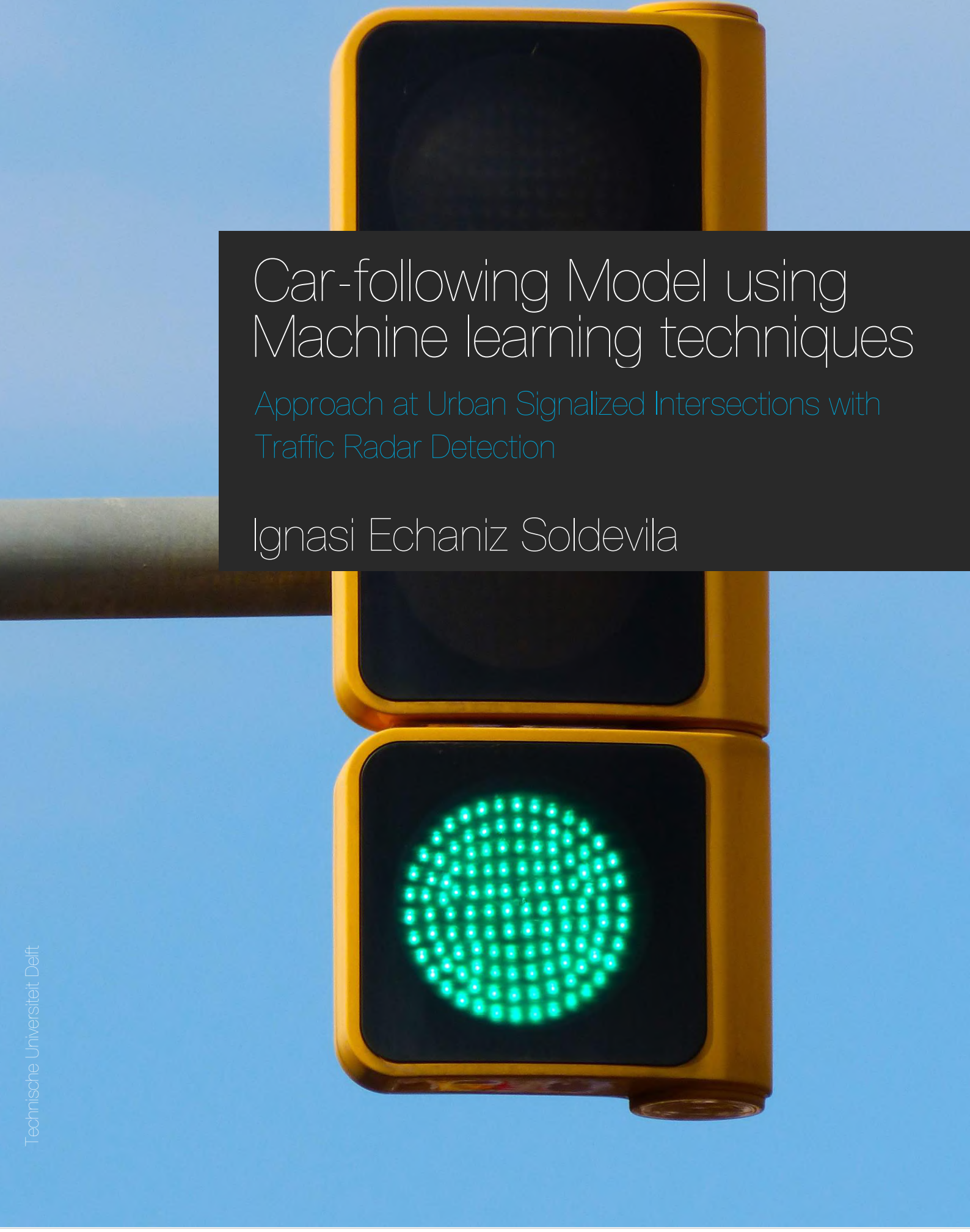


Car-following Model using Machine learning techniques

Approach at Urban Signalized Intersections with Traffic Radar Detection

Ignasi Echaniz Soldevila

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Car-following Model using Machine Learning techniques

Approach at Urban Signalized Intersections with Traffic Radar Detection

by

Ignasi Echaniz Soldevila

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday 12^{th} of October 2017, at 16:00.

Student number:	4516680	
Project duration:	March 2^{nd}, 2017 – October 12^{th}, 2017	
Thesis committee:	Prof. dr. ir. S. P. Hoogendoorn,	TU Delft, Chairman
	Dr. V. Knoop,	TU Delft, Daily Supervisor
	Dr. J. Alonso-Mora,	TU Delft, External Supervisor
	Ir. J. Steenbakkers,	INCONTROL Simulation Solutions

An electronic version of this thesis is available at http://repository.tudelft.nl/.

This thesis has been conducted in cooperation with INCONTROL Simulation Solutions

TU Delft Delft University of Technology

INCONTROL Simulation Solutions

Preface

From the moment I started my bachelor in Civil Engineering in my native country, I knew that traditional civil engineering was not my path. I did not enjoy designing structures of bridges, whereas I was interested on how the layout of the bridge should be designed to optimise traffic flows. Thus, there was no other choice than joining Transport and Planning department at TUDelft to carry out my master studies. Honestly, the professional environment in this department has exceeded my expectations. My master studies in Transport and Planning have brought to me the opportunity to learn how to improve people's life by means of solving daily transport problems and by efficiently anticipating, predicting and managing undesired situations. Finally, with this thesis, I hope to have acquired enough knowledge, so from the day after my graduation I can actually start solving mobility problems in dense urbanised regions.

I would like to thank each university member of the assessment committee of my thesis. First of all, I would really like to thank my daily supervisor, Victor Knoop. In every meeting, new thesis paths and opportunities were discovered. I especially enjoyed realising how together we could both push our minds and think far away from transport traditional modelling techniques. Special thanks to chair of the committee, Prof. Serge Hoogendoorn, whose advises have always been been of great importance and full of content. Finally, I wish to thank Javier Alonso-Mora, my external supervisor, for accepting this challenge of guiding a master thesis from a completely different engineering field from his expertise.

This work would also not have been possible without the support INCONTROL Simulations Solutions. I am deeply grateful to all colleagues who have been continuously asking for the status of my thesis. Special thanks to Jeroen Bijsterbosch, who gave me a lot of technical programming feedback during my project. Most important of all, I would like to thanks Jeroen Steenbakkers, who since the first time he interviewed me, he has always believed in me.

I would like to thank my wonderful family, who have always been with me even if they were physically thousands of kilometres away. We have always been a close and warm family and this will endure forever. I wish to thank my master classmates of TIL and TP and my flatmates, who became my Dutch family in these years far away from my lovely Barcelona. Closing the acknowledgement and most importantly, I would like to thank from the bottom of my heart my loving and supportive partner, Alessandra, whose selfless time and care were sometimes all that kept me going. She has always been next to me in the joyful and critical moments. I will be always grateful to her.

Ignasi Echaniz Soldevila
Utrecht, October 2017

Summary

Since the introduction of fast PCs in the last decades of the 20^{th} century, the way that traffic engineers approach modelling and test transport infrastructures and traffic scenarios has completely changed. Nowadays engineers rely on microscopic traffic simulation software to tackle conventional traffic management problems. Car-following models are those submodels that describes the longitudinal behaviour of drivers. Despite existent car-following models are seen as accurate, they might miss several variables relationships in urban signalised intersections. Historically, existent parametric car following models have usually been calibrated using traditional optimisation techniques and small, yet accurate datasets. Recently, large datasets are becoming available thanks to technology improvements and governmental efforts. Opposite traditional datasets, new available datasets contain large amounts of data, e.g. thousands of trajectories. Nonetheless, these datasets frequently contain errors and inconsistencies and they also present significant noise. One example of this is traffic radar detection technology, which has started to being used for massive traffic data collection and traffic management in The Netherlands. Opposite to on-board units, traffic radars are installed in existent infrastructure such as light pole or traffic lights and simultaneously detect and register multiple vehicle driving in the radar road section range. However, occlusion and interferences might occur, leading to data gaps or data errors. Moreover, noise is frequently found in the position measurements. The characteristics of the data do not recommend to use parametric models to describe driving longitudinal behaviour. Those models do not benefit from inaccurate, noisy and large datasets in calibration phases due to computational time issues and its rigid model formulation. Alternatively, non parametric models derived from new machine learning techniques are rapidly becoming popular to deal with this type of datasets. Non parametric models are often derived using fast computational methods that "learn" information directly from the data and by relying on a generic model formulation, which adaptively improve their performance as the number of samples available for learning increase.

This master thesis aims to gain new empirical insights into longitudinal driving behaviour of following vehicles, particularly at urban signalised intersections, by means of the enumeration of a new hybrid car-following model which combines parametric and non parametric mathematical formulation. Particularly, the goal focus on obtaining a calibrated car following model in stop and go urban traffic conditions which predicts acceleration of a driver based on a set of predictor variables such as drivers' speed or spacing. As part of the methodology, the thesis explores the viability of non parametric models using machine learning techniques and trained by large datasets with significant errors and noise. All the above mentioned contributed to formulating the following research question:

How can the longitudinal urban driver behaviour at signalized intersections be modelled using non parametric models and machine learning techniques?

In order to answer the main research question, new relationship with variables rarely included in car-following models such as distance to traffic light or its status are studied. Fur-

thermore, traditional variables such as speed, spacing and speed difference between the following driver and its leader are also included in the analysis. This study, opposite to most of the literature reviewed, focus exclusively on urban environment instead of freeways. Traffic detection radar data of the PPA project in Amsterdam has been used. The data was not collected for the purpose of this thesis. Only a short road section of less than 100 metres is analysed, which unfortunately do not include the first 15 meters in front of the traffic light due to the radar range limitations. The main benefit of traffic radar technology is the amount of data available. Nonetheless, data measurements frequently contain errors, gaps and noise. Therefore, the first part of the thesis focuses on processing the data in order to get reliable predictors and response variables measurements. First, the noisy data out of the theoretical range of the radar is deleted. Then, trajectories are independently smoothed for x and y coordinates. Immediately after, measurements are mapped to lanes and variables such as speed and acceleration are computed. Later, the preceding assignment is carried out by first mapping all incomplete trajectories by means of a linear assignment problem. This step is essential and the suggested approach is considered a success. Despite losing 2% of reliable data due to wrong estimations, it is avoided that 10% of all reliable points present a wrong preceding assignment. Finally, the distance to traffic light and its status is registered to each measurement. Data analysis of the processed dataset depicts hundreds of thousand of leader and following reliable measurements. Although most variables obtained from data processing are accurate, spacing sometimes present extremely low values (<1m). This is mainly due to inaccurate position measurements of the radar, which do not always guarantee the front part of a vehicle as a reference point due to the radars' location.

Gaussian Process Regression (GPR) for machine learning (ML) has been used to take advantage from the large data set available and to ensure a complete model outside those space regions were no training data was found. GPR's mathematical approach offers a combination between traditional parametric models and machine learning techniques by incorporating the so-called basis function. The basic idea is that the GPR relies on the training data if new data input for prediction is not far apart of the training set, and it relies on a basis function (parametric model) when no correlation between new input and training data exists. In this project, the optimal velocity model has been chosen as an underlying model, i.e. basis function. The prediction of the GPR model is the mean acceleration and its variance (normal distribution). In this thesis it has been assumed that the acceleration prediction is directly the mean. Three types of conceptual GPR models have been trained in this master thesis. The first family of model belongs to GPR models fully optimised. This means that all hyper-parameters of the GPR are optimised during the training phase, including the parameters of the basis function. However results showed that in most of the cases optimising the basis function leads to bias results. In order to solve that, a second family of models has been trained by fixing the basis function. Moreover, in this case different optimisation procedure and objective function has been used to adapt the formulation to the needs of traffic theory. Finally, it was also decided to derive models without basis function, in order to see the real power of this machine learning technique.

In total, 47 models are tested and benchmarked according to two key performance indicators: the mixed error, an index that measures the error between two consecutive trajectories, and the total number of collisions. In every family of models, all possible variable combinations are included, leading to multiple models per family. The best GPR models achieve less than 20% of mixed error. This error is consistent with typical error ranges in literature. Furthermore, we have calibrated the OVM parametric model with the same radar dataset. This is carried out to analyse if there is an improvement by applying GPR formulation to turn a para-

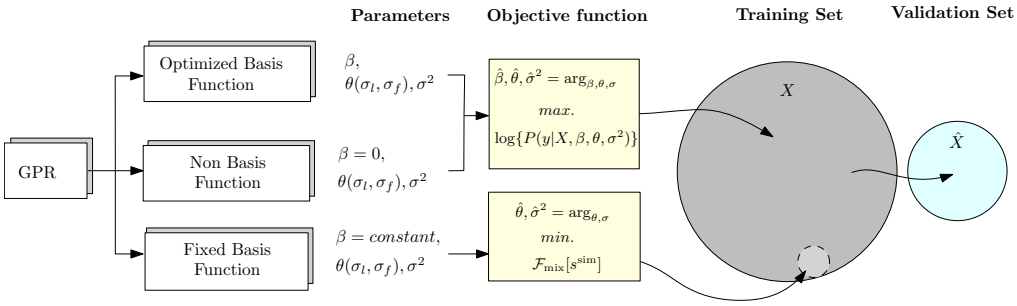


Figure 1: Conceptual Map of the methodology

metric model into an hybrid parametric and non parametric model. GPR model with fixed basis function scores better results than OVM in terms of mixed error in the space regions of the validation set (17.5% vs. 18.6%). Moreover, whereas OVM suffers significant overfitting issues and predictions become unrealistic outside validation set ranges, GPR formulation guarantee a complete model thanks to the fixed basis function. However, still the original OVM model ensures no collisions, while the GPR model occasionally predicts collisions. Taking a look to the variables importance, speed difference appears to be the most important variable to describe traffic behaviour. Opposite to OVM, GPR models are not capable to accurately describe traffic behaviour with only spacing and speed variables in the given the data set. The results also highlight that the status of the traffic lights affects traffic behaviour. Generally speaking, the most accurate models are achieved by including spacing, speed, speed difference and the status of the traffic light as predictor variables. Therefore, for first time in the literature reviewed in this thesis, the traffic light status is included into the mathematical formulation, becoming a relevant variable to explain the longitudinal traffic behaviour. Finally, distance to the traffic light seems not significantly affecting the results.

This thesis shows a new methodological approach of deriving mathematical hybrid models. Particularly, the thesis depicts how any process described by a parametric model, can be improved in a specific space regions where new data is available by using Gaussian Process Regression and machine learning techniques. Results of this thesis proved that GPR models can improve a traditional parametric car following model such as OVM in terms of performance, but still they present some violations of traffic physics (e.g. collisions) and computational times issues. On one hand, spacing, which should be one of the main relevant variables to describe traffic behaviour in stop and go traffic conditions, presents significant noise due to inaccurate position measurements. This issue highly affect GPR predictions, particularly in deceleration phases,as spacing measurements in training data re not completely accurate. On the other hand, computational time issues in prediction can be faced by precomputing offline of all the possible model solutions so they might not represent a big challenge for its real market application. Therefore, with a more accurate position measurement, GPR models will not face any problem and seems a promising new methodology to derive microscopic traffic models. Compared to parametric models, GPR formulation allows having a complete model and avoids overfitting the data. It also allows flexibility in its formulation and it is sure that the listed issues will be overcome in future research. Last but not least, few literature of machine learning techniques applied to car-following models is found, proving the innovative approach of this thesis. Results are not outstanding, but they definitely give some insights of a new powerful mathematical techniques that can be applied to describe driving behaviour or any modelled process.

Contents

List of Figures

List of Tables

1 Introduction

1.1. Motivation

Since the introduction of powerful microsimulation tools in the last decades of the 20^{th} century, the way that traffic engineers approach modelling and test transport infrastructures and traffic scenarios has completely changed. Nowadays engineers rely on microscopic traffic software to examine signalised roundabouts, optimise signalised intersection, to test a wide range of traffic management measures such as ramp metering, high occupancy lanes or rerouting due to road works, and to estimate traffic emissions among others. Fast PCs have made it possible to develop advanced traffic micro-simulation software packages. Today, the number of traffic microscopic simulation models is vast and the simulation approaches and model applications are to a large extent differentiated. The main objection of traffic microsimulation models is that any attempt to model the actions and interactions of individual vehicles is flawed as is challenging to understand and calibrate the controlling parameters (Wood, 2012). However, microscopic traffic modelling has opened up opportunities for engineers to tackle problems where conventional models were found wanting. Moreover, microscopic traffic models are, generally speaking, proved to accurately describe the traffic behaviour and its output such as average travel time or speed it is now being used as an input in macroscopic models (Olstam & Tapani, 2004).

Microscopic traffic models are based on a combination of mainly three models: car-following models, lane changing models and gap acceptance models. Car-following models are those sub-models that describes the interactions with preceding vehicles in the same lane (Olstam & Tapani, 2004). A lot of research has been carried out on this topic, from Gazis-Herman-Rothery (GHR) model at the General Motors research labs in Detroit in the fifties and earlier sixties until modern models such as the Intelligent Driver Model (IDM) in the current century. Overall, those models are seen as accurate and research is currently more focused on lane changing models, where still further investigation is needed. However, current microscopic car-following models present a smaller discharge rate on urban signalised intersections than in real observations (Hidas, 2006). In order to solve this issue, existing software's allow to set as an input the discharge of an intersection and then parameters are automatically calibrated accordingly to the discharge rate. Hence, this kind of models might miss several vari-

ables relationships making them not capable to accurately capture driving behaviour at urban signalised intersections. The main hypothesis is that drivers tend to accelerate more and assume extra risks in traffic lights such assuming relatively small spacing with the preceding vehicle to avoid waiting to an extra traffic light cycle. Other hypotheses are that drivers might be willing to accelerate if they have a clear vision of two cars ahead from them or that drivers may tend to smooth down their speed when approaching a signalised junction in red cycle in order to avoid stopping before the traffic light it turns green, i.e. coasting drivers' behaviour. This could mean that drivers pay less attention to preceding vehicles in this kind of situations. Nonetheless, no empirical demonstrations are performed and these hypotheses can only be found under the future academic research.

During the past decades, existent parametric car following models have been usually calibrated using traditional techniques and small yet accurate datasets. Traditional optimisation calibration techniques simply consist on maximising the fit of a particular parametric equation to the data, given a set of parameters and an objective function. A dataset collected in a test done by Robert Bosch GmbH in 1995 (Schulz et al., 2003) represents a clear example of datasets usually used in this techniques. It was carried out in Stuttgart in a one lane road with traffic lights and "stop and go" traffic conditions. An instrumental vehicle equipped with an on-board radar, registered speeds difference, acceleration and spacing between itself and its predecessor every 100 milliseconds during 300 seconds. Yet, it seems obvious that drivers' behaviour cannot be fully described by two vehicles driving during 5 minutes. Last years, large datasets are becoming available thanks to technology improvements and governmental efforts such as the Next Generation Simulation (NGSIM) datasets promoted by the Federal Highway Administration of the U.S National Department of Transportation (Federal Highway Administration, n.d.). Opposite to the before mentioned datasets, these datasets contain large amount of data, e.g. thousands of trajectories. Nonetheless, these datasets are plenty of errors and present noise. In the Netherlands, traffic radar detection technology has started to being used for traffic data collection. Opposite to other on-board units, traffic radar is installed in existent infrastructure such as light poles or traffic lights and simultaneously detects and register vehicle driving in the radar road section range. Thus, its main benefit is the large amount of data that can be collected from a single radar detector. Depending on traffic density, there can be up to 80 reflectors (vehicles) captured simultaneously at a frequency of 20 cycles per second in a maximum range of 300 metres. According to (Mende, 2010), traffic radars provide more accurate and reliable measurements for intersection control, vehicle counting and speed than loop detectors or basic GPS systems. However, doubts arise whether this data collection technology is accurate enough to derive microscopic traffic models. Occlusion and interference's might occur, leading to data gaps or data errors. Moreover, noise is frequently found in the measurements. Calibration of parametric models with radar data would not be beneficial. First, large amounts of data represents a challenge to traditional optimisation schemes due to computational times issues. Second, one of the main purposes of using large amounts of data is to fully capture drivers' behaviour and its stochasticity. However seems difficult to shape this behaviour in a fixed parametric model by only altering the values of its parameters. Third and last, parametric models might suffer from overfitting if the data is from an small space region. Alternatively, non parametric models derived from new machine learning techniques are rapidly becoming popular to deal with this type of datasets. Hence, this thesis investigates non parametric car following models using machine learning techniques to solve the listed issues. This approach allows the model to learn from the data instead directly relying on a parametric equation. Overall, using traffic radar technology and non parametric model is challenging and represents a new line of research.

The final goal of this master thesis is to obtain a calibrated car following model in stop and go urban traffic conditions. This projects aims to gain new empirical insights into longitudinal driving behaviour by means of the enumeration of a new non parametric car-following model using machine learning techniques. Traffic radar data of the arterial S106 road in Amsterdam is going to be used. The first aim of the thesis is to find dynamic (over time) relationship between variables such as spacing, acceleration and speed of preceding cars both in acceleration and braking phases. Moreover, other variables which are rarely include in currently car-following models such as distance to traffic light or its status will be studied. Then, the project will use the large data set to accurately describe a new model using Machine learning techniques. Finally a comparison between an existent parametric car following model and the model derived in this thesis will be conducted to test its performance. Generally speaking, the project might represent a considerable challenge as new data collection technology and new estimation model techniques are used.

1.2. Research Objectives and Questions

The final goal of this thesis is to obtain a calibrated non parametric car-following model at signalised intersection with stop and go traffic conditions by using machine Learning techniques and radar detection data. In order to overcome the challenges of the characteristics of new datasets available, non parametric models using machine learning techniques are investigated. Thus, the main research question and sub-questions to be answered are:

How can the longitudinal urban drivers' behaviour at signalized intersections be modelled using non parametric models and machine learning techniques?

(a) *What is the quality of preceding trajectories in the traffic radar detection data set?*

(b) *Which are the main significant dynamic variables relationship between preceding vehicles?*

(c) *How considerable are variable relationships that takes into account traffic light distance and status?*

(d) *Which Machine Learning technique should be used to fit the processed data taking into account the data characteristics?*

(e) *What is the new model accuracy terms of the selected KPI compared to existent parametric models?*

The sub-questions can be divided in two main groups. The first sub-questions, i.e. *a,b* and *c*, investigate on the radar tracking data quality and the variable relationships that can be derived from it. The second group of sub-questions, *d* and *e*, aim to find an appropriate model to use this data to accurately describe longitudinal drivers' behaviour. This set of research sub-question will help to finally answer the main research question.

1.3. Research Approach

The steps suggested to perform this master thesis are depicted in **Figure 1.1**. There are three main parts that need to be completed in this specific order: data, machine learning and discussion and conclusions. The output of each section provides input to the following one. The first part consist on reviewing car following models and also constists on describing, processing and analysis radar detection data. Then, in the second part, data will be used to fit a non parametric model using machine learning techniques. Finally, the third part includes the discussion and the conclusion of this thesis, where results and methodology are deeply examined.

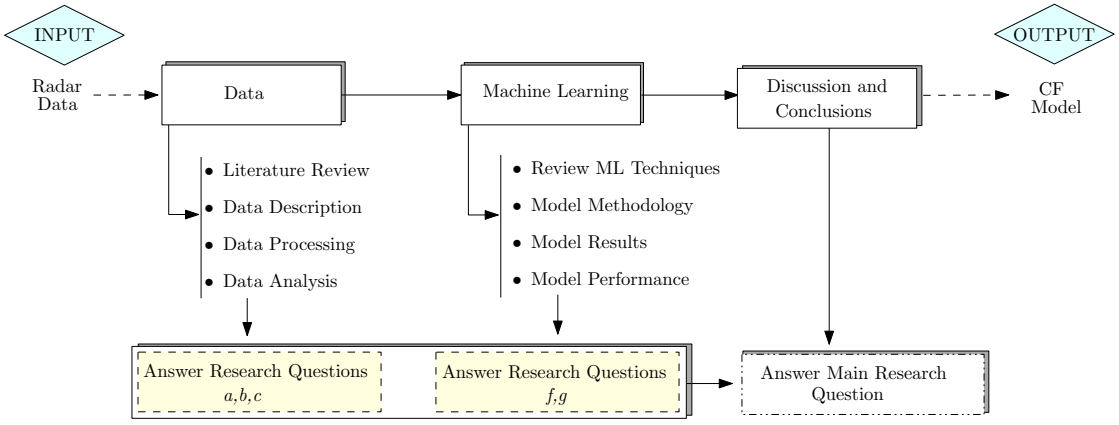


Figure 1.1: Flowchart of the Methodological Approach

1.4. Thesis Outline

A short description of the outline of the research project is given in this section. The designed approach was mas made to meet the research objectives of this master thesis and form the following parts and chapters and its respective content:

1 **Introduction**
Motivation of this master thesis and illustration of the research question and the research approach.

I Data

2 **Literature Review**
Provide a theoretical background on current car-following models and insights on the traffic radar detection technology.

3 **Radar Data Processing and Analysis**
Derivation of all variables needed to fit a car-following model from the dataset available.

II Machine Learning

4 **Machine Learning Techniques**
Introduction to machine learning techniques with a special focus on Gaussian Process Regression.

5 **Methodology for GPR Model Derivation and Validation**
Explanation of the proposed methodology to derive Gaussian Process Regression models from the available dataset and to validate the models.

6 **Results**
Results of all combinations of trained models.

III Discussion and Conclusions

7 **Discussions**
Critical assessment on the results and the methodology chosen.

8 **Conclusions and Recommendations**
Final conclusions and recommendations, including recommendations for current practice and future work.

I

Data

2

Literature Review

To reach the goal of this research thesis it is essential to carry out an extensive literature review of the past and current work in this topic. Hence, in Section 2.1, the most relevant available car-following models and particularly, its approach to signalised intersections, are examined. It is crucial to know which response and predictor variables are currently taken into account in each model. Later, Section 2.2 focuses on getting extra knowledge regarding traffic radar technology. This non intrusive road side data collection technique was used to collect the traffic data used in this master thesis.

2.1. Car-following models

Car-following models, also known longitudinal behaviour models, are mathematical models that controls drivers behaviours with respect to the preceding vehicle in the same lane (Olstam & Tapani, 2004). Most of the models consider three main regimes: car-following, free flow and emergency regime. If the driver behaviour is constrained by another vehicle downstream, it is consider that the driver is driving in the car-following regime, i.e. its acceleration partly depends on the behaviour of the downstream vehicle. When a vehicle is not constrained by any other downstream vehicle, it is consider that the vehicle is driving at the free flow regime and, in general, its driving at its desired speed. Finally, when a driver is in the emergency regime, the driver needs to make an emergency braking to avoid a collision.

2.1.1. Models Classification

Car-following models are commonly divided into classes or types depending on the utilised logic (Olstam & Tapani, 2004). (Brackstone & McDonald, 1999) suggests 5 different families while other researches like (Saifuzzaman & Zheng, 2014) includes two extra family to incorporate models of the 21^{st} century. The following subsections will describe each family of car-following models.

Gazis-Herman-Rothery models (GHR)

Gazis-Herman-Rothery (GHR) models, developed in the General Motors research lab in Detroit, are one of the most well-known and old family of models (Brackstone & McDonald, 1999). The set of model describes the vehicles acceleration according to its own speed, the speed difference and space headway between the vehicle and its leader. According to the original model enumerated by (Chandler et al., 1958) in 1958, driver acceleration is described as a stimulus-response function. The acceleration is proportional to increment of speed, or deviation from a set following distance, which can itself be speed dependent. During the second part of the 20^{st} century, several researchers tried to improve its accuracy by converting the equation formulation to a non-lineal function, resulting in new calibrated and validated adaptations of the original GHR model such as (May Jr & Harmut, 1967). However, it is now being used less frequently, significantly because of the large number of contradictory findings (Brackstone & McDonald, 1999).

Safety distance or collision avoidance models

The safety distance or collision avoidance models seek to specify a safe following distance within a collision would be avoidable. The main difference with GHR models is that drivers react to the spacing between itself and the preceding vehicle, rather than their speed difference. The most well known model of this family was formulated by (Gipps, 1981). In Gipps car-following model, vehicles are classified as free or constrained by the vehicle in front. When constrained, the follower tries to adjust its speed in order to obtain safe space headway to its leader taking into account the maximum deceleration rate of both vehicles. A specific headway is considered safe if it is possible for the follower to respond to any reasonable leader action without colliding with the leader. When free, the vehicle's speed is constrained by its desired speed and its maximum acceleration (Olstam & Tapani, 2004). Although it is an old model, it is still considered an accurate model an even a calibrated version of the Gipps model is used in AIMSUN traffic simulation software.

Optimal Velocity Models

The optimal velocity model (OVM), introduced by (Bando et al., 1995) has received considerable attention in the car-following literature according to the number of literature available. The model assumes that each vehicle has an optimal (safe) velocity, which depends on the spacing from the preceding vehicle. Then, the acceleration of the vehicle can be determined according to the difference between the actual velocity and the optimal velocity. Note that the model is simply described by two explanatory variables. Although OVM model was designed to address the issue of the unrealistically high acceleration and deceleration observed in Newell's model, comparison with the field data shows that it still produces high unrealistic accelerations and decelerations. The reason is that the optimal velocity is dependent on spacing. Hence, the density is still affecting the model (Saifuzzaman & Zheng, 2014). In order to solve that, (Helbing & Tilch, 1998) added velocity difference as a predictor variable to the OVM model.

