

Driver Modeling and Lane Change Maneuver Prediction

C.J. van Leeuwen

1460919

October 28, 2010

Master Thesis
Artificial Intelligence

Kunstmatige Intelligentie
Rijksuniversiteit Groningen

Internal supervisor:

Marco Wiering (Kunstmatige Intelligentie, Rijksuniversiteit Groningen)

External supervisor:

Bart Netten (Monitoring Systems, TNO Science & Industry, Stieltjesweg 1 Delft)



**rijksuniversiteit
groningen**



Abstract

Modeling driver behavior is a key aspect of traffic simulations. Accurate driver models can be used to study the effect of new infrastructure components, new vehicle interfaces or congestion reduction methods. The aim of this study was twofold: to model driver following behavior, and to create a classifier for predicting lane change maneuvers.

A preprocessing method using cubic smoothing splines was proposed to extract smooth vehicle trajectories from video monitoring data of highway traffic. After comparing the smoothed trajectories to reference measurements it was shown that the position information was obtained with high accuracy, but this accuracy decreased with every differentiation.

The driver models that were used are based on the Intelligent Driver Model (IDM) and the Human Driver Model (HDM) for which a Genetic Algorithm was used to find the optimal parameters for each individual driver. HDM was not able to significantly more accurately fit observed trajectories than IDM, but it did show more realistic traffic congestion patterns.

The most useful features for maneuver prediction were the lateral position and its derivatives. The longitudinal measures and information on the neighboring vehicles were shown to be far less informative, although when using a larger training set some additional features did improve the performance. The final lane change maneuver classifier was able to predict 90% of the exit maneuvers 1.44 seconds in advance with a false alarm rate of 8%. Although the classification accuracy varies for different maneuvers, in overall it was able to classify the maneuver a vehicle is in with high accuracy.

Acknowledgments

First of all I would like to express my gratitude to all people who supported me both technically and morally during my study. Without all the help of those around me it would not have been possible to complete this thesis.

I would like to thank Bart Netten as my supervisor at TNO for helping me out with all material that was eventually used, or not used in this thesis. He introduced me at the TNO department of Computer Vision and Statistics where I was later accepted for a job. I would like to thank him for helping me get started in this field of research and for the discussions we had on the subject. I also owe a lot of thanks to Marco Wiering as my supervisor at the university for his advice and the practical support for the progress of this thesis.

I would like to thank all the people at the Computer Vision and Statistics department for giving feedback on my thesis and for providing a nice atmosphere at the workspace. I am looking forward to working with you all as my future colleagues.

Furthermore I would like to thank my girlfriend for her love and understanding and for supporting my decisions. Also I want to say thanks to all of my friends for making my life as a student unforgettable.

Last but not least I would like to thank my parents for their parental wisdom and all the support they gave me throughout my study.

Contents

| | |
|--|-----------|
| Abstract | 3 |
| Acknowledgements | 5 |
| Contents | 8 |
| List of Figures | 9 |
| List of Tables | 10 |
| 1 Introduction | 11 |
| 1.1 Structure of thesis | 12 |
| 1.2 Research questions | 13 |
| 2 Literature | 14 |
| 2.1 ACC systems | 14 |
| 2.2 Driver models | 14 |
| 2.3 Gap acceptance | 15 |
| 2.4 Lane change detection & prediction | 16 |
| 3 Databases | 17 |
| 3.1 A67 | 17 |
| 3.2 A270 | 19 |
| 3.3 ENDOR | 19 |
| 4 Preprocessing | 21 |
| 4.1 Different types of smoothing | 21 |
| 4.2 Evaluation criteria | 24 |
| 4.3 Interpolation | 25 |
| 4.4 Smoothing results | 26 |
| 4.5 Differentiation | 29 |
| 5 Driver Models | 31 |
| 5.1 IDM | 31 |
| 5.2 HDM | 31 |
| 6 A270 Experiment | 35 |
| 6.1 Experimental setup | 35 |
| 6.2 Data | 36 |
| 6.3 Comparing measurements | 37 |
| 6.4 Fitting the models | 39 |
| 6.5 Resulting models | 42 |
| 6.6 Simulation | 44 |

| | | |
|-----------|---|-----------|
| 7 | Classification Experiments | 47 |
| 7.1 | Experimental setup | 47 |
| 7.2 | Lane change detection | 48 |
| 7.3 | Maneuver segmentation | 49 |
| 7.4 | Feature extraction | 51 |
| 7.5 | Building the Naive Bayes classifier | 52 |
| 7.6 | Modelling in HMM | 53 |
| 7.7 | Hybrid classifier | 54 |
| 8 | Classification Results | 56 |
| 8.1 | Feature distributions | 56 |
| 8.2 | Single maneuver classifier | 60 |
| 8.3 | Prediction time-offset | 61 |
| 8.4 | Maneuver prediction classifier | 61 |
| 8.5 | Validation on ENDOR database | 64 |
| 9 | Discussion | 66 |
| 9.1 | A270 experiments | 66 |
| 9.2 | Classification experiments | 67 |
| 10 | Conclusion | 69 |
| | References | 70 |
| | Appendices | 73 |
| A | Smoothing Results | 73 |
| B | HDM Parameters | 76 |
| C | Confusion Matrices from Cross Validation Experiments | 78 |
| D | Confusion Matrices from ENDOR Validation Experiments | 80 |

List of Figures

| | | |
|----|---|----|
| 1 | An example of a traffic situation in the Netherlands | 11 |
| 2 | A67 road layout | 17 |
| 3 | A67 snapshots | 18 |
| 4 | A270 snapshots | 19 |
| 5 | SEMA filter window | 22 |
| 6 | Loess filter window | 23 |
| 7 | Butterworth filter gain | 24 |
| 8 | Interpolation priority in preprocessing | 26 |
| 9 | Differentiation priority in preprocessing | 28 |
| 10 | Reintegrating preprocessed tracks | 29 |
| 11 | Wiener process example | 32 |
| 12 | HDM distance estimation | 33 |
| 13 | A270 experiment location | 35 |
| 14 | A270 experiment trajectories | 37 |
| 15 | Comparison of measurements | 38 |
| 16 | Velocity fitness landscape | 40 |
| 17 | Model behavior in different places on the fitness landscape | 40 |
| 18 | Acceleration fitness landscape | 41 |
| 19 | Mean squared error of IDM and HDM | 42 |
| 20 | Mean squared error of HDM model fits | 43 |
| 21 | Shockwave track simulations | 44 |
| 22 | Extended track simulations | 46 |
| 23 | Lateral position histogram for lane detection | 49 |
| 24 | Track segmentation | 50 |
| 25 | HMM state flow | 53 |
| 26 | Lateral position distributions | 56 |
| 27 | Lateral velocity and acceleration distributions | 57 |
| 28 | Longitudinal velocity and acceleration distributions | 57 |
| 29 | Right and left follower distance distributions | 58 |
| 30 | Lead distance distributions | 59 |
| 31 | HDM Model error distribution | 59 |
| 32 | The results of the binary classifier | 60 |
| 33 | Time-offset prediction performance | 61 |
| 34 | HMM classification performance | 63 |
| 35 | Validation on ENDOR database results | 65 |
| 36 | Moving average filter smoothing result | 73 |
| 37 | SEMA filter smoothing result | 73 |
| 38 | Loess filter smoothing results | 74 |
| 39 | Local Polynomial Regression filter smoothing results | 74 |
| 40 | Butterworth filter smoothing results | 75 |
| 41 | Cubic smoothing spline results | 75 |
| 42 | HDM parameter distributions | 76 |
| 43 | HDM parameter distributions | 77 |

List of Tables

| | | |
|----|---|----|
| 1 | A67 maneuver counts | 18 |
| 2 | Measures on Moving Average Filter | 26 |
| 3 | Measures on SEMA Filter | 27 |
| 4 | Measures on Loess Filter | 27 |
| 5 | Measures on Local Regression Filter | 27 |
| 6 | Measures on Butterworth Low-pass Filter | 27 |
| 7 | Measures on Cubic Smoothing Spline Filter | 27 |
| 8 | Smoothing methods scores | 28 |
| 9 | Comparison of smoothing errors | 38 |
| 10 | HMM state numbering | 51 |
| 11 | Features used for maneuver classification | 52 |
| 12 | Classification maneuver numbering | 62 |
| 13 | Naive Bayes classification performance | 78 |
| 14 | Nearest Neighbor classification performance | 78 |
| 15 | SVM classification performance | 78 |
| 16 | Hybrid classifier performance | 79 |
| 17 | HMM classification performance | 79 |
| 18 | KNN validation on ENDOR | 80 |
| 19 | Hybrid classifier validation on ENDOR | 80 |
| 20 | HMM classifier validation on ENDOR | 80 |

1 Introduction

In a society where people are traveling more and more, traffic congestion is a serious problem occurring in most countries in the world. Not only is it a daily cause of frustration for drivers, it also costs governments and companies a tremendous amount of money. Annual traffic congestion costs in the EU are estimated at over €100 billion (Europa.eu, 2008).

A lot of research is done on how to reduce traffic congestion. Of course there are many different possibilities, but the foremost solution to improve traffic flow and reduce overall traveling time is to build more roads, or broaden the existing ones. Then there are possibilities to reduce congestion by manipulating traffic flows using dynamic speed limitations or ramp metering. Another solution is discouraging people from using their cars altogether, either by increasing the cost of vehicles usage, or by improving public transport.

Now some of these methods are more successful than others, but many of them only relocate the problem to another bottleneck instead of really getting to the cause of the problem. A new approach in which vehicles communicate with each other in order to optimize traffic flow is called cooperative driving, and it's showing promising results (Ioannou, Wang, & Chang, 2007).

An important part of any research concerning traffic flow, is modeling driver behavior. A realistic driver model helps to understand how drivers react and how traffic congestion forms and how it could be mitigated. In this thesis the driver modeling process will be approached from two perspectives. In the first few chapters a vehicle following model will be created and tested which only incorporates the longitudinal behavior. In the next chapters a model will be created which focuses on the lane changing behavior.

The longitudinal models shall be trained by analyzing the results of an experiment on cooperative driving (Broek, Netten, Hoedemaeker, & Ploeg, 2010). In this experiment the vehicles communicated their acceleration with vehicles around them in order to reduce



Figure 1. An example of a traffic situation in the Netherlands

the so called shockwave effect. This shockwave effect is a form of congestion which occurs even though there is no apparent reason for it. The data from these experiments shall be used to train driver models by finding the optimal parameters for the Intelligent Driver Model (IDM) (Treiber, Hennecke, & Helbing, 2000) and the Human Driver Model (HDM) (Treiber, Kesting, & Helbing, 2006).

The lateral models shall be used to create a predictive lane change maneuver classifier. Classifiers that are able to predict whether a vehicle is going to change lanes in the short term future, might be used to improve driving assistance systems. If such a system would detect a neighboring vehicle that is going to cut in, it might respond accordingly to avoid sudden braking and provide a smoother and safer ride. The classifiers will be implemented by form of pattern recognition algorithms. Specifically the Naive Bayes algorithm, K-Nearest Neighbor, Support Vector Machines and Hidden Markov Models will be used.

In this thesis the traffic data is gathered by a traffic monitoring system called Video Based Monitoring (VBM) which extracts vehicle trajectories from video data from road side cameras (Huis, Baan, & Loon, 2008). This data shall be used to analyze the vehicles' behavior and train the models.

1.1 *Structure of thesis*

First, in chapter 2 some background literature will be discussed. Some previous work on Adaptive Cruise Control systems, driver models, gap acceptance and lane change maneuver predictions shall be summarized.

Next in chapter 3 the different databases that were used throughout the rest of the thesis will be introduced. Three different databases will be used for different purposes, so the main differences will be discussed between the databases and the situations under which the data was obtained.

In chapter 4 a preprocessing method for the raw data from VBM will be proposed in order to obtain smooth and accurate data. Various smoothing methods will be evaluated, as well as the order in which the various preprocessing steps should be performed.

Chapter 5 will explain the different driver models that are used in this thesis. The driver models are approximations on how the human drivers behave when following another vehicle.

Chapter 6 elaborates on an experiment that was held to test a system that advises drivers on how fast they should accelerate or decelerate. These experiments will provide the data from which the vehicle following models shall be created. The resulting models will also be evaluated in this chapter.

In chapter 7 the lane change maneuver classification experiments will be explained. Both the experimental setup as well as the classifiers that predict the behavior of observed vehicles will be discussed. The results of these experiments shall be presented in chapter 8.

Finally in chapter 9 the results of the experiments shall be discussed and the conclusions will be presented in chapter 10.

1.2 *Research questions*

In this thesis the problem will be addressed of how to predict driver behavior in the short term future. Specifically the following questions will be researched:

- How should VBM data be preprocessed in order to obtain accurate trajectory information on which driver models can be trained?
- Will IDM or HDM be able to simulate traffic situations that display realistic traffic congestion phenomena such as shockwaves?
- What features could be used to train a predictive lane change maneuver classifier?
- Based on these features, what method is best able to classify the lane change maneuvers, and how much in advance could they be classified?

2 Literature

2.1 ACC systems

Cruise Control systems allow drivers to instruct the vehicle to automatically maintain a constant velocity. This is primarily to increase the driving comfort, as the driver is then able to drive without having to continuously regulate its speed. Adaptive Cruise Control (ACC) extends this method as it takes into account the distance to preceding vehicles in order to improve the safety. However, studies have shown that this also influences the traffic flow. In one study on ACC it was shown that the travel times were significantly reduced, even when only a small percentage of the vehicles are using this system (Kesting, Treiber, Schönhof, & Helbing, 2008).

In other studies ACC systems were developed using longitudinal vehicle following models, which were capable of guaranteeing global stability (Zhang & Ioannou, 2004). The systems proposed in this study are specifically designed to reject disturbances and create a smooth response. In a continued study a method is discussed that combines this system with a controller in the infrastructure (Ioannou et al., 2007). These studies showed that when using this type of systems, it was possible to reduce the total time vehicles would have to spend on the road, relieving congestion significantly depending on the amount of vehicles equipped with the system.

These studies indicate that communicating vehicles is a plausible way to reduce traffic congestion. In these studies the vehicles communicate with other vehicles or with a system in the road, in order to improve the traffic flow. These cooperative driving systems can be created in a way so that they will behave in a self-organized cooperative manner (Stiller, Färber, & Kammel, 2007).

In a study by Driel and Arem (2007) a system was simulated where drivers were warned by vehicles up the road when approaching a traffic jam, and helped with stopping and starting when in a traffic jam. This system significantly improved the traffic flow as it allowed for closer time headways and smoother decelerating when approaching traffic jams.

Another study also using simulated driver models showed that Cooperative Adaptive Cruise Control (CACC) systems are indeed capable of changing the overall traffic flow, but it was noted that realistic driver models are needed (Schakel, Arem, & Netten, 2010).

2.2 Driver models

Modeling driving behavior is very important whenever some research involves traffic simulations. In a research by Bando et al. (1995) a driver model was proposed called the Optimal Velocity Model (OVM). The core of this model is a differential equation where the input is a function of the headway of the predecessors to produce an acceleration. This function has some parameters allowing the authors to fine tune the model in order to reproduce some features of traffic flow quite accurately. Specifically they are quite well in predicting the critical car-density between free and congested flow.

A complete and comprehensive driver model incorporating both the acceleration model as well as the lane change model is described in a research by Ahmed (1999). This study combines different driver models for different situations. When validating the proposed combined model it performed much better than any of the individual models, and

significantly better than the Microscopic Traffic Simulator (MITSIM) (Yang & Koutsopoulos, 1996), which was used as a reference model.

Pellecchia, Igel, Edelbrunner, and Schoner (2005) created a driver model using attractor dynamics. Their model involves a set of parameters which they show are very suitable to be found using evolutionary algorithms. They also show that this allows them to have models using different strategies such as cautious driving or aggressive driving.

Another interpretation of driver modeling is to model the cognitive processes that occur in the human driver. These type of models can be used to obtain more insight in the human aspect of the driving, but it is also possible to predict when people will accept gaps to change lanes (Absil, 2008; Hlavacek, 2010).

In the human-kinetic model an acceleration integral is used to model driver behavior (Tampère, 2004; Tampère, Hoogendoorn, & Arem, 2005). It incorporates driver variability meaning that drivers will not all behave in the same way when put in the same situation. This quite complex model is capable of producing a wide range of congestion phenomena such as stop and go shockwaves, wide traffic jams and homogeneous congested traffic. The authors claim that without the inter-driver differences, the congestion dynamics can not be correctly modeled.

The Intelligent Driver Model (IDM) was developed by Treiber et al. (2000) and is also in principle a differential equation for the acceleration as a generalization of the Optimal Velocity Model. The vehicles' own velocity, time headway and the velocity difference with the predecessor are used to determine the acceleration of a vehicle. For this it uses a set of "logical" parameters such as the desired velocity in free flow, the minimum and maximum acceleration, the desired safety distance and the minimal distance to the predecessor. It turns out this model is quite successful in predicting the traffic flow in both congested and in free flow states. Using a distribution of parameters observed in real life it can be used to model different types of trajectories (Kesting & Treiber, 2008). IDM will be explained in more detail in chapter 5.1.

In a follow-up research to IDM the Human Driver Model (HDM) was created, which in fact is a meta-model to include human reaction times, estimation errors and both temporal and spatial anticipations (Treiber et al., 2006). This allows for creating traffic flows that are even more realistic, in that IDM seems quite predictable, and HDM is much more lifelike. The theory on HDM will be explained more elaborately in chapter 5.2.

2.3 Gap acceptance

Lane changes can be divided into two categories: Mandatory Lane Changes and Discretionary Lane Changes (Ahmed, Ben-Akiva, Koutsopoulos, & Mishalani, 1996). Mandatory Lane Changes (MLC) occur when a driver has to change lanes in order to take an offramp or on an intersection, simply because they have to go in that direction. Discretionary Lane Changes (DLC) occur when drivers are not satisfied with their current lane. In the article the author elaborates on the gap acceptance behavior when changing to a new lane. The resulting model that handles MLC and DLC separately is able to capture the dynamics of the lane changing decision process.

In a study using a large dataset created by the NGSIM community (Next Generation SIMulation), Bham (2009) also validated a model on gap acceptance behavior. They obtain what they call the critical time gap distributions using maximum likelihood estimation.

The critical time gap is defined as the threshold for a gap to just be accepted or rejected. The leading critical time gap was found to be 1.17 second with a variance of 0.82 and the trailing critical time gap was 1.26 seconds with a variance of 0.87.

2.4 Lane change detection & prediction

Salvucci, Mandalia, Kuge, and Yamamura (2007) describe a cognitive driver modeling approach that is very suitable for lane change detection. In this research the authors create a system that compares observed data with a model prediction. The internal situation of the ACT-R model that most accurately explains current and past observations, is then used to classify whether the observation at hand is in a lane change maneuver or not.

A study on sequence modeling of driving behavior showed how driving behavior could be segmented into small *drivemes*, similar to *phonemes* in speech recognition (Torkkola, Venkatesan, & Liu, 2005). These drivemes can then be used to model driving maneuver sequences using a Hidden Markov Model implementation.

In a study on driving maneuver recognition and prediction by Dagli, Brost, and Breuel (2003) a Dynamic Belief Network was used to predict future actions of drivers. These future actions can then be used to improve existing ACC systems by recognizing vehicles that are about to cut-in in front of another vehicle (Dagli, Breuel, Schittenhelm, & Schanz, 2004). The system was able to recognize a cutting-in vehicle approximately one second before it actually changed lanes.

Minimizing Overall Braking Induced by Lane-changing (MOBIL) is a method to determine if a vehicle is likely to change lanes (Kesting, Treiber, & Helbing, 2007). It uses a vehicle following model to predict the action of the surrounding vehicles if it would perform a lane change and if it wouldn't perform a lane change. The authors claim that a vehicle will perform the action that leads to the least amount of braking for all vehicles. The system would therefore be able to predict lane changes when another vehicle is approaching from the back, or when the vehicle itself is going to overtake another vehicle. Using thresholds they were able to model vehicles that behave in an egoistic, an altruistic or in any intermediate manner.

Finally in a study by Dubelaar (2009) on predicting the short term behavior of a driver, a lane change maneuver prediction system was developed using a Bayesian network. The validation method was quite computationally expensive however, so the validation was done on only 8 trajectories. In this thesis a similar approach will be used, but with methods that perform faster, to create the possibility to develop a real-time application.

3 Databases

The data used in this thesis is almost all obtained by a system called Video Based Monitoring (VBM) (Huis et al., 2008). It is a roadside based monitoring system, which uses cameras to observe vehicles. The video files it captures are stored on a hard drive and processed later off-line. Current developments allow the videos to be processed real-time allowing for insight on live traffic congestion or other events.

In still images the locations and widths of the vehicles are detected and stored. The length is not available in the database, but it can be recovered from the video files. For this thesis however the length is not very important. The detections of one vehicle in multiple image frames are then put together to form a trajectory. Trajectories that last fewer than a certain amount of frames are discarded, as they are most likely to be false detections as true detections should be obtained during a longer range of time.

To analyze longer trajectories, the results of sequential cameras can be put together to form a single long trajectory. For this a combination of heading and predicted position is used, but also a fingerprinting method using SIFT features (Renkema, 2009). Using this method it is not even required for the cameras to overlap, but of course then the behavior in between two cameras has to be approximated. In this thesis the following datasets will be used.

3.1 A67

The dataset of the A67 consist of four times 15 minutes of traffic on the Dutch highway near Venlo. The trajectories are the combined results of four cameras placed along a piece of highway, together covering approximately 700 meters. In figure 2 you can see the layout

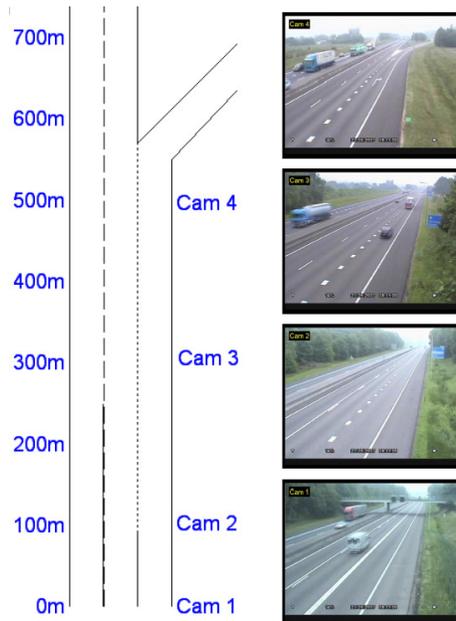


Figure 2. The road layout of the A67 database. Figure from (Renkema, 2009)



Figure 3. Two examples of what a camera has recorded in the A67 dataset. Note that the situation with back light has a very different contrast. This will lead to poorer detection.

of the road. There were three lanes: the rightmost lane being a combined entry/exit lane, the middle lane being the “normal” lane and the leftmost lane the “overtaking” lane. Note that in the Netherlands you are only allowed to overtake other vehicles on their left side, so therefore the leftmost lane is the only one where vehicles are allowed overtake to other vehicles.

All of the recordings were made in May 2007, with no obvious variation in weather conditions, except the sun creating back light in one of the recordings, see figure 3. The traffic density varies somewhat during the recordings, but there are no real traffic jams or any other noticeable forms of congestion. There is one truck which stands still on the side of the road for a long time and is therefore removed from the dataset as the passing vehicles should not be considered as overtaking it.

This database is a recording of “natural” driving behavior, thus it contains various maneuvers. In table 1 some statistics on the different types maneuvers are shown.

| Event | Observations | Percentage |
|----------------------------|--------------|------------|
| All trajectories | 1284 | 100% |
| All maneuvers | 795 | 62% |
| Overtaking | 38 | 3% |
| Returning to middle lane | 104 | 8% |
| Merging | 39 | 3% |
| Exiting | 496 | 39% |
| Exiting from leftmost lane | 118 | 9% |

Table 1: The amount of different maneuvers observed in the A67 dataset

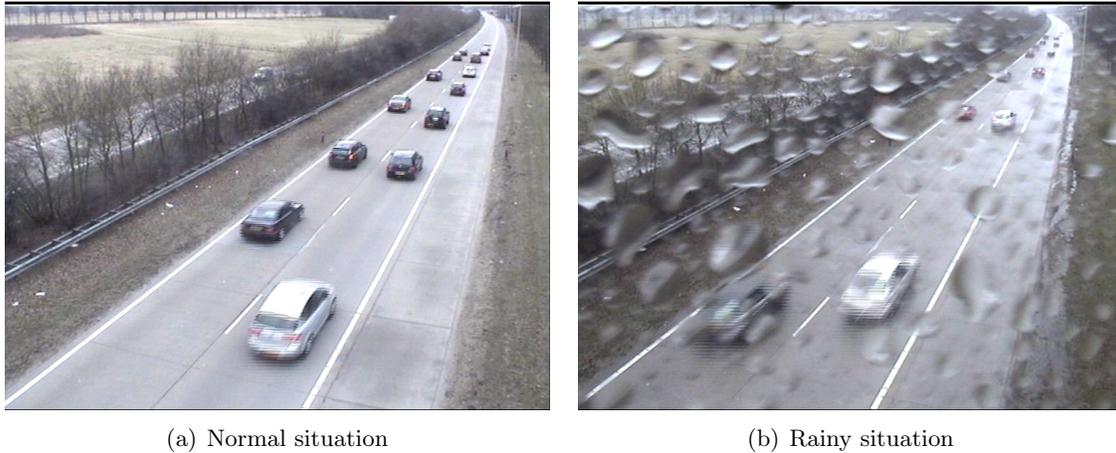


Figure 4. Two examples of what a camera has recorded in the A270 dataset.

3.2 A270

The A270 database consists of video material taken during 17 experiments divided over three days. The experiments were on the effect of a system that suggests drivers on how to accelerate to damp out traffic density shockwaves. This experiment is more elaborately described in chapter 6.

The tracks are observed by 20 camera's which are all approximately one hundred meters apart, so in all they cover two kilometers of the Dutch highway. The setting of the experiment was on the main highway between the cities Helmond and Eindhoven. About one hundred participants partook in the experiment which were separated in an experimental group and a control group. Therefore there are over a thousand trajectories available. The participants were instructed to not change lanes during the experiment, so except for one instance in which a driver accidentally started in the wrong lane, there were no lane-changing maneuvers.

The weather conditions varied much more than in the A67 database. The first two days were partly cloudy to cloudy with no significant wind, but the last day it was raining severely and quite windy. The camera view was obscured by raindrops on the lens, and the image shook significantly due to the wind. For an impression of how severe the impact of the weather was, see figure 4.

3.3 ENDOR

From other research projects, there is a very large database containing months of video material. The videos in the database are recorded on two Dutch highways the A12 and the A58, on 5 and 3 different locations respectively. The data was obtained on differing dates and times for the different locations, but all of it was between March and December 2009.

All of this video data is processed by the video based monitoring system, so a lot of data is available. However, the cameras used for this database are too far away from each other to allow for accurately merging the data from two video files. Therefore the

resulting tracks are on average only 125 meters long, with only very few tracks longer than 200 meters. Because of these relatively short tracks, the data from this database will not be used to train the models, but only to validate the previously obtained models.

Since the database is so extensive, it contains most of the weather and lighting conditions you could expect within the given dates in the Netherlands. Not all the trajectories in the database will be used to verify our models, since this would take a very long time, probably without any added useful information. Therefore only a fraction of the video files will be used where the conditions were good, so that the VBM system can accurately detect the vehicle positions. The actual portion of the ENDOR database that will be used for validation contains 792 trajectories with an average duration of 5.1 seconds.

4 Preprocessing

Before any analysis on the VBM data can be done, it first needs to be preprocessed. In this chapter the following steps of the preprocessing will be discussed: smoothing, interpolation and differentiation. A few methods of smoothing will be proposed, in order to evaluate what kind of smoothing would be best to preprocess the data in the VBM dataset. The order in which the rest of the preprocessing steps will also be evaluated (interpolating and derivation of the velocity and acceleration) in order to get the most reliable data.

4.1 Different types of smoothing

There are many types of smoothing, all having different properties, and thus different pros and cons. The methods that will be discussed are the following:

- Moving Average
- SEMA Filter (Thiemann et al., 2008)
- Locally-weighted running line-smoothers (Cleveland, 1979)
- Local Linear Regression
- Butterworth Low-pass Filter
- Cubic Smoothing Spline

Moving average The moving average filter is a very common way of smoothing data. It simply involves iteratively taking a predetermined number of sequential points around a data point and using the average value of the sequence as the smoothed data point. It has the advantage that it is very intuitive, and is very fast to implement. Outliers are definitely reduced, but their effect spreads out to neighboring data points. Also, because data points are taken completely into account from one point to the other, the resulting line may still be jagged.

SEMA filter The Symmetric Exponential Moving Average (SEMA) filter as described by Thiemann et al. (2008) is based on the moving average filter, except that it uses an exponential window to weigh the points near the smoothed data point. This means that points near the edge of the window have less effect than in a regular moving average filter, and this method therefore results in a smoother signal.

The window they use is a symmetrical exponential window, which compares to a triangular window, but it decreases in an exponential manner, therefore points near the center of the window are weighed less than in a regular triangular window, but the points near the edge are weighed somewhat more. An illustration of the window can be seen in figure 5.

The smoothed version of a data point x at time t_i where $i = 1 \dots N$ can be calculated according to the formula shown in equation 1 where Z is the normalization factor shown in equation 2.

$$\tilde{x}(t_i) = \frac{1}{Z} \sum_{k=i-D}^{i+D} x(k) e^{-|i-k|/\Delta} \quad (1)$$

$$Z = \sum_{k=i-D}^{i+D} e^{-|i-k|/\Delta} \quad (2)$$

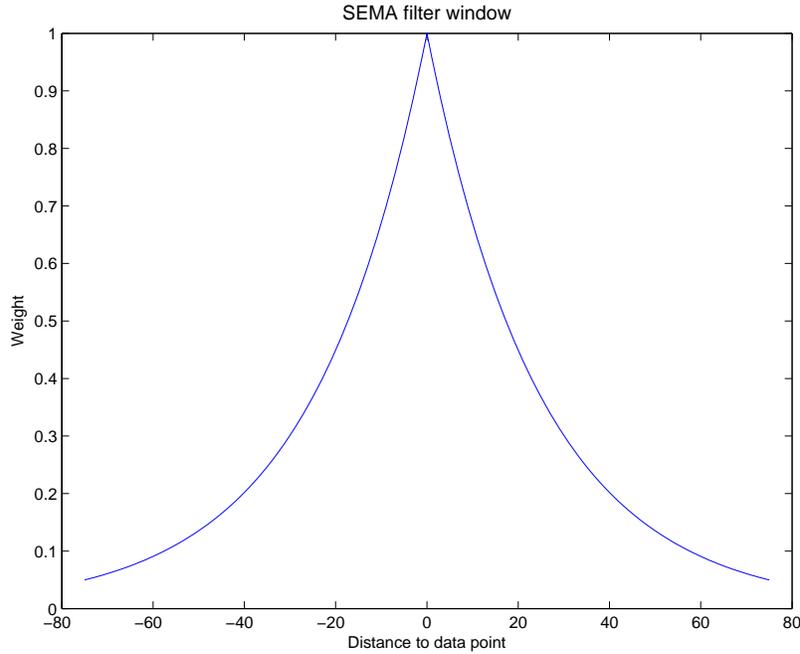


Figure 5. The shape of the window as used in the SEMA filter

In these equations 1 and 2 Δ is the base of the smoothing width given by $\Delta = T/dt$ where T is the smoothing time in seconds and dt is the period of the signal (for the VBM data in this report $dt = \frac{1}{25}$ s). And finally D is the maximal width of the exponential kernel given by $D = \max(3\Delta, i - 1, N - i)$. This basically means that it is normally 3Δ unless the window is near the beginning or the end of the signal. This is what makes sure the window is symmetrical.

Locally-weighted running line-smoothers The next smoothing method is originally described by Cleveland (1979) and is also known by the name *loess*. It is comparable to the SEMA filter, but instead of using an exponential window, it applies a *tri-cube* weighing function. Another difference is that it takes a predetermined percentage of the data, weighed according to the distance between the data points and the center of the window. After normalizing the distances u according to the furthest neighbor, it weighs the data points according to the formula shown in equation 3. The shape of this function is shown in figure 6.

$$W(u) = \begin{cases} (1 - |u|^3)^3 & \text{if } 0 \leq |u| \leq 1; \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

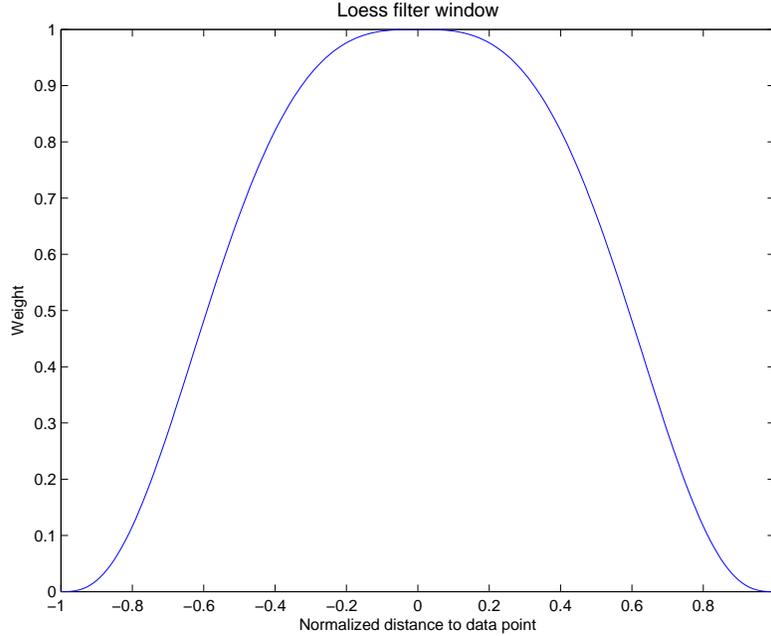


Figure 6. The shape of the window as used in the Loess filter

Local linear regression Another method is fitting a polynomial to a piece of the signal and using that polynomial, or a part of it, as the smoothed signal. Since this method applies a polynomial to the fragments of the data, apart from the window span, it needs the order of the polynomial as input. This is of large influence in the process, as a too low order polynomial might miss crucial information, but a too high order polynomial is susceptible of overfitting. In this thesis only second order polynomials are used.

A disadvantage of this technique is that just as with the moving average filter, points are taken into account from the one moment to the other causing some jaggedness.

Butterworth low-pass filter The Butterworth low-pass filter is a specific implementation of a low-pass filter (Butterworth, 1930). Butterworth filters are characterized by a magnitude response that is optimally smooth around the cut-off frequency in order to avoid any filter induced artifacts. Like a low-pass filter, it removes all components of the signal with a frequency higher than a given threshold. How sharp this threshold is, can be defined by the order of the Butterworth filter. The gain of an n -order filter, is characterized as in equation 4.

$$G^2(\omega) = \frac{G_0^2}{1 + (\omega/\omega_c)^{2n}} \quad (4)$$

In this equation G_0 is the base gain for the passing frequencies, ω is the angular frequency in radians per second normalized to the Nyquist frequency (half the sampling rate), and ω_c is the cut-off frequency. In figure 7 you can see the gain for the filters of different orders all with identical cut-off frequencies of $\omega_c = .5$. For this thesis only second order Butterworth filters will be evaluated, but with varying cut-off frequencies.

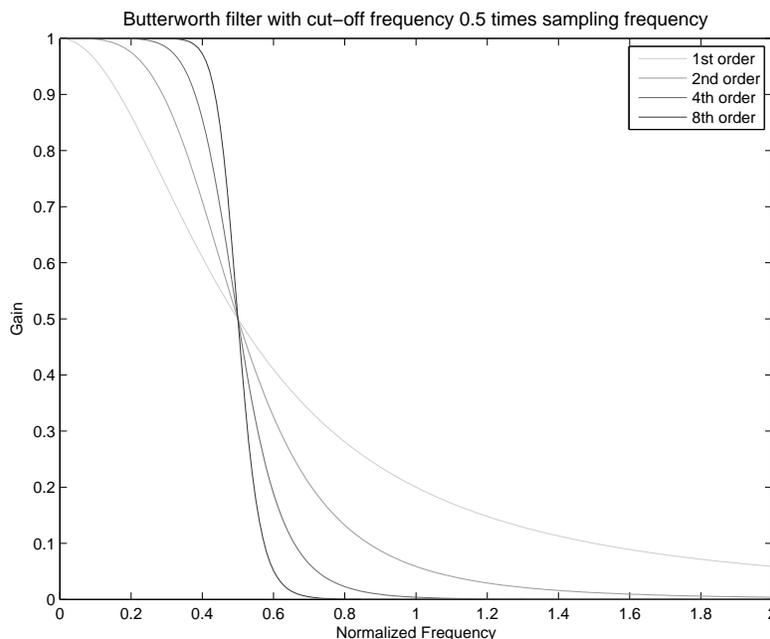


Figure 7. The gain of the different orders Butterworth low-pass filters

Cubic smoothing spline The cubic smoothing spline is a collection of splines through a set of ‘knots’. These knots can be the original data points, in which one would just get the interpolated version of the signal. However the cubic spline smoothing function allows to relocate the knots in one dimension to smooth the signal by minimizing the residual sum of squares and the total curvature as in equation 5.

$$\sum_{i=1}^N \{y_i - f(x_i)\}^2 + \lambda \int \{f''(x)\}^2 dx \quad (5)$$

In which (x_i, y_i) represent the knots, and $f(x)$ is a spline function. In equation 5 the first term penalizes large residuals, and the second term penalizes curvature. The parameter λ is the relative cost of curvature, so that for $\lambda \rightarrow \infty$ the curvature approaches a straight line, and for $\lambda \rightarrow 0$ the smoothed signal approaches the unsmoothed signal. This seems to be a highly complex problem, but as shown by (Hastie & Tibshirani, 1990), there is a unique solution for every λ .

4.2 Evaluation criteria

When comparing the different smoothing methods it is necessary to determine which one is best for this thesis. In order to do so, a set of evaluation criteria is needed. The following three criteria will be used for comparing the different smoothing methods:

- Bias
- Mean Squared Error
- Smoothness

A bias can occur if the smoothed data is on average lower or higher than the original signal. Therefore the mean of the original data will be compared to the mean of the smoothed data. If there is no bias, they should be equal. The mean squared error is somewhat similar to the bias. The mean squared error is a measure for the mean deviation of the smoothed curve at every point. It will therefore return a higher value if for instance the smoothed signal crosses the original line, whereas the bias then could still be zero. It is therefore an objective measurement of how much the smoothed function deviates from the original signal. It is defined by the formula shown in equation 6. In this equation x_i denotes the value of the original curve at point i and \tilde{x}_i is the smoothed value.

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - \tilde{x}_i)^2 \quad (6)$$

The last criterion, the smoothness of the curve, is a bit hard to measure. Ideally the smoothing function would give back a signal that is as smooth as possible. This is especially difficult since some of the smoothing functions do not return a continuous signal. Therefore the the sum of the angles between all of the line segments of the smoothed curve will be used. Since the tracks will all be interpolated at every time step, they will have the same number of segments, and the sum of the angles will give information about how jagged a curve is. In all of these criteria a lower value indicates a better smoothing function.

4.3 Interpolation

When smoothing the data, it will have to be interpolated first. This is because the functions are not able to handle series with irregular time intervals, except for the cubic smoothing spline. This chapter is therefore not relevant to the cubic smoothing spline. This seems to be problematic, because interpolation between outliers would cause the outliers to get a large influence. However, it is not true that because of interpolation all the outliers will gain influence, only the ones that lie in the part of the signal where there are few observations to begin with.

Outliers that have direct neighbors will not cause any interpolated points, and will still be smoothed out. Points that lie in a sparse period will gain extra weight, but since they are the only data in that sparse period, they will have to be assumed to be correct anyway. Another reason to start with interpolation is that during interpolation any previously smoothed data would be stretched out. Two sequential points would be moved away from each other, causing a new jaggedness in the smoothed signal.

So if interpolation were to take place before smoothing there would be some side effects since the points in the dataset are spaced irregularly in time. Figure 8 shows an example of such effects. It shows the difference between a signal that was interpolated first, and a signal that was smoothed first. As can be seen the dotted line makes some irregular curves where the signal was interpolated. This is because the originally smoothed data is stretched during the interpolation process. The effect is also measurable by MSE which is slightly lower in the scenario where the interpolation was done first. In this example the Loess filter was used, but the same effect, and even to worse extents, occurs with the other smoothing methods.

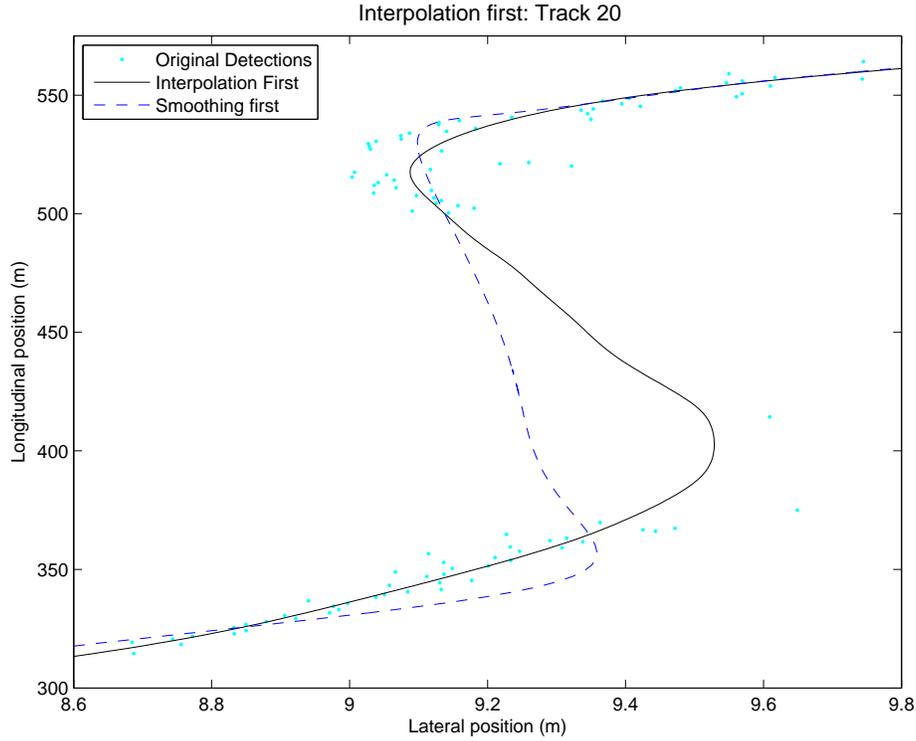


Figure 8. A closeup of a track in which interpolation took place first or smoothing took place first. The dotted line is smoothed first, the continuous line is interpolated first

4.4 Smoothing results

In the following chapter you can see the results of the different smoothing operations. In appendix A you can see the overall result of the smoothing operations as well as a closeup. The overall image provides insight of how the smoothed signal follows the original signal, and the closeup show how jagged the smooth signal is.

In tables 2 through 7 you can see the various measurements from the smoothed signals. The MSE of the y position and both biases are multiplied by 100 to gain more insightful numbers. Also note that the spans do not correlate to each other directly, they are simply a range of spans that seemed to be within the range which is interesting. This means that only the within method ratios of the measurements are relevant, but not the between ratios.

| Span | MSE x | MSE y | Bias x | Bias y | Smoothness x | Smoothness y |
|------|-------|-------|--------|--------|--------------|--------------|
| 5 | 0.114 | 0.181 | 0.032 | -0.000 | 20.880 | 6.423 |
| 11 | 0.164 | 0.235 | 0.085 | -0.053 | 10.639 | 3.131 |
| 21 | 0.188 | 0.311 | 0.637 | -0.251 | 6.516 | 1.821 |
| 41 | 0.273 | 0.484 | 2.915 | -1.015 | 4.308 | 1.208 |
| 81 | 0.657 | 1.634 | 8.787 | -3.397 | 3.175 | 0.861 |

Table 2: Measures on Moving Average Filter

| Span | MSE x | MSE y | Bias x | Bias y | Smoothness x | Smoothness y |
|------|-------|-------|--------|--------|--------------|--------------|
| 0.25 | 0.170 | 0.273 | 0.942 | -0.334 | 2.830 | 0.795 |
| 0.5 | 0.266 | 0.487 | 3.387 | -1.258 | 2.055 | 0.534 |
| 0.75 | 0.434 | 1.017 | 6.422 | -2.473 | 1.997 | 0.520 |
| 1 | 0.703 | 1.904 | 10.221 | -3.756 | 2.003 | 0.528 |
| 1.5 | 1.571 | 4.610 | 18.825 | -5.911 | 2.034 | 0.536 |

Table 3: Measures on SEMA Filter

| Span | MSE x | MSE y | Bias x | Bias y | Smoothness x | Smoothness y |
|------|-------|-------|--------|--------|--------------|--------------|
| 21 | 0.135 | 0.194 | -0.077 | 0.003 | 4.445 | 1.412 |
| 31 | 0.153 | 0.228 | -0.052 | 0.017 | 2.183 | 0.685 |
| 51 | 0.178 | 0.286 | -0.183 | 0.019 | 1.031 | 0.329 |
| 91 | 0.218 | 0.354 | 0.247 | 0.003 | 0.629 | 0.214 |
| 131 | 0.290 | 0.404 | 1.454 | -0.074 | 0.461 | 0.197 |

Table 4: Measures on Loess Filter

| Span | MSE x | MSE y | Bias x | Bias y | Smoothness x | Smoothness y |
|------|-------|-------|--------|--------|--------------|--------------|
| 10 | 0.163 | 0.234 | -1.869 | 0.085 | 12.296 | 3.239 |
| 15 | 0.170 | 0.269 | -1.006 | 0.073 | 8.604 | 2.262 |
| 25 | 0.194 | 0.337 | -1.488 | 0.191 | 5.369 | 1.336 |
| 50 | 0.293 | 0.401 | -1.217 | 0.113 | 2.664 | 0.723 |
| 100 | 0.744 | 0.962 | -2.582 | -0.208 | 1.986 | 0.591 |

Table 5: Measures on Local Regression Filter

| Cut-off Frequency | MSE x | MSE y | Bias x | Bias y | Smoothness x | Smoothness y |
|-------------------|-------|-------|--------|--------|--------------|--------------|
| 2 | 0.126 | 0.186 | -0.116 | 0.011 | 3.912 | 1.238 |
| 1 | 0.165 | 0.247 | 0.060 | 0.030 | 1.515 | 0.469 |
| 0.75 | 0.197 | 0.294 | 0.849 | 0.081 | 1.348 | 0.328 |
| 0.5 | 0.356 | 0.494 | 4.237 | 0.306 | 1.348 | 0.250 |
| 0.33 | 1.359 | 1.386 | 14.507 | 0.992 | 1.319 | 0.215 |

Table 6: Measures on Butterworth Low-pass Filter

| $1 - \lambda$ | MSE x | MSE y | Bias x | Bias y | Smoothness x | Smoothness y |
|---------------|-------|-------|--------|--------|--------------|--------------|
| 0.001 | 0.270 | 0.435 | -0.000 | -0.000 | 0.807 | 0.255 |
| 0.0005 | 0.283 | 0.459 | 0.000 | -0.000 | 0.697 | 0.224 |
| 0.0001 | 0.331 | 0.525 | 0.000 | -0.000 | 0.529 | 0.186 |
| 0.00005 | 0.371 | 0.581 | 0.000 | -0.000 | 0.470 | 0.176 |
| 0.00001 | 0.613 | 1.093 | -0.000 | -0.000 | 0.308 | 0.152 |

Table 7: Measures on Cubic Smoothing Spline Filter

| | MovAv | SEMA | Loess | Regression | Low-pass | Spline |
|------------|-------|------|-------|------------|----------|---------|
| Setting | 41 | 0.75 | 91 | 30 | .5 | 0.00005 |
| Bias | ± | - | + | + | ± | ++ |
| MSE | + | ± | + | + | + | + |
| Smoothness | - | ± | + | - | + | + |

Table 8: Relative score of the smoothing methods

In table 8 an overview is shown of how well the smoothing methods work. Note that these are not absolute measures, but rather subjective indications of how well they suit the needs for this thesis. The parameter that is shown in table 8 is the setting chosen to evaluate for that type of smoothing method. They are indicated in tables 2 through 7 as the rows with a gray background color. According to the tables and the figures it seems that the cubic smoothing spline filter best suits the needs for this study. It scores positively on all criteria, and shall therefore be used for the rest of the thesis. It also has the advantage that it does not need interpolation before filtering, which is probably also what causes the bias to be zero.

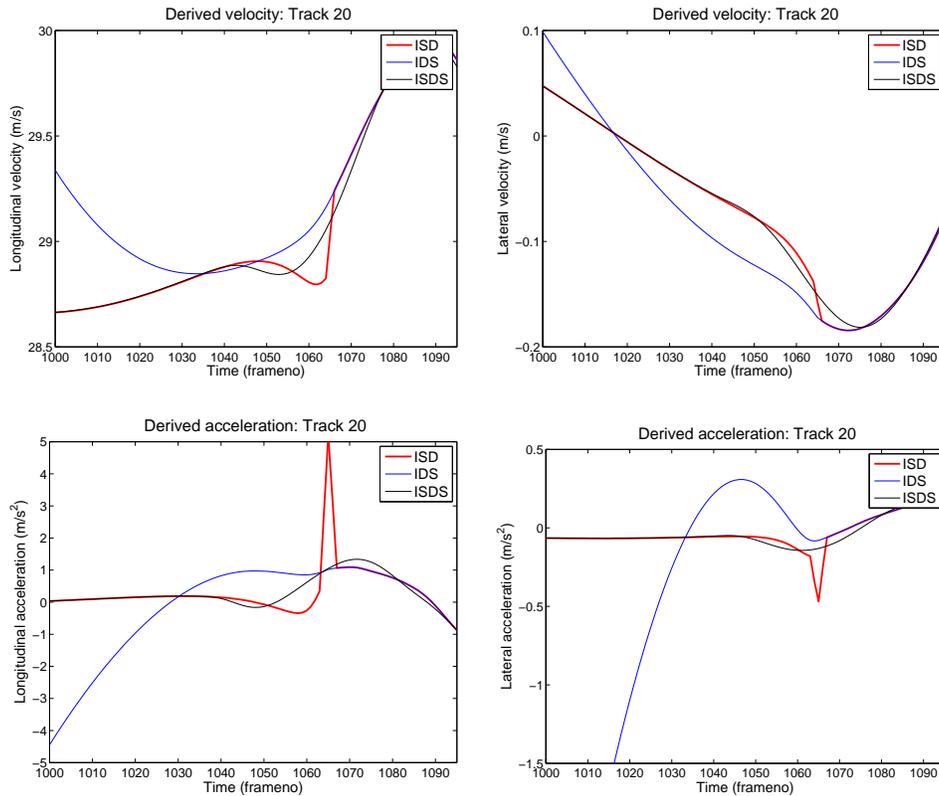


Figure 9. Differentiating the position in different orders. The letters in the legend indicate the order in which the preprocessing steps took place. I indicates interpolation, S stands for smoothing and D denotes differentiation

4.5 Differentiation

Deriving the velocity and the acceleration is the last step in preprocessing the data. As was determined in chapter 4.3 interpolation is necessary before smoothing the data, and for similar reasons the differentiation also will have to be done after interpolation. The obtained values should not be stretched out by the interpolation, which would make them incorrect. Also it is much more natural to derive a function that is defined at every point in time than a function which is defined on an irregular basis.

As a criterion for how reliable the velocity and acceleration are, reintegrating the acceleration back into the position should lead to back to the original track back. This might sound trivial, but as shall be seen it is possible that the acceleration in no way represents the actual positions that were originally measured. Another criterion for the velocity and acceleration is that the acceleration can not have very high values that would physically be very unlikely. For instance a vehicle accelerating faster than $4m/s^2$ is already very implausible for stock vehicles.

In figure 9 you can see a part of a track's derivatives. It shows three curves about the velocity and acceleration for a track. The first track is interpolated and smoothed before

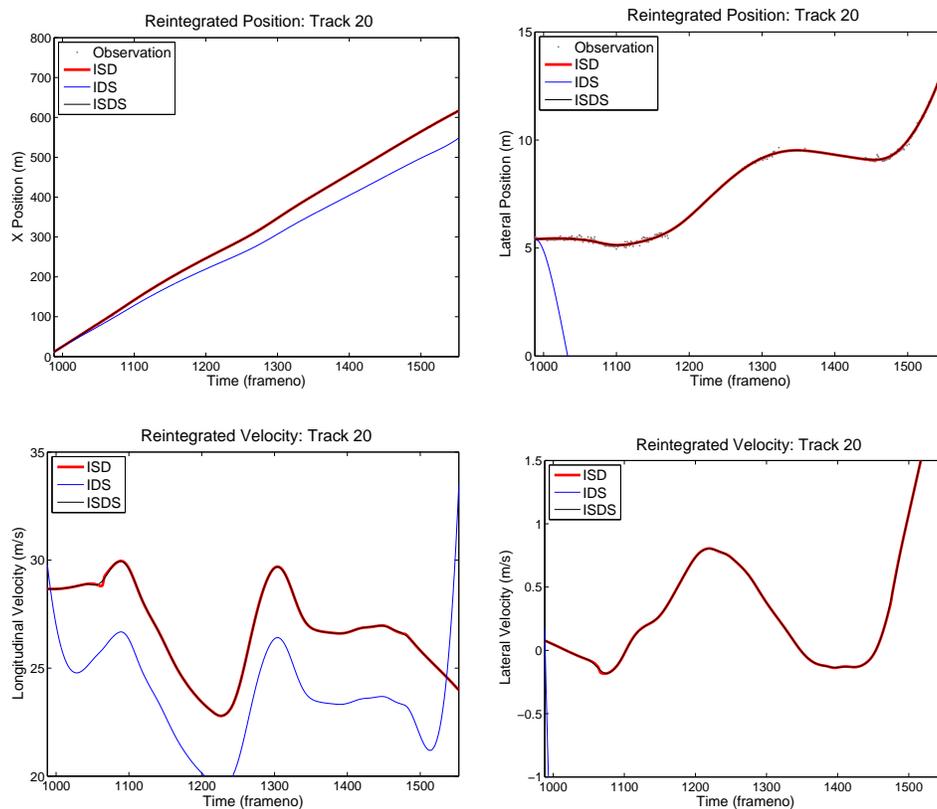


Figure 10. Integrating the acceleration back into the positions may induce some errors, but on average works pretty well. The same line styles are used as in figure 9. Additionally the original observed locations are plotted as gray dots.

differentiating, the black track is the same but smoothed again slightly in the end and the last curve is the track interpolated and then differentiated before it is smoothed. It is very clear that all of the curves agree to some extent, but some of them have acceleration values that are too extreme. So it seems that in order to obtain a plausible estimation of the accelerations you have to finish the preprocessing by smoothing the curve. However it is also required to differentiate a smoothed curve, so in order to meet both these criteria the smoothing will have to take place twice.

In figure 10 you can see the locations and the velocity as they are integrated back from the acceleration. As can be seen, the curve which represents the derivatives before smoothing, is way off. Even though the velocity profile is shaped approximately correct it is continuously estimated too low, causing a position that deviates enormously. Also note that there is essentially no difference between the curve that is smoothed once and the curve that is smoothed twice. So from this point of view, smoothing the acceleration does not really change the velocity it integrates to.

It seems a natural choice to assume the black curve in figures 10 and 9 best represents the actual velocity and acceleration, but without a ground truth there is no way of saying for sure that they are correct. In chapter 6.3 the measures from VBM data will be compared to more reliable reference measures.

5 Driver Models

5.1 IDM

The Intelligent Driver Model (IDM) is a vehicle following model proposed by Treiber et al. (2000). It is defined as the following formula for the longitudinal acceleration of vehicle α .

$$\dot{v}_{IDM}(s_\alpha, v_\alpha, \Delta v_\alpha) = a \cdot \left[1 - \left(\frac{v_\alpha}{v_0} \right)^4 - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s} \right)^2 \right] \quad (7)$$

$$s^*(v_\alpha, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}} \quad (8)$$

In which v is the velocity of a vehicle, and thus \dot{v} is the acceleration, s is the distance between the vehicle and its immediate predecessor and Δv is the velocity difference between a vehicle and its leader. In equation 8 s^* stands for the desired distance to its predecessor.

IDM uses five parameters that influence the behavior of the model. These parameters are all “logical” in the sense that they have an intuitive meaning. The parameters that can be set are the following:

a is the maximally possible acceleration.

b is the maximum comfortable deceleration. However the model can perform “emergency brakes” which decelerate faster than this value. Note that this parameter still always has to be positive.

v₀ is the desired velocity of the driver. In free flow mode it will try to reach this velocity.

s₀ is the minimum distance to the leader in meters. If the vehicles come to a complete standstill, this is the distance it will keep to its predecessor.

T is the desired time headway. In cases where the vehicle is following its leader it will follow with this time gap.

5.2 HDM

HDM is a meta model later defined by Treiber et al. (2006). Not really being a model on its own, but a meta model, it transforms the input data before it is put into a vehicle following model. In this thesis it will be applied to IDM. It takes in account the human reaction time, temporal and spatial anticipation and estimation errors. Estimation errors are modeled using a Wiener process (Gardiner, 1985), thus producing pseudo-random data.

The reaction time is included as taking a quantity not at the current time, but as that quantity a certain reaction time T' ago.

$$\chi_{t-T'} = \beta \chi'_{t-n-1} + (1 - \beta) \chi'_{t-n} \quad (9)$$

In which χ denotes either the gap between two vehicles s , the velocity of a vehicle v or the difference in velocities between to vehicles Δv . Note that χ' is the sequence of the corresponding parameter with discrete time intervals Δt . The equation is simply a linear interpolation between two observations that are about T' seconds ago. Since the reaction

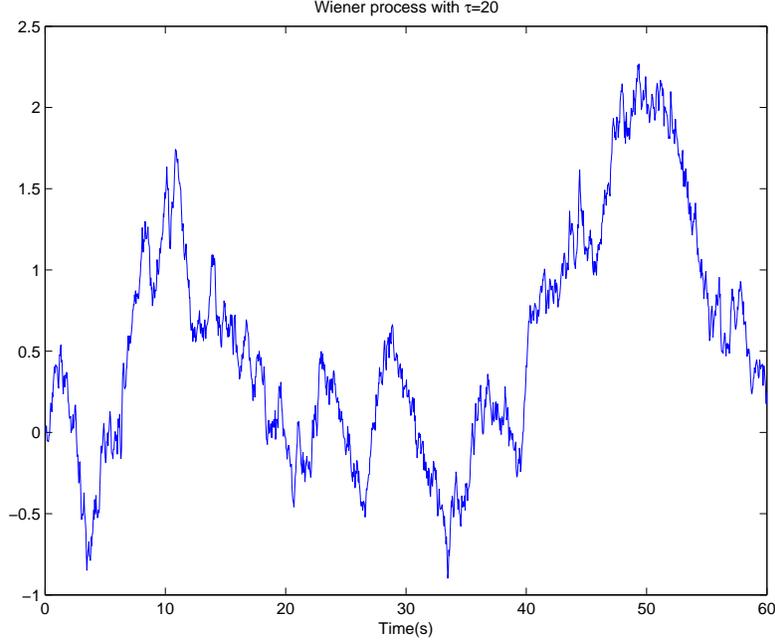


Figure 11. An example of a random Wiener process with $\tau = 20$ s

time is not necessarily a discrete amount of time steps, n and β are respectively defined as the integer and the decimal part of $T'/\Delta t$.

The estimation errors are modeled by adding a term involving an approximation of a random Wiener process. This is similar to a random walk, but its step sizes are normally distributed with a mean value of 0 and a variance of 1. The equation for the Wiener process approximation is:

$$w_{t+\Delta t} = e^{-\Delta t/\tau} w_t + \sqrt{\frac{2\Delta t}{\tau}} \eta_t \quad (10)$$

In this equation τ is the average periodicity of the Wiener process, and η is a normally distributed random variable which is realized at every iteration. An example of what such a Wiener process might look like is shown in figure 11

This Wiener process is incorporated as the w_s and the $w_{\Delta v}$ term in the following two equations for the estimation errors:

$$s^{est}(t) = s(t) e^{V_s w_s(t)} \quad (11)$$

$$\Delta v^{est}(t) = \Delta v(t) + s(t) r_c w_{\Delta v}(t) \quad (12)$$

In these equations, s^{est} and Δv^{est} are the estimated values of the distance to the leading vehicle and the velocity difference respectively. Note that the velocity v will not be estimated since the driver is assumed to directly obtain this from the speedometer. The

parameters V_s and r_c are the severities of the distance error and the velocity difference error. With increased values of V_s and r_c the traffic simulations become more unstable.

The temporal anticipation is then defined by linearly extrapolating the delayed gap and the velocity. This means that the quantities in the following equations will be used as input for the model instead of the actual distance and the actual velocity. Note that the velocity difference is not interpolated, because that would also require the acceleration of the leader, instead a zero acceleration is assumed for the predecessors.

$$s'_\alpha(t) = \left[s_\alpha^{est} - T' \Delta v \right]_{t-T'} \tag{13}$$

$$v'_\alpha(t) = \left[v_\alpha^{est} + T' a_\alpha \right]_{t-T'} \tag{14}$$

$$\Delta v'(t) = \Delta v^{est}(t - T') \tag{15}$$

The gap between two vehicles and the velocity of a vehicle are anticipated by the rate at which they changed T' seconds ago, this is the because of the assumption of constant acceleration. For the velocity difference however the model assumes a constant velocity, because the acceleration of the predecessor vehicles are not likely to be estimated by the human drivers. For a schematic figure of how the velocity is estimated see figure 12.

And then finally the model is split into two parts, namely a free flow model, and a braking action for all n predecessors, which represents the spatial anticipation.

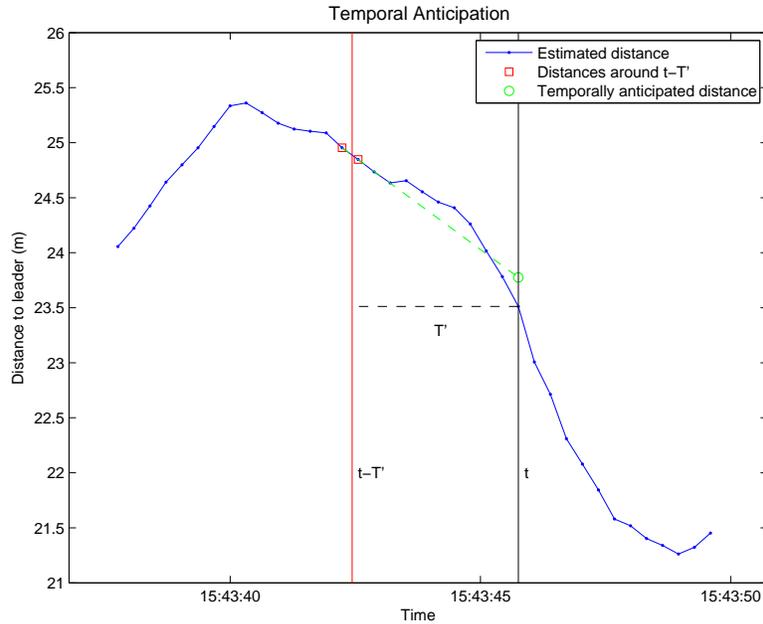


Figure 12. This figure shows how a HDM model estimates the distance to its predecessor. It takes a distance T' seconds ago, adds some estimation error, and extrapolates it T' seconds to the current moment. In this figure $T' = 3$ which is never used in this thesis, but it illustrates what happens in the model.

$$\dot{v}_{HDM} = \dot{v}_{\alpha}^{free} + \sum_{\beta=\alpha-n}^{\alpha-1} \dot{v}_{\alpha\beta}^{brake} \quad (16)$$

$$\dot{v}^{free} = a \cdot \left[1 - \left(\frac{v'_{\alpha}}{v_0} \right)^4 \right] \quad (17)$$

$$\dot{v}_{\alpha\beta}^{brake} = -a \cdot \left(\frac{s^*(v'_{\alpha}, \Delta v'_{\alpha\beta})}{s'_{\alpha\beta}} \right)^2 \quad (18)$$

In which $\Delta v'_{\alpha\beta}$ and $s'_{\alpha\beta}$ are the estimated difference in position and velocity according to the model as given by the following two equations:

$$\Delta v'_{\alpha\beta} = v'_{\alpha} - v'_{\beta} \quad (19)$$

$$s'_{\alpha\beta} = \sum_{j=\beta+1}^{\alpha} s'_j \quad (20)$$

Note that in equation 18, $s_{\alpha\beta}$ is the sum of all estimated gaps between the vehicles α and β .

6 A270 Experiment

Recently a field test was held on cooperative driving on the Dutch highway A270 between the cities Helmond and Eindhoven with 96 vehicles. The goal of this study was to analyze the effects on traffic flow of Cooperative Adaptive Cruise Control (CACC) systems and to increase public awareness of the possibilities of cooperative driving.

Since the instrumentation of such a large amount of vehicles with CACC is not feasible yet, a specific driver support system has been developed for this field test. The functionality of this system can best be characterized as an advisory form of CACC, providing the driver with a recommended acceleration or deceleration which should lead to a better traffic flow. More detailed information about this experiment can be found in the article by Broek et al. (2010).

For this thesis, it will provide insight in how the drivers behave when they are following other vehicles as both IDM and HDM models shall be fitted to the observed trajectories. It will also provide the opportunity to validate how accurate the preprocessed VBM data is since in this study there will be measurements with other systems besides VBM as well.

6.1 Experimental setup

A new test site was created as the location for this experiment. The A270 between Helmond and Eindhoven in the Netherlands was closed for these experiments, see figure 13. The experiments took place on a public road to demonstrate to the general public that these types of experiments can take place at any given location. The highway was closed to avoid non-instructed drivers during the experiment.

There were three experiment days, on the first day February 7th, 2010 there was a pilot experiment. According to this pilot the final experimental setup was determined. The two actual experiments were performed on February the 21th and 28th, 2010. On both days there were 6 runs of the experiments with slightly different conditions.

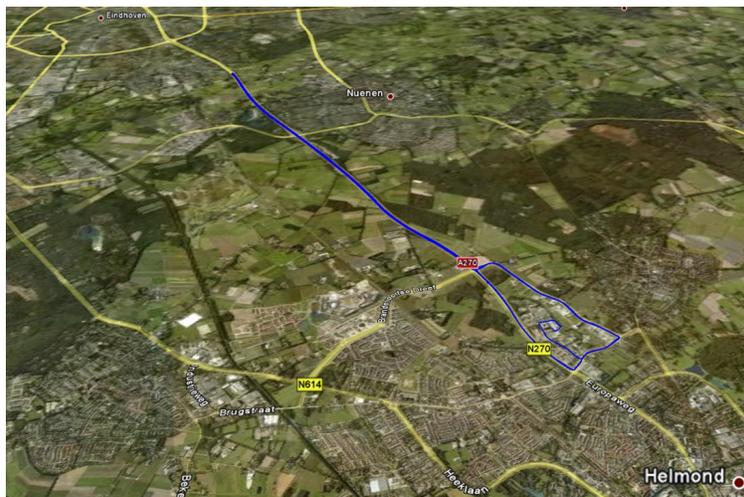


Figure 13. The location of the experiments on the A270. (Source: Google Earth)

During the experiments the participants were separated into two lanes of 48 vehicles: one lane was equipped with the advisory system and the other lane was not equipped, thus acting as a control group. The acceleration advice system consisted of a computer which calculated what the “best” acceleration was in order to damp any shockwave patterns. It did this by communicating with the units in the preceding vehicles via WiFi and analyzing their accelerations. This computer was connected to a TomTom navigation module which ran customized software so that it could instruct the driver to go faster or slower using both visual and auditory cues.

In front of both lanes was a lead vehicle which accelerated and decelerated at different locations and times to introduce shockwaves for the rest of the platoon. The exact behavior of this lead vehicle varied between experiments. In some experiments it introduced a single hard brake, in some multiple braking actions, and in other experiments it accelerated in a somewhat sinusoidal manner.

There were 96 participants who all volunteered to participate. All participants were instructed to stay in their lane, and to drive as in a normal situation as if they were just going to work. The drivers on the equipped lane were furthermore instructed to try to obey the advisory CACC system as well as possible. During the test day the vehicles remained in the same position. Between test days the position of vehicles were changed.

6.2 Data

From the experiments much data was logged by four different measurement systems:

- Visual Based Monitoring (VBM)
- TomTom
- On Board Unit (OBU)
- INstrumented CAr (INCA)

One example of the trajectories in an experiment is shown in figure 14. In this chapter the different logging methods will be explained, and in the next subsection the different measurements will be compared to each other.

The visual based monitoring system (Huis et al., 2008) is a road-side system consisting of multiple cameras that can detect and track vehicles using computer vision methods. Since both the instrumented and the non-instrumented vehicles are observed by the system, this is the main measurement method, and it will most likely yield a high accuracy in the location domain.

The TomTom data consist of location information of only a few non-instrumented vehicles. All of the instrumented vehicles were equipped with a TomTom, but the information from those vehicles are not logged in the experiments. However it did form the input of the OBU, and since that logging is available, the TomTom measurements are indirectly available as well.

The on board unit is the system that the instrumented vehicles were equipped with. So these measurements are the same as the input for the system used to advise the drivers. It contains the acceleration and velocity of its own vehicle, and from five of its preceding vehicles. However since all of the equipped vehicles are known, for this thesis this information is redundant. The OBU gets the acceleration information from an accelerometer,

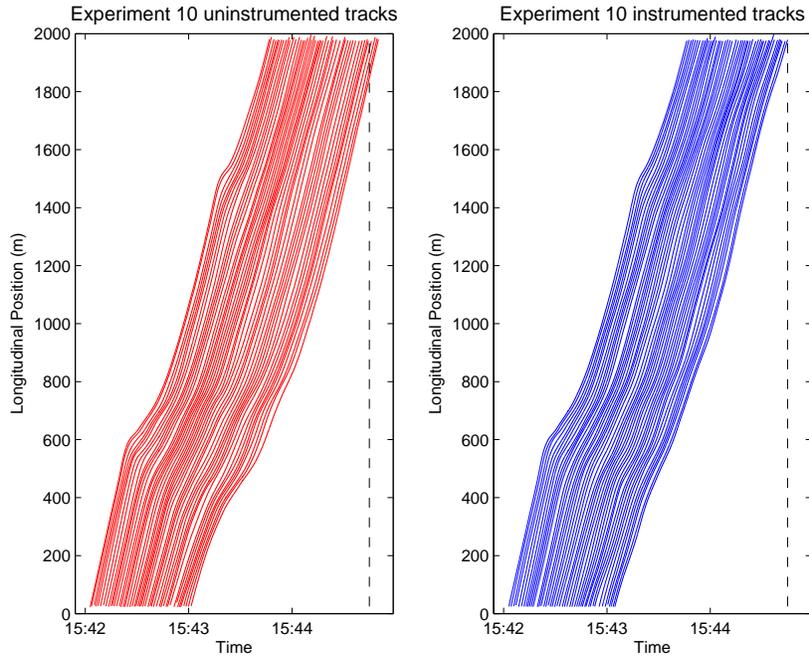


Figure 14. The position of the instrumented and the non-instrumented vehicles from an experiment as observed by VBM. On the left are the non-instrumented vehicles and on the right the instrumented vehicles. The dotted line indicates when the last instrumented vehicle reached the two kilometer point

which are corrected and filtered by means of a Kalman filter which combines GPS velocity measurements and raw acceleration data. Therefore it was assumed that these filtered accelerations logged by OBU are highly accurate.

Finally there was one vehicle participating as one of the instrumented vehicles which was instrumented with highly accurate measurement systems. This vehicle is referred to as the INstrumented CAR (INCA). It logged the location in the form of GPS coordinates much more accurately than the TomTom device does. The device that was used to measure the coordinates is called Ublox, and it is not as accurate as DGPS, but nevertheless much more accurate than the standard GPS module in TomTom. INCA logging also contains a highly accurate measurement of the velocity from a system called CAN.

6.3 Comparing measurements

Since only limited measurements of the most accurate methods OBU and INCA is available, the VBM measurements should be compared to them in order to evaluate how accurate they are. In chapter 4 the preprocessing method of the VBM data was discussed. In figure 14 all preprocessed tracks are shown for one experiment. The positions of a single vehicle are shown in figure 15(a) as they were measured by the different systems. Note that the CAN locations are actually reintegrated from the velocity just as the velocity from VBM measurements are differentiated from their position. In figure 15(b) the velocity measurements are compared and finally in figure 15(c) the accelerations are shown.

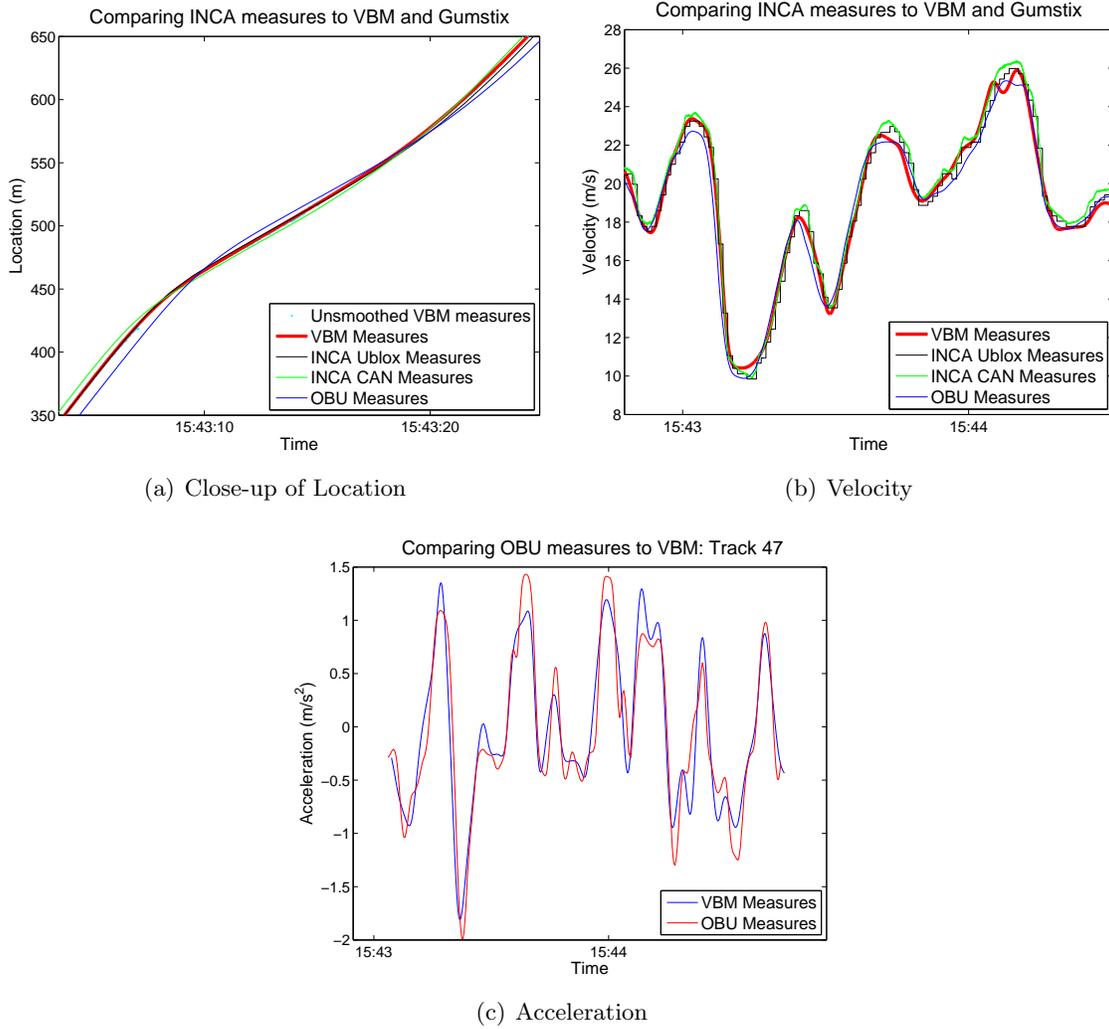


Figure 15. Comparing the measurements of different systems

| | Mean Error | RMSE | Relative Mean Error | Relative RMSE |
|--------------|---------------|-------------|---------------------|---------------|
| Location | $-0.58m$ | $1.97m$ | 0.00018 | .0031 |
| Velocity | $0.36m/s$ | $0.45m/s$ | 0.017 | .023 |
| Acceleration | $-0.016m/s^2$ | $0.20m/s^2$ | 1.25 | 1.85 |

Table 9: The errors of the VBM measurements compared to their corresponding reference measurements. RMSE is the root of the mean squared error as defined in equation 6

In figure 15 it can be seen that the position and the velocity measurements of VBM are highly accurate. The errors of the VBM measure and the reference measures are shown in table 9. The VBM position is almost indistinguishable from the Ublox measure, which was expected since the cameras are positioned next to the road at fixed positions, and thus each time a vehicle enters the field of view of a camera it has a new point of reference.

The velocity is also very accurate as it deviates only slightly from the reference velocity (CAN), however note that the peaks of the velocity profile are somewhat flattened due to the smoothing. Finally the acceleration of VBM is a bit less accurate. Differentiating the position twice and the smoothing process eventually lead to an acceleration that is not very accurate anymore, but it still follows the reference acceleration (OBU) reasonably well. The fact that the acceleration integrates back to accurate positions, as seen in figure 10, is most important for this thesis. This means that the accelerations are realistic and represent the observed positions.

6.4 Fitting the models

To analyze the vehicle behavior in how they follow their predecessors, a model will be created for every individual vehicle. These models will be analyzed in order to find some patterns in the driving behavior. The proposed driver models in chapter 5 will be fitted for every individual vehicle by changing its parameters so that the resulting model behavior best matches the observed behavior.

IDM uses five parameters to describe the driver behavior, and HDM uses nine parameters in total which all influence the model in a different way. However there are two parameters which simply create pseudo random noise in the form of estimation errors by the driver. These two parameters will be left out, and thus HDM uses seven parameters. To find the best parameter set for each vehicle a genetic algorithm (Mitchell, 1998) approach was used, to find the best parameters. In this genetic algorithm every instance is represented by a set of parameters. Instances are rewarded based on how well they are able to predict the behavior of the vehicle according to its predecessors.

In this study a pool size of 100 instances was used, and every in generation the top 10 instances automatically advanced to the next round. These winners also create six *mutants* each which are slight variations by changing the parameters randomly. The amount by which the parameters changed decreased with every generation. Additionally the top 20 instances are crossed over by incorporating a random number of parameters from another instance also in the top 20. The remaining instances are invaders, which are created by randomly choosing a set of parameters. These settings proved to maintain a good genetic variation and within reasonable time a good fit could be found.

However it is not immediately clear what the fitness function of the genetic algorithm should be. Obviously some error of the model relative to the observed track should be minimized, but what error? The most logical answer seems to minimize the location error, because it is most accurately measured by VBM. To illustrate what the fitness landscape looks like, in figures 16(a) and 16(b) the mean squared errors of the position of the best performing models can be seen. These figures are obtained by brute-force trying a large number of parameters for one single trajectory. However it is not feasible to do this for all trajectories within the timespan of this study. Even then it would still be restricted to some resolution within the problem space, but the genetic algorithm is not.

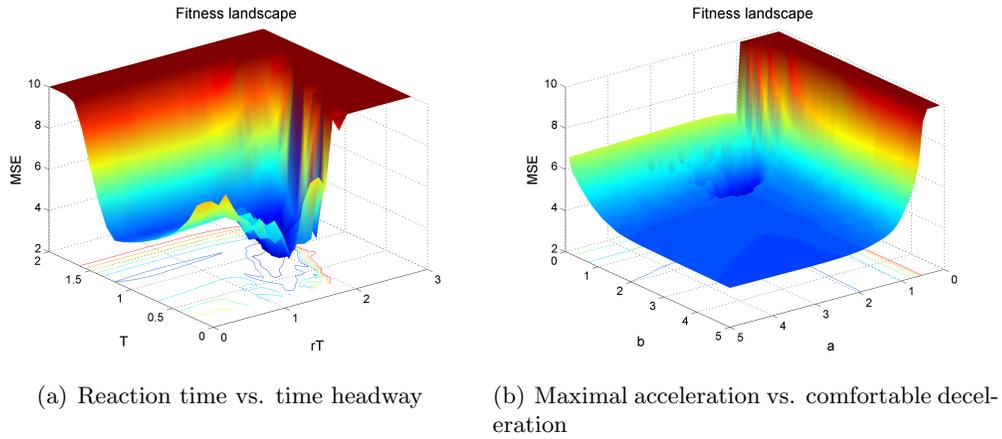


Figure 16. The fitness landscape of the mean squared error in position of the best models. The parameters not along the axes are chosen in each point to minimize the error

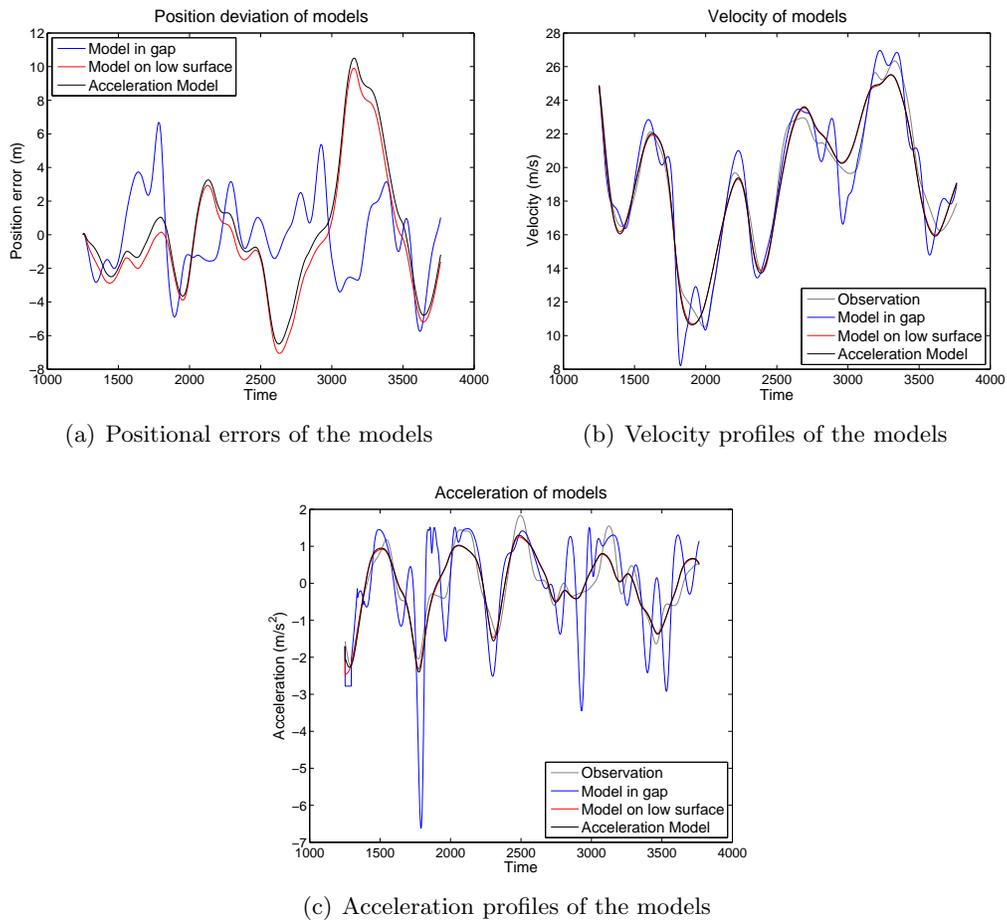


Figure 17. The behavior of a model that lies in the gaps in figure 16, a model on the lowest point of the surface in the same figure, and a model at the lowest point in figure 18.

Immediately noticeable is the sudden drop in MSE around $rT = 1.8$, $T = 0.9$, $a = 1.5$ and $b = 0.9$. It looks like the model fits the best for that set of parameters. In figure 17(a) you can see the error in meters at each time step, and it seems that especially at the end the model in the gap fits a bit better. However, according to the velocity profile of the model in the gap as shown in figure 17(b), the model in the gap changes its velocity quite quickly. This is even more obvious in the acceleration profile shown in figure 17(c).

It appears that some over fitting occurs in certain regions of the parameter space where the model really accurately explains the observed position, but to do so it needs really extreme accelerations. In fact when the parameter space is extended even further, better models can be found but the behavior is far from realistic. Obviously these are not realistic solutions, since the behavior is very unlikely. Therefore it is necessary to choose another fitness criterion.

When fitting the model according to the acceleration error, the fitness landscape looks quite differently all of a sudden as can be seen in figures 18(a) and 18(b). The drop of the error is no longer present when minimizing the acceleration error. The best performing model in this landscape is also put in figures 17(a) through 17(c), and there it can be seen that its performance is very similar to that of a model on the low surface from the landscape of the position error. Because the possibility exists that a model is found in this strange gap which would lead to undesirable behavior, the acceleration error will be minimized in the genetic algorithm.

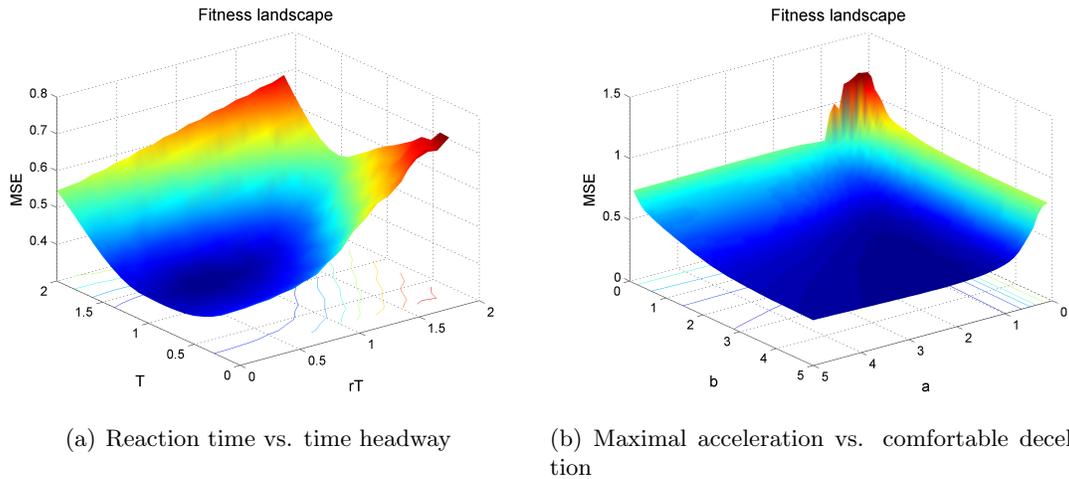


Figure 18. The fitness landscape of the mean squared error in acceleration of the best models. The parameters not along the axes are chosen in each point to minimize the error

6.5 Resulting models

The resulting model parameters can be compared in order to analyze the driver behaviors. Note that the measures are all from fitting models to individual observations, so the training and testing is done on the same data. The goal of this chapter is to analyze how well the models are able to explain the observation, not to create a generalized model from it. The resulting models give us insight in how the drivers behave in terms of the model parameters, and the corresponding MSE says something about how accurate that explanation is.

In figure 19 you can see the histogram of the MSE of all the best fitting IDM and HDM parameters. The MSE of the best IDM parameters have a mean value of 0.077 and for the HDM parameters this is 0.067. Furthermore the standard deviation of the IDM MSE is 0.041 and for HDM this is 0.037. So the differences may not be very large, but since this is an average of over 400 samples, there is some evidence that the HDM allows for a better fit than IDM.

The distribution of the mean squared errors of the best fitting model of all tracks is shown in figure 20. The values of the parameters for the instrumented versus the unin-

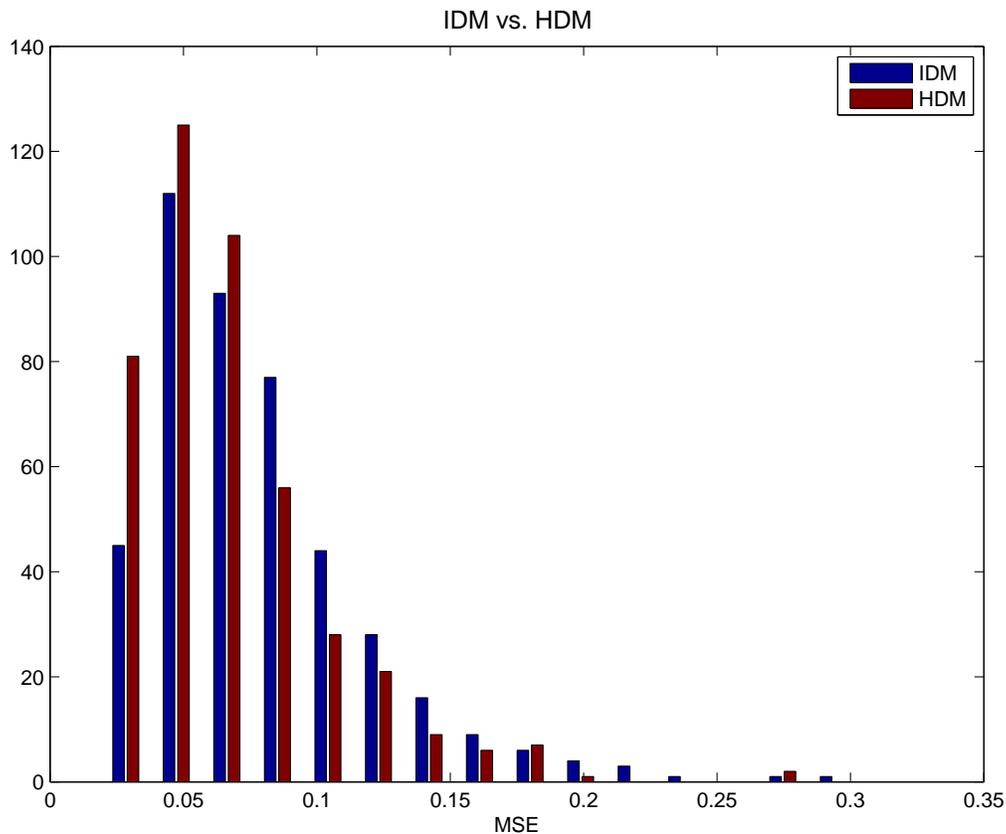


Figure 19. The mean squared error of the acceleration of all the best fitting IDM parameter sets and the best fitting HDM parameter sets. On average the HDM error is slightly lower.

strumented models are shown in appendix B. As can be seen there seems to be no clear difference in the parameters of the instrumented vehicle models and the uninstrumented vehicle models. Apparently the behavior of the instrumented drivers is not changed that much by the system in terms of the model parameters. The only parameter that does seemingly differ between the two types of drivers is the desired time headway T . For the other parameters the inter-driver variability is larger than the inter-group variability.

What also can be seen from the results, is that there is a very large spread of the parameters. In fact all attempts to cluster the data failed, as for each parameter distribution seems to be distributed all over the search space, and no significant correlations could be found between parameters. For example the minimum distance to the predecessor s_0 takes very different values from one driver to another, and even for the same driver from one experiment to another. A possible way to explain this is that the type of traffic that is observed in the experiment is too monotonous. Even though a variety of speeds and accelerations is observed, there never really is a traffic jam or something. Even though the models give a very good fit on the observed data, there is still some work to be done in order to get a good model for traffic in general.

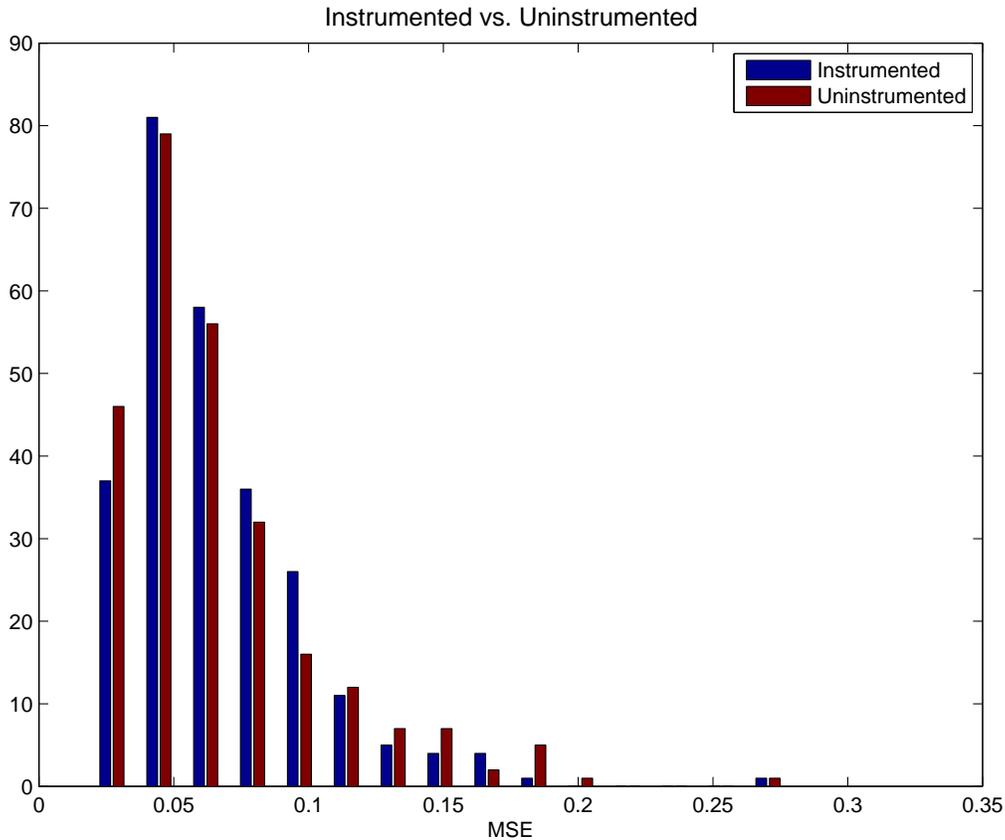


Figure 20. The mean squared error of the acceleration of the observed vehicles and the models. Note that there is no obvious difference between the instrumented and the uninstrumented tracks.

6.6 Simulation

The obtained models can now be used to simulate driver behavior similar to that in the experiment. With the correct starting conditions it should even be possible to roughly recreate the shockwaves. In figure 21 you can see trajectories of the simulated tracks as well as a the reference observed trajectories.

In all of these simulations the observed trajectory of the lead vehicle was used as the first vehicle, and models would follow in the same way as in the experiment. The models get their initial position, velocity and acceleration at the point the corresponding vehicle first entered the first camera's field of view. Since the lead vehicle in the simulation performs the same maneuver as the real vehicle, the response of the platoon should be similar. However with accumulating deviations, the models won't get the exact same input as when fitting the model. Given the same input it would perform very similar, but since there is different

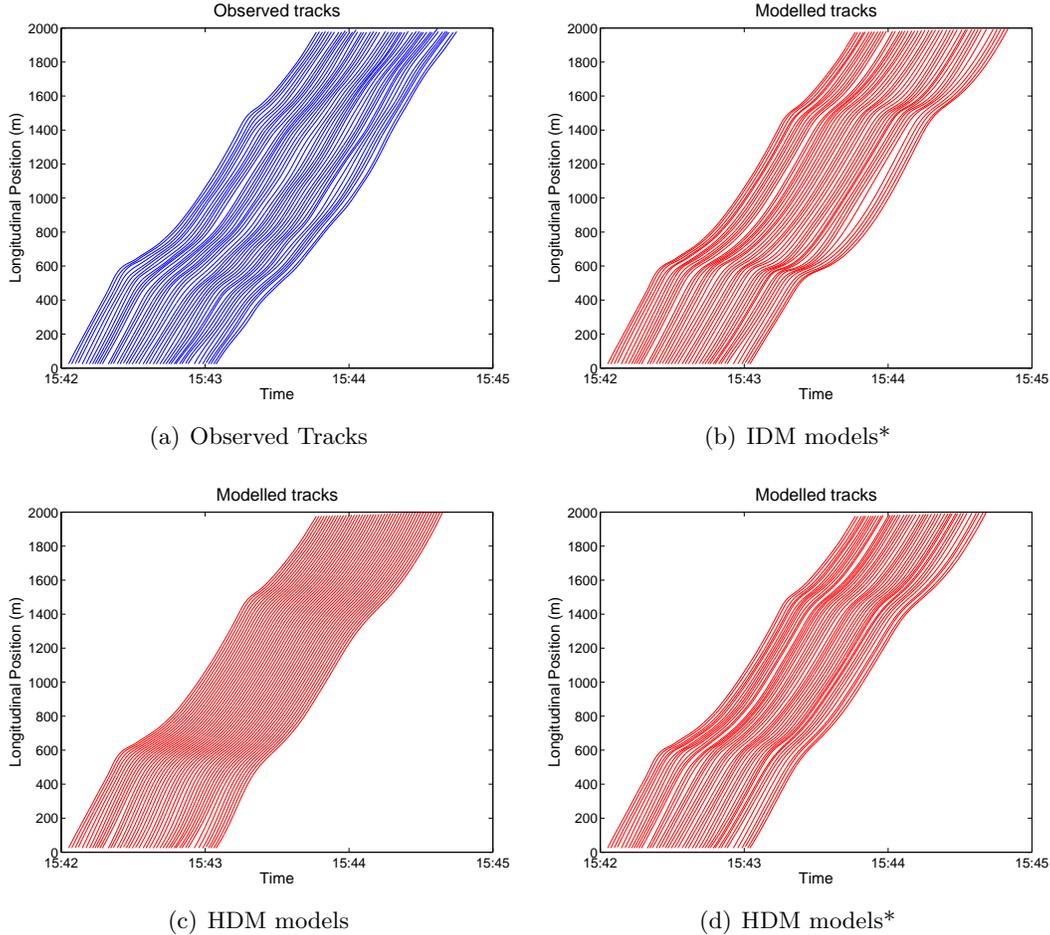


Figure 21. Simulations of the shockwave in the experiment. 21(a) shows the observed tracks, 21(b) is a simulation of IDM with for each model its best fitting parameters, 21(c) is a simulation of HDM with every model the same parameters, and 21(d) HDM with for each model its best parameters

input, there will be different output.

The trajectories in figure 21(b) are obtained by using the IDM model, with for each model the parameters that best fitted the observed track. Notice that the shockwave is no longer reduced over time. It does show some variation between the vehicles which is good, but the model lacks the ability to absorb the shockwaves. This also results in the fact that the second shock is not very different from the first shock; a phenomenon which can be seen in the reference observed trajectories in figure 21(a).

The trajectories shown in figure 21(c) are simulated by the HDM model, but now using the same set of parameters for every model. In this simulation the parameter set was used that minimized the overall acceleration error of all tracks. As you can see because all vehicles behave in the same manner, the flow is way too smooth with equal distance between vehicles. Even though the shockwave is damped to a similar degree as the observed track and the second shock has a different effect than the first one, this simulation is not very realistic for recreating human driver behavior.

Finally in figure 21(d) the trajectories are shown that were made by the HDM models and this time using the best set of parameters for each individual track. In this simulation both a dampening of the shockwaves occur and since there is a variation between the drivers there is a more realistic diversity in behavior. In the simulations of the HDM model the estimation error is implemented, and the corresponding parameters are set to the same values used by Treiber et al. (2006).

In another simulation 100 random initializations of models were used, with all parameters distributed as they were in the experiments. Then all of the models are placed in a row behind each other with an initial speed of the lead vehicle, and the initial distance to their predecessor equal to $v_0 * T + s_0$, which should be their preferred distance. Then the models all follow the lead vehicle from the experiment for the first two kilometers and after that they are “let loose”.

The results of these simulations are different every time, since the parameter distribution is random, and so are the estimation errors. An example of such a simulation is shown in figure 22. Again it is clear that the models are capable of damping out the shockwave, and there is a diversity in behavior similar to the real data. Because of the estimation error, there are also spontaneous shockwaves after approximately two and three and a half kilometers. This happens more often in other simulations, and even more extreme. In this case the driver only introduces a shockwave similar to what the lead vehicle induced, but in other simulations actual traffic jams occurred.

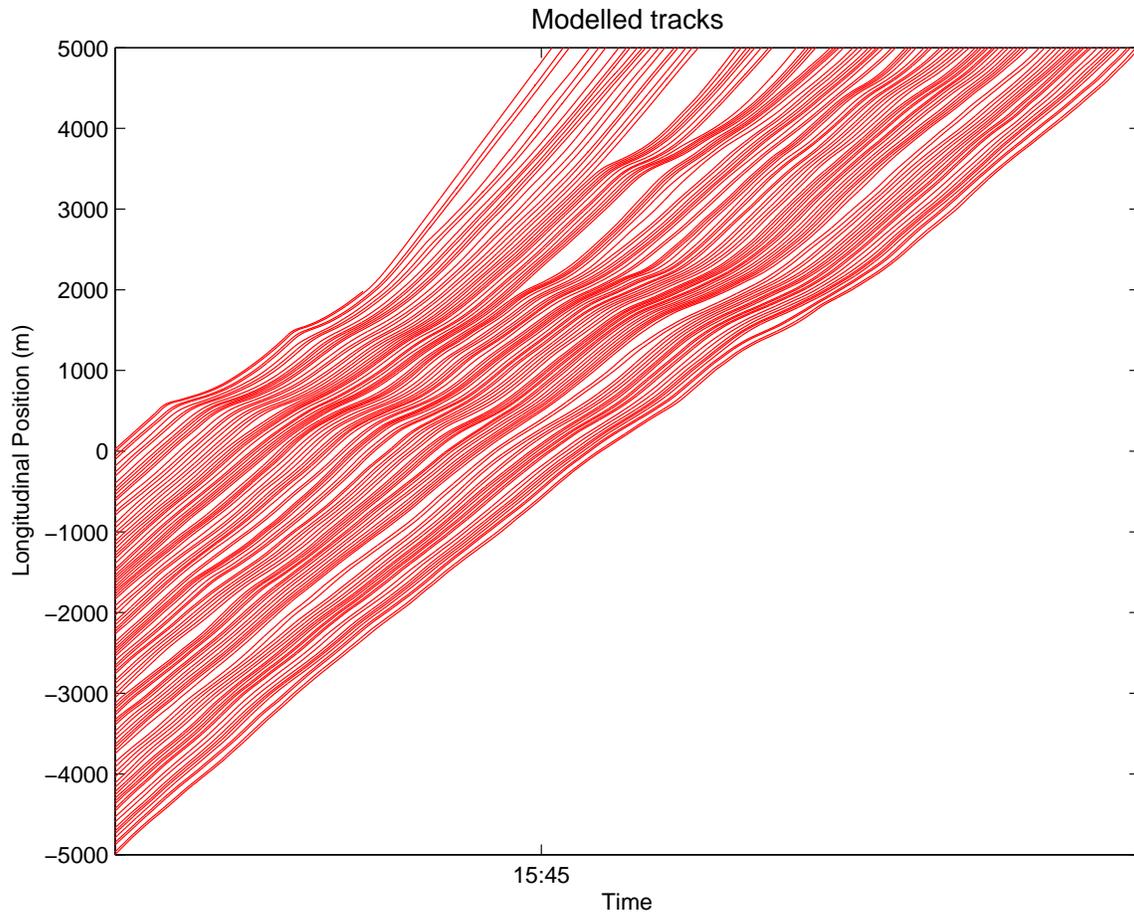


Figure 22. Trajectories of the simulated tracks when letting them continue past the two kilometer point

7 Classification Experiments

In this chapter the classification and lane change maneuvers prediction methods shall be discussed. In the first case the complete information of the tracks shall be used. This means that at any moment there will be no prediction of lane changes, but instead afterwards determining if and when an observed vehicle changed lanes. This is of course a lot easier than the prediction task, but it is necessary to obtain some *true* maneuvers which later can be predicted. After this initial classification, it will be possible to use pattern recognition techniques to predict the maneuvers. The prediction methods will be discussed in the latter parts of this chapter. First of all the experimental setup shall be explained and after that, the various stages of the experiments shall be discussed.

7.1 Experimental setup

For the majority of the experiments the A67 database is used, which has been more elaborately described in chapter 3.1. The database contains trajectory information of vehicles driving on the Dutch A67 highway near Venlo. Using the cubic smoothing spline filter as described in chapter 4, the data was preprocessed. Subsequently for each track the time and space headway to its neighboring vehicles were determined. For the final evaluation some data from the ENDOR database shall be used, for which the same preprocessing methods shall be used.

The first part of the experiment is about obtaining the “ground truth” for the actual prediction experiment. In this case the complete information of the track shall be used, and it will only be determined whether it changed lanes or not. If it did change lanes, the time at which it did shall also have to be determined. To verify its performance, the data in the A67 database was first manually classified for each vehicle based on whether it can be seen changing lanes in the video or not.

With the information about each vehicle if and when it changes lanes, a classifier can be made for each maneuver type separately. Each classifier then determines whether some example situation corresponds to a vehicle starting a lane change or not. To train these classifiers a set of example situations were chosen which are within the maneuver starting segment, and some examples of situations in the same lane but when not starting the maneuver. These examples are “snapshots”, which means they do not take into account the vehicle’s history.

To maintain a realistic setting, the ratio of maneuver to non-maneuver examples corresponds to the observed maneuvers in the training set. This means that the amount of non-maneuver examples will always outnumber the number of maneuver examples, and that there will be relatively more exit maneuvers than overtaking maneuvers. (See table 1 for the exact amount of maneuvers in the A67 database.) Features can then be extracted from the obtained examples, creating a database of targets and non targets, and some numeric values for their features.

At first a Naive Bayes classifier shall be created, this is a relatively simple initial approach, and it will provide insight in the underlying feature distributions. Therefore it will be possible to analyze the feature distributions and the intermediate probabilities. These feature distributions will later be reused to form the groundwork for a Hidden Markov Model.

Besides using a random set of snapshots to classify observations, another interesting question is how much a maneuver can be predicted in advance. In order to analyze this, the training of the classifier will stay the same, but the testing shall be done on all observations up to a certain amount of seconds before the vehicle in question actually touches the lane divider. Repeating this iteratively taking observations more in advance of the lane crossing, the prediction accuracy relative to the time-offset can be determined.

After these binary classifiers the multi-class classifiers shall be trained, which can determine at any time what maneuver a vehicle is performing. For this a toolbox called Rapid Miner is used to train and test classifiers using 10-fold cross validation. 10000 randomly selected examples are taken from our database and stored into a format readable for Rapid. In each iteration of the cross-validation 9000 observations are used to train the classifier, and the rest of the observations are used to test the performance of the classifier.

Different pattern recognition approaches will be applied, so that results of the Naive Bayes classifier, the K-Nearest Neighbor classifier and a Support Vector Machine can be compared to each other. The Hidden Markov Model that was used was not part of Rapid Miner, but the experimental setup is the same and its results will be presented in the same form. Multiple methods were used to see what approach would perform best. The particular methods that were chosen are all relatively easy to implement, but they are well known methods and can perform quite well. These methods are simple yet effective, and can provide a reliable classification with low computational complexity.

Besides cross-validation using the same database over and over, there will also be an experiment where the models are trained on the complete A67 dataset, and then the performance is determined of its ability to classify the maneuvers of vehicles in the ENDOR database.

7.2 Lane change detection

First of all, it is necessary to obtain the lateral position of the lane dividers. Inspired by the gray-level thresholding method proposed by Otsu (1975) a way of finding the lane dividers was implemented. Assuming that the most highly occupied lateral positions are the centers of the lanes, a histogram is created of the lateral positions and the highest maxima are then chosen as the lane centers. Correspondingly the lane dividers are localized just in between them. For an illustration see figure 23.

When the lane divider positions are known it is easy to determine for each vehicle at each time step in what lane it is. Finding out which vehicles changed lanes and which did not is now as simple as checking for each track whether the lane information changes or not. After checking with manually labeled tracks this simple classifier detected 95% of the labeled lane changes with a precision of 100%.

The few cases in which it missed the lane change are purely due to missing information. Some vehicles were not detected by the VBM system in the first part of the first camera image, and some were not detected in the last part of the last camera image. Now this could be solved by (linearly) extrapolating the missing information of the tracks, but this was not done for this thesis. When only looking at the actually detected parts of the trajectories, 100% of the lane changes were found.

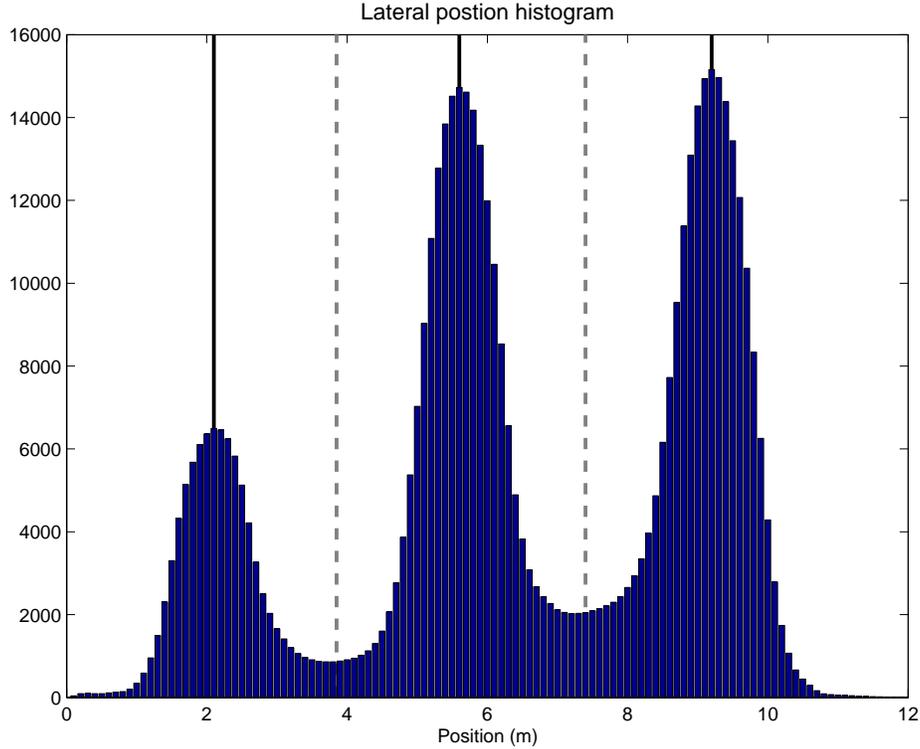


Figure 23. The histogram of the lateral positions in the A67 database. The dark lines represent the most occupied positions (lane centers) and the gray dashed lines are the positions of the lane dividers

7.3 Maneuver segmentation

Besides knowing from each vehicle whether it changes lanes, it is also necessary to know when it changes lanes, and when it started and ended the maneuver. Finding the time of the lane changes simply means finding the time step in which the lane is different from the previous time step. This time step corresponds to the moment at which the vehicle is just in between two lanes, so the vehicle's center is directly over the lane divider. This time step will be denoted by t_{change} . Using t_{change} and the vehicles width, which is provided by VBM, it is possible to find the time at which the vehicle first and last touches the lane divider with the following equation:

$$y(t_{in}, t_{out}) \approx y(t_{change}) \pm 0.5 * w \quad (21)$$

With the additional constraints that $t_{in} < t_{change} < t_{out}$. In this equation w is the observed vehicle's width. Finally the moment at which the driver started the maneuver, and at which point the driver ended the maneuver are also required. Now the moment at which the driver decides to perform the maneuver is impossible to observe from the outside. Therefore it is impossible to determine exactly from when to when the maneuver takes place. As an approximation the begin of the maneuver t_{start} is defined as the last time step

before a lane change in which the vehicle had a lateral velocity of less than some variable v_{min} . Vice versa the end of the maneuver t_{end} is the last time step at which the vehicle had a lateral velocity higher than v_{min} . A value of 0.05 meters per second was used throughout this thesis, although in other literature higher values have been proposed as well.

If a track is then segmented according to these times t_{start} , t_{in} , t_{out} and t_{end} a track segmentation is obtained as shown in figure 24. The track segments are numbered for later reference, the numbers and their description can be found in table 10.

So far the methods to classify the various lane changing maneuvers have been discussed. The point at which the driver started a maneuver can be determined using a set of strict rules. Now these classifications are stored as a ground truth of the vehicle maneuvers. These are used to compare the methods discussed in the following chapter in which a lane change maneuver will be classified, before it is actually seen by the camera. In order to predict a lane change maneuver, the vehicle should be classified as performing a maneuver if it is in the period between t_{start} and t_{in} . This corresponds to the second segment of the trajectory in figure 24 which has state number 2.

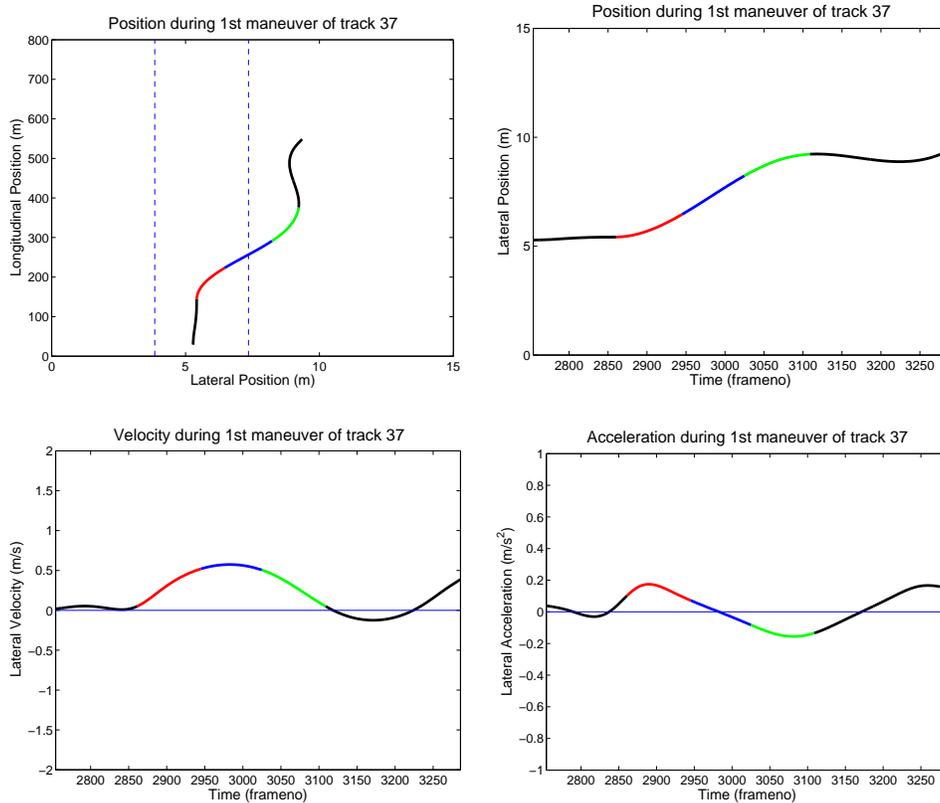


Figure 24. An example of the track segmentation: this vehicle changes from the middle lane to the exit lane.

| State | Name | Description |
|-------|---------------|--|
| 1 | follow | Following the leader (no maneuver) |
| 2 | exitStart | Starting an exit maneuver |
| 3 | exit | Performing the exit maneuver |
| 4 | exitEnd | Ending an exit maneuver |
| 5 | overtakeStart | Starting an overtaking maneuver |
| 6 | overtake | Performing the overtaking maneuver |
| 7 | overtakeEnd | Ending an overtaking maneuver |
| 8 | mergeStart | Starting a merging maneuver |
| 9 | merge | Performing the merging maneuver |
| 10 | mergeEnd | Ending a merging maneuver |
| 11 | returnStart | Starting a maneuver returning after overtaking |
| 12 | return | Performing the maneuver returning after overtaking |
| 13 | returnEnd | Ending a maneuver returning after overtaking |

Table 10: The numbering of the states used in the experiment.

7.4 Feature extraction

In order to train any predictive classifier, a set of features needs to be extracted for each instance. Each observation already contains some information directly from VBM, and the other features are derived from it. However, in the results chapter, not every feature will appear to be as discriminative as the other.

As the input of the classification algorithms, combinations of features in table 11 will be used. The first couple of features are fairly simple as they contain the lateral and longitudinal information of the location, velocity and acceleration of the vehicles. The other features all involve the position of the neighboring vehicles as well. The *neighbor* part of the features can relate to any of the six neighboring vehicles (left leader, leader, right leader, left follower, follower or right follower). Then for each neighbor the distance and the time gap can be determined, but also the time-to-collision. The time-to-collision is defined as $\frac{\Delta x}{\Delta v}$ and represents the time it would take for two vehicles to collide if they kept the same velocity. The inverse time-to-collision is also used, as this gives some measure of how dangerous the situation is (Gettman & Head, 2003).

The last feature in table 11, the model error is a very interesting one. Using the Human Driver Model, a prediction is made of how much the driver should be accelerating or deceleration taking into account the immediate predecessor (if there is any). The difference between the observed and the predicted acceleration can then be used as a feature. As parameters for HDM the parameters are used that on average fit all of the tracks best. Using an estimation of the parameters was also considered, by fitting the model to the previous couple of seconds. However, this is not very practical in terms of efficiency and it is also very prone to overfitting. This resulted in most model errors to be near equal to zero, and there was a very high variety in models from the one moment to the other, so instead one predetermined set of parameters was used.

| Feature | Description |
|-----------------------|--|
| yy | The lateral position relative to the road |
| lane | The lane the vehicle is in |
| laneOffset | The lateral position relative to the lane center |
| vy | The lateral velocity |
| ay | The lateral acceleration |
| vx | The longitudinal velocity |
| ax | The longitudinal acceleration |
| <i>neighbor.xDist</i> | The longitudinal distance to some neighbor |
| <i>neighbor.xGap</i> | The longitudinal time gap to some neighbor |
| <i>neighbor.ttc</i> | The projected time-to-collision to some neighbor |
| <i>neighbor.ittc</i> | The inverse time-to-collision to some neighbor |
| modelErr | The error of the HDM prediction |

Table 11: The various features used for maneuver classification

7.5 Building the Naive Bayes classifier

Once a set of training examples is established and the features are selected, the training of a classifier can begin. It should be able to separate the situations in which a vehicle is starting a maneuver (targets) from the non-maneuver situations (distractors). In this chapter a Naive Bayes method will be explained, because the parameter distributions are very interesting and the Naive Bayes method uses those distributions to classify the data.

First a histogram will be created of the features from the targets and the distractors. These histograms will be shown in the results chapter 8.1, as they provide a lot of insight in why some features are more suitable for separating the targets from the distractors. These feature distribution histograms will also be used for the Hidden Markov Modeling in chapter 7.6.

Discretizing the features into histogram bins in the training phase, allows to determine the probability that some feature f_i is observed given the fact whether that observation was in the target or the distractor class $P(f_i|c)$. Then, calculating the probability that an observation was either during a maneuver or a non-maneuver can be done using Bayes' Theorem in equation 22.

$$P(c|o) = \frac{P(o|c) * P(c)}{P(o)} \quad (22)$$

$$P(o|c) = \prod_{i=1}^n P(f_i|c) \quad (23)$$

$$P(o) = P(o|c_{change}) * P(c_{change}) + P(o|c_{follow}) * P(c_{follow}) \quad (24)$$

$$pred = \begin{cases} change & \text{if } P(c_{change}|o) > P(c_{follow}|o) \\ follow & \text{otherwise} \end{cases} \quad (25)$$

In which $P(o)$ stands for the probability of some observation, $P(f_i)$ stands for the probability of observing one of the n features, and $P(c)$ for the *a priori* probability for one of the classes. Equation 25 says that when the probability of changing given the observation is larger than the probability of staying in the same lane given the observation, the algorithm will predict that the vehicle is going to change lanes. Based on the application it would also be possible to implement a threshold here.

Equation 23 simply says that the probability of a certain observation given a class is equal to the product of the probabilities of all features in the observation. Here it is assumed that the features are independent given the maneuver, which is probably not really the case. This assumption is what makes the Naive Bayes classifier naive.

7.6 Modelling in HMM

Using the distribution of the features conditional to the segments of the maneuvers, it is possible to apply Hidden Markov Models to find out what the actual driving maneuvers are. Markov models can be used to model a certain series of observations that are conditional on a set of states. The maneuvers aren't directly observable so therefore the system would be in fact a Hidden Markov Model. The observations in the Hidden Markov Model are the longitudinal and lateral position, velocity and acceleration, and possibly any other features from table 11. The states are the maneuver segments from figure 24 and the corresponding state numbers which are listed in table 10. An illustration of the state flow is shown in figure 25.

Modeling the maneuvers as Hidden Markov Models allow us to calculate the probability that some series of maneuvers was performed and thereby what the most likely path is. This makes it effectively another maneuver classifier, and therefore it will be possible to compare its results to the other method. To create a model the following parameters are needed:

- The initial state distribution
- The state transition probability matrix
- The state conditional observation emission probabilities

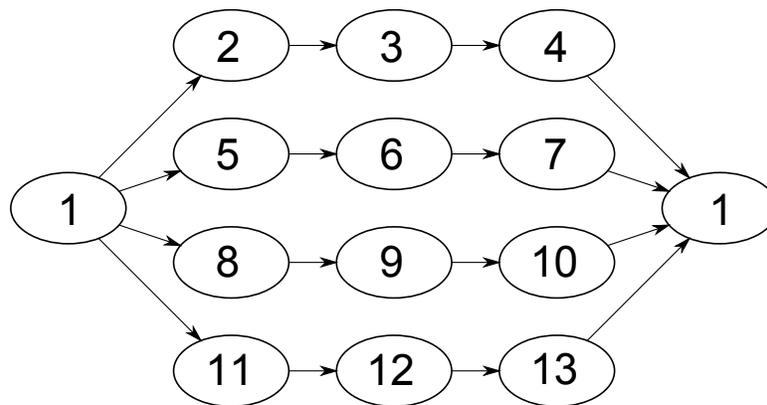


Figure 25. The Hidden Markov Model flow, the state numbers correspond to the numbers in table 10

The initial state distribution is relatively easy to obtain. Of course it would be possible to check for every track what the first maneuver segment is, but in the experiment it was simply assumed that every vehicle starts out in the state where it is not performing any maneuver. So the initial probability of state 1 is one, and of the other states zero.

The state transition probability matrix is calculated by simply keeping track of how often each state follows another. This way a matrix can be constructed of how often each state occurs after every other state. By normalizing so that the sum of all rows adds up to one, the transition probabilities are obtained.

The state transition matrix can be thresholded to some minimum value to make sure a really unlikely transition doesn't occur. If in the training phase a particular transition somehow occurs once, then under some circumstances the algorithm might suggest that this transition occurred. However, because of the way that the states are defined, there is absolutely no way that the system could go for instance from state 5 to state 7, or even from 8 to 4. In chapter 8.4 the effect of this thresholding method shall be presented.

Finally the state conditional emission probabilities is where the feature distributions come in that were calculated for the Naive Bayes classifier. Now the feature distribution for every state is needed, instead of just for one target class, so for each state there will be a separate histogram of the feature distribution. Because not every possible value will be present in the training phase, the first and the last bin of the histogram correspond to negative and positive infinity respectively, and each bin will have a minimum value of ϵ .

To find the most likely sequence of maneuver segments the Viterbi algorithm (Forney Jr, 1973) will be used. This most likely path should be the same (or at least similar) to the state segments in the trajectory. The errors of this most likely path will be used to measure the performance of the HMM.

7.7 Hybrid classifier

Finally, a hybrid classifier was built which combines the methods of rule based decision trees and Naive Bayes classification. For this classifier, knowledge is used about the road and about the definitions of the maneuvers. Additionally this classification method classified an entire track sequentially, and so to some extent it takes in account the history of the trajectory as well.

It is assumed that the lane divider position is known, and so the classifier can detect if a vehicle is getting closer to the divider, but also if it has crossed it. When the vehicle has crossed the divider, the classifier "knows" that the next couple of sequences are definitely part of the maneuver, so it can classify the instances accordingly. When it has not recently crossed a line however, the classifier uses a set of predetermined rules to identify the possible maneuvers.

What this classifier basically does is the following checks for every maneuver:

1. Is the maneuver possible according to the road layout?
2. Is there any reason for the vehicle to perform the maneuver?
3. Is there a gap large enough in the target lane?
4. Is there any evidence in terms of lateral movement for the maneuver?
5. If there are still multiple maneuvers possible, what is the most likely maneuver?

The first thing the classifier does is check what lane the vehicle currently is in. In every lane there is a maximum of three maneuvers possible: either continue in the lane, change to the left or change to the right. However in the outer lanes, there are only two maneuvers possible, so these possibilities are ruled out.

The classifier then looks at where the neighboring vehicles are. For overtaking maneuvers it is required that there is an immediate predecessor. If there is none, the vehicle does not need to perform an overtaking maneuver so that possibility is ruled out.

Next the classifier checks if there is enough room in the target lane for each maneuver. Using the feature distributions from the Naive Bayes classifier, it calculates the probability that the vehicle will perform a lane change maneuver given the proximity of the neighboring vehicles. If this probability is lower than some threshold, the maneuver is also ruled out.

It then uses a similar rule for the lane offset, the lateral velocity and the lateral acceleration. If the probability that the vehicle will perform a maneuver given the feature is lower than a certain threshold it is ruled out. It then also checks if the combined probability given these three features is lower than some threshold.

If after all these rules there are still more than one possible maneuver, the classifier predicts the maneuver that has the largest combined probability of all features. This is done in the same way as a Naive Bayes classifier would.

8 Classification Results

In this chapter the results of the experiments described in chapter 7 will be presented. At first the feature distributions will be shown, next the results of the single maneuver classifiers and finally the multi-class classifiers.

8.1 Feature distributions

First the different features shall be presented in order to show how well they allow to separate the data. Since there are 32 possible features and four different maneuvers, it would be unrealistic to analyze all possibilities, and present them all in this thesis. Therefore some interesting examples have been selected and will be presented here, however a lot more were tried out.

The first feature of interest is the lateral position. In figure 26(a) one can see the distribution of the lateral positions of vehicles that are going to change from the middle lane to the right lane and the vehicles that stay in the middle lane. As could have been expected there is a clear difference, since the lateral position defines the lane change. In figure 26(b) the distribution of the lane offset is shown. Of course this figure is essentially the same as the lateral position, except that the figure is now mirrored and centered differently. However the lane offset is very important because it is not location specific in contrary to the “global” position. This means that the trained model will be more applicable to other situations, instead of only to locations that are identical to the training situation.

The next obvious features are the lateral velocity and acceleration, in figure 27 these distributions are plotted for the same vehicles as in figure 26. Since these values are really just derivatives of the position, they offer similar information as the position distribution. In figure 27(a) there still is an obvious difference between the changing vehicles and the non-changing vehicles. This means that the velocities occurring during a lane change are far higher than the velocities during weaving when a driver stays in the lane. Also note the sharp cut-off in the changing vehicles’ velocity around $v = 0.05$: this is due to the

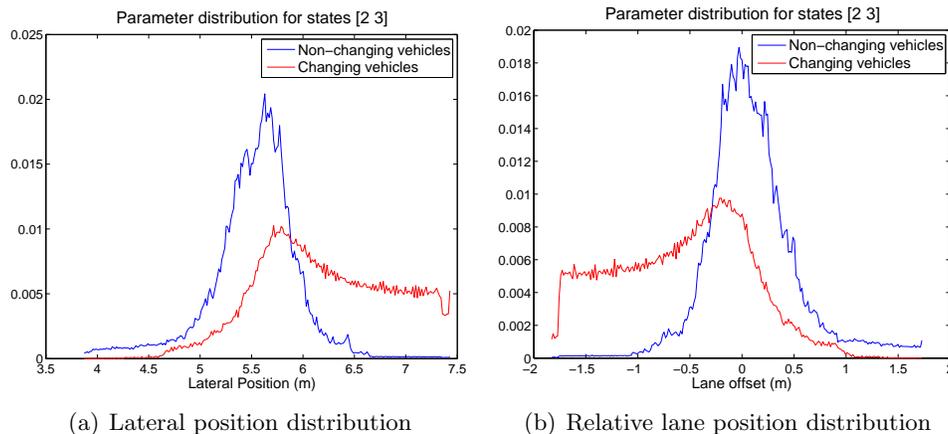


Figure 26. The lateral position albeit relative to the lane or relative to the road makes for an obvious feature to make a decision

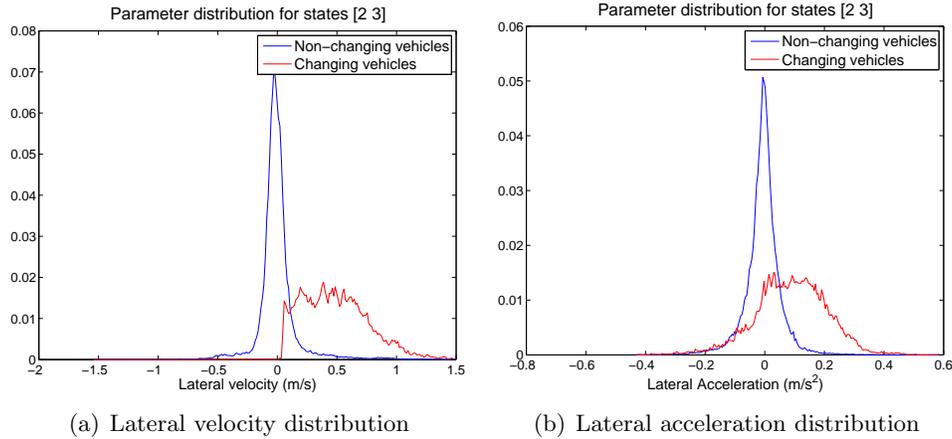


Figure 27. The lateral velocity and acceleration allow for clear discrimination between changing vehicles and non-changing vehicles

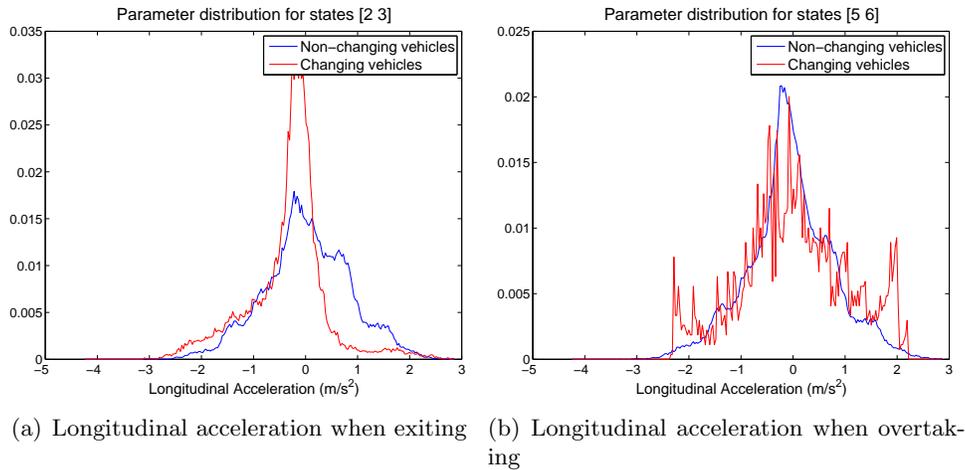


Figure 28. The longitudinal velocity and acceleration seem to be not very good for discriminating lane changing vehicles from non-changing vehicles

definition of when a maneuver starts and stops (see chapter 7.3). When looking at the lateral acceleration in figure 27(b) there is still a difference between the two distributions, but since the acceleration is yet another abstraction from the initial measurement and perhaps because it is measured less accurately, the difference is somewhat smaller.

In figure 28 you can see the longitudinal acceleration distribution of two maneuvers. In figure 28(a) you see the distribution of acceleration when starting an exit maneuver. Here there are still some differences. Apparently the drivers that change lanes are braking more than non-changing vehicles, or at least they are less likely to accelerate. Apparently it is possible to measure that they are reducing their speed in order to merge with the exit lane and to take the offramp.

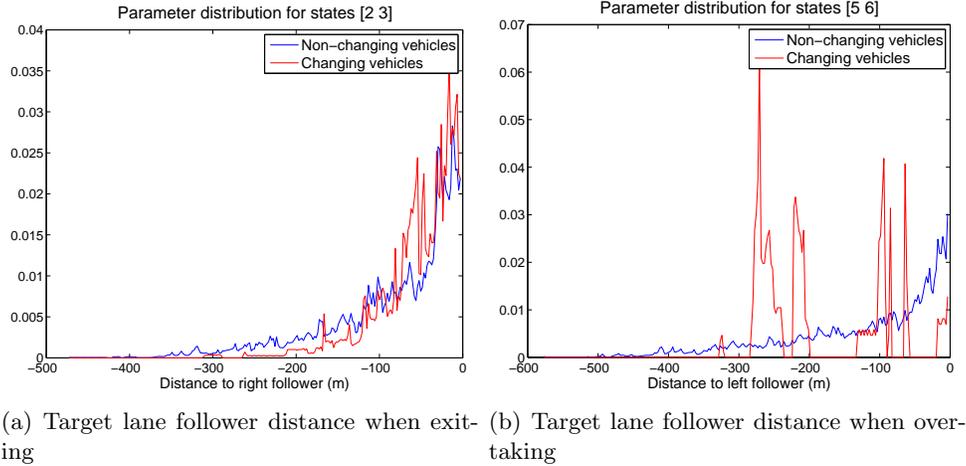


Figure 29. The right and left follower distance distributions seem not to be very suitable for discriminating lane changes from non-lane changes

In figure 28(b) you can see the acceleration when the driver initiates an overtaking maneuver. In this situation there seems to be no evidence that drivers accelerate when starting an overtaking maneuver. Apart from the more “noisy” graph, which is simply because there are less examples, the distribution seems to be nearly identical to the non-overtaking vehicles.

Now let’s take a look at the somewhat more complex features. In figure 29 the distances to the follower in the target lane are shown. The first thing to notice is that if there is some decision boundary it is far less clear than in the previous figures. In figure 29(a) distance to the follower in the target lane during an exit maneuver is shown. Here there seem to be no difference between changing and non-changing vehicles.

In figure 29(b) the same is shown but for overtaking maneuvers. Even though it is not very clear, it seems that here the distance to the neighbor is more likely to differ than when performing an exit maneuver. When initiating an overtaking maneuver, there is a smaller chance that there is a vehicle nearby in the target lane than where there is no maneuver. The fact that figure 29(b) is very irregular is due to the fact that there are far less examples of overtaking maneuvers and because there is not always a left follower present.

In figure 30(b) the distance to the immediate leader can be seen before initiating an overtaking maneuver. It was expected that there will be no need to change to the left lane if there is no immediate predecessor; the distribution confirms these expectations. The overtaking maneuvers are mostly only initialized when there is a leader relatively close to the vehicle. The same does not hold for the exiting maneuvers which are mandatory lane changes (Ahmed et al., 1996). However, there does seem to be an effect that their immediate predecessors are really close when performing the maneuver as can be seen in the leftmost part in figure 30(a). So apparently a predecessor within about 30 meters is some evidence that the vehicle will perform an exit maneuver.

Finally in figure 31 the error of the HDM prediction is plotted. Unfortunately it seems that there is no evidence that changing vehicles behave less like the model than

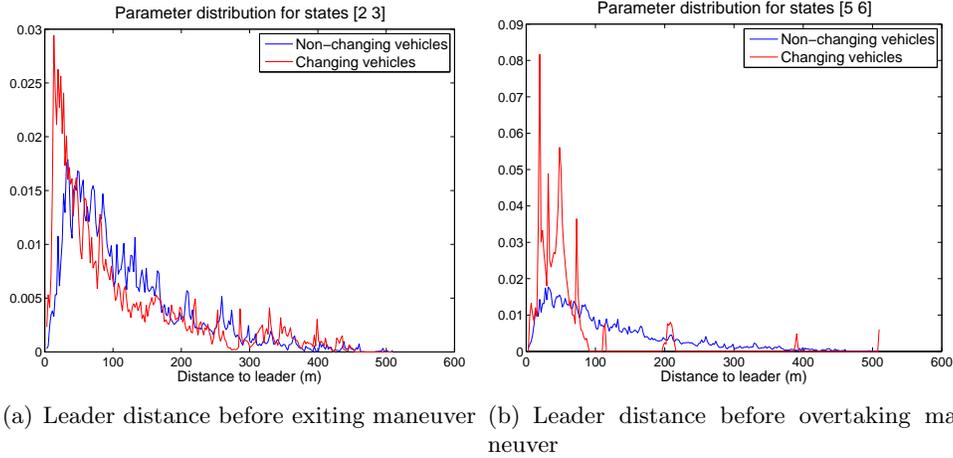


Figure 30. The distribution of distances to the immediate leader just before initiating a lane change

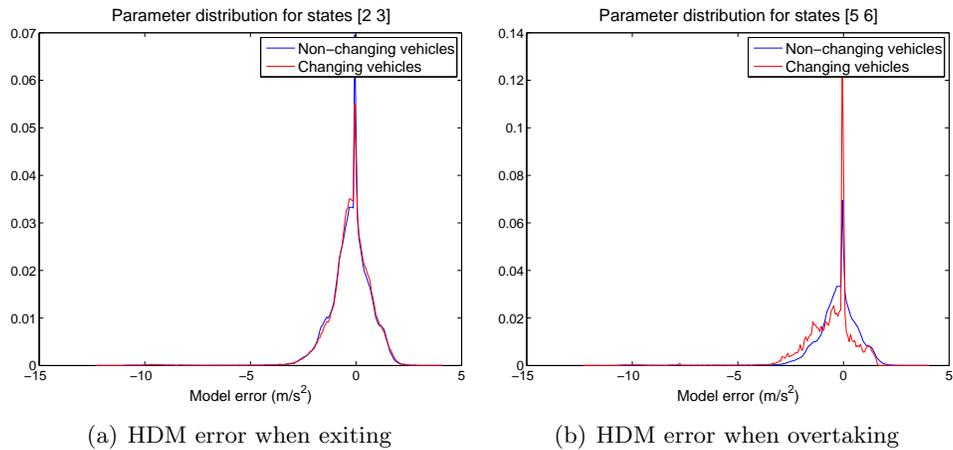


Figure 31. The distributions of the HDM model error show no indication of the model being less applicable to lane changing maneuvers than to following maneuvers

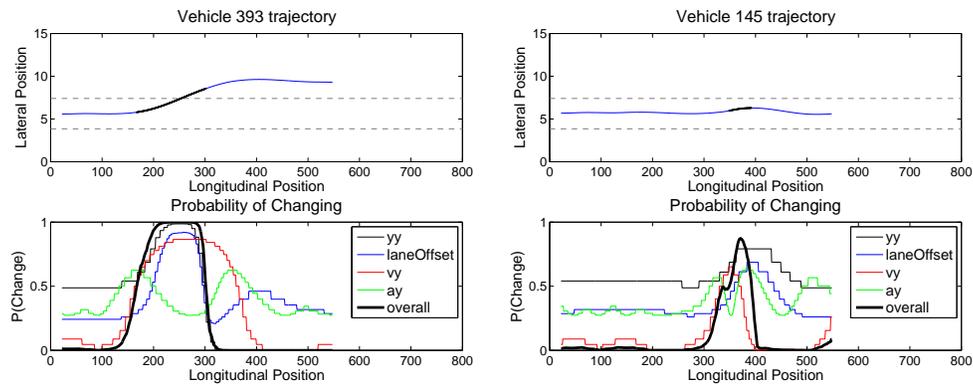
non-changing vehicles. It does seem that most predictions are reasonably accurate, as the majority of the errors is around zero. Still there are some predictions that are off by up to $4m/s^2$. This occurs when a driver is suddenly very close to a predecessor, or is just driving very slowly. The model can not explain this because it does not take into account the history of the trajectory.

Using the set of features there are over 4 billion possible ways to combine the features. However from the combinations that were analyzed for the classification, it was found that the combination of only the lateral position, velocity and acceleration actually performs best in practically all of the cases. Some additional features just did not change the result, and other added features caused a decrease in accuracy. This will be more elaborately discussed in chapter 9.

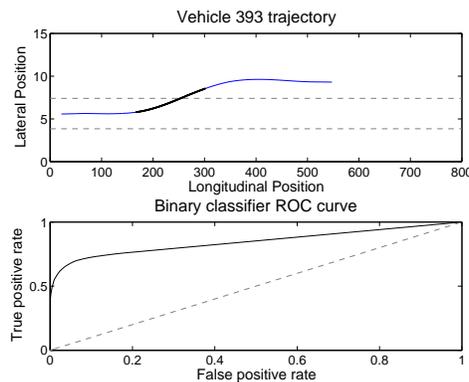
8.2 Single maneuver classifier

Using the distributions of the different features, a Naive Bayes Classifier was implemented as described in chapter 7.5. In figures 32(a) and 32(b) the probabilities can be seen of the different features. Each line in the bottom graph represents the probability that the vehicle is going to change from lane 1 to lane 2 according to its corresponding feature. The thick black line is the overall estimated probability that the vehicle is going to overtake. In the top graph you can see the trajectory of the vehicle, and the thick black line represents the part of the track where the classifier would predict that the vehicle is going to overtake.

In figure 32(c) the corresponding ROC curve is plotted which represent true positive and false positive ratios for different thresholds. An optimal value seems to be around the threshold of 0.25 which means that the probability of changing lanes should be higher than 0.25 before the classifier predicts that the vehicle will change, in order to obtain a false positive rate of about 0.05 and a true positive rate of 0.7.



(a) An example of a successful prediction (b) An example of an erroneous prediction



(c) ROC curve

Figure 32. The Naive Bayes classifier predicts the maneuver based on individual features. In (a) and (b) figure you can see the underlying probabilities of the classifier. In (3) you can see the ROC curve of the predictions

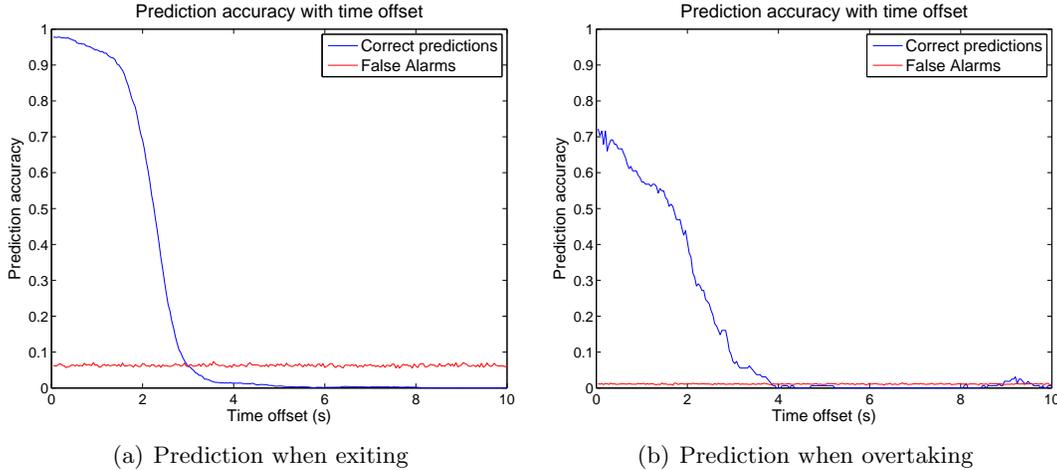


Figure 33. The prediction accuracy with increasing time-offsets to the moment the vehicle first crosses the lane divider

8.3 Prediction time-offset

In the next experiments the prediction accuracies were obtained as a vehicle approaches the lane divider. Obviously as it comes closer, the prediction shall be more accurate. In the first experiment the exiting maneuvers will again be classified. Using the Naive Bayes classifier it was possible to predict 50% of the lane changes 2.24 seconds before the vehicle first crosses the line. At 1.44 seconds before the vehicle crosses the line the accuracy was 90%. Meanwhile the false alarm rate was about 0.08. For this only the lateral position, velocity and acceleration were used, as the other features only led to a decrease in performance. In figure 33(a) you can see the results of the experiment.

In figure 33(b) you can see the result of the same experiment, but when looking at overtaking maneuvers. Here the performance is lower due to the fact that there are far less examples of vehicles overtaking. However, 1.68 seconds before the moment the vehicle first touches the lane divider, the classifier was able to predict with a 50% accuracy that the vehicle is going to overtake. In this experiment there was an average false alarm rate of only 0.01.

8.4 Maneuver prediction classifier

In this chapter the results of the cross-validation experiment will be presented. As mentioned in chapter 7 a set of random “snapshot” examples is selected from the database which are labeled by their segment number. However to maintain the overview in this thesis the separate classes are converted to the maneuvers. Therefore the classification is done according to the labels in table 12. In appendix C you can see the confusion matrices that were obtained during the experiments.

Naive Bayes Using only the three simple features (lateral position, velocity and acceleration) the Naive Bayes classifier obtained an accuracy of 89.49%. When incorporating the longitudinal velocity and acceleration as well, the accuracy decreased to 89.06%. Using

| Class | Name | Description |
|-------|----------|---------------------------------------|
| 1 | follow | Following the leader (no maneuver) |
| 2 | overtake | An overtaking maneuver |
| 3 | return | A maneuver returning after overtaking |
| 4 | merge | A merging maneuver |
| 5 | exit | An exit maneuver |

Table 12: The numbers which represent the maneuvers that the classifiers outputs.

the time gaps of the neighbors resulted in a decrease to 87.36%. When adding the HDM model error the accuracy became 89.00%. Finally when all of these features were combined the accuracy was 82.29%.

These differences might not be significant but the same effect is observed in other classifiers. These effects are not very surprising when considering the feature value distributions. Only the simple features are very different from one class to another. Any learned differences concerning the more complex features in the training phase are coincidental and therefore result in a decrease in performance. Therefore it seems that it is better to leave the complex features out of the process. In table 13 in appendix C you can see the confusion matrix of the Naive Bayes classifier.

K-Nearest Neighbor When using the K-Nearest Neighbor algorithm the classifier performed best using $k = 5$ and only using the three simple features of lateral position, velocity and acceleration. The Euclidean distance was used as a distance measure between two examples. With these settings the accuracy was 94.22%.

This time when adding the more complex features however, the performance decreased more drastically. For instance when adding just the longitudinal velocity and acceleration the accuracy dropped to 88.00%. Adding the time gaps to the neighbors made it drop to 79.97% and using all features again resulted in an accuracy of 78.62%. In table 14 in appendix C you can see the confusion matrix of the 5-Nearest Neighbor classifier

Support Vector Machines Next is the Support Vector Machine classifier, again with the most simple features it performed best with an accuracy of 83.46. When using the complex features as well the accuracy drops to 69.76%. In both cases a Cost Support Vector Classification (C-SVC) algorithm was used (Chang & Lin, 2001) with polynomial kernels. This method is available in Rapid Miner application. It should also be noted that this method took much more time than any other method. The resulting confusion matrix with the simple features can be seen in table 15 in appendix C.

Hybrid Classifier In the cross-validation experiment the decision tree and Naive Bayes hybrid classifier was able to classify the maneuvers with an accuracy of 89.01%. It has a predetermined set of features, including the target lane following and leading distances, the distance to the leader in the same lane, the lane offset and the lateral velocity and acceleration. The corresponding confusion matrix can be seen in table 16 in appendix C.

Hidden Markov classification Finally the Hidden Markov Model approach also performed best using only simple features just: it obtained an accuracy of 92.98%. With the more complex features it performed with an accuracy of 87.46%. When the thresholding method for the transition matrix was applied, the accuracy of the model with simple features increased to 93.07%. The same method with the added complex features yielded an accuracy of 87.53%. The resulting confusion matrix of the model with the simple features can be seen in table 17 in appendix C.

In figure 34 you can see the results of the Hidden Markov Model algorithm as well, but presented in another manner. It shows the fraction of the tracks which are incorrectly classified. They are sorted so that the tracks that are fully correctly classified are on the left side, and those that are mostly incorrectly classified are on the right side. The area below the curve represents the incorrectly classified instances. The different curves represent the results of the HMM with slightly different settings. In this figure you can see how the performance drops when more features are added.

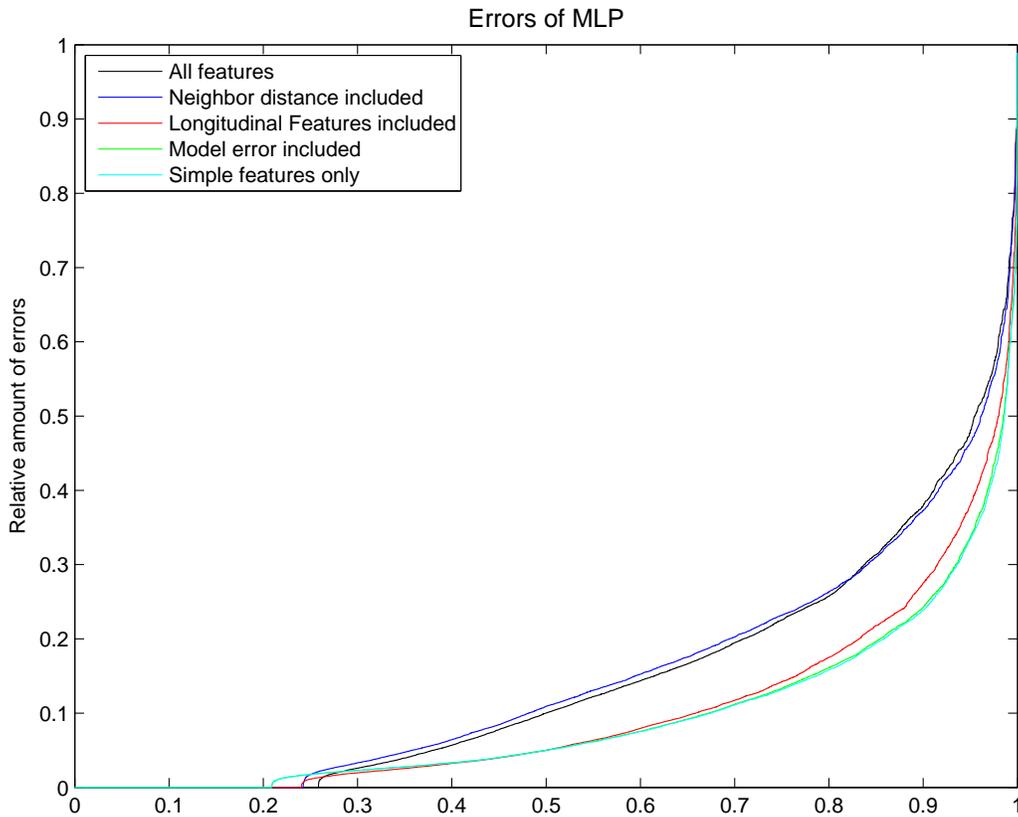


Figure 34. The performance graph of the HMM classifier, different curves are different features sets which are specified in the legend.

8.5 Validation on ENDOR database

As a final validation the classifiers are tested on a file from the ENDOR database as well. Instead of cross-validation the model was trained on the complete dataset of the A67, and used one video from the ENDOR database as test set. In this chapter the results of the K-Nearest Neighbor classifier, the hybrid classifier and the Hidden Markov Model classifier will be compared since they performed best in the cross-validation experiment. In appendix D the exact confusion matrices of the corresponding experiments can be found.

In these experiments using the simple features as earlier, the performances are a lot poorer. This is because in this case the road layout doesn't match exactly, and therefore the lateral positions do not line up very well. So for these experiments the lateral position relative to the road was switched to the lane and the lane offset information.

KNN classifier Using the simple features with the lane offset and the lane information, the K-Nearest Neighbor classifier could classify the vehicle maneuvers with an accuracy of 91.27%. Using all features the accuracy became 87.24%. Including only the longitudinal acceleration yielded the best results with an accuracy to 93.27%. These results were obtained using $k = 50$. The results of this classification can be seen in more detail in table 18 in appendix D.

Hybrid classifier The hybrid classifier was able to classify the vehicle maneuvers in the ENDOR database with an accuracy of 94.81%. The corresponding confusion matrix can be seen in table 19 in appendix D. This classifier is built specifically for one set of features, and it incorporates both the distances to the neighbors in target lanes as well as the leader in the initial lane. Therefore no data of variants of this classifier are available.

Hidden Markov Model For the Hidden Markov Model the initial classifier with the lane and lane offset information the accuracy was 82.98%. Using all of the features did somewhat increase the performance, as it gained an accuracy of 83.49%, and when only adding the longitudinal acceleration and no other features, the performance was best with an accuracy of 85.36%. In table 20 in appendix D you can see the confusion matrix of the HMM validation on the ENDOR database. In figure 35 the performance can be seen of various Hidden Markov Models in the same manner as figure 34.

Errors in the ENDOR database It can be seen in the confusion matrices in appendix D that many errors are made in instances that are classified as some maneuver, but the "true" class is following (class 1). Especially true for the Hidden Markov Model, this is possibly due to the fact that the tracks in the ENDOR database are very short, and therefore the actual crossing of the lane divider is not seen. The classifier that obtains the ground truth using the full trajectory therefore doesn't see a certain instance as part of a maneuver, even though the classifier does see it.

In a small experiment to test this hypothesis, the video data of the tracks were analyzed, where a lot of these errors occurred. 20 tracks were tested to see if the vehicles were actually changing lanes, and in 7 of them it was definitely true that the vehicle was going to performing a maneuver, but the VBM tracking system just failed to see the actual crossing of the lane. In 3 cases the vehicle just finished the maneuver before it entered the

field of view of the camera, so the VBM system couldn't have seen the lane crossing. In the other 10 cases the vehicles just weaved a bit, or drove very close to the lane divider.

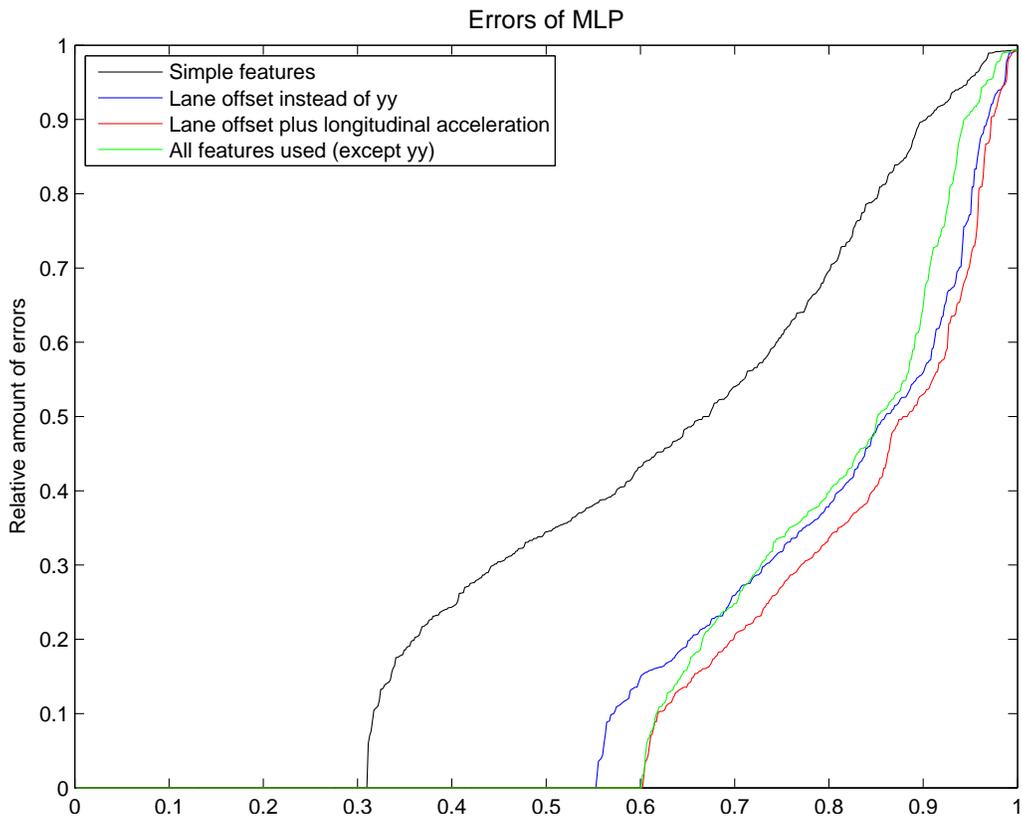


Figure 35. The performance of the HMM classifier on the ENDOR database. Here some additional feature are improving the classification results

9 Discussion

In this chapter the results of the experiments shall be discussed. First the measurement comparisons and the vehicle following models that resulted from the experiment at the A270 will be discussed and how these results could be interpreted. Next the feature distributions and the predictive classifiers of the classification results shall be analyzed.

9.1 A270 experiments

The A270 experiment showed that the proposed VBM trajectory filtering method is highly accurate. For longitudinal position the VBM data does not differ significantly from the GPS measurements (Ublox). The velocity is also very accurate as it deviates only slightly from the velocities measured in the CAN data. Whenever a vehicle enters the field of view of a new camera, the velocity deviates a bit as the position is then measured from another perspective. This leads to only small differences in position, but the effect is noticeable on the velocity. This problem is beyond the scope of this thesis however, since it seems to some sort of a calibration offset in the VBM process. Also it seems that near peaks in the velocity profile the spline filter tends to cut off a bit, but this effect is still insignificant. Finally the acceleration data makes these small differences more obvious, but it still gives a fair estimate of the acceleration. For the purposes of this thesis it is good enough, but for later research this might be something to improve.

In chapter 6.4 the Genetic Algorithm allowed to find accurate estimations of both the IDM and the HDM parameters, corresponding with the findings by Pellecchia et al. (2005). However it is important to fit the accelerations of the model, and not the locations. Fitting to the locations does lead to a model that very accurately predicts the location, but the accelerations needed to get that model are far beyond realistic. Therefore it is absolutely necessary to fit the model acceleration to the observed acceleration when trying to obtain realistic model behavior.

It seemed that the HDM model allows for a slightly better fit when trying to recreate observed data. Of course this can be attributed to the fact that it has two extra variables, and therefore two extra degrees of freedom. Then again, this would also mean that a longer time is needed for the Genetic Algorithm to find the best set of parameters, so extending the time to find the parameters will probably have more influence on the HDM results than on the IDM results.

Besides better fitting the observed data, the simulations using the HDM also better predicts the dissolving of the shockwaves. In the simulations it also seems that the inter-driver variability has a very large influence on the overall behavior of the traffic flow, which corresponds to findings in other studies (Tampère, 2004). In the final simulations using HDM, the resulting driver models are very lifelike. They seem to spontaneously produce shockwave patterns and in overall behave in a way not very dissimilar from the observed vehicles.

The parameters of the models fitting the observations were spread throughout the parameter space in a seemingly random cloud. Clustering the parameters in order to find for instance a group of aggressive and a group of passive drivers is not possible. There seems to even be no noticeable difference in the model parameters for instrumented vehicles

and uninstrumented vehicles. The inter-driver variability seems to be higher than the inter-group variability.

Better driver models are still needed. Even though HDM is able to generate a lot of realistic data, there are still some unaddressed aspects of driver modeling. For instance HDM does not allow to set different driver preferences in different situations. When for instance the overall density of traffic is higher, the driver might want to reduce its preferred velocity and/or decrease its desired time headway. Although there are many features that might be included, it will be extremely difficult to fully capture the human driving behavior.

As it was not the aim of this thesis the results of the A270 experiments weren't analyzed thoroughly, but they showed that the CACC system did increase the average velocity of the equipped vehicles in comparison to the non equipped vehicles. Also did it reduce the variability of the time headway of the equipped vehicles (Broek et al., 2010). This last result also emerged in the estimated driver models, as the parameter T was the only one that seemed to differ from the instrumented to the uninstrumented group.

9.2 Classification experiments

From the feature distributions in the classification experiments in chapter 8.1 it appeared that many features aren't very useful when trying to classify the maneuver of vehicles. Of course the lateral position relative to the road, and relative to the lane are extremely significant, since they define the lane change. The first and second derivatives of the position: the velocity and the acceleration are also very useful for maneuver classification, although it seems that every level of derivation decreases the usability. This is due to the fact that the velocity and acceleration may hold the same values during a maneuver as during a non-maneuver i.e. the vehicles may swerve a bit without actually changing lanes. Also these features are just not as reliably measured as the position, which might be of influence.

The longitudinal velocity and acceleration seem to be less influenced by the maneuver the vehicle is in, but still they allow for some improvement when included in the classifier. For some maneuvers it is true that the acceleration gives some evidence on whether the vehicle is likely to perform a maneuver or not. In the validation experiments the classification performance improved when the longitudinal acceleration was included, but not in the cross-validation experiments. This might be because the training set size is too small in a 10-fold cross validation setup, but when more data is added, useful rules can be learned.

The distance to neighboring vehicles are only marginally of influence on the lane change predictions. Although some statistical analysis is possible, and the Hybrid Classifier does use the information on neighboring vehicles, the differences are small. In most cases when information about neighboring vehicles is included the classification performance goes down. It seems that this is because the training set is once again too small, particularly because neighbors are not always present. Additionally the lane changes in the dataset were primarily Mandatory Lane Changes which are less influenced by the neighboring vehicles (Ahmed et al., 1996).

Finally the last feature that was analyzed was the HDM error. It was hypothesized that because HDM gives an indication of how drivers follow other vehicles, the prediction error would be larger whenever a maneuver were to be executed. However no evidence was found that supports this hypothesis. Possibly the Human Driver Model does not give an

accurate enough estimation to detect this effect from the noise error. It is possible that such effects can be found in larger databases with better following models, but for that more research would be needed. For instance it would be possible that there are differences in the parameters of the best fitting models. Because in this thesis a predetermined set of parameters was used, this effect was not analyzed.

In the classification experiments, the Naive Bayes classifier was able to predict 50% of the exit maneuvers 2.24 seconds before the vehicle touches the lane divider. Additionally 90% of the exit maneuvers were predicted 1.44 seconds in advance. For the overtaking maneuver this performance was definitely lower as 50% of the lane changes were predicted 1.68 seconds before lane change. Apparently the amount of overtaking maneuvers is still too low to extract a reliable method for predicting lane changes. This could probably be improved by training the classifier using a more extensive database.

The best classifier in the cross-validation setup was the K-Nearest Neighbor algorithm using $k = 5$. With these settings the classification accuracy was 94.22%. The Hidden Markov Model approach however gets very near with an accuracy of 93.07%. The Naive Bayes and the Hybrid classifiers perform somewhat poorer with 89.49% and 89.01% respectively, and the Support Vector Machine performed worst of the analyzed methods with an accuracy of only 83.46%. The reason for this is unclear, since usually Support Vector Machine tend to outperform the Naive Bayes classifier. Perhaps the training data was not sufficient to generalize an appropriate model. It should be noted that the Support Vector Machine was not extensively researched, and with additional effort it might be possible to create better performing SVM classifiers.

What also can be learned from the experiments is that some classifiers are more robust against noisy features than others. For instance the K-Nearest Neighbor and the Support Vector Machine methods seemed to be very sensitive, as adding the complex features drastically decreased their performance. The Naive Bayes and the Hidden Markov approaches however were more robust, as the performance only slightly decreased.

It seems very interesting that the performance of the hybrid classifier is very high when validating on the ENDOR database. A lot better even than the Hidden Markov Classifier, even though in the cross validation experiment this was the other way around. This is perhaps due to the fact that the training set becomes too small when using cross validation, and our hybrid classifier therefore fails. In the cross validation the probability distribution is made up from only a few examples. Apparently in cross-validation there are some unseen values for features in the training set, which causes maneuvers to be discarded too quickly in the test phase. Now of course this should also apply for the Hidden Markov Model, but apparently that is more robust.

The final performance of the lane change maneuver prediction is quite good. The accuracy of the hybrid classifier in the validation using the ENDOR database is 94.77%. It has to be noted however that the precision values for the individual maneuvers are much lower for the lane changing maneuvers. The following maneuver is extremely abundant in the data, and therefore plays a large role in the validation. In an application it might be necessary to shift the thresholds a bit so that the precision increases and the recall decreases. This might lead to a worse overall accuracy because many lane change maneuvers would no longer be recognized, but the current system would give way too many false alarms.

10 Conclusion

In chapter 4 we have determined that fitting a cubic smoothing spline to the VBM data allows for a very smooth and realistic estimate of the vehicle's trajectories. Later in chapter 6.3 we validated the processed measurements and we determined that the estimated location is highly accurate when comparing to reference measurements. Although the velocity and acceleration data are less reliable, they still provide smooth and accurate data. Therefore this preprocessing method makes it possible to use VBM as a tool to extract continuous and reliable information about location, velocity and acceleration on different vehicles. It also creates the possibility obtain inter-vehicle times and distances and possibly even more information.

In chapter 6 we have seen that HDM provides a slightly better fit than IDM to observed trajectories. Also HDM simulations show a more realistic traffic congestion behavior. This could be used for future research on new vehicle techniques, infrastructures or other traffic flow related issues. The model should definitely include inter-driver variability and multiple vehicle anticipation as the resulting traffic flow is much more realistic than the flow of the simulation with monotonous IDM vehicles. Although it is probably possible to create even better and more realistic models, HDM allows for a lot of possibilities.

In chapter 8.1 we have seen that the different features are not all very suitable to predict future lane changes. Mostly the lateral position and its derivatives seem to be useful, which is not very surprising since they define the actual lane change. In some cases the longitudinal acceleration also seemed to be useful for lane change prediction, and the distances to neighboring vehicles were only marginally relevant for the hybrid classifier. However it seems plausible that with a larger database and even more accurate data, their influence will be more significant. Unfortunately we were not able to use the acceleration model error as a predictive feature.

The predictive lane change classifier seems to be reliable enough for driver assistance application purposes. With an accuracy of over 90% we could help create a system that provides information about other vehicles cutting in, in the short term future. A Naive Bayes classifier was able to predict 90% of the exit maneuvers 1.44 seconds in advance. Additionally it can predict 50% of the overtaking maneuvers 1.68 seconds in advance. The methods we proposed in this thesis are quite flexible, and therefore allow for the possibility to train models for different highway situations. Ideally we would create a model that is accurate everywhere, but we do not have the means to test this yet.

Obviously for reliable methods there is still more research needed, but in this thesis we showed that it is possible to create models of how drivers will behave in the short term future. This could be used in traffic simulations to create realistic driver behavior, but it could also lead to applications in intelligent transport systems where vehicles or roadside units are aware of future actions of nearby vehicles.

References

- Absil, N. (2008). *Driver Behaviour Model for the Multi-Agent Real-time Simulation*. Unpublished master's thesis, Delft University of Technology.
- Ahmed, K. (1999). *Modeling drivers' acceleration and lane changing behavior*. Unpublished doctoral dissertation, Massachusetts Institute of Technology.
- Ahmed, K., Ben-Akiva, M., Koutsopoulos, H., & Mishalani, R. (1996). Models of freeway lane changing and gap acceptance behavior. *Transportation and Traffic Theory. New York: Elsevier Science Publishing*, 501–515.
- Bando, M., Hasebe, K., Nakanishi, K., Nakayama, A., Shibata, A., & Sugiyama, Y. (1995). Phenomenological study of dynamical model of traffic flow. *J. Phys. I France*, 5, 1389–1399.
- Bham, G. (2009). Estimating Driver Mandatory Lane Change Behavior on a Multi-lane Freeway. *Proceedings of the 88th Transportation Research Board*.
- Broek, T. van den, Netten, B., Hoedemaeker, M., & Ploeg, J. (2010). The experimental setup of a large field operational test for cooperative driving vehicles at the A270. In *13th International IEEE Annual Conference on Intelligent Transport Systems*.
- Butterworth, S. (1930). On the Theory of Filter Amplifiers. *Wireless Engineer*, 536–541.
- Chang, C., & Lin, C. (2001). *LIBSVM: a library for support vector machines*. Citeseer.
- Cleveland, W. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 829–836.
- Dagli, I., Breuel, G., Schittenhelm, H., & Schanz, A. (2004). Cutting-in vehicle recognition for ACC systems-towards feasible situation analysis methodologies. *Proceedings of the IEEE Intelligent Vehicles Symposium*, 925–930.
- Dagli, I., Brost, M., & Breuel, G. (2003). Action recognition and prediction for driver assistance systems using dynamic belief networks. *Lecture Notes in Computer Science*, 179–194.
- Driel, C. van, & Arem, B. van. (2007). Traffic flow impacts of a congestion assistant. In *87th Annual Meeting of the Transportation Research Board, Washington DC*.
- Dubelaar, B. (2009). *Short term driver behavior prediction an online approach*. Unpublished master's thesis, Delft University of Technology.
- Europa.eu. (2008). *Greening Transport Package – Frequently asked questions*. Available from <http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/08/492>
- Forney Jr, G. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, 61(3), 268–278.
- Gardiner, C. (1985). *Handbook of stochastic methods*. Springer Berlin.
- Gettman, D., & Head, L. (2003). Surrogate safety measures from traffic simulation models. *Transportation Research Record: Journal of the Transportation Research Board*, 1840(-1), 104–115.
- Hastie, T., & Tibshirani, R. (1990). Cubic Smoothing Splines. In *Generalized Additive Models* (pp. 27–29). Suffolk, Great Britain: Chapman & Hall.
- Hlavacek, S. (2010). *Integrated Driver Behavior Model*. Unpublished master's thesis, HAN University of Applied Sciences.
- Huis, J. van, Baan, J., & Loon, A. van. (2008). *Traffic Monitoring using Video-based Vehicle Tracking* (Tech. Rep.). TNO.

- Ioannou, P., Wang, Y., & Chang, H. (2007). *Integrated Roadway/Adaptive Cruise Control System: Safety, Performance, Environmental and Near Term Deployment Considerations*. California PATH Research Report, UCB-ITS-PRR-2007-08.
- Kesting, A., & Treiber, M. (2008). Calibrating Car-Following Models by Using Trajectory Data: Methodological Study. *Transportation Research Record: Journal of the Transportation Research Board*, 2088(-1), 148–156.
- Kesting, A., Treiber, M., & Helbing, D. (2007). General lane-changing model MOBIL for car-following models. *Transportation Research Record: Journal of the Transportation Research Board*, 1999(-1), 86–94.
- Kesting, A., Treiber, M., Schönhof, M., & Helbing, D. (2008). Adaptive cruise control design for active congestion avoidance. *Transportation Research Part C: Emerging Technologies*, 16(6), 668–683.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. The MIT press.
- Otsu, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11, 285–296.
- Pellecchia, A., Igel, C., Edelbrunner, J., & Schoner, G. (2005). Making driver modeling attractive. *IEEE Intelligent Systems*, 20(2), 8–12.
- Renkema, S. (2009). *Object fingerprinting*. Unpublished master’s thesis, Rijksuniversiteit Groningen.
- Salvucci, D., Mandalia, H., Kuge, N., & Yamamura, T. (2007). Lane-change detection using a computational driver model. *Human factors*, 49(3), 532.
- Schakel, W., Arem, B. van, & Netten, B. (2010). Effects of Cooperative Adaptive Cruise Control on Traffic Flow Stability. In *13th International IEEE Annual Conference on Intelligent Transportation Systems*.
- Stiller, C., Färber, G., & Kammel, S. (2007). Cooperative cognitive automobiles. *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 215–220.
- Tampère, C. (2004). Human-kinetic multiclass traffic flow theory and modelling. With application to Advanced Driver Assistance Systems in congestion.
- Tampère, C., Hoogendoorn, S., & Arem, B. (2005). A behavioural approach to instability, stop and go waves, wide jams and capacity drop. In *Proceedings of the 16th International Symposium on Transportation and Traffic Theory* (pp. 205–228).
- Thiemann, C., Treiber, M., Kesting, A., Baker, G., Hague, J., Opsahl, T., et al. (2008). Estimating Acceleration and Lane-Changing Dynamics Based on NGSIM Trajectory Data. In *Proceedings of the 87th Annual Meeting of the Transportation Research Board*.
- Torkkola, K., Venkatesan, S., & Liu, H. (2005). Sensor sequence modeling for driving. *Proceedings of the 18th International FLAIRS Conference*, 15–17.
- Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2), 1805–1824.
- Treiber, M., Kesting, A., & Helbing, D. (2006). Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A: Statistical Mechanics and its Applications*, 360(1), 71–88.
- Yang, Q., & Koutsopoulos, H. (1996). A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies*, 4(3), 113–129.

Zhang, J., & Ioannou, P. (2004). Integrated roadway/adaptive cruise control system: safety, performance, environmental and near term deployment considerations. *UC Berkeley: California Partners for Advanced Transit and Highways (PATH)*.

Appendices

A Smoothing Results

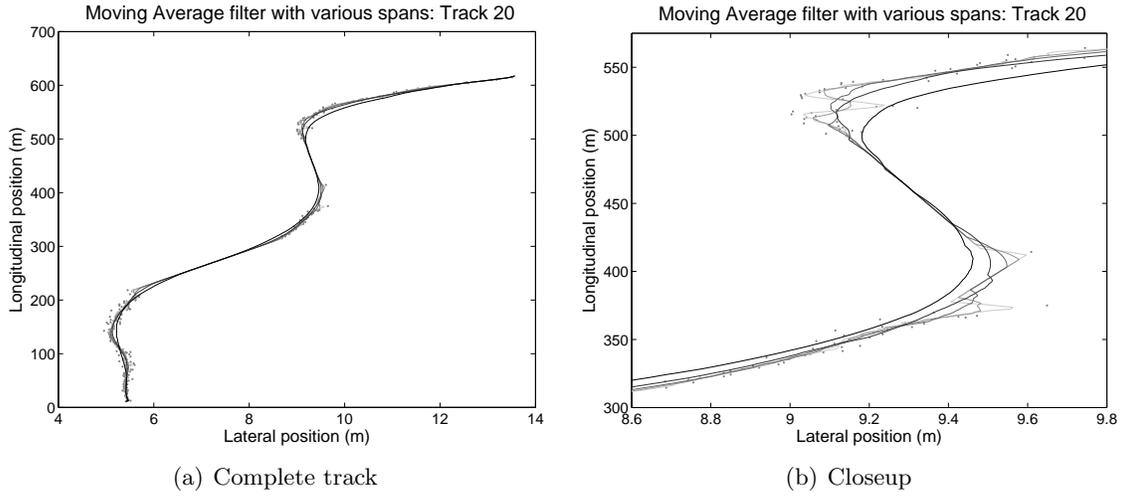


Figure 36. The smoothing of the Moving Average filter of track 20 using different spans. The larger the span, the darker the color

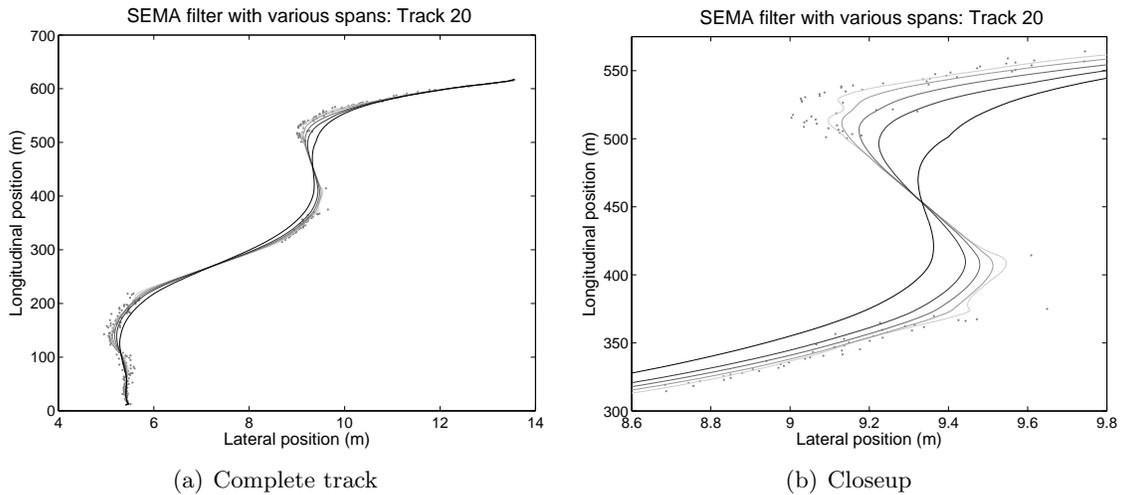


Figure 37. The smoothing of the SEMA filter of track 20 using different spans. The larger the window, the darker the color

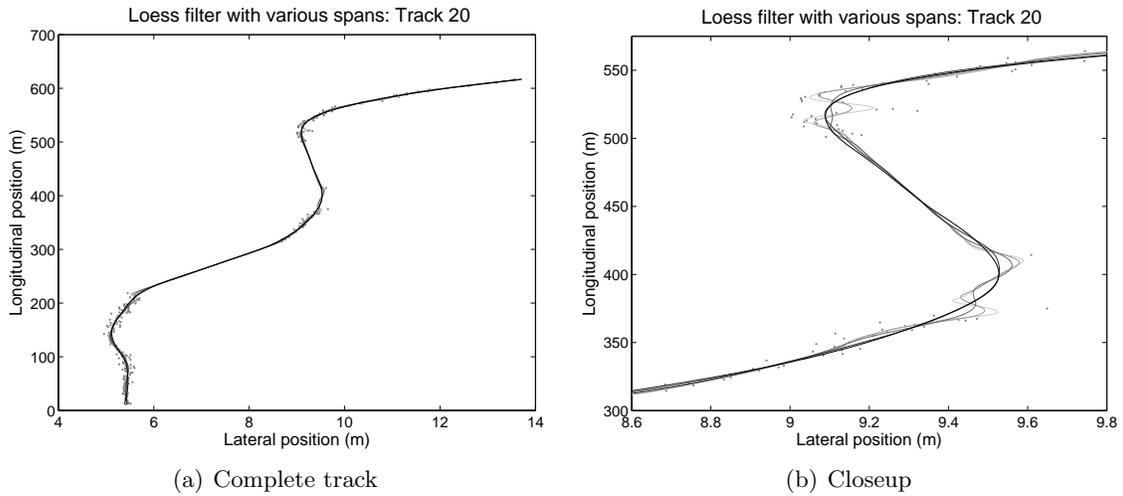


Figure 38. The smoothing of the Loess filter of track 20 using different spans. The larger the span, the darker the color

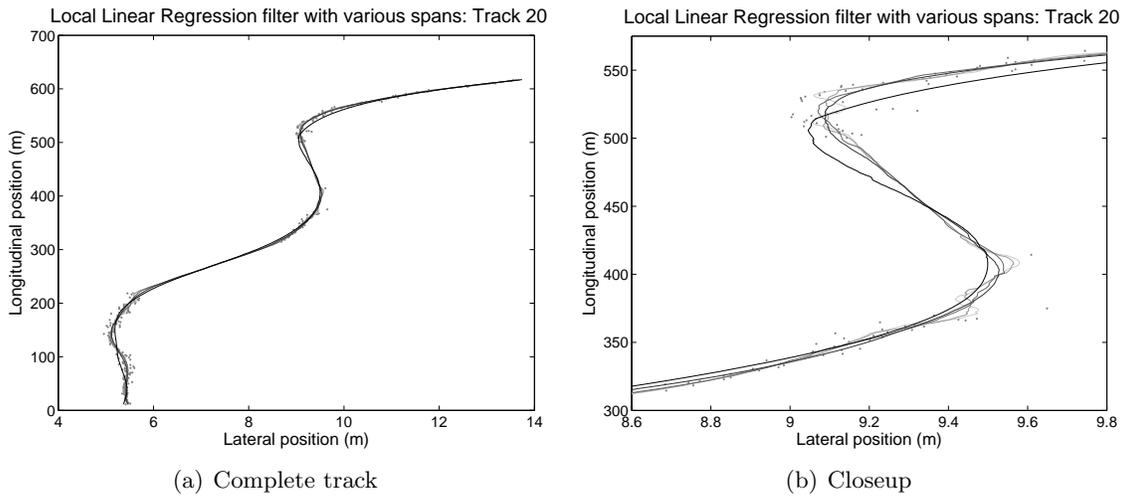


Figure 39. The smoothing of the Local Polynomial Regression filter of track 20 using different spans. The larger the span, the darker the color

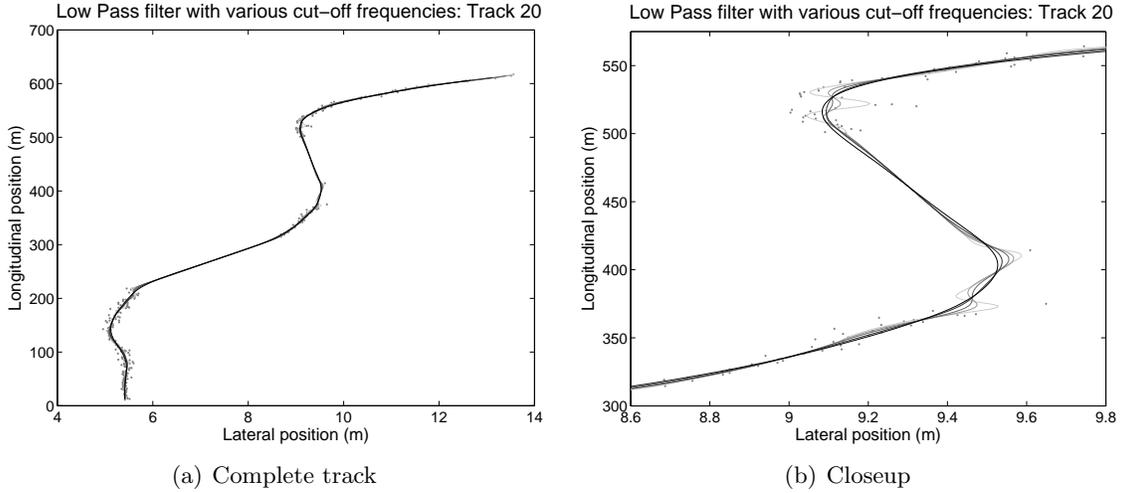


Figure 40. The smoothing of the Butterworth low-pass filter of track 20 using different cut-off frequencies. The lower the cut-off frequency, the darker the color

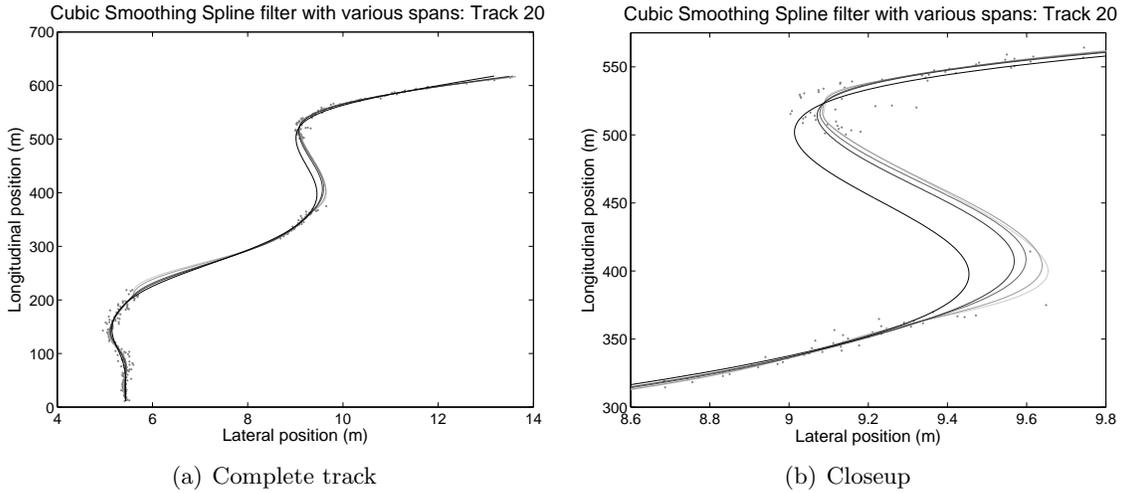


Figure 41. The smoothing of the Cubic Smoothing Spline filter of track 20 using different spans. The smaller the curve penalty, the darker the color

B HDM Parameters

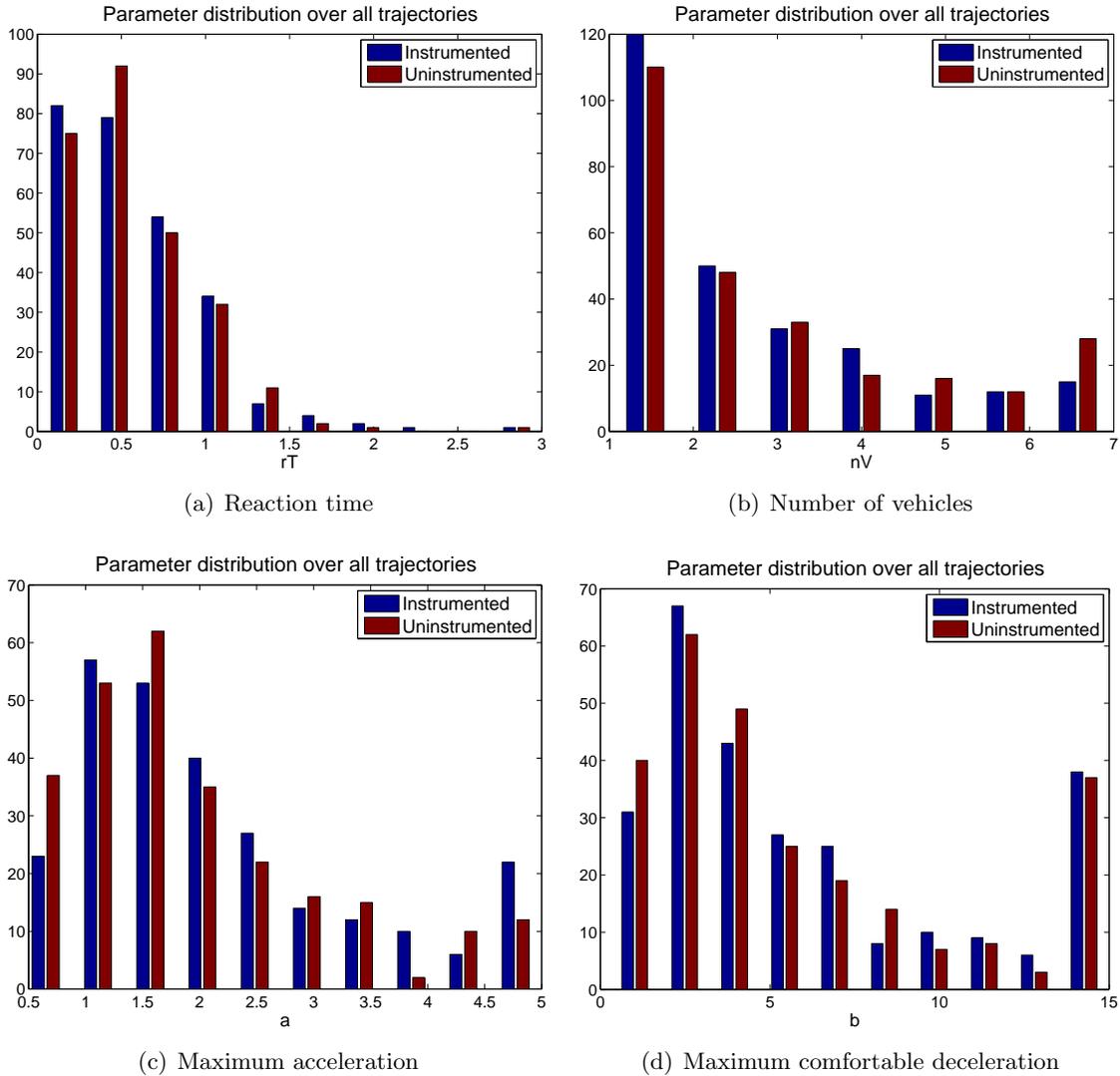


Figure 42. The distribution of HDM parameters of the models that best fit the A270 data

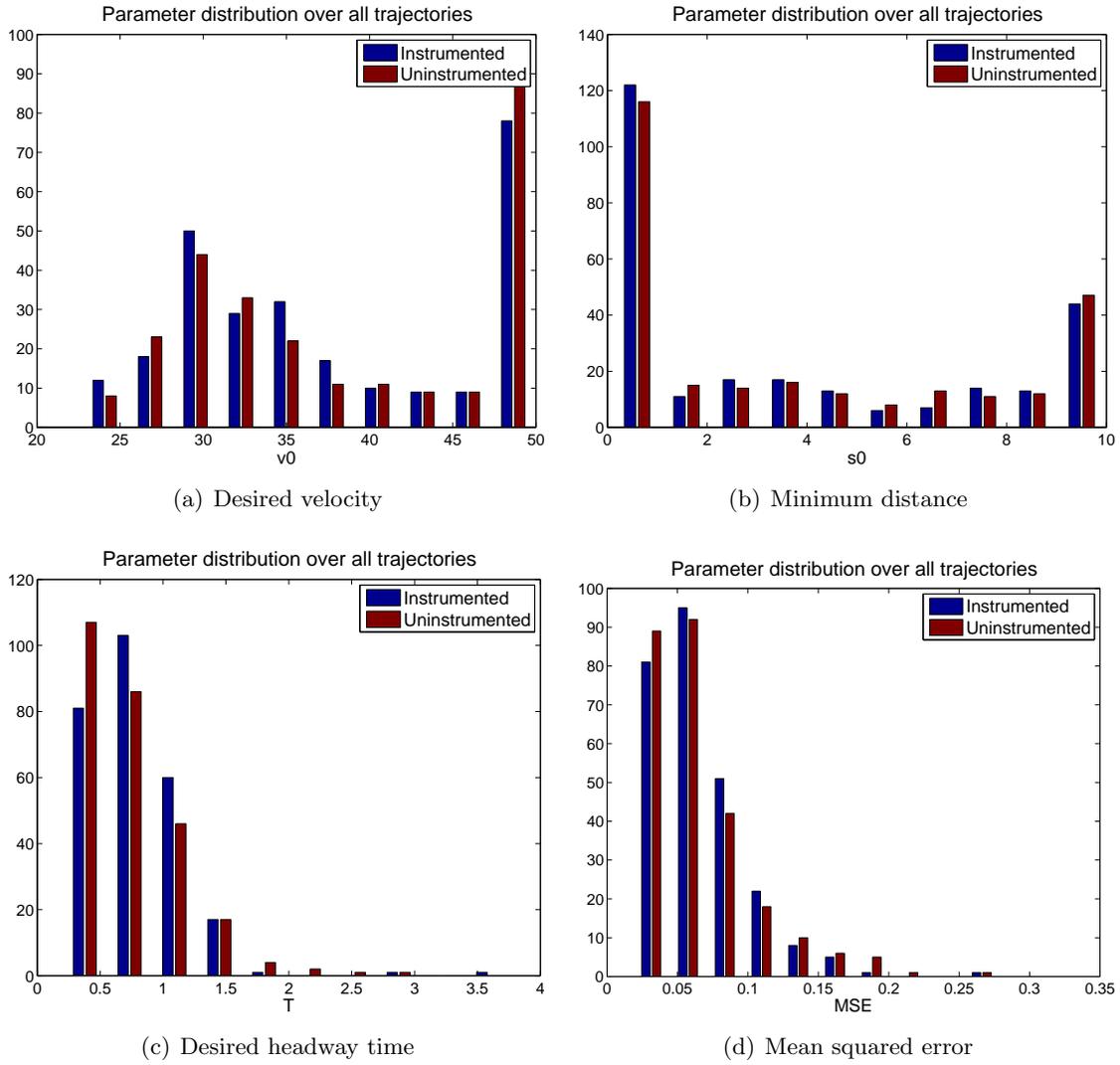


Figure 43. The distribution of HDM parameters of the models that best fit the A270 data

C Confusion Matrices from Cross Validation Experiments

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|--------|--------|--------|-----------|
| pred. 1 | 6610 | 34 | 140 | 66 | 458 | 90.45% |
| pred. 2 | 11 | 74 | 0 | 4 | 0 | 83.15% |
| pred. 3 | 48 | 2 | 437 | 1 | 11 | 87.58% |
| pred. 4 | 5 | 1 | 0 | 82 | 0 | 93.18% |
| pred. 5 | 216 | 0 | 46 | 8 | 1746 | 86.61% |
| class recall | 95.94% | 66.67% | 70.14% | 50.93% | 78.83% | |

Table 13: The classification results of the Naive Bayes Classifier using the simple feature set. Training and testing with the A67 dataset using cross validation

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|--------|--------|--------|-----------|
| pred. 1 | 6652 | 35 | 91 | 51 | 148 | 95.34% |
| pred. 2 | 7 | 75 | 0 | 1 | 0 | 90.36% |
| pred. 3 | 72 | 0 | 527 | 0 | 8 | 86.82% |
| pred. 4 | 8 | 1 | 0 | 109 | 0 | 92.37% |
| pred. 5 | 151 | 0 | 5 | 0 | 2059 | 92.96% |
| class recall | 96.55% | 67.57% | 84.59% | 67.70% | 92.96% | |

Table 14: The classification results of a 5 Nearest Neighbors algorithm using the simple feature set. Training and testing with the A67 dataset using cross validation

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|--------|--------|--------|-----------|
| pred. 1 | 6855 | 83 | 568 | 165 | 715 | 81.74% |
| pred. 2 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 3 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 4 | 0 | 0 | 0 | 0 | 0 | 0.00% |
| pred. 5 | 121 | 0 | 2 | 0 | 1491 | 92.38% |
| class recall | 98.27% | 0.00% | 0.00% | 0.00% | 67.59% | |

Table 15: The classification results of a Support Vector Machine classifier using the simple feature set. Training and testing with the A67 dataset using cross validation

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|--------|--------|--------|-----------|
| pred. 1 | 6772 | 65 | 151 | 133 | 296 | 91.30% |
| pred. 2 | 5 | 21 | 0 | 0 | 18 | 47.73% |
| pred. 3 | 104 | 0 | 434 | 0 | 253 | 54.87% |
| pred. 4 | 6 | 0 | 0 | 35 | 0 | 85.37% |
| pred. 5 | 45 | 18 | 0 | 0 | 1635 | 96.29% |
| class recall | 97.69% | 20.19% | 74.19% | 20.83% | 74.25% | |

Table 16: Confusion matrix of the results of the Hybrid classifier. Training and testing with the A67 dataset using cross validation

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|--------|--------|--------|-----------|
| pred. 1 | 6456 | 42 | 27 | 42 | 71 | 97.26% |
| pred. 2 | 50 | 57 | 0 | 2 | 0 | 52.29% |
| pred. 3 | 150 | 2 | 546 | 0 | 17 | 76.36% |
| pred. 4 | 55 | 2 | 0 | 122 | 1 | 67.78% |
| pred. 5 | 219 | 0 | 11 | 1 | 2115 | 90.15% |
| class recall | 93.16% | 55.34% | 93.49% | 73.05% | 95.96% | |

Table 17: The classification results of a Hidden Markov Model classifier using the simple feature set. Training and testing with the A67 dataset using cross validation

D Confusion Matrices from ENDOR Validation Experiments

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|--------|--------|--------|-----------|
| pred. 1 | 9019 | 199 | 52 | 48 | 11 | 96.68% |
| pred. 2 | 0 | 8 | 0 | 0 | 0 | 100% |
| pred. 3 | 48 | 18 | 136 | 0 | 0 | 67.33% |
| pred. 4 | 3 | 8 | 0 | 39 | 0 | 78.00% |
| pred. 5 | 270 | 0 | 15 | 1 | 125 | 30.41% |
| class recall | 96.56% | 3.43% | 67.00% | 44.32% | 91.91% | |

Table 18: Confusion matrix of the results of the K-Nearest neighbor classifier trained with the A67 data, and applied on the ENDOR data. In this setup the longitudinal acceleration was included and $k = 50$.

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|---------|--------|--------|-----------|
| pred. 1 | 8892 | 44 | 0 | 10 | 6 | 99.33% |
| pred. 2 | 4 | 179 | 0 | 0 | 0 | 97.81% |
| pred. 3 | 143 | 0 | 210 | 0 | 0 | 59.49% |
| pred. 4 | 84 | 0 | 0 | 82 | 0 | 49.40% |
| pred. 5 | 219 | 9 | 0 | 0 | 112 | 32.94% |
| class recall | 95.18% | 77.16% | 100.00% | 89.13% | 94.92% | |

Table 19: Confusion matrix of the results of the Hybrid classifier trained with the A67 data, and applied on the ENDOR data.

| | true 1 | true 2 | true 3 | true 4 | true 5 | precision |
|--------------|--------|--------|--------|--------|--------|-----------|
| pred. 1 | 7898 | 5 | 2 | 8 | 6 | 99.73% |
| pred. 2 | 166 | 249 | 0 | 9 | 0 | 58.73% |
| pred. 3 | 265 | 0 | 200 | 0 | 0 | 43.01% |
| pred. 4 | 392 | 0 | 0 | 67 | 0 | 14.60% |
| pred. 5 | 590 | 0 | 19 | 0 | 116 | 16.00% |
| class recall | 84.82% | 98.03% | 90.50% | 79.76% | 95.08% | |

Table 20: Confusion matrix of the results of the HMM classifier trained with the A67 data, and applied on the ENDOR data. In this setup the longitudinal acceleration was included.