle had to interact with its environment and other vehicles (e.g making a left or right turn, stopping, etc). No particular routes were selected for this experiment, but there was an attempt to include the main features of urban areas. Mitrovic [31] used a left-to-right model over the general model as the latter was better able to capture the changing characteristics of the data. The relevant data collected for training and validation were the vehicle speed, longitudinal and lateral acceleration. In total, 238 maneuvers including 41 right curve, 16 right on the roundabout, 36 right turns, 56 left curves, 20 left on a roundabout, 59 left turns, 10 straight on a roundabout, were recorded as the dataset. The codebook size of 16 was selected based on a trade-off between recognition accuracy and error. For the training purpose, Mitrovic [31] used 30% of the original data, and the rest was used for testing of the model. The commands were presented to the driver (for e.g stop, turn left or right on the next intersection, change line, and pass the car in front). The HMMs were trained to be able to recognize the driver's intent 2 s after the text command was presented to him. A limited number of sensors were used and the results showed that the model was still quite accurate (an overall accuracy of 98.3% was obtained), robust and trustworthy. This framework was different from the previous work carried out as the data collection was done by inexpensive sensors present on board on the car preferably.

The use of HMM to detect driver intention was also made by Berndt et al. [32]. The selected scenario for this research included road intersections, lane change and turn maneuvers. The critical instance when a maneuver has to be detected is at the start. This part is very relevant as it could be used to alert the driver of any illegal action or behavior to prevent accidents. Three hours of freeway driving were used to extract 100 and 50 lane change maneuvers for training and testing respectively. Several hours of urban driving resulted in 60 and 35 turn maneuvers for training and testing respectively. A window size of about 4 and 2 s was used for the turn and lane change maneuvers respectively. Steering wheel angle and yaw rate were used during the study. A left-to-right model was used which allowed the use of the same parameters as the overall model for the extracted sub-models. HMMs with six, nine and twelve states were used and Berndt et al. [32] found that the model with nine states performed better than the rest. For the training purpose at intersections, the third until the sixth states were extracted from the overall model. But, as the first two states were not extracted, the turning action was detected a bit late. Berndt et al. [32] also wanted to find the best combination of the observation signals for the model, but incorporating other signals didn't improve the performance. As the amount of data available was restricted, it was possible that the performance was not properly determined and having more data could help solve this issue. Berndt and Dietmayer [33] carried a similar study and promising results were obtained as Berndt et al. [32].

Boyraz et al. [34] presented a framework based on HMM to recognize the driver's intentions in urban conditions and to classify it as good or bad. The speed and steering wheel angle data was collected from a driving simulator. Twenty drivers took part in the experiment and hence, the collected data promised to be interesting due to the diversity. 5 right turn, 5 left turns, 5 U-turn and 5 roundabouts were collected for training and testing. Results showed that the model was able to capture the maneuvers and also classify the driving as good or bad based on the class division done.

Moreover, Zou and Levinson [35] used the HMM approach to describe conflicting and non-conflicting driving behaviors. Only the intersection and left-turn maneuvers were considered at road intersections. Fourteen radar and two lidar sensors were placed at the intersections. The radar sensors captured the vehicle speed and position while the lidar sensors gave an indication of the vehicle size. In total, six states were chosen to keep the analysis simple and yet interesting. A single HMM was initially used to model the whole data set of five hours of driving. 1963 vehicles were extracted from the data for estimating the intent with a frequency of 10 Hz. Initial results showed that one HMM could not be used for both conflicting and non-conflicting vehicles. Hence, two different HMMs were then used for conflicting and non-conflicting vehicles. Some part of the sequence was extracted to train the model to estimate to which cluster the sequence belonged. The rest was used for validation that is, to check really to which cluster it belonged to. In this way, 30% was used for estimation and 70% was used to check in reality. The results obtained from the clustered HMM showed that the new approach is more accurate than the traditional Gipps model. The Gipps model was selected for comparison as it uses the same data signal as the proposed model.

Tang et al. [36] tackled the modeling of a dynamic and complex decision-making process using HMMs. This is quite common at road intersections due to the high interaction between vehicles and other road users and hence making this study relevant for consideration. The traditional method (Baum-Welch algorithm) was used for the training of the HMM while the decoding was done based on the Viterbi algorithm. Four cases were considered at three different intersections for the collection of video data. A total of 698 vehicle trajectories including 179 trucks and 519 passenger cars were extracted for training and testing. The vehicle speed and acceleration were divided into different categories to facilitate the training and intent recognition. Results showed that the HMM was able to predict the intention of the driver with a high accuracy and half of the drivers used a two-step decision making. This conclusion helped to consolidate the belief of Tang et al. [36] that the driver does not use a single-step decision as assumed for the conventional models.

Meng et al. [37] presented a new architecture for modeling driving behavior which could be used in controlling the number of thefts. The application might be different, but the method used was to distinguish between the driving behavior of the owner and a thief. Hence, this study is very important and interesting for this review. Meng et al. [37] used a simulator to collect acceleration, braking, and steering angle data in urban scenarios. For each driver, a different HMM was trained to find the optimal parameters and the model could be used to identify any unknown driver. Six states were chosen for each driver and a total of seven HMMs were created for the seven drivers. The results obtained were very promising as for some drivers, the model attained a highest success rate of 100% and a lowest rate of 40%. For the application of identifying an unknown driving behavior, the model attained a success rate of 80%.

Kuge et al. [38] assessed the driving behavior when performing a lane change and pioneered the research on the performance of HMM in capturing human behavior. A driving simulator with a motion system was set up for the collection of behavior data. Different subjects were asked to drive in the simulator for different lane change situations. Again, the Viterbi and the Baum-Welch algorithm were used in the training of the HMM to find the optimal parameters. When the models were trained with the steering angle and steering angle velocity, a recognition rate of 100% was recorded for lane change and lane keeping. Results also showed that the HMM were able to learn the behavior and the models were able to predict the behavior at an early stage with a time interval of 5 ms.

Chapter 3

Methodology

This chapter will focus upon the development of the interaction model and the intent recognition using Hidden Markov Model (HMM). HMM have been established in the field of speech recognition, pattern recognition [28] and robotics [39]. HMM is gaining ground in the field of driver behavior modeling and recognition of intent, route, violation and distraction. Many studies have shown that a decision-making process can be considered as a sequence of internal states, for example, a lane change maneuver or stopping at an intersection [29]. Hence, the internal states of the driver or cyclist can be recognized by training the HMM.

3-1 Developing the Interaction Model

An interaction between road users is defined as a complex decision-making process where different road users cross each other and interact over a certain time period. This process occurs in such a way that the road users are most of the time able to use the road without collision [40]. Hence, the name "interaction model" was coined as it is a method used for describing the detection of another road user and its decision influenced by the ego vehicle.

There are different parameters which help a road user to take a decision at an unregulated intersection, for example, the traffic rules, hand gestures, estimating the distance to the intersection and the speed of other road users. In this study, normal traffic rules are considered where the right way has priority while hand gestures are excluded. Since the distance to the intersection (center) and the speed of other road users seem to be predictive parameters, these will be used for the interaction model as well.

Table 2-3 describes the parameters (including the interaction parameters) selected based on the Literature study. These parameters will be used like another road user would estimate them, to define the architecture of the interaction model. For example, a driver approaching an intersection will estimate how fast another road user is approaching from another direction. The architecture of the interaction model is shown in Figure 3-1.

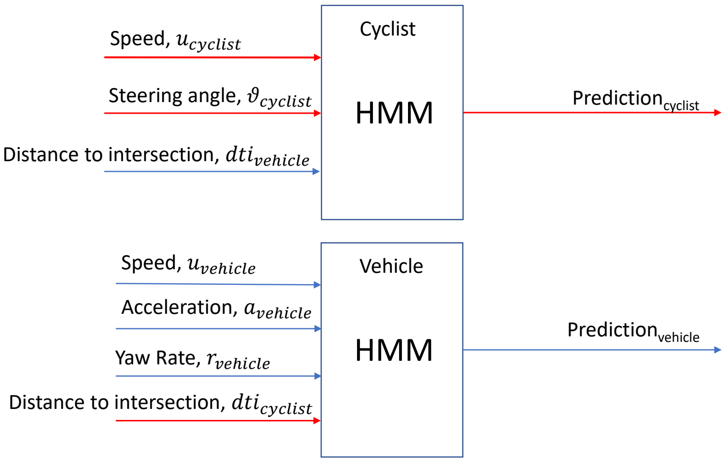

Speed, $u_{cyclist}$
Steering angle, $\vartheta_{cyclist}$
Distance to intersection, $dti_{vehicle}$
Cyclist
HMM
$\text{Prediction}_{\text{cyclist}}$
Speed, $u_{vehicle}$
Acceleration, $a_{vehicle}$
Yaw Rate, $r_{vehicle}$
Distance to intersection, $dti_{cyclist}$
Vehicle
HMM
$\text{Prediction}_{\text{vehicle}}$

Figure 3-1: Input and output parameters of the cyclist and vehicle HMM models.

3-2 Brief Introduction to Hidden Markov Model

HMM is an extension of discrete Markov models. Simple Markov models consist of states, an observation (i.e., an event) corresponding to each state, and transition probabilities representing the switch between each state. This way of modeling has some limitations to many practical applications. Hence, this concept is extended to represent the observation as a probabilistic function of the state. Hence, instead of having a direct mapping of the observation to the state, the process is stochastic (i.e., hidden) in nature.

A HMM can be characterized as follows:

1. N, the number of distinct states present in the model.
   Even though the states of a HMM are hidden, for many applications the states have a physical meaning associated with the states. For this research, the states attached to the cyclist $N_{cyclist} = 3$ are straight, turn right or stop, while the states for the vehicle $N_{vehicle} = 2$ are straight and right turn. The states are connected in a way that a state can be reached from another state (i.e., an ergodic model). The state at time t is denoted as q_t and individual states are defined as

$$S = \{S_1, S_2, \dots, S_N\} \tag{3-1}$$

2. M, number of distinct observation symbols for each state.
   The observation symbols are the outputs (measured) of the system that is being studied. The symbols are denoted by

$$\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2, \dots, \mathrm{v}_M\} \tag{3-2}$$

3. State transition probabilities, $A = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i,\ j \leq N. \tag{3-3}$$

   These present the probability of switching from state i to j and for this research $a_{ij} > 0$, as all the states are reachable.

4. Observation probability distribution function in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = P[\text{v}_k \text{ at } t | q_t = S_j], \quad \begin{array}{l} 1 \leq j \leq N \\ 1 \leq k \leq M \end{array} \tag{3-4}$$

5. Initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N \tag{3-5}$$

After the appropriate values of $N, M, A, B, \text{and} \, \pi$, the HMM can be used for determining how well it matches the given observation sequence. The steps are given below:

1. Determine the initial state $q_1 = S_i$ based on the initial state distribution π.
2. Set $t = 1$.
3. Choose $O_t = \text{v}_k$ based on the symbol probability distribution function in state S_i, i.e., $b_i(k)$.
4. Switch to a new state $q_{t+1} = S_j$ based on the state transition probability distribution for the prior state S_i, i.e., a_ij.
5. Set $t = t + 1$; Go to Step 3 if $t < T$; otherwise terminate.

Based on the description above, the specification of an HMM includes two the model parameters ($N \text{and} M$) and three probability distributions $A, B \text{and} \, \pi$. The parameter set for updating are defined as follows for convenience

$$\lambda = (A, B, \pi) \tag{3-6}$$

3-3 Developing the HMM-based Intent Recognition System

3-3-1 The Basic Problems of HMM

Based on the description of the HMM in Section 3-2, there exists three problems that must be solved to be able to implement the HMM in a real life application. The problems are listed as follows (refer to [41]):

Problem 1: How to determine $P(O|\lambda)$, the probability of having a sequence ($O = O_1 \, O_2 \ldots O_T$) given a model $\lambda = (A, B, \pi)$.

Problem 2: How to determine the state sequence $Q = q_1 q_2 \ldots q_T$ which is optimal for a given observation sequence $O = O_1 O_2 \ldots O_T$ and model $\lambda = (A, B, \pi)$.

Problem 3: How to update the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$

3-3-2 Solutions to The Basic Problems of HMM

A. Solution to Problem 1

Solution to Problem 1 involves finding the probability that the observation sequence $O = O_1\,O_2 \ldots O_T$ was generated by the model λ (i.e., choosing the model which best describes the observations). A possible way of doing this by defining all the possible state sequence of length T. Consider a fixed state sequence as follows:

$$Q = q_1\,q_2 \ldots q_T \tag{3-7}$$

where q_1 is the state at $t = 1$. The probability of having an observation sequence O for the state sequence defined in Equation 3-7 can be expressed as

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda) \tag{3-8}$$

Assuming that the observations are statistically independent, Equation 3-8 simplifies to

$$P(O|Q,\lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \ldots b_{q_T}(O_T) \tag{3-9}$$

The probability of having such a sequence Q with the model λ can be expressed as

$$P(Q|\lambda) = \pi_{q1} a_{q1q2} a_{q2q3} \ldots a_{q_{T-1}q_T} \tag{3-10}$$

The probability that both O and Q occur simultaneously is the product of Equation 3-9 and 3-10

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda) \tag{3-11}$$

The probability of having the observation sequence O given the model λ is obtained by summing the joint probability for all possible state sequence q

$$P(O|\lambda) = \sum_{allQ} P(O|Q,\lambda)P(Q|\lambda) \tag{3-12}$$

$$= \sum_{q_1,q_2\cdots,q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1q_2} b_{q_2}(O_2) \ldots a_{q_{T-1}q_T} b_{q_T}(O_T) \tag{3-13}$$

Equation 3-13 is not computationally possible as it requires $(2T-1)N^T$ multiplications and $N^T - 1$ additions. Hence, a more effective solution named forward-backward procedure will be used. Consider a forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(O_1O_2 \ldots O_t, q_t = S_i|\lambda) \tag{3-14}$$

$\alpha_t(i)$ is the probability of having a partial observation sequence for time t given being in state S_i and the model λ. The following steps can be used to solve for $\alpha_t(i)$:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N. \tag{3-15}$$

2. Induction

$$\begin{aligned}\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad & 1 \leq t \leq T-1 \\ & 1 \leq j \leq N.\end{aligned} \tag{3-16}$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3-17}$$

In a similar way, consider a backward variable $\beta_t(i)$ defined as

$$\beta_t(i) = P(O_{t+1}O_{t+2}\ldots O_T, q_t = S_i|\lambda) \tag{3-18}$$

$\beta_t(i)$ is the probability of having a partial observation sequence from time $t+1$ to T given being in state S_i at time t and the model λ. The following steps can be used to solve for $\beta_t(i)$:

1. Initialization

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N. \tag{3-19}$$

2. Induction

$$\begin{aligned}\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \qquad & t = T-1, T-2, \ldots 1 \\ & 1 \leq i \leq N.\end{aligned} \tag{3-20}$$

Using the variables $\alpha_t(i)$ and $\beta_t(i)$ reduced the overall computation to $2N^2T$.

B. Solution to Problem 2

The solving Problem 2 involves finding the best state sequence corresponding to the given observation sequence. The challenge is to find the best optimality criterion. One possible solution is to select the states q_t which are individually most probable. To apply this solution, consider the variable

$$\gamma_t(i) = P(q_t = S_i|O, \lambda) \tag{3-21}$$

$\gamma_t(i)$ represents the probability of being in state S_i at time t given an observation sequence O, and a model λ. Using the forward-backward terms, Equation 3-21 can be simplified as follows:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum\limits_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{3-22}$$

Being in state S_i at time t, $\alpha_t(i)$ and $\alpha_t(i)$ takes into account the partial observation sequences $O_1O_2 \dots O_t$ and $O_{t+1}O_{t+2} \dots O_T$ respectively. $P(O|\lambda)$ acts as a normalizing factor so that

$$\sum_{i=1}^{N} \gamma_t(i) = 1 \tag{3-23}$$

Solving for the most probable individual state q_t at time t, can be expressed as

$$q_t = \arg\max_{1 \le i \le N} [\gamma_t(i)], \quad 1 \le t \le T \tag{3-24}$$

The drawback of using Equation 3-24 is that the resulting state sequence might not be a practical solution. The technique used to solve Problem 2 is called Viterbi algorithm, and it involves finding the best state sequence $Q = q_1q_2 \dots q_T$. Consider $\delta_t(i)$ as follows:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1q_2 \dots q_t = i, O_1O_2 \dots O_t|\lambda] \tag{3-25}$$

Using induction,

$$\delta_{t+1}(j) = [\max_i \delta_t(i)a_{ij}] \cdot b_j(O_{t+1}) \tag{3-26}$$

To obtain the best state sequence, the argument which maximized Equation 3-26 must be checked. This is done by introducing the array $\psi_t(i)$. The Viterbi algorithm is described as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N. \tag{3-27}$$

$$\psi_1(i) = 0 \tag{3-28}$$

2. Recursion

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T$$
$$1 \leq j \leq N \tag{3-29}$$
$$\psi_t(j) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \quad 2 \leq t \leq T$$
$$1 \leq j \leq N \tag{3-30}$$

3. Termination

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$
$$q_t^* = \arg\max_{1 \leq i \leq N} [\delta_T(i)] \tag{3-31}$$

4. Path backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1. \tag{3-32}$$

C. Solution to Problem 3

Solution to Problem 3 involves adjusting the model parameters (A, B, π) to optimize the probability of an observation sequence given the model. The Baum-Welch (or the expectation-maximization (EM)) method Consider $\xi_t(i,j)$ which is the probability of being in state S_i at time t and state S_j at time $t+1$, given the observation sequence and model,

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | o, \lambda) \tag{3-33}$$

Using the forward and backward variables, Equation 3-33 can be written as

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{3-34}$$

As $\gamma_t(i)$ was defined as the probability of being in state S_i at time t given an observation sequence O, and a model λ, $\gamma_t(i)$ and $\xi_t(i,j)$ can be related as

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3-35}$$

The number of transitions from S_i is obtained if $\gamma_t(i)$ is summed over time t, while the number of transitions from S_i to S_j is obtained if $\xi_t(i,j)$ is summed over time t.

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{Number of transitions from } S_i$$
$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{Number of transitions from } S_i \text{ to } S_j \tag{3-36}$$

Using the Equations defined above, the re-estimation equations for λ, A and B can be expressed as

$$\begin{aligned}
\bar{\pi}_i &= \text{Number of times in state } S_i \text{ at time } (t=1) \\
&= \gamma_1(i) && \text{(3-37)} \\
\bar{a}_{ij} &= \frac{\text{Number of transitions from } S_i \text{ to } S_j}{\text{Number of transitions from } S_i} \\
&= \frac{\sum\limits_{t=1}^{T-1} \xi_t(i,j)}{\sum\limits_{t=1}^{T-1} \gamma_t(i)} && \text{(3-38)} \\
\bar{b}_k &= \frac{\text{Number of times in state } j \text{ and observing symbol } \mathrm{v}_k}{\text{Number of times in state } j} \\
&= \frac{\sum\limits_{\substack{t=1 \\ \text{s.t.} O_k=\mathrm{v}_k}}^{T-1} \gamma_t(i)}{\sum\limits_{t=1}^{T-1} \gamma_t(i)} && \text{(3-39)}
\end{aligned}$$

If the current model is defined as $\lambda = (A, B, \pi)$ and is used to calculate re-estimated parameters, the re-estimated model can be expressed as $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. If $\bar{\lambda}$ is used in place of λ, and the re-estimation procedure is repeated, the probability of O being observed with the model can be increased until a saturation point. The final outcome of the re-estimation procedure is called the maximum likelihood estimate of a HMM.

3-3-3 Using Continuous Observation Densities

The observation used for modeling are continuous in nature and hence, some restrictions must be used on the Probability Density Function (PDF). This is to ensure that the HMM parameters can be re-estimated. The PDF can be expressed in a finite mixture as

$$b_j(\mathbf{O}) = \sum_{m=1}^{M} c_{jm} \zeta[\mathbf{O}, \mu_{jm}, U_{jm}], \quad 1 \leq j \leq N \tag{3-40}$$

where $\mathbf{O}$ is the model vector, c_{jm} is the mixture coefficient for the m^{th} mixture and ζ is a Gaussian distribution with mean μ_{jm} and covariance matrix U_{jm} The mixture coefficient satisfies the stochastic constraint as follows

$$\sum_{m=1}^{M} c_{jm} = 1, \quad 1 \leq j \leq N \tag{3-41}$$

$$c_{jm} \geq 0, \quad 1 \leq j \leq N, \; 1 \leq m \leq M \tag{3-42}$$

Equations 3-41 and 3-42 allows the PDF to be normalized

$$\int_{-\infty}^{\infty} b_j(\mathbf{x}) d\mathbf{x} = 1, \quad 1 \leq j \leq N. \tag{3-43}$$

3-3-4 Using Multiple Observation Sequences

The derivation carried out above was for a model with a single observation sequence. Some applications require the use of multiple observation sequences. As shown in Figure 3-1, 3 observations would be used for the cyclist while 4 observations (including the interaction parameters) would be utilized for the vehicle intentions prediction. Hence, the re-estimation procedure must be modified to accommodate the use of multiple sequences. Consider a set of K observation sequences

$$\mathbf{O} = [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \ldots, \mathbf{O}^{(\mathrm{k})}] \tag{3-44}$$

where $\mathbf{O}^{(\mathrm{k})} = [O_1^{(\mathrm{k})} O_2^{(\mathrm{k})} \ldots O_{T_\mathrm{k}}^{(\mathrm{k})}]$ is the kth observation sequence. It is assumed that each observation sequence is independent of the other observation sequence. The goal is to update the parameters to maximize

$$P(O|\lambda) = \prod_{\mathrm{k}=1}^{\mathrm{K}} P(\mathbf{O}^{(\mathrm{k})}|\lambda) \tag{3-45}$$

$$= \prod_{\mathrm{k}=1}^{\mathrm{K}} P_\mathrm{k} \tag{3-46}$$

As the re-estimation formulas depend on the number of occurrence of different events, the formulas for the multiple sequences are adjusted by summing the individual occurrence for each sequence. The modified equation can be expressed as

$$\bar{\pi}_i = \pi_i \tag{3-47}$$

$$\bar{a}_{ij} = \frac{\sum\limits_{\mathrm{k}=1}^{\mathrm{K}} \frac{1}{P_{\mathrm{k}}} \sum\limits_{t=1}^{T_{\mathrm{k}}-1} \alpha_t^{\mathrm{k}}(i) a_{ij} b_j(O_{t+1}^{(\mathrm{k})}) \beta_{t+1}^{\mathrm{k}}(j)}{\sum\limits_{\mathrm{k}=1}^{\mathrm{K}} \frac{1}{P_{\mathrm{k}}} \sum\limits_{t=1}^{T_{\mathrm{k}}-1} \alpha_t^{\mathrm{k}}(i) \beta_t^{\mathrm{k}}(i)} \tag{3-48}$$

$$\bar{b}_j(\ell) = \frac{\sum\limits_{\mathrm{k}=1}^{\mathrm{K}} \frac{1}{P_{\mathrm{k}}} \sum\limits_{\substack{t=1 \\ \text{s.t. } O_t = \mathrm{v}_\ell}}^{T_{\mathrm{k}}-1} \alpha_t^{\mathrm{k}}(i) \beta_t^{\mathrm{k}}(i)}{\sum\limits_{\mathrm{k}=1}^{\mathrm{K}} \frac{1}{P_{\mathrm{k}}} \sum\limits_{t=1}^{T_{\mathrm{k}}-1} \alpha_t^{\mathrm{k}}(i) \beta_t^{\mathrm{k}}(i)} \tag{3-49}$$

The parameter π_i is not re-estimated as $\pi_1 = 1,\ \pi_i = 0,\ i \neq 1$.

3-3-5 Equal-width Binning

Discretization is needed for the signal, namely distance to the intersection. Discretization will allow the use of bins which improves the training of the HMM for the intent prediction. For this purpose, the equal-width discretization technique is used where data is divided into K number of bins as defined by Equation 3-50. Equal width bins are bins having the same width for the entire observation sequence. Based on [21], a bin size of $0.1m$ is selected for the distance to the intersection while the other observation sequences remain continuous in nature.

$$W = \frac{d_{max} - d_{min}}{K} \tag{3-50}$$

Based on the formulation described above, discretization allows the flexibility to give a higher weighting to the original parameters ($\lambda = (\pi, A, B)$) compared to the re-estimated parameters ($\bar{\lambda} = \bar{\pi}, \bar{A}, \bar{B}$).

$$\lambda_{updated} = (1 - \epsilon)\lambda + \epsilon\bar{\lambda} \tag{3-51}$$

where ϵ is a small number.

This technique makes the model robust to outliers or sudden change in the observation sequence and hence, enhancing the recognition performance.

3-4 Validation of the interaction model

The most commonly used method of validation comprises of dividing the whole data into two parts, namely training (70%) and testing (30%) data. For this method, a large amount of data is needed and if a relatively small dataset is available, then cross-validation is a better method for assessing the performance [42]. With a cross-validation, each time different data is left out as the test set and the training set contains the remaining maneuvers. The concept used in this validation, is that the test data must be unknown to the trained model.

Chapter 4

Experimental Data

This chapter will elaborate on the setup of the car-cyclist experiments and the dataset collected from it, that is used for the learning and validation of the car-cyclist interaction model. A description of the procedure used in preparing the data is also provided here. Furthermore, it is explained how the data was used in the validation of the interaction model.