Psychological or action point models (AP)

Psychological or AP models are based in the assumption that a driver will perform an action when a threshold is reached (Panwai & Dia, 2005a). The most famous model in this family is the one formulated by (Wiedemann, 1974) and improved in (Wiedemann & Reiter, 1992). Both models are currently used in the well-known platform simulation VISSIM (PTV).

Three main threshold in the speed difference-spacing plane are implemented leading to four regimes: free driving, following, closing in and braking mode. Acceleration is computed according to the regime and a set of parameters. Another famous model of this family is the car-following model proposed by (Fritzsche, 1994). The main difference with Wiedmann model is that following regime is dived in two, leading to an extra total regime. A adaptation and calibrated version of this model is currently used in PARAMICS.

Fuzzy logic-based models

Fuzzy logic-based models try to divide their inputs into a number of overlapping 'fuzzy sets', each one describing how adequately a variable fits the description of a 'term'(Brackstone & McDonald, 1999). A common used fuzzy rule would be: "if the spacing with the predecessor is -close- and the speed difference leads to -closing-, the driver response would be -brake-". The original model of this family was formulated by (Kikuchi & Chakroborty, 1992), who attempted to 'fuzzify' the traditional GHR model. However, this family of models have been rarely used in practice as its formulation seems not realistic as acceleration of the leader vehicle is incorporated, which is highly debatable whether a following driver can notice it (Brackstone & McDonald, 1999).

Desired measures models

Further studies of the original lineal GHR model, took Helly to enumerate a new linear model in 1959. He proposed to include additional terms in the original formulation. His idea was that acceleration of a driver varies according to whether the vehicles in front are braking. According to (Brackstone & McDonald, 1999), the linear model has little uses in current simulation models. Recently, one of the most popular models using desired measures is the intelligent driver model (IDM) proposed by (Treiber et al., 2000). This model describes the acceleration of a driver according to both the desired speed and the desired space headway. Later, (Treiber & Helbing, 2003) extended IDM to capture driver's adaptation effect to the surrounding environment using a memory function (IDMM) (Saifuzzaman & Zheng, 2014). The extension of the model is based on the observation that, after being in congested traffic for some time, most drivers adapt their driving style such as increasing their preferred time gap.

2.1.2. Approach at Signalised Intersections

Generally speaking, all before mentioned mathematical models do not explicitly take into account traffic lights. Usually, models formulation allow to simulate a traffic light as a standstill vehicle. For instance, when the traffic lights becomes red, a virtual standstill vehicle is placed in the location of the traffic light, forcing vehicles to decelerate. The leader vehicle instantly becomes a follower of an invisible vehicle -the traffic light-, and its deceleration rate is computed depending of the model explanatory varies such as spacing (distance to the traffic light), its own speed and the speed difference (own speed as traffic light is an obstacle). Hence, models assume that the driver behaviour towards a traffic light is identical to a standstill car (i.e. same parameters and variables). Moreover, models generally assume that vehicles following a leader vehicle who faces a red traffic light, will not drive according to traffic light variables, only as a function of the leader vehicle itself. Thus, traffic lights are taken into account only in the first vehicle upstream the traffic light. There have some studies which try to simulate drivers behaviour in traffic lights, but they especially focus on the decision making of drivers to accelerate or brake when the traffic light turns yellow (Kesting & Treiber, 2008a).

2.1.3. Conclusions

All reviewed model assume identical drivers behaviour between vehicles and obstacles (i.e. traffic lights). Moreover, traffic lights are only taken into account in the first vehicle upstream. These both assumptions contradicts one hypothesis of this thesis: traffic lights status might influence drivers' behaviour. To summarise this section, **Table 2.1** gives an overview of the main car-following models and the variable considered.

Table 2.1: Overview of independent variables taken into account in current car-following models

	Models						
Variables	GHR	Gipps	Helly	Wiedmann	Fritzsche	IDM	OVM
Acceleration	***R***	-	***P*&*R***	***P*&*R***	***R***	***R***	***R***
Speed	***P***	***P*&*R***	***P***	***P***	***P*** [2]	***P***	***P***
Speed Difference	***P***	-	***P***	***P***	***P*** [3]	***P***	-
Spacing	***P***	***P***	***P***	***P***	***P***	***P***	***P***
Distance to the traffic light	-	-	-	-	-	-	-
Status of the traffic light	-	-	-	-	-	-	-
Number of cars downstream	-	-	***P*** [1]	-	-	-	-

Legend: '**R**' response variable, '**P**' predictor variable, '**P**&**R**' predictor and response variable, '-' not included.

[1] Two cars downstream are considered $(n-1)$ and $(n-2)$.

[2] Own speed (v_n) and the preceding vehicle speed (v_{n-1}) independently.

[3] Difference of two squares between speeds.

2.2. Radar Technology for Traffic Data Collection Purpuses

Radar is defined in (Booth & Kurpis, 1993) as "a device for transmitting electromagnetic signals and receiving echoes from objects of interest such as targets within its volume of coverage". For traffic data collection purposes, two kinds of radar can be distinguished: on-board driver units and road side units. In this master thesis, we will refer to radars as those road side units which are permanently installed to road infrastructure such as light poles or traffic lights. The echoes (reflections) from individual vehicles are picked up by detection signal processing software which computes the location, radial speed and length of vehicles. Moreover, new radars incorporate innovate algorithms that automatically assign ID (unique vehicle identifiers) to all logs, by means of mapping consecutive observation. Traffic radar for traffic purposes is considered a non intrusive vehicle detection technology as systems are positioned in light pols or traffic lights next to the road. Consequently, road streams do not to be closed during its installation or maintenance. The main benefit compared to other intrusive and non intrusive road side data collection technologies is the large amount of data that can be collected from a single radar detector in a short timespan. Depending on traffic density, there can be up to 80 reflectors (vehicles) captured simultaneously at a frequency of 20 cycles per second in a maximum range of 300 metres (Mende, 2010). Furthermore, opposite to intrusive traffic data collection techniques such as classic inductive loop detectors, radar detection requires few maintenance and installation time and they are not subject to roadway conditions overall leading to a cheaper technology (Medina et al., 2012).

(Mende, 2010) argues that radars provide more accurate and reliable measurements for intersection control, vehicle counting and speed than loop detectors or regular GPS systems. According to all of the above said, the radars' output are apparently complete ideal trajectories of single vehicles in a specific road section (radar range). However, this only works in theory. Real measurements clearly show that this technology is not perfect. Occlusion and interferences regularly occur, leading to data gaps or data errors. For instance, radars might report stopped vehicles, although the radar might not really detect stopped vehicles as vehicles are too close between each other at similar speed (standstill). The radar detectors incorporate an algorithm that keep track of moving vehicles, and based on past observations, it can then guess where a vehicle must be standing still at a certain location. Although the algorithm generally is pretty accurate, it does make mistakes. Sometimes, the data might indicate stopped vehicles that actually are not there anymore or the same ID is assigned to different vehicles. Furthermore, occlusion causes extra gaps in the data, especially when there is a queue and vehicles are close to each other (see **Figure 2.1**). Generally, radars can only be positioned in light pols with the consecutive location and height restrictions. Consequently, usually radars are not able to measure vehicles behind a queue of about 50-75 metres due to occlusion. Taking into account data errors and gaps, data processing before the analysis is essential to avoid bias results.

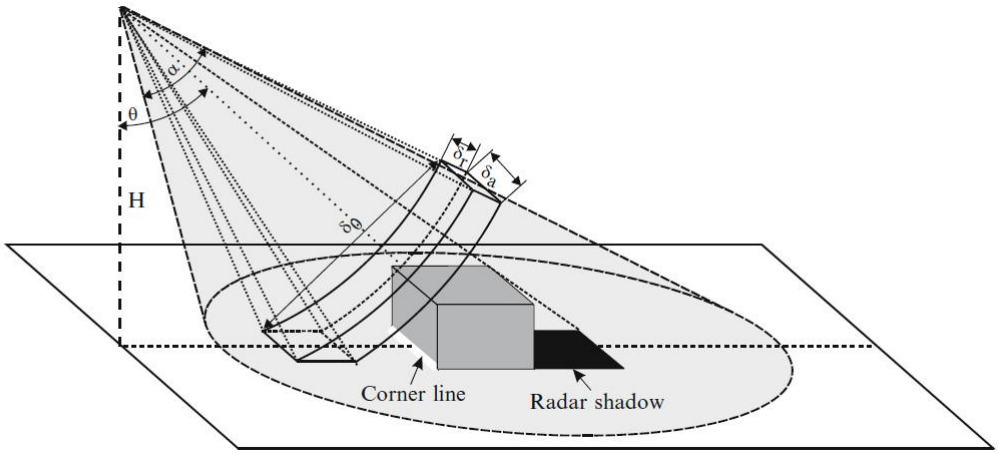
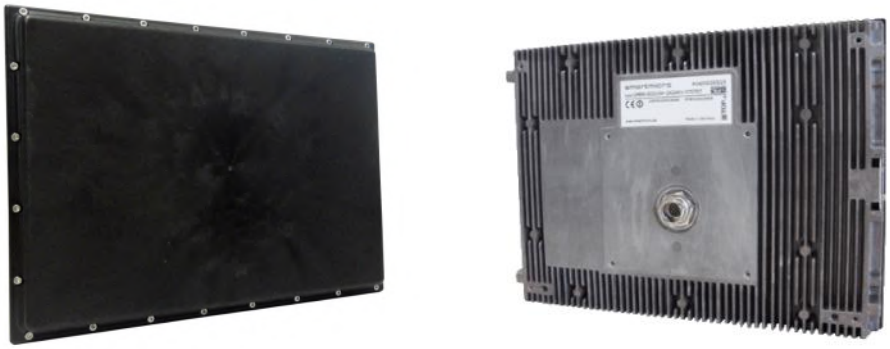


Figure 2.1: Sketch: Radar occlusion (Soergel, 2010)

This last decade, Radar detection technology has started to commonly being used for traffic data collection purposes thanks to the technology improvement in terms of vehicle accuracy detection. Radars has been recently successfully used in the Praktijkproef Amsterdam or PPA (Field Operational Test Integrated Network Management Amsterdam), one of the first large-scale field operational tests testing coordinated network wide deployment of traffic management in practice (Hoogendoorn et al., 2013). Radar devices were used to track vehicles at intersection level in order to estimate queue length of traffic road streams. Overall, this non-intrusive traffic detection technology proved its validity and seems a promising new line of research. Apparently, it can perfectly substitute classic inductive loop detectors for intersection management. However, a lot of doubts arise if radar technology can substitute high precision GPS or radar on-board devices to describe traffic behaviour, and even calibrate or validate current models car following models. None studies have been published yet regarding this topic. Compared to high precision on-board devices, main benefits is the amount of data available with only one device and the main drawback is the noise and gaps errors

in vehicle tracking. Overall, using this technology for our purpose represents an enormous challenge.

For the development of this thesis, it has been used data collected from 5 radars provided by the German manufacturer Smartmicro during the before mentioned PPA project in Amsterdam. Each radar measured the location, speed and length of vehicles relative to its own location. Four out of 5 radars were detectors *UMRR-0C Type 40* and the other one was *UMRR-0A Type 30*. While the first type of radar is characterised by a high range up to 350m on passenger cars / 450m on trucks, and narrow beam of <± 18°, the second type is characterised by a small range up to 105m and a wide beam of <±35°. Type 40 can detect more vehicles simultaneously up to a maximum of 256 vehicles compared to 64 of type 30. According to the manufacturer, both type of sensor provides excellent vehicle separation capabilities if there is a radial speed value difference between vehicles greater than 0.25 m/s or there is a different range value between vehicles by 2 to 6m. If a vehicle is detected, the range and speed accuracy is outstanding, with <±0.25% m and <± 1% m/s respectively. Extra information of both radars specifications can be found in (Smartmicro, 2017) and (Smartmicro, 2016). The choice of radar type during the PPA project was done according to the location characteristics as showed in section 3.2.1.



(a) front *(b)* rear

Figure 2.2: Tracking Radar UMRR-0C Type 40 (Smartmicro, 2017)

3

Radar Data Processing and Analysis

Chapter 3 explains how all set of predictors and response variables have been derived from the given traffic radar dataset. Overall, the chapter proves how challenging is to use a radar data technology to describe microscopic drivers' longitudinal behaviour. After a sharp introduction in Section 3.1, the chapter includes the data description in Section 3.2, which inspects the raw data and summarises the errors and challenges of using this data collection technique. Then Section 3.3 contains all steps carried out in order to get reliable predictor and responses variables measurements. Finally, Section 3.4 analyses the data and discusses next steps in this thesis.

3.1. Introduction

This chapter includes the process of inspecting, processing, and modelling data with the aim of discovering useful information. Raw data usually include errors, gaps and noise. Thus, it is needed an extensive process to derive reliable variables measurements from the data before starting the analysis of the data and fitting a model. **Table 3.1** and **Figure 3.1** depict the response and predictor variable proposed in this thesis. The response variable of this thesis is the acceleration. This means that we aim to fit a model that describes the driver acceleration of vehicle n at time step t based on a set of predictor variables on $(t,...,t-k)$ time steps. The proposed predictor variable are the own speed of vehicle n, the spacing distance and the speed difference between vehicle n and vehicle $n-1$ -downstream vehicle-, the distance of vehicle n to the downstream traffic light, the status of the downstream traffic light of vehicle n and the number of vehicles downstream of vehicle n.

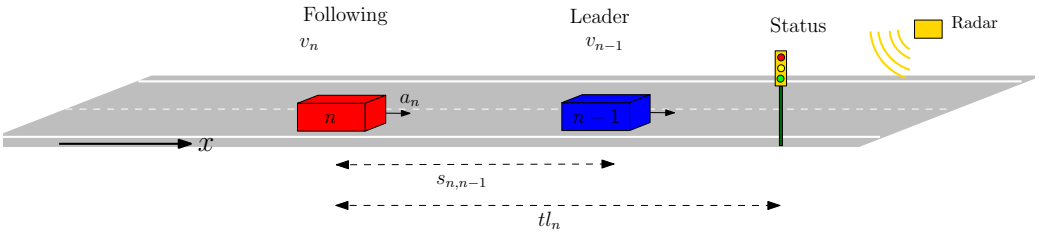


Figure 3.1: Response and predictor variables

Table 3.1: Response and predictor variables

Response Variable	Units	Predictor Variables	Units
Acceleration$_n$	m/s^2	Speed$_n$	m/s
		Speed Difference $_{n;n-1}$	m/s
		Spacing$_{n;n-1}$	m
		Traffic Light Status$_n$	Dummy variable
		Distance to traffic light$_n$	m
		Number of cars downstream$_n$	bins

3.2. Data Description

This master thesis has used radar tracking data from the S106 arterial road in Amsterdam. The data was collected during the PPA project during the months of June and July of 2016. This section aims to give insight into the provided radar data. It first describes the radars location and the description of the road streams analysed. Later it describes the content of the data and ends with the explanation of the data quality and its challenges for the thesis purpose.

3.2.1. Location

The five radars were located in the southeast of the city-center of Amsterdam in June and July of 2016 (see **Figure 3.2**). The devices were positioned in the S106 arterial road close to the Dutch national freeway A10. Radars were located in that specific spot to accurately estimate vehicle queues length in the prior intersection to the north and south on-ramps to the highway. The idea behind this measure was to effectively coordinate traffic flows through the network, distributing queues over intersection streams in order to avoid traffic gridlocks (Hoogendoorn et al., 2013). The figure also depicts several real trajectories from the 17^{th} of June of 2016 tracked by each radar. The theoretical ranges of the radars can also be seen in the same figure. For the development of this thesis, only data from radar IV is going to be used. Radar I tracked vehicles at the southern on-ramp. Data collected by this radar might not be relevant for the thesis purpose as on-ramp traffic light signalling is quite different compared to a regular one. Radar II and III were located at the east side of the S106 arterial road. However, those radars were positioned too far away of the traffic signal. Furthermore, the radars did not have a direct line of vision with the traffic light due to natural obstacles such as trees that might have caused occlusion and data gaps. Radar IV and V were located in the west side of the S106 arterial road and generally its position suits the requirements. They covered a long road stretch of 450 metres approximately, including a traffic light. Both devices were tracking vehicles coming from the west and aiming to access the A10 via the northern or southern on-ramp, or continuing the S106 to the city centre of Amsterdam (see **Figure 3.3**). Note that in between both traffic directions there exist a tram track with a stop 100 metres away of the traffic light. Trams and pedestrians were usually tracked by both radars.Finally, taking into account all the above mentioned, it has been considered to analyse only data from radar IV. This radar range covers 100 metres approximately from to the traffic light and it is assumed that vehicles will not change their driving behaviour before this range. Moreover, if radar V had been taken into account, there would have existed several traffic streams in the analysis

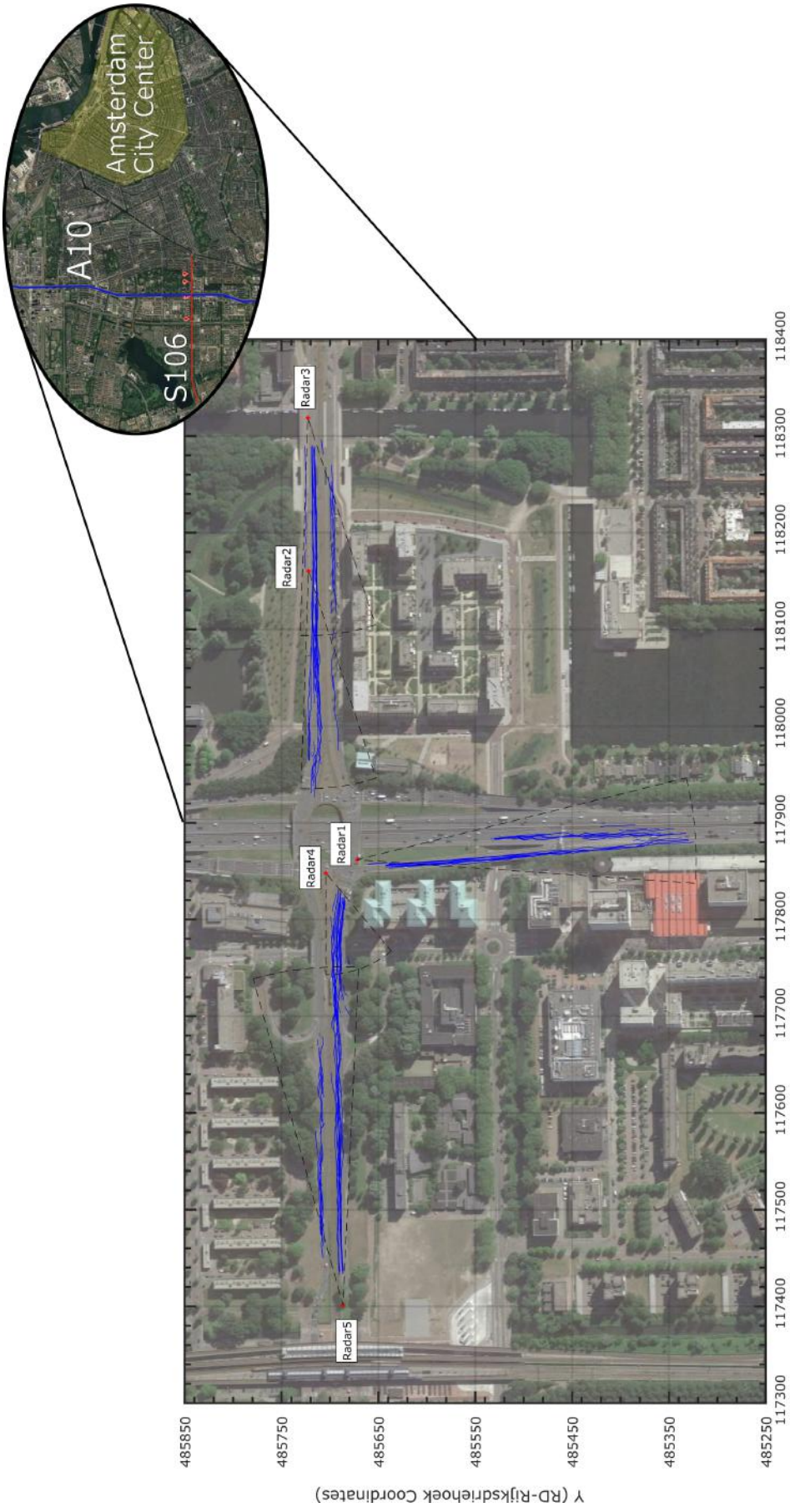


Figure 3.2: Radars' location. Real trajectories from the 17^{th} of June of 2016 tracked by each radar (blue lines).

as there exist a split and an incorporation in its road stretch. Also, the ID assigned to the same vehicle are different per radar, meaning that some extra work needed to be done to match both radars. Hence, it seems reasonable to do not study data from radar V and avoid overloading the workload of this thesis.

Figure 3.3 depicts selected road stretch. As it can be observed, it covers 150 metres approximately before a set of traffic lights. Thanks to the standard infrastructural design of this road stretch, results from this thesis might be extrapolated to other urban intersections. The road presents different number of lanes depending on the distance to the traffic light. First, 150 metres away from the traffic light and the radar, three lanes with different directions are found. Right lane is frequently used by vehicles that want to turn right and access the southern on-ramp of A10. Centre lane is used to continue in the S106 arterial road towards Amsterdam city centre. Finally, left lane is usually used by drivers to access the northern on-ramp of the A10. An extra lane is added to the left and centre lane close to the traffic line to increase the intersection accumulation capacity. Each of the three directions has an independent traffic light (independent signal) with standard Dutch coding: left turning -09.1 and 09.2-, straight -08.1 and 08.2- and right turning -07.1 and 07.2-. Finally note that the radar is not completely perpendicular to the traffic streams. This might represent a problem as the radar usually detects the closest part of the vehicle, which might not be the front part.

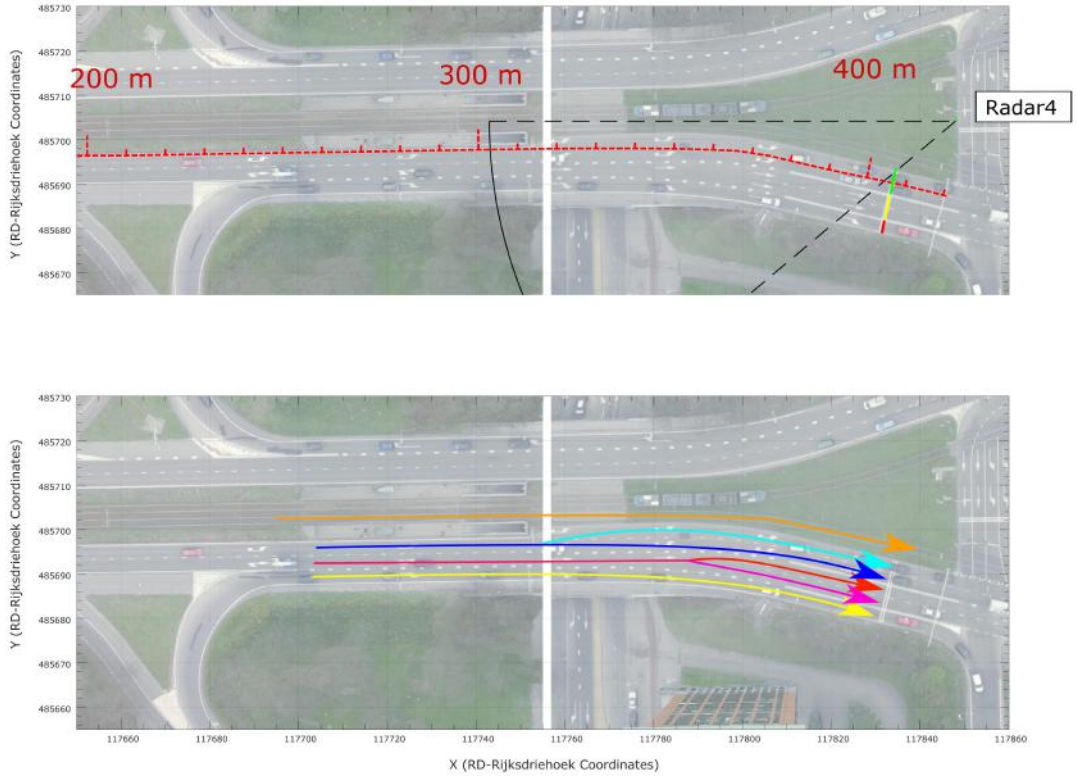


Figure 3.3: Road section to be analysed

3.2.2. Raw Data

The data contains measurements from last June and July 2016. Each radar measured the location, speed and length of most vehicles relative to its own location. Later, the data owners converted the locations into map coordinates in Rijksdriehoekstelsel (the Dutch Mercator

projection) and they gave a reliable flag to each log according to conditions explained later in this subsection. The radar time interval was 0.2275 s, meaning that every 0.2275 s, information of all vehicles in the road was collected. **Table 3.2** enumerates the information in each vehicle log, i.e. measurement.

Table 3.2: Raw data included in each measurement

Measurament	Unit	Description
ID	-	Unique identifier per vehicle
Xpos	m	X component in *Rijksdriehoekstelsel* coordinates
Ypos	m	Y component in *Rijksdriehoekstelsel* coordinates
Xspeed	km/h	X component of the speed (towards / from the radar)
Yspeed	km/h	Y component of the speed (sideways)
Time	HH:MM:SS.FFF	Time
Vehicle Length	m	Vehicle length of the measured vehicle
Reliable	0 or 1	Reliable flag given based on certain conditions

Radar automatically assigned ID's to each vehicle, i.e. unique vehicle identifiers, to each measurement based on past observations. Moreover, the owners of the data in a first data analysis gave a flag to all measurements: (1) reliable and (0) unreliable. The unreliable flag was based on the following observations:

i When the radar stopped observing a vehicle because it was too close and thus, out of the range, an internal radar algorithm extrapolated where the vehicle had to be based on its past trajectory. This extrapolation was only performed over the x-axis, which means that the y-location was exactly constant (relative to the radar). This process generally gave quirky trajectories, and since this project is only interested in real observations the flag of these extrapolations has been kept as unreliable. This is usually observed next to radar IV close to the traffic light stop line

ii In a few cases the radar detected bizarre trajectories of a vehicle that flipped back and forward between two lanes. In those cases, the measurements were also flagged as unreliable

There were quite a few communication issues with the radars during the PPA project in Amsterdam. Thus, the data is incomplete with some radars being offline for complete days. Another fact to take into account is that there were roadworks on the west side of S106 for the entire month of July, completely blocking one lane. For the performance of this master thesis, traffic light data is also available. In this road stretch, there exist three different independent traffic lights with standard code 9.7 (right turn), 9.8 (straight stream) and 9.10 (left turn). An event was registered each time a traffic light changed its state (see **Table 3.3**). For instance, if a traffic light turned red, the event 3 and time were registered.

3.2.3. Data Quality

One of the main challenges of this thesis is to use radar data to describe drivers' longitudinal behaviour. Hence, the quality of the data provided necessarily needs to be examined. However, this project does not have any ground truth data, so all evaluation will be carried out by

Table 3.3: Traffic light state description

Event	State Traffic Light
0	Unknown State
1	Green State
2	Yellow State
3	Red State
4	Signal Off

traffic knowledge and common sense. A first insight into the data can be examined from X-Y plots, time-space diagrams and lateral positioning histograms.

Figure 3.4 depicts X-Y plots of radar IV the 17^{th} of June, between 9AM and 10AM. Each dot represents a log (measurement). Blue logs are labelled as reliable, while red logs are not reliable measurements (estimated by the radar). Note that most of these unreliable data are found either out of the range of the radar or next to the radar itself. Range detection of radars is much greater than the range stated in the certifications. However, out of the specified range, data seems less accurate and lanes are rarely distinguished. Thus, it might be a wise measure to delete (cut) the data in the radar range to avoid using inaccurate data.

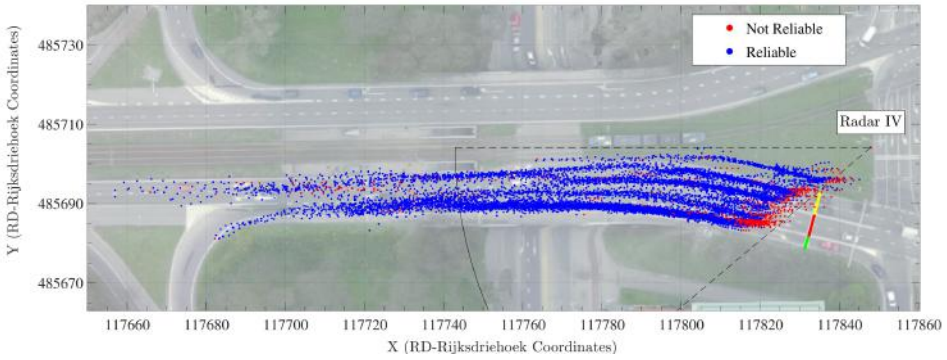


Figure 3.4: Aerial (X-Y) scatter of Radar IV observations. Data from 17/06/2016 from 09:00AM to 10:00AM.

It is also possible to create the so-called time-space diagrams, which are used to clearly depict vehicles trajectories. Each measurement is projected to the reference line and then 2D diagrams over time and space become possible. **Figure 3.5** shows time-space diagrams from radar IV the 17^{th} of June of 2016 from 09:15AM to 09:18AM. Each log is coloured according to the mapped lane. It can be easily seen that there are gaps in the raw data. Trajectories are incomplete, leading to single trajectories with few points. The quality and accuracy of the measured data seems right as no bizarre trajectories are found. Only measurements close to radar IV (410-430 metres in the reference line) present inaccurate behaviour such as spontaneous lane changing or long standstill phases. Fortunately, most of this measurements are already marked as unreliable.

Another way to visually check the accuracy of the radar detection is performing histograms of the lateral position of position measurements at a certain road section. In a specific road section, the perpendicular distance to the reference line is computed for each log and then is grouped in intervals of 0.5m. **Figure 3.6** depicts the lateral positioning on the road of radar IV measurements 50 metres away of the traffic light the 17^{th} of June of 2016 from 9AM

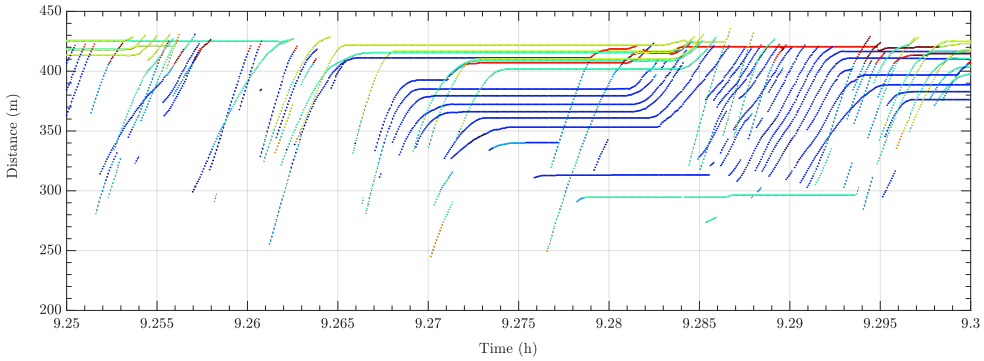


Figure 3.5: Time-space diagram of radar IV raw data. Data from 17/06/2016 from 09:15AM to 09:18A.

to 10AM. On this specific spot, 4 lanes are found. From the histogram can be clearly observed the four lanes. Both turning left lanes have a similar number of observations. This fact seems reasonable as traffic is usually uniformly distributed between same direction lanes at stopping lines. Centre lane can also be observed in between 3.5m and 6.5m perpendicular from the reference line and in this case few logs are measured. Finally, the right lane with a high number of observation can be mapped from 7.5m to 10.5m perpendicular from the reference line. Note that a higher number of observations can mean two things: higher flows or higher red traffic light phase time, which increase the number of measurements due to standstill vehicles longer times. Overall, the mapping accuracy of the radar seems acceptable and suitable for this project.

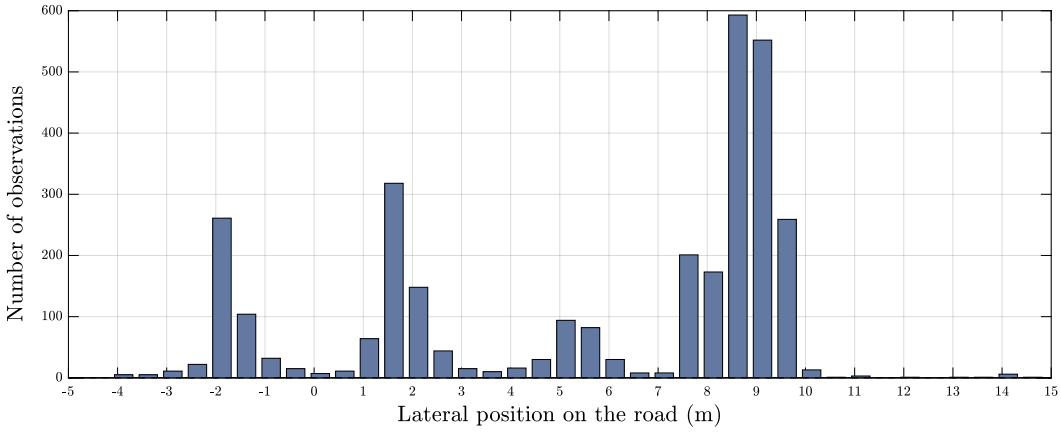


Figure 3.6: Lateral positioning radar IV raw data on the road at road section 50 metres away from the traffic light (380 m reference line (see **Figure 3.3**). Data from 17/06/2016 from 9AM to 10AM. Bins of 0.5 metres.

The previous figure showed that accuracy of individual points seems adequate. However, single trajectories of vehicles are noisy within a lane (see **Figure 3.7**). X and Y coordinates varies substantially within consecutive measurements, leading to instability. If speed and acceleration were directly be derived from the noisy position data, they would also inherit the noisy behaviour. The radar manufacturer does not ensure that every log of the radar belongs to the same part of the vehicle. For instance, imagine that the first measurement of the radar, it uses the position of the front part of the vehicle as a reference point, but later in the second measurement, the measurement belonging to the right part of the vehicle is used. This leads to instability between consecutive logs, specially when vehicles are standstill.

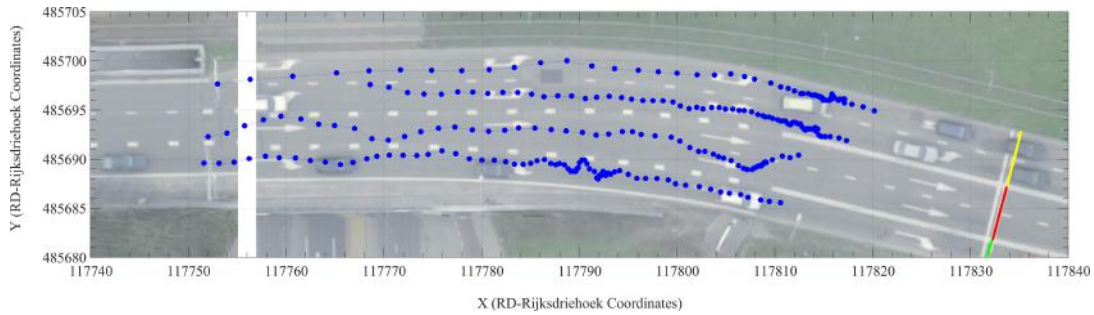


Figure 3.7: Example unstable/noisy trajectories. Only reliable points are illustrated.

Finally, radars also measure vehicle length. This measurement seems not reliable. Most of the vehicles contain the same measurement (either of 4.6 or 5). Moreover it varies within the same vehicle measurements. Generally speaking, the data is good enough for vehicle classification. A clear distinction can be found between trams and trucks -higher lengths-, cars - medium lengths- and pedestrians/motorbikes -small lengths-. However, it is not accurate enough to include it in microscopic calculations, e.g. computation of net spacing.

Summarising all mentioned above, the data main problems are:

i Data errors

ii Data gaps

iii Inestability or noisy trajectories

iv Vehicle length reliability

Using this data is seen as an innovative great opportunity. It is out of the scope of the thesis to improve the data characteristics (i.e. accurate estimation of missing trajectories). However, the project needs to deal with the current data quality. Consequently, it is essential to deeply examine the data quality and exactly know per log if we can rely on it or not. Setting a big amount of data as a not reliable should not represent a problem, as we fortunately have a big sample. The following paragraphs deeply depict the fourth main data quality problems and explain why they do represent a problem for our thesis, and explains how to solve it.

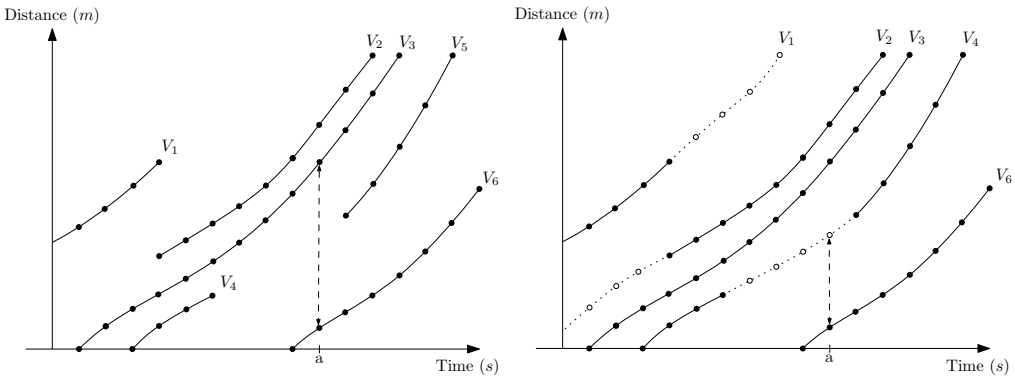
Data Errors

Data errors do not represent a big issue for this thesis. Fortunately, data errors are mainly non reliable logs such as radar estimation measurements. For this reason, those logs are not going to be used for data analysis. For instance, if a log of a leader car is labelled as unreliable, the log is not going to be considered even if the preceding vehicles logs are reliable. At the same time, in order to reduce the probability of having inaccurate reliable logs, data too close and too far away of the radar theoretical range will not be taken into account in the analysis.

Data Gaps

Space-time trajectory graphs clearly depict the existence of data gaps (incomplete tra-

jectories). **Figure 6.4a** illustrates an example of this particular issue. Usually, trajectories are not tracked from the theoretical start range of the radar (V_2) or the other way around, vehicle measurements stop quite far away from the radar (V_1). Sometimes vehicles are tracked at the beginning and at the end, missing some logs in between. Moreover, radar assigns each part of the trajectory to a different ID when in reality both trajectories belong to the same vehicle (V_4 and V_5). All these issues represent a major problem for our aim to study the relationships between preceding vehicles. For instance, imagine that we are studying the log of vehicle V_6 at time a. As trajectories are incomplete, the preceding vehicle of V_6 seems to be V_3. However, in reality, the preceding vehicle of V_6 is V_4 (see **Figure 6.7b**). Hence the importance of having complete trajectories. As previously mentioned, this thesis do not aim to estimate accurately those missing trajectories. However, an option is to discard all those reliable logs that it is sure that there is a vehicle in-front but there is no direct information of it. A way to do so, is to map trajectories such as V_4 and V_5, and create invented logs by interpolating last point of V_4 and the first point of V_5. Moreover, we can also extend trajectories and estimating missing logs by assuming the speed of last logs (V_1 and V_2). Then, all estimated logs must be labelled as not reliable. This process is discribed in detail in section section 3.3.3.



(a) Before mapping trajectories *(b)* After mapping trajectories

Figure 3.8: Before and after of mapping vehicle trajectories

Noisy trajectories

Another relevant problem is that trajectories seem to be quite unstable due to noisy observations. Most of vehicles seem to oscillate within a lane, especially close the the maximum radar range, when vehicle speeds are low or even standstill. This might indicate a lack of accuracy in the car detection as it has been explained before. **Figure 3.9** visually depicts the difference between the real trajectory and the observed one by the radar. This observed behaviour sometimes causes momentary lane changing, meaning that for a single or few logs a vehicle changes lane and returns to the previous lane. This is considered as unrealistic behaviour. In order to solve this issue, x and y positions are going to be smoothed. Moreover, it is going to be assumed that each log refers to the front of the vehicle as the radar is approximately located perpendicular to the vehicles. This assumption becomes important to compute one predictor variables: net distance between vehicles, i.e. net spacing.

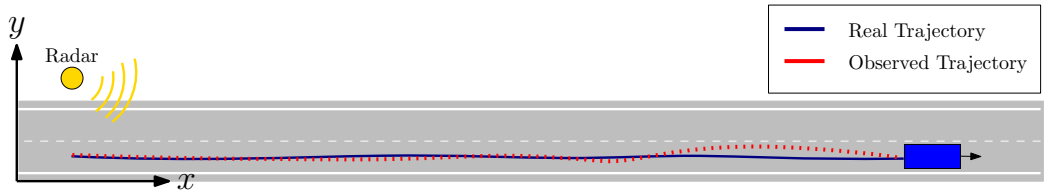


Figure 3.9: Sketch: Real versus observed vehicle trajectory

Vehicle length measurement

The raw data also includes the vehicle length of each log. However, they are not reliable. Therefore, we will use a standard vehicle length of 5 metres in the calculations of the spacing.

3.3. Data Processing

The main goal of the data processing is to get reliable response and predictor variables measurements. The data processing steps are summarised in **Figure 3.10**. First, the data out of the range of the radar is deleted. Then, data is smoothed independently for both x and y coordinates. Once the lanes are mapped and variables such as speed and acceleration are computed, the preceding assignment is carried out. Later on, the traffic light status is assigned to each measurement. Finally, the rest of variable needed for data analysis are computed. The following subsections precisely describe each of the steps in the data processing.

3.3.1. First Filtering Process

The first step in this process is to delete some parts of the data that are actually out of the range provided by the radar owner. It will be beneficial to delimit the data to the considered theoretical threshold provided by the radar manufacturer. Data from radar 4 has been delimited until a range of 105 metres -theoretical range provided by the manufacturer-. Moreover, in the same radar, logs closer to 15 metres have also been deleted. Most of these logs are not reliable as only X position component is measured assuming constant Y position (horizontal lines close to the radar).

3.3.2. Smoothing Trajectories

Position data is noisy within a lane. As it is shown in **Figure 3.7**, consecutive data points have great variability within consecutive x and y positions, leading to unrealistic trajectories. Speed and acceleration will be derived from the position data and thus, they would inherit this instability. Hence, is it strongly recommended to smooth the position data to avoid transmitting the noisy behaviour to the rest of variables. Alternatively, we could use the speed measured by the radar and derive acceleration from it. Nonetheless, from practice it is highly recommended to derive all variables from either measured speed or measured position and to not combine both of them. This subsection will also depict the difference on speed and acceleration by using both alternatives.

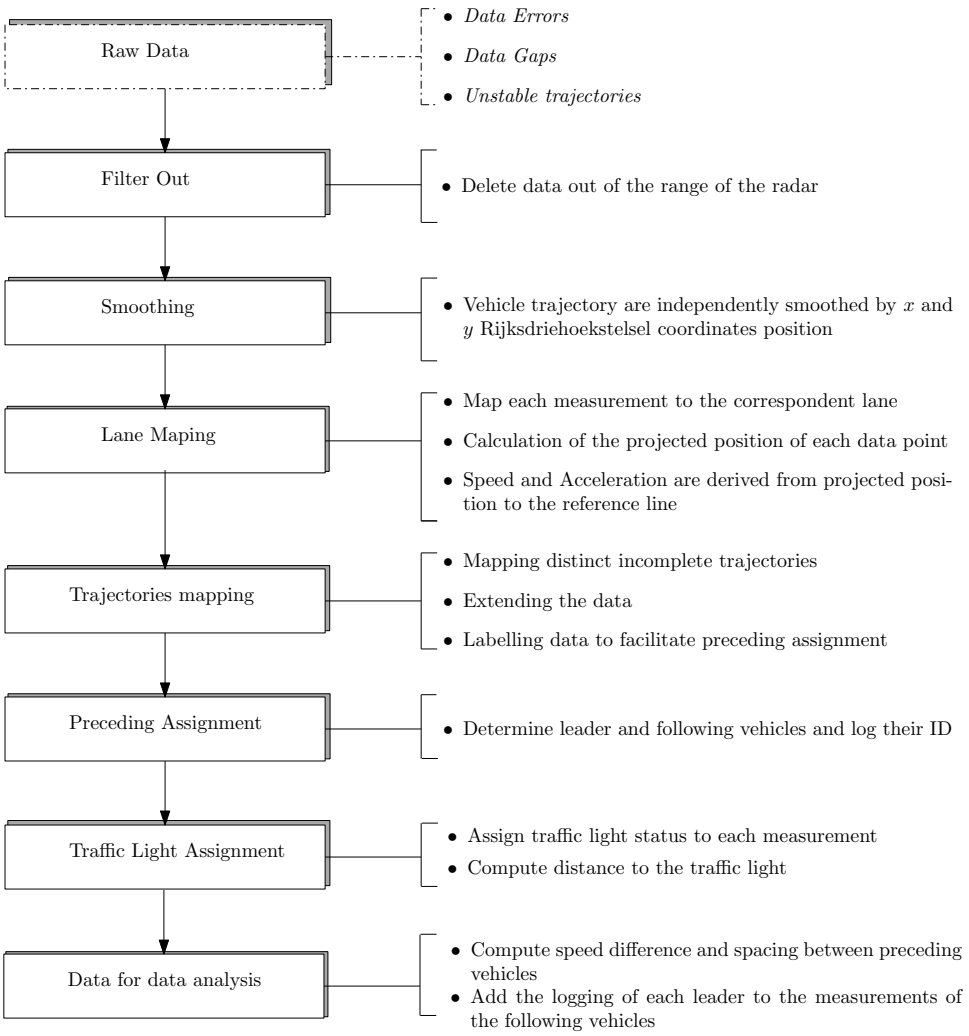


Figure 3.10: Flowchart of the Data Processing

Trajectories of each vehicle have been independently smoothed per x and y position. To do so, it has been used the well-known "moving average method", which locally replace each data point with the average of the neighboring data points defined within the span (see Equation **3.1**). According to the equation, $y_s(i)$ (in our specific case either x or y in Rijksdriehoekstelsel coordinates) is the smoothed value for the ith data point. N is the number of neighboring data points on either side of $y_s(i)$, and $2N+1$ is the span. For instance, a span of 7, would mean that the resulted smoothed data point is the average of the 6 nearest points (three on each side) and itself (centre point). Note that the span needs to be an odd number. In the borders points (i.e. $i = 3$) the span is adjusted to accommodate the the maximum neighbour data points available until the spam is entirely achieved. More information about this method and how it is applied in Matlab can be found in (Mathworks, n.d.-c).

$$y_s(s) = \frac{1}{2N+1}\left(y(i+N) + y(i+N-1) + ... + y(i-N)\right) \tag{3.1}$$

Figure 3.11 shows an example of a single smoothed trajectories with three different spans: 5 points (1.14 seconds), 11 (2.5 seconds) and 21 points (4.78 seconds). The red trajectory illustrates the smoothed trajectory, while the blue depicts the observed trajectory. As can

be observed, higher spans lead to extra straight trajectory. However, this also means that we are deviating too much from a possible reality.

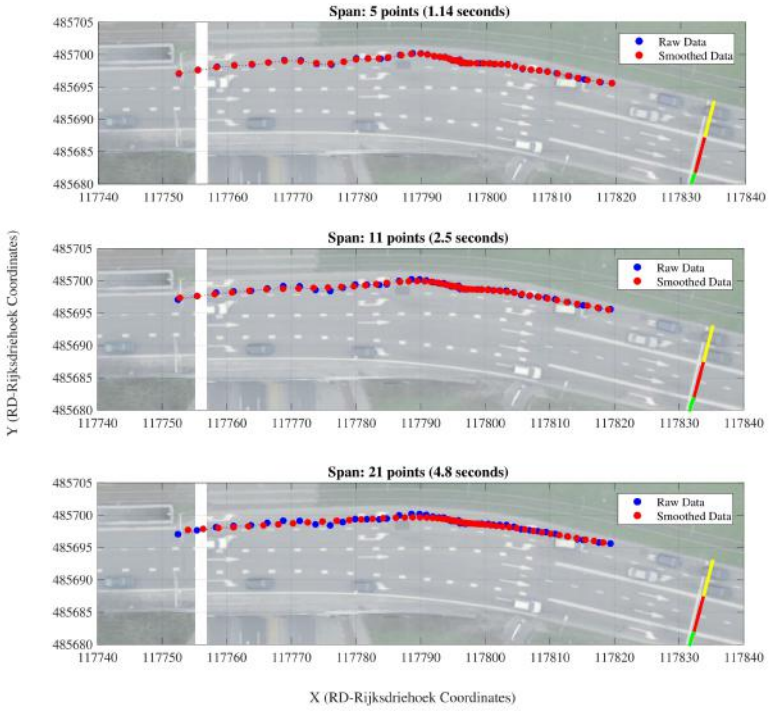


Figure 3.11: Example of vehicle trajectory independently smoothed by x and y Rijksdriehoekstelsel coordinates

In order to choose the appropriate span level, it is also important to take a close look to how smoothing span effects to the projected position. **Figure 3.12** depicts the projected position after applying the same set of spans. It can be seen that a span of 21 points (4.78 seconds) might be missing driving behaviour (data deviates to much from the raw data). From the previous figure we saw that a span of 5 data points did not improve enough the noisy trajectory. Therefore, it is considered that a span of 11 points is the most suitable span for this data set. This span represents a smoothing of 2.5 seconds according to the radar time interval.

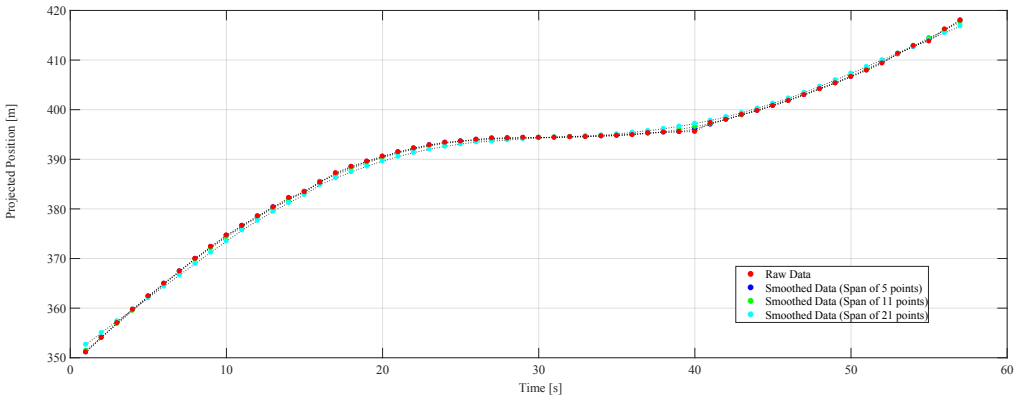


Figure 3.12: Example of vehicle trajectory independently smoothed by x and y Rijksdriehoekstelsel coordinates and their resultant projected position to the reference line.

The accuracy of the position data has also great impact on the speed and acceleration.

Figure 3.13 depicts speed and acceleration derived from different data of a single trajectory. In this case, the vehicle reduces its speed (braking) as the traffic light is red. After being stopped for approximately 1 minute, the vehicle accelerates to speed up again. Speed can be derived directly from the radar (red), from x, y raw data (green), from x, y smoothed data (blue), from projected raw data (yellow) and from projected smoothed data (cyan). Afterwards, acceleration can be directly derived from speed. On one hand, deriving speed and acceleration from raw position data (either x,y or projected position) leads to noisy results (green and yellow). On the other hand, deriving speed and from smoothing data (span of 11 points) leads to more plausible results (blue and cyan). Both speed and acceleration results from x and y coordinates and from projected position are quite similar, especially speed results. Moreover, those estimations are similar compared to speed directly measured by the radar (red). Therefore, in order to avoid using variables from different measurement origin, speed and acceleration will be computed from the smoothed projected position instead of directly from the radar.

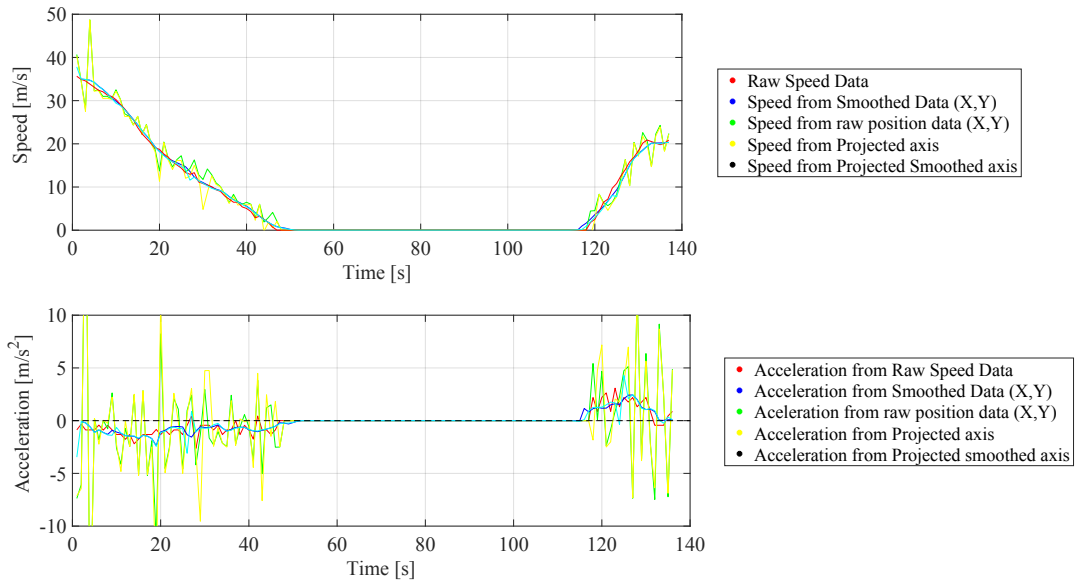


Figure 3.13: The effects of smoothing into speed and acceleration

3.3.3. Lane Mapping

Car following models focus on individual lanes, so it is essential to map each measurement to the correspondent lane. Lane mapping not only will help to filter vehicles per lane, but also will help to find out vehicles that change lane. The main idea of this mapping is to know if a data point (x, y) is inside or not a certain poligon area (lane). This can be done quite straightforward in Matlab using predefined functions such as *inpolygon*. In order to get the area (coordinates) of each lane, a Google Earth image is imported to Matlab. Generally speaking, Google Earth has an inaccuracy of 1 metre approximately, so an area can not be directly drawn it this platform to get the coordinates. Alternatively, the image needs to be imported to Matlab and displayed into a figure together with the data points. Then, it is needed a manual synchronisation process by mapping the image to the points. Finally, the coordinates of the polygon of each lane can be derived from the figure by using predefined Matlab function *ginput*. **Figure 3.14** depicts an example of lane mapping of radar IV for 1 hour data measurements. Each measurement is labelled according to the lane and it is shown in the figure with

a different colour. Note that the radar is also capable to detect trams (orange).

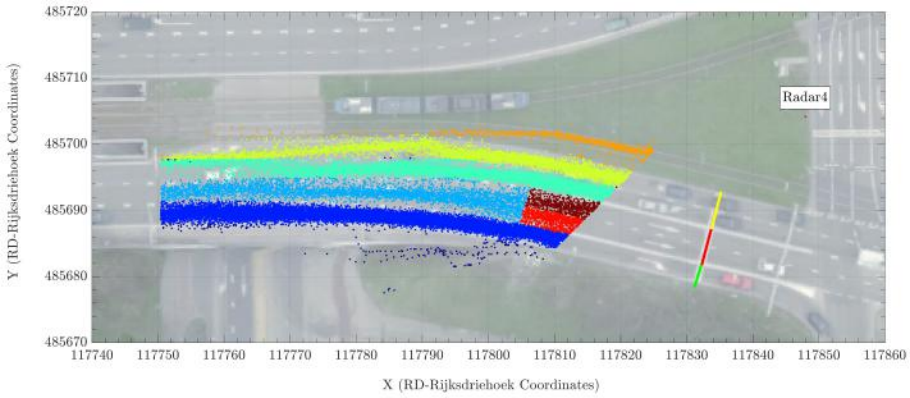


Figure 3.14: Lane maping at radar IV. Data from 17/06/2016 from 9AM to 10AM.

In this process, the projected position to the reference line of each measurement is also computed. This variable become essential in later steps of the data analysis processing. It is mainly used as a criteria to list cars within a lane, to create time-space diagrams and to improve algorithm performance (i.e. breaks in loops). Moreover, the projected position is used to compute speed and acceleration of each measurement as indicated in the previous subsection.

3.3.4. Mapping and Extending Trajectories

One of the most problematic issue of the raw data are the data gaps. Trajectories are incomplete, which could lead to an incorrect preceding assignment. For this reason, mapping trajectories belonging to the same vehicles and extending those trajectories still incomplete becomes essential.

The first step in this process is to map distinct trajectories that belong to the same vehicle but they are assigned to different vehicles ID. This mainly occurs when radar misses some measurements due to occlusion or inferences and later on it detects again the same vehicle. However, the internal radar algorithm is not capable to match both trajectories and they are automatically assigned to different vehicles (different ID). This problem is partly solved by performing a linear assignment problem in the time-space diagram environment. The general idea of the linear assignment problem is to find to each incomplete trajectory all the other incomplete trajectory candidates which might belong to the same vehicle. Then, it is assigned a cost/weight in every possible combination of trajectories and finally the assignment is performed. The constraints of the assignment to select candidates to one incomplete trajectory are:

i *Position constraint*: the position of the first measurement of a possible candidate is greater than the last position of studied trajectory

ii *Temporal constraint*: the time of the first measurement of a possible candidate is greater compared to the time of the last measurement of the studied trajectory. Moreover, both trajectories cannot belong to a difference of time greater than 3 minutes to increase algorithm efficiency

iii *Lane constraints*: the lane difference between the first measurement of a possible candidate to the last measurement of the studied trajectory is 1 (only a lane change between consecutive lanes is allowed)

To each candidate trajectory a weight is assigned according to:

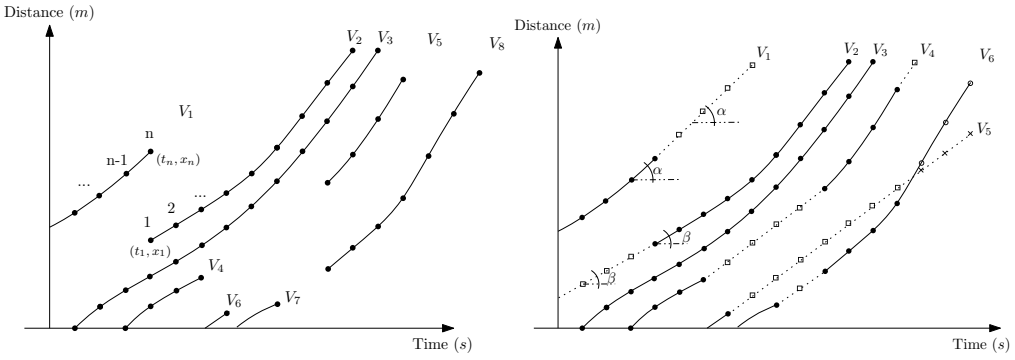
i Euclidean distance over time and space between candidate and studied last point of the studied trajectory

ii The speed difference between the first measurement of the candidate and last measurement of the studied trajectory

iii If there is or not a lane change

Once the weights of all possible combinations between incomplete trajectories candidates are known, the Hungarian method is applied to solve the linear assignment problem. This type of assignment basically minimises the global cost of all candidates and gives a unique solution. Note that it is not necessary to have the same number studied trajectories than candidates, as the method might not assign any candidate to one trajectory. More information of this assignment method can be found in (Kuhn, 1955).

This linear assignment problem applied to trajectories is illustrated as an example in **Figure 3.15a**. First, trajectories V_1, V_4, V_6 and V_7 are selected as candidates because their last or first measurement is smaller or greater than initial or end distance determined by the radar range (incomplete trajectories). For each of these trajectories, candidates are searched and a weight (or cost) are assigned to each possible combination that satisfy the constraints. For instance, V_1 do not have any possible candidates, as its last distance measurement is greater than any other first measurement of all trajectory (position constraint: $V_{1x_n} > V_{2x_1}$). Thus, all weight/cost of the candidates of this trajectory are assigned to infinite. V_4 might have two candidates: V_5 and V_8. Both candidates fulfil the constraints, so a weight is given to both candidates. Finally, a matrix containing all combination of candidates is obtained and the assignment method is applied. The ideal result would be that V_4 is assigned to V_5 and V_7 to V_8, while V_1, V_2 and V_6 remain incomplete. For those trajectories which are mapped, a linear interpolation between both last and first measurement is done to get complete trajectories. The interpolated points are labelled as not reliable and the ID of the data points of the second trajectory are changed to the ID of the first one.



(a) Sketch: Original data. *(b)* Sketch: Solution to the linear assignment problem.

Figure 3.15: Sketch: Linear Assignment Problem

The second step is to extend the missing incomplete trajectories either at the starting or end distance. Following the same example, V_1, V_2 and V_6 are linearly extended using the last or the first measurement speed. For instance, V_1 is linearly extended until the end distance

with slope α (speed between two last measurements). If the vehicle is standstill in their last or first measurement (slope equal to zero), then a constant speed of 1 m/s is applied. This is done to avoid horizontal trajectories over time. Finally, in order to have complete trajectories, a simple interpolation is carried out. Again, all interpolated points are labelled as not reliable (estimation).

This procedure will successfully lead to a better preceding assignment as no data gaps are found. However it is not a perfect method, especially when only few data points of a vehicle trajectory are tracked. A measurement can generally be used if the measurement itself and its predecessor measurement is reliable. However, due to errors in the linear assignment problem, estimated trajectories may occasionally overlap others in the same lane (see V_5 and V_6 in **Figure 3.15b**). This result is not feasible as this might indicate a collision between vehicles. The assignment itself is not a problem, as those data points from V_5 are labelled as not reliable and therefore are not going to be used. However, this wrong assignment might lead to an error to other reliable points of the second vehicle. For instance, data points of vehicle V_6 after the overlapping, should not be used as there is a car upstream which the assignment do not show. Therefore, those data points should be labelled as not reliable. Nonetheless, other cars downstream of V_6 could use those points as leader.

To deal with all the before mentioned cases, the labelling in **Table 3.4** is suggested and a real example is illustrated in **Figure 3.16**. Labelling 0 (red colour) and label 1 (blue colour) is the labelling inherit from previous process steps and corresponds to the original label given by the radar owners. This labelling simply indicates whether the data is reliable or it is an estimation of the algorithm of the radar itself. Label 2 (magenta) indicates whether the measurement is an estimation done by the linear assignment problem. It could be either estimation done to extend the data or mappings between two trajectories. Label 3 (cyan) and label 4 (yellow) are labels to solve the issue with wrong estimations in the assignment problem. Label 3 depicts all those measurements where the measurement itself is reliable, but it cannot be used as a preceding vehicle because upstream there should be a vehicle which eventually has not been well estimated by the assignment. However, this point could be used itself as reliable measurement if a reliable following data point is found, i.e. can be used as a leader. Finally, label 4 (yellow) is set to all data estimations (not reliable data points) which after overlapping need to be considered as transparent points. This means that the points are not considered as preceding and automatically the search for a preceding needs to jump to next upstream point.

Table 3.4: Data labelling according to its reliability

Label	Meaning	Description
0	Not Reliable	Radar estimation
1	Reliable	Real measurement
2	Not Reliable*	Estimation: mapping between two trajectories or extension
3	Reliable*	Real Measurement but cannot be used as leader vehicle
4	Not Reliable**	Bad estimation, point converted to transparent

Table 3.4 depicts the percentage of data of each label in a whole day dataset from the 7^{th} of June of 2016. On one hand, if we take a look to the original data provided, nearly the majority of the 928.859 measurements are reliable. Only 1% of this data is labelled 0 , i.e. radar own estimations. On the other hand, after processing the data, the dataset size increases to 1.241.865 measurements as new estimations are added to facilitate next data process steps. Therefore the reliable measurements represent 74% of the total data size. This labelling process is essential to label all measurements in order to know whether we can use or not each

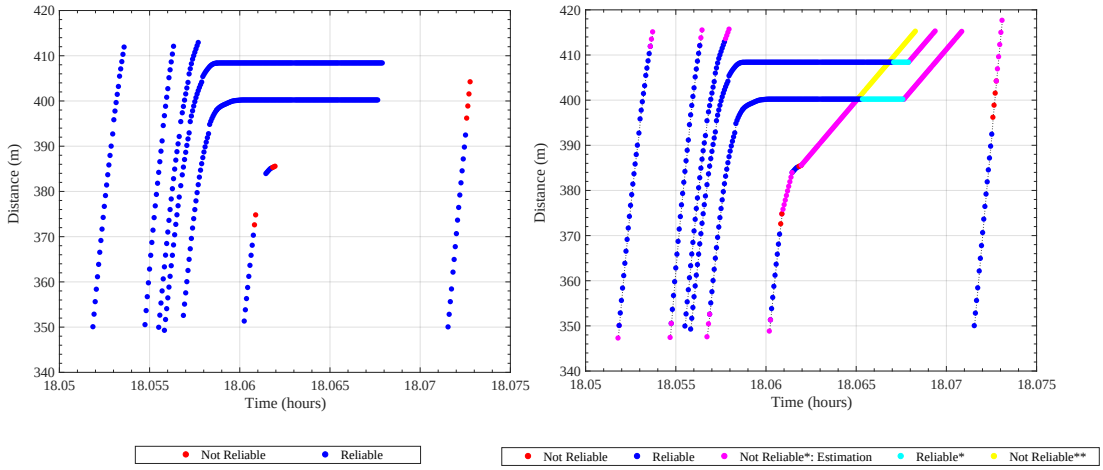


Figure 3.16: Real example of the linear assignment problem. The left graph is original data at this procedure. Right graph is the solution of the linear assignment problem and its labelling.

logging. Hence, this procedure will avoid selecting wrong information and will lead to get better results in the data analysis.

Table 3.5: Statistics of data reliability. Percentage of data from 7^{th} of June of 2016 of each label. Left column refers to the original data (928.859 measurements) while the right column refers to the processed data (1.241.865 measurements).

Label	**Percentage**	
0	1,1%	0,8%
1	98,9%	74,0%
2	-	18,3%
3	-	4,9%
4	-	2,0%

3.3.5. Preceding Assignment

Most of car-following models determine the acceleration of a vehicle based on their own speed and the spacing and speed difference with its preceding vehicle. Therefore, it becomes essential to assign to each data point the preceding vehicle. This step is simply determining which vehicles are leaders and which vehicles are following another one in every time instant. The ID of the preceding vehicle at each time instant is included in each measurement. If no preceding vehicle can be found (ie. leader vehicle), ID is set to "NaN" value. The labelling from previous section helps to determine whether the preceding vehicle is a real vehicle or it was an estimation error. The ID assignment is also done for several previous time step. Each measurement includes the vehicle ID of its preceding from the previous logs (i.e. 1^{st} log/0.22s,..., 4^{th} log/0.44s). This is done to see whether acceleration in time step n varies per log taken into account $(n-1,...,n-4)$ in the predictor variables such as speed difference, the so-called reaction time.

Figure 3.17 depicts the preceding assignment using the same example as previous sub-

section. Each reliable data point is checked whether it can be used. If it can be used, it is sub-classified as following vehicle (red) or as leader vehicle (blue). The condition for being an usable measurements is that the following and its leader vehicle measurements need to be both reliable (label 1). Furthermore, it can also be used combinations of reliable leaders (label 1) with following drivers with label 3 or that both the leader and the following are labelled as 3, i.e. both trajectories are overlapped by the same error measurement. Transparent points are skipped and jumped in the search of preceding vehicle as seen in the figure (yellow points).

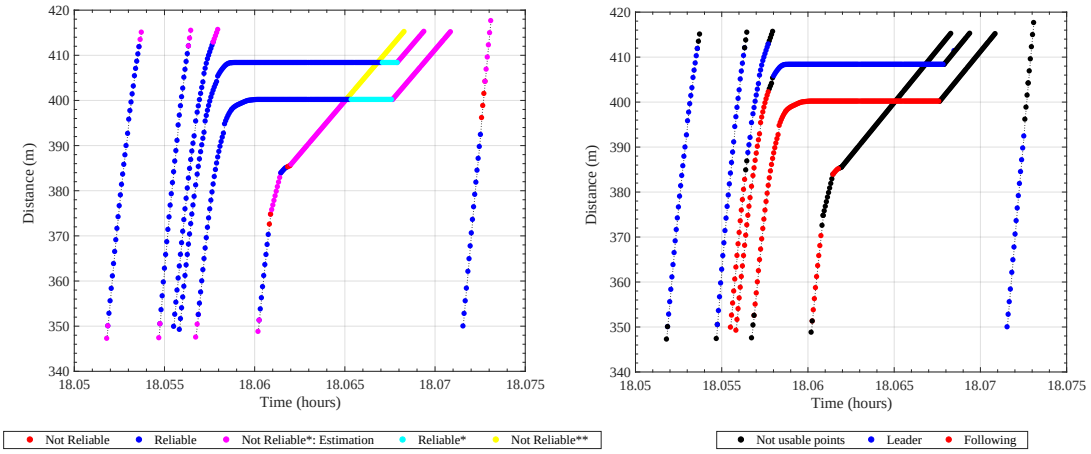


Figure 3.17: Preceding Assignment with real data. the left graph is original data at this procedure. Right graph is the solution of the preceding assignment problem and its labelling.

Using the same example as in the previous subsection, from the 918.640 original reliable measurements, 52% are labelled as "following" measurements and 46% as "leaders". The 2% missing are reliable data points that cannot be used due to a wrong estimation of the added data points, i.e. combinations of following measures with label 1 and leader measures of label 3. A quick check in the assignment shows the effectiveness of the preceding assignment approach. If we had not used this approach, there would have been 95.304 wrong assignments (10.3% of reliable points). This can be easily checked by counting all those following vehicles that its leader is an estimated measurement (label 2). Overall, this proves that the data processing approach suggested in this thesis is a success. Despite losing 2% of reliable data due to wrong estimation, it is avoided that 10% of all reliable points present a wrong preceding assignment.

During the preceding assignment, the number of cars downstream is also computed. In this assignment, for each data point, vehicles downstream are searched and sorted. Hence, it is a great opportunity to determine the number of cars downstream until the traffic light. Unfortunately, we might be missing some cars downstream as the range of the radar misses 15 metres approximately in front of the traffic light.

3.3.6. Traffic Light Assignment

One of the main hypotheses in this project is that traffic lights state might influence drivers' longitudinal behaviour. Thus, it is essential to assign this information to each vehicle log. As mentioned in section section 3.2.2, there exist three independent traffic light on this stream

(left turning, straight and right turning). Hence, it is essential to know beforehand the mapped lane of the vehicle in order to assign the right traffic light status. **Figure 3.18** depicts the traffic light status assignment. For instance, vehicles driving on those streams where traffic light status is red, they are assigned a "Red" label. Note that the traffic light is assigned according to the assigned lane in the last reliable measurement. That means that if a vehicle changes lane, the traffic light belonging to the lane of its last measurement will be assigned (the traffic light that it is actually taking). Lastly, in this step it is also computed the net distance to the stopping line of the traffic light using the reference line (projected position).

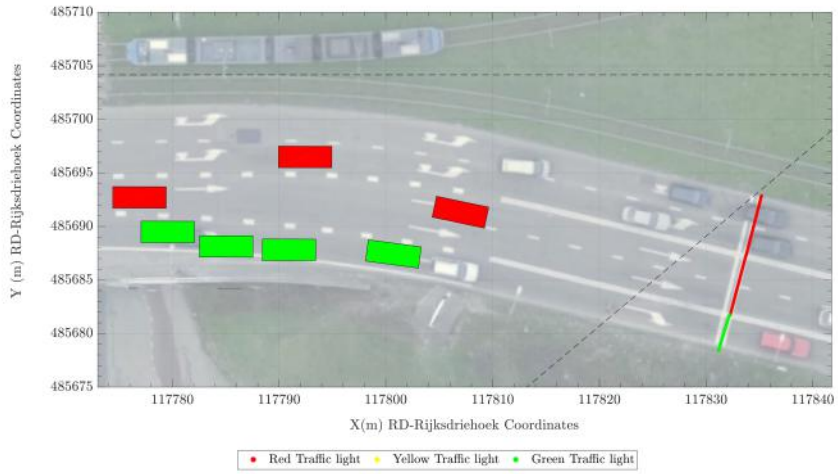


Figure 3.18: Traffic light Assignment. Each vehicle is assigned a traffic light state at each log.

3.3.7. Last Processing Steps and Data Obtained

Response and predictor variables are described in Section 3.3. Predictor variables that still need to be computed such as speed difference and spacing between the following car and its leader are computed in this process step. Moreover, it is recommended to include in the response variable measurement (acceleration), all predictor variables measurements from different time steps in order to facilitate the data analysis. For instance, for each acceleration measurement of time step n of vehicle i, it is added the predictor variables at time steps $n-1,...,n-k$ of vehicle i. Later on, in the data analysis it will be analysed which time step(s) should be considered to better describe longitudinal driving behaviour. **Table 3.6** depicts all information attached to each acceleration measurement (response variable) per time step. Finally, data is ready now to be analysed and to be used.

3.4. Data Analysis

A first simple analysis of processed data analysis can be easily carried out by creating a matrix of scatter plots where each figure contains a scatter plot of one variable against another variable. **Figure 3.19** depicts direct relationships figures between acceleration, speed, speed difference and spacing with the preceding vehicle and distance to the traffic light. Each data point is coloured according to the status of the traffic light -green, red and yellow-. Moreover, in the diagonal we can see the histogram bars of the accumulated function of each variable.

Table 3.6: Information included to each measurement of acceleration (response variable)

Variable	**Explanation**
Log Difference	Log difference between the response and the predictors variables
Preceding Label	Label indicating if the measurement can be used or not
Time	Time of the predictor variables
Speed	Speed of vehicle i
Preceding ID	ID of vehicle $i-1$
Spacing Difference	Net distance between vehicle i and $i-1$ (m)
Speed Difference	Absolute speed difference between i and $i-1$ (m/s)
Distance to traffic light	Net distance to the stopping line of the traffic light (m)
N° of cars downstream	Number of vehicles downstream of vehicle i

The measurements belong to 2 hours data set (from 8:00 AM to 10:00AM) of the 17^{th} of June of 2016. For this specific plot, only data belonging to the left lane and data belonging to following vehicles is used. The first row of figures might indicate that there is a clear relationship between response variables such as speed, spacing and speed difference and, the response variable, the acceleration. Moreover the acceleration is usually higher in absolute terms in green phases compared to deceleration in red traffic light phases -vehicle standstill accelerate whenever it gets green while they decelerate whenever the traffic light gets red-. The figure also depict some interesting relationships between predictor variables. For instance, the distance to the traffic light - speed plane depicts that vehicle reduce their speed as they get close to a red traffic light, while the behaviour is the opposite for green phases. In that case, vehicles increase their speed while they get close to the traffic light -vehicles accelerate gaining speed after being standstill due to a red traffic light or because they might want to cross the traffic light before getting red-. Another interesting example is the speed difference-spacing plane. The figure depicts the phenomena described by the Wiedemann principle, which is revised in (Hoogendoorn et al., 2011). The principle states that drivers only react if the required action is above threshold and that generally following drivers overreacts. While drivers get close to leaders they reduce their speed. At some point, the following vehicle n realise that its spacing with the leader $n-1$ is increasing as its speed was reduced too much. This leads to a positive speed difference ($V_{n-1} > V_n$). Consequently, the following driver will accelerate turning to a negative speed difference ($V_{n-1} < V_n$). Then, the following driver will again realise that the spacing with the leading vehicle is less than the safety braking distance and it will slightly brake starting again the cycle. The process is repeated and repeated until the equilibrium. The overall behaviour leads to the circles in the speed difference - spacing plane, which can be easily observed in the figure. Overall, the figure proves that the previous steps in the data processing have been a success as figures and measurements do not generally present a strange behaviour. Nonetheless, it is also worthy to mention that obviously sometimes strange measurements can be observed. For instance, there are few low spacing measurements registered in the processed data, between 0 and 1, with high speed difference or speeds. This might indicate that the spacing, which is derived from position measurement, is not an completely accurate measurement due to radar inaccuracy.

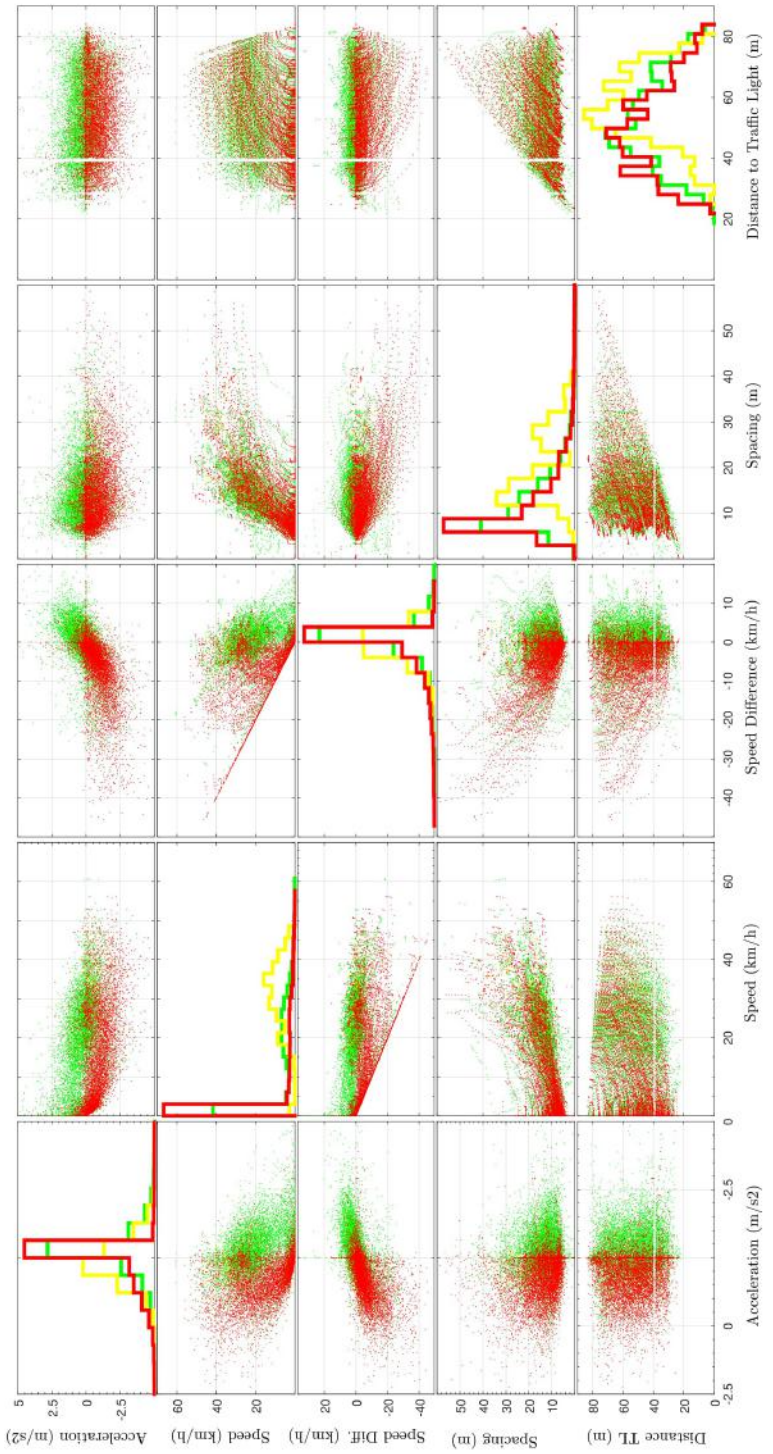


Figure 3.19: Variable Relationships

3.4.1. Discussion

The previous figure proves the uniqueness of this data set and it encourages us to look for new and innovative techniques to fit a new model. Taking a look to the available processed data, for a single day (from 6AM to 20PM), 250.000 reliable points approximately can be used for the analysis for each single lane (varies according to the flow and traffic light cycle time of each lane). That means that for 2 months data, tens of millions of data measurements could be used. Furthermore, data only belongs to a limited road section of less than 100 metres and inevitably the process data still have errors. Traditional techniques usually assume a underlying parametric model, and by means of optimisation techniques they try to find the optimal parameters that fit the dataset. Hence, we could opt to enumerate a parametric car-following model based on observed mathematical relationships between variable, or use an existent parametric model and slightly modify it to adapt it to our variables. Nonetheless, both alternatives could fall in optimisation computational times problems if all data would be used. Moreover, they could lead to an overffiting issue as the dataset belongs to a relatively small spot. Finally, we might be assuming variables parametric relationships which do not really let 'talk' the data.

According to the last paragraph, it sincerely seems a lost opportunity to adopt just traditional techniques. Thus, it might be useful and challenging to try new techniques able to deal and to take advantage from big amounts of data such as machine learning techniques. Furthermore, it is necessary to explore how this possible method can deal with space regions outside our dataset. Finally, it is interesting to see whether new methods can deal with some inconsistencies and noise found in the processes data.

II

Machine Learning

4

Machine Learning Techniques

Chapter 4 aims to give insights to the reader on the so-called Machine Learning (ML) techniques, and especially on the chosen family model to describe drivers' acceleration: the Gaussian regression process. Section 4.1 introduces the topic of Machine learning. That means, explaining the goals of ML, which techniques families do exist and for what they are currently used in the scientific field. Later on, Section 4.2 focuses on Gaussian regression processes (GPR) for machine learning as it is the selected approach to satisfactorily model driver behaviour at signalised intersection. This section motivates the choice of GPR model among others, it includes and extensive formulation of the mathematical model and finally it ends with several practical and visual examples of what the model is able to do.

4.1. Introduction to Machine Learning

Machine learning (ML) is a computational technique that trains models to learn from experience. Machine learning algorithms use computational methods to "learn" information directly from data relying on a generic model formulation. ML models are able to find natural patterns in data that generate insight and help to make more accurate predictions. One of the main benefits of these techniques is that, ML adaptively improve their performance as the number of samples available for learning increase (Mathworks, 2016). Over last decades, machine learning models have been applied to solve many daily problems like, among others, spoken language recognition, fraud detection, customer relationship management or gene function prediction (Ławrynowicz & Tresp, 2014).

There exist two types of ML techniques: unsupervised and supervised learning (see **Figure 4.1**). On one hand, unsupervised learning aims to find hidden patterns or intrinsic structures in input data. It becomes useful when you want to explore your data but you do not yet have a specific goal or you are not sure what information the data contains. It might be also a good technique to reduce the dimensions of your data (Mathworks, 2016). On the other hand, supervised learning focuses on deriving predictive models based on input -predictor(s)- and output -response- from the so-called training set data. Thus, this family of techniques learns from past observations to predict the future. Supervised learning is divided in two main techniques depending on the target. If the target of the model is discrete (e. g. nominal or ordi-

nal), then the given task is called classification. Classification models are trained to classify data into categories such as for example, whether an email is genuine or spam. If the target is continuous, the task is called regression (Ławrynowicz & Tresp, 2014). Some examples are changes predictions, temperature or fluctuations in electricity demand or forecasting stock prices.

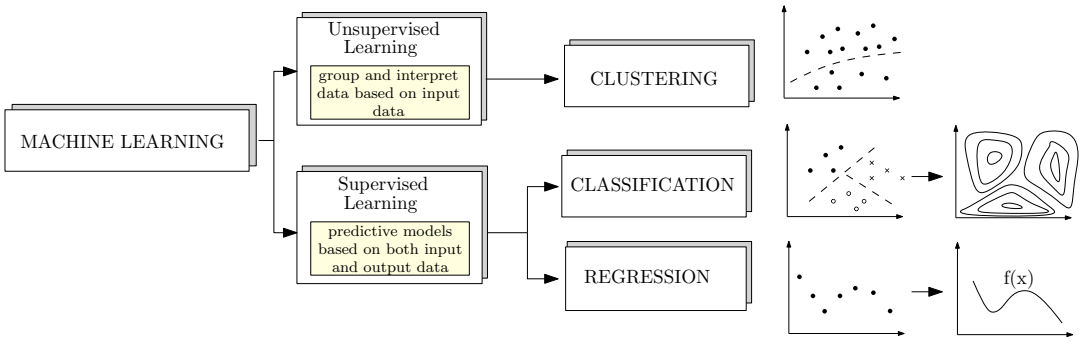


Figure 4.1: Machine Learning techniques. Adapted from (Mathworks, 2016)

As the goal of the thesis is to derive a predictive model able to simulate driver's acceleration, supervised learning with regression techniques will be used. Given a training set of examples of h, the goal is to return a function f that best approximates h. In our specific case, given the acceleration of vehicles and a set of predictors such as speed and spacing, the goal of the machine learning regression process is to learn a function able to make accurate predictions of the acceleration given a new data set of predictors variables.

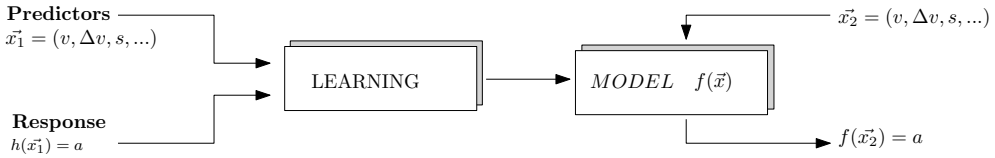


Figure 4.2: How machine learning works. Adapted from (Mathworks, 2016)

4.1.1. Common Regression Algorithms (Supervised Learning)

There exist several types of regression algorithms and these are listed as follows (Mathworks, 2016):

i **Linear and non linear Regression:** These are statistical modelling techniques that are used to describe a continuous response variable as a linear/non linear function of one or more predictor variables.

ii **Gaussian Process regression (GPR):** Non parametric models that are used for predicting the value of a continuous response variables. It can also be defined as a collection of any random variables, any Gaussian process finite number of which have a joint Gaussian distribution (Rasmussen & Williams, 2006). They are widely used in the field of spatial analysis for interpolation in the presence of uncertainty.

iii **Support Vector Machines (SVM) Regression:** This technique is based on regression algorithms that find a model that deviates from the measured data by a value no

greater than a small amount, with parameter values that are as small as possible in order to minimise sensitivity to the error.

iv **Decisions Trees:** It lets you predict responses to data by following the decisions in the tree from the root (beginning) down to a leaf node. A tree consists of branching conditions where the value of a predictor is compared to a trained weight.

v **Neural Networks:** It consists of highly connected networks of neurons that relate the inputs to the desired outputs. The network is trained by iteratively modifying the strengths of the connections so that given inputs map to the correct response.

4.1.2. Machine Learning in Traffic Theory

Since the end of last century with the rapid improvement of personal computer, machine learning technique was rapidly adopted by experts and researches in the transportation field. Research have mainly focus on macroscopic traffic theory topics and traffic management data. For instance, (Lv et al., 2015) enumerates a deep learning approach to predict traffic flows, and states that this technique has superior performance than traditional techniques using shallow traffic prediction models. (Hofleitner et al., 2012) proposes an hybrid approach of traditional flow modelling techniques and machine learning to forecast urban travel time with streaming GPS probed data. Results show that combining both approaches results on a significant improvement compared to a data-driven baseline algorithm. Moreover they point out that a traditional flow model of traffic is essential to ensure consistent results according to physics of traffic. Lately, numerous studies can be found in literature focusing on automated driving, especially in the sign recognition and navigation range such as (Pomerleau, 1991) and (De la Escalera et al., 2003).

Moving to microscopic traffic modelling and especially with similar aim than this master thesis, little literature can be found. Artificial neural network is the only ML family used in practice, i.e. none literature is found for GPR or SVM. (Panwai & Dia, 2005b) elaborates a car following model using reactive agent techniques based on a neural network approach for mapping perceptions to actions. However the model is not completely specified and it mainly aims to classify five drivers modes (e.g. free driving, following , danger...) with ANN according to speed difference and spacing inputs, to later on apply certain response rules according to the driver mode. Recently, (Khodayari et al., 2012) described a complete ANN model given 4 inputs -spacing, speed difference, speed and reaction time-, 1 output -acceleration- and only one hidden layer. Unlike other models, the reaction time was not considered fixed and linearly depended on the spacing and the current acceleration of the driver. In this case, the model was trained using a large data trajectory set from a 640 metres road section in a highway located in San Francisco (California,US). The results were outstanding, with RMSE around 0.25 (m/s^2), proving that this technique is able to incorporate hidden driver behaviour, specially in the reaction time. The main reason to find few literature in this topic is certainly the lack of accurate large data set, which would benefit from the characteristics of ML techniques. In the past, it have rarely been large datasets of consecutive vehicle trajectories. Traditional parametric models techniques filled the requirements as few trajectory data points have been usually used to enumerate and calibrate longitudinal driver behaviour models. As in other major engineering fields, with the improvement of traffic data collection techniques, it will definitely start to be become popular topic for researcher in the near future.

4.2. Gaussian Process for Machine Learning

This section contains all information regarding Gaussian process regression models for machine learning. First, the choice of this model among others is justified. Then, formulation regarding this approach is enumerated. Note that Gaussian process can also be applied for classifications purposes. However, in this thesis, we only refer Gaussian Process for regression, i.e. continuous responses. Finally, visual and practical examples are provided to the reader to facilitate the understanding of what the selected model is able to do.

4.2.1. Motivation

Gaussian process regression (GPR) models, also known as Kriging regression models, are non-parametric kernel-based probabilistic models (Mathworks, n.d.-d). Opposite to parametric models, where predictions by the model are done by assuming an underlying distributions and fitted parameters, non-parametric models relies directly on the data, which tells to the model how to make predictions. The basic idea of GPR is that by training the model with a large data set, the model is able to compute a predictive mean and its variance. When a new input is introduced for prediction, the model checks how 'close' is the new input to the training set, and based on that it makes a prediction. There are two main reasons why GPR might help us in our aim of modelling drivers' behaviour at urban signalised intersection. First, our data is not complete enough to create a complete traffic model. For instance, our data was collected in less than 80 metres road length, meaning that high speed, high spacing and high absolute speed difference between vehicles will not be found in the training set. Furthermore, small spacing between drivers are rarely found as no collisions are presumably observed. Hence, the model might not know what to 'predict' in this kind of situation. If we aim to have a complete model, any ML non-parametric model may fall into over-fitting the current data set and even might violate some traffic principles outside the data boundaries (e.g. collisions). However, we still want to benefit from non-parametric model characteristics of relying in the data rather than in an underlying model. GPR offers a combination between both types of model. The GPR can rely on the data if new points are not far apart of the training set, and it can be modified to rely on a parametric model (basis function) when no correlation between new input and training data exists. The only assumption is that the prediction of a GPR given a new input, is a Gaussian distribution with a predicted mean and a variance of the input. The radar data used in this master thesis contains noise, which unfortunately is unknown. GPR models estimates the noise based on the training set, and predicts the variance. That means that the GPR model computes how certain the prediction is. From literature, ANN have already been used to enumerate car following model, while GPR does not. Nonetheless, one of our main purpose, as mentioned before, is to have a complete model, even outside space data set regions. ANN do not present any hybrid mathematical solution to internally predict uncorrelated input to the trained set. Moreover, ANN models might be difficult to interpreter depending on the number and complexity of the hidden layers. Similarly, decisions tree models might also be too complex. In order to predict continuous responses (regression), decision trees define really small decisions regions, resulting in hundreds of decision layers. Overall, GPR seems the most feasible and simple way to describe microscopic longitudinal driver behaviour given our data set and our purpose.

4.2.2. Formulation

A Gaussian process is a type of continuous stochastic process which defines a probability distribution for functions (Papoulis & Pillai, 2002). Consider a training set: $\{(x_i, y_i); \quad i = 1,2,...,n\}$, where $\vec{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, drawn from an unknown distribution. A Gaussian process regression model addresses the question of predicting the value of a response variable y_*, given a new input vector x_* and the training data set. The key point is that a Gaussian process is defined as a distribution over functions:

$$f \sim \mathcal{GP}(m, K) \tag{4.1}$$

Now suppose that we pick a particular finite subset of set of random variables indexed by a continuous variable: $f(x)$, $f = \{f_1, f_2, ..., f_n\}$, with indices x_i. In a GP, any such set of random function variables are distributed multivariate Gaussian (Snelson, 2008).

$$P(f \mid X) \sim \mathcal{N}(u, K) \tag{4.2}$$

where $m : \mathcal{X} \to \mathbb{R}$ is the **mean function**

$$m(x) = \mathbb{E}[f(x)] \tag{4.3}$$

and $K : \mathcal{X}^2 \to \mathbb{R}$ is the **co-variance function** of a real process $f(x)$:

$$K(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \tag{4.4}$$

Thus, a Gaussian process is completely specified by its mean function and co-variance. The last one can be defined by various kernel functions. It is usually parameterized in terms of the kernel parameters in a vector $\theta = (\log(\sigma_l), \log(\sigma_f))$, where θ_l is the characteristic length scale, and θf is the signal standard deviation.

$$k(X, X) = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix} \tag{4.5}$$

Where K_{ij} can be for example the squared exponential kernel between others:

$$K_{ij} = (x_i, x_i) = \sigma_f^2 \exp\left[-\frac{1}{2}\frac{(x_i - x_j)^T (x_i - x_j)}{\sigma_l^2}\right] \tag{4.6}$$

Kernel parameters in the kernel function (σ_l, σ_f) essentially describe how far apart the input values X_{ij} are from each other and, i.e. correlated between each other. If a new input x_* is provided, $K(x_*, X)$ describes how much the new input is correlated to the training set. Both σ_l and σ_f needs to be greater than 0, and this is enforced by the unconstrained logarithmic parametrization (Rasmussen & Williams, 2006).

Prediction free Noisy Observations

Now suppose that we have a (X, y) training set, and we want to predict y_* based on a new input point x_*. The joint distribution of the training outputs, y , and the test outputs y_* according to the prior is (assuming noisy free observation X and zero mean prior):

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,x_*) \\ K(x_*,X) & K(x_*,x_*) \end{bmatrix}\right) \tag{4.7}$$

Then, the predictive distribution is obtained by conditioning on the observed training outputs (and using the inverse of a partitioned matrix identity):

$$P(y_* \mid y, X, x_*) \sim \mathcal{N}(y_* \mid \mu_*, \Sigma) \tag{4.8}$$

Where, the predictive mean of the new input point is μ_* and its variance Σ_* is:

$$\mu_* = K(x_*^T, X)\left[K(X,X)\right]^{-1} y \tag{4.9}$$

$$\Sigma_* = K(x_*, x_*) - K(x_*^T, X)\left[K(X,X)\right]^{-1} K(X, x_*^T) \tag{4.10}$$

Prediction using Noisy Observations

Typically, training observation incorporates noise ($y = f(X) + \varepsilon$). Thus, assuming the training set, (X, y), with additive independent identically distributed Gaussian noise ε with variance σ_n^2, a new input point x_* and the desired y_*, the co-variance function and the joint distribution of the observed target values and the function values at the test locations under the prior are:

$$\text{cov}(y) = K(X,X) + \sigma_n^2 I, \tag{4.11}$$

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{4.12}$$

Again, the predictive distribution is obtained by conditioning on the observed training noisy outputs:

$$P(y_* \mid y, X, x_*) \sim \mathcal{N}(y_* \mid \mu_*, \sigma^2 + \Sigma_*) \tag{4.13}$$

Where, the predictive mean of the new input point is μ_* and its variance Σ_* is:

$$\mu_* = K(X_*^T, X)\left[K(X,X) + \sigma_n^2 I\right]^{-1} y \tag{4.14}$$

$$\Sigma_* = K(x_*, x_*) - K(x_*^T, X)\left[K(X,X) + \sigma_n^2 I\right]^{-1} K(X, x_*^T) \tag{4.15}$$

Incorporating Explicit Basis Functions

One of the main reasons of choosing Gaussian Process to model our data, is the theoretical facility of the GPR to model those space regions where we possess information but also those regions where few or no data points in the training set are found. In the previous subsection (with and without noise), the Gaussian process has been defined with zero mean. Hence, new data points uncorrelated with the training set, i.e. points far apart the training set, will results with a predictive mean of zero. For this thesis, it seems necessary to include a non-zero mean function to model those space regions with no data points in the training set. For instance, few data points can be found with small spacing values (e.g. smaller than 0.5 metres) or high spacing values (e.g. bigger than 50 metres). In order to posses a complete model, an underlying parametric model H with parameters β can be defined to model longitudinal drivers' behaviour outside space regions of the training data. As before, the predictive distribution is obtained by conditioning on the observed noisy training outputs:

$$P(y_* \mid y, X, x_*) \sim \mathcal{N}(y_* \mid H(x_*)\beta + \mu_*, \sigma^2 + \Sigma_*) \tag{4.16}$$

Where, the predictive mean of the new input point is μ_* and its variance Σ_* is:

$$\boldsymbol{\mu}_* = K(x_*^T, X) \underbrace{\left[K(X,X) + \sigma_n^2 I\right]^{-1} (y - H(X)\beta)}_{\alpha} \tag{4.17}$$

$$\boldsymbol{\Sigma}_* = K(x_*, x_*) - K(X_*^T, X) \left[K(X,X) + \sigma_n^2 I\right]^{-1} K(X, x_*^T) \tag{4.18}$$

Note that in this case, when x_* is uncorrelated with X, the predictive mean μ_* tends to zero as $K(x_*^T, X)$ is small. Thus, the resultant mean function of the predictive distribution is mainly the underlying mean function of the explicit basis function evaluated in the new input $H(x_*)\beta$. The model can be then seen as an hybrid GPR model which combines parametric and non-parametric formulation.

Hyper-parameters Optimisation and Practical Implementation

The previous sections have described the Gaussian process. To summarise, making predictions using a GPR mode for new data points based on a training set requires:

- Knowledge of the coefficient vector of the basis function β.
- Knowledge on the so-called kernel parameters $\theta = (\sigma_l, \sigma_f)$.
- Knowledge of the noise variance σ^2 of the training set.

Consequently, all these parameters need to be optimally estimated to make accurate predictions. Generally speaking, applying the GPR model consists in two main phases. First, it is needed to adequately estimate the hyper-parameters $(\beta, \theta, \sigma^2)$ from the training data (X, y). Second, the GPR model is build using the optimal hyper-parameter set and predictions of y_* can be performed given new data points x_*. The main approach to estimate or "learn" the hyper-parameters is by maximising the log-likelihood $P(y \mid X)$ as a function of β, θ and σ^2. This method is described in (Rasmussen & Williams, 2006) and is especially designed for GPR for machine learning in order to benefit of big amounts of data. A practical implementation of this approach is shown in **Algorithm 1**. The main idea is that in each iteration, first $\hat{\beta}(\theta, \sigma^2)$ is computed. Then, the algorithm maximises the β profiled log-likelihood over θ and σ^2 (Mathworks, n.d.-b). Overall, this approach represents a major advantage of GPs over other methods as the method is able to select co-variance hyper-parameters from the training data directly, rather than use a scheme such as cross validation (Snelson, 2008). In the maximisation, well-known optimisation solvers algorithms such as quasi-newton or interior-point algorithms can be used. The output of the training phase is a set of hyper-parameters $(\beta, \theta, \sigma^2)$ which maximises the log-likelihood.

The second step is to build and predict new data points using the trained model (see **Algorithm 2**). Similarly to the training algorithm, first the GPR model is built according to the optimal parameters found, i.e. calculation of C and α in steps 2 and 3. Later, given a new point to predict x_*, the algorithm returns its predictive mean $\mathbb{E}(y_*)$ and variance $\mathbb{V}[y_*]$.

Algorithm 1 Gaussian Processes for Machine Learning: Training

Input: X (inputs-predictors-), y (target-Response-), $(\beta_0, \theta_0, \sigma_0^2)$ (Initial hyper-parameters)

1: **Objective Function:** $\hat{\beta}, \hat{\theta}, \hat{\sigma}^2 = \arg_{\beta,\theta,\sigma} \quad \max \quad \log\{P(y|X,\beta,\theta,\sigma^2)\}$

2: **Initialisation:**

3: $C := (K(X,X|\theta) + \sigma^2 I_n)^{-1}$

4: $\hat{\beta}(\theta,\sigma^2) := [H(X)^T C H(X)]^{-1} H(X)^T C y$

5: $\log P(y|X,\hat{\beta}(\theta,\sigma^2),\theta,\sigma^2) := -\frac{1}{2}(y - H(X)\hat{\beta}(\theta,\sigma^2))^T C (y - H(X)\hat{\beta}(\theta,\sigma^2)) - \frac{n}{2}\log 2\pi - \frac{1}{2}\log C$

Output: (β,θ,σ^2) (Hyper-parameters)

Algorithm 2 Gaussian Processes for Machine Learning: Prediction

Input: (X,y) (training set), (β,θ,σ^2) (Hyper-parameters), x_* (new data points)

1: **Initialisation**

2: $C := (K(X,X|\theta) + \sigma^2 I_n)^{-1}$

3: $\alpha := C(y - H(X)\beta)$

4: $u_* := \alpha K(x_*, X|\theta)$

5: $\Sigma_* := K(x_*,x_*) - K(X_*^T, X) C K(X, x_*^T)$

6: $E(y_*|y,X,x_*,\beta,\theta,\sigma^2) := H(x_*^T)\beta + u_*$

7: $V(y_*|y,X,x_*,\beta,\theta,\sigma^2) := \Sigma_* + \sigma^2$

Output: $\mathbb{E}(y_*)$ (predictive mean), $\mathbb{V}[y_*]$ (variance)

Computational Complexity

Training and predicting using a GPR model for large data sets is usually quite expensive in terms of computational time and memory usage. On one hand, as is shown in in **Algorithm 1**, training requires the inversion of the kernel function (calculation of C). The computational complexity of this step is $\mathcal{O}(n^2)$, where n is the number of observations in the training set. Later, an evaluation of $\log P(y|X)$ scales the computation complexity to $\mathcal{O}(n^3)$. Thus, the final computational complexity of training is $\mathcal{O}(k \cdot n^3)$[1], where k is the number of iteration in the optimisation process. On the other hand, prediction requires again the calculation of C, leading to a complexity of $\mathcal{O}(n^2)$. Therefore, dealing with a large data sets such as in the machine learning field becomes an expensive operation, especially in training phases. However, several methods currently exist to reduce the computation complexity of large data sets, both in training and prediction (Snelson, 2008):

i **Subset of Data (SD):** This basic computational technique consists on simply reducing the size of the data set. By selecting a subset of data of size m, the computational complexity is reduced from $\mathcal{O}(k \cdot n^3)$ to $\mathcal{O}(k \cdot n \cdot m^3)$. This technique is especially satisfactory for large and redundant data sets, where adding extra points results in a little extra information. How a subset of data can be optimally chosen is explained later in this section.

ii **Subset of Regressors (SR):** This approximation method consists of replacing the exact kernel co-variance function K by an approximation Q to facilitate its inversion. According to (Quiñonero Candela, 2004), one of the main problems of this technique is that by approximating the co-variance function k, the optimised model might rise the predictive variances in space regions away of the subset points.

[1] Matlab maximum memory usage in 64-bit Windows XP or later and Matlab 7.5 or later is 8TB (Mathworks, n.d.-a)), meaning that assuming $k = 200$ number of iterations, the maximum number of observations possible in this operation in training is 11206

iii **Fully Independent Conditional Approximation (FIC):** This approach aims to solve predictive variance problem of SR. Again the GPR kernel function K is approximated, but in this case ensuring to mantain still a valid a Gaussian process in the approximation.

iv **Block Coordinate Descent Approximation (BCD):** This approach is only used for prediction, and it aims to approximate the value of α (see **Algorithm 2**).

Other relevant approaches can also be found in literature such as projected process (PP) or the Bayesian committee machine (BCM), with the same aim: finding an approximation of the kernel co-variance matrix. As described in (Snelson, 2008), best practices include first to reduce smartly your data set (SD), and then, if still less computational is desired, any of rest of techniques listed above can be additionally performed. Reducing the data set becomes essential for methods such as SR or FIC as they would fail in the goal of reducing computational time with large data sets -finding an approximation might take more time than directly inverting an already reduced matrix-. Then, it becomes essential how subsets are selected, the so-called active sets. Usually sophisticated schemes use various information criteria that score how much the model improves by including an extra point into the subset. Otherwise, another valid option is to directly select randomly the active set. The different techniques are briefly described as follows (more information can be found in (Herbrich et al., n.d.) and (Snelson, 2008)):

i **Random:** The simplest method is to simply choose randomly the subset of data required.

ii **Entropy:** The reduction in differential entropy of the Gaussian process latent variable having observed the corresponding response variable is used as a criterion for selection of new points into the active set. This selection technique is also known by the name informative vector machine (IVM).

iii **Sparse Greedy Matrix Approximation (SGMA):** This techniques evaluates a new point by creating an approximation to the true kernel function using an active set and checks the approximation error between the true kernel function and its approximation.

iv **Likelihood:** New points are evaluated based on the approximation to the marginal likelihood of the GPR model. The criterion for accepting a new point into the active set is the change in the log-likelihood upon adding a new point to the active set.

Choices between techniques used for data reduction and selection mainly depend on the data characteristics and your computational and memory usage requirements, both in training and predicting. Literature does not ensure that one method is better than another one. Therefore, in ML all techniques should be performed and evaluated. To summarise, for large data sets, the practical implementation of Gaussian Processes for machine learning consists of: first a subset of data is selected, then the model is trained and finally prediction can be performed.

4.2.3. Theoretical Interpretation of Gaussian Processes

In order to get insights on what does Gaussian process regression model looks like, a theoretical example and a real example are illustrated below. Imagine that we train a GPR with some data points and zero mean (see **Figure 4.3**). Then, we use the GPR trained to make prediction using new data input. Note that for a single data point input, i.e. x_*, the GPR prediction is a Gaussian/normal distribution with mean u_* and variance $\sigma^2 + \Sigma_*$. Consequently, the blue line in **Figure 4.3** represents the predictive mean function evaluated to all x axis. As can be observed in the same figure, the predicted mean tends to the data when relatively close observation are found and to zero where there is no data nearby. Furthermore, the variance is obviously smaller close to training data points. All hyper-parameters in in the GPR certainly influence the results, i.e. $f(x_*)$. The kernel co-variance parameters, the characteristic length scale σ_l and amplitude σ_f, tell how data points are correlated between them. σ_f can be interpreted as the relation of two points in the y axis while σ_l as indicates the relation over the x axis. According to **Equation** 4.6, when any of those parameter tend to be small, the resultant kernel function is also small, meaning that the GP rapidly will tend to 0 when no data points are found. The parameter indicating the noise of the data, σ^2, also plays a major role as highly affects the total variance of the GPR prediction, i.e. the width of the prediction confidence interval -dashed line-. Finally, β defines the shape of the basis function.

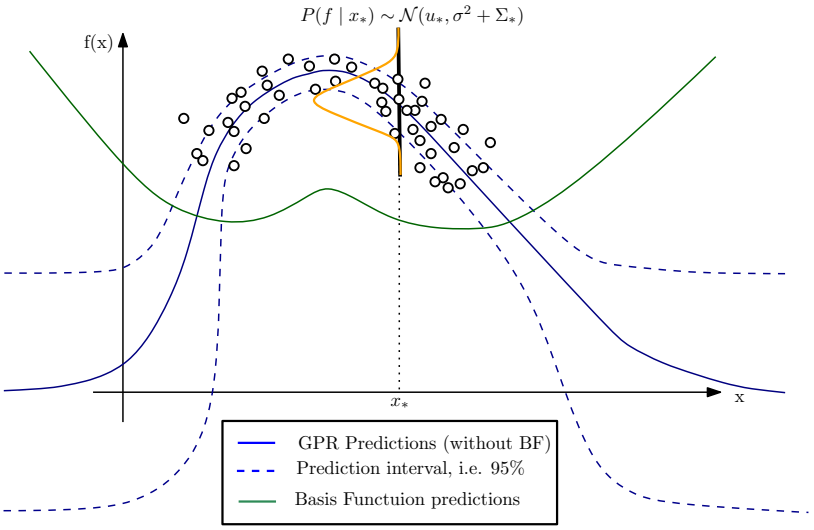


Figure 4.3: Gaussian process regression with zero mean

Now imagine that we incorporate a basis function as a mean in the GPR, instead of zero. The idea is that we would like the GPR to rely on data points in those regions spaces where is training data, and to rely on a certain basis function where new data points are uncorrelated with the training set (see **Figure 4.4**). This is possible by incorporating a basis function as a mean function of the GPR. Then, when making a prediction of a new data point, i.e. x_*, the predictive mean is the sum of the evaluation of the new point in the basis function $(H(x_*)\beta)$ plus the GPR term (u_*), which depends on how this point is correlated with the training set (kernel function). On one hand, if $K(x_*, X) = 0$ -no correlation-, the predicted mean will be simply the basis function. On the other hand, if $K(x_*, X) > 0$ -correlation-, then the predicted mean will deviate from the basis function according to the kernel co-variance function. The key point when a basis function is incorporated, is to ensure an appropriate transition phase between data points and basis function and also how far apart are those two function origi-

nally.

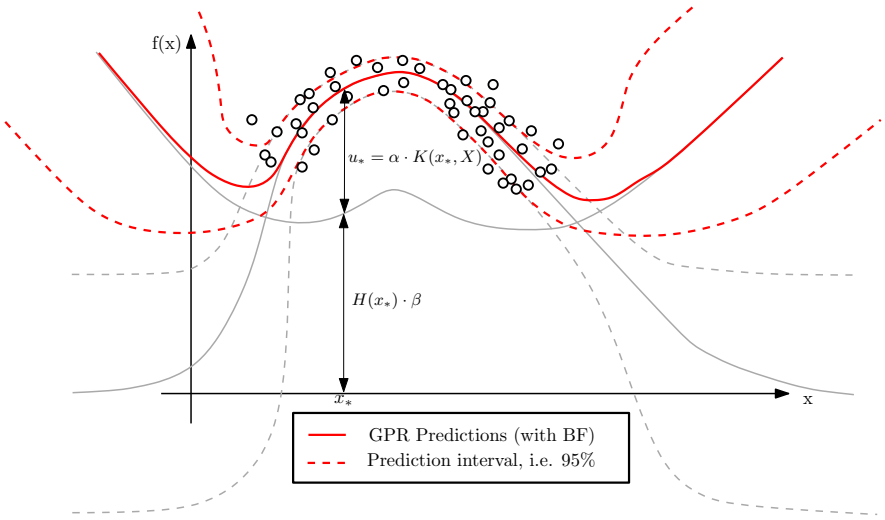


Figure 4.4: Gaussian process regression with basis function: theoretical example

Figure 4.5 depicts a real example from the data set used in this master thesis. The figure shows a GPR trained to predict the acceleration mean using spacing and speed difference between leader and following cars as a explanatory variables. It can be easily seen from the figure, how by giving a predefined basis function, the model predicts values of the basis function where there is no data and relies on the data points in those space regions where there is data. As mentioned before in this chapter, the incorporation of a basis function might be essential to have a complete model. If small acceleration and high speed difference are given as an input to the model, the predicted acceleration mean would be 0 if no basis function had been set. However, if a basis function that complies the physics of traffic theory is given to the model as in this example, the model predicts a high non-positive value of acceleration (deceleration) to avoid a possible collision.

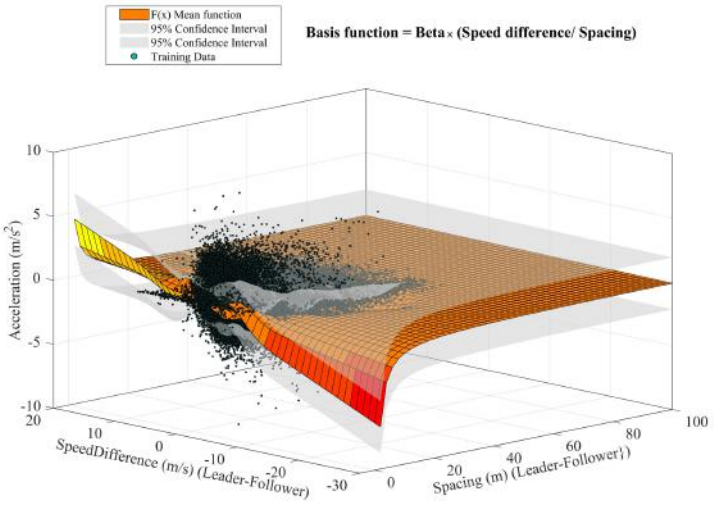


Figure 4.5: Gaussian process regression with basis function trained with real radar data

5

Methodology for GPR Model Derivation and Validation

Chapter 5 focuses on explaining the methodology applied to derive and validate the different set of GPR created. First, Section 5.1 includes a short section which summarises the key points of the different family models created in this master thesis, including a conceptual map to facilitate the comprehension to the reader. Then, a detailed explanation of how the models have been derived and the data that has been used is given in Section 5.2. Finally, Section 5.3 concludes this chapter giving insights on how models have been benchmarked and validated.

5.1. Introduction

From the previous chapter, we have seen that there exist several ways of building GPR models, which one might differ from another on which hyper-parameters are optimised -'learned'- and included in the GPR formulation. In this thesis, three types of conceptual GPR models have been created (see **Figure 5.1**). The first family of model belongs to GPR regression models fully optimised. This means that all hyper-parameters, i.e. $(\beta, \theta, \sigma^2)$, are optimised during training phase, including the parameters of the basis function. As it will be later depicted in the results section, these models do not fully ensure a complete model as the optimal basis function lies far apart from the original parametric model. In order to solve this issue, a set of models are trained without optimising the basis function, i.e. the coefficient vector of the basis function β is constant. The third family of models are simple GPR models trained without basis function , i.e. β is zero. Note that these kind of models do not present any underlying traffic equation, meaning that the GPR is exclusively derived from data. Two different objective function in the hyper-parameter learning optimisation have been used. On one hand, the first objective function applied in the thesis is to maximise the log-likelihood of the probability that the model prediction are actually the training data, as it is explained the previous chapter. This procedure is recommended in the machine learning field (Rasmussen & Williams, 2006), as it can deal with large data sets such as the one we possess. On the other hand, we also use the minimisation of the mixed error as an objective function. This error indicator measures the difference between consecutive trajectories (see Section 5.3.1 for further details of this error measure). The author believes that this objective function is more appropriate to

assess good fits in traffic behaviour. However, all comes with a price, computing the mixed error is computationally expensive as it is needed to simulate all trajectories in the training set. Hence, only a small training set can be actually used for training the model. In every family of models, all possible variable combinations will be included, leading to multiple models per family. Finally, all models will be evaluated with the same validation set in order to fairly asses the accurateness of each model.

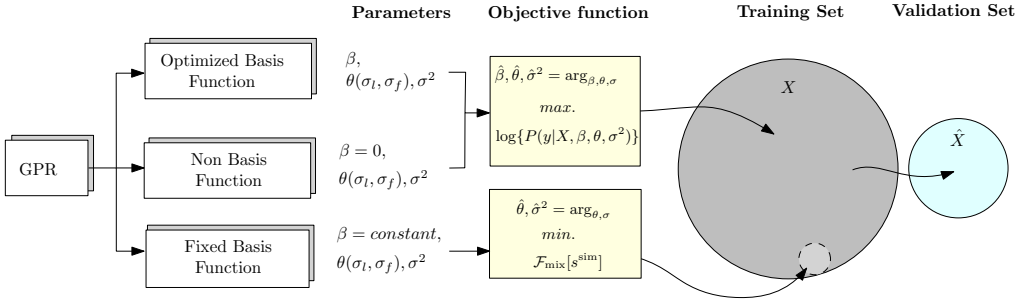


Figure 5.1: Conceptual Map of the methodology

5.2. Model Derivation

This section elaborates on how GPR models in this master thesis have been derived -trained-. First, the section illustrates which variables are going to be used to describe traffic behaviour. Then, the basis function used in this project and the initial hyper-parameters are introduced. Later, the section details both algorithms used to train the GPR models. Finally, it concludes giving some insights into the data used for training the models.

5.2.1. Variables

In this master thesis, only following longitudinal driving behaviour is aimed to be modelled. From our data set, we can never ensure that a theoretical vehicle leader, is actually the leader. The data belongs to a road section of 80 metres, which makes challenging to ensure that an actual leader from our data is not actually 'following' a vehicle outside the radar range. Therefore, our training data will specifically focus on following behaviour. However, that does not mean that our final model cannot simulate leader behaviour. GPR models with basis function should definitely be able to model leaders behaviour according to the basis function, as those points should have no correlations at all with training set, i.e. infinite spacing. For the same reason that we cannot guarantee leaders in our dataset, we also cannot ensure the number of cars downstream from a certain vehicle. This variable unfortunately cannot be used to derive a model. Consequently, this thesis will aim to accurately describe the acceleration of a following driver, based on spacing distance and the speed difference between the driver and its leader, its own speed, the status of the traffic light and its distance to the traffic light. We would like to explore the variable significance to describe traffic behaviour. Thus, we will derive a set of different variable combinations of each model. Furthermore, we will not explore the effect of the reaction time, which is left for future work. We assume that the reaction time is 3 logs, i.e. 0.7 s approximately. That means that we assume that acceleration of time step i, is modelled according to the predictor variables in time step $i-3$. Last decades, literature

from different fields has intensively tried to determine a certain reaction time value. Most of them conclude that it mostly depends on the age and psychological conditions of the driver, e.g. tiredness, but also they indicate that also might depend on variables such as speed of which the driver is driving. In order to avoid increase the thesis complexity, the reaction time is considered constant and specifically of 3 logs.

5.2.2. Basis function

In this master thesis the optimal velocity model (OVM) is chosen as a basis function with the exception of those models trained without basis function, i.e. zero mean.The Optimal Velocity Model (OVM) is a time-continuous model that describes acceleration of a driver based on its own speed and the spacing with its predecessor. Originally intruded by (Bando et al., 1995), the model adapts the actual speed v, to the optimal velocity v_{opt} on a time scale given by the adaptation time τ.

$$\dot{v}_{\text{OVM}}(s, v) = \frac{(v_{\text{opt}}(s) - v)}{\tau} \tag{5.1}$$

where,

$$v_{\text{opt}}(s) = v_0 \frac{\tanh(\frac{s}{\Delta s} - \beta) + \tanh\beta}{1 + \tanh\beta} \tag{5.2}$$

The optimal speed v_{opt}, which depends on the current spacing s, smoothly increased while spacing increases until the desired speed v_0 is reached on a certain spacing and afterwards it keeps constant. All OVM parameters τ, v_0, Δs and β are defined by positive values and typical values are depicted in **Table 5.1**.

Table 5.1: Standard parameters for OVM used in simulation. Adapted from (Treiber & Kesting, 2013).

Parameter	Highway	City Traffic	Remarks
Adaptation time τ	0.65 s	0.65 s	Higher values for trucks, e.g. 1.7s
Desired speed v_0	120 km/h	54 km/h	Smaller values for trucks, e.g. 80 km/h
Transition width Δs	15 m	8 m	-
Form factor β	1.5	1.5	-

Figure 5.2 depicts the acceleration predicted by the OVM given a specific spacing and speed and using the suggested parameters of **Table 5.1**. According to the model, drivers accelerate when the spacing with the predecessor is large and they are driving at lower speeds. The model tends to accelerate until a certain desired speed is achieved, which depends on both spacing and speed variables (red line). Deceleration is found in combination of small spacing and high speed values.

The main reason to choose this model among others, is that few number of variables are used to describe acceleration of a vehicle. OVM only includes two explanatory variables in order to predict acceleration. This will facilitate the interpretation of the performance of

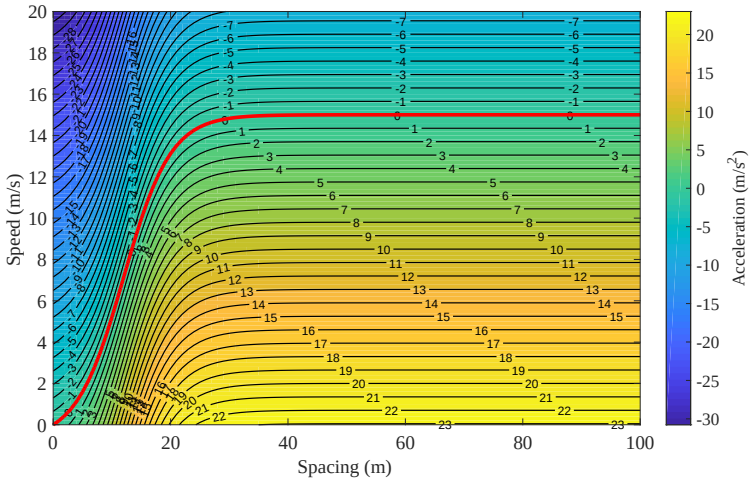


Figure 5.2: Predicted acceleration of OVM using parameters illustrated in **Table 5.1**. Red line shows the speed-spacing combinations where acceleration is zero.

the basis function in the GPR models. Moreover, according to the formulation of the GPR, the variables included in the basis function must be also included as a predictor variable. Therefore, all GPR models with basis function should at least have speed and spacing as a predictor variables. If we had chosen another car-following model with extra variables as a basis function, e.g. IDM, less model variable combinations would have been possible. Finally, the basis function is introduced into the GPR formulation as $H(X)\beta$, where H is a vector containing all variables operations and β the vector containing all parameters.

5.2.3. Optimisation procedures

Training the hyper-parameters is the key point of any GPR model. Two main optimisation procedures have been designed in this master thesis. Generally speaking, the first procedure, focuses on taking profit of big large data sets. The second procedure tries to address some issues in the results of the first procedure. As will be later explained in Section 6.1, by fully optimising the basis function, we do not completely accomplish the purpose of enforcing the model to comply with traffic physics in all space regions. Therefore, a new optimisation procedure for fixed basis function is designed.

The first procedure to derive accurate models by optimising hyper-parameters is shown in **Algorithm 3**. This optimisation procedure is both used in GPR models fully optimised and also models without a basis function. This procedure is exclusively designed to overcome the challenges of training GPR processes with large data sets. The algorithm first selects an optimal subset of 2000 points. Starting from a single point, 60 observations are randomly selected from the training set and they are scored according to its log-likelihood. The point with a maximum value is incorporated to the subset. This process is repeated until 2000 points are included in the subset or when the highest score point in any of the 2000 iteration, is smaller than a certain stopping criteria, i.e. extra points do not include extra information to the subset. In the subset selection, sparse greedy matrix approximation (SGMA) method is used to compute the $\log P(\hat{y})|\hat{X},\beta,\theta,\sigma^2)$. This is to reduce computational time avoiding directly inverting every evaluation in each iteration the inversion of the kernel function. For extra infor-

mation of the SGMA mathematical method please check (Smola & Schölkopf, 2000). Once the subset is selected, optimisation procedure described **Algorithm 1** is used to find the optimal hyper-parameters. In this case the exact method is applied, meaning that the real inversion of the kernel function is performed in every iteration of the optimisation. Finally, the hyper-parameters -(β,θ,σ^2)- are found. This process is repeated 5 times. The subset is selected in every of the 5 iterations using the hyper-parameters found in the previous iteration. This is done to avoid bias results of selecting the subset based on non optimal hyper-parameters. After the 5^{th} iteration, we assume that we posses an accurate and reliable model. Once an optimal model is found, prediction can be easily done using **Algorithm 2**. Initial parameters chosen for the kernel function are θ_0 =[mean(std(predictors))) ; std(response)/sqrt(2)] and variance, σ_0^2 =[std(response)/sqrt(2)]. Initial parameters chosen for the basis function, i.e. β, are the values suggested in **Table 5.1**.

Algorithm 3 Model Derivation Scheme 1

Input: Training Set $(X,y) \in A$, $(\beta_0,\theta_0,\sigma_0^2)$ (Initial hyper-parameters) H (Basis function)

for i=1 $\rightarrow$ 5 **do**
  while $subsetB < 2000$ **do** ▷ Subset selection
    $(\hat{X},\hat{y}) \in C \subset A$, $|C| = 60$ ▷ Select randomly a set of candidates C
    for j=1 $\rightarrow$ 60 **do**
      $\log P(\hat{y}(j))|\hat{X}(j),\beta(i-1),\theta(i-1),\sigma^2(i-1))$ ▷ SGMA method approximation
    end for
    Add max $(\log P(\hat{y}(j))|\hat{X}(j),\beta(i-1),\theta(i-1),\sigma^2(i-1)))$ to the subset B
  end while
  Algorithm 1 ▷ Hyper-parameters optimisation using Exact method
  Input Training Set B, $(\beta(i-1),\theta(i-1),\sigma^2(i-1))$ H
  return $(\beta(i),\theta(i),\sigma^2(i))$
end for

Output: (β,θ,σ^2) (Hyper-parameters)

The second procedure to derive accurate models by optimising hyper-parameters is shown in **Algorithm 3**. The approach can be seen as a traditional optimisation using GPR formulation. In every iteration a GPR is built and tested. Therefore, contrary to last scheme, the building and prediction algorithm plays an essential role (**Algorithm 2**). In this case, only (θ_0,σ_0^2) are optimised, as this algorithm is only used for fixed basis function, i.e. the vector of the basis function parameter β is constant an equal to city parameters depicted in **Table 5.1**. The algorithm is basically a minimisation of the mixed error, which measures the difference between the simulated and observed following trajectory respect to the observed leader. This error is explained in detail in the next section. The minimisation works as follows. First, the GPR is built based on **Algorithm 2**, depicted in previous chapter. Afterwards, all trajectories in the training set are predicted by the GPR using the initial point of each trajectory and the trajectory of the leader as an input. Finally, the mixed error is computed. There exist two main limitations of this method in terms of computational time. On one hand, in every iteration of the optimisation, all the trajectories in the training set need to be simulated in order to compute the mixed error. The computational time increases with the required number of simulated trajectories. On the other hand, the size of data used to build the GPR highly affects the occupational time, e.g. inversion involved in the computation of C in **Algorithm 2**. Therefore, the training set for both reasons need to be reduced. The training subset of complete following trajectories will be randomly picked from the big training set of scheme 1. Hence, no subset selection is performed compared to scheme 1. In the previous scheme, this was

carried out by adding new data points with high log-likelihood. However, this cannot be done here as training set must include complete following trajectories instead of independent measurements. Note that in the previous algorithm, it was not necessary to ensure that training points in the subset were in the same trajectory as data points are independent in the optimisation scheme. A similar subset approach could be certainly been performed in this scheme. For instance, trajectories could be iterative added to a subset. However, mixed error highly depends on the parameters. Thus, similarly to the previous scheme, the model derivation should be performed several times to avoid picking bias trajectories in the initial phase. Overall, this would be dramatically expensive in terms of computation time due to the simulation of trajectories in any calculation of the mixed error and the number of predictions required. This negative characteristic is compensated by the fact that GPR are build with a relatively higher training set (16.000 measurements) compared to scheme 1 (only 2000 measurements), as GPR model needs to be built less times in **Algorithm 4** compared to **Algorithm 3**.

Algorithm 4 Model Derivation Scheme 2

Input: Training Subset$(\hat{X}, \hat{y}) \in B \subset A$, (θ_0, σ_0^2) (Initial hyper-parameters) (β, H) Basis Function

1: **Objective Function:** $\hat{\theta}, \hat{\sigma}^2 = \arg_{\theta,\sigma} \quad \min \quad \mathscr{F}_{\text{mix}}[s^{\text{sim}}](\theta, \sigma^2)$

2: **Algorithm 2** from 2-3 ▷ Derivation of GPR

3: Predict each point in the trajectories using **Algorithm 2** from 4-7

4: Calculation of $\mathscr{F}_{\text{mix}}$ from the simulated trajectories

Output: (θ, σ^2) (Hyper-parameters)

5.2.4. Training Data

GPR models highly rely on training data to make predictions. In this case, we will use data processed from the Tuesday 7^{th} and Wednesday 8^{th} of June of 2016 from 6 AM to 20 AM for training purposes. Those specific days have been selected as they were favourable meteorological conditions. This means that no rain was registered and that the visibility was good all day[1]. Note that by selecting these days and its time-line we aim to avoid any influence of bad meteorological conditions and night light to drivers' behaviour. Moreover, both were working (business) days and no miscommunications between the radar and the registration box were reported. Certainly, there were around 10 more days available with the same conditions. However, according to some primarily tests, no significant result were obtained by using more than 2 days in the training set. At the same time, there was no difference on results depending on the day selected. Note that this amount of data still has never been seen in literature. In total, there are 788.481 measurements available in the training set, where each one includes all variables previously described. The data selection method will be later evaluated and discussed in Chapter 7.2.

As we have just specified in the previous subsection, the model derivation Scheme 2 needs a smaller data training set and complete trajectories. Moreover, it is needed the trajectory of the leader vehicles, as simulated trajectories do depend on it. For this reason, 275 complete following trajectories have been randomly selected from the previous data set with its respective leader trajectories. **Table 5.3** depicts the number of following trajectories that has a certain number of observation. In total more than 16.0000 measurements are included.

[1] According to historical meteorological data from Weather Underground

Table 5.2: Training data for model derivation Scheme 2. Number of following trajectories with certain number of observations

Observations	Trajectories
50 to 100	192
100 to 150	29
150 to 200	28
>200	6
Total	275

5.3. Validation

This section contains the description on how the performance of the multiple models generated are going to be assessed. First, the key performance indicators used to benchmark and assess the different set of GPR model created are introduced. Later, the data used in this assessment will be detailed. Finally, a subsection is included to briefly illustrate how models are going to be assessed outside space regions of the validated data.

5.3.1. Key Performance Indicator for Benchmarking

The reliability and robustness of the different GPR variable combinations fits are assessed by applying a specific performance criteria, which measures the deviations between the GPR simulated predictions and the real data. In order to find the robustness model, one main key performance indicators (KPI) is used to benchmark the different set of models: a mixed error measure.

$$\mathscr{F}_{\text{mix}}[s^{\text{sim}}] = \sqrt{\frac{1}{\langle |s^{\text{data}}| \rangle} \left\langle \frac{\left(s^{\text{sim}} - s^{\text{data}}\right)^2}{|s^{\text{data}}|} \right\rangle} \tag{5.3}$$

The mixed error $\mathscr{F}_{\text{mix}}$, defined in **Equation 5.3**, measures the difference between consecutive trajectories in terms of spacing and was proposed by (Kesting & Treiber, 2008b). Each simulated trajectory is initialised by the initial position and speed of the observed trajectory from the validation data (see **Figure 5.3**). From the second time step, the simulated trajectory becomes independent from the observed trajectory and exclusively depends on itself and the observed leader. Each time-step, the spacing square difference between the observed and the simulated trajectory is computed. Later, a temporal average is done according to the amount of the data points of the trajectory. Finally, the average from different mixed error of every specific trajectory is carried out. As shown in the formulation of this KPI, s^{data} is partially excluded from the main term. (Kesting & Treiber, 2008b) suggests this approach to avoid overestimation errors for large gaps (at high velocities) and to avoid systematically overestimates deviations of the observed headway in the low velocity range. The conceptual interpretation of the mixed error is the percentage error between the simulated and the observed trajectory.

As GPR models are not collision free, i.e. its mathematical formulation do not explicitly

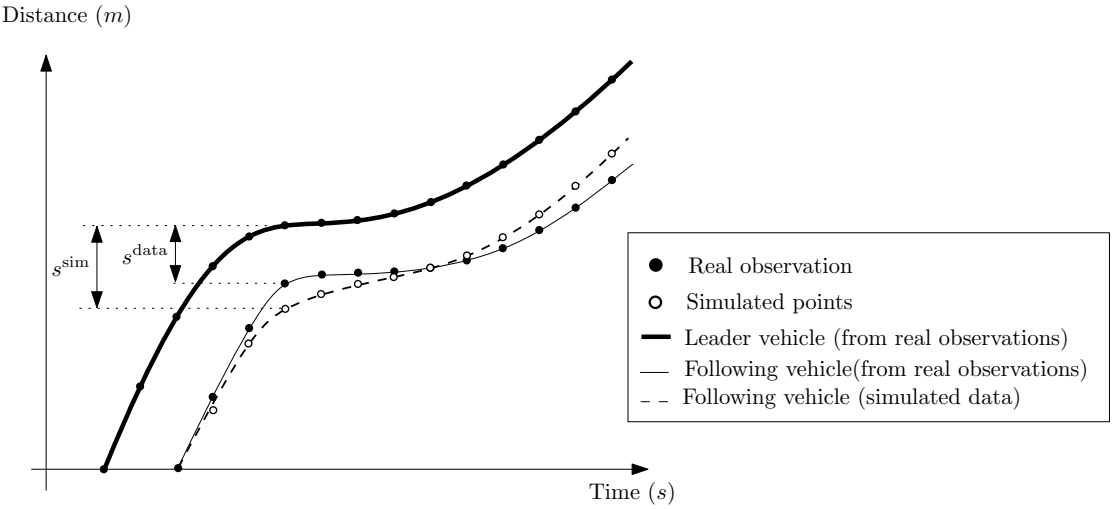


Figure 5.3: Interpretation of the mixed error

ensure no collisions, the number of collisions observed in the simulated trajectories are also going to be used as a KPI. If multiple collisions are observed in the same simulated trajectory, only 1 collisions is going to be counted. For instance, imagine that a simulated trajectory overpasses the leader trajectory, it later brakes and let the leader pass, and finally it overpasses the leader once again. Two collisions will be then observed, but only one will be counted as do not make sense to count collisions once there has already been one. It might happen that a model scores better in terms of mixed error, but it presents a bigger number of collisions. Therefore, it is essential to make a final assessment by complementing both KPI. Finally, note that the simulated spacing is always derived from the predicted acceleration mean. Thus, in this master thesis we assume the mean of the GPR as the true prediction, whereas the variance do not play any role.

Different measures for the deviations between the GPR simulated predictions and the real data are described in literature. One interesting approach to benchmark predictions is to reset at each time step the position and speed of the simulated vehicle as suggested by (Ossen, 2008). However, in this specific case, it does not seem a good option to use this KPI. The radar frequency of the measurements is rather small (e.g. 0.22 seconds). Hence, few errors could be then noticed as time steps are small. Another approach is to use the root mean square error (RMSE), which measures the difference between the acceleration in the data and the predicted in the GPR model. RMSE obviously gives a relevant insight on how good is your model. However, the author believes that it is not fully suitable for benchmarking car-following models, and especially bad for optimisation/calibration. The main reason is that acceleration points become independent in the calculation and, therefore, measurements are not affected by the previous points in the same trajectory. For instance, when the RMSE measures the difference between the simulated acceleration and the real acceleration, it could be that the input would have never been the actual input given to the GPR, e.g. the GPR might had decelerated in previous points of the trajectory and therefore the input would be radically different of the one using in the RMSE. This would become unacceptable if this KPI is used for optimisation/calibration purposes as it will lead to bias results depending on the model formulation. Hence, only the mixed error together with the number of collisions are going to be used to benchmark GPR models. Moreover, the mixed errror is also used as a objective function in GPR trained with fixed basis function as explained in previous section. Moreover, the mixed error it will also be used to calibrate existent models to compare their performance with the GPR models.

5.3.2. Validation Data

A complete day data set of one lane is exclusively used for validation purposes. Thus, this data set it is not included in the training data set. Specifically, data from the 09^{th} of June of 2016 is used. Using the same criteria as in previous section, favourable conditions were observed that specific day. **Table 5.3** depicts the number of trajectories that posses a certain number of observations which are used for validation purposes. In total, 2790 individual following trajectories with a minimum of 10 continuous measurements are used to assess the models. By using a large number of trajectories, we aim to reduce the sensitivity of the results and to dissipate possible errors in the validation data itself. Note that together with the following trajectories, it is also needed the leader trajectories as simulated trajectories depend on it.

Table 5.3: Validation data: number of following trajectories with certain number of Observations

Observations	**Trajectories**
10 to 50	1851
50 to 100	308
100 to 150	264
150 to 200	157
>200	78
Total	2790

5.3.3. Model Assessment Outside Validation Data Space Regions

By performing an empirical validation with the before mentioned KPI, we are exclusively assessing the validity of the model inside data space region of the validation (and training) data, e.g. road section of 100 metres in a urban signalised intersection. However, one of the goal of this project is to derive a complete model. Hence, it becomes essential to assess also the completeness of a model. In order to do so, a simulator has been built in Matlab programming software. The best GPR models are going to be visually assess in the urban signalised environment simulated in the software. Leader trajectories will be always simulated using existent car-following models, while following vehicles will be generated by the GPR models.

6

Results

Chapter 6 contains the fundamental results of this master thesis. First, the results of the 47 GPR models created are presented and summarised in Section 6.1. Then, results details of each of the three model families are given and compared, including a set of tables and figures to help the reader to understand the relevance of each model. After, in Section 6.2, results are visually assessed outside validation data space regions. Finally, the results are compared to the existent OVM parametric model in Section 6.3

6.1. Model Results

The results of this master thesis are summarised in **Table 6.1**. Several models are created by combining different predictor variable included to train models (e.g. spacing $s_{n,n-1}$, speed v_n, speed difference $\Delta v_{n,n-1}$, the distance to the traffic light x_{TL} and status of the traffic light) and by optimising, fixing or removing the basis function. In total, 47 models are tested and evaluated according to the KPI described in the previous chapter and using the validation dataset. By visually checking the trajectories simulated by GPR trained models, $\mathscr{F}_{\text{mix}}$ with smaller values than 30%, represent favourable results. Models with $\mathscr{F}_{\text{mix}}$ between than 30% and 50% are intermediate results, as the average error per trajectory is near to the half of the relative spacing. Finally, models with $\mathscr{F}_{\text{mix}}$ greater than 50% represent bad fitted models to our specific data. Overall, results depict that several GPR models are able to make accurate predictions, i.e. $\mathscr{F}_{\text{mix}}$ smaller than 20%. However, we can not guarantee any free collision model. For example, there are nearly 200 collisions in best models, which represents a collision of 6% approximately of the trajectories in the validation set. Speed difference definitely seems the most important variable to describe driver behaviour at urban signalised intersection according to our dataset as even models without basis function exclusively trained with this variable predict reasonable acceleration. On the other hand, the rest of variable do not seem relevant enough to separately describe the traffic behaviour. Generally speaking, adding more variable results in more accurate models, i.e. lower $\mathscr{F}_{\text{mix}}$ and less collisions. It should be highlight that these results only refer to the space region that we possess data, i.e. the assessment is performed according to how similar is the model compared to the validation data. Again, note that some variable combinations are not possible while training a GPR with basis function. Following three subsections describe the results of each type of trained model.

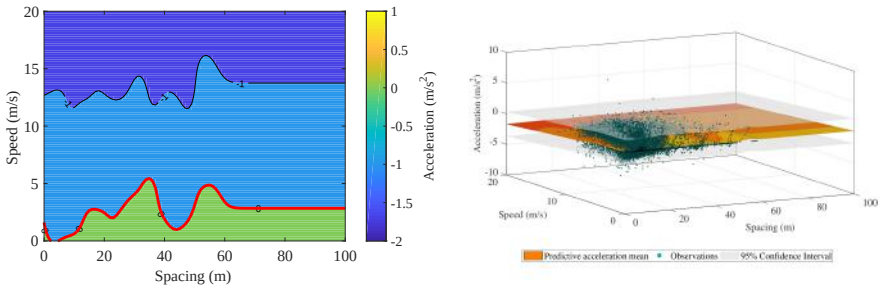
Table 6.1: Combination results

		Variables					Results					
							Optimised Basis Function**		**Fixed Basis Function****		**Without Basis Function**	
		$s_{n,n-1}$	v_n	$\Delta v_{n,n-1}$	x_{TL}	Status$_{TL}$	$\mathcal{F}_{\text{mix}}[s]$	Collisions**	$\mathcal{F}_{\text{mix}}[s]$	Collisions**	$\mathcal{F}_{\text{mix}}[s]$	Collisions**
1	**One Variable**	✓					-	-	-	-	**355,8%**	965
2			✓				-	-	-	-	**134,8%**	940
3				✓			-	-	-	-	**31,0%**	394
4					✓		-	-	-	-	**225,7%**	927
5						✓	-	-	-	-	**185,6%**	955
6	**Two Variables**	✓	✓				**172,6%**	968	**38,0%**	811	**285,2%**	978
7		✓		✓			-	-	-	-	**102,9%**	460
8		✓			✓		-	-	-	-	**396,8%**	975
9		✓				✓	-	-	-	-	**542,6%**	960
10			✓	✓			-	-	-	-	**28,3%**	340
11			✓		✓		-	-	-	-	**139,3%**	901
12			✓			✓	-	-	-	-	**62,1%**	760
13				✓	✓		-	-	-	-	**34,8%**	347
14				✓		✓	-	-	-	-	**25,4%**	275
15					✓	✓	-	-	-	-	**208,4%**	851
16	**Three Variables**	✓	✓	✓			**115,0%**	509	**19,1%**	177	**73,8%**	363
17		✓	✓		✓		**113,1%**	933	**35.2**	774	**320,7%**	969
18		✓	✓			✓	**53,9%**	594	**32.6**	659	**92,1%**	604
19		✓		✓	✓		-	-	-	-	**147,2%**	647
20		✓		✓		✓	-	-	-	-	**69,3%**	352
21		✓			✓	✓	-	-	-	-	**468,3%**	934
22			✓	✓	✓		-	-	-	-	**37,1%**	432
23			✓	✓		✓	-	-	-	-	**23,1%**	194
24			✓		✓	✓	-	-	-	-	**60,7%**	741
25				✓	✓	✓	-	-	-	-	**29,4%**	257
26	**Four Variables**	✓	✓	✓	✓		**146,4%**	689	**20,8%**	634	**56,1%**	332
27		✓	✓	✓		✓	**22,1%**	215	**17,5%**	256	**24,0%**	177
28		✓	✓		✓	✓	**151,4%**	839	**30,7%**	267	**86,1%**	731
29		✓		✓	✓	✓	-	-	-	-	**112,6%**	503
30			✓	✓	✓	✓	-	-	-	-	**28,8%**	330
31	**Five Variables**	✓	✓	✓	✓	✓	**120,9%**	529	**17,9%**	192	**52,4%**	355

* Some variable combinations are not possible while training a GPR with the selected basis function
** Collision out of the total number of following trajectories in the data set (2790)

6.1.1. Results with Optimised Basis Function

GPR fully trained, i.e. when the basis function is also optimised, presents significant different results depending on the selected variables. The mixed error, $\mathscr{F}_{\text{mix}}$, ranges from 22 to 172% in the 8 different GPR models trained. **Figure 6.1** depicts results of a GPR model trained with only spacing and speed as an explanatory variables, which are the same predictor variables than the OVM. This GPR model represents a bad fitted model as it presents a mixed error of nearly 175% and 968 collisions out of 2709. As can be seen in the figure, acceleration mostly depends on speed rather than spacing. Moreover, the predicted mean has few variability and the basis function tend to a negative constant value outside space regions of the training data. Overall this result significantly differs from the original OVM parametric formulation depicted in **Figure 5.2**



(a) 2D plot *(b)* 3D plot

Figure 6.1: Predicted mean acceleration of a GPR model trained with spacing and speed as predictor variables and with optimised OVM basis function. Red line shows the speed-spacing combinations where acceleration is zero mean.

Figure 6.2 illustrates two simulated trajectories using the previus GPR model. As can be observed in the figure, simulated drivers (red circles) do not even brake to avoid collision with the observed leader (grey circles). Hence, it is proved the bad $\mathscr{F}_{\text{mix}}$ score of this GPR model (172%) and its high number of collisions. This result might indicate that both speed and spacing alone, cannot describe drivers' longitudinal acceleration given the dataset and also that this GPR methodology might not be satisfactory.

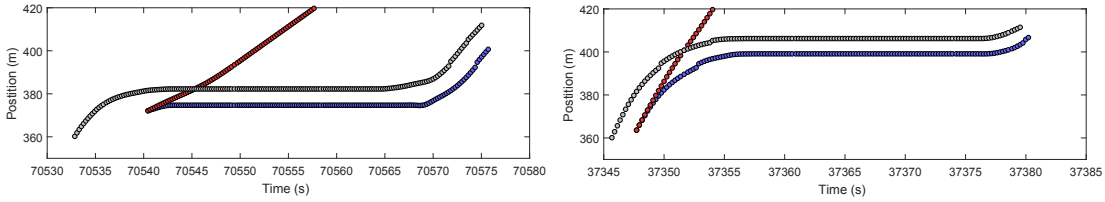


Figure 6.2: Simulated versus observed trajectories. GPR model trained with spacing and speed as predictor variables and an optimised basis OVM function.

Overall, results are not satisfactory according to both KPI, i.e. high mixed errors and high number of collisions (see **Table 6.1**). However, there is one exception. To highlight, the GPR model trained with spacing, speed, speed difference and the status of the traffic light presents satisfactory results, with only 22% of mixed error and 215 collisions (7% of the total amount of trajectories validated). **Figure 6.3** depicts several simulated trajectories with this model from the validation set. Hence, the incorporation of the traffic light and especially speed difference variable to the GPR seems essential.

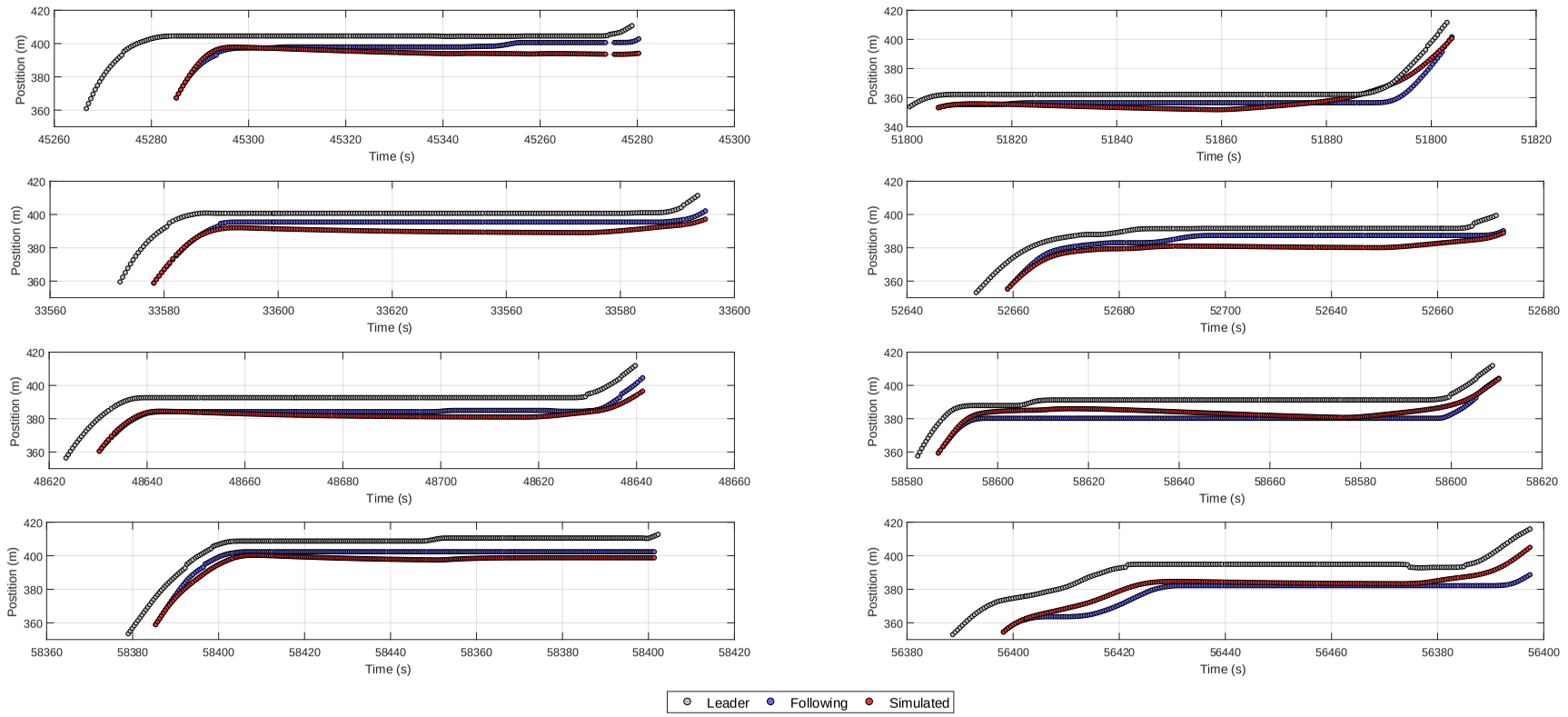
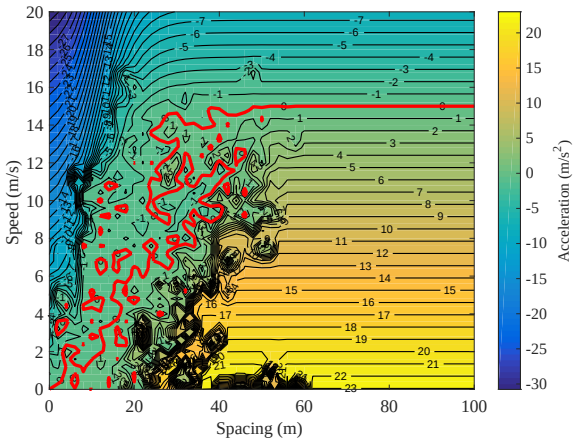


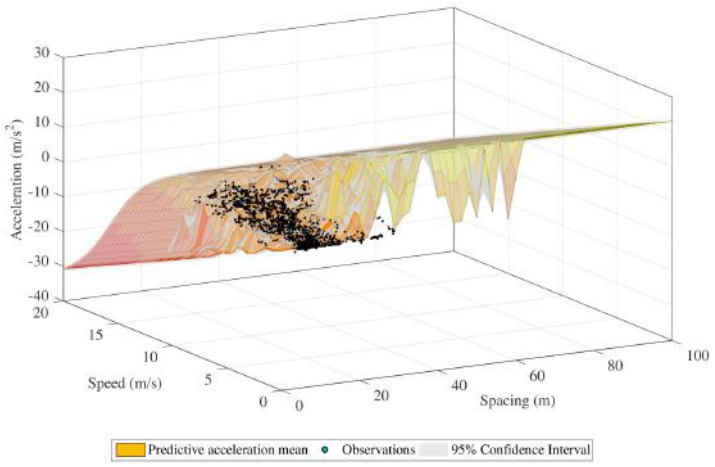
Figure 6.3: Simulated versus observed trajectories. GPR Model trained with spacing, speed, speed difference and status of the traffic light as predictor variables and an optimised basis OVM function

6.1.2. Results with fixed Basis Function

GPR Models with fixed basis function have been designed to try to address two main problems found in the GPR models with optimised basis function results. First, the optimised GPR models seem not performing well according to both KPI in most of the models. Second, the optimised basis function usually tends to constant negative values close to zero. Therefore, the objective function might be forcing the basis function to be constant and nearly zero, $\beta \approx 0$. As explained in the Section 5.2.3, the main idea to solve this is to use **Algorithm 4** and remain β constant and equal to the values depicted in **Table 5.1**. Another main difference with the previous GPR models is that in this case, $\mathcal{F}_{\text{mix}}$ is used as an objective function instead of the log-likelihood. Generally speaking, significantly better results are obtained according to both KPI(s). Similarly to previous subsection, **Figure 6.4** depicts an example of GPR model trained with speed and spacing as predictor variables.



(a) 2D plot



(b) 3D plot

Figure 6.4: Predicted mean acceleration of a GPR model trained with spacing and speed as predictor variables and with fixed OVM basis function. Red line shows the speed-spacing combinations where acceleration is zero mean.

There are major differences between the previous approach, i.e. fully optimising the basis function, and this approach, i.e. fixing the basis function, as is shown both in **Figure 6.1**.

Both models rely on one hand in the basis function when predictions are needed in space regions not found in the training data, and in the other hand models rely in the data when predictions are required in correlated space regions between the new input and the training set. However, by fixing the basis function to specific values, we force the model to have a particular shape when no correlation between new input and the training data is found. Thus, a recognisable solution is found compared to the results in previous subsection. To highlight, the transition between the parametric and non parametric model seems to play a major role in this kind of GPR models. Overall, this model presents a mixed error of 38%, much lower compared to the optimised GPR model (172%). However, there is not a similar drastic reduction in the number of collision (from 968 to 811 collisions). Therefore, we can conclude that by fixing the basis function we slightly improve the performance of the GPR model, but still both variables do not fully describe the driver behaviour according to the dataset. The main reason to explain a drastic reduction of the mixed error ($\mathscr{F}_{\text{mix}}$), but not in the number of collisions is that the predicted acceleration of the optimised basis function model tends to zero or a small negative value outside training data space regions. After a collision, spacing turns negative. Obviously, this is never observed in the training data, so predicted acceleration becomes 0 or slightly negative, usually leading to an increase of negative spacing in every iteration when the leader is stopped. This substantially increases the total average mixed error. Opposite to that, the models with fixed basis function realise that there has been a collision thanks to the basis function as the spacing turns negative. Then, this GPR models try to decelerate and even go backwards, leading to relatively better mixed error compared to the previous models. Therefore, similar numbers of collisions are registered in both models but of course, a small mixed error is found if a fixed basis function is used.

In total, 8 different models are trained using different variable combinations. Generally speaking, significantly better results are obtained by fixing the basis function according to both KPI(s) compared t (see **Table 6.1**). Best GPR models with fixed basis function and according to the mixed error ($\mathscr{F}_{\text{mix}}$) are models trained with spacing, speed, speed difference and the status of the traffic light, with only 17.5%. **Figure 6.5** depicts several randomly picked trajectories of this GPR model. It can be observed how sometimes GPR models present collisions. The best model according to the number of collisions is a GPR model trained with spacing, speed and speed difference as explanatory variables, with 177 collisions out of 2790 trajectories (6%). Therefore, we can conclude that by fixing the basis function we improve the performance of the GPR model, but still collisions occur.

6.1.3. Results without Basis Function

Results with optimised basis function and without a basis function are quite similar. The main reason is that, as explained previously, the optimal basis functions resulted from the optimisation procedure predominately tend to zero mean, which equals the structure of the GPR without a basis function, where $\beta = 0$. This can be observed by comparing **Figure 6.6** and **Figure 6.1**. Both models acceleration predictions far apart from the training set space regions are approximately zero. Again, GPR models without a basis function and trained with speed and spacing are not capable to predict accurate accelerations.

Nonetheless, deriving models without a basis function give us much freedom as there are no variables requirements from the basis function. In total, 31 models are trained, with combinations of 1 to 5 variables. Results show that the mixed error, $\mathscr{F}_{\text{mix}}$, ranges from 22 to

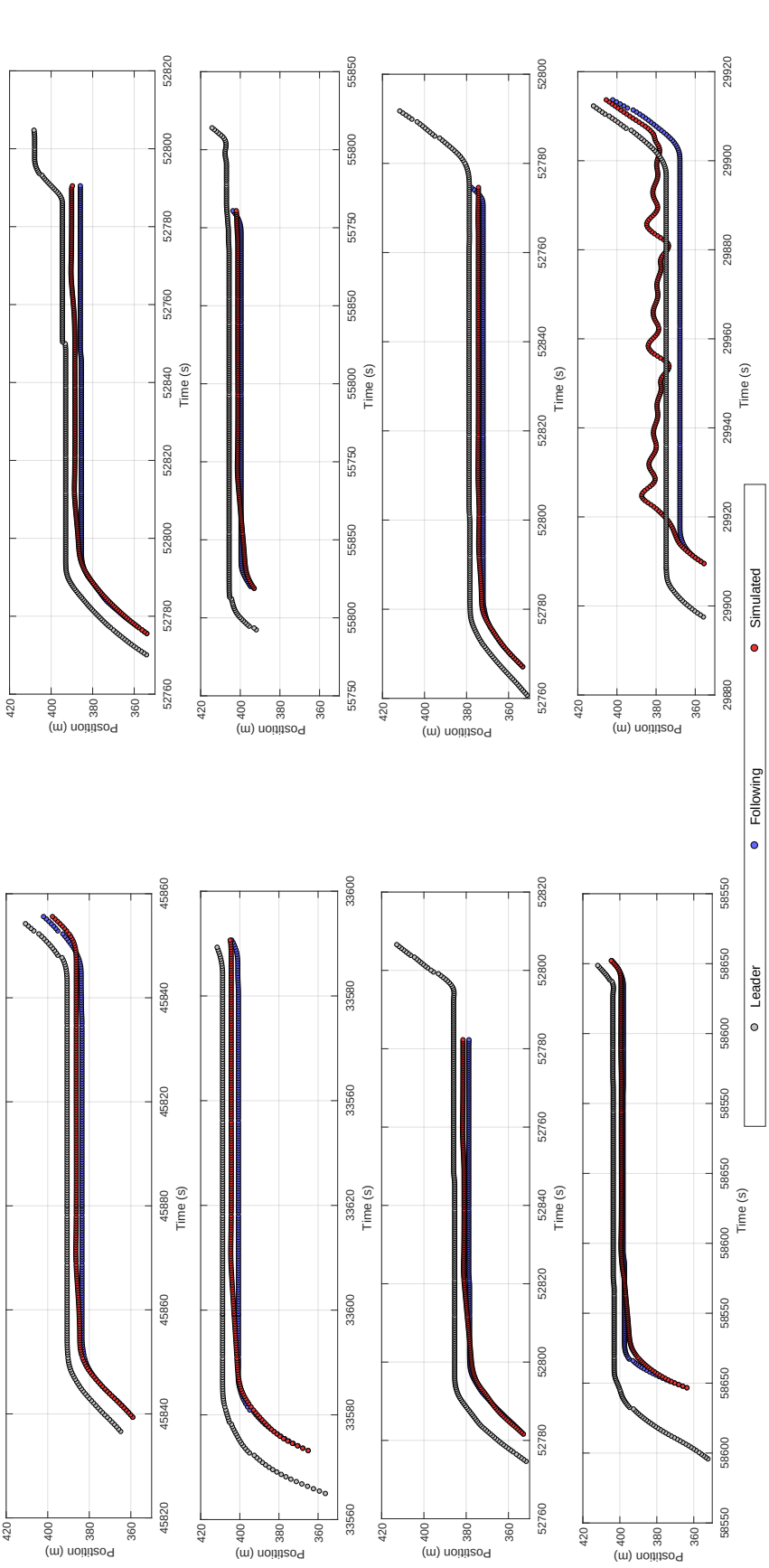
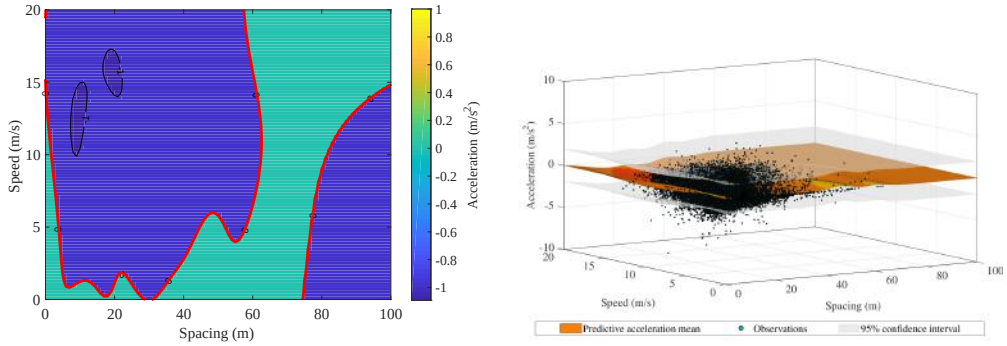


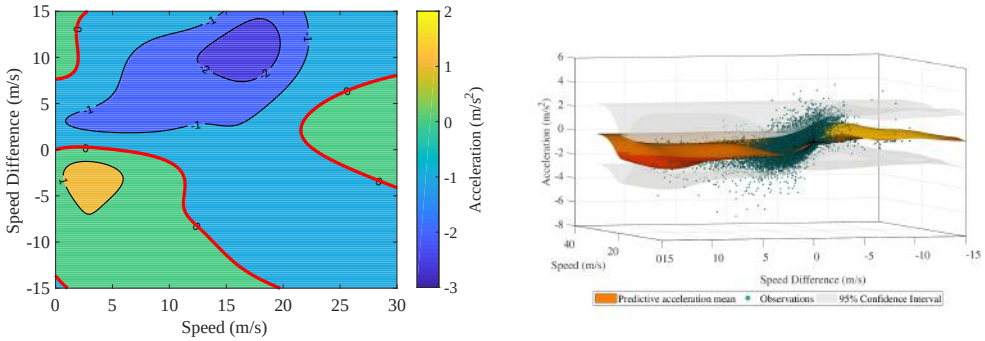
Figure 6.5: Simulated versus observed trajectories.GPR Model trained with spacing, speed, speed difference and status of the traffic light as predictor variables and a fixed basis OVM function



(a) 2D plot *(b)* 3D plot

Figure 6.6: Predicted mean acceleration of a GPR model trained with spacing and speed as predictor variables and without basis function. Red line shows the speed-spacing combinations where acceleration is zero mean.

542%. Speed difference definitely is the most significant variable. A GPR trained only with this variable achieves a mixed error of only 31% and 394 collisions. By adding speed, the model slightly reduces its mixed error to 28% and 340 collisions. **Figure 6.7** depicts the predicted acceleration according to speed and speed difference between the driver and its predecessor as an example of an intermediate performing model. On one hand, when drivers are slower than the car in-front (negative speed difference) and their speed is low, the predicted acceleration is obviously positive. On the other hand, higher speed and positive speed difference leads to drivers deceleration. Hence, results are reasonable leading to a favourable model. Note that there exist two isolated regions in the model: 1. combinations of high positive speed difference and nearly 0 speed values; 2. combinations of high speed and small speed difference values-.The acceleration prediction in those regions is zero as no training data is found. Thus, the model behaves as expected. Generally speaking, results without basis function are promising. No traffic underlying equation is set to the model. Instead, independent measurements are given to the GPR, which learns the underlying behaviour of the data. Therefore, it is necessary to highlight the good results of the GPR model such as the model derived with spacing, speed, speed difference and the status of the traffic, which achieves a mixed error of 24 % and 177 collisions out of 2790 (6%).



(a) 2D plot *(b)* 3D plot

Figure 6.7: Predicted mean acceleration of a GPR model trained with speed and speed difference between preceding vehicles as predictor variables and without OVM basis function. Red line shows the speed-speed difference between preceding vehicles combinations where acceleration is zero mean.

6.2. Model Completeness

In this section we are going to briefly evaluate the completeness of models in simulated real urban signalised intersection. One of the main reasons to incorporate a basis function into the formulation, was to ensure that the model could deal with inexperienced situation, i.e. situations not found in the training set. Therefore, we should assess the so-called completeness of each model. A simple signalised intersection simulation has been built in Matlab software, where the leader is always simulated using IDM and the following vehicles using the selected GPR model. The assessment is simply being carried out visually. Best models according to the mixed error, $\mathscr{F}_{\text{mix}}$, of each GPR family have been picked and simulated. The following figures give an insight of how the three types of models behave in a real simulator.

Figure 6.8 depicts a simulation using GPR model with an optimised basis function. Overall, its performance seems reasonable. However, some collisions can be observed, particularly when small spacing are registered. Then, the model seems not capable to brake on time to avoid a collisions. Overall, as explained in previous section, the main purpose of a basis function is not fully achieved.

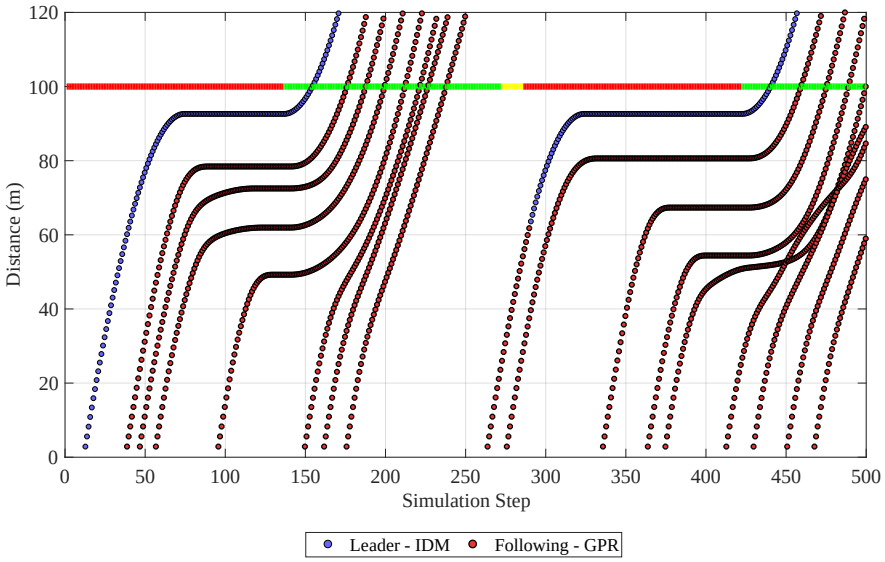


Figure 6.8: Assessment of GPR model with optimised basis function completeness

Figure 6.9 depicts a simulation using GPR model with fixed basis function. In this case, better performance can be observed. Simulated vehicle trajectories seem reasonable. However, occasionally collisions can be observed if the simulation is repeated several times, proving that the model is not unfortunately free of collisions even if the basis function is fixed.

Figure 6.10 depicts a simulation using GPR model without basis function. This model was learned without any basic traffic equation. Therefore results show what machine learning can achieve in the microscopic traffic field. Once a collision is occasioned, differently to GPR models with fixed basis function, the model do not how to react and zero acceleration is predicted (constant speed).

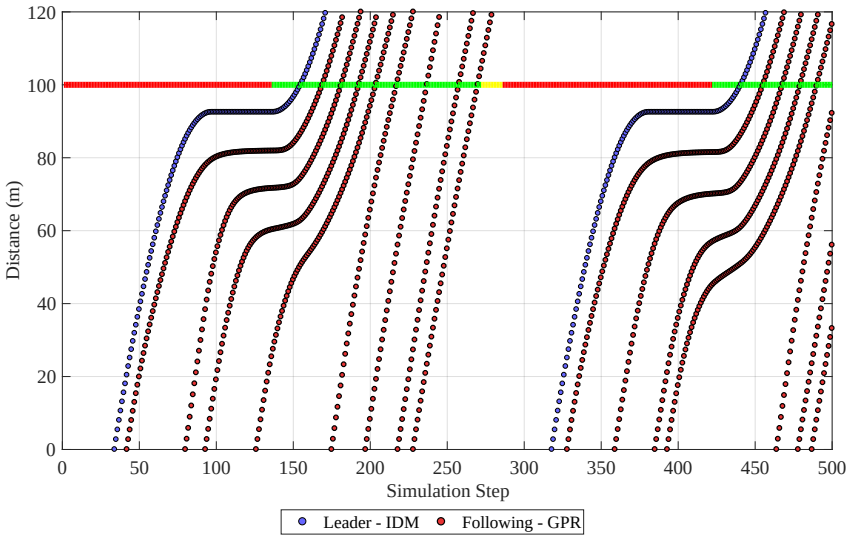


Figure 6.9: Assessment of GPR model without basis function completeness

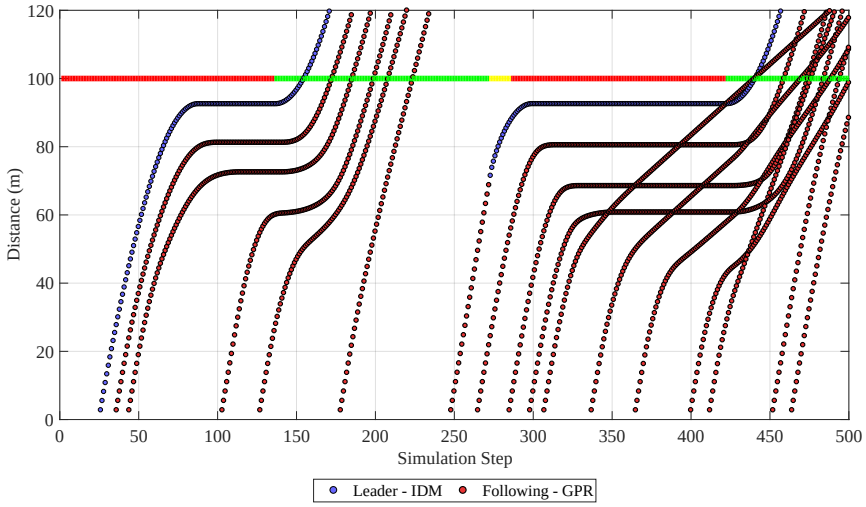


Figure 6.10: Assessment of GPR model without basis function completeness

No empirical conclusions can be derived from this section. For instance, trajectories are initialised randomly, which might affect the driver behaviour predicted by the GPR. However, this section gives a visual insight on how GPR models rely and learn from experience data, but at the same time can present underlying traffic equation to guarantee appropriate traffic behaviour when there are no measurements. Thus, the section visually proves that the three different types of models behave according to what we were expecting while enumerating its formulation.

6.3. Model Comparison

This sections compares the main findings of this thesis to current practice. This is carried out by contrasting the GPR models results to the OVM parametric model. In other words, we analyse if there is an improvement by applying GPR formulation to turn a parametric model into an hybrid model. First, this section explains the traditional calibration procedure and results of the OVM model given the same dataset of this thesis. Finally, the results are analysed and compared to the GPR results.

6.3.1. Traditional Calibration Results

The OVM has been traditionally calibrated using the reduce dataset described in Section 5.3, which was also used for training GPR models in the second model derivation scheme. Traditional calibrations techniques consist on finding an optimal parameter set according to the OVM car-following parametric formulation that best fit the dataset. This is done by means of a nonlinear optimization problem and using a classic optimisation algorithm, i.e. interior point. Initial parameters in the optimisation were set to the values in **Table 5.1** and parameters constraints were set according to (Treiber et al., 2000). The mixed error, $\mathscr{F}_{\text{mix}}$, is used as an objective function (minimisation of the mixed error). Similarly to model derivation scheme 2, this calibration scheme also involves the simulation of all trajectories in every iteration of the optimisation and thus requires a reduce dataset. In order to fairly compare these models with the GPR models, once the optimal parameters were found, OVM has also been validated with the same large validation dataset that was used to validate the GPR models. **Table 6.2** depicts the calibration results.

Table 6.2: OVM Calibration Results

Dataset	Optimisation Set	Validation Set	
KPI	$\mathscr{F}_{\text{mix}}[s^{\text{sim}}]$	$\mathscr{F}_{\text{mix}}[s^{\text{sim}}]$	Collisions
Results	19.4%	18.6%	0

Optimal model parameters found in the optimisation are illustrated in **Table 6.3**.

Table 6.3: Optimal OVM parameters

Parameter	**Value**
Adaptation time τ	0.48 s
Desired speed v_0	28.60 km/h
Transition width Δs	1.56 m
Form factor β	3.67

6.3.2. Comparison

Results shows that OVM score worst than the best GPR models in terms of the mixed error, i.e. 17.5% vs. 18.6%. Nonetheless, OVM presents no collisions while best GPR models present

around 175 collisions out of 2790 trajectories. **Figure 6.10** depicts the acceleration predicted by the OVM given a specific spacing and speed values and using the optmimal parameters found in the calibration and illustrated in **Table 6.3**.

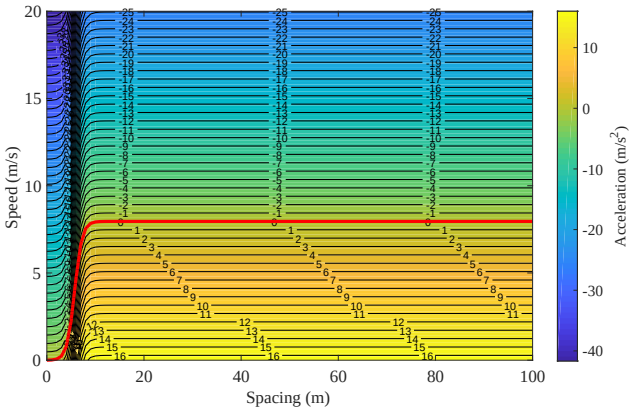


Figure 6.11: [Predicted acceleration of OVM using optimal parameters illustrated in **Table 6.3**. Red line shows the speed-spacing combinations where acceleration is zero.

The figure clearly depicts one of the main problems of traditional optimisation techniques applied to parametric models using data in relatively small space data regions: overfitting. According to the results, the calibrated OVM model is quite accurate in similar space regions of the dataset and presents typical mixed error ranges found in literature (Treiber et al., 2000). Yet, outside space regions of the dataset, the acceleration predictions are poor. This is evident while comparing this figure to the solution shown in **Figure 6.3** using other generic parameters. In that case, the desired speed is 54 km/h. Thus, if spacing is big enough, drivers only accelerate until achieving that specific speed value and then they keep their speed constant (zero acceleration). Nonetheless, in this case, the desired speed is 28 km/h, which might be a low value to represent desired speed in city traffic and represents wrong model estimation. Overall, all the above mentioned shows the importance of the GPR hybrid formulation to ensure a complete model and avoid overfitting issues such as this one.

III

Discussion and Conclusions

7

Discussion

In this chapter, a discussion will be presented on the content of this master thesis. Section 7.1 will first discuss the relevancy of the results. In order to derive the proposed GPR, several assumptions, choices and simplifications were made that led to certain implications or drawbacks. This, will be discussed in Section 7.2. Section 7.3 discusses the applicability of these kind of models to commercial software. Finally, Section 7.4 argue about the future of these models

7.1. Discussion on the Results

In this master thesis we aimed to gain new empirical insights into longitudinal driving behaviour, particularly at urban signalised intersections and using traffic radar detection. This technology represents an alternative to classic technologies to obtain massive data. However, the radar data available for this thesis was full of errors, inconsistencies and noise as seen in Section 3. Moreover, the dataset was limited to 80 metres of road section, which unfortunately did not include first 15 metres in front of the traffic light due to the radar's range. Therefore, GPR model was thought to be the best ML technique to take profit of the large dataset available, to avoid overfitted results and to ensure a complete model outside those space regions were no training data was found. GPR's mathematical approach offers a combination between traditional parametric models and machine learning techniques by incorporating the so-called basis function. Three types of conceptual GPR models have been trained in this master. The first family of model belongs to GPR regression models fully optimised. This means that all hyper-parameters, i.e. $(\beta, \theta, \sigma^2)$, are optimised during training phase. However, results shows that in most of the cases optimising the basis function leads to bias results. In order to solve that, a second family has been trained by fixing the basis function. Moreover, different optimisation procedure and objective function have been used. Finally, it was also decided to derive models without basis function in order to see the real power of machine learning, which can learn out of data to derive a mathematical formulation. On one hand, best models results of each of the three family of model depicted in Section 6.1 are similar according to both KPI: the mixed error and the number of collisions. These results depict that the core of the GPR formulation model completely relies on the data. Consequently, no big difference is found if there is a fix, optimised or non basis function. On the other hand, only by fixing the

basis function we achieve a complete model as is it visually shown in Section 6.2. Therefore, it is not recommended to optimise the basis function and it is better to keep the basis function fixed according to your needs outside the training set data space regions. This implies that the basis function do not really influence the GPR when there is correlation between the input of the prediction and the training data. This can also be derived by the relatively high values of the kernel optimised parameters in the optimal hyper-parameters set from the best fitted models. Those high values forces the model to highly rely in the data when there is actually data.

Several models with different variables combination have been trained in this master thesis. Taking a look to variable importance, speed difference appears to be the most important variable to describe traffic behaviour according to both KPI. One main reason to explain the importance of this variable is that stop and go traffic conditions are frequently found in our dataset. Therefore, speed difference between drivers and its leader becomes a relevant variable in this kind of traffic conditions, e.g. approaching vehicle to a standstill leader. Opposite to OVM, GPR models are not capable to describe accurately the traffic behaviour with only spacing and speed variables in the given dataset. The results also highlight that the status of the traffic lights affects traffic behaviour. Generally, the most accurate models are achieved by including spacing, speed, speed difference and the status of the traffic light. For the first time in literature, this variable is included in the mathematical formulation to describe traffic behaviour. Finally, distance to the traffic light seems not significantly affecting the results. Hence, results do not allow us to state that drivers behaves according to their distance to the traffic light. However, it is also worth to mention that only measurements between 15 to 80 metres to the traffic light are included into the dataset. Thus, we might miss relevant traffic behaviour close to traffic light itself (e.g. drivers choice to accelerate or decelerate in combination of yellow traffic light and short distances).

The best GPR models achieve less than 20% of mixed error. This errors are consistent with typical error ranges in literature (Treiber et al., 2000). Furthermore, we have calibrated the OVM parametric model with the same radar dataset. This is carried out to analyse if there is an improvement by applying GPR formulation to turn a parametric model into an hybrid parametric and non parametric model. GPR model with fixed basis function scores better results than OVM in terms of mixed error (17.5% vs. 18.6%). However, still the original OVM model ensures no collisions, while the GPR model occasionally predicts collisions. When collision occur in one trajectory of the validation set, bad results of that specific trajectory are registered, e.g >100%. Thus, this means that the model is really accurate in those trajectories where there are no collision in order to compensate bad results in all trajectories where collisions are registered. Collisions are mainly predicted by the GPR model when following drivers are driving quite close to the leader (small spacing values). Then, when the leader stops in a traffic light, the model is not able to predict high deceleration values and therefore vehicles sometimes are not able to brake on time. The main hypothesis is that spacing measurements are sometimes not reliable due to the noisy position measurements collected by the radar. In this thesis we assumed that each measurement belonged to the front part of a vehicle. Furthermore, we assumed all vehicle length of 5 metres. Therefore, to obtain the net spacing, the length of a vehicle was subtracted to the distance measured between consecutive drivers position measurements. However, in reality, radar might be measuring other parts of the vehicle such as the side part of a vehicle instead of the front point. Moreover, by smoothing the position measurements in the data processing section we are definitely altering the reference point. Consequently, we might be committing a considerable error of dozens of centimetres and even few meters. However, few dozens of centimetres in spacing are essential for any

car-following to simulate stop and go traffic conditions and to avoid collisions. In the training set can be sometimes observed really small spacing (i.e. smaller than 0.5 metres) with a great variability of speeds and speed difference values, which seems quite unrealistic. This leads to high variance and an inaccurate predicted mean acceleration when spacing are relatively small (<2m), which are usually found in deceleration phases. Other variables such as speed and speed difference are also derived from position measurements. Therefore, they should also present significant errors. However, this is not the case. Position measurements are projected to a reference line after smoothing trajectories in both coordinates, and then speed is derived. This procedure helps to reduce the noise in the speed, and consequently speed difference, in the dataset. Obviously, some errors are inherit but they are not significant given the big range of values (e.g. no big difference in car following models between 2 or 2.5 m/s). The fact that sometimes collisions are registered proves that one the purposes of the basis function is not achieved. Overall, the formulation of the GPR is good enough to deal with noise. The GPR solution is the predictive mean and the variance of the acceleration. The solution is directly derived from the hyper-parameters, which includes the noise of the measurements. One of the benefits of the GPR formulation, is that σ (noise) can also be optimised if its value is unknown as in this thesis. However, the noise and errors from the spacing might be too significant to ensure a free collisions model, especially given the fact that spacing is one of the most important variable as depicted in literature in Chapter 2, particularly in stop and go traffic conditions.

Opposite to GPR, the OVM do not seem highly affected by the noisy spacing measurements. OVM parametric formulation is simple, but at the same time robust against noisy data in small space regions. However, traditional calibration of the OVM results in an overfitted model. Although prediction are quite accurate in the validation set space regions, an analysis of the results outside those space regions depicted unrealistic results (see Section 6.3.2). This shows that traditional techniques may fall into overtitting problems if reduced space regions are used to calibrate the model. GPR formulation do not present overfitting issues due to its hybrid formulation.

7.2. Discussion on the Methodology

Empirical results of this master thesis are not outstanding, but they definitely give some insights of new powerful mathematical techniques such as GPR that can be applied to describe longitudinal drivers' behaviour. Therefore, is needed an appropriate discussion on the methodology.

i **Radar Data:**
The data has great implications to GPR model. GPR predictions highly depend on observations if new inputs for prediction are correlated with training dataset. Therefore the characteristics of the data have a great impact on the results. On one hand, in order to reduce the impact of inaccurate and noisy observation, it is essential to better understand and quantify the noise and errors. It would be necessary that the manufacturer could provide extra information regarding this issue and could be included in the formulation of the GPR (e.g. fixed value of noise σ). On the other hand, a better positioning of the radars would also help to obtain better results. For this master thesis, radar detection data from the PPA project in Amsterdam has been used. This data

was not collected for the purpose of this thesis as it has been pointed out in Section 3.2. The major drawback is that there is no data registered within 15 metres in front of the traffic light. In order to register complete data, the radar should be placed in such a way that the radar measures a road section of at least 100 metres, including the front and the rear sections of the traffic light. According to the technology description described in Section 2.2, the radar should be placed completely perpendicular to the road streams to favour front measurements and improve spacing variable accuracy. If all these conditions are not possible with just one radar, multiple coordinated radars should be used. However, other challenges such as synchronisation will then arise.

ii **Optimisation:**
On one hand, **Algorithm 3** was chosen according to generic Matlab standards codes for GPR in the machine learning field. It was thought that this algorithm and its optimisation algorithm would lead to satisfactory results. The algorithm tries to overcome computational time challenges by finding a subset of data that fully represents the whole dataset. Results shows satisfactory results and variables insights but also proved that the procedure is not fully able to describe accurate longitudinal traffic behaviour without collisions. One hypothesis is that the subset of 2000 points is not enough representative. However, it is not feasible for regular computers to highly increase this number, as in every iteration to find a new subset point, the GPR needs to be computed for all candidates. On the other hand, a new optimisation procedure was designed explicitly to deal longitudinal traffic behaviour, where the objective function focused on reducing the error between simulated and observed trajectories (e.g.**Algorithm 4**). Therefore, measurements were not independent anymore and they belonged to a certain trajectory. A smaller training set of 16.000 points (270 trajectories) was used to train the models. However, this time no subset selection was carried out and the 16.0000 points were directly used to built the GPR in each iteration. Results showed that even if the mixed error was reduced in all models, still similar collisions to the previous procedure were observed. Therefore, it is proved that the optimisation method worked in terms of mixed error (objective function) but still failed to find an optimal solution free of collision. The reduction of the dataset to 16.000 points seems not really affecting the results as the optimisation results are similar to the results in the large validation dataset.

iii **Changes:**
The methodology presented in this thesis do not ensure a collision free model. By including the basis function, it was thought that collisions could be avoided. However, space regions where collisions occur are too spatially close to the training set (small spacing and low speeds). The GPR model tends to believe the training data instead of relying into the basis function, which would predict much higher deceleration rates and would avoid a collision. This indicates that the transition between the non parametric model to the parametric model is not strict enough to avoid traffic physics violations. Therefore, seems essential to modify some parts of the methodology to try to solve this issue. One option would be to (manually) force the GPR model to rely on the basis function whenever a collision is expected in near future time steps (really small spacing registered). This could be seen as an emergency brake. Nonetheless, this measure might lead to noncontinuous model and frequent spontaneous radical change of driver behaviour. Alternatively, it would have been interesting to design an optimisation procedure that somehow introduces the concept of collision. This has been tried without success. A penalty error was introduced in **Algorithm 4** when a collision was registered (e.g. a high value was given to the mixed error of that specific trajectory). However,

result highly depended on the penalty value that was given to the trajectory.

Looking back, another way of trying to solve this issue is to focus data quality instead of methodology itself. It is necessary further analysis of spacing in the data analysis section in order to make correct assumptions (e.g. vehicle length). Spacing is proven from literature to be relevant in all car-following models, but results of this thesis illustrate that this variable do not play a major role.

7.3. Applicability of GPR models to commercial software

Calibrated parametric car-following models are widely used in commercial microscopic traffic simulators such as VISSIM, PARAMICS or AIMSUN. Those models allow for fast acceleration predictions of all simulated drivers in small time-steps thanks to its simplistic mathematical formulation. GPR models, opposite to parametric models, are methods computationally expensive to make predictions as explained in chapter 4. It is true that part of its prediction formulation can be carried out offline, i.e. calculation of α, but still in any prediction of a single driver, X_*, the calculation of steps 3-4 and 6 in **Algorithm 4** need to be computed. This includes the calculation of the kernel matrix $K(X_*, X)$, which defines how close is the new input to the training set. Hence, depending on the size of the training set, this calculation can be really expensive in terms of computational time and memory usage. Therefore, if high accuracy models are required, then fast online prediction become impossible in regular computers. A possible solution to that is to perform offline the prediction of all possible combinations of variable given a certain limits. All combination should be stored and sorted in a matrix. Then when a prediction is needed, a search algorithms such as the one presented in (Shen, 1997) could be applied to rapidly find the required combination and give the prediction. Obviously, this prediction method would be still slower than traditional parametric models, but might be a feasible solution to overcome this challenge.

7.4. Future of GPR Models

Results of this thesis proved that GPR models can improve a traditional parametric car following model such as OVM in terms of performance, but still they present some violations of traffic physics (e.g. collisions) and computational times issues. The main question that arises is if GPR can at some point be able to fully derive accurate car following models. The mathematical formulation of the GPR is complete and robust, and it can be applied to any field. It is obvious that more powerful computers in the future will help to develop these kind of techniques, but it is also true that not always are needed big datasets. Hence, two scenarios can be differentiated. On one hand, if relatively small datasets but with few inconsistencies and noise are found, GPR can definitely be used. Parallel to this thesis, a GPR models were trained with a relatively small training set from simulated trajectories using other CF models and using stochastic desired speed per driver. Results were outstanding and showed that GPR model could predict accurately the drivers trajectories and no traffic violations were registered. On the other hand, if large datasets with inconsistencies, noise and stochastic behaviour are found as in this thesis, GPR predictions are more accurate than traditional parametric models but sometimes violate traffic physics. However, the GPR formulation allows

flexibility and it is sure that the issues described in this thesis will be overcome. Generally speaking, the overall results are satisfactory and give a first insight of what this kind of powerful mathematical models can actually do. Summarising, this project shows a new methodological approach of deriving mathematical hybrid models to describe any process that want to be improved in a specific space regions by data observations.

8

Conclusion and Recommendations

This last chapter contain the conclusions and recommendation of this master thesis. First, in Section 8.1 the main findings are presented, including the answer to the research questions of this project. Next, in Section 8.2 includes recommendations both for practical use and for further research.

8.1. Conclusions

The goal of this research was to gain new empirical insights into longitudinal driving behaviour, especially at urban signalised intersections. Particularly, the goal focused on obtaining a calibrated car following model in stop and go urban traffic conditions which predicts acceleration of drivers. In order to overcome the challenges of the characteristics of new datasets available, non parametric models using machine learning techniques were investigated. To reach this goal, the research question was supported by a set of sub-question that needed to be answered:

i **What is the quality of preceding trajectories in the traffic radar detection data set?**
Traffic radar data measurements of this thesis frequently contain errors, gaps and noise. Therefore, an extensive data processing is carried out in order to get reliable predictors and response variables measurements. First, the noisy data out of the range of the radar is deleted. Then, trajectories are independently smoothed for x and y coordinates. Immediately after, measurements are mapped to lanes and variables such as speed and acceleration are computed. Later, the preceding assignment is carried out by first mapping all incomplete trajectories. Despite losing 2% of reliable data due to wrong estimation, it is avoided that 10% of all reliable points present a wrong preceding assignment. Finally, the distance to traffic light and its status is assigned to each measurements. It is also worthy to mention that only measurements between 15 to 80 metres to the traffic light are included into the data set. Thus, we might miss relevant trajectories information and this has represented a major challenge. Data analysis of the processed dataset depicts hundreds of thousand of leader and reliable points. Although most variable obtained from data processing are accurate, spacing sometimes present extremely low values and has represented a drawback to avoid collisions. This is mainly due to inaccurate

position measurements of the radar, which do not always guarantee the front part of a vehicle as a reference point due to the radars location.

ii **Which are the main significant dynamic variables relationship between preceding vehicles?**
In this thesis we have assumed that the reaction time is 3 logs, i.e. 0.7 s approximately. That means that we assumed that acceleration depends dynamically on predictors variables of 0.7 seconds in the past. Speed difference is most important variable to describe traffic behaviour. The main reason to explain this fact is that stop and go conditions are frequently found in our data set. Therefore, speed difference between drivers and its leader becomes the most relevant variable in this kind of traffic conditions, e.g. approaching vehicle to an standstill leader. Surprisingly, spacing is not a really relevant, mainly due the errors depicted in this thesis. GPR models are not capable to accurately describe traffic behaviour with only spacing and speed variables in the given the data set. The results also highlight that the status of the traffic lights affects traffic behaviour. Generally speaking, most accurate models are achieved by including spacing, speed, speed difference and the status of the traffic light, meaning that this variable are the main significant variables between preceding vehicles. Finally, distance to the traffic light seems not significantly affecting the results.

iii **How considerable are variable relationships that takes into account traffic light distance and status?**
The most accurate models are achieved by including spacing, speed, speed difference and the status of the traffic light. For first time in literature, this variable is included in the mathematical formulation to describe traffic behaviour and therefore seem relevant. Contrarily, distance to the traffic light seems not significantly affecting the results. Nonetheless, it is also worthy to mention that only measurements between 15 to 80 metres to the traffic light are included into the data set. Thus, we might miss relevant traffic behaviour close to traffic light itself (e.g. drivers choice to accelerate in combination of yellow and short distances).

iv **Which Machine Learning technique should be used to fit the processed data taking into account the data characteristics?**
Gaussian Process Regression (GPR) for machine learning (ML) has been used to take advantage from the large data set available and to ensure a complete model outside those space regions were no training data was found. GPR's mathematical approach offers a combination between traditional parametric models and machine learning techniques by incorporating the so-called basis function. The basic idea is that the GPR relies on the training data if new data input for prediction is not far apart of the training set, and it relies on a basis function (parametric model) when no correlation between new input and training data exist. In this project, the optimal velocity model has been chosen as an underlying model, i.e. basis function. The prediction of the GPR model is the mean acceleration and its variance (normal distribution).

v **What is the new model accuracy terms of the selected KPI compared to existent parametric models?**
The best GPR models achieve less than 20% of mixed error. This error is consistent with typical error ranges in literature. Furthermore, we have calibrated the OVM parametric model with the same radar dataset. This is carried out to analyse if there is an improvement by applying GPR formulation to turn a parametric model into an hybrid parametric and non parametric model. GPR model with fixed basis function scores bet-

ter results than OVM in terms of mixed error in the space regions of the validation set (17.5% vs. 18.6%). Moreover, whereas OVM suffers significant overfitting issues and predictions become unrealistic outside validation set ranges, GPR formulation guarantee a complete model thanks to the fixed basis function. However, still the original OVM model ensures no collisions, while the best GPR model occasionally predicts collisions.

Finally, the main research question can be answered:

How can the longitudinal urban driver behaviour at signalized intersections be modelled using non parametric models and machine learning techniques?

Longitudinal urban driver behaviour at signalised intersection can be modelled using Gaussian Process Regression models with machine learning techniques and trained by traffic radar detection. This thesis shows a new methodological approach of deriving mathematical hybrid models. Particularly, the thesis depicts how any process described by a parametric model, can be improved in a specific space regions where new data is available by using Gaussian Process Regression and machine learning techniques. Focusing on longitudal driver behaviour, acceleration can be predicted by GPR models trained with spacing, speed, speed difference and traffic light as predictor variables. Results of this thesis proved that GPR models can improve a traditional parametric car following model such as OVM in terms of performance, but still they present some violations of traffic physics (e.g. collisions) and computational times issues. Spacing, which should be one of the main variables to describe traffic behaviour in stop and go traffic conditions, presents significant noise due to inaccurate position measurements. Computational time issues in prediction can be faced by precomputing offline of all the possible model solutions so they might not represent a big challenge for its real market application. Therefore, with a more accurate position measurement, GPR models would not face any problem and seems a promising new methodology to derive microscopic traffic models. Compared to parametric models, GPR formulation allows having a complete model and avoids overfitted models. It also allows flexibility in its formulation and it is sure that the listed issues will be overcome in future research. Last but not least, few literature of machine learning techniques and car-following models is found in this topic, proving the innovative approach of this thesis. Results are not outstanding, but they definitely give some insights of a new powerful mathematical techniques that can be applied to describe drivers behaviour or any modelled process.

8.2. Recommendations

While doing this project, knowledge of the field was acquired and ideas for additional working directions arose. Thus, all these can be written as suggestions for practical use of GPR models or for further scientific research.

Recommendations for Practice

i **Use appropriate radar data:** Traffic radar detection seems a feasible way to collect massive microscopic traffic data for ML techniques. However, in this specific

project, the radar data was not appropriately installed for our purpose. Hence, we missed several essential parts of the road. If radars are installed as mentioned in the discussions, then variables such as distance to the traffic light and others driver behaviours such as coasting in red traffic light cycles will be easily observed.

ii **Election of the Basis Function:** Basis function do not play a major role in the core of the GPR according to the results and as discussed in the previous chapter. Therefore, the basis function and its parameters should be exclusively selected according to the user needs outside the training dataset space regions. Furthermore, the parameters of basis function need to remain fixed while optimising the other hyper-parameters.

iii **Traffic light:** Results proved that traffic light status should be included as a predictor variable to describe drivers acceleration. Nonetheless, the distance between drivers and the traffic light seems not affecting the drivers behaviour according to our dataset. Thus, we cannot recommend the inclusion of this explanatory variable into the GPR model.

iv **Optimisation Algorithm and Objective function:** Algorithm 4 is an adequate algorithm to optimise the hyper-parameters to describe longitudinal driving behaviour. The mixed error within trajectories is a better KPI to describe longitudinal traffic behaviour compared to the log-likehood as the whole trajectory is considered instead of independent measures.

v **Trade off between accuracy and training and prediction time:** One of the main drawbacks of Machine learning techniques, and particularly for the GPR, is that training and predicting is expensive in terms of computational time and memory. Therefore, crucial choices need to be done in order to reduce training sets which not always can be fully empirically justify. Unfortunately, as in other fields, there is no receipt that tells you how much you can reduce your dataset in order to still have accurate results on a reasonable computational time. This mainly depends on your dataset. Then, there is no other choice that to try different parameters combinations and check its performance to determine if all the dataset is relevant or not for your purpose.

Recommendations for Future Research

i **Use simulated data as training data:** the first recommendation would be to use real observed data plus simulated data to train the GPR model. Instead of using a basis function and an hybrid model, the idea is to simulate all missing data according to the basis function and include this data directly into the training set. Hence, only non parametric kernel GPR formulation is necessary. By doing so, we might eliminate the issues with the transition phases between parametric and non parametric model, but certainly will increase the workload of simulating all missing trajectories. Moreover, the completeness of the model needs to be ensured manually by including all trajectories possible.

ii **Further analysis of the GPR variance:** In this master thesis, the variance did not play any major role. Predictions are assumed the mean of the GPR. Therefore, a suggestion is to explore the how the variance can be incorporated in the predictions of trajectories to incorporate drivers stochastic behaviour.

iii **Use different hyper-parameters per variable:** Another relevant change that would presumably increase the model accuracy is the inclusion of different kernel function parameters for each predictor variable. This might increase the optimisation complexity, but in the other hand would help to easily interpret variables importance and could finally lead to more accurate results. Similarly, the noise could also differ per explanatory variable, i.e. higher noise in spacing measurements compared to speed measurements.

iv **Evaluate and analyse different driver behaviour:** Another way to directly analyse different driver behaviour and at the same time benefit from large data set, could be training several models. First, the data could be clustered according using other machine learning techniques such as neural networks. Then a different set of models could be trained from the different subset of data. Hence, every individual model depending the clustered objective, could depict different traffic behaviour according to the driver characteristics, meteorological conditions, or day type among others.

v **Analysis of the drivers reaction time:** In thesis, we did not explore the effects of the reaction time and it was assumed to by of 3 consecutive measurements. Therefore, given the amount of data available, seems reasonable to research how the reaction time differs among drivers and even if that depends on other factors related to traffic characteristics, e.g. stop and go conditions, or meteorological conditions, e.g. rainy vs. sunny day, among others.

vi **Macroscopic Relations:** another recommendation is to analyse the macroscopic implication of the GPR models. Using micro-macro relation between spacing-density and headway-flow, the fundamental diagram (FD) could be derived. From this diagram, the model consistency and robustness could also be analysed.

References

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., & Sugiyama, Y. (1995). Dynamical model of traffic congestion and numerical simulation. *Physical review E*, *51*(2), 1035.

Booth, C. J., & Kurpis, G. P. (1993). *The new ieee standard dictionary of electrical and electronics terms* [Book]. IEEE New York, USA.

Brackstone, M., & McDonald, M. (1999). Car-following: a historical review [Journal Article]. *Transportation Research Part F: Traffic Psychology and Behaviour*, *2*(4), 181-196.

Chandler, R. E., Herman, R., & Montroll, E. W. (1958). Traffic dynamics: studies in car following [Journal Article]. *Operations research*, *6*(2), 165-184.

De la Escalera, A., Armingol, J. M., & Mata, M. (2003). Traffic sign recognition and analysis for intelligent vehicles [Journal Article]. *Image and vision computing*, *21*(3), 247-258.

Federal Highway Administration, U. (n.d.). *Fnext generation simulation (ngsim)*. Retrieved from https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

Fritzsche, H.-T. (1994). A model for traffic simulation [Journal Article]. *Traffic Engineering+ Control*, *35*(5), 317-21.

Gipps, P. G. (1981). A behavioural car-following model for computer simulation [Journal Article]. *Transportation Research Part B: Methodological*, *15*(2), 105-111.

Helbing, D., & Tilch, B. (1998). Generalized force model of traffic dynamics. *Physical review E*, *58*(1), 133.

Herbrich, R., Lawrence, N. D., & Seeger, M. (n.d.). Fast sparse gaussian process methods: The informative vector machine [Conference Proceedings]. In *Advances in neural information processing systems* (p. 625-632).

Hidas, P. (2006). Evaluation and further development of car following models in microscopic traffic simulation [Journal Article]. *WIT Transactions on The Built Environment*, *89*.

Hofleitner, A., Herring, R., & Bayen, A. (2012). Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning [Journal Article]. *Transportation Research Part B: Methodological*, *46*(9), 1097-1122.

Hoogendoorn, S., Hoogendoorn, R., & Daamen, W. (2011). Wiedemann revisited: new trajectory filtering technique and its implications for car-following modeling [Journal Article]. *Transportation Research Record: Journal of the Transportation Research Board*(2260), 152-162.

Hoogendoorn, S., Landman, R., Van Kooten, J., & Schreuder, M. (2013). Integrated network management amsterdam: Control approach and test results. In (p. 474-479).

Kesting, A., & Treiber, M. (2008a). Calibrating car-following models by using trajectory data: Methodological study [Journal Article]. *Transportation Research Record: Journal of the Transportation Research Board*(2088), 148-156.

Kesting, A., & Treiber, M. (2008b). Calibrating car-following models by using trajectory data: Methodological study [Journal Article]. *Transportation Research Record: Journal of the Transportation Research Board*(2088), 148-156.

Khodayari, A., Ghaffari, A., Kazemi, R., & Braunstingl, R. (2012). A modified car-following model based on a neural network model of the human driver effects. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *42*(6), 1440–1449.

Kikuchi, S., & Chakroborty, P. (1992). Car-following model based on fuzzy inference system [Journal Article]. *Transportation Research Record*, 82-82.

Kuhn, H. W. (1955). The hungarian method for the assignment problem [Journal Article]. *Naval research logistics quarterly*, *2*(1-2), 83-97.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic flow prediction with big data: a deep learning approach [Journal Article]. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865-873.

Mathworks. (n.d.-a). *Display memory information.* Retrieved from http://nl.mathworks.com/help/releases/R2013a/matlab/ref/memory.html#brl1pdy

Mathworks. (n.d.-b). *Exact gpr method.* Retrieved from https://nl.mathworks.com/help/stats/exact-gpr-method.html

Mathworks. (n.d.-c). *Filtering and smoothing data.* Retrieved from https://nl.mathworks.com/help/curvefit/smoothing-data.html

Mathworks. (n.d.-d). *Gaussian process regression models.* Retrieved from https://nl.mathworks.com/help/stats/gaussian-process-regression-models.html

Mathworks. (2016). *Introducing machine learning.*

May Jr, A. D., & Harmut, E. (1967). Non-integer car-following models [Journal Article]. *Highway Research Record*(199).

Medina, J. C., Benekohal, R. F., & Ramezani, H. (2012). *Field evaluation of smart sensor vehicle detectors at intersections–volume 1: Normal weather conditions* (Report No. 0197-9191).

Mende, R. (2010). A different kind of radar all together [Magazine Article]. *Thinking Highways*, *6*.

Olstam, J. J., & Tapani, A. (2004). *Comparison of car-following models* (Report No. 0347-6030).

Ossen, S. J. L. (2008). Longitudinal driving behavior: theory and empirics [Journal Article].

Panwai, S., & Dia, H. (2005a). Comparative evaluation of microscopic car-following behavior [Journal Article]. *IEEE Transactions on Intelligent Transportation Systems*, *6*(3), 314-325.

Panwai, S., & Dia, H. (2005b). A reactive agent-based neural network car following model. In *Intelligent transportation systems, 2005. proceedings. 2005 ieee* (pp. 375–380).

Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables, and stochastic processes* [Book]. Tata McGraw-Hill Education.

Pomerleau, D. A. (1991). Efficient training of artificial neural networks for autonomous navigation [Journal Article]. *Neural Computation*, *3*(1), 88-97.

Quiñonero Candela, J. (2004). *Learning with uncertainty-gaussian processes and relevance vector machines* (Thesis).

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian processes for machine learning* (Vol. 1) [Book]. MIT press Cambridge.

Saifuzzaman, M., & Zheng, Z. (2014). Incorporating human-factors in car-following models: a review of recent developments and research needs [Journal Article]. *Transportation research part C: emerging technologies*, *48*, 379-403.

Schulz, A., Brockfeld, E., Kelpin, R., Parnitzke, A., & Wagner, P. (2003). Clearing house for transport data & transport models-concept and implementation.

Shen, H. (1997). Optimal algorithms for generalized searching in sorted matrices. *Theoretical computer science*, *188*(1-2), 221–230.

Smartmicro. (2016). *Umrr traffic sensor type 30 data sheet* (Report). Author.

Smartmicro. (2017). *Umrr traffic sensor type 40 data sheet* (Report). Author.

Smola, A. J., & Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning.

Snelson, E. L. (2008). *Flexible and efficient gaussian process models for machine learning* [Book]. University of London, University College London (United Kingdom).

Soergel, U. (2010). Review of radar remote sensing on urban areas [Book Section]. In *Radar remote sensing of urban areas* (p. 1-47). Springer.

Treiber, M., & Helbing, D. (2003). Memory effects in microscopic traffic models and wide scattering in flow-density data [Journal Article]. *Physical Review E*, *68*(4), 046119.

Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations [Journal Article]. *Physical review E*, *62*(2), 1805.

Treiber, M., & Kesting, A. (2013). Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*.

Wiedemann, R. (1974). Simulation des strassenverkehrsflusses [Journal Article].

Wiedemann, R., & Reiter, U. (1992). Microscopic traffic simulation: the simulation system mission, background and actual state [Journal Article]. *Project ICARUS (V1052) Final Report*, *2*, 1-53.

Wood, S. (2012). Traffic microsimulation–dispelling the myths [Journal Article]. *Traffic Engineering and Control*, *53*(9), 339-344.

Ławrynowicz, A., & Tresp, V. (2014). Introducing machine learning [Journal Article]. *Perspectives On Ontology Learning. AKA Heidelberg*.