Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2017

Driver Model for Mission-based Driving Cycles

Marcus Almén

LINKÖPING UNIVERSITY

Master of Science Thesis in Electrical Engineering

Driver Model for Mission-based Driving Cycles

Marcus Almén

LiTH-ISY-EX--17/5086--SE

Supervisor: **Sogol Kharrazi**
VTI, Linköping

Examiner: **Erik Frisk**
ISY, Linköping University

Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

When further demands are placed on emissions and performance of cars, trucks and busses, the vehicle manufacturers are looking to have cheap ways to evaluate their products for specific customers' needs. Using simulation tools to quickly compare use cases instead of manually recording data is a possible way forward. However, existing traffic simulation tools do not provide enough detail in each vehicle for the driving to represent real life driving patterns with regards to road features.

For the purpose of this thesis data has been recorded by having different people drive a specific route featuring highway driving, traffic lights and many curves. Using this data, models have then been estimated that describe how human drivers adjust their speed through curves, how long braking distances typically are with respect to the driving speed, and the varying deceleration during braking sequences. An additional model has also been created that produces a speed variation when driving on highways. In the end all models are implemented in MATLAB using a traffic control interface to interact with the traffic simulation tool SUMO.

The results of this work are promising with the improved simulation being able to replicate the most significant characteristics seen from human drivers when approaching curves, traffic lights and intersections.

Acknowledgments

I would like to thank Erik Frisk and Sogol Kharrazi for their valuable input during our meetings and for helping me narrow down the scope of the thesis. Many thanks also to Anders, Jonas, Erik, Andreas and Laban for making my time at VTI very enjoyable and for making me feel welcome. I would like to send a big thank you to the people at Universidad Nacional de Colombia involved in making the TraCI toolbox for MATLAB without which an implementation of this work would still be far away.

And of course I would like to thank my friends for their fantastic company and my parents for their support during my time at the university.

Linköping, August 2017
Marcus Almén

Contents

Notation

Abbreviations

Abbreviation	Meaning
FTP-75	Federal Test Procedure
UDDS	Urban Dynamometer Driving Schedule
NEDC	New European Driving Cycle
UDC	Urban Driving Cycle
EUDC	Extra Urban Driving Cycle
SUMO	Simulation of Urban Mobility
TraCI	Traffic Control Interface
XML	Extensible Markup Language
GPS	Global Positioning System
IMU	Inertial Measurement Unit
CAN-bus	Controller Area Network Bus
RMSE	Root Mean Square Error
MA	Moving Average

1 Introduction

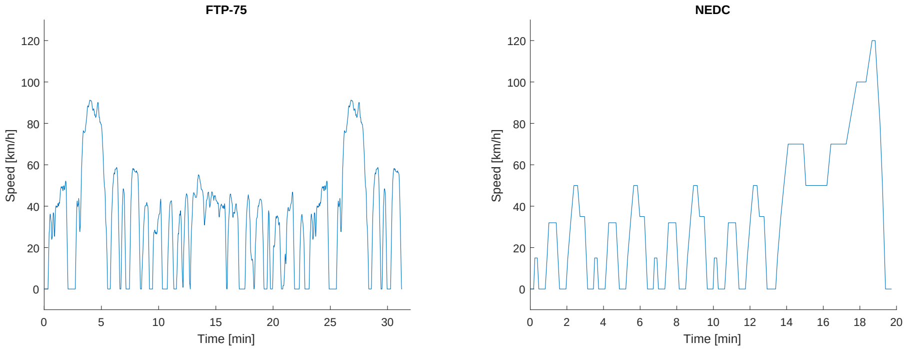
This chapter serves as a broad introduction to the different areas of research that this thesis builds upon.

1.1 Driving cycles

Driving cycles is an important tool for the evaluation of engine emissions and fuel consumption and are used to give vehicles environmental certification in a country or region. Driving cycles are also used by vehicle manufacturers in order to optimize powertrains based on specific use cases. A driving cycle is essentially a speed profile recorded over a certain amount of time designed to mimic regular usage of the vehicle type it's supposed to test. Numerous driving cycles are used around the world to assess the environmental impact of vehicles. Some of the most well known driving cycles are the New European Driving Cycle (NEDC) used for car certification in Europe, and the EPA Federal Test Procedure (FTP-75 and its successors) used for certification in the United States.

Different driving cycles are used for different vehicle types since often the regular usage can vary greatly. Bus engines for example are typically tested on a driving cycle containing lots of starts and stops, mimicking a bus going between stops, while an engine for a truck designed to carry heavy loads is tested on a driving cycle with longer segments of rather constant speed representing long range highway/freeway driving.

This thesis however is mainly concerned with driving cycles for cars, but the work may be extended to other vehicle types as well. Driving cycles for cars have differing characteristics depending on what type of driving they aim to represent. For example, urban driving contains lots of stops and relatively low speeds while driving on rural roads or highways means very few stops and generally higher speeds. There are driving cycles that have been designed to cover urban and



(a) The FTP-75 driving cycle, assembled from real-world driving data. The cycle contains a partial repetition (at ≈ 23 minutes) in order to include a hot start sequence.

(b) The NEDC, a purely theoretical driving cycle put together by four repeated urban driving cycles, followed by extra-urban driving.

***Figure 1.1:** A comparison between the NEDC and FTP-75 driving cycles.*

highway driving separately and there are also ones where both types of driving are present.

Many driving cycles (including FTP-75, see Figure 1.1a) are created by recording the speed of vehicles driving specific routes and then using averages and piecing together different parts to create one driving cycle that's deemed representative of a car's regular usage. Since this type of method is based on real world data it produces driving cycles with speed variation close to what is observed from human drivers in real traffic and road situations. The reliance on real world data however means that the driving cycle is more favourable to the type(s) of car that was driven during the test. The FTP-75 driving cycle consists of the UDDS (Urban Dynamometer Driving Schedule) driving cycle with an added repetition of the first 505 s (Environmental Protection Agency (EPA), 2017).

Some other driving cycles (for example NEDC, Figure 1.1b) are created in theory, only using real driving as a guideline in its creation. The NEDC is made up of four repeated UDC (Urban Driving Cycle) followed by an EUDC (Extra-Urban Driving Cycle) (United Nations, 2013). The result is a driving cycle that has sections of acceleration/deceleration with fixed gear changes, and sections of constant speed. This type of driving cycle is not very representative of real driving and the NEDC has been heavily criticized for producing fuel consumption and emission figures that are far from what can be achieved in real life (Mock et al., 2012). The predictable shape and repetition of the driving cycle has also meant that car manufacturers can optimize their powertrains to get better results from the specific driving cycle while not actually producing better results in real world driving, something referred to as “cycle beating” (Kågesson, 1998).

1.1.1 Related Research

With regards to the problems faced by using predetermined driving cycles, research has been conducted on generating driving cycles with characteristics similar to real driving. The goal with these methods is to eliminate the need for predetermined driving cycles (such as FTP-75 or NEDC) and instead be able to generate driving cycles that can be used for engine testing and emission evaluation (Nyberg et al., 2016; Souffran et al., 2011).

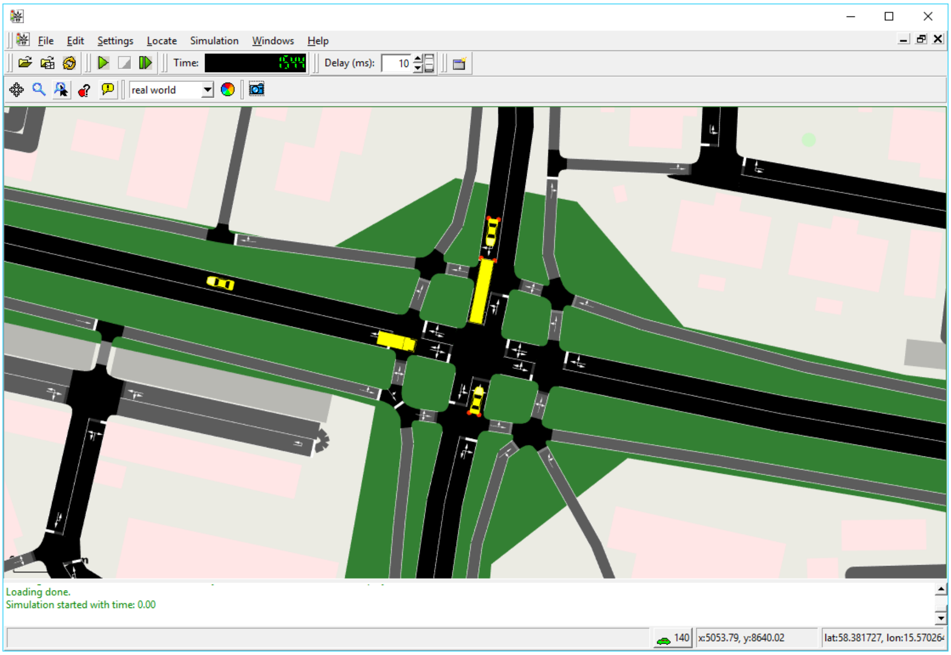
One approach to generate driving cycles is to make use of a stochastic process that given data recorded from many test drives is able to construct a driving cycle that represents real driving. Some papers (Gong et al., 2011; Nyberg et al., 2016; Souffran et al., 2011) make use of Markov models to generate these new driving cycles. The approach presented by Nyberg et al. (2016) has the added possibility to generate equivalent driving cycles that provide the same excitation as a given standardized driving cycle, e.g. NEDC.

Another approach with the same goal is to use real world driving data to calibrate engines for a specific set of driving missions carried out in real life (Tong et al., 1999). This approach can be used for passenger cars as well as other types of vehicles. A set of driving missions can be defined which represent the most likely driving that the vehicle will be exposed to during its life time. The drawback with this approach is that recording driving data takes a lot of time and is expensive. There is also very little control over test parameters such as traffic when driving in real life. If vehicle routes are simulated inside a traffic simulation where parameters can be set for traffic, vehicles and driver behaviour this method could prove more viable. This thesis will make use of a traffic simulation to simulate driving missions.

1.2 Microscopic Traffic Simulation

Traffic simulation is an area of research that is concerned with the understanding of phenomena such as traffic flow, traffic jams, and vehicle interaction in traffic through the use of simulation. Simulating these phenomena accurately can give valuable information about efficient road construction to minimize traffic jams and improved event planning in large cities when roads are closed off (Krauß, 1998).

Traffic simulation is usually divided into two groups: Macroscopic- and microscopic simulation. Macroscopic simulation models are aimed at describing traffic in large road networks where the main point of interest is traffic flow and how it evolves over time. Macroscopic models are thus based on flow equations representing single vehicles as particles within a fluid moving along the roads. Microscopic models on the other hand describe each vehicle and its behaviour on the road individually and a larger traffic context is created by having all vehicles interact according to the given model. The models generally consist of differential equations with parameters based on empirical studies of human driver behaviour. Microscopic models generally give more detailed results than macroscopic simulation but may have problems simulating larger road networks (Brackstone and



Loading done.
Simulation started with time: 0.00

x:5053.79, y:8640.02
lat:58.381727, lon:15.57026

Figure 1.2: *Screenshot of the SUMO graphical interface showing an intersection with several vehicles present.*

McDonald, 1999).

1.2.1 Car-following Models

Since it's the interaction between vehicles on the road that serves as the source of different traffic phenomena, a family of models exist based on the concept of car-following. Car-following models model the relationship between one vehicle and the vehicle in front of it based on the distance and speed difference between the two. Usually a car-following model considers the vehicle and the driver as a single unit where attributes such as reaction time, acceleration and braking ability consider both the vehicle itself and the driver (Brackstone and McDonald, 1999).