4-1 Experimental Set-up

Cyclist and vehicle data was collected over a period of two days by Integrated Vehicle Safety (IVS), TNO in 2014. Data was recorded at the parking space of TNO as shown in Figure 4-1. The information about the participants are given below:

- The cyclists were aged between 25-55.
- Four different cyclists volunteered (1 female, 3 male).
- The driver (male) had a driving license.
- The cyclists were instructed to get to a speed of approximately 15 km/h before approaching the intersection.
- The driver was instructed to drive at a speed of 30 km/h before approaching the intersection.

The experiments were performed in a controlled environment that was created using cones as shown in Figure 4-2. This enabled the tests to be carried out quite fast and only the selected traffic was allowed to perform the maneuvers. Two scenarios, namely a cyclist crossing and turning right at an intersection were chosen as shown in Figure 4-3.

As described in Figure 4-3 cyclist were instructed to perform the following maneuvers:

A. Stop

B. Straight

C. Right Turn

The driver (the other road user) was instructed to perform the following maneuvers:

1. Straight
2. Right Turn

such that the vehicle-cyclist crossing and right turn scenarios were created.

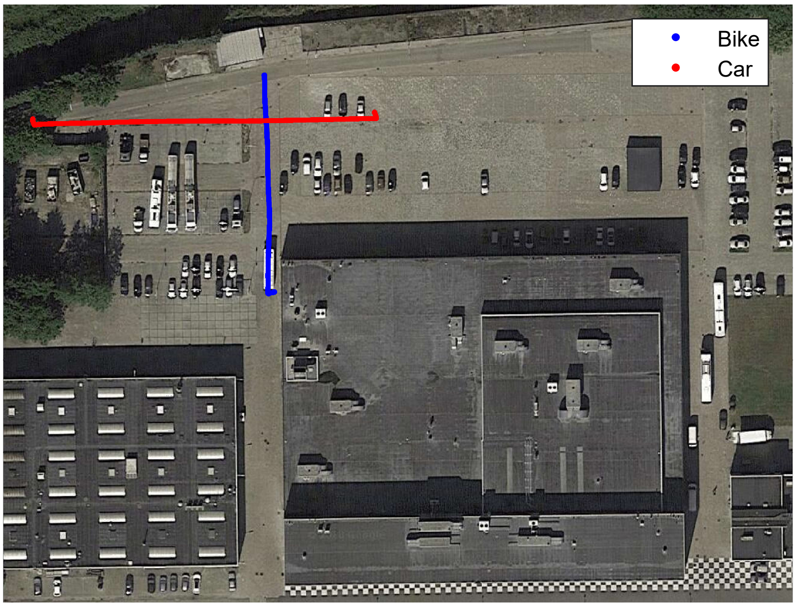


Figure 4-1: Map showing the car and bike trajectory at TNO's parking space

The data was checked for loss of signals for both the driver and cyclists. The maneuvers having all the required signals were considered for training and validation. The tests resulted in a successful collection of 61 maneuvers including, 16 straight, 21 right turns, and 24 stop for all the cyclists. The most relevant signals to this study logged from the bike and car during the experiments are described in Table 4-1 and 4-2. Video data was also collected using one camera installed in the car and one camera at the intersection as shown in Figure 4-2.

4-2 Preparation of Data

Some preprocessing had to be carried out to prepare the data for the validation of the model. This includes extracting the required signals from the raw loggings and cut off the unwanted segment of the data.

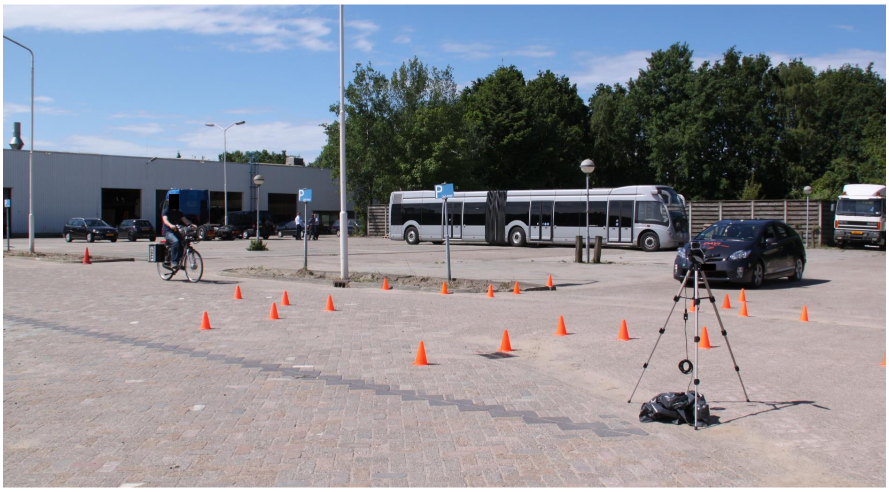


Figure 4-2: Intersection used for Data collection

Signal Measured for Car	Unit	Frequency
Longitudinal Speed	m/s	100 Hz
Lateral Speed	m/s	100 Hz
Longitudinal Acceleration	m/s^2	100 Hz
Lateral Acceleration	m/s^2	100 Hz
Yaw Rate	deg/s	100 Hz
GPS longitude	deg	100 Hz
GPS latitude	deg	100 Hz

Table 4-1: Important signals measure for the car

4-2-1 Selection of Dataset

If any of the signals mentioned in Table 4-1 and 4-2 are not present, the data was not taken into account. The main challenge was for the GPS signal, as sometimes the data didn't match the location as shown in Figure 4-1.

4-2-2 Finding a reference point

As the data was recorded on two different platforms, a common reference point was needed to couple the dataset of the bicycle and the vehicle. The trajectory of each road user was plotted and compared with the video recording. As the lanes were quite narrow, both road users were driving approximately in the center of their lanes. Hence, the center of the intersection was assumed to be the point of intersection $(x = 0, y = 0)$ of the trajectories as shown in Figure 4-1. Therefore, we assume that the point of collision would be at the center of the intersection.

The GPS coordinates of each road user are converted in the XY plane and are eventually used to define the distance to the center of the intersection as follows:

$$dti = \sqrt{(X^2 + Y^2)} \tag{4-1}$$

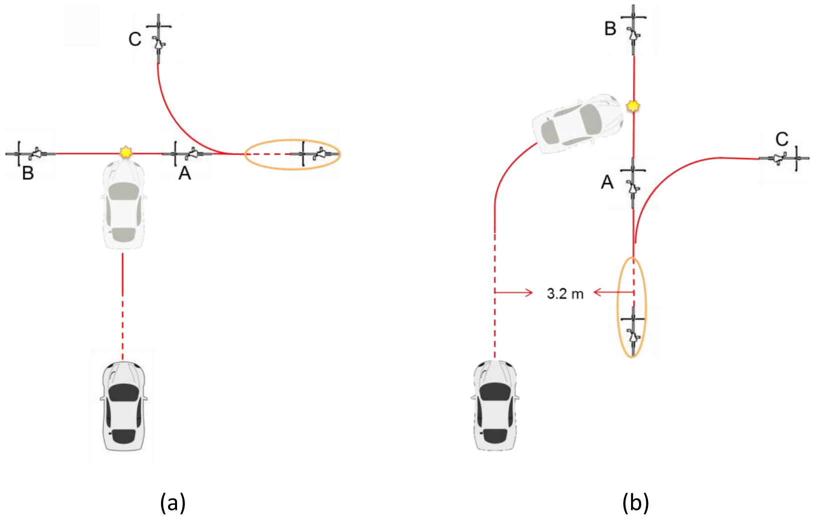


Figure 4-3: Scenarios selected for data collection: (a) Vehicle-cyclist crossing scenario (b) Vehicle-cyclist turning right scenario

[43]

Signal Measured for Bike	Unit	Frequency
Longitudinal Speed	m/s	100 Hz
Lateral Speed	m/s	100 Hz
Longitudinal Acceleration	m/s^2	100 Hz
Lateral Acceleration	m/s^2	100 Hz
Roll Angle	deg	100 Hz
Steering Angle	deg	100 Hz
GPS longitude	deg	100 Hz
GPS latitude	deg	100 Hz

Table 4-2: Important signals measure for the bike

4-2-3 Synchronizing the vehicle and bicycle data

Both recording platforms were time-synchronized, as a measure was required to be able to determine the relative distance to the intersection and signals of each road user. Time-synchronization means that the clock on each platform was set at the same time. As each platform had its start and stop feature, each recording was started manually. Hence, even if one recording was started later than the other, it would still be possible to match the signals with the help of the time synchronization.

After matching the time of each dataset, the common portion of the signals was extracted. The trajectory for each road user was plotted and matched with the video recording as shown in Figure 4-4. Hence, the animated plot helps in checking that the synchronization was performed correctly and as a reference to check the consistency of the actions of the cyclist and driver (kinematic data) with the video data. As distance and time to the intersection are important parameters to analyze the performance of the model in this study, the whole signal was shifted in a way that $t = 0$ corresponds to the time when the road user is at the center of the intersection. In this way, $t < 0$ and $t > 0$ correspond to the bicycle or vehicle approaching and leaving the intersection respectively.

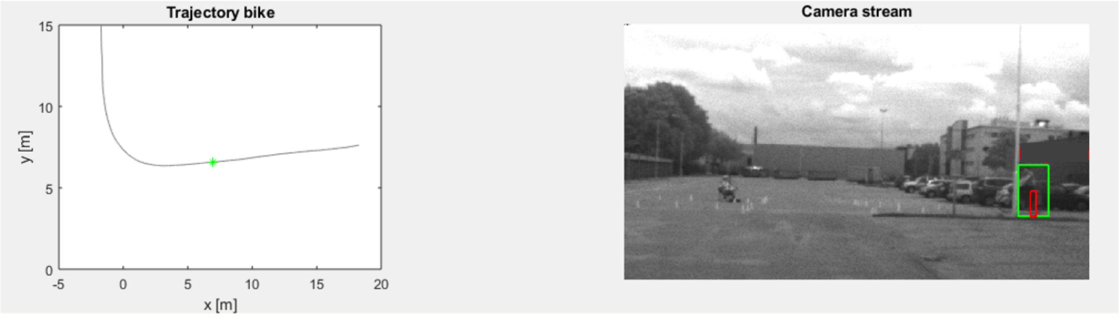


Figure 4-4: Matching the data with the video recording

4-2-4 Data Selection

After the processes described above are performed, the data still contains segments which are not interesting for this study, for example, when the bike is getting up to speed. These segments are found usually at the start and end of the dataset as shown in Figure 4-5 As the interaction is the main focus of this study, the initial segment (with speed less than the maximum speed, $V_{max,cyclist}$) of the cyclist data can be neglected. Considering the fact that one road user will reach the intersection earlier, the last portion was discarded once both road users have crossed and traveled a minimum distance of 7m. The conditions can be described as follows:

$$V_{initial} \geq \frac{1}{2} V_{max,cyclist} \tag{4-2}$$

$$\min\{dti_{cyclist}, dti_{vehicle}\} = 7m \tag{4-3}$$

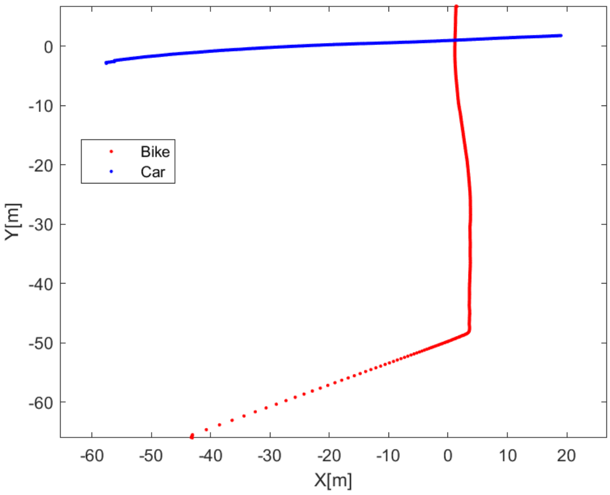


Figure 4-5: Complete vehicle and cyclist trajectories

When the conditions are implemented, the data becomes suitable for the learning and validation of the interaction model as shown in Figure 4-6. The true states of each road user were also included in the dataset, where straight, turn and stop actions were denoted by 1, 2 and 3 respectively.

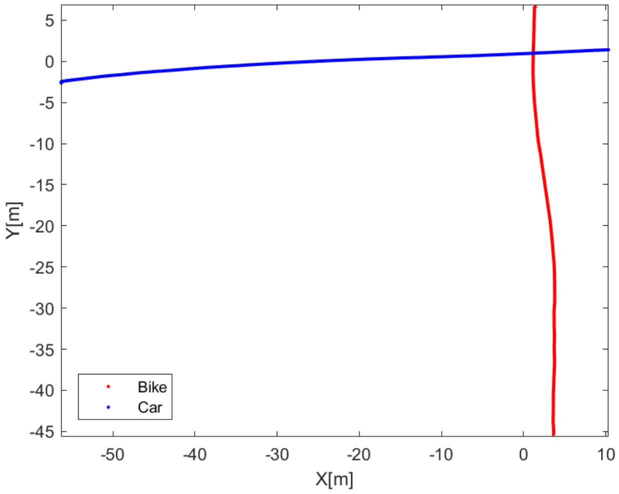


Figure 4-6: Selected part of the trajectories for determining the parameters of the interaction model

Chapter 5

Results

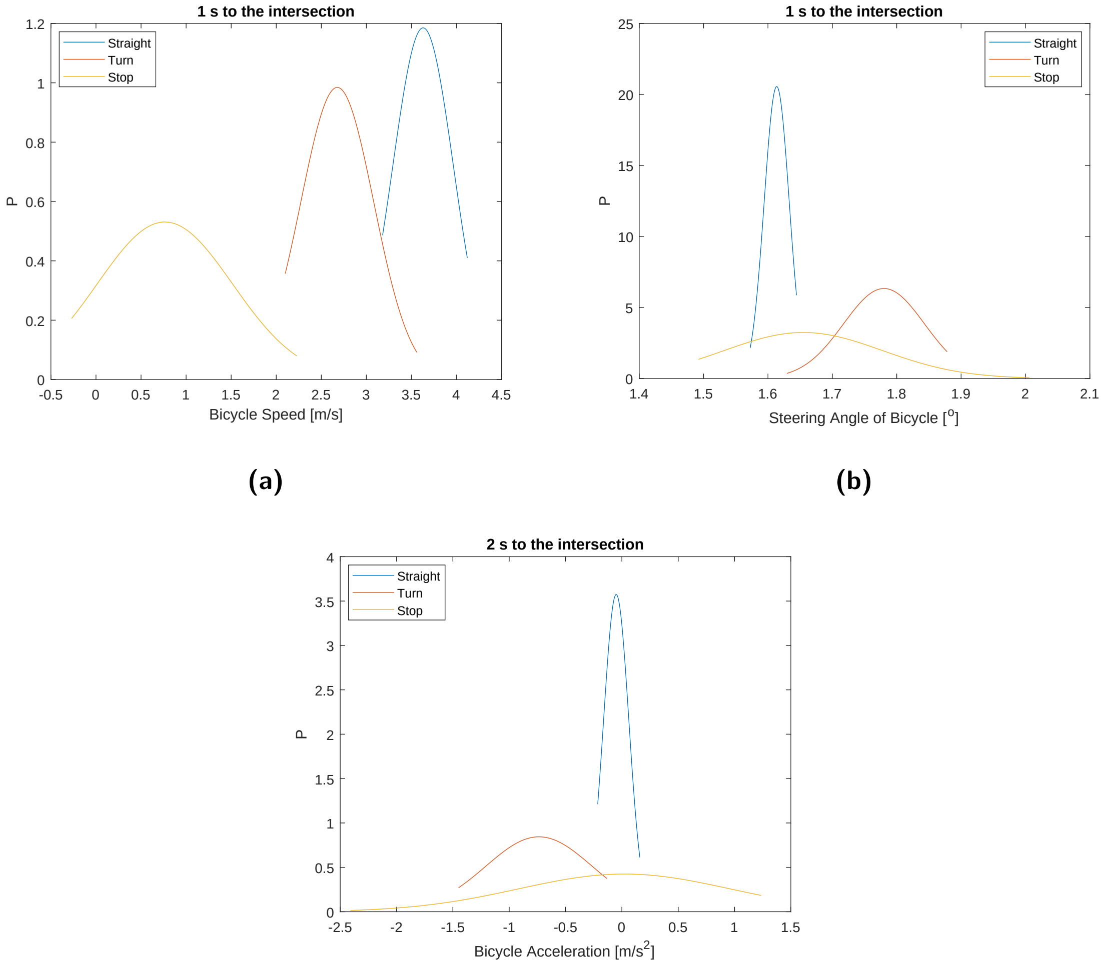
In this chapter, the results of the simulations are presented and discussed. Based on the architecture of the model described in Section 3-1, the final parameters for the prediction of the interaction between cyclist and car including their maneuvers are selected. The parameter selection is based on a sensitivity analysis. Next, the method of validation was selected based on the available datasets and implemented to generate the results.

5-1 Selection of parameters for the vehicle and cyclist intent prediction

Before training and cross-validation, the right combination of parameters from the architecture as described in Section 3-1 was selected. This was performed by means of a method applied earlier at TNO for the selection of parameters for cyclist maneuver prediction [21].

Three different windows were selected for both the cyclist and vehicle, namely $1s$, $2s$, $4s$ to the center of the intersection. At each instance, the bicycle (speed, acceleration, and steering angle) and vehicle (speed, acceleration, and yaw rate) parameters were fitted to a normal distribution. The normal distribution of the bicycle parameters for 1 s, 2 s, and 4 s to the intersection are shown in Figures A-1 to A-3, Figures A-4 to A-6, and Figures A-7 to A-9 respectively. The normal distribution of the vehicle parameters for 1 s, 2 s, and 4 s to the intersection are shown in Figures A-10 to A-12, Figures A-13 to A-15, and Figures A-16 to A-18 respectively. Figures 5-1 and 5-2 show the most important plots which will be used for the sensitivity analysis of the bicycle and vehicle parameters respectively.

In Figure 5-1, the cyclist speed, and acceleration show the least overlap in the normal distribution curves of all maneuvers and hence it can be concluded that these are the most important parameters for the cyclist intent recognition. Referring to Figure 5-1a, the normal distribution curves for the straight and turn maneuvers have no overlap while the turn and



(a)

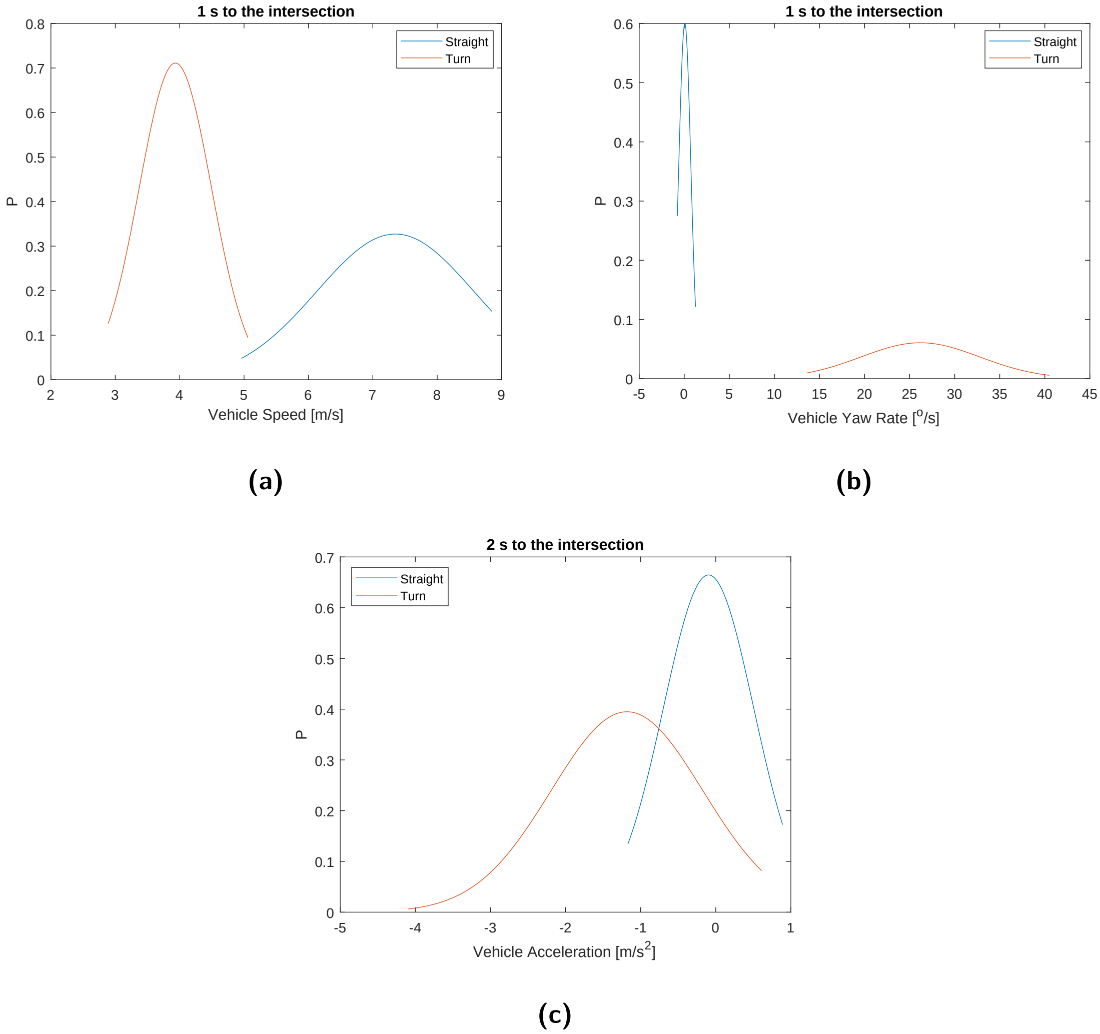
(b)

(c)

Figure 5-1: Fits of normal distributions to the (a) Speed (b) Steering angle (c) Acceleration of the bicycle

stop maneuvers still have some overlap. Based on the study [21], only steering angle and speed were selected initially. However, the results from the parameter sensitivity analysis encourage the use of acceleration. On incorporating acceleration as an additional input parameter of the interaction model, it was seen that the performance of the model improved with a maximum difference of 27.58 % as shown in Tables A-25 and A-26.

Applying the sensitivity analysis to the vehicle's speed, yaw rate, and acceleration (see Figure 5-2), it was concluded that yaw rate and speed are parameters that can provide a good prediction. In addition, the performance was checked with a combination of yaw rate, speed, and acceleration, and incorporating acceleration improved the prediction by a maximum of 32.76% as shown in Tables A-25 and A-26. Hence, this combination was used for checking the influence of the interaction parameters as described in Section 5-4.



(a) **(b)**

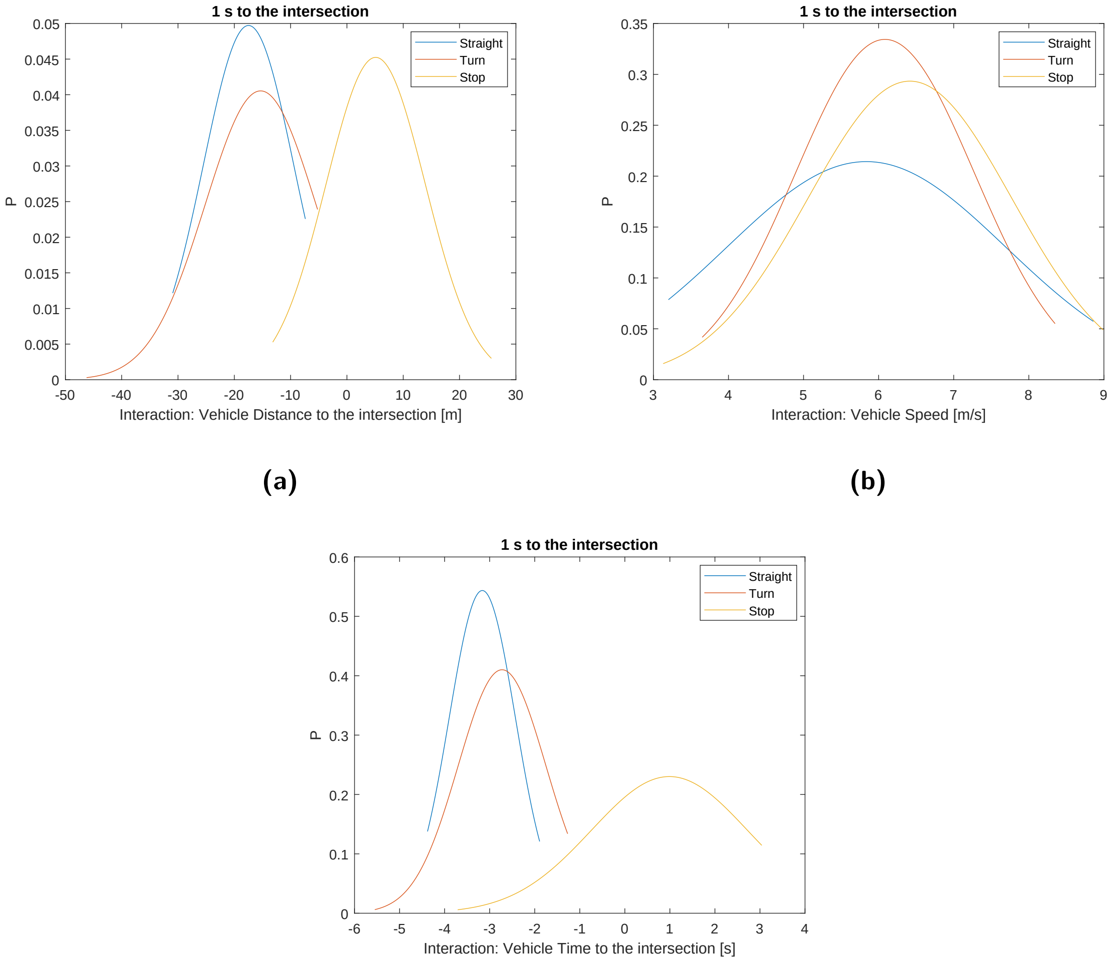
(c)

Figure 5-2: Fits of normal distributions to the (a) Speed (b) Yaw Rate (c) Acceleration of the vehicle

5-2 Selection of interaction parameters

As explained in 3-1, the interaction parameters used in the intention prediction of the cyclist and the vehicle are the distance to the intersection of the vehicle and cyclist respectively. After the development of the architecture, other interaction parameters could also be used, such as speed and time to the intersection. This section focuses on comparing the different interaction parameters and choose the best combination for an optimum prediction. Similar to Section 5-1, windows of $1s$, $2s$, $4s$ to the center of the intersection will be used for the comparison. The normal distribution of the interaction parameters for the vehicle and bicycle are shown in Figures B-1 to B-9 and Figures B-10 to B-18 respectively.

Figure 5-3 and 5-4 show the interaction parameters used for the intent prediction of the cyclist and vehicle respectively. Comparing the plots in Figure 5-3, it can be seen that the results of vehicle time to the intersection (TTI) and the distance to the intersection. Also,



(a)

(b)

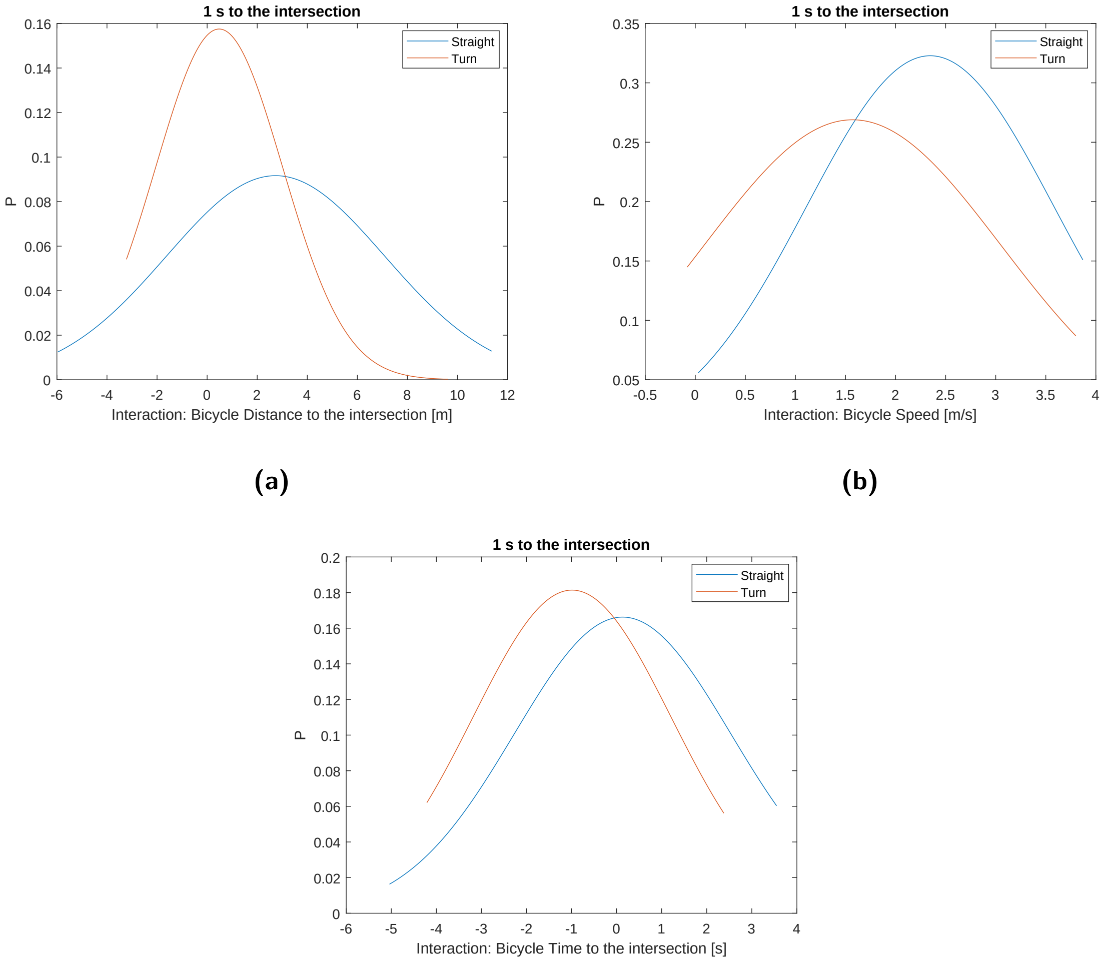
(c)

Figure 5-3: Fits of normal distributions to the interaction parameters of the bicycle: (a) distance to the intersection (b) speed (c) time to the intersection (TTI) of the vehicle

the parameters are quite similar, as they only differ by a factor (i.e. the speed). However, Figure 5-4 shows that only bicycle distance to the intersection has the least overlap for the maneuvers compared to the other interaction parameters. Hence, distance to the intersection was selected for predicting the intention of both the cyclist and vehicle.

5-3 Selection of Validation Method

The standard method of validation is not selected as it requires a large number of datasets are available, which is not available in this study (a total of 61 maneuvers). Hence, cross-validation is preferred over the standard method. Three different maneuvers were always used to start the learning of the Hidden Markov Model (HMM) and therefore were not considered for calculating the overall accuracy of the prediction. These maneuvers included at least



(a)

(b)

(c)

Figure 5-4: Fits of normal distributions to the interaction parameters of the vehicle: (a) distance to the intersection (b) speed (c) time to the intersection (TTI) of the bicycle

a stop, straight, and turn for the cyclist, and at least a straight and turn for the vehicle. Accordingly, 1 prediction was given with the HMM trained on the rest 57 maneuvers. This training and prediction process was repeated for 58 maneuvers and the overall accuracy of the model was calculated.

5-4 Cross Validation

After selecting the parameters of the interaction model, its performance is evaluated. The concept of confusion matrix is applied for checking the overall accuracy. Table 5-1 shows an example of a confusion matrix. The main diagonal of the confusion matrix represents the true positives while the other values represent the false positives. The overall accuracy of the prediction is derived as follows:

$$\text{Overall Accuracy} = \frac{\text{Number of True Positives}}{\text{Number of True Positive} + \text{Number of False Positives}} \tag{5-1}$$

Figure 5-5 shows the final results of applying Equation 5-1 on Tables C-1 till C-24. To assess the performance further, the cyclist's intent prediction of the interaction model was plotted from the time instance when the cyclist was at 15 m from the intersection until he reaches the center of the intersection ($t_{cyclist} = 0$). To plot the vehicle's intent prediction, the time instance of the vehicle corresponding to the cyclist being 15 m from the intersection, was again considered until the time the vehicle reaches the center of the intersection ($t_{vehicle} = 0$). This procedure was repeated both for the HMM with and without the interaction parameters. An example of each state of the cyclist and the vehicle (with and without interaction) are shown in Figures 5-6 to 5-15.

Actual Cyclist Maneuvers at TTI= 0.5 s	**Predicted Cyclist Maneuvers (Interaction) at TTI = 0.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	14	0	1
Right Turn	2	18	0
Stop	0	0	23

Accuracy = 94.83%

Table 5-1: An example of the Confusion Matrix

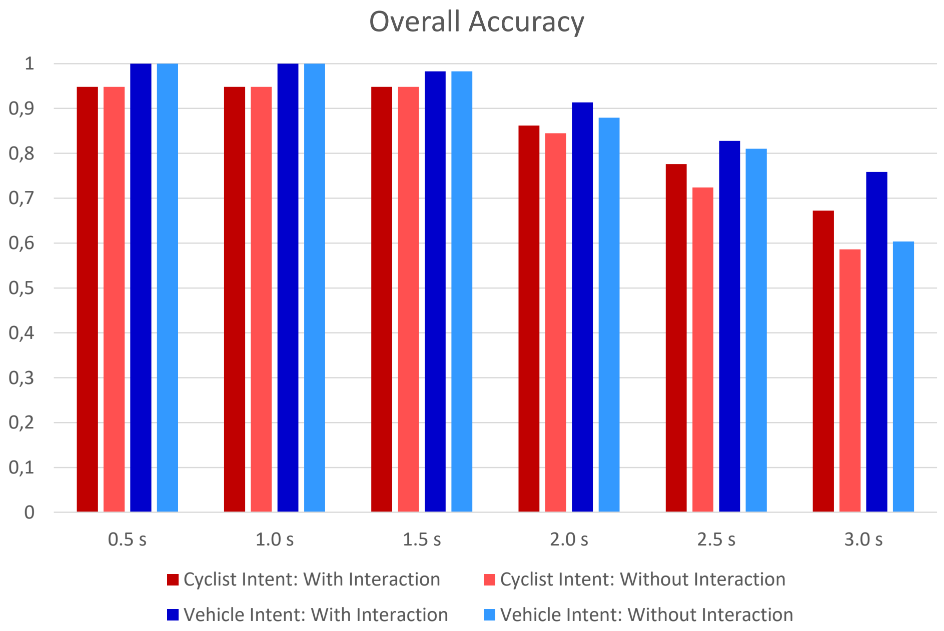


Figure 5-5: Overall Accuracy of the interaction model

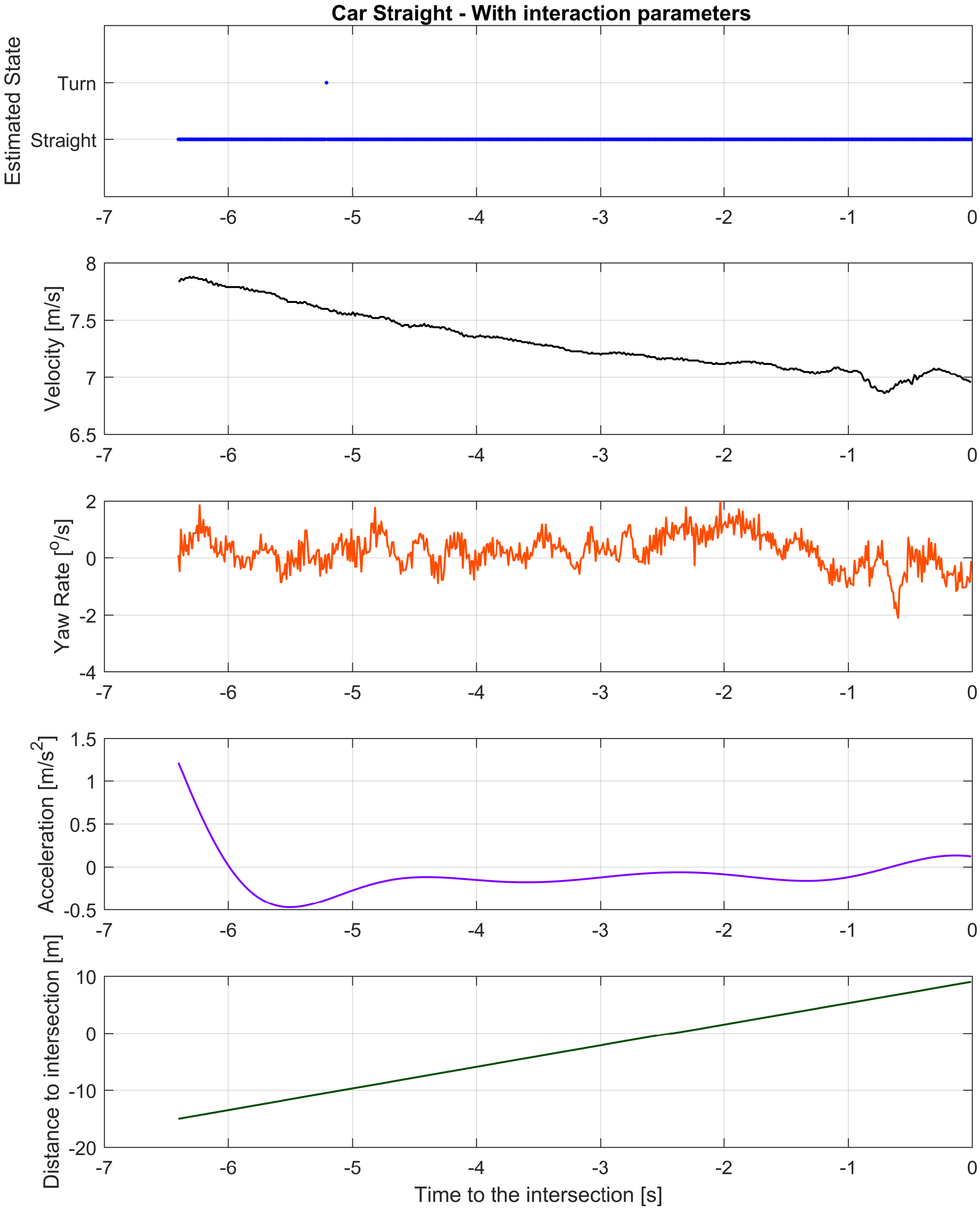


Figure 5-6: Example of intent recognition for vehicle going straight using interaction parameters

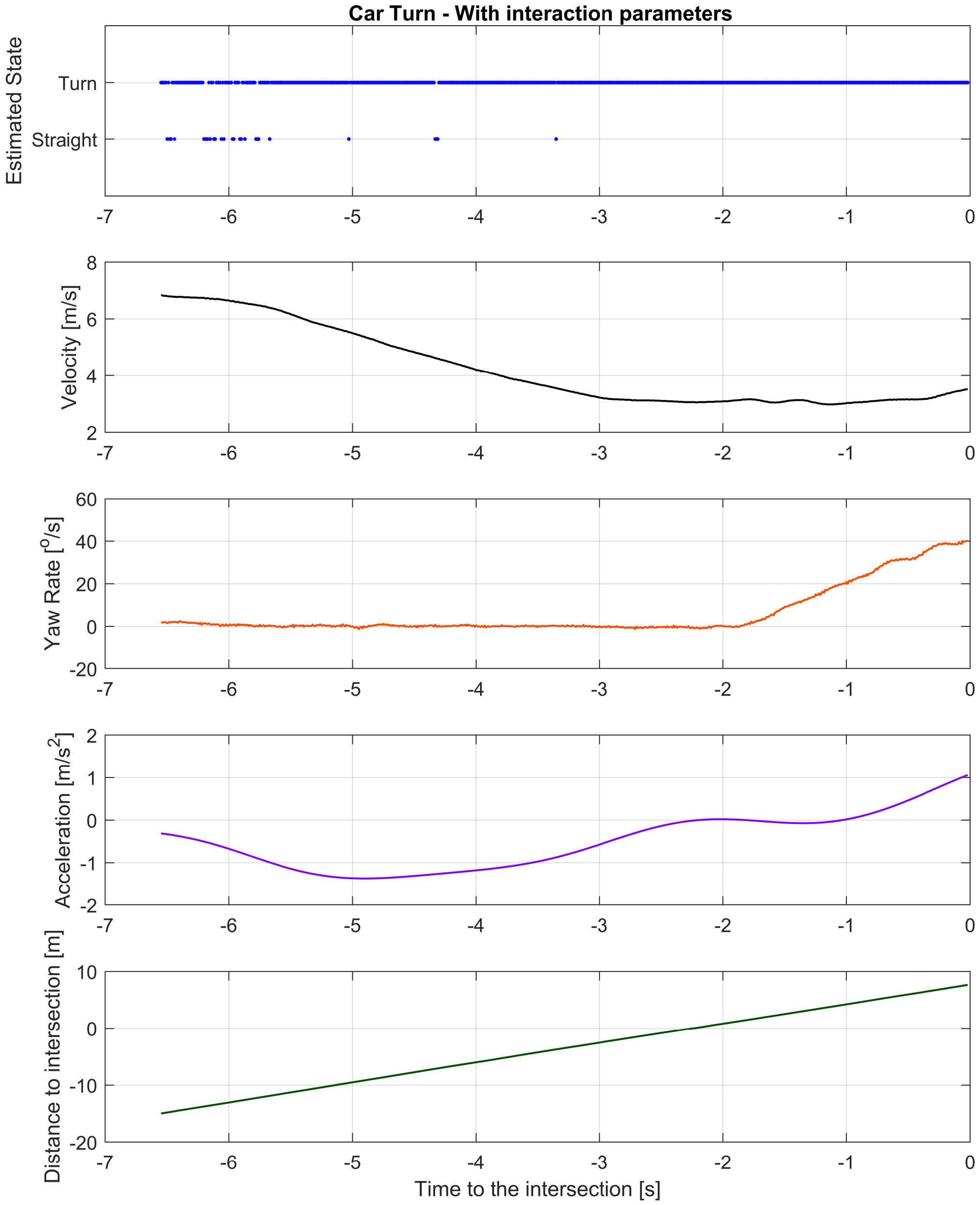


Figure 5-7: Example of intent recognition for vehicle taking a right turn using interaction parameters

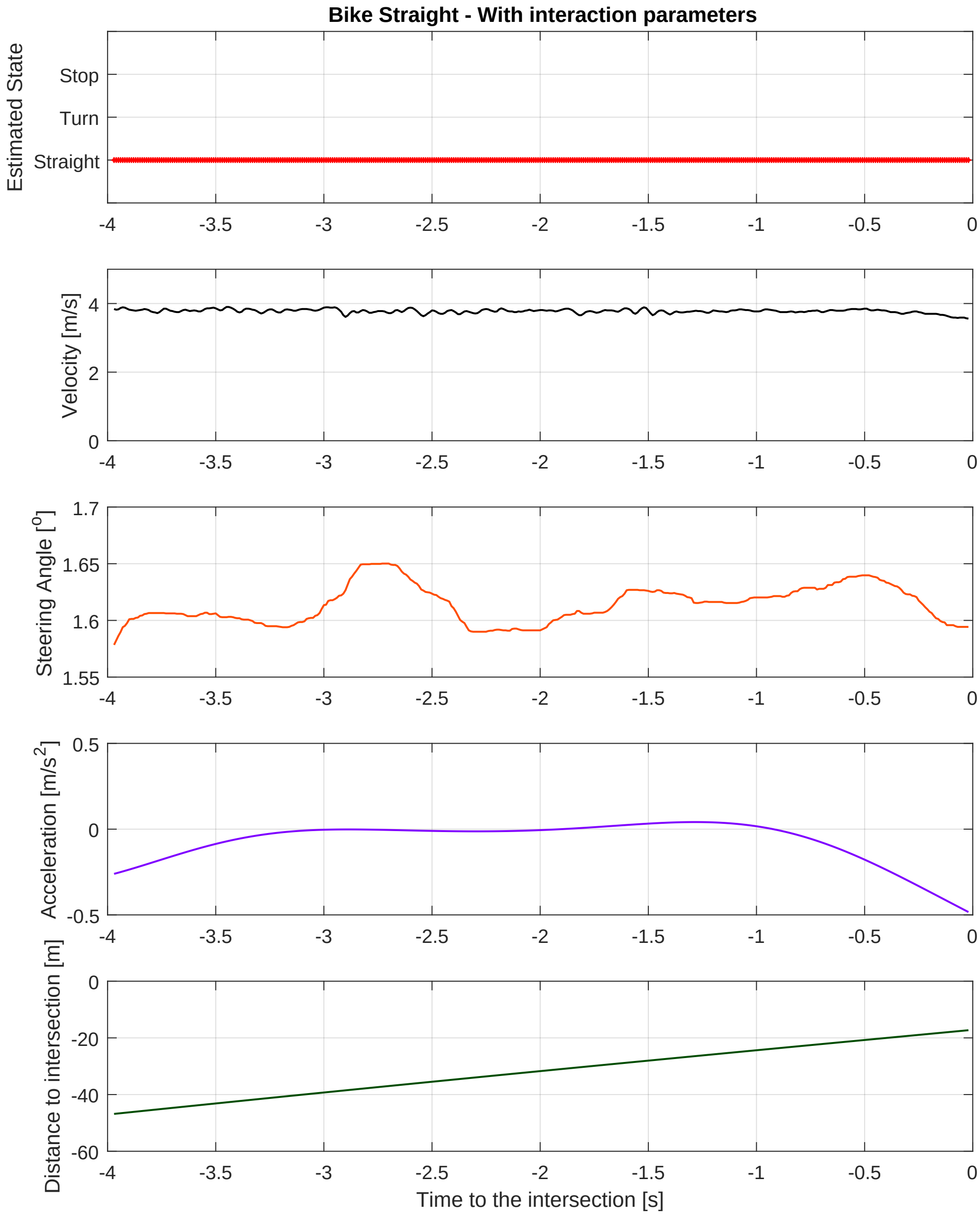


Figure 5-8: Example of intent recognition for cyclist going straight using interaction parameters

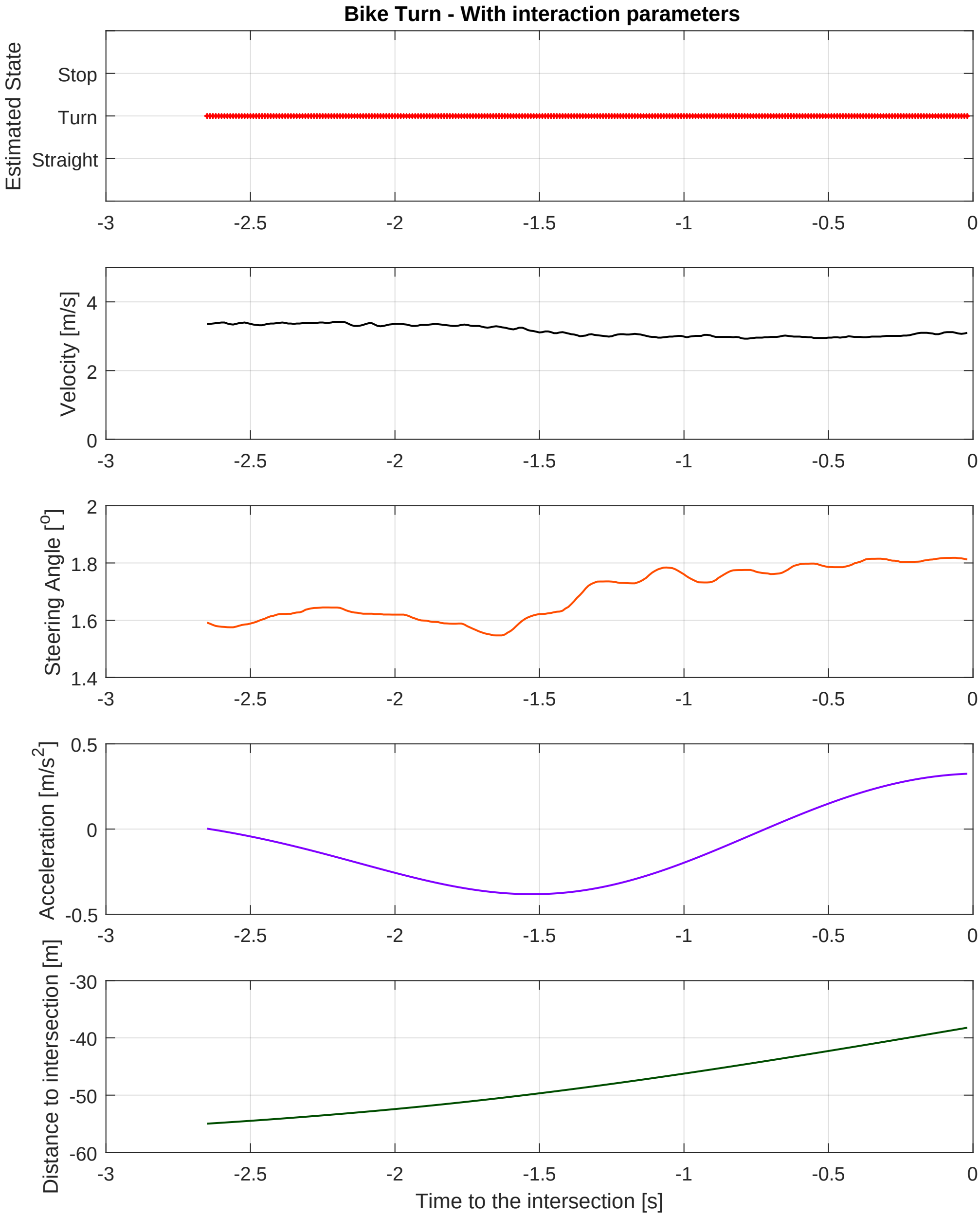


Figure 5-9: Example of intent recognition for cyclist taking a right turn using interaction parameters

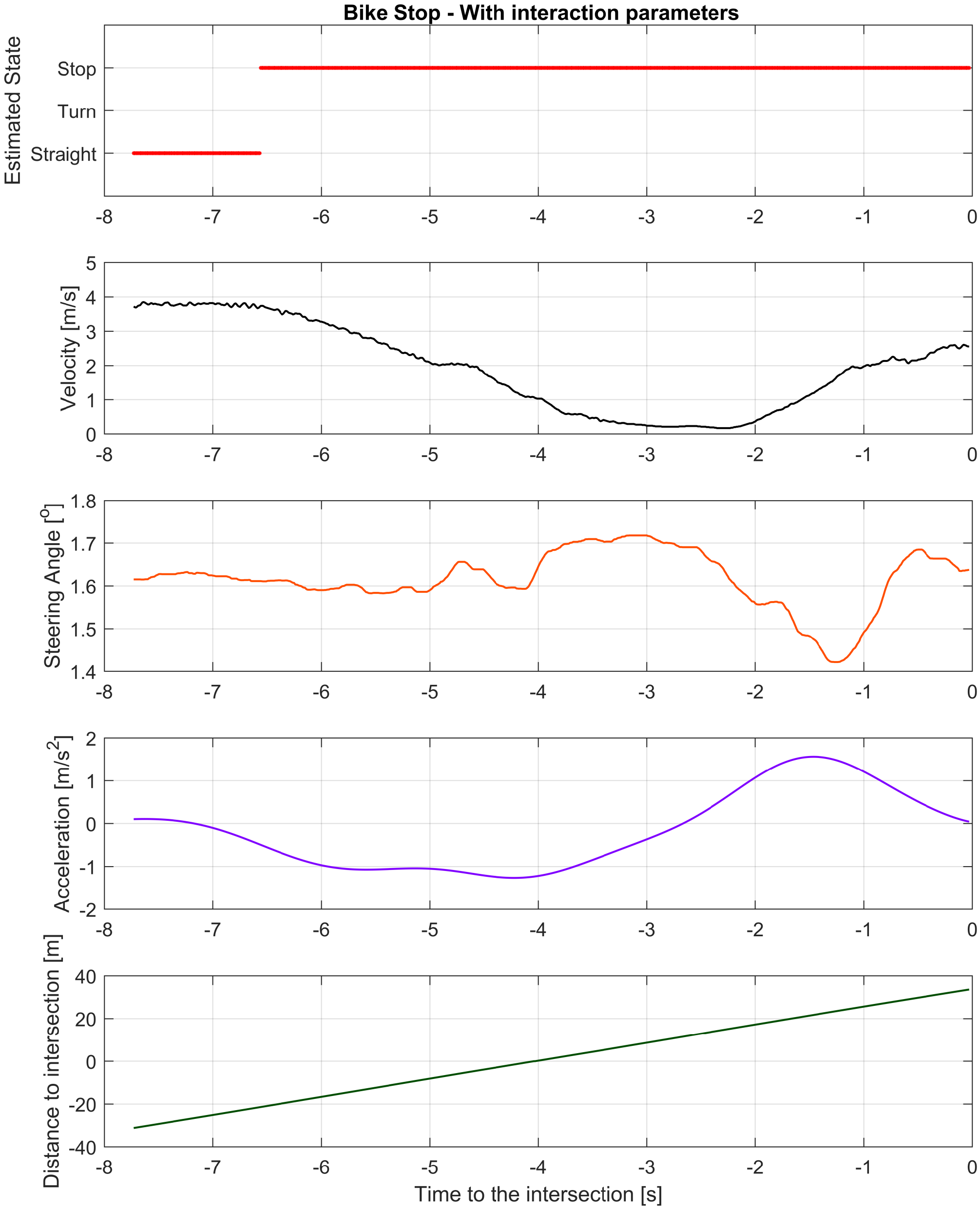


Figure 5-10: Example of intent recognition for cyclist stopping using interaction parameters

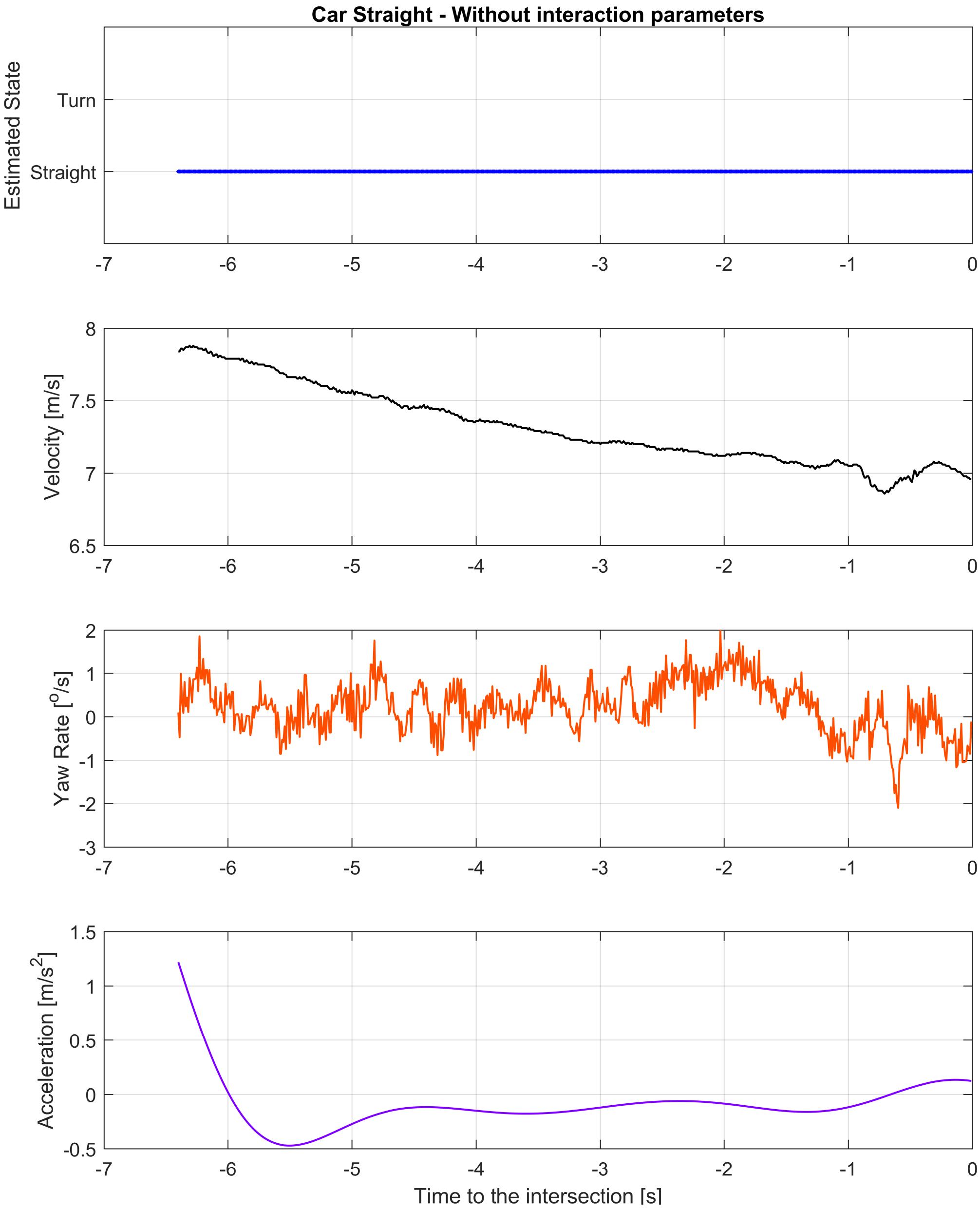


Figure 5-11: Example of intent recognition for vehicle going straight without interaction parameters

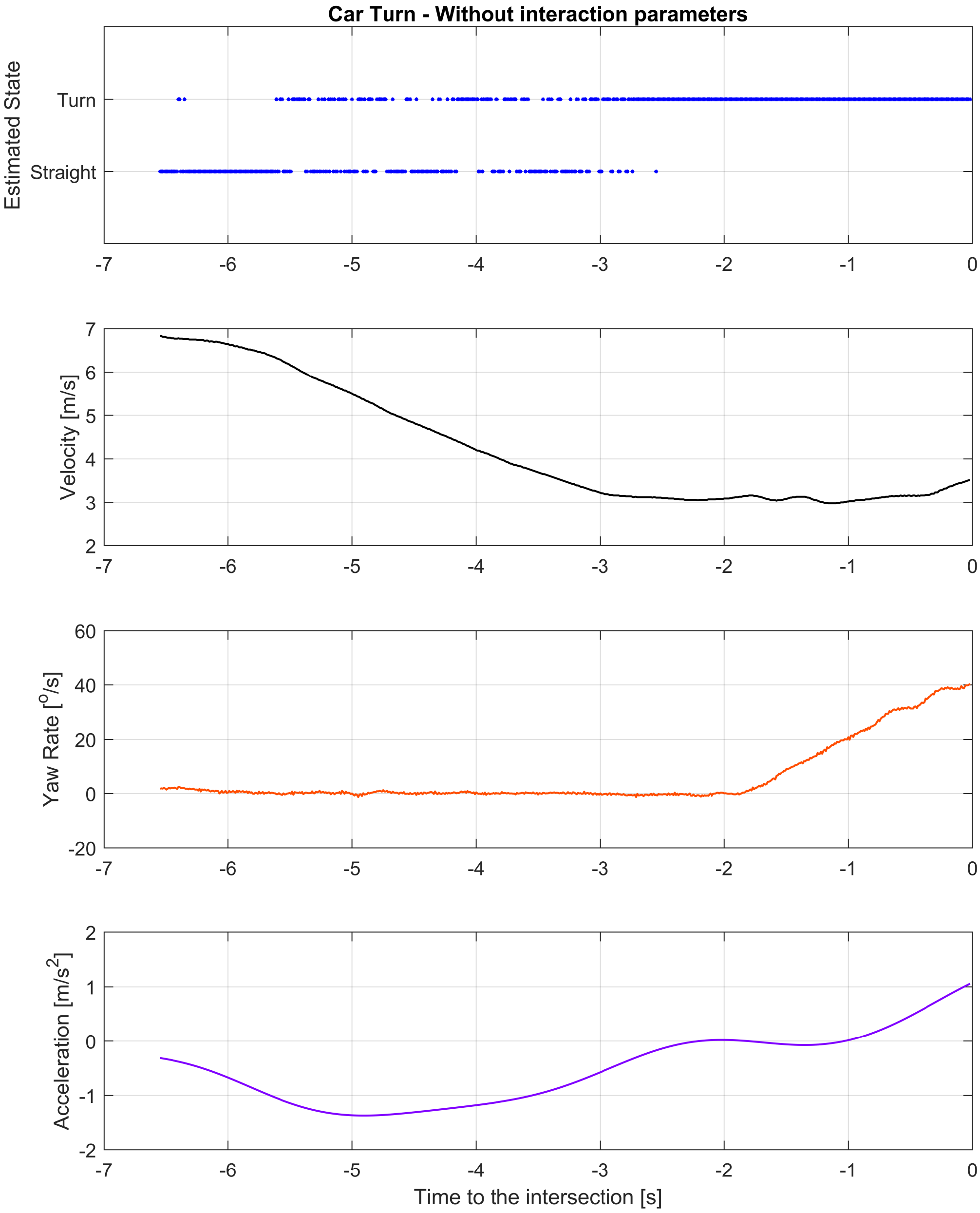


Figure 5-12: Example of intent recognition for vehicle taking a right turn without interaction parameters

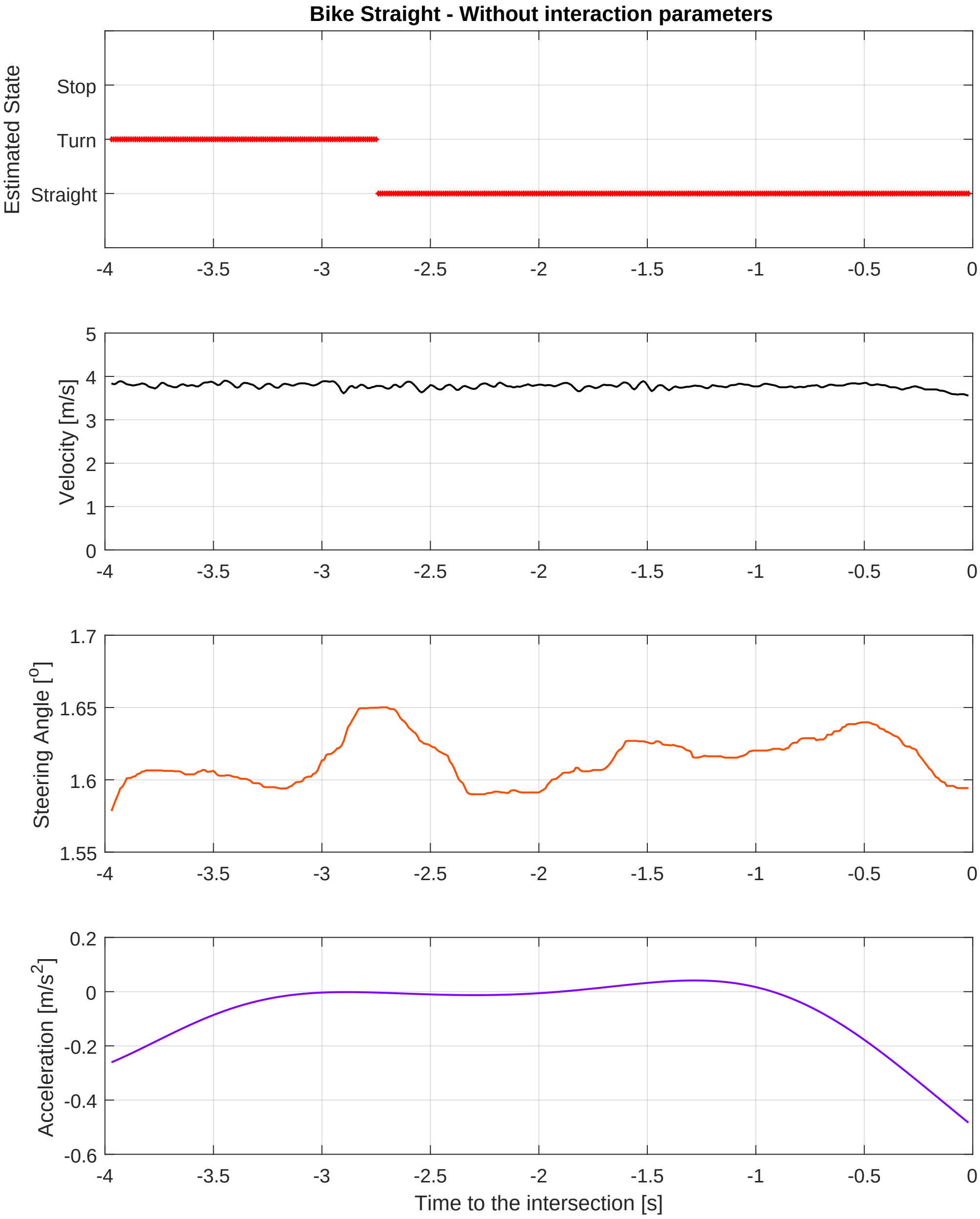


Figure 5-13: Example of intent recognition for cyclist going straight without interaction parameters

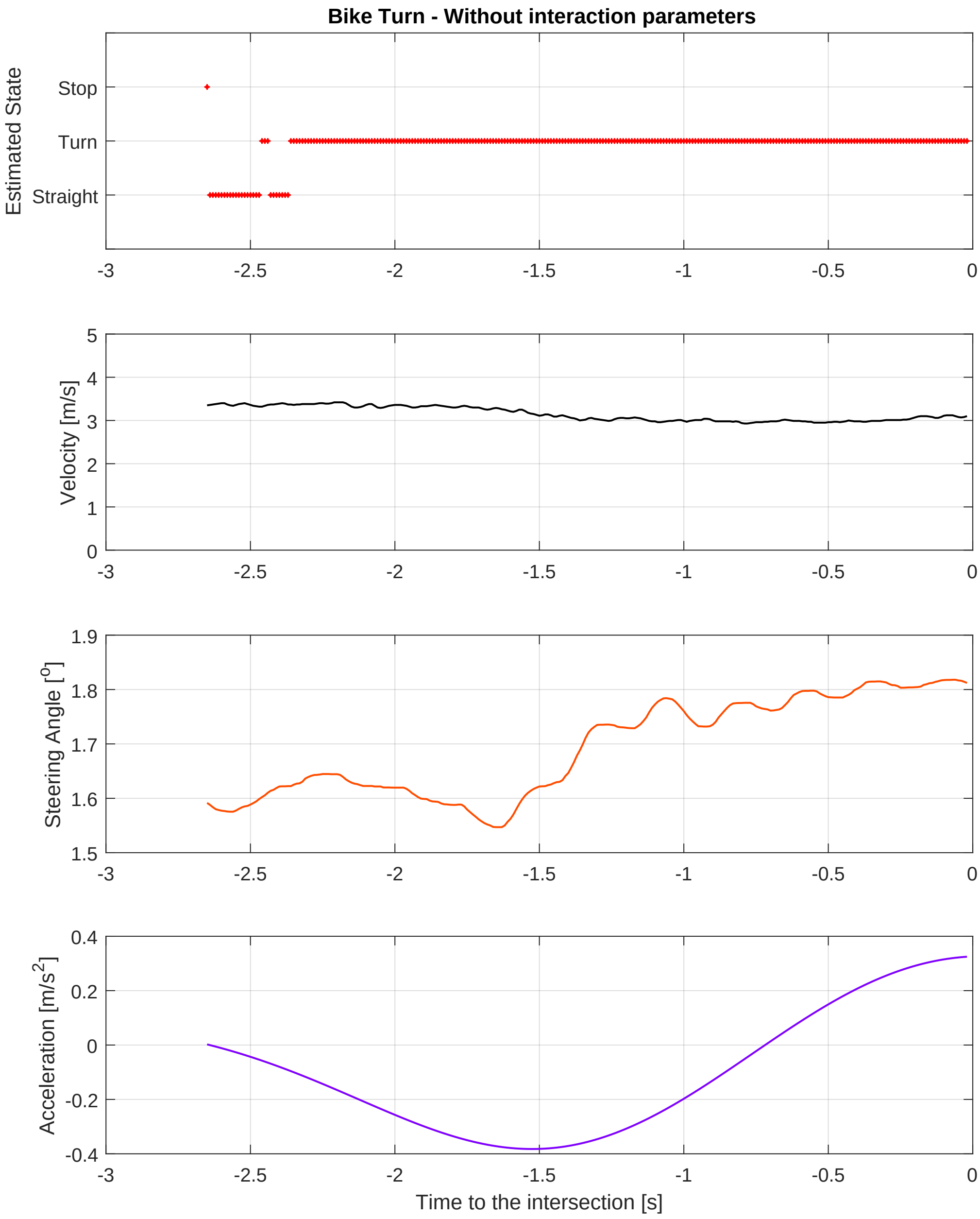


Figure 5-14: Example of intent recognition for cyclist taking a right turn without interaction parameters

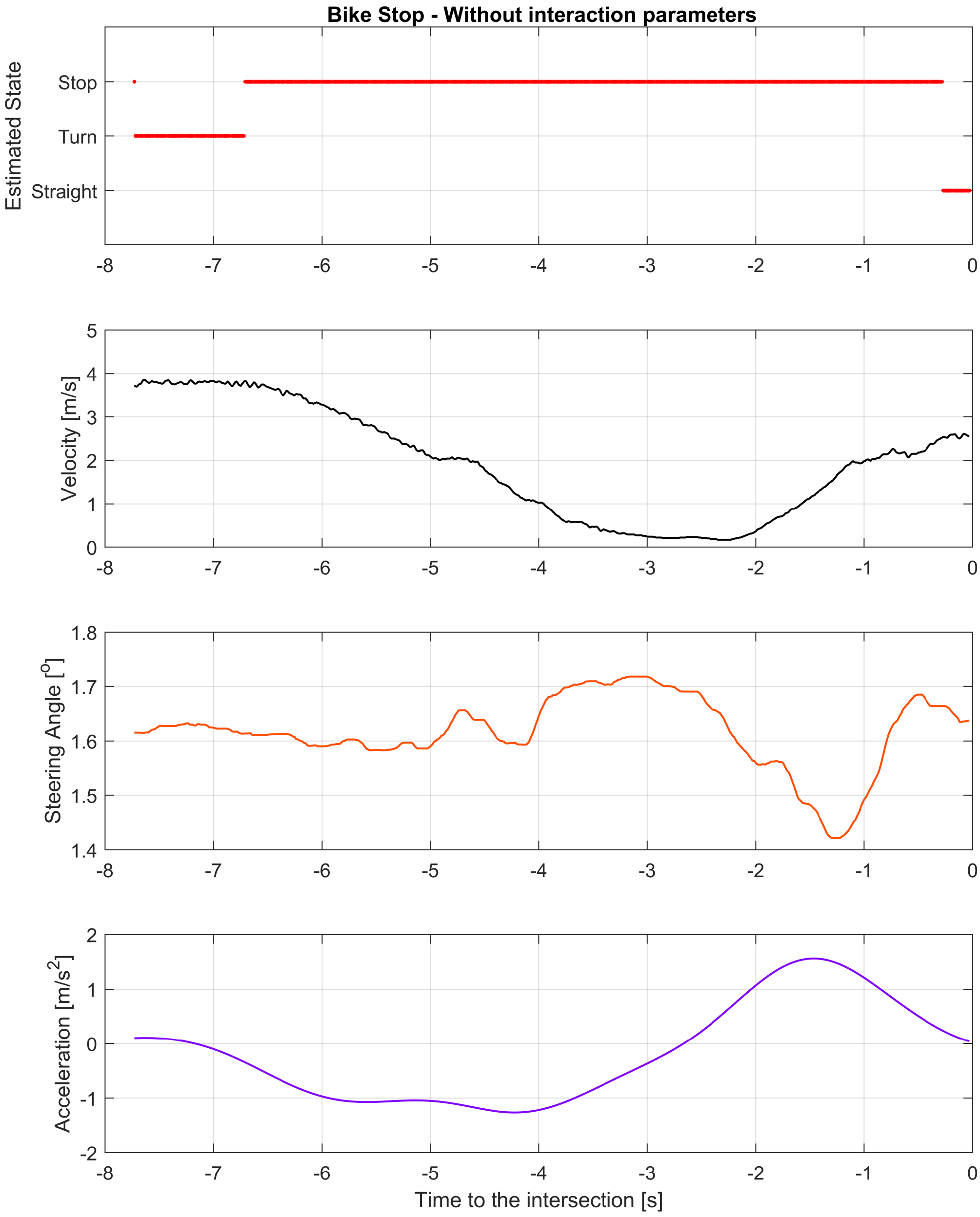


Figure 5-15: Example of intent recognition for cyclist stopping without interaction parameters

5-5 Discussion of results

The method of cross-validation was selected based on the limited amount of data available for this study. Furthermore, this validation method allows the use of unbalanced datasets as the number of recorded maneuvers are not equal. Also, cross-validation gives a better insight into the performance of machine learning methods with a limited amount of data.

It was concluded that the selection of parameters (including the interaction parameters) plays an important role in improving the overall performance of the HMM. Applying the method described in [21] lead to the results that steering angle, speed, and acceleration of the bicycle are the best parameters for the intent recognition of a cyclist. Next, speed, yaw rate, and acceleration are the best parameters for recognizing a driver's intention near an intersection. When a driver and a cyclist interact at an intersection, each road user estimates how far and how fast does the other road user approach the intersection. In this way, the road users influence each other's decision to stop or to go on.

Table 5-2 summarizes the difference in performance of the HMM with and without interaction parameters. The addition of the interaction parameters in predicting a driver's or a cyclist's decisions further improved the performance for a prediction of more than $2s$. On comparing Figures 5-7 and 5-12, it can be concluded that the interaction parameters help to have a better and more constant prediction. In real life, if a road user has an indication about the decisions of the other road users then, this could help him in making his decision. In a similar way, if an early prediction (>2s) is required, then having the interaction parameter helps to have a better prediction.

Time to the intersection [s]	Type of Road User					
	Vehicle			Cyclist		
	HMM with interaction [%]	HMM without interaction [%]	Overall Performance [%]	HMM with interaction [%]	HMM without interaction [%]	Overall Performance [%]
0.5	100	100	0	94.83	94.83	0
1.0	100	100	0	94.83	94.83	0
1.5	98.28	98.28	0	94.83	94.83	0
2.0	91.38	87.93	3.45	86.21	84.48	1.73
2.5	82.76	81.03	1.73	77.59	72.41	5.18
3.0	75.86	60.34	15.52	67.24	58.62	8.62

Table 5-2: Comparison of the HMM with and without interaction parameters

Chapter 6

Discussion, Conclusions, and Recommendations for Model Improvements

6-1 Discussion

The number of accidents involving Vulnerable Road Users (VRU) is increasing each year and introduction of Advanced Driver Assistance Systems (ADAS) functionalities are small steps towards safer roads. The objective of this MSc project was to develop a vehicle-cyclist interaction model for the generation of realistic variations on vehicle-cyclist crossing scenarios.

A literature study was first conducted in order to find the modeling methods for describing a driver's and a cyclist's behavior, which could then be used in the development of a vehicle-cyclist interaction model. Based on the Literature Study, the architecture of the interaction model was proposed as shown in Figure 3-1. It was also seen that extensive research has been performed on modeling a driver behavior and limited on modeling a cyclist's behavior. Hidden Markov Model (HMM) was the method most often used for modeling the driver's behavior, and hence, was selected with the same motivation for the development of the interaction model.

The Literature study also gave an indication of what parameters could be used for the intent recognition of the driver and the cyclist independent of each other. Based on how a road user takes a decision at an intersection, there are many possibilities for the interaction parameters. These include distance to the center of the intersection, the speed, or time to the crossing of the other road user approaching the intersection. Sensitivity analysis was used to select the right combination of parameters to avoid having too many parameters which would result in high computational cost and overfitting of the model.

As the interaction model requires time-synchronized data both from the vehicle and the bicycle, the choice for a suitable dataset was quite limited. Experiments were conducted at

TNO in a controlled environment where data was collected from a vehicle interacting with a cyclist at an intersection. The data was time-synchronized, however, the data was not ideal since the tests were not naturalistic. Also, for safety reasons, no critical encounters between vehicle and cyclists could be made. Furthermore, the number of scenarios collected was less than the actual scenarios possible at an intersection, and the number of collected vehicle-cyclist encounters was also limited. Hence, the available data restricted the study on the possible states of each participant and the model still requires more data to be able to test for all real situations at an intersection.

As a limited amount of data was available, cross validation was preferred over the most commonly used validation method (dividing the data in a 7:3 ratio for training and testing respectively). In this case, cross validation allows a better evaluation of the performance as each time a different dataset is selected for prediction. Nevertheless, using cross validation requires more computational cost as a new HMM is trained with each iteration. Compared to the cross validation, the most commonly used validation method involves training HMM once and testing it for 30% of the data, which means less computational time in total.

The simulation results of this research are very promising despite the limited data set. The selection of the parameters can already be applied for the development and assessment of ADAS functionalities, whereas the interaction model needs more data to be further optimized for application in the scenario generation method. The current interaction model is limited as the maneuvers of each participant is less than the possible maneuvers at an intersection (i.e. go straight, stop, turn right or turn left). For the complete training and validation, the interaction model requires naturalistic data collected at a real unregulated crossing. Furthermore, the participants must not be instructed to stop for each other. For an ideal dataset, the participants should cycle or drive as they would do in everyday life, however for safety reasons, they should follow the traffic rules that apply.

6-2 Conclusions

In this thesis, a vehicle-cyclist interaction model was developed for the generation of scenarios for virtual testing of ADAS or for validation of randomly generated scenarios. Also, the best parameters were determined by means of using experiments performed with a vehicle and cyclists. From the preceding Literature Study, an initial architecture was proposed for a vehicle-cyclist interaction model as shown in Figure 3-1.

A sensitivity analysis as described in [21] was applied to the available dataset of vehicle and cyclist kinematic data, in order to select potential parameters for the cyclist, driver as well as their interaction. It was concluded that speed, acceleration, and steering angle were the best parameters for the cyclist, while speed, acceleration, and yaw rate were the best parameters for the driver. On comparing the interaction parameters for the cyclist, it was concluded that both distance and time to the intersection (TTI) gave similar results as they differ by a factor, namely the velocity. But for the vehicle, it was concluded that distance to the intersection is a better interaction parameter as the normal distribution curves had the least overlap compared to the other parameters. Hence, distance to the crossing was selected as the interaction parameter to maintain a uniformity between the cyclist and the vehicle. Using the selected parameters for predicting the cyclist's and driver's intent, a final conclusion is drawn on the architecture of the interaction model, as shown in Figure 6-1.

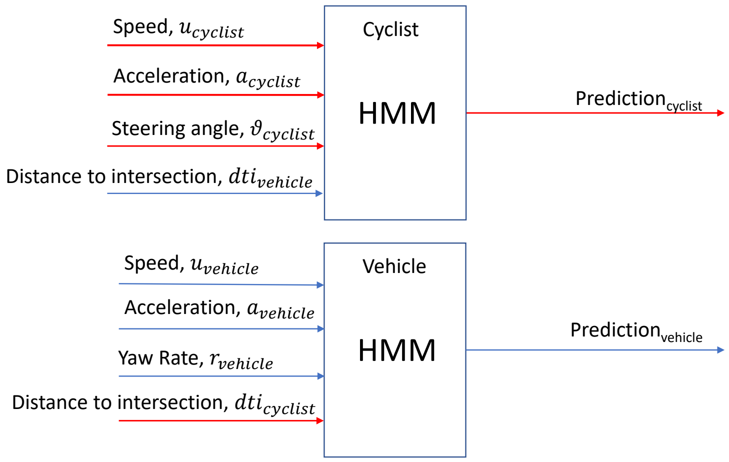

Speed, $u_{cyclist}$
Acceleration, $a_{cyclist}$
Steering angle, $\vartheta_{cyclist}$
Distance to intersection, $dti_{vehicle}$
Cyclist
HMM
Prediction$_{cyclist}$
Speed, $u_{vehicle}$
Acceleration, $a_{vehicle}$
Yaw Rate, $r_{vehicle}$
Distance to intersection, $dti_{cyclist}$
Vehicle
HMM
Prediction$_{vehicle}$

Figure 6-1: Final architecture of the interaction model.

Using the final architecture and the parameters, the interaction model was developed, trained and validated. Using the Confusion Matrix, it was seen that including the interaction parameters improved the prediction for more than $2s$ with a minimum and maximum difference of 1.73 and 15.52% respectively. Figures 5-6 to 5-15 also showed that the interaction model has a more stable performance. From this research, we can conclude that using the interaction parameters improves the performance for an early prediction (>2s) as well as reduces the fluctuations between different states.

6-3 Recommendations for model improvements

Since this study was a first attempt to model the vehicle-cyclist interaction for the generation of scenarios for virtual testing of ADAS, several recommendations for further development of this model are described below.

The current model is limited due to the data that was used for the training and validation. Additional data is needed in order to have all the possible states at an intersection, namely going straight, stopping, taking a right or left turn. Also, it is recommended to collect the data at a real intersection to have a naturalistic representation of traffic and interaction. Furthermore, it would be very efficient for data analysis if the data collection could be done on one platform which receives data from the vehicle and bicycle instead of using two different platforms which needed to be time-synchronized.

The model excludes input from the camera and it could be interesting to couple it. In this way, hand gestures or head movements of the cyclist could be used to get additional input parameters for turns. Hence, the interaction model could be extended with additional parameters to simulate a vehicle-cyclist interaction using recorded data from the camera.

Finally, the vehicle-cyclist interaction model needs to be made applicable for creating scenarios for virtual testing of ADAS (for example, cyclist-AEB, driver warning for cyclists, etc.). Ideally, in future the model would be trained with data that are directly measured at the vehicle in daily life during a certain period of driving in cities with many cyclists. If the interaction model is trained using more kinematic data, more realistic variations of the scenarios can be generated.

6-4 Future Applications

As critical scenarios are more interesting for assessing the performance of ADAS functionalities, these scenarios should also be collected. But, this is not allowed when testing with volunteers. For safety reasons physical testing is performed with cyclist dummies, however, besides the unnatural kinematics, the interaction of the cyclist with the vehicle is missing. Therefore, critical vehicle-cyclist scenarios should be obtained from real-life scenarios. This could be obtained from camera measurements at a crossing as performed for the EU project InDEV [44], or from driving around with a vehicle with special instrumentation for collecting the cyclist kinematics. In this way, more critical scenarios could be generated based on the model and the critical scenarios that were measured.

Once the vehicle-cyclist interaction model is finalized, it can be used as a filter on an instrumented vehicle for filtering vehicle-cyclist scenarios that have already been recorded in the database. In this way, a storage and computational power management could be attained where only new scenarios that have statistical relevance and discard all the rest.

The model can also be used to develop convenient and safe actuation times for ADAS, like Automated Emergency Braking (AEB), or future functionalities to avoid accidents with cyclists. Based on this model, a safe actuation time could be selected for the performance assessment of the ADAS. Depending on how well the system matches this timing, the performance could be evaluated (for example, failing to match the timing would result in discarding the system).

Finally, the interaction model could also be used for predicting the intention of road users when implemented in a vehicle. In this way, if a driver has an indication about the future action of a cyclist at an unregulated intersection, an accident or an unnecessary deceleration could be avoided to improve the fuel consumption and to drive more comfortably and safely. This could also help in the case of a VRU, where information could be given to other road users through a display installed on the vehicle body as in[45], for example, a cyclist or a pedestrian could be allowed to cross the road safely. The interaction parameters could be recorded using sensors (e.g. Lidar, Radar, stereo camera) or when measure at the bicycle could be communicated to the vehicle using wireless communication.

Appendix A

Selection of Parameters for Intention Prediction

A-1 Parameters for the bicycle

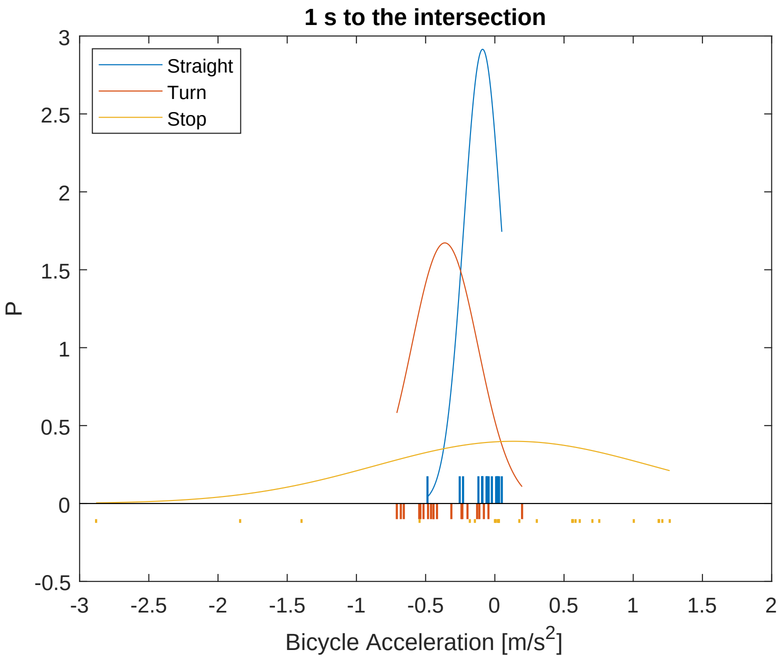


Figure A-1: Fits of normal distributions to the bicycle acceleration at TTI = 1.0 s

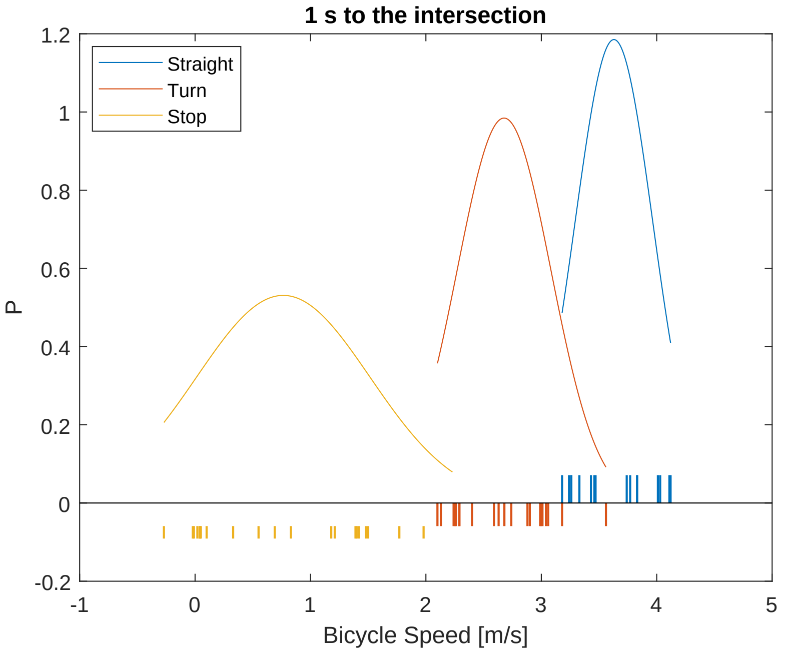


Figure A-2: Fits of normal distributions to the bicycle speed at TTI = 1.0 s

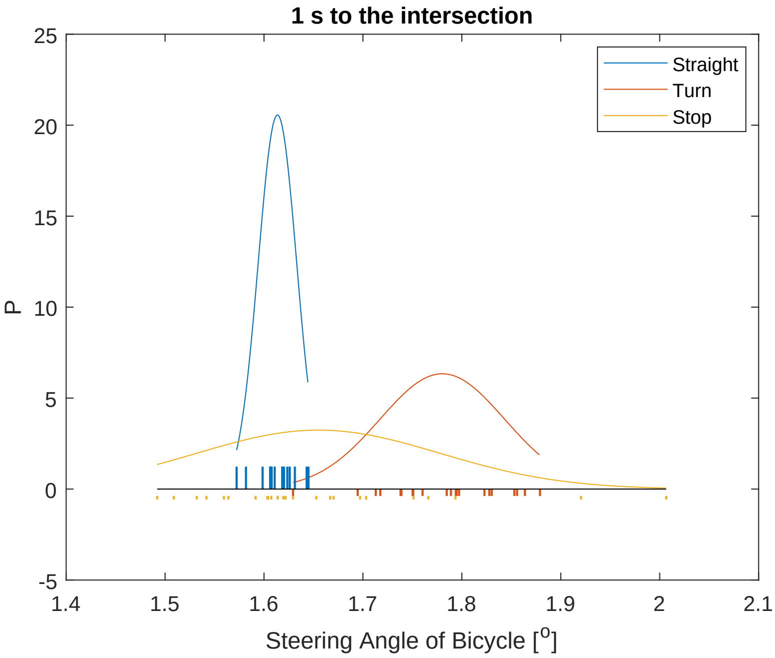


Figure A-3: Fits of normal distributions to the steering angle of the bicycle at TTI = 1.0 s

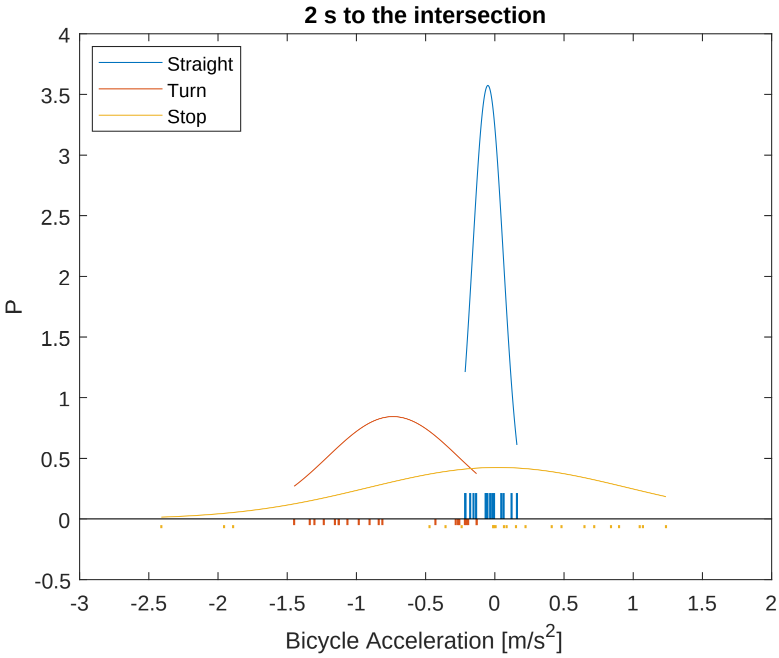


Figure A-4: Fits of normal distributions to the bicycle acceleration at TTI = 1.0 s

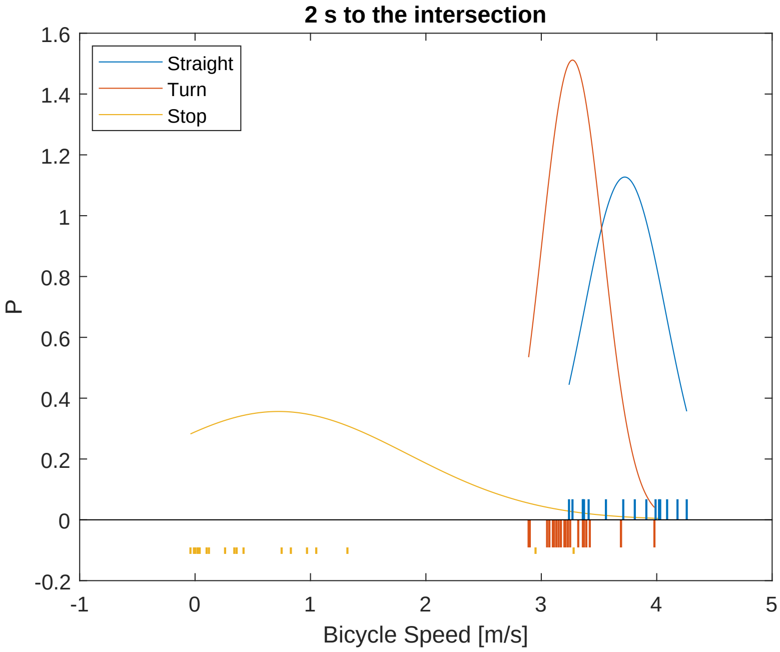


Figure A-5: Fits of normal distributions to the bicycle speed at TTI = 2.0 s

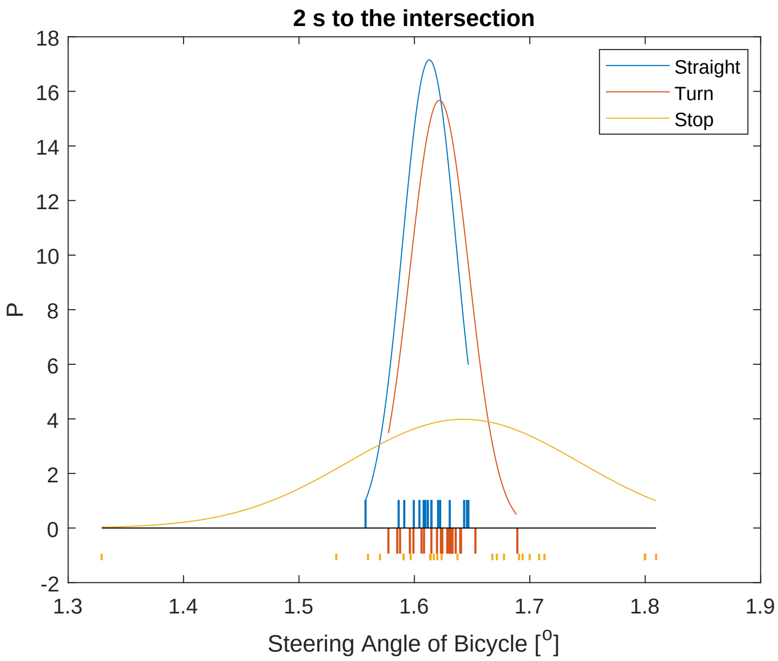


Figure A-6: Fits of normal distributions to the steering angle of the bicycle at TTI = 2.0 s

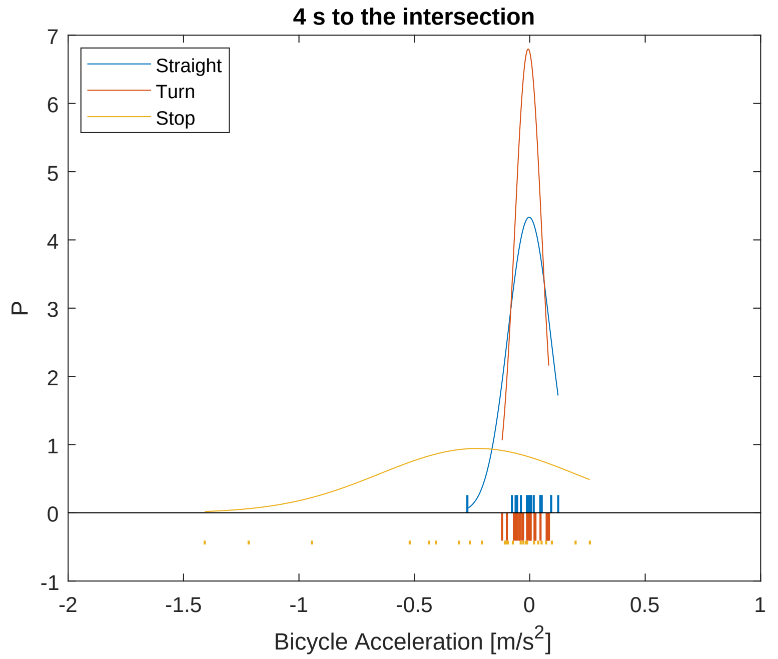


Figure A-7: Fits of normal distributions to the bicycle acceleration at TTI = 4.0 s

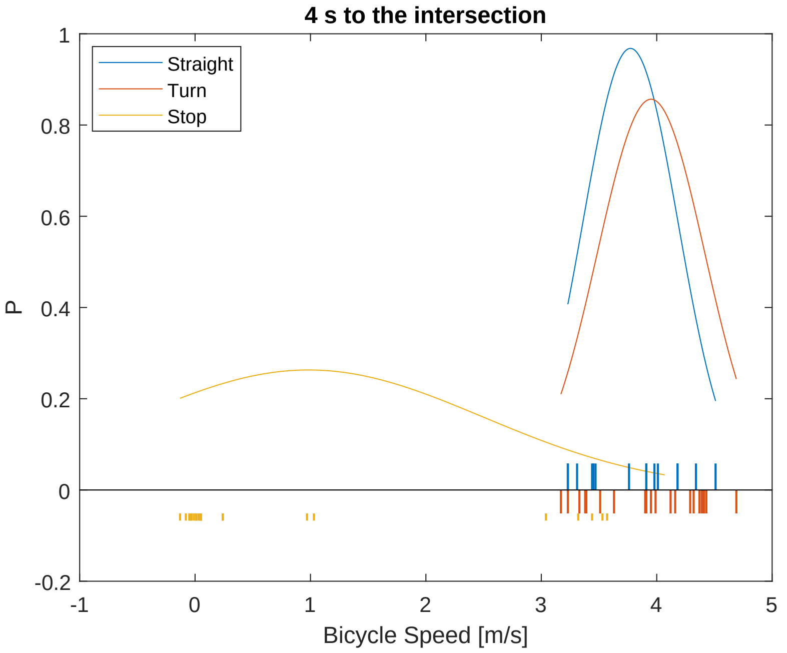


Figure A-8: Fits of normal distributions to the bicycle speed at TTI = 4.0 s

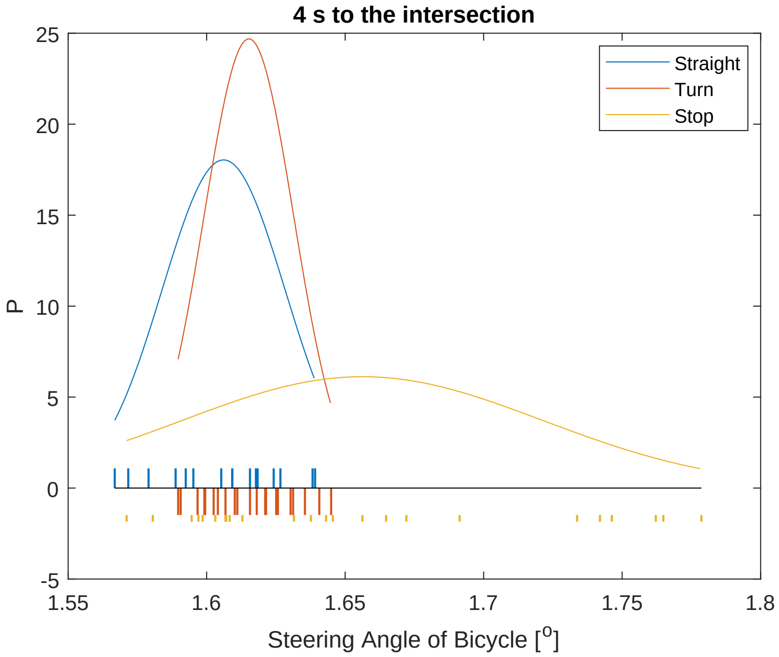


Figure A-9: Fits of normal distributions to the steering angle of the bicycle at TTI = 4.0 s

A-2 Parameters for the vehicle

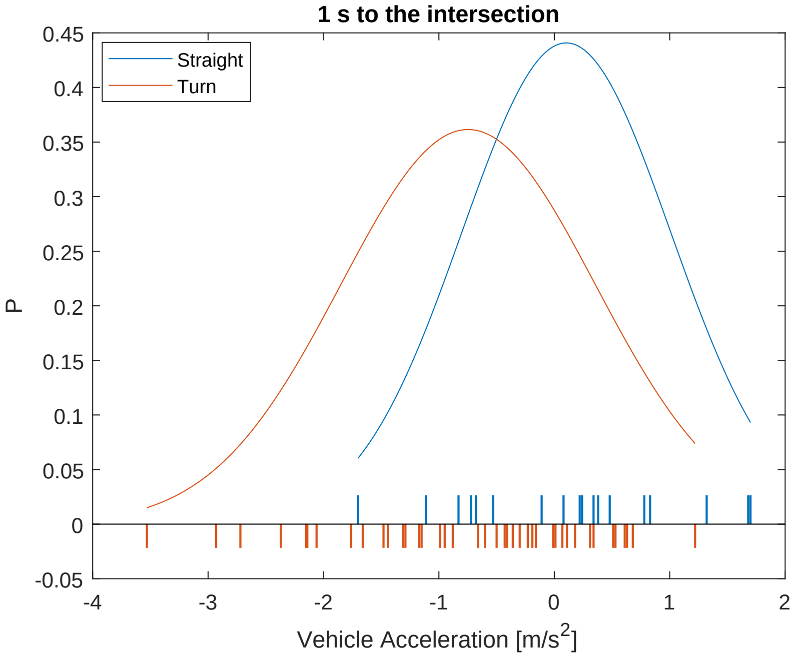


Figure A-10: Fits of normal distributions to the vehicle acceleration at TTI = 1.0 s

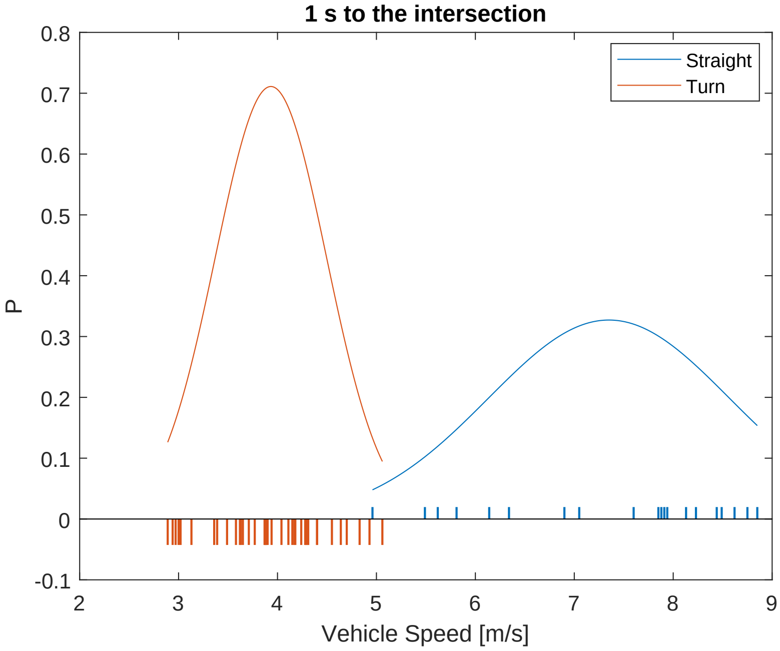


Figure A-11: Fits of normal distributions to the vehicle speed at TTI = 1.0 s

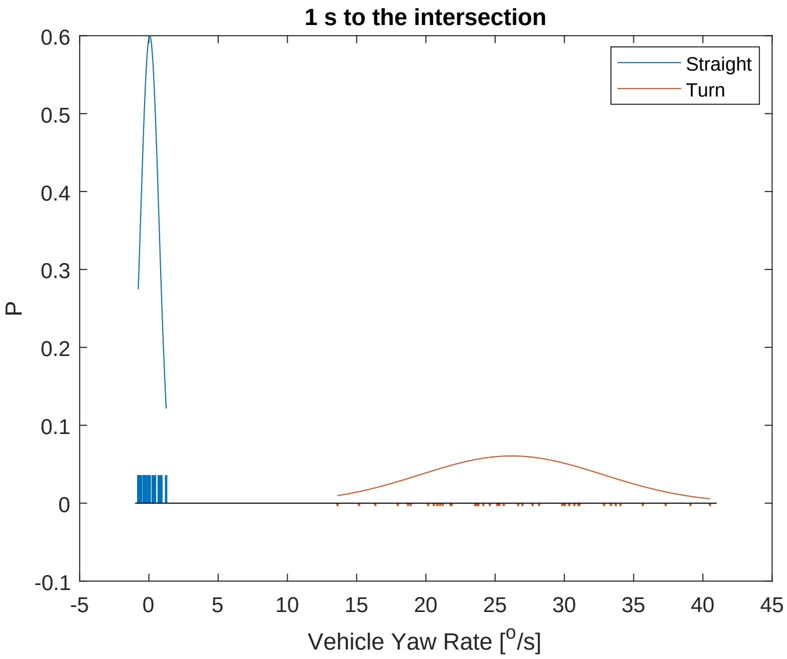


Figure A-12: Fits of normal distributions to the vehicle yaw rate at TTI = 1.0 s

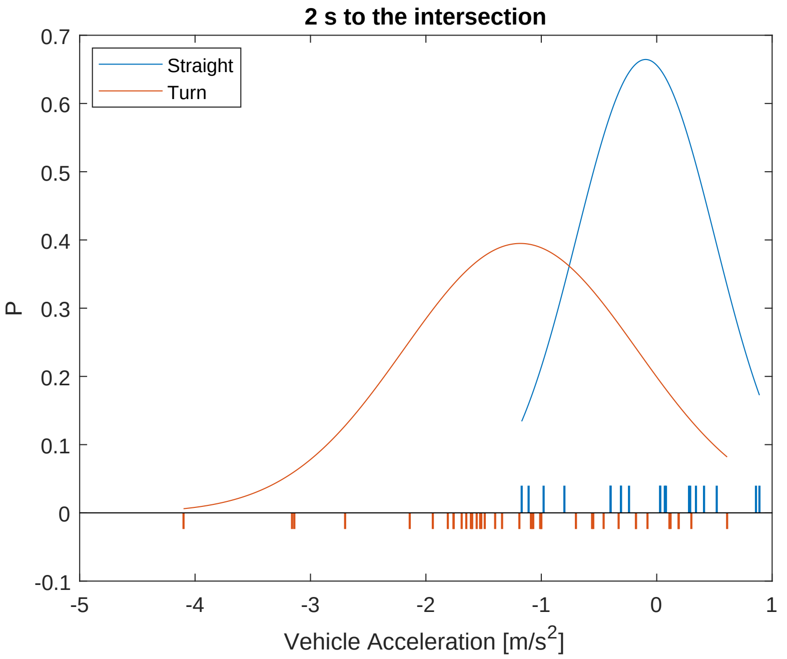


Figure A-13: Fits of normal distributions to the vehicle acceleration at TTI = 2.0 s

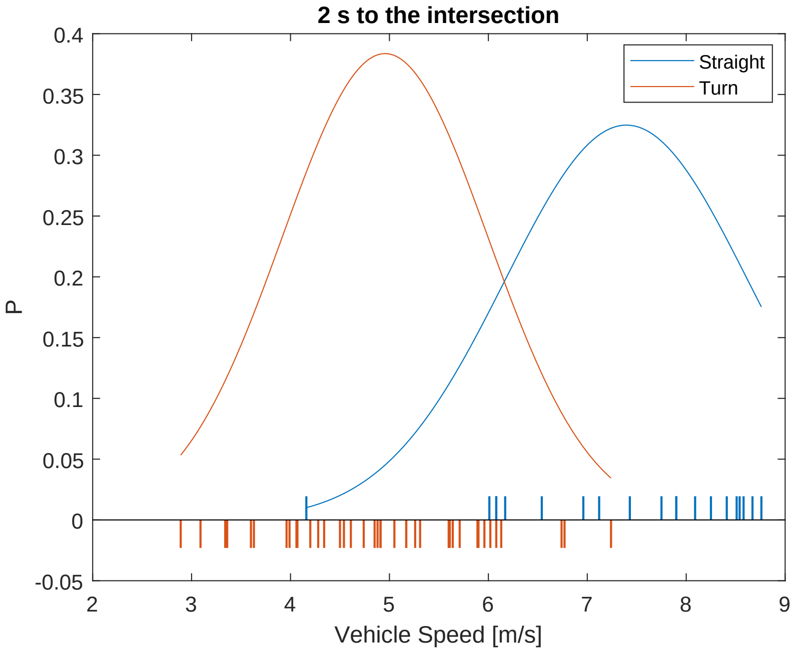


Figure A-14: Fits of normal distributions to the vehicle speed at TTI = 2.0 s

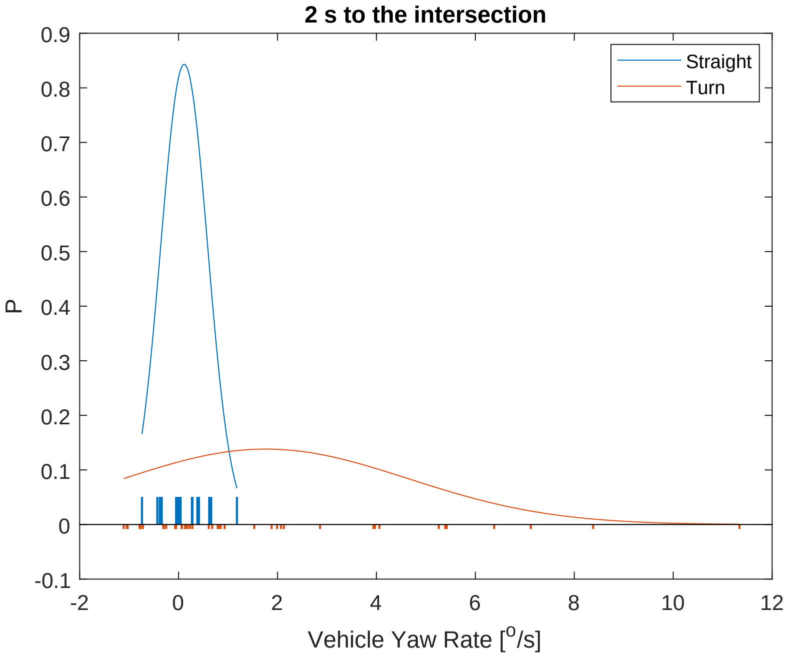


Figure A-15: Fits of normal distributions to the vehicle yaw rate at TTI = 2.0 s

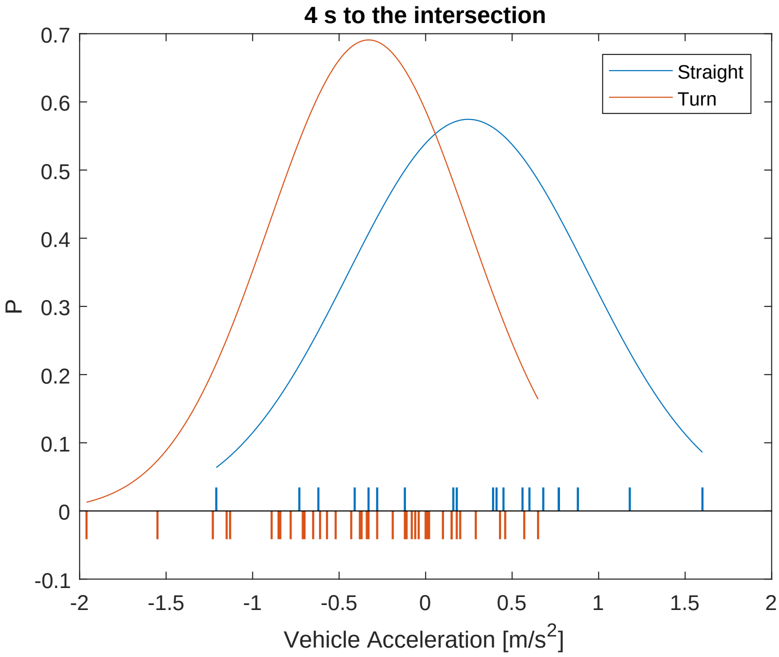


Figure A-16: Fits of normal distributions to the vehicle acceleration at TTI = 4.0 s

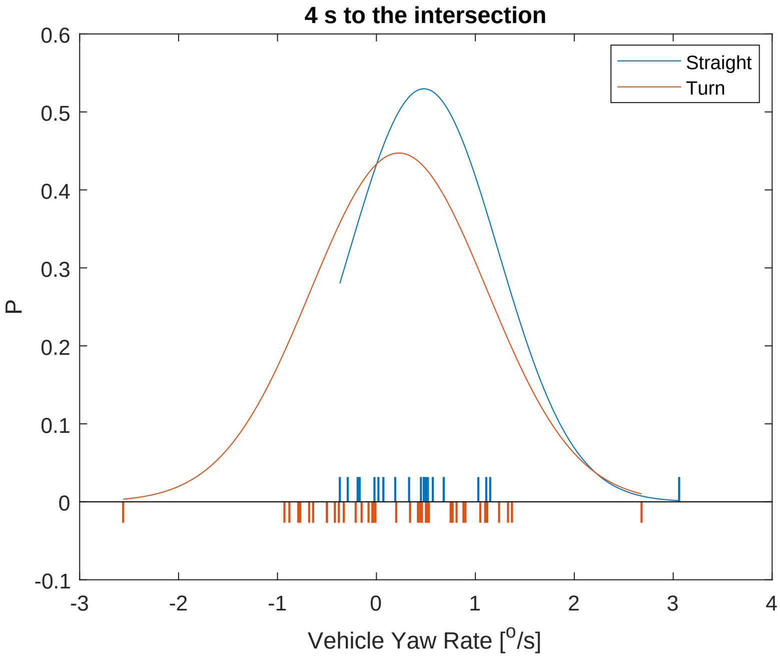


Figure A-18: Fits of normal distributions to the vehicle yaw rate at TTI = 4.0 s

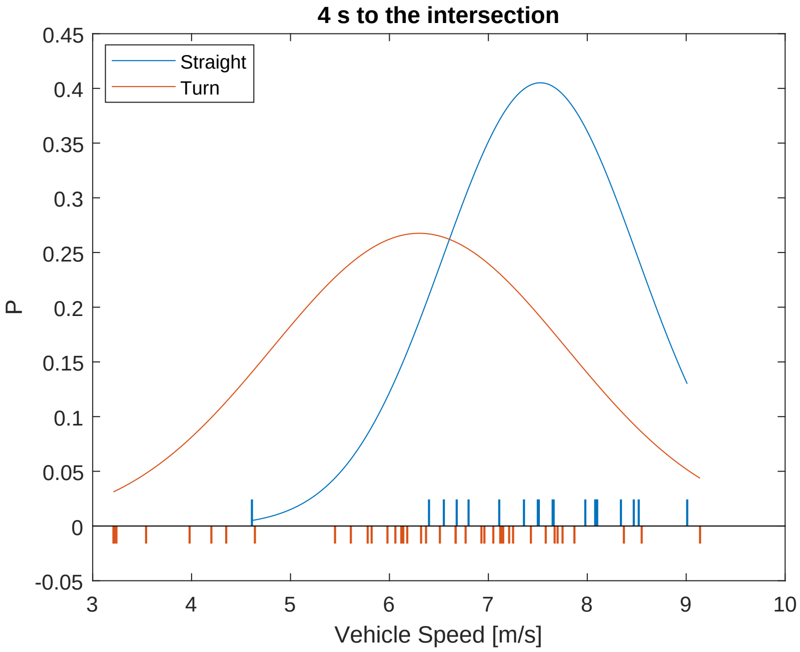


Figure A-17: Fits of normal distributions to the vehicle speed at TTI = 4.0 s

A-3 Cross Validation-Without Acceleration

A-3-1 Implementation of the bicycle parameters-With Interaction

Actual Cyclist Maneuvers at TTI= 0.5 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 0.5 s		
	Straight	Right Turn	Stop
Straight	15	0	0
Right Turn	2	18	0
Stop	0	4	19

Accuracy = 89.66%

Table A-1: Confusion Matrix for cyclist maneuvers with interaction (excluding acceleration) at TTI=0.5 s

Actual Cyclist Maneuvers at TTI= 1.0 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 1.0 s		
	Straight	Right Turn	Stop
Straight	14	1	0
Right Turn	3	17	0
Stop	0	2	21

Accuracy = 89.66%

Table A-2: Confusion Matrix for cyclist maneuvers with interaction (excluding acceleration) at TTI=1.0 s

Actual Cyclist Maneuvers at TTI= 1.5 s	**Predicted Cyclist Maneuvers (Interaction) at TTI = 1.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	13	2	0
Right Turn	5	14	0
Stop	1	0	22

Accuracy = 84.48%

Table A-3: Confusion Matrix for cyclist maneuvers with interaction (excluding acceleration) at TTI=1.5 s

Actual Cyclist Maneuvers at TTI= 2.0 s	**Predicted Cyclist Maneuvers (Interaction) at TTI = 2.0 s**		
	Straight	**Right Turn**	**Stop**
Straight	12	3	0
Right Turn	13	7	0
Stop	3	0	20

Accuracy = 67.24%

Table A-4: Confusion Matrix for cyclist maneuvers with interaction (excluding acceleration) at TTI=2.0 s

Actual Cyclist Maneuvers at TTI= 2.5 s	**Predicted Cyclist Maneuvers (Interaction) at TTI = 2.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	11	4	0
Right Turn	20	0	0
Stop	3	0	20

Accuracy = 53.45%

Table A-5: Confusion Matrix for cyclist maneuvers with interaction (excluding acceleration) at TTI=2.5 s

Actual Cyclist Maneuvers at TTI= 3.0 s	**Predicted Cyclist Maneuvers (Interaction) at TTI = 3.0 s**		
	Straight	**Right Turn**	**Stop**
Straight	11	4	0
Right Turn	20	0	0
Stop	3	0	20

Accuracy = 53.45%

Table A-6: Confusion Matrix for cyclist maneuvers with interaction (excluding acceleration) at TTI=3.0 s

A-3-2 Implementation of the bicycle parameters-Without Interaction

Actual Cyclist Maneuvers at TTI= 0.5 s	**Predicted Cyclist Maneuvers at TTI = 0.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	14	0	1
Right Turn	0	20	0
Stop	0	2	21

Accuracy = 94.83%

Table A-7: Confusion Matrix for cyclist maneuvers (without acceleration) at TTI=0.5 s

Actual Cyclist Maneuvers at TTI= 1.0 s	**Predicted Cyclist Maneuvers at TTI = 1.0 s**		
	Straight	**Right Turn**	**Stop**
Straight	12	3	0
Right Turn	2	18	0
Stop	0	2	21

Accuracy = 89.66%

Table A-8: Confusion Matrix for cyclist maneuvers (without acceleration) at TTI=1.0 s

Actual Cyclist Maneuvers at TTI= 1.5 s	**Predicted Cyclist Maneuvers at TTI = 1.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	5	10	0
Right Turn	0	18	2
Stop	1	0	22

Accuracy = 77.59%

Table A-9: Confusion Matrix for cyclist maneuvers (without acceleration) at TTI=1.5 s

Actual Cyclist Maneuvers at TTI= 2.0 s	**Predicted Cyclist Maneuvers at TTI = 2.0 s**		
	Straight	**Right Turn**	**Stop**
Straight	0	12	3
Right Turn	1	13	6
Stop	2	1	20

Accuracy = 56.90%

Table A-10: Confusion Matrix for cyclist maneuvers (without acceleration) at TTI=2.0 s

Actual Cyclist Maneuvers at TTI= 2.5 s	**Predicted Cyclist Maneuvers at TTI = 2.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	0	10	5
Right Turn	0	11	9
Stop	2	1	20

Accuracy = 53.45%

Table A-11: Confusion Matrix for cyclist maneuvers (without acceleration) at TTI=2.5 s

Actual Cyclist Maneuvers at TTI= 3.0 s	Predicted Cyclist Maneuvers at TTI = 3.0 s		
	Straight	Right Turn	Stop
Straight	0	11	4
Right Turn	0	13	7
Stop	0	4	19

Accuracy = 53.45%

Table A-12: Confusion Matrix for cyclist maneuvers (without acceleration) at TTI=3.0 s

A-3-3 Implementation of the vehicle parameters-With Interaction

Actual Vehicle Maneuvers at TTI= 0.5 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 0.5 s	
	Straight	Right Turn
Straight	18	0
Right Turn	0	40

Accuracy = 100%

Table A-13: Confusion Matrix for vehicle maneuvers with interaction (excluding acceleration) at TTI=0.5 s

Actual Vehicle Maneuvers at TTI= 1.0 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 1.0 s	
	Straight	Right Turn
Straight	18	0
Right Turn	0	40

Accuracy = 100%

Table A-14: Confusion Matrix for vehicle maneuvers with interaction (excluding acceleration) at TTI=1.0 s

Actual Vehicle Maneuvers at TTI= 1.5 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 1.5 s	
	Straight	Right Turn
Straight	18	0
Right Turn	1	39

Accuracy = 98.28%

Table A-15: Confusion Matrix for vehicle maneuvers with interaction (excluding acceleration) at TTI=1.5 s

Actual Vehicle Maneuvers at TTI= 2.0 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 2.0 s	
	Straight	Right Turn
Straight	17	1
Right Turn	17	23

Accuracy = 68.97%

Table A-16: Confusion Matrix for vehicle maneuvers with interaction (excluding acceleration) at TTI=2.0 s

Actual Vehicle Maneuvers at TTI= 2.5 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 2.5 s	
	Straight	Right Turn
Straight	16	2
Right Turn	27	13

Accuracy = 50.00%

Table A-17: Confusion Matrix for vehicle maneuvers with interaction (excluding acceleration) at TTI=2.5 s

Actual Vehicle Maneuvers at TTI= 3.0 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 3.0 s	
	Straight	Right Turn
Straight	16	2
Right Turn	31	9

Accuracy = 43.10%

Table A-18: Confusion Matrix for vehicle maneuvers with interaction (excluding acceleration) at TTI=3.0 s

A-3-4 Implementation of the vehicle parameters-Without Interaction

Actual Vehicle Maneuvers at TTI= 1.5 s	Predicted Vehicle Maneuvers at TTI = 1.5 s	
	Straight	Right Turn
Straight	18	0
Right Turn	0	40

Accuracy = 100.00%

Table A-19: Confusion Matrix for vehicle maneuvers (excluding acceleration) at TTI=0.5 s

Actual Vehicle Maneuvers at TTI= 1.0 s	Predicted Vehicle Maneuvers at TTI = 1.0 s	
	Straight	Right Turn
Straight	17	1
Right Turn	0	40

Accuracy = 98.28%

Table A-20: Confusion Matrix for vehicle maneuvers (excluding acceleration) at TTI=1.0 s

Actual Vehicle Maneuvers at TTI= 1.5 s	Predicted Vehicle Maneuvers at TTI = 1.5 s	
	Straight	Right Turn
Straight	17	1
Right Turn	0	40

Accuracy = 98.28%

Table A-21: Confusion Matrix for vehicle maneuvers (excluding acceleration) at TTI=1.5 s

Actual Vehicle Maneuvers at TTI= 2.0 s	Predicted Vehicle Maneuvers at TTI = 2.0 s	
	Straight	Right Turn
Straight	17	1
Right Turn	14	26
		Accuracy = 74.14%

Table A-22: Confusion Matrix for vehicle maneuvers (excluding acceleration) at TTI=2.0 s

Actual Vehicle Maneuvers at TTI= 2.5 s	Predicted Vehicle Maneuvers at TTI = 2.5 s	
	Straight	Right Turn
Straight	17	1
Right Turn	24	16
		Accuracy = 56.90%

Table A-23: Confusion Matrix for vehicle maneuvers (excluding acceleration) at TTI=2.5 s

Actual Vehicle Maneuvers at TTI= 3.0 s	Predicted Vehicle Maneuvers at TTI = 3.0 s	
	Straight	Right Turn
Straight	15	3
Right Turn	27	13
		Accuracy = 48.28%

Table A-24: Confusion Matrix for vehicle maneuvers (excluding acceleration) at TTI=3.0 s

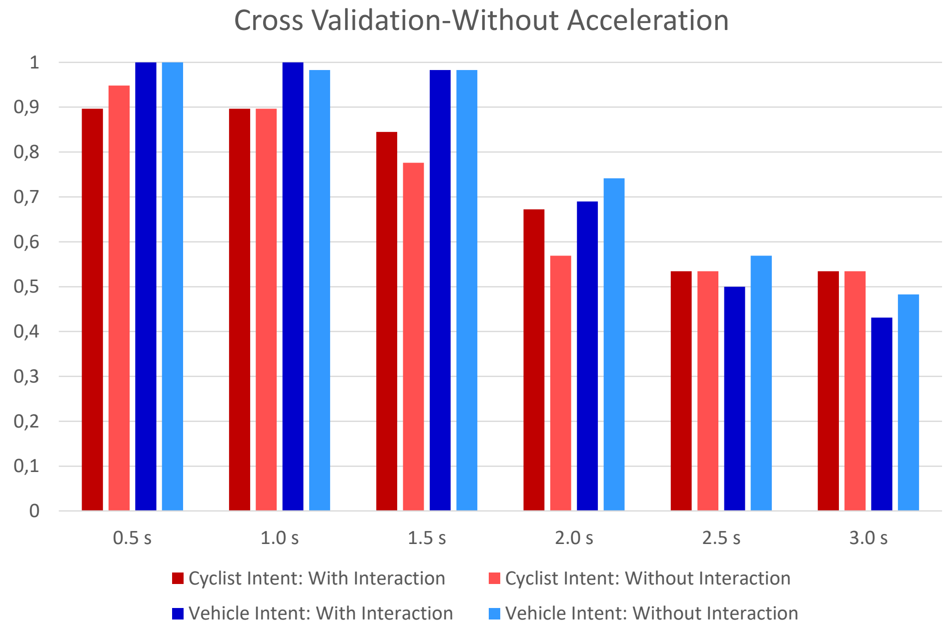


Figure A-19: Overall Accuracy of the interaction model (excluding acceleration)

Time to the Crossing [s]	Type of Road User					
	Vehicle (Interaction)			Cyclist (Interaction)		
	HMM with acceleration [%]	HMM without acceleration [%]	Overall Performance [%]	HMM with acceleration [%]	HMM without acceleration [%]	Overall Performance [%]
0.5	100	100	0	94.83	89.66	5.17
1.0	100	100	0	94.83	89.66	5.17
1.5	98.28	98.28	0	94.83	84.48	10.35
2.0	91.38	68.97	22.41	86.21	67.24	18.97
2.5	82.76	50.00	32.76	77.59	53.45	24.14
3.0	75.86	43.10	32.76	67.24	53.45	13.79

Table A-25: Comparison of the interaction model with and without acceleration

Time to the Crossing [s]	Type of Road User					
	Vehicle (Without Interaction)			Cyclist (Without Interaction)		
	HMM with acceleration [%]	HMM without acceleration [%]	Overall Performance [%]	HMM with acceleration [%]	HMM without acceleration [%]	Overall Performance [%]
0.5	100	100	0	94.83	94.83	0
1.0	100	98.28	1.72	94.83	89.66	5.17
1.5	98.28	98.28	0	94.83	77.59	17.24
2.0	87.93	74.14	13.79	84.48	56.90	27.58
2.5	81.03	56.90	24.13	72.41	53.45	18.96
3.0	60.34	48.28	12.06	58.62	53.45	5.17

Table A-26: Comparison of HMM with and without acceleration

Appendix B

Selection of Interaction Parameters

B-1 Interaction Parameters of the Vehicle

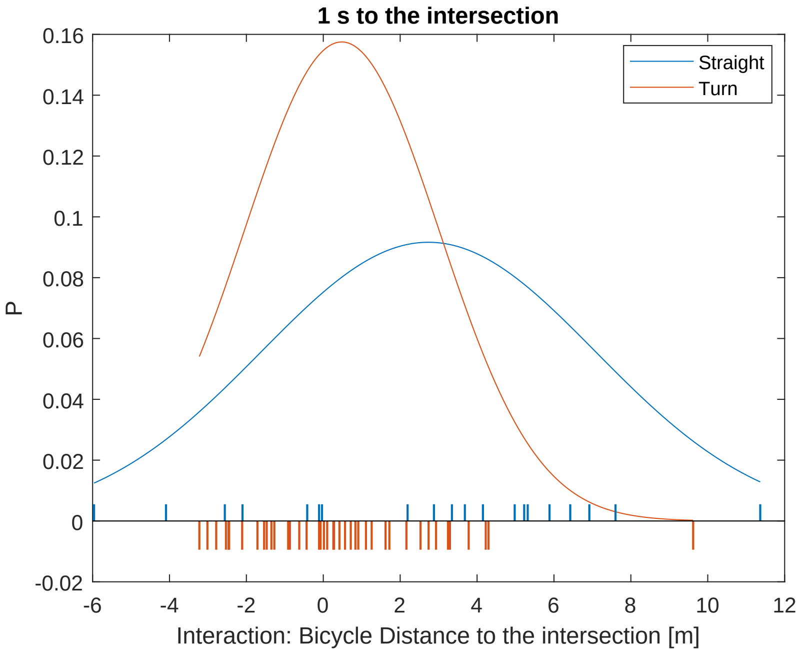


Figure B-1: Fit of normal distributions to the bicycle distance to the intersection at TTI = 1.0 s

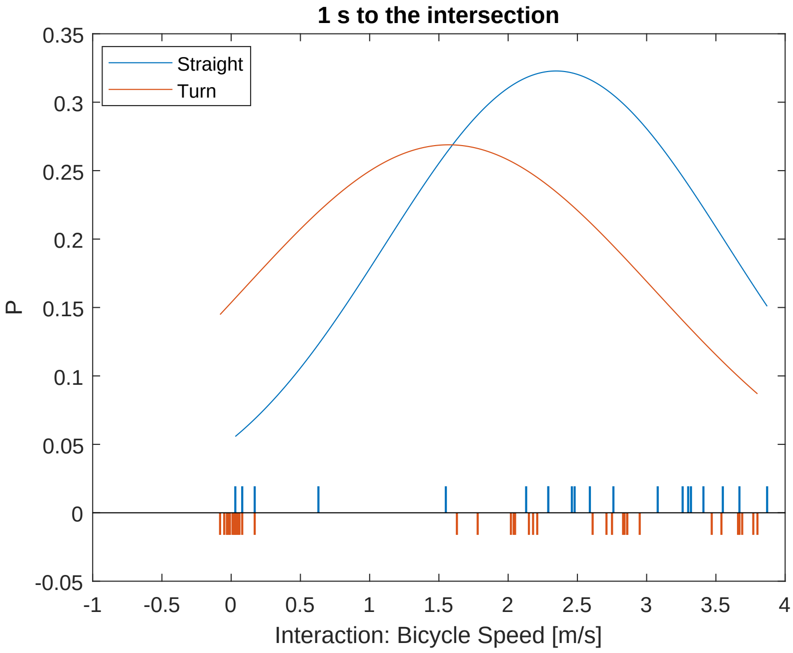


Figure B-2: Fit of normal distributions to the bicycle speed at TTI = 1.0 s

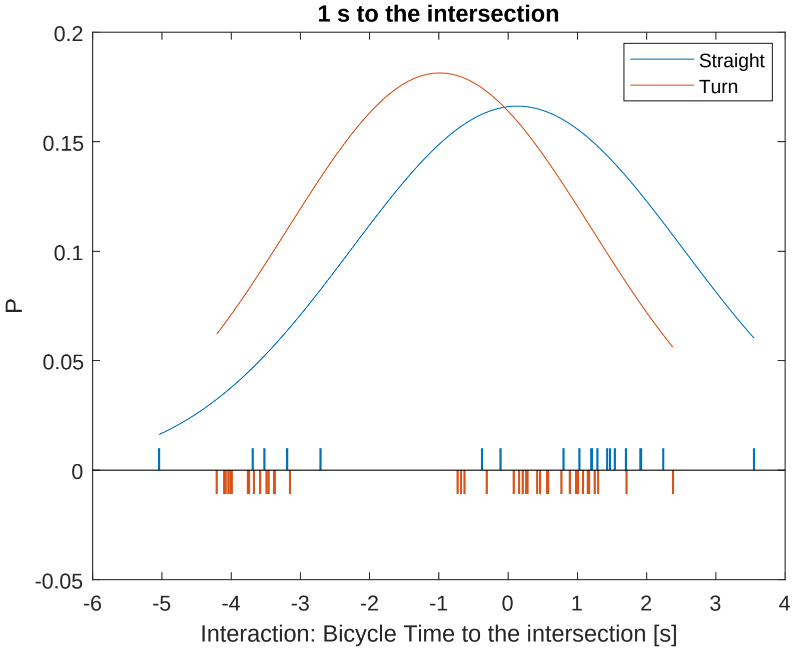


Figure B-3: Fit of normal distributions to the bicycle time to the intersection (TTI = 1.0 s)

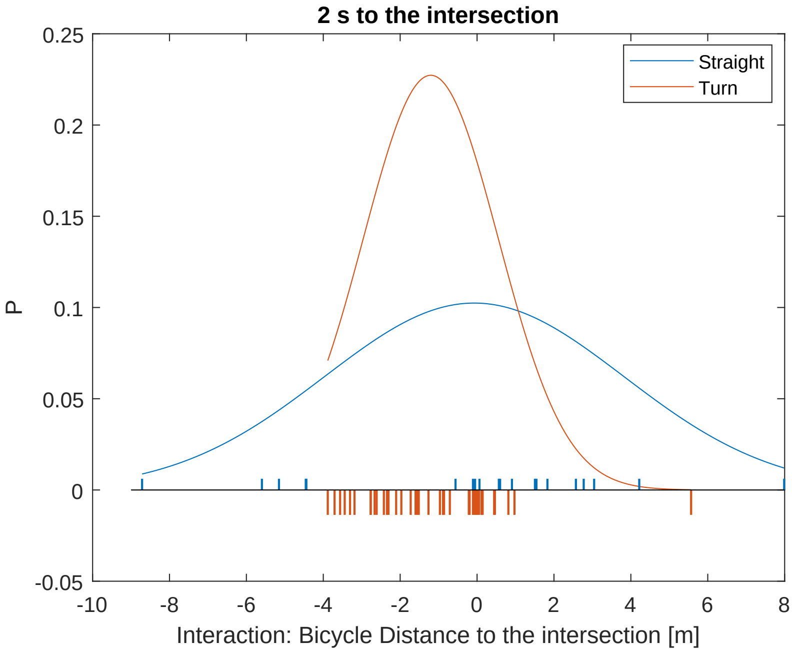


Figure B-4: Fit of normal distributions to the bicycle distance to the intersection at TTI = 2.0 s

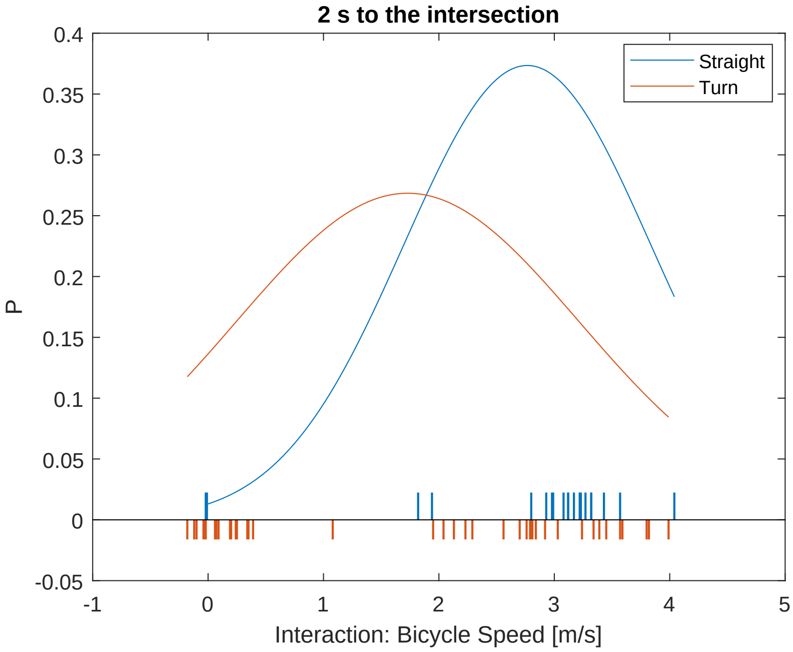


Figure B-5: Fit of normal distributions to the bicycle speed at TTI = 2.0 s

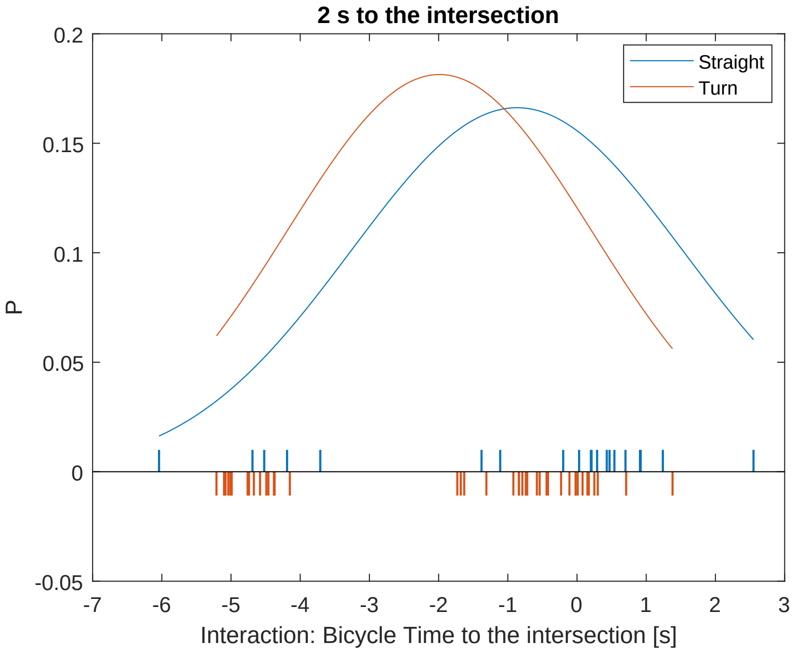


Figure B-6: Fit of normal distributions to the bicycle time to the intersection (TTI = 2.0 s)

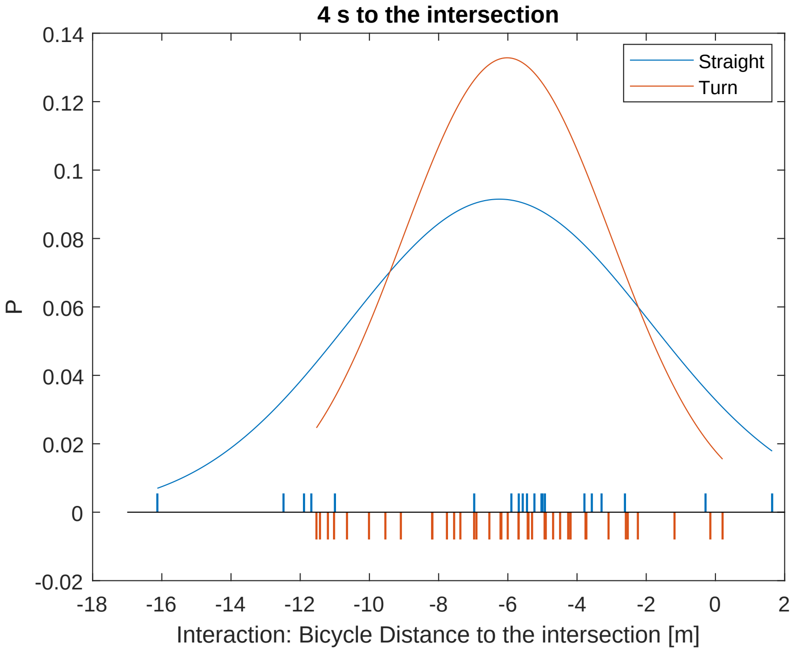


Figure B-7: Fit of normal distributions to the bicycle distance to the intersection at TTI = 4.0 s

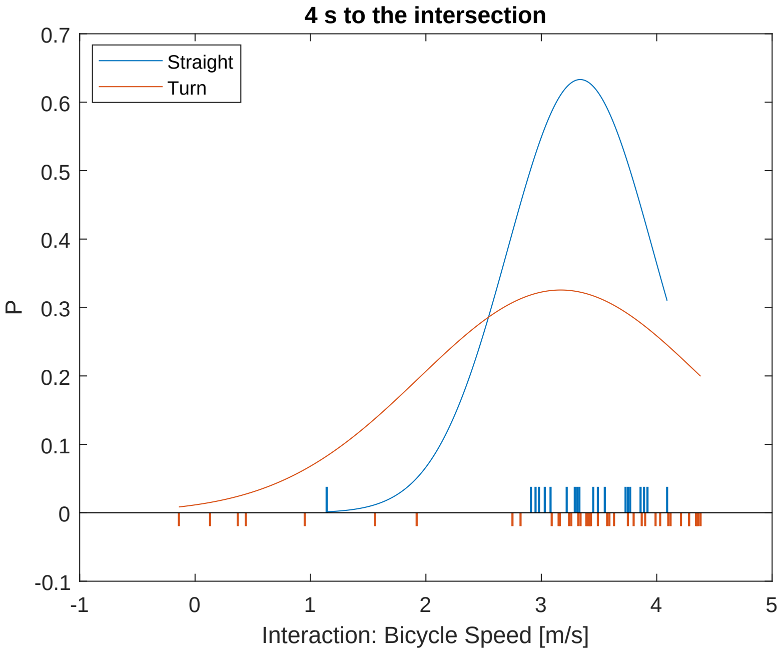


Figure B-8: Fit of normal distributions to the bicycle speed at TTI $= 4.0$ s

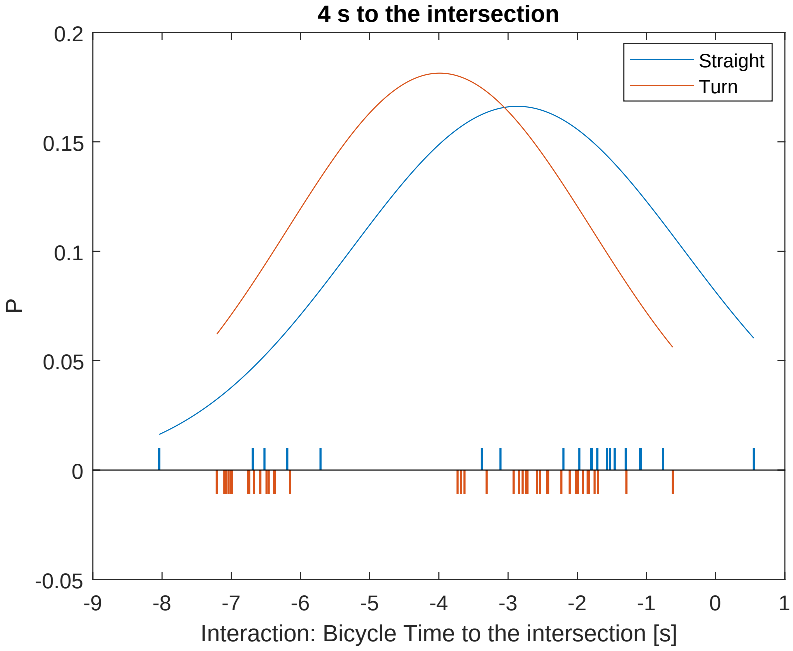


Figure B-9: Fit of normal distributions to the bicycle time to the intersection (TTI $= 4.0$ s)

B-2 Interaction Parameters of the Bicycle

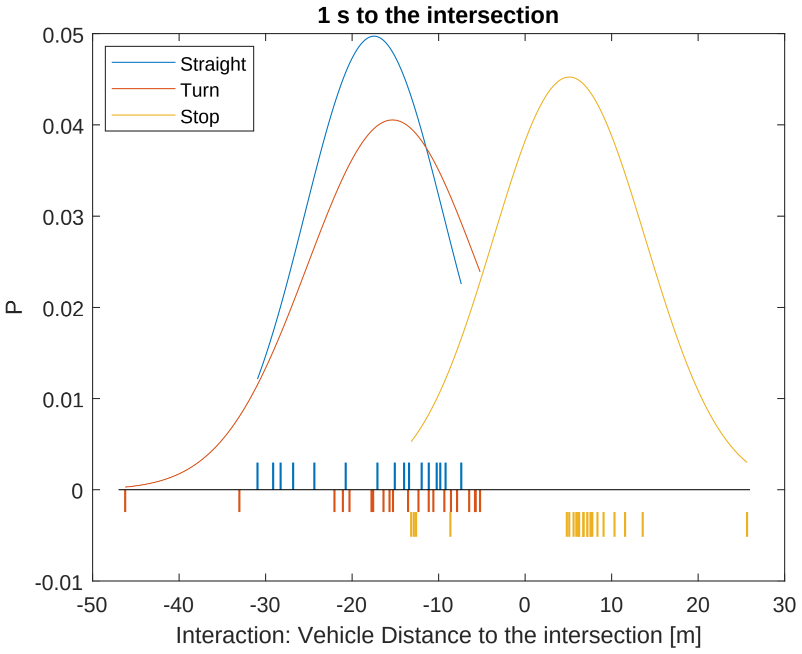


Figure B-10: Fit of normal distributions to the vehicle distance to the intersection at TTI = 1.0 s

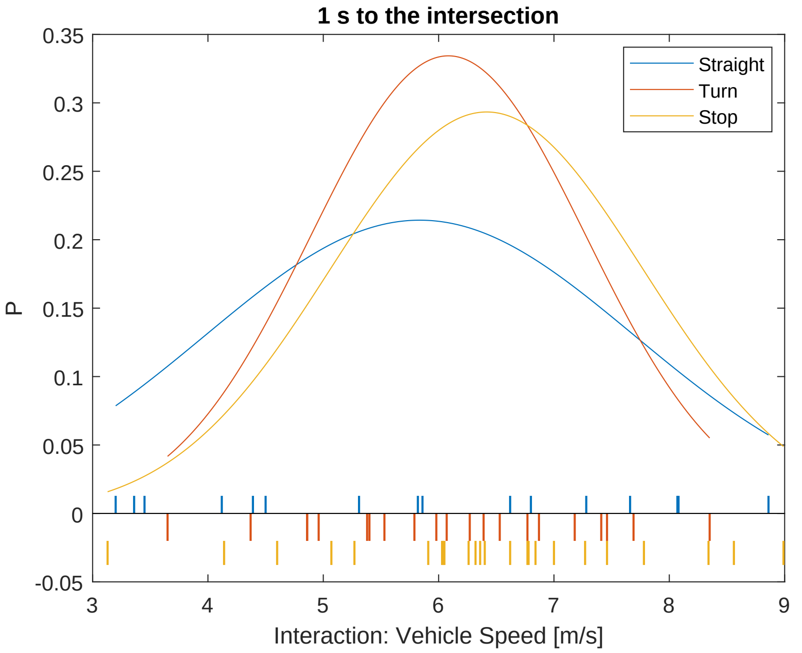


Figure B-11: Fit of normal distributions to the vehicle speed at TTI = 1.0 s

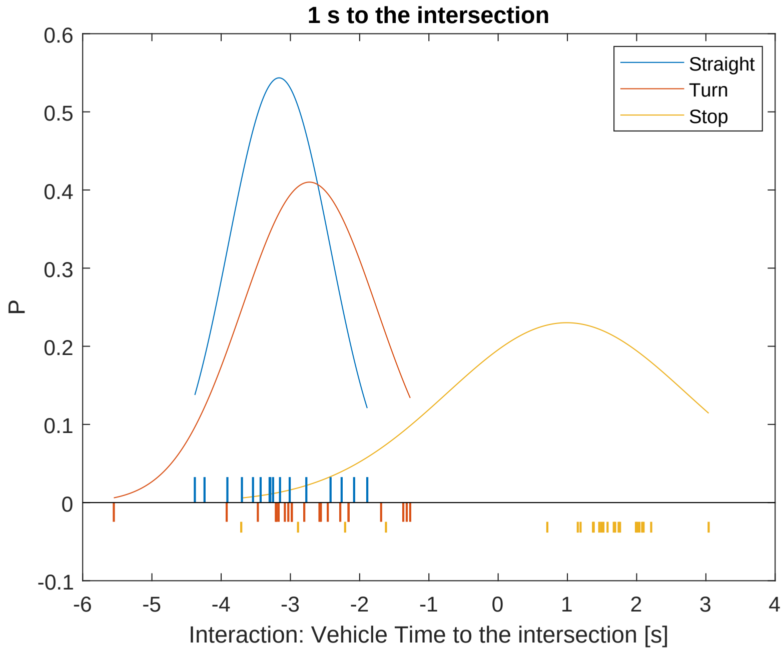


Figure B-12: Fit of normal distributions to the vehicle time to the intersection (TTI = 1.0 s)

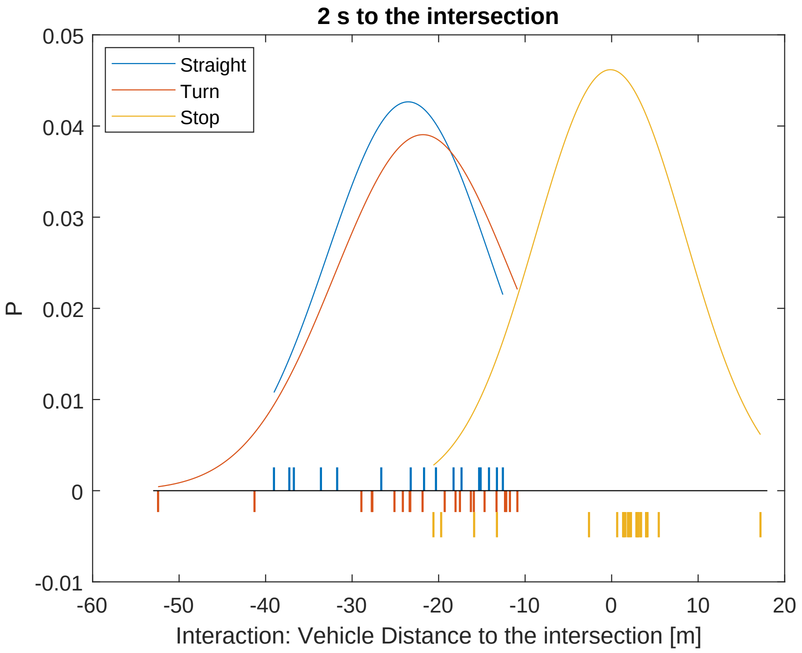


Figure B-13: Fit of normal distributions to the vehicle distance to the intersection at TTI = 2.0 s

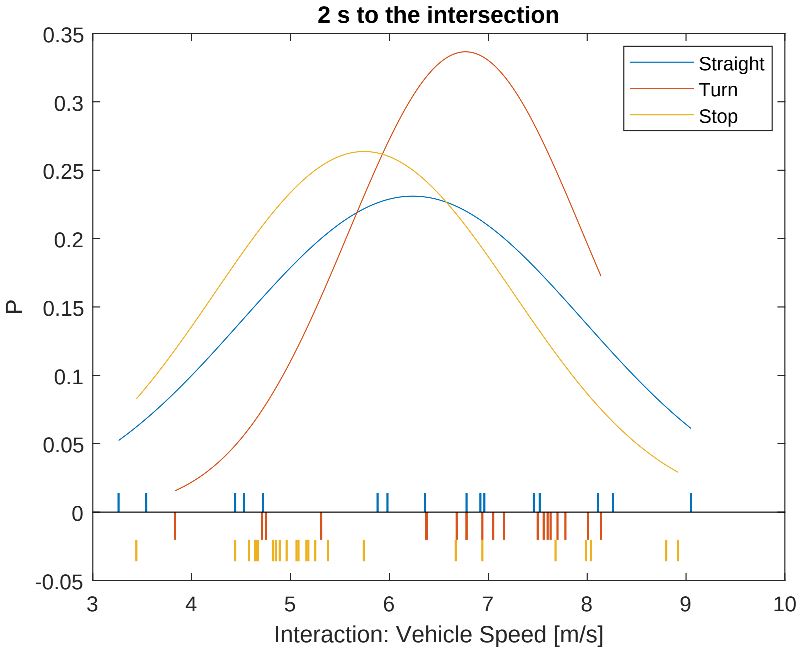


Figure B-14: Fit of normal distributions to the vehicle speed at TTI = 2.0 s

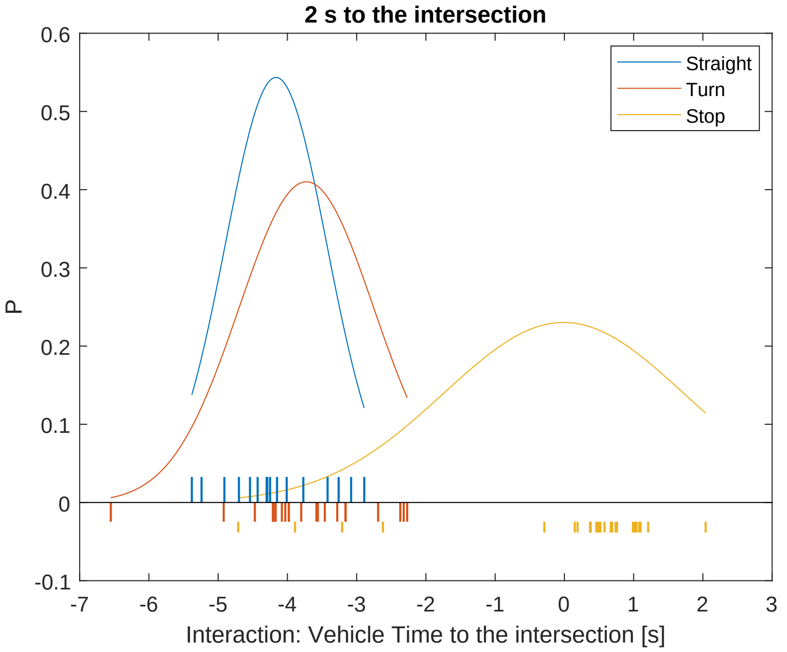


Figure B-15: Fit of normal distributions to the vehicle time to the intersection (TTI = 2.0 s)

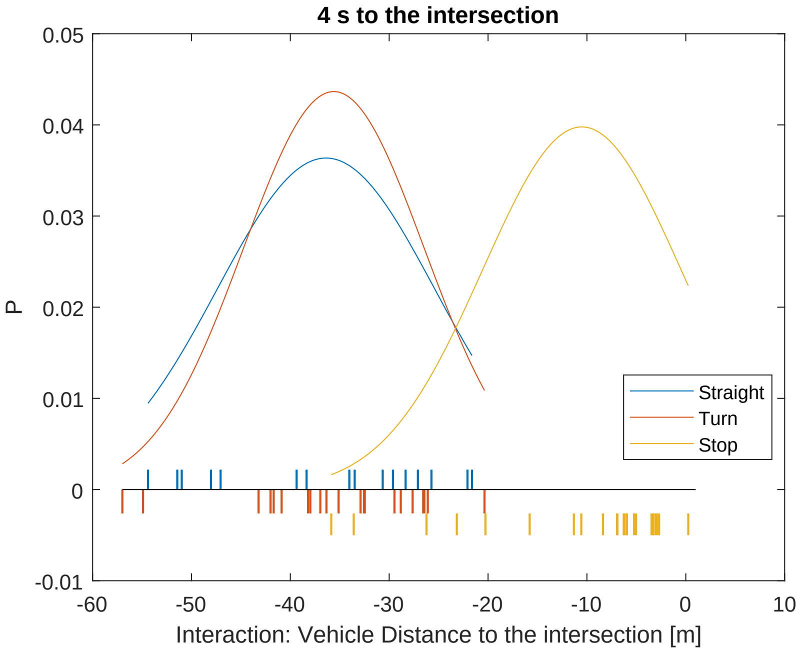


Figure B-16: Fit of normal distributions to the vehicle distance to the intersection at TTI = 4.0 s

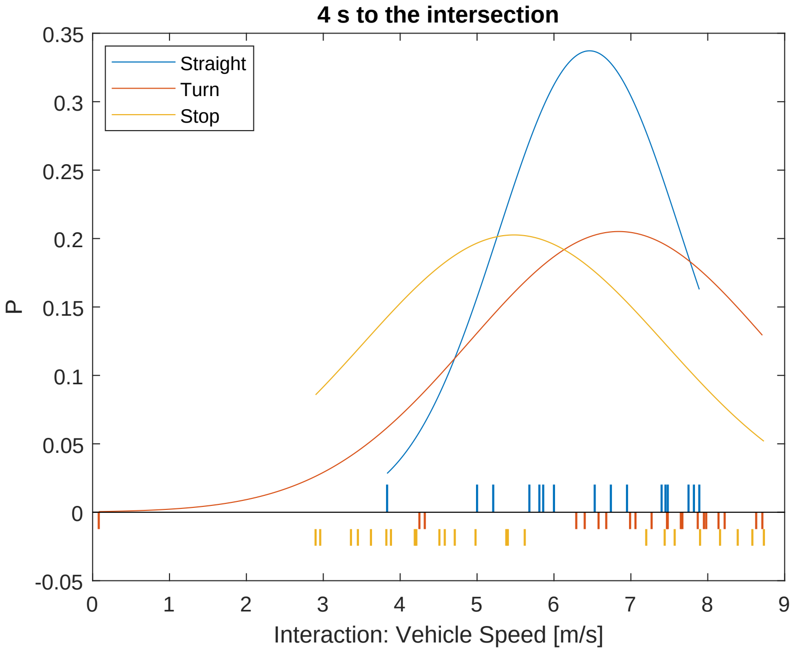


Figure B-17: Fit of normal distributions to the vehicle speed at TTI = 4.0 s

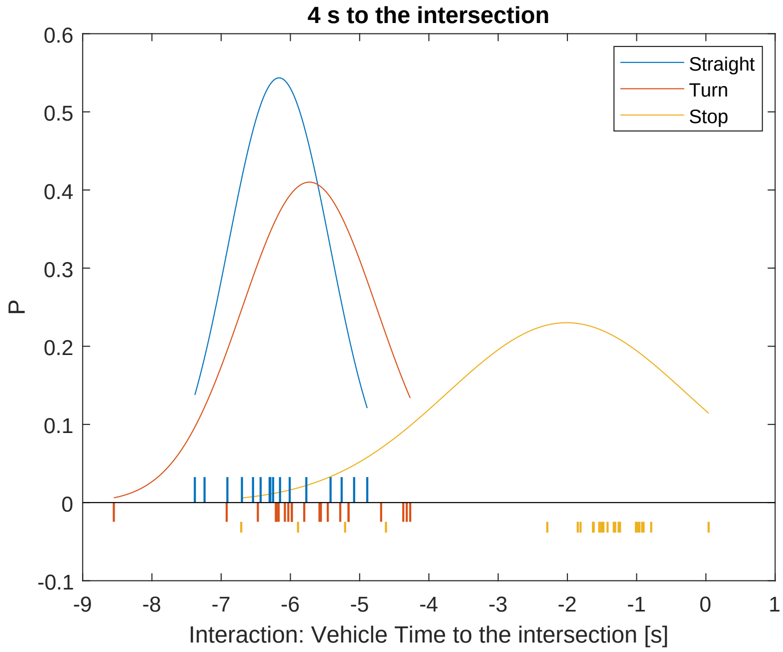


Figure B-18: Fit of normal distributions to the vehicle time to the intersection (TTI = 4.0 s)

Appendix C

Cross Validation

C-1 Cyclist-With interaction

Actual Cyclist Maneuvers at TTI= 0.5 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 0.5 s		
	Straight	Right Turn	Stop
Straight	14	0	1
Right Turn	2	18	0
Stop	0	0	23
			Accuracy = 94.83%

Table C-1: Confusion Matrix for cyclist maneuvers with interaction at TTI=0.5 s

Actual Cyclist Maneuvers at TTI= 1.0 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 1.0 s		
	Straight	Right Turn	Stop
Straight	14	1	0
Right Turn	2	18	0
Stop	0	0	23
			Accuracy = 94.83%

Table C-2: Confusion Matrix for cyclist maneuvers with interaction at TTI=1.0 s

Actual Cyclist Maneuvers at TTI= 1.5 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 1.5 s		
	Straight	Right Turn	Stop
Straight	15	0	0
Right Turn	2	17	1
Stop	0	0	23
			Accuracy = 94.83%

Table C-3: Confusion Matrix for cyclist maneuvers with interaction at TTI=1.5 s

Actual Cyclist Maneuvers at TTI= 2.0 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 2.0 s		
	Straight	Right Turn	Stop
Straight	14	1	0
Right Turn	5	14	1
Stop	1	0	22

Accuracy = 86.21%

Table C-4: Confusion Matrix for cyclist maneuvers with interaction at TTI=2.0 s

Actual Cyclist Maneuvers at TTI= 2.5 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 2.5 s		
	Straight	Right Turn	Stop
Straight	14	1	0
Right Turn	9	11	0
Stop	2	1	20

Accuracy = 77.59%

Table C-5: Confusion Matrix for cyclist maneuvers with interaction at TTI=2.5 s

Actual Cyclist Maneuvers at TTI= 3.0 s	Predicted Cyclist Maneuvers (Interaction) at TTI = 3.0 s		
	Straight	Right Turn	Stop
Straight	14	1	0
Right Turn	9	11	0
Stop	2	1	20

Accuracy = 67.24%

Table C-6: Confusion Matrix for cyclist maneuvers with interaction at TTI=3.0 s

C-2 Cyclist-Without interaction

Actual Cyclist Maneuvers at TTI= 0.5 s	Predicted Cyclist Maneuvers at TTI = 0.5 s		
	Straight	Right Turn	Stop
Straight	14	0	1
Right Turn	2	18	0
Stop	0	0	23

Accuracy = 94.83%

Table C-7: Confusion Matrix for cyclist maneuvers at TTI=0.5 s

Actual Cyclist Maneuvers at TTI= 1.0 s	Predicted Cyclist Maneuvers at TTI = 1.0 s		
	Straight	Right Turn	Stop
Straight	14	0	1
Right Turn	2	18	0
Stop	0	0	23

Accuracy = 94.83%

Table C-8: Confusion Matrix for cyclist maneuvers at TTI=1.0 s

Actual Cyclist Maneuvers at TTI= 1.5 s	**Predicted Cyclist Maneuvers at TTI = 1.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	15	0	1
Right Turn	2	17	1
Stop	0	0	23

Accuracy = 94.83%

Table C-9: Confusion Matrix for cyclist maneuvers at TTI=1.5 s

Actual Cyclist Maneuvers at TTI= 2.0 s	**Predicted Cyclist Maneuvers at TTI = 2.0 s**		
	Straight	**Right Turn**	**Stop**
Straight	9	4	2
Right Turn	2	17	1
Stop	0	0	23

Accuracy = 84.48%

Table C-10: Confusion Matrix for cyclist maneuvers at TTI=2.0 s

Actual Cyclist Maneuvers at TTI= 2.5 s	**Predicted Cyclist Maneuvers at TTI = 2.5 s**		
	Straight	**Right Turn**	**Stop**
Straight	7	8	0
Right Turn	3	15	2
Stop	2	1	20

Accuracy = 72.41%

Table C-11: Confusion Matrix for cyclist maneuvers at TTI=2.5 s

Actual Cyclist Maneuvers at TTI= 3.0 s	**Predicted Cyclist Maneuvers at TTI = 3.0 s**		
	Straight	**Right Turn**	**Stop**
Straight	0	15	0
Right Turn	4	13	3
Stop	1	1	21

Accuracy = 58.62%

Table C-12: Confusion Matrix for cyclist maneuvers at TTI=3.0 s

C-3 Vehicle-With interaction

Actual Vehicle Maneuvers at TTI= 0.5 s	**Predicted Vehicle Maneuvers (Interaction) at TTI = 0.5 s**	
	Straight	**Right Turn**
Straight	18	0
Right Turn	0	40

Accuracy = 100%

Table C-13: Confusion Matrix for vehicle maneuvers with interaction at TTI=0.5 s

Actual Vehicle Maneuvers at TTI= 1.0 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 1.0 s	
	Straight	Right Turn
Straight	18	0
Right Turn	0	40

Accuracy = 100%

Table C-14: Confusion Matrix for vehicle maneuvers with interaction at TTI=1.0 s

Actual Vehicle Maneuvers at TTI= 1.5 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 1.5 s	
	Straight	Right Turn
Straight	17	1
Right Turn	0	40

Accuracy = 98.28%

Table C-15: Confusion Matrix for vehicle maneuvers with interaction at TTI=1.5 s

Actual Vehicle Maneuvers at TTI= 2.0 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 2.0 s	
	Straight	Right Turn
Straight	14	4
Right Turn	1	39

Accuracy = 91.38%

Table C-16: Confusion Matrix for vehicle maneuvers with interaction at TTI = 2.0 s

Actual Vehicle Maneuvers at TTI= 2.5 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 2.5 s	
	Straight	Right Turn
Straight	11	7
Right Turn	3	37

Accuracy = 82.76%

Table C-17: Confusion Matrix for vehicle maneuvers with interaction at TTI = 2.5 s

Actual Vehicle Maneuvers at TTI= 3.0 s	Predicted Vehicle Maneuvers (Interaction) at TTI = 3.0 s	
	Straight	Right Turn
Straight	11	7
Right Turn	3	37

Accuracy = 75.86%

Table C-18: Confusion Matrix for vehicle maneuvers with interaction at TTI = 3.0 s

C-4 Vehicle-Without interaction

Actual Vehicle Maneuvers at TTI= 0.5 s	**Predicted Vehicle Maneuvers at TTI = 0.5 s**	
	Straight	**Right Turn**
Straight	18	0
Right Turn	0	40

Accuracy = 100%

Table C-19: Confusion Matrix for vehicle maneuvers at TTI=0.5 s

Actual Vehicle Maneuvers at TTI= 1.0 s	**Predicted Vehicle Maneuvers at TTI = 1.0 s**	
	Straight	**Right Turn**
Straight	18	0
Right Turn	0	40

Accuracy = 100%

Table C-20: Confusion Matrix for vehicle maneuvers at TTI=1.0 s

Actual Vehicle Maneuvers at TTI= 1.5 s	**Predicted Vehicle Maneuvers at TTI = 1.5 s**	
	Straight	**Right Turn**
Straight	17	1
Right Turn	0	40

Accuracy = 98.28%

Table C-21: Confusion Matrix for vehicle maneuvers at TTI=1.5 s

Actual Vehicle Maneuvers at TTI= 2.0 s	**Predicted Vehicle Maneuvers at TTI = 2.0 s**	
	Straight	**Right Turn**
Straight	16	2
Right Turn	5	35

Accuracy = 87.93%

Table C-22: Confusion Matrix for vehicle maneuvers at TTI=2.0 s

Actual Vehicle Maneuvers at TTI= 2.5 s	**Predicted Vehicle Maneuvers at TTI = 2.5 s**	
	Straight	**Right Turn**
Straight	15	3
Right Turn	8	32

Accuracy = 81.03%

Table C-23: Confusion Matrix for vehicle maneuvers at TTI=2.5 s

Actual Vehicle Maneuvers at TTI= 3.0 s	**Predicted Vehicle Maneuvers at TTI = 3.0 s**	
	Straight	**Right Turn**
Straight	15	3
Right Turn	20	20

Accuracy = 60.34%

Table C-24: Confusion Matrix for vehicle maneuvers at TTI=3.0 s

Bibliography

[1] POPULATION REFERENCE BUREAU, “Road traffic accidents increase dramatically worldwide,” https://www.prb.org/roadtrafficaccidentsincreasedramaticallyworldwide/, 2006, [Online; February 11, 2018].

[2] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, and C. Mathers, “World report on road traffic injury prevention,” http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/en/, 2004, [Online; February 11, 2018].

[3] WHO, “Projections of mortality and causes of death, 2015 and 2030,” http://www.who.int/healthinfo/global_burden_disease/projections/en/, 2012, [Online; February 11, 2018].

[4] WHO, “Road traffic injuries,” http://www.who.int/mediacentre/factsheets/fs358/en/, 2018, [Online; February 12, 2018].

[5] SWOV, “Road deaths in the netherlands,” https://www.swov.nl/en/facts-figures/factsheet/road-deaths-netherlands, 2017, [Online; February 13, 2018].

[6] J. Pieters, “Netherlands cyclists most likely in eu to be hurt in traffic,” https://nltimes.nl/2016/04/12/netherlands-cyclists-likely-eu-hurt-traffic, 2016, [Online; February 13, 2018].

[7] H. G. Korving, C. Schagen, I. van Weijermars, W. Bijleveld, F. Wesseling, S. Bos, and H. NM & Stipdonk, “Monitor verkeersveiligheid 2016: achtergrondinformatie en onderzoeksverantwoording.” 2016.

[8] B. Dutch, “More cycling fatalities than deaths in cars,” https://bicycledutch.wordpress.com/2018/04/25/more-cycling-fatalities-than-deaths-in-cars/, 2018, [Online; February 12, 2018].

[9] S. Pawlett, “Autonomous vehicles will reduce accidents,” https://www.autosphere.ca/autojournal/dealerships-articles/2017/08/03/autonomous-vehicles-will-reduce-accidents/, 2017, [Online; February 12, 2018].

[10] C. Tingvall and N. Haworth, "Vision zero: an ethical approach to safety and mobility," in *6th ITE International Conference Road Safety & Traffic Enforcement: Beyond*, vol. 1999, 2000, pp. 6–7.

[11] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 1455–1462.

[12] R. W. Butler and G. B. Finelli, "The infeasibility of experimental quantification of life-critical software reliability," *ACM SIGSOFT Software Engineering Notes*, vol. 16, no. 5, pp. 66–76, 1991.

[13] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[14] R. van Tongeren, O. Gietelink, B. De Schutter, and M. Verhaegen, "Traffic modelling validation of advanced driver assistance systems," in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 1246–1251.

[15] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations," *Vehicle System Dynamics*, vol. 44, no. 7, pp. 569–590, 2006.

[16] A. Pütz, A. Zlocki, J. Bock, and L. Eckstein, "System validation of highly automated vehicles with a database of relevant traffic scenarios," *situations*, vol. 1, p. E5.

[17] C. Roesener, F. Fahrenkrog, A. Uhlig, and L. Eckstein, "A scenario-based assessment approach for automated driving by using time series classification of human-driving behaviour," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 1360–1365.

[18] E. de Gelder, J.-P. Paardekooper, J. Ploeg, H. Elrofai, O. Op den Camp, A. K. Saberi, and B. de Schutter, "Ontology of scnearios for the assessment of automated vehicles," Submitted in 2018.

[19] Ministry of Infrastructure and the Environment, "ITS-plan the netherlands 2013-2017," https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwiZ-fen7p7dAhUGLVAKHX08BNEQFjAAegQIARAC&url=https%3A%2F%2Fec.europa.eu%2Ftransport%2Fsites%2Ftransport%2Ffiles%2Fthemes%2Fits%2Froad%2Faction_plan%2Fdoc%2F2012-its-plan-the-netherlands-2013-2017.pdf&usg=AOvVaw1B0k61dPHz-DtSXJLsbeVk/, 2012, [Online; February 13, 2018].

[20] R. Puranjay, "Development of vehicle-cyclist interaction model for scenario generation/-validation," Submitted in 2018.

[21] M. Kunert, D. Large, G. Burnett, R. Meijer, J.-P. Paardekooper, E. van Dam, F. Flohr, S. Krebs, D. Gavrila, M.-P. Bruyas, M. Gabor, G. Othmezouri, H. Stoll, A. Schneider, M. Petersson, L. Sanz, and I. Cieslik, "Vru motion and intent modelling, situation analysis and collision risk estimation," *Proactive Safety for Pedestrians and Cyclists (Prospect)*, vol. 74, pp. 38–44, 2018.

[22] S. B. Amsalu and A. Homaifar, "Driver behavior modeling near intersections using hidden markov model based on genetic algorithm," in *Intelligent Transportation Engineering (ICITE), IEEE International Conference on.* IEEE, 2016, pp. 193–200.

[23] M. Oudelha and R. N. Ainon, "Hmm parameters estimation using hybrid baum-welch genetic algorithm," in *Information Technology (ITSim), 2010 International Symposium in*, vol. 2. IEEE, 2010, pp. 542–545.

[24] A. Kurt, J. L. Yester, Y. Mochizuki, and Ü. Özgüner, "Hybrid-state driver/vehicle modelling, estimation and prediction," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on.* IEEE, 2010, pp. 806–811.

[25] V. Gadepally, A. Krishnamurthy, and U. Ozguner, "A framework for estimating driver decisions near intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 637–646, 2014.

[26] S. B. Amsalu, A. Homaifar, A. Karimoddini, and A. Kurt, "Driver intention estimation via discrete hidden markov model," in *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on.* IEEE, 2017, pp. 2712–2717.

[27] S. B. Amsalu, A. Homaifar, F. Afghah, S. Ramyar, and A. Kurt, "Driver behavior modeling near intersections using support vector machines based on statistical feature extraction," in *Intelligent Vehicles Symposium (IV), 2015 IEEE.* IEEE, 2015, pp. 1270–1275.

[28] A. Sathyanarayana, P. Boyraz, and J. H. Hansen, "Driver behavior analysis and route recognition by hidden markov models," in *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on.* IEEE, 2008, pp. 276–281.

[29] A. Pentland and A. Liu, "Modeling and prediction of human behavior," *Neural computation*, vol. 11, no. 1, pp. 229–242, 1999.

[30] J. Wu, Z.-m. Cui, P.-p. Zhao, and J.-m. Chen, "Traffic vehicle behavior prediction using hidden markov models," in *International Conference on Artificial Intelligence and Computational Intelligence.* Springer, 2012, pp. 383–390.

[31] D. Mitrovic, "Reliable method for driving events recognition," *IEEE transactions on intelligent transportation systems*, vol. 6, no. 2, pp. 198–205, 2005.

[32] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden markov models," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on.* IEEE, 2008, pp. 1189–1194.

[33] H. Berndt and K. Dietmayer, "Driver intention inference with vehicle onboard sensors," in *Vehicular Electronics and Safety (ICVES), 2009 IEEE International Conference on.* IEEE, 2009, pp. 102–107.

[34] P. Boyraz, M. Acar, and D. Kerr, "Signal modelling and hidden markov models for driving manoeuvre recognition and driver fault diagnosis in an urban road scenario," in *Intelligent Vehicles Symposium, 2007 IEEE.* IEEE, 2007, pp. 987–992.

[35] X. Zou and D. Levinson, "Modeling pipeline driving behaviors: Hidden markov model approach," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1980, pp. 16–23, 2006.

[36] K. Tang, S. Zhu, Y. Xu, and F. Wang, "Modeling drivers' dynamic decision-making behavior during the phase transition period: An analytical approach based on hidden markov model theory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 206–214, 2016.

[37] X. Meng, K. K. Lee, and Y. Xu, "Human driving behavior recognition based on hidden markov models," in *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*. IEEE, 2006, pp. 274–279.

[38] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, "A driver behavior recognition method based on a driver model framework," SAE Technical Paper, Tech. Rep., 2000.

[39] J. Yang, Y. Xu, and C. S. Chen, "Human action learning via hidden markov model," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 27, no. 1, pp. 34–44, 1997.

[40] M. Wilbrink, A. Schieben, R. Markowski, F. Weber, T. Gehb, J. Ruenz, F. Tango, M. Kaup, J.-H. Willrodt, V. Portouli, N. Merat, R. Madigan, G. Markkula, R. Romano, C. Fox, M. Althoff, S. SÃűntges, and A. Dietrich, "Designing cooperative interaction of automated vehicles with other road users in mixed traffic environments," European Council, Tech. Rep., 2017.

[41] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[42] J. Brownlee, "A gentle introduction to k-fold cross-validation," https://machinelearningmastery.com/k-fold-cross-validation//, 2018, [Online; October 27, 2018].

[43] R. Meijer, S. de Hair, J. Elfring, and J. Paardekooper, "Predicting the intention of cyclists," in *6th Annual International Cycling Safety Conference, 21-22 September 2017, Davis, California*, 2017.

[44] E. Commission, "In-depth understanding of accident causation for vulnerable road users," https://machinelearningmastery.com/k-fold-cross-validationhttps://ec.europa.eu/inea/en/horizon-2020/projects/h2020-transport/safety/indev, 2018, [Online; October 27, 2018].

[45] G. K. De Clercq, A. Dietrich, J. P. N. Velasco, J. De Winter, and R. Happee, "External human-machine interfaces on automated vehicles: Effects on pedestrian crossing decisions," 2018.

Glossary

List of Acronyms

ADAS	Advanced Driver Assistance Systems
ADS	Automated Driving Systems
AEB	Automated Emergency Braking
CSS	continuous-state system
DSS	discrete-state system
EM	expectation-maximization
GA	Genetic Algorithm
HIL	Hardware in the Loop
HMM	Hidden Markov Model
HSS	Hybrid-State System
IVS	Integrated Vehicle Safety
KDE	Kernel Density Estimation
KNN	K-nearest neighbor
PDF	Probability Density Function
SIL	Simulation in the Loop
SVM	Support Vector Machine
TTI	time to the intersection
TU Delft	Delft University of Technology
VRU	Vulnerable Road Users
WHO	World Health Organization