1.2.2 The SUMO Traffic Simulation

SUMO is an open source traffic simulation software package which includes a simulation environment with accompanying graphical interface, a network importer and editor allowing maps from OpenStreetMap to be imported and used in the simulation, among other tools. The simulation models in SUMO are based on the car-following principle.

1.2.3 Related Research

Numerous studies have been undertaken in search of modelling and understanding human driving behaviour. The entire field of car-following models aims to replicate interactions between human drivers in traffic on a vehicle by vehicle basis, see Brackstone and McDonald (1999) for an overview. Some notable research on car-following are Chandler et al. (1958) which includes one of the first studies on traffic interactions, and Gipps (1981) in which a car-following model is presented that serves as the base for many modern simulation models.

Relating specifically to human driving Treiber et al. (2006) presents their so called Human Driver Model built upon a previous driver model presented in Treiber et al. (2000). In order to more accurately simulate a human driver this model includes a finite reaction time, imperfect estimation capabilities of the distance between vehicles, and anticipation several vehicles ahead in traffic. This thesis will take inspiration from some of the car-following research though the work itself will not be based on car-following.

1.3 Problem Statement

With further demands on vehicle emissions and performance it's becoming increasingly important for vehicle manufacturers to have quick and reliable ways to evaluate their vehicles for specific customers' needs. By using a microscopic traffic simulation tool to simulate driving missions in real road networks it should be possible to repeat the same missions under varying circumstances without having to drive there in real life, thereby saving time and resources. If speed profiles from a traffic simulation could be made representative of real-world driving this approach would be valid. At the moment the traffic microsimulation tool SUMO does not generate speed profiles with the required amount of detail.

The purpose with this thesis is to create models that together can describe how human drivers adjust their speed with respect to common road features such as curves and intersections as well as a model describing the general speed variation when driving on straight roads. The models will be used together with the SUMO traffic simulation tool to simulate driving missions on real road networks which can then be compared and evaluated against actual driving data from the same roads.

1.4 Delimitations

The goal with this thesis is not to develop a new traffic simulation model but merely to add new features to an existing model. The new functions will be limited to adjusting the speed (and acceleration) of vehicles in different ways mostly related to road features such as curves and intersections and not vehicle interactions. Furthermore these additions will have to be made to a model that's already implemented in SUMO.

The decision has been made to utilize TraCI (Traffic Control Interface) in SUMO as the injection point of the new models. Through TraCI it's possible to control a SUMO simulation via MATLAB or Python for example, making it faster and easier to iterate on code without having to recompile SUMO. Since this work will utilize MATLAB for data processing and subsequent modelling, using MATLAB with TraCI will make the transition between modelling and implementation simple. This way MATLAB's computational flexibility can also be used in the implementation of the models. The main drawback is that the work will be bound by what functions are available through TraCI, which is more limited than programming directly in the SUMO codebase using C++. SUMO can be downloaded from `http://sumo.dlr.de/wiki/Downloads`, and the TraCI implementation for MATLAB used here called TraCI4Matlab is available at `http://mathworks.com/matlabcentral/fileexchange/44805-traci4matlab`

This work does not aim to create a general model applicable to all different types of roads. The modelling will be based on data from one or two routes which will consist mainly of rural roads and highways, so city driving is unaccounted for, though city driving should mainly be governed by car-following interactions and not road features which reduces the impact any of the models would have.

1.5 Thesis Outline

In Chapter 2 the background for the work is presented highlighting differences between simulation and measurements where improvements will be made.

Chapter 3 shows how a route was decided, and what data was recorded with the instrumented test vehicle. The data that will be used to for each model is shown along with the selection procedure for each of the data sets.

Chapter 4 contains the modelling procedures for all models. Model structures, estimation procedures and model outputs are shown.

Chapter 5 presents principally how the models are implemented to work with SUMO and also how simulation performance is impacted by the implemented behavioural models.

The results are shown in Chapter 6 in relation to the measurements, as well as simulation results before the implemented modifications.

In Chapter 7 the conclusions are presented followed by possible future work that can be carried out to improve the results.

2
Background

In this chapter the focus of the thesis will be presented in relation to the aspects of the SUMO behavioural model that will need to be improved in order to more accurately reflect human driving behaviour. Observations from human driving will be used as the basis for these changes.

2.1 Car-following Models in SUMO

The vehicle behaviour in SUMO is defined by an underlying car-following model described by the set of equations in (2.1) - (2.4). A lane changing algorithm is also implemented but is not described here. This car-following model which was presented by Stefan Krauss in Krauß (1998) can be seen as a variant of the car-following model proposed by P. G. Gipps in Gipps (1981) that guarantees the safety of vehicles on the road by including a safety distance between them.

$$v_{\mathrm{safe}}(t) = v_{\mathrm{l}}(t) + \frac{g(t) - g_{\mathrm{des}}(t)}{\tau + \tau_{\mathrm{b}}} \tag{2.1}$$

$$v_{\mathrm{des}}(t) = \min\left[v_{\mathrm{max}},\ v(t) + a(v)\Delta t,\ v_{\mathrm{safe}}(t)\right] \tag{2.2}$$

$$v(t + \Delta t) = \max\left[0,\ v_{\mathrm{des}}(t) - \eta\right] \tag{2.3}$$

$$x(t + \Delta t) = x(t) + v\Delta t \tag{2.4}$$

where the default model has the following parameters

$$g_{\mathrm{des}} = \tau v_{\mathrm{l}}(t) \tag{2.5}$$

$$\tau_{\mathrm{b}} = \frac{v_{\mathrm{l}}(t) + v(t)}{2b} \tag{2.6}$$

with the definitions in Table 2.1.

Table 2.1: *Variable definitions for the default car-following model in* SUMO

Notation	Definition	Unit
$v(t)$	Current speed of the vehicle	m/s
$v_l(t)$	Current speed of the leading vehicle	m/s
$v_{\text{safe}}(t)$	Current speed that guarantees that no collision will occur with the leading vehicle	
$v_{\text{des}}(t)$	The vehicle's current desired speed	m/s
v_{max}	The vehicle's maximum speed depending on the speed limit	m/s
$a(v)$	The vehicle's acceleration capability given the speed	m/s^2
b	Typical deceleration used by driver/vehicle	m/s^2
$g(t)$	Current distance between the current vehicle and its leader	m
$g_{\text{des}}(t)$	Current desired distance for the current vehicle to the leader	m
τ	Reaction time of the driver-vehicle unit	s
τ_b	Time scale	s
Δt	Length of a simulation time step	s
η	Random perturbation (> 0) on the speed	m/s
$x(t)$	The vehicle's current (one dimensional) position on the road	m

When adjusting the speed of a vehicle in the simulation the vehicle's acceleration parameter as well as car-following behaviour should still be adhered to. If the vehicle's speed is only adjusted with v_{max}, car-following and acceleration will work as normal, see equation (2.2). This is the way speed adjustment works via TraCI and is how all speed adjustments will be applied. The acceleration parameter $a(v)$ can be adjusted without impacting the car-following behaviour.

Because of how changing the speed and acceleration works, any properties of the model such as the car-following behaviour and safety distance to vehicles etc. will remain intact.

2.2 Identifying Limitations in SUMO

This section details some areas in SUMO where models are needed in order to generate realistic human driving with respect to speed keeping. Figure 2.1 shows the same driving mission carried out in real life as well as in SUMO, with areas of special interest shown in greater detail in Figure 2.2. For more information about the measurements, see Chapter 3.

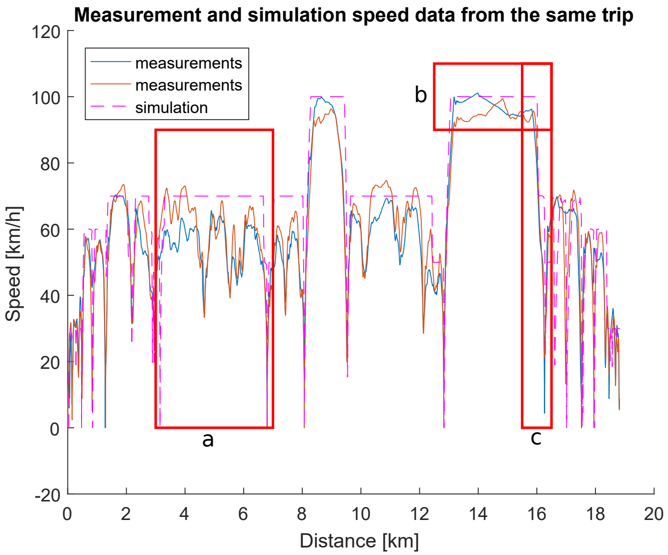


Figure 2.1: A driving mission carried out in both real life and simulation. The rectangles marked a, b, c *correspond to the respective subfigures in Figure 2.2.*

2.2.1 Adjusting Speed to Curvature

In SUMO there is no system in place that changes vehicle speeds depending on road geometry. In Figure 2.2a it's shown that there exists a relationship between road curvature and driving speed for human drivers. In order to safely pass any curves, drivers must adjust their driving speed according to what they perceive as safe. The speed adjustment can be seen to vary between drivers and the amount of adjustment could also be affected by parameters such as visibility around the curve, the road surface and weather conditions. The speed-curvature relationship will be investigated to see what parameters contribute to the speed adjustment and if a model can be constructed that can replicate the speed well.

2.2.2 Speed Variation

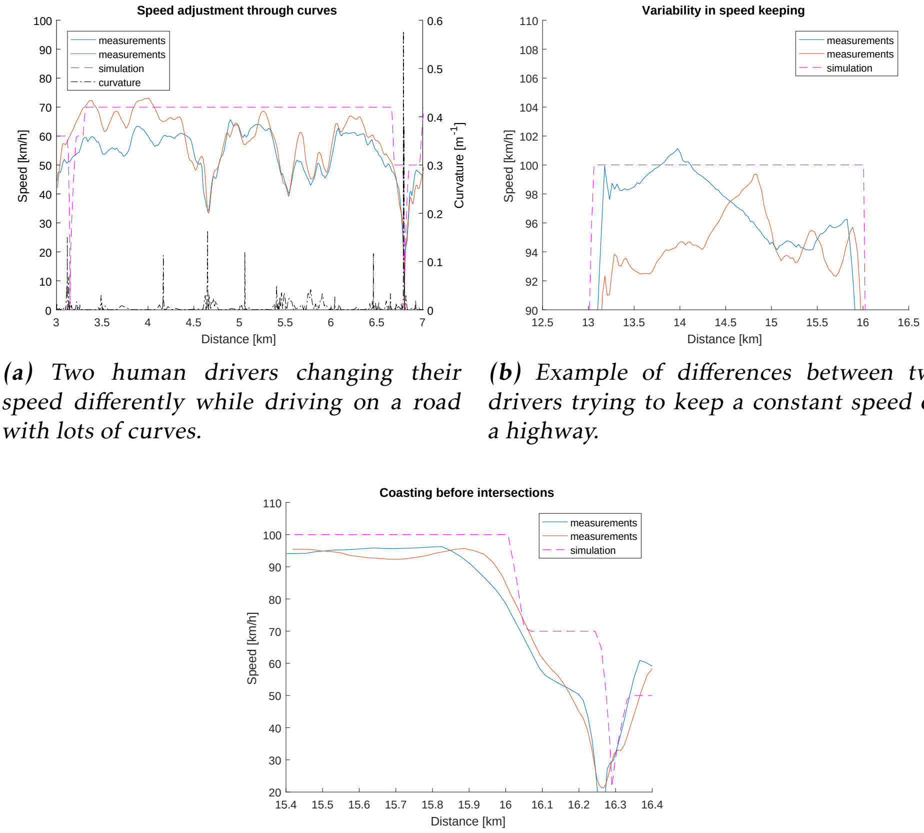
When driving on a relatively straight road with a constant speed limit and with no traffic interactions, human drivers exhibit a speed variation of varying amplitude as seen in Figure 2.2b. Realistically this speed variation could be dependent on the type of driver and on the type of road. This thesis will aim to find a representation of this speed variation with the use of randomly generated noise.

2.2.3 Anticipation

Human drivers will often avoid hard braking and instead let their vehicle coast for a while or only brake lightly if they are able to anticipate obstacles on the road ahead of them in a sufficient amount of time. This can be observed by looking at how drivers behave when they're approaching an intersection (in this case a roundabout) from a rural highway with a speed limit of 100 km/h, see Figure 2.2c. It's apparent that different drivers utilize coasting to different degrees. Anticipation also affects how drivers adjust to speed limits and curves. Coasting will be modelled by considering an anticipation distance for drivers which is dependent on the speed and thus letting them begin decelerating long before intersections or other obstacles when driving at high speeds.

2.2.4 Acceleration

Acceleration in SUMO only includes one parameter value which means that the acceleration when vehicles are not bound by car-following is constant. In reality the acceleration changes depending on which gear the driver is using and at which speed the vehicle is going. A gear changing algorithm will have to be implemented if the simulated accelerations are going to be representative.



(a) *Two human drivers changing their speed differently while driving on a road with lots of curves.*

(b) *Example of differences between two drivers trying to keep a constant speed on a highway.*

(c) *An example of drivers using coasting to varying degrees before entering an intersection.*

Figure 2.2: *Three different areas where changes will be made in the simulation so that it more closely resembles a human's driving.*

3 Data Acquisition and Processing

This chapter explains the data acquisition that will be used for the modelling procedure later on in the thesis. Shortcomings in the data will be discussed and data used for estimation and validation will be clearly shown before moving on to the modelling in the next chapter.

3.1 Driving Data

For the purpose of this thesis, data has been recorded from several runs over a specific route with an instrumented vehicle. Different drivers were used for every test drive in order to get more varied data and to avoid drivers getting accustomed to the specific route. In total the route was driven by 15 different people who volunteered to participate without compensation. 12 of the drivers were male, 3 female and most were in the ages between 24-35 years old with a few being older.

Measurements carried out on a different route as part of a research project a few years prior were also made available for this thesis. The measurements were carried out with the same car equipped with the same instruments, so the data was comparable. While the two routes contain parts of the same roads they were deemed sufficiently different to be used as estimation and validation data respectively.

3.1.1 Route for Measurements

When deciding a route to drive for recording data there were a few points to consider

1. The route should contain a good amount of curves and intersections/traffic lights to get sufficient data for curvature speed modelling, anticipation and

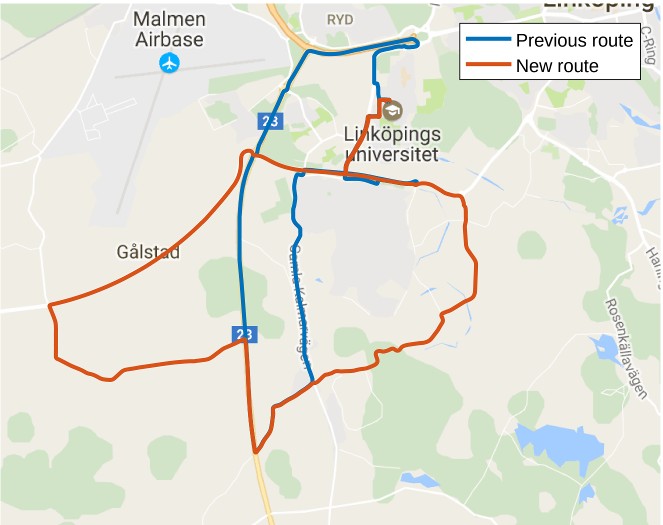


Figure 3.1: The route chosen for measurements (red) and the previously recorded route (blue) in the vicinity of Linköping University. Map image taken from Google Maps.

braking.

2. The route should include highway driving so that stationary speed variation can be estimated.
3. The route should preferably avoid roads that had been driven on in a previous research project.
4. The route should not take longer than 30 min to drive in order to make it easier to get people to volunteer.

These requirements led to the route which can be seen in Figure 3.1 where the previously driven route is also shown. The route is 19 km long and includes six traffic lights, around 30 significant curves and approximately 5 km highway driving. The route is in general very flat so any speed fluctuations due to slopes can reasonably be neglected.

3.1.2 Available Measurements

The instrumented vehicle is equipped with a CAN-bus which records data from the car's on-board computer, a GPS, and an IMU. All relevant signals from these sources can be found in Table 3.1.

Some problems were encountered while recording data where measurements from the GPS and IMU sources would occasionally be missing. Additionally the GPS would sometimes have problems finding the correct position, rendering its

Table 3.1: *Table describing the measured signals from the instrumented vehicle that were relevant for the work in this thesis.*

Measurement unit	Measurements
CAN-bus	Time Angular velocities of all wheels Throttle pedal angle Clutch Brake
GPS	Time Latitude Longitude Speed
IMU	Time x, y, z accelerations φ, θ, ψ angles

measurements uncertain for some period. Thankfully the data from the CAN-bus was always available without any problems.

Since this thesis focuses on analyzing speed profiles the speed was the most important measurement and while the speed was measured by the GPS the data from the CAN-bus was seen as a better source for two reasons: Firstly the above mentioned unreliability of the GPS rendering parts or the entirety of the recording useless. Secondly the speed from the GPS is also filtered leading to some inaccuracies when the car stops. Meanwhile the CAN wheel angular velocity data doesn't suffer from any of these problems, and by taking the mean angular velocity of all four wheels and multiplying with the wheel radius the car's speed could be obtained in a reliable way throughout all test drives, see equation (3.1) where F, R, l and r denote front, rear, left and right respectively.

$$v(t) = r_{\text{wheel}} \frac{\omega_{\text{F,l}}(t) + \omega_{\text{F,r}}(t) + \omega_{\text{R,l}}(t) + \omega_{\text{R,r}}(t)}{4} \tag{3.1}$$

By using dead reckoning from the wheel speeds it was possible to get a sufficiently precise measurement on the driven distance. Using a stop sign along the route as a syncing point for all runs the largest difference in estimated distance between runs was approximately 20 m which was deemed to be within the margin of error for the 19 km route.

The IMU was not fully aligned with the car so measured accelerations in (x, y, z) had to be rotated according to the measured angles roll, pitch and yaw (denoted (φ, θ, ψ)) to give the acceleration in the proper directions, see Figure 3.2 for a visualization of the coordinates. Of main interest here is the acceleration in x which can be used to judge the braking action, and also the acceleration in y which might be useful when analyzing driving through curves.

The signals for the throttle, clutch and brake pedals were judged to potentially be useful when determining braking and acceleration behaviour of drivers. Both

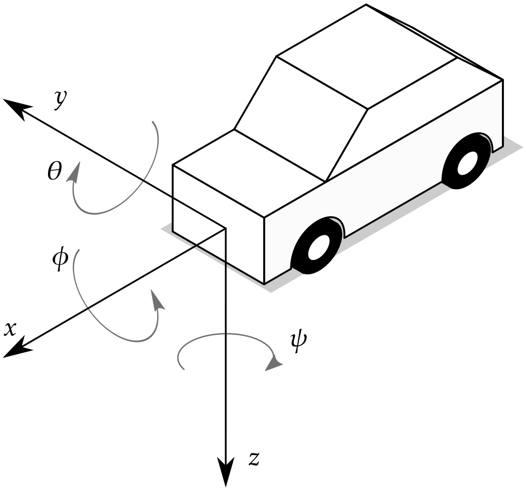


Figure 3.2: The coordinate system of the car. (*φ*, *θ*, *ψ*) *represent roll, pitch and yaw respectively.*

brake and clutch can be applied to different degrees and in that way affect the deceleration of the vehicle differently, but the clutch and brake signals were only binary i.e. on or off, so their usefulness were limited.

Something else to note is that the three measurement units all have their own internal sampling rate meaning that the data needed to be resampled at a common rate in order to use them together for modelling purposes.

3.2 Map Data

Since SUMO has the ability to import map data from OpenStreetMap and simulate traffic on those road networks, that same data was also used for the parts of modelling that require knowledge of road features, e.g. curvature, speed limits and intersections. While a network imported like this was found to generally work quite well there were instances where speed limits were incorrect or intersections were not connected properly to incoming roads. These things had to be manually edited to agree with real-life.

3.2.1 Improving Road Geometry with Splines

Definition 3.1. For any curve $C(s)$ the curvature $\kappa(s)$ is defined as

$$\kappa(s) = \frac{1}{r(s)} = \left\| \frac{d\mathbf{T}(s)}{ds} \right\| \tag{3.2}$$

where $r(s)$ is the radius of curvature and $\mathbf{T}(s)$ is the unit tangent vector to the curve for a given s.

Using Definition 3.1 the curvature of a plane curve given in the Cartesian coordinates $(x(s), y(s))$ is given by equation (3.3), where $'$ is used to denote the derivative $\frac{d}{ds}$. This expression has been used to calculate the curvature of roads.

$$\kappa = \frac{|x'y'' - x''y'|}{(x'^2 + y'^2)^{\frac{3}{2}}} \tag{3.3}$$

The geometry of the imported road network was generally accurate, but since the road geometry is only C^0 continuous (points are connected with no continuous derivatives) calculating the curvature according to equation (3.3) with discrete differentiation led to discontinuities and curvature values that were not realistic. Additionally the large variance in length between some road segments led to inaccurate curvature at the connecting points.

To alleviate this problem the road geometry was interpolated using cubic spline interpolation with equal spacing along the road. The geometry used by the spline interpolation was modified with corners being cut so that the spline wouldn't deviate too much from the original geometry at sharp corners, see Figure 3.3a for an example. The resulting spline $S(d)$ describes the route coordinates $[x, y]$ as a function of the route distance d. It has the representation seen in equation (3.4).

$$\begin{bmatrix} x \\ y \end{bmatrix} = S(d) = \begin{cases} \begin{bmatrix} P_{x,1}(d) \\ P_{y,1}(d) \end{bmatrix}, \; d_0 \leq d \leq d_1 \\ \vdots \\ \begin{bmatrix} P_{x,n}(d) \\ P_{y,n}(d) \end{bmatrix}, \; d_{n-1} \leq d \leq d_n \end{cases} \tag{3.4}$$

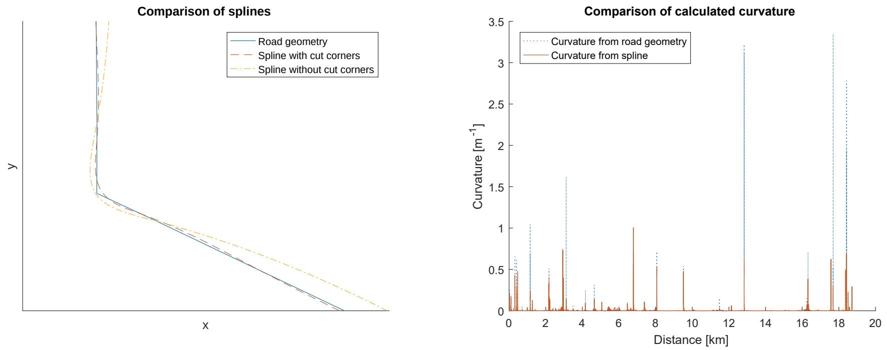
with the cubic polynomials $P_{x,k}(d)$, $P_{y,k}(d)$ on the form

$$P_k(d) = p_{k,1}d^3 + p_{k,2}d^2 + p_{k,3}d + p_{k,4} \tag{3.5}$$

where the parameters $p_{k,1}, p_{k,2}, p_{k,3}, p_{k,4}$ are determined with spline interpolation and where k denotes each separate geometry segment.

The cubic spline is C^2 continuous and representing the geometry with a cubic spline means that the polynomials describing each segment can be analytically differentiated when calculating the curvature which is more reliable than using discrete derivatives. Figure 3.3b shows the difference between curvature calculated from road geometry and curvature calculated from a spline. The road geometry data used for calculating curvature has had some processing where segments shorter than 0.1 m were removed or otherwise the highest calculated curvature was in the region of 10^4 which would have been unusable for any modelling purpose.

Something to take note of is that the calculated curvature is still not representative of the real road curvature since a curvature of 1 equals a curve radius of 1 m. Realistically a curvature of around 0.2 (curve radius of 5 m) would be about the highest expected curvature on any road. The limitation here is most



(a) *Example of road geometry interpolated with splines where the corners are cut at $^1/_5$th of the segment length, and where corners are not cut. The spline with cut corners more closely resembles the actual road geometry.*

(b) *Comparison between curvature calculated from road geometry of the entire route with some processing, and curvature calculated from an interpolating spline based on the same geometry.*

likely that the map data from OpenStreetMap isn't sufficiently detailed to get accurate road curvature from. Modelling will have to be carried out on this data regardless.

3.3 Preparation of Data

In order to easily compare and use the recorded data for modelling, every test drive was resampled and adjusted so that all of them would cover the same distance and be sampled equally over this distance (all tests were sampled with 1 m resolution). The resampling was done over distance so that the driving speed and the time standing still at traffic lights and intersections wouldn't have an effect on the amount of samples. With this data preparation in place it was then simple to calculate a mean speed profile consisting of the mean speed at every distance sample. Doing this results in a speed profile without any individual driver behaviour while random disturbances are also reduced so that it's mostly the speed adjustments for curves and intersections that is visible. Some considerations were taken before calculating the mean. Speed profiles with extraordinary disturbances such as the occasional tractor on the road were removed because these could not represent normal driving. Ultimately 9 speed profiles (out of 15) were used to calculate the mean speed profile. The mean speed profile will mainly be used for estimating speed through curves.

The individual resampled speed profiles will still be used when modelling the anticipation distance and the braking behaviour before stops at intersections and traffic lights.

3.3.1 Data for Curvature Speed Model

For identifying a curvature speed model, points of local maxima in curvature κ_{max} along the route were matched to local minima, v_{min}, in the mean speed profile. Since there is only one set of input data (curvature), estimation on all speed profiles would result in the same model as estimation on the mean speed profile. Data corresponding to intersections was removed because the speed is affected by other factors than curvature there. Figure 3.3 shows part of the selected data in speed and curvature.

Only using data at the extreme points in both data series means that modelling will only be carried out on the most relevant data. The extreme points were selected with criteria for prominence and distance, i.e. selected points had to have a prominence over surrounding points higher than 0.1 m/s and be at least 60 m apart. As an effort to counter the large variation in peak curvature, the integral of curvature within 50 m around each peak position was calculated and used for modelling as well. The integral was calculated with the trapezoidal method:

$$\kappa_{\text{int}}(p) = \frac{1}{2}\sum_{n=1}^{2L}(\kappa(p-L+n-1)+\kappa(p-L+n)) \tag{3.6}$$

where p is the peak position (the distance in the route where the peak was found), and L is the distance before and after the peak to integrate over, in this case $L = 50$.

In the end only 30 points of speed and curvature were selected, which isn't a significant amount so precautions will have to be made during the modelling procedure.

3.3.2 Data for Braking Distance Modelling

For modelling the braking distance d_{brake}, local minima in speed were identified as described above but this time intersections were included in the data as well. Then a point prior to the minimum was found corresponding to where the braking starts by analyzing the car's acceleration in the x direction. These points are denoted v_{start} and v_{end} respectively. The braking distance was then calculated as the difference in position between these two points. Stretches of road with traffic lights were removed from the data because of the unpredictability of the traffic light state leading to braking over shorter distances than under normal conditions.

For this model all different speed profiles could be used since the model variables v_{start}, v_{end} and d_{brake} aren't the same for any tests. From all tests then a selection of approximately 500 samples was obtained which is significantly better than the amount of data for the curve speed model. An example of the data selection is shown in Figure 3.4 where the mean speed profile has only been used so a comparison can easily be made to Figure 3.3 and 3.5 however, all tests are used as data sources.

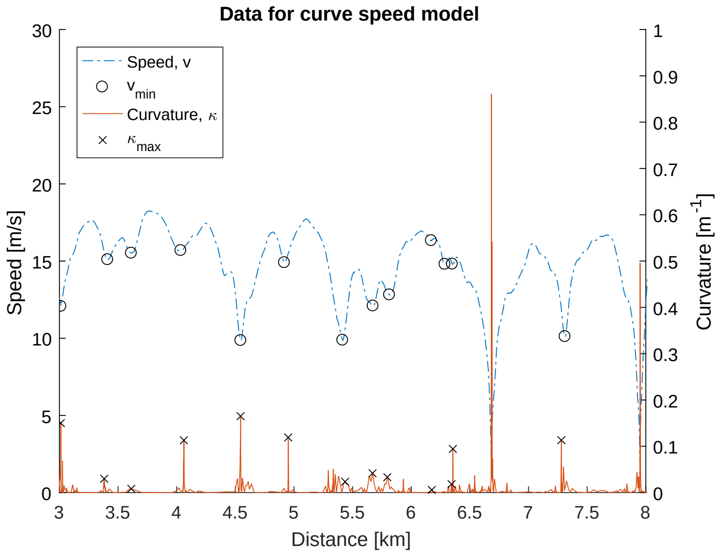


Figure 3.3: *Part of the data that will be used for curve speed modelling. Data from intersections has not been selected.*

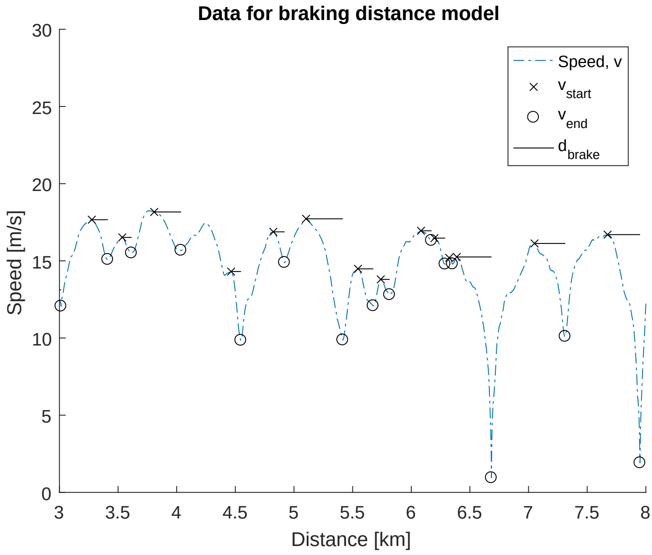


Figure 3.4: *Visualizing the data that will be used for anticipation/braking distance modelling. Data from intersections is used here.*

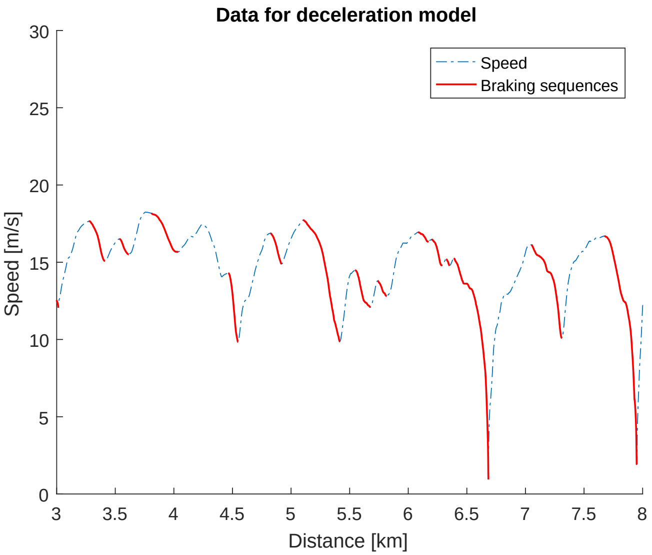


Figure 3.5: *Visualizing the data that will be used for braking sequence modelling.*

3.3.3 Data for Braking Sequence Modelling

Braking sequences again make use of found minima in speed, here denoted v_{end}, along with the identified speed where braking starts, v_{start}. In order to model the varying behaviour during an entire braking sequence the model estimation must be carried out on series of data instead of just isolated points, where each series consists of all speed and distance values taken between v_{start} and v_{end}.

Braking sequences from all tests could be used here as well as data from traffic lights and intersections to ultimately get a large selection of data consisting of about 350 different braking sequences with varying amounts of samples. The modelling will most likely have to be separated into braking for curves, and braking for intersections and traffic lights since the behaviour is different when braking to a stop, see Chapter 4.

Figure 3.5 shows the identified braking sequences over the same segment on the route as Figure 3.3 and 3.4, again the mean speed profile is used here for ease of comparison but it's not used as a data source.

3.3.4 Data for Speed Variation Modelling

The speed variation when driving without external influences will be modelled from data corresponding to free driving on highways without any traffic interactions. This is done so that hopefully no speed adjustments will occur due to external sources and only the driver's internal speed variation will show. Data was taken from the 9 test runs that make up the mean speed profile and the relevant data was selected by hand so that the criteria could be fulfilled. In Figure 3.6 the data from the second part of highway driving is shown.

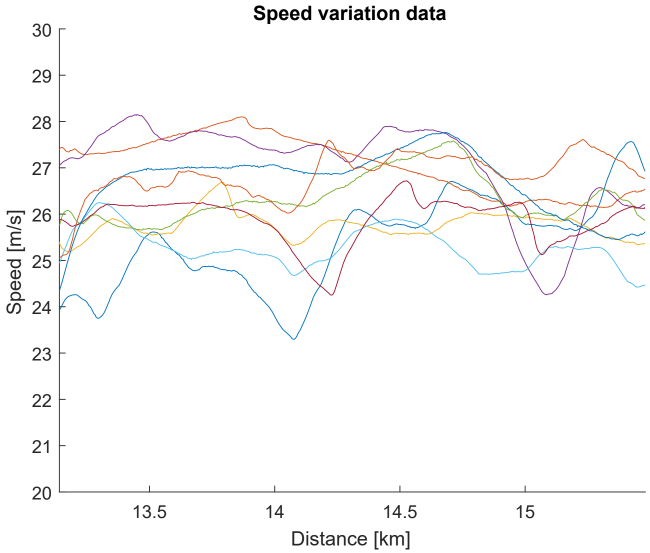


***Figure 3.6:** The data that will be used for speed variation modelling. The speed limit here is* 100 km/h *(*27.78 m/s*).*

4

Modelling

This chapter presents the modelling procedures for the different areas presented in Chapter 2 using the data laid out in Chapter 3. Throughout the chapter the coefficient of determination, R^2 will be used as a measure of fit for the models, i.e. how well the model fits to the data, and root mean square error (RMSE) will be used to assess the models' performance, i.e. the deviation between model values and measurements, see Definition 4.1 and 4.2 below, taken from Montgomery and Runger (2003, chapter 11).

Definition 4.1. A vector $\mathbf{y}$ has N values and for each value in $\mathbf{y}$ there exists a modelled or predicted value in $\hat{\mathbf{y}}$. The coefficient of determination, R^2 is then defined as

$$R^2 \equiv 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{4.1}$$

where $\bar{y}$ is the mean of $\mathbf{y}$,

$$\bar{y} \equiv \frac{1}{N}\sum_{i=1}^{N} y_i$$

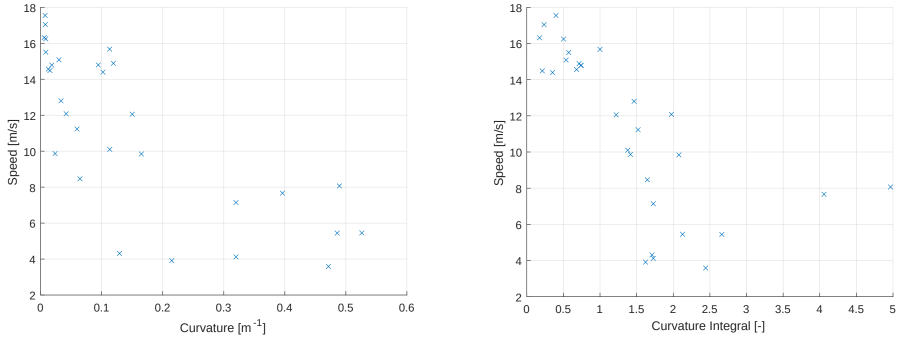
Definition 4.2. A vector $\mathbf{y}$ has N values and for each value in $\mathbf{y}$ there exists a modelled or predicted value in $\hat{\mathbf{y}}$. The root mean square error (RMSE) between the two is defined as

$$\text{RMSE} \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.2}$$

The root mean square error has the same unit as the elements in $\mathbf{y}$ and $\hat{\mathbf{y}}$.

Table 4.1: Notation for the curve speed model. Boldface is used to indicate vectors.

Notation	Explanation
$\kappa, \boldsymbol{\kappa}$	Curvature
$\kappa_{\text{int}}, \boldsymbol{\kappa}_{\text{int}}$	Integral of curvature
$v_{\text{min}}, \mathbf{v}_{\text{min}}$	Speed at curve



(a) Speed through curves $\mathbf{v}_{min}$ plotted against the curvature $\boldsymbol{\kappa}$.

(b) Speed through curves $\mathbf{v}_{min}$ plotted against the integral of curvature $\boldsymbol{\kappa}_{int}$ around the max curvature.

Figure 4.1: A downward trend in speed is visible for both increasing curvature and increasing curvature integral.

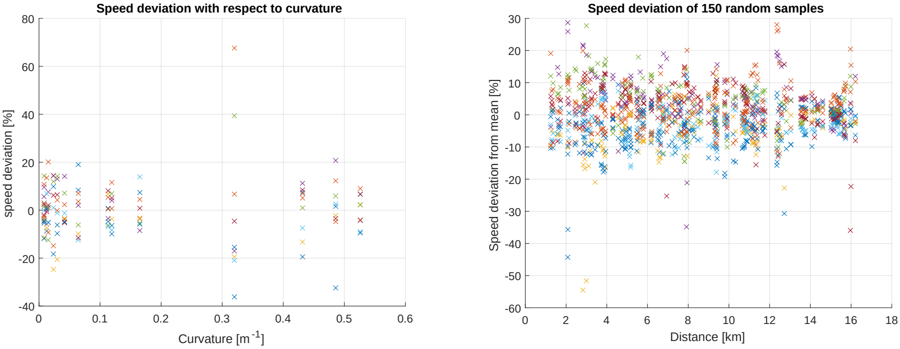
4.1 Speed Adjustment Through Curves

This section describes how the model for curve speed adjustment is derived. Data is first analyzed to expose any relationships in the variables and then the model expression is presented. Lastly parameters are estimated and the model performance is reviewed. The notation used for this model can be found in Table 4.1.

4.1.1 Analyzing the Data

Before constructing the model, the data must first be analyzed to see what shape a possible model would take. A relationship between curvature and speed must be established and Figure 4.1 shows that lower speeds are generally associated with both an increase in curvature and curvature integral, as to be expected. As described in Section 3.3, the data for this model is taken from the mean speed profile. The lack of data points makes it hard to draw definitive conclusions about the function itself but the trends are reasonably clear to at least state that there is a general relationship.

Something to consider is how different drivers react to curvature. Is a person who drives fast in general more likely to also drive faster through curves, or is



*(**a**) Proportional speed deviations from the mean speed profile through curves by 9 drivers. There is no visible correlation between curvature and the speed deviation.*

*(**b**) Proportional speed deviations from the mean speed profile by 9 drivers over 150 random samples.*

***Figure 4.2:** Speed deviation through curves compared with speed deviation of random samples.*

the amount of curve adjustment separate from how fast one drives on straight roads? To determine if the speed through curves has comparable characteristics to normal driving the proportional speed deviation from the mean is analyzed. By looking at the proportional deviation instead of the absolute deviation the variation should likely be in the same region for high and low speeds. As a way to assess the speed deviation for general driving, 150 random speed samples were taken from the route at positions in the range from 1.2 km to 16.5 km to avoid any traffic lights which interfere with the speed distribution between drives. Figure 4.2 shows the deviations side by side, and there isn't any visible difference between them. The data used here is from the 9 test drives that make up the mean speed profile.

***Table 4.2:** Difference in mean μ and standard deviation σ of the proportional speed deviation between the curve speeds and the random speed samples of the four different drivers seen in Figure 4.3, clockwise from upper left.*

$\mu_{\text{curve}} - \mu_{\text{rand}}$ [%]	$\sigma_{\text{curve}} - \sigma_{\text{rand}}$ [%]
0.655	1.84
−0.310	2.34
−0.552	−1.39
0.157	3.55

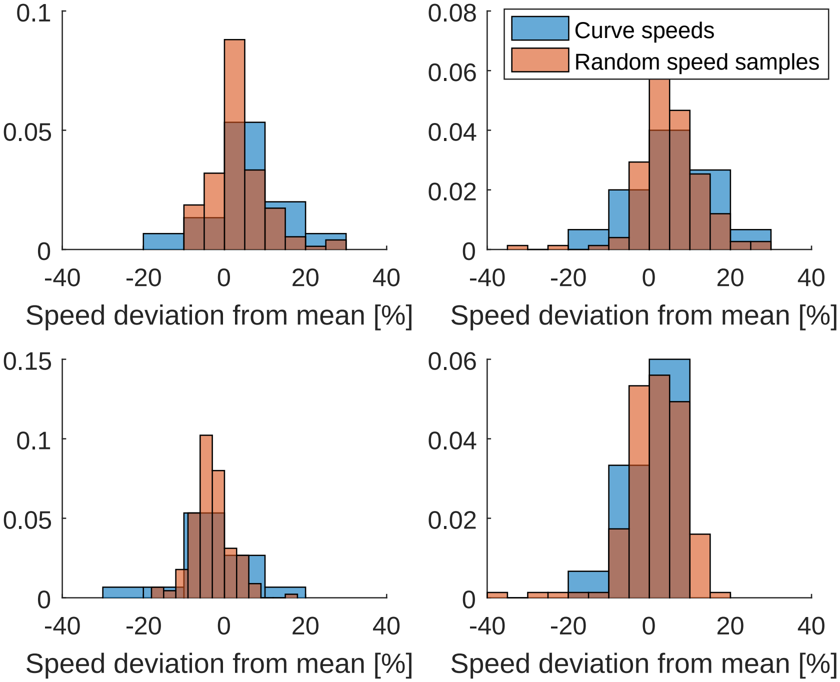


Figure 4.3: *Comparison between the distribution of speed deviations when driving through curves and speed deviation for random speed samples, for four separate drivers. The histograms are normalized so that the area of each bar is the relative number of observations. The sum of the bar areas is 1.*

Looking at separate drivers, Table 4.2 and Figure 4.3 show that the distribution of speed deviations are similar for curves and random speed samples with regards to both the bias and standard deviation. While it's hard to draw conclusions from such a small data set there is nothing that clearly points towards a different behaviour in curves specifically. With no evidence supporting a different speed deviation in curves it will be assumed that the proportional speed deviation is the same for curves as it is for general driving.

4.1.2 Constructing the Model

Figure 4.1 shows a decreasing speed for increasing curvature and curvature integral as one would expect. In mathematical terms this would imply a function that's strictly decreasing for any $\kappa, \kappa_{\text{int}} \geq 0$. The speed through a curve must also be positive to make driving through it possible. The model properties can be summarized as follows

$$v_{\min} = f_{\text{curve}}(\kappa, \kappa_{\text{int}}) \tag{4.3}$$

subject to

$$\kappa, \kappa_{\text{int}} \geq 0 \tag{4.4}$$

$$f_{\text{curve}}(\kappa, \kappa_{\text{int}}) > 0 \tag{4.5}$$

$$\frac{\partial f_{\text{curve}}(\kappa, \kappa_{\text{int}})}{\partial \kappa} < 0 \tag{4.6}$$

$$\frac{\partial f_{\text{curve}}(\kappa, \kappa_{\text{int}})}{\partial \kappa_{\text{int}}} < 0 \tag{4.7}$$

A function that fulfills the criteria in (4.4) - (4.7) would be the exponential function of a negative variable, e^{-x}, leading to the model expression

$$f_{\text{curve}}(\kappa, \kappa_{\text{int}}) \equiv \beta_1 e^{-\kappa} + \beta_2 e^{-\kappa_{\text{int}}} + \beta_3 \tag{4.8}$$

where $\beta_1, \beta_2, \beta_3 > 0$. This is a linear model in the parameters β_1, β_2 and β_3 which means that linear regression can be used to estimate their values according to equation (4.9).

$$\mathbf{v}_{\text{min}} = \mathbf{X}\boldsymbol{\beta} \tag{4.9}$$

where

$$\mathbf{X} = \begin{bmatrix} e^{-\kappa} & e^{-\kappa_{\text{int}}} & 1 \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

Linear regression can be used on the model (4.8). Ideally the data should be divided into training and validation data, but since the data selection is so small in this case cross validation has been used instead. Cross validation is the process of estimating multiple models with different data partitions and then averaging the results to get a better view of model performance for small data sets. Using linear regression, β_1, β_2 and β_3 received the following values

$$\begin{aligned} \beta_1 &= 14.25 \\ \beta_2 &= 9.95 \\ \beta_3 &= -4.09 \end{aligned}$$

The values were not bounded during the estimation and β_3 ended up being negative which is not allowed for this function. Setting $\beta_3 = 0$ and only estimating β_1 and β_2 gives

$$\begin{aligned} \beta_1 &= 8.45 \\ \beta_2 &= 11.15 \end{aligned}$$

with cross validation resulting in

$$R^2 = 0.745$$

$$\text{RMSE} = 2.38\,\text{m/s}$$

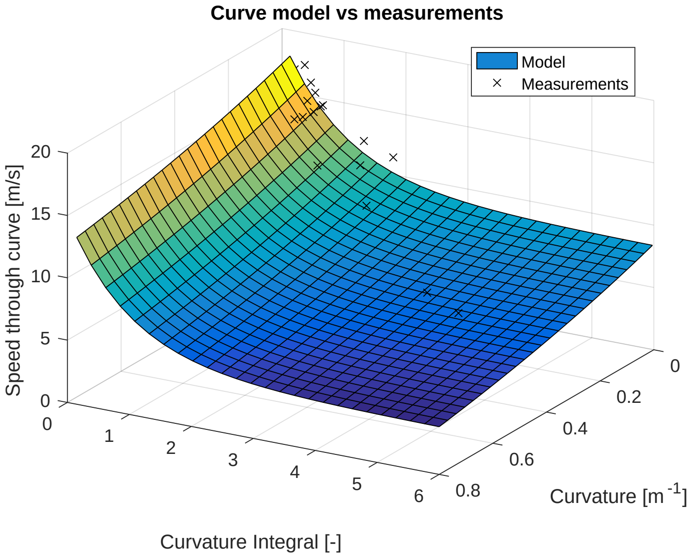


Figure 4.4: *The model in* (4.8) *with the parameters in* (4.10) *plotted as a 3-d surface together with the data points in* $\mathbf{v}_{min}$.

For the definitions of R^2 and RMSE see Definition 4.1, 4.2. The model fit to the data is reasonably good but the error is rather high, which in part could be due to lack of data and/or bad data. The curvature has been pointed out as not being entirely accurate previously and only using 9 drivers to create the mean speed profile could mean that some random variations are still visible.

The model is plotted as a 3-d surface in Figure 4.4 together with the data points used in the estimation and cross validation process.

4.2 Speed Variation

It's not possible to model the speed variation as the other models with regression simply because there is no input that maps to an output. Instead the speed variation must be modelled as a stochastic process which varies naturally in a way that replicates the speed when driving undisturbed.

Definition 4.3. A Wiener process w (or Brownian motion) is a stochastic process in which each sample k is an increment on the previous sample. The increments x are taken from an independent Gaussian normal distribution with mean 0 and variance σ^2 (Hida, 1980).

$$w_0 = 0 \tag{4.10}$$

$$x_k \sim \mathcal{N}(0, \sigma^2), \quad k = 1, \dots, N \tag{4.11}$$

$$w_k = w_{k-1} + x_k \tag{4.12}$$

In order to make sure the noise doesn't deviate too far from zero a threshold is set above which the noise will be sampled from a distribution with a mean ≠ 0 that moves it back within the threshold. The complete process can be described as follows

$$
\begin{aligned}
w_0 &= 0 && (4.13)\\
\mu_k &= \begin{cases} 0, & |w_{k-1}| < b \\ (\operatorname{sign}(w_{k-1})b - w_{k-1})a, & \text{otherwise} \end{cases} && (4.14)\\
x_k &= \mathcal{N}(\mu_k, \sigma^2) && (4.15)\\
w_k &= w_{k-1} + x_k && (4.16)\\
b, a, \sigma &\quad \text{design parameters}
\end{aligned}
$$

where b is the threshold within which the process should stay, a is the gain applied to the process outside the threshold, σ is the standard deviation. These parameters can be changed to fit different driving styles.

In order to make the process applicable to different speeds it's applied in proportion to the desired speed.

$$v_{\text{new}} = v_{\text{des}}(1 + w_k) \tag{4.17}$$

A threshold $b = 0.05$ means that the speed variation will try to stay within 5% of the desired speed and is easily observable from measurements. a and σ are harder to estimate and a trial and error approach is the best way to find values for them. Increasing a means that the driver more aggressively adjusts the speed when it drifts outside the threshold. $a = 0.01$ was found to match one particular driver reasonably well. σ controls how much variation there is between every new sample, where $\sigma = 0.001$ roughly represents the behaviour of one driver.

In order for the process to have frequency components close to human drivers it is also filtered with an MA (moving average) zero-phase filter. The process is created at 100 Hz and then downsampled to match the simulation's step length, in this case 1 Hz.

Definition 4.4. A moving average (MA) filter applied on a vector x calculates a new vector x_f where every element is the mean of all elements in x within the sliding window of size α.

$$x_f(n) = \frac{1}{\alpha}(x(n) + \cdots + x(n - (\alpha - 1))) \tag{4.18}$$

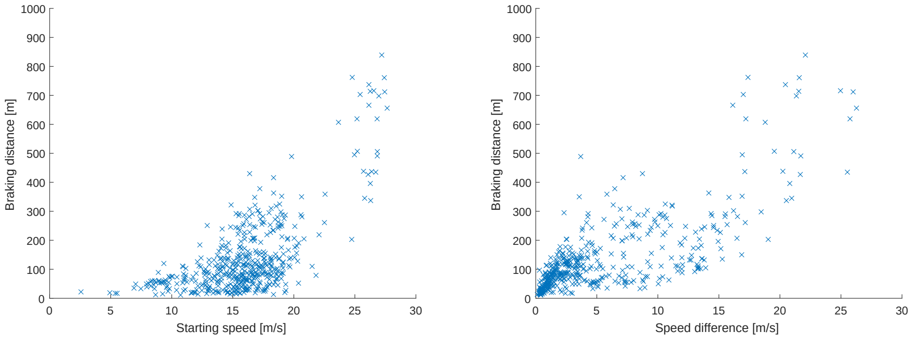
For a zero-phase filter x_f is reversed after filtering and the filter is then applied a second time to eliminate the phase shift that happens otherwise (Oppenheim, 1999).

4.3 Braking Distance

The goal with modelling the braking distance is to get a function that describes the distance that is used for braking before an obstacle on the road given the current speed and the desired speed at the obstacle. "Obstacle" here refers to static

Table 4.3: *Notation for the braking distance model. Boldface is used to indicate vectors.*

Notation	Explanation
$d_{\text{brake}}, \mathbf{d}_{\text{brake}}$	Braking distance
$v_{\text{start}}, \mathbf{v}_{\text{start}}$	Speed before braking
$v_{\text{end}}, \mathbf{v}_{\text{end}}$	Speed after braking
$\Delta v, \Delta \mathbf{v}$	Speed difference, $v_{\text{start}} - v_{\text{end}}$



*(**a**) The braking distance $\mathbf{d}_{brake}$ plotted against the starting speed $\mathbf{v}_{start}$.*

*(**b**) The braking distance $\mathbf{d}_{brake}$ plotted against the speed difference $\Delta\mathbf{v}$.*

Figure 4.5: *The braking distance is increasing in both v_{start} and Δv, though the function is not the same for both.*

features of the road such as curves and intersections with stops. Traffic lights are not included since they are not static features. The notation used throughout this section and Section 4.4 is described in Table 4.3.

4.3.1 Analyzing the Data

The data being used here is described in 3.3.2 with sections of road with traffic lights being removed. Figure 4.5 shows how the braking distance d_{brake} is dependent on both the initial speed v_{start} and the speed difference Δv of the entire braking sequence. Clear increasing trends are visible in both variables.

Model estimation will be carried out on 12 of the 15 test drives while 3 test drives are used as validation data.

4.3.2 Constructing the Model

It seems clear that the braking distance should be a positive distance that's increasing for higher values in v_{start} and Δv. The properties can be summarised

Table 4.4: *Parameters values and corresponding p-values for the quadratic model* (4.25) *estimated with linear regression.*

Parameter	Value	p-value
β_0	-25.9	-
β_1	3.97	1.96e-57
β_2	-1.27	2.60e-65
β_3	1.98	3.20e-19
β_4	-0.01	0.94
β_5	-1.26	1.61e-20

as

$$d_{\text{brake}} = f_{\text{brake}}(v_{\text{start}}, \Delta v) \tag{4.19}$$

subject to

$$\begin{aligned} v_{\text{start}} &\geq 0 && (4.20)\\ \Delta v \equiv v_{\text{start}} - v_{\text{end}} &\leq v_{\text{start}} && (4.21)\\ f_{\text{brake}}(v_{\text{start}}, \Delta v) &\geq 0 && (4.22)\\ \frac{\partial f_{\text{brake}}(v_{\text{start}}, \Delta v)}{\partial v_{\text{start}}} &\geq 0 && (4.23)\\ \frac{\partial f_{\text{brake}}(v_{\text{start}}, \Delta v)}{\partial \Delta v} &\geq 0 && (4.24) \end{aligned}$$

Given the shape of measurement data, a quadratic function on the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 \tag{4.25}$$

where $y = d_{\text{brake}}$, $x_1 = v_{\text{start}}$, $x_2 = \Delta v$ has been investigated. The constraints in (4.22) - (4.24) can then be expressed as

$$\begin{aligned} \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 &\geq 0 && (4.26)\\ \beta_1 + \beta_3 x_2 + 2\beta_4 x_1 &\geq 0 && (4.27)\\ \beta_2 + \beta_3 x_1 + 2\beta_5 x_2 &\geq 0 && (4.28) \end{aligned}$$

The parameter values have been estimated with linear regression and in Table 4.4 the values and corresponding p-values of the parameters are shown. Looking at the p-values, β_4 stands out with a very high value and also a parameter value close to 0, meaning that the term $\beta_4 x_1^2$ is insignificant and removing it should not impact the model much. Furthermore, the constraints (4.27), (4.28) imply that $\beta_0, \beta_2 \geq 0$ while at the moment they are negative. Fixing these values to 0 reduces the model expression to

$$y = \beta_1 x_1 + \beta_3 x_1 x_2 + \beta_5 x_2^2 \tag{4.29}$$

The constraint (4.27) is fulfilled for all $x_2 \geq 0$ when $\beta_1, \beta_3 \geq 0$, as they are with the current values. Constraint (4.28) states that

$$\left.\frac{\partial y}{\partial x_2}\right|_{0 \leq x_2 \leq x_1} = \beta_3 x_1 + 2\beta_5 x_2 \geq 0 \tag{4.30}$$

which can be evaluated at $x_2 = x_1$ to yield

$$\beta_3 + 2\beta_5 \geq 0 \tag{4.31}$$

With the values for β_3, β_5 in Table 4.4 the expression evaluates to < 0. Fixing the expression to 0 results in the following relationship

$$\beta_5 = -\frac{1}{2}\beta_3 \tag{4.32}$$

Finally the model (4.25) can be expressed with only 2 parameters (changing the name β_3 to β_2, and substituting y, x_1, x_2 for $d_{\text{brake}}, v_{\text{start}}, \Delta v$ respectively)

$$d_{\text{brake}} = \beta_1 v_{\text{start}} + \beta_2 \left(v_{\text{start}} \Delta v - \frac{1}{2}(\Delta v)^2 \right) \tag{4.33}$$

With linear regression the parameters are estimated to

$$\begin{aligned} \beta_1 &= 2.72 \\ \beta_2 &= 1.49 \end{aligned} \tag{4.34}$$

with

$$R^2 = 0.813$$
$$\text{RMSE} = 60.4\,\text{m}$$

on the validation data. The model does provide a good fit but the error is high which is partly due to the larger error at longer braking distances skewing the data, but there could also be refinements made to the way data is selected for modelling. Taking into account that the data is not adjusted for specific drivers and that the drivers themselves aren't particularly consistent the model error is acceptable. By comparison, estimating the complete model (4.25) with all parameters $\beta_0, \ldots, \beta_5$ yields RMSE = 57.7 m which is only a minor improvement over the model which has been used and also proves that this model structure cannot perform better with the current data selection. A 3-d surface plot of the estimated function is shown in Figure 4.6.

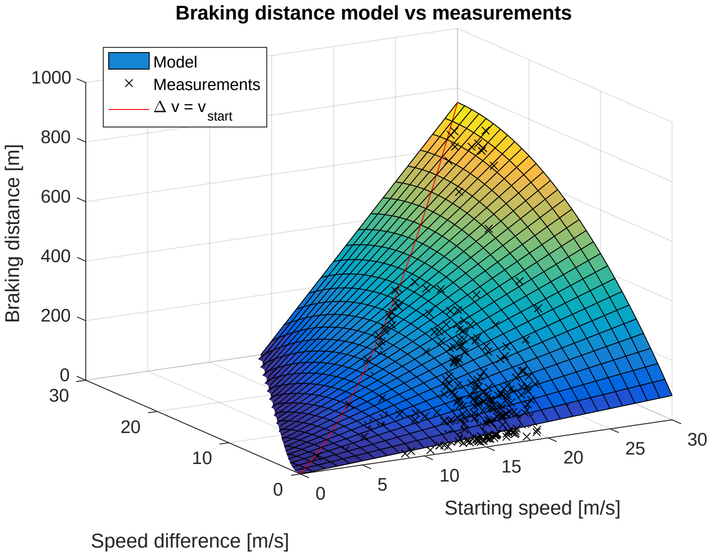


Figure 4.6: *The model in* (4.33) *with the parameters in* (4.34) *plotted as a 3-d surface together with the data points in* $\mathbf{d}_{brake}$. *The model is not defined for any values to the left of the line* $\Delta v = v_{start}$.

4.4 Braking Sequences

Braking sequences are modelled so that the deceleration before curves and intersections becomes more gradual and varied compared to the purely on/off state of the braking currently used in the simulation. Some data is first presented and then a neural network model is used to model the braking sequences. The notation in this section is the same as in the previous section.

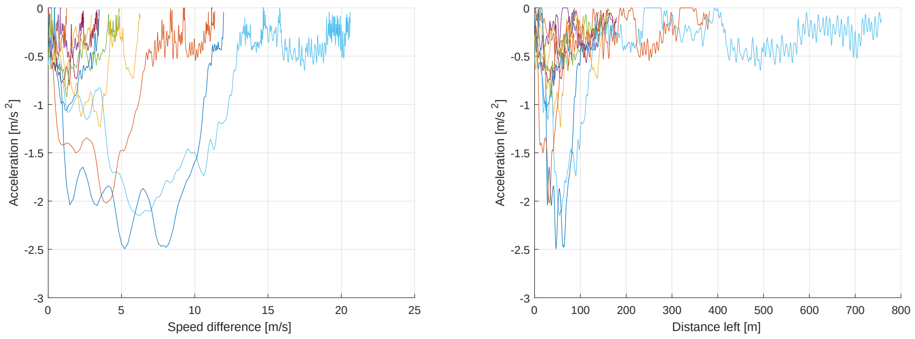
4.4.1 Analyzing the Data

The data used for modelling the braking sequences includes all speed samples between v_{start} and v_{end} and using data from all test drives results in a large selection of samples. The acceleration data is quite noisy so an MA zero-phase filter with window size 3 has been applied to the data, see Definition 4.4.

Furthermore there are a few samples with accelerations above 0 but since the model should only cover braking all data was limited to values below 0 in order to avoid accidental positive accelerations being produced by the model. In Figure 4.7 5 adjusted braking sequences are shown.

4.4.2 Constructing the Model

Due to the nonlinear nature of the data combined with the data having no clear qualities to narrow down a model from, i.e. trends or shapes, it has been decided



(a) *The acceleration during 5 braking sequences plotted against the speed difference.*

(b) *The acceleration during 5 braking sequences plotted against the remaining distance.*

Figure 4.7: *The acceleration behaviour during braking sequences looks to be highly nonlinear in nature.*

that an artificial neural network will be used for function fitting on the data. The neural network type that will be considered here is the so-called feedforward neural network consisting of an input, a hidden layer with a certain amount of neurons (also called nodes), and an output layer that produces the function output (Haykin, 1994). The neuron functions for the hidden layer are sigmoid transfer functions and the output layer contains one neuron with a linear transfer function. Whenever this thesis is referring to the network size, it's the amount of neurons in the hidden layer that's being considered. Figure 4.8 shows an overview of the network structure. This type of neural network has the ability to approximate any function given enough data and a sufficient amount of neurons in the hidden layer (Hornik et al., 1989).

The main design parameter in the neural network is to determine the amount of neurons that should be present in its hidden layer. Adding more neurons to the function generally provides a better fit to the data but with diminishing returns for every added neuron. Additionally overfitting might become a problem when using many neurons, meaning that the model does not generalize well to other input data. Since the data is taken from all drivers there are differences in disturbances and speed variations between the data sets.

The data has been divided into training and validation sets. The validation set constitutes data from three test drives amounting to approximately 20% of the data. Neural networks of different sizes have been trained on the data and the results can be seen in Table 4.5. The mean square error (MSE) has been used as a performance metric during training and training was stopped when the error increased for more than 20 consecutive iterations. The Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963) for nonlinear estimation was used during training and since it's not guaranteed to find the global minimum the re-

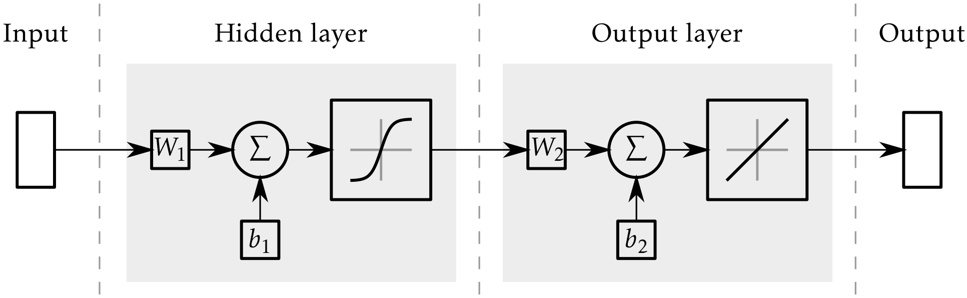


Figure 4.8: *The two layer neural network structure that will be used for approximating braking sequences. The hidden layer consists of a number of neurons that use a sigmoid function while the output layer has a single neuron that uses a linear function. The layers have weights W and biases b associated with each neuron and input. The structure shown here assumes that the weights and biases are applied via matrix operations. In this case the input is two dimensional* $(\Delta v, \Delta d)$ *and the output is scalar* (a)*, leading to* W_1 *being N×2 dimensional and* W_2 *being 1×N dimensional where N is the network size.*

sults here are not definitive. Also the network weights are randomized every time training starts which sometimes leads to different results for training. Effort was taken to avoid this problem by training multiple iterations of the same network and then choosing the one with the best performance.

Looking at Table 4.5 convergence seems to be a problem for network sizes > 6 since the network with 6 neurons actually performs better than the ones with more neurons. Another algorithm, Bayesian regularization (Williams, 1995) was tried out on these networks to see if the convergence could be more reliable by using that method. The result was better performance (RMSE) and fit (R^2) but with undesirable characteristics when extrapolating.

The network with 6 neurons performs the best and also significantly better than the one with 5 neurons while no better fit was achieved with higher amounts of neurons. Therefore it will be used as the function approximating the braking sequences. A 3-d plot of the function is shown in Figure 4.9.

The same network structure is also used to estimate a separate model for braking before traffic lights, using data from those braking sequences to train and validate on instead.

Table 4.5: *The performance of the neural network generally improves by increasing the number of neurons in its hidden layer. Multiple iterations of each network were trained in an effort to find the best possible performance for the given amount of neurons.*

Network size	RMSE	R^2_{train}	R^2_{val}
1	0.0477	0.576	0.629
2	0.0432	0.671	0.691
3	0.0396	0.733	0.781
4	0.0387	0.762	0.820
5	0.0352	0.783	0.830
6	0.0270	0.886	0.895
7	0.0290	0.867	0.880
8	0.0298	0.859	0.864
9	0.0283	0.871	0.878

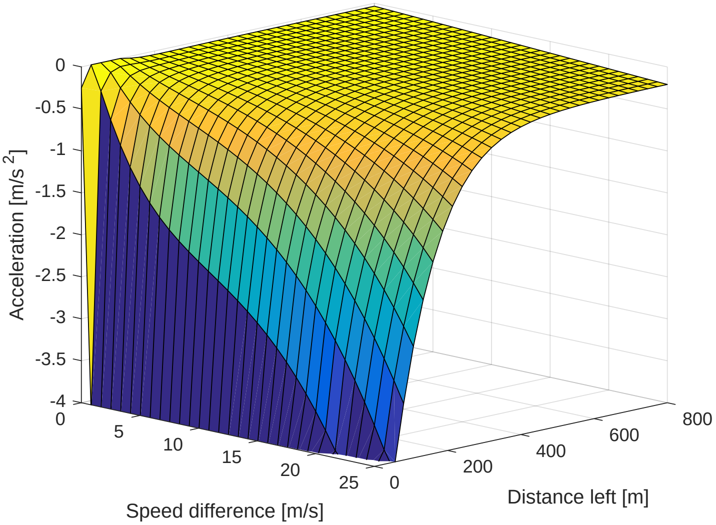


Figure 4.9: *The 6 node neural network that will be used for braking sequence approximation. The output shown here was generated with v_{end} = 5 m/s.*

4.5 Gear Changing Model

A rudimentary gear changing model has been created to allow the vehicle's acceleration parameter to take on different values depending on its current speed. If the amount of gears is g, speed values are defined in a vector $\mathbf{v}_{\text{gc}}$ of length $g - 1$ defining when gear changes will occur and the different gear accelerations are defined in another vector $\mathbf{a}_{\text{gear}}$ of length g. The algorithm initiates a gear change when the speed goes above any of the values defined in $\mathbf{v}_{\text{gc}}$. The gear change can take an arbitrary amount of time t_{gc} and during the gear change the acceleration is 0.

Sampling some acceleration sequences from one driver leads to the following parameter values

$$\begin{aligned}
g &= 5 \\
t_{\text{gc}} &= 1\ \text{s} \\
\mathbf{a}_{\text{gear}} &= \begin{bmatrix} 1.9 & 1.7 & 1.4 & 0.9 & 0.6 \end{bmatrix}\ \text{m/s}^2 \\
\mathbf{v}_{\text{gc}} &= \begin{bmatrix} 20 & 40 & 60 & 80 \end{bmatrix}\ \text{km/h}
\end{aligned}$$

In the simulation t_{gc} is bound to be a multiple of the iteration step length, which has been set to 1 s.

An extra element that has been added to acceleration sequences is a gradual decrease in acceleration when the speed is close to the desired speed. When the current speed value is over 85% the value of the desired speed, the acceleration will start to decrease.

5 Implementation

This chapter will briefly describe how the implementation is made in terms of algorithms and how it performs compared to the basic simulation just using SUMO.

5.1 Parameters for Vehicles

Options have been added to the simulation to control which of the new models are used and also to assign the added vehicle parameters.

Table 5.1: *Available vehicle parameters for the simulation with the implemented models.*

Parameter	Data type	Description
doSpeedAdjustment	boolean	Activates the dynamic speed adjustments for road features
doNewNoise	boolean	Activates the speed variation
doGearChanges	boolean	Activates the gear changing algorithm
accVec	$[1 \times g]$ double	Vector containing the acceleration values at different gears
accSwitchSpeed	$[1 \times (g-1)]$ double	Vector containing the speed when gear changes occur
timeToSwitch	double	Time it takes to switch gears
decel	double	Maximum deceleration the vehicle is capable of
speedFactor	double	Factor that is applied to all speeds by the vehicle
noiseParams	$[1 \times 3]$ double	Noise parameter vector

5.2 Algorithm

The algorithm that determines the vehicle's speed during the simulation consists of several steps detailed below.

1. **Find objects to brake for**

   The braking distance model is used to search a length of road ahead of the vehicle. By evaluating the function at $\Delta v = v$ the maximum possible braking distance for the current speed is received. Within this distance local maxima in curvature are retrieved as well as the road's speed limits and any traffic lights. The distances between these features and the vehicle are recorded and the curve speed model is used to calculate the speed through the identified curves. If a traffic light is red, the desired speed at that position is 0.

2. **Determine if braking should start**

   The braking distance model is used again to calculate the braking distance to all of the found speed reductions. Different cases can occur depending on what is found within the look-ahead distance:

   (a) **Calculate braking action**

   If any of the calculated braking distances are shorter than the current distance between the vehicle and the speed reduction, a braking action will be calculated using the neural networks. The maximum braking action out of these candidates is then applied to the vehicle.

   (b) **Keep the current speed**

   If speed reductions are found but no braking distance is currently short enough to initiate braking, the vehicle will keep it's current speed until braking starts. Also occurs if neither speed reductions nor speed increases are found.

   (c) **Calculate acceleration**

   If no speed reductions are found but the speed limit is higher than the current speed, acceleration is calculated with gear changes.

5.3 Performance

It's important to highlight that performance in SUMO is significantly impacted when using TraCI (Traffic control interface) with MATLAB. TraCI is also available for Python, Java and C++ and since these versions have not been tested it's not known if performance could be improved by using any of these interfaces instead of the MATLAB one. Looking at the breakdown in Table 5.2 it's apparent that communication via TraCI takes up about the same amount of time that the simulation itself takes in this case. TraCI commands are used extensively to continually save data from the controlled vehicle, such as speed, position and lane

Table 5.2: *Simulation performance when running the implemented code in MATLAB without the SUMO GUI. Only a single vehicle was present in the simulation. 3500 simulation steps were carried out in total, of which the controlled vehicle was present for around 1300. The numbers represent the total time spent during the entire simulation. Performance data was gathered using MATLAB's profiler tool.*

Function task	**Time taken** [s]
Initializing the simulation	3.88
SUMO performing simulation steps	12.56
Reading data via TraCI	11.98
Sending data via TraCI	1.91
Reading XML-files with Python	9.33
Calculating spline and curvature	3.33
Calculating new speeds in MATLAB	1.68
Other MATLAB functions	1.37
Total time	**46.04**

occupancy, data which is needed for determining new speeds during the simulation and also for analysis after the simulation is finished. The total simulation time for the example with TraCI communication in Table 5.2 is 46 s while running the same simulation without TraCI takes 17 s, almost three times shorter. Improvements that can be made with regards to performance are optimizations to reading the XML files and minimizing the amount of data that is accessed via TraCI.

6

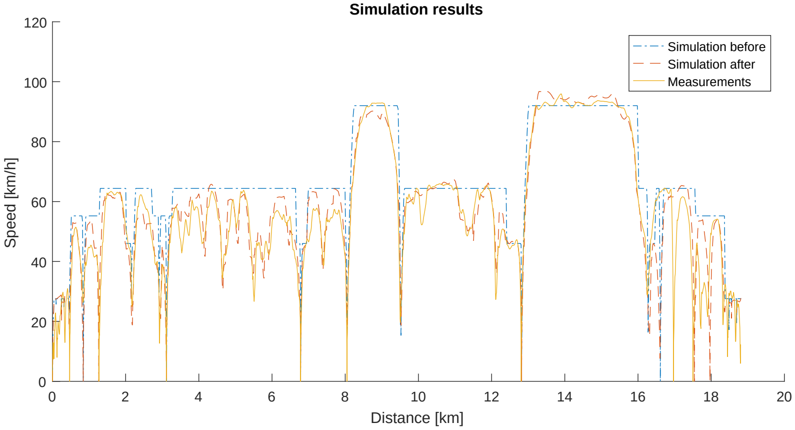
Results and Discussion

This chapter presents the results of the thesis with comparisons between the simulation and measurements and also some discussion on the quality of the results.

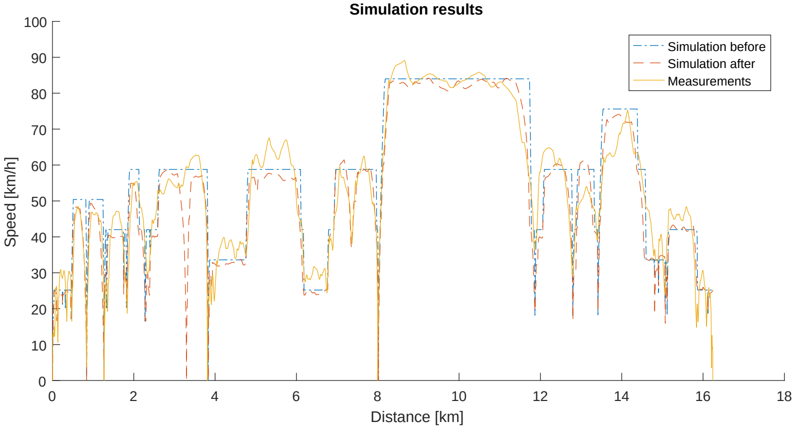
6.1 Comparisons

The figures on the following pages show speed profiles from the route planned for this thesis and speed profiles from the route used in a previous research project. The simulation before any adjustments are made is compared with the simulation after the adjustments from this thesis are implemented, and one measured speed profile from a single driver. Parameters have been adjusted in the models to match the drivers as close as possible. Any sudden stops occurring during the first 3 km and last 2.5 km are due to traffic lights and since traffic lights were not synced between the simulation runs and the measurements these stops might differ. While recording data for this thesis it was noted whether the driver was affected by traffic in any way so for the comparison here, a test drive could be chosen where the amount of traffic was minimal. Traffic information was not included in the previously recorded data however, so a test drive was selected where the driving looked suitably “clean” for the most part but it’s not known where the driver could have been affected by traffic and to what extent.

The speed profile belonging to the measurements carried out for this thesis seen in the figures below was previously used as part of the validation data set for the braking distance and braking sequence models. Since cross validation was used for the curve speed model, the curves in these measurements have been used for both training and validation. The speed profile belonging to the old route has not been involved during modelling but some sections of the route are shared with the route that was used for modelling. Note that the comparison between the two routes does not compare the same sections of road, rather sections of road with similar properties.

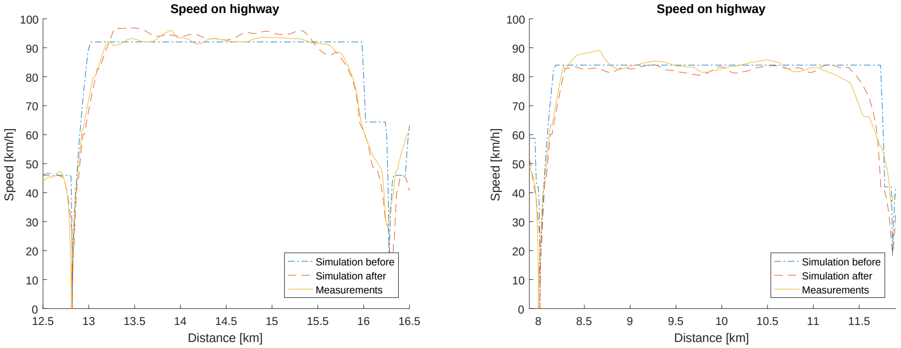


(a) The route for this thesis, used for modelling.



(b) The route used in a previous research project, not used during modelling.

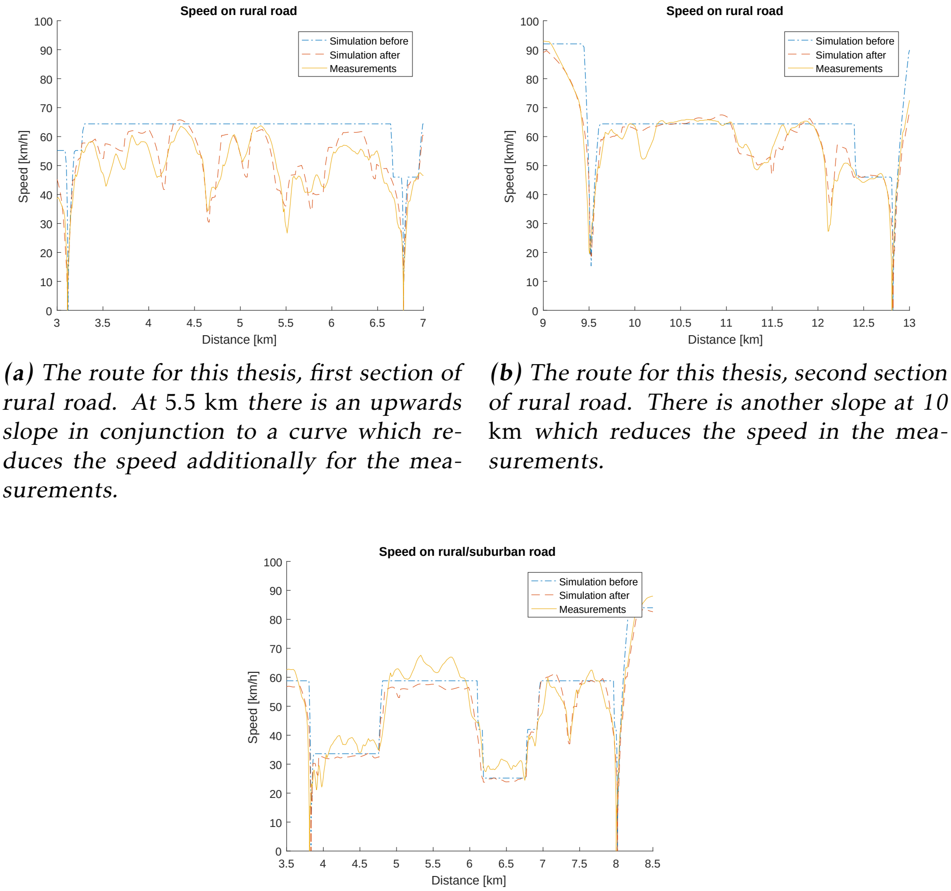
Figure 6.1: *In general the adjusted simulation manages to follow the measurements well, disregarding the random disturbances. It's much improved compared to how the simulation behaves without the adjustments made in this thesis.*



*(**a**) Data recorded for this thesis.*

*(**b**) Data from the old route.*

Figure 6.2: *In these plots the effect of the gear changing algorithm can clearly be seen during the accelerations, changing the available acceleration depending on the current speed. When cruising, the speed variation model in the adjusted simulation is observed to produce a speed variation that closely resembles the measurements in terms of amplitude and frequency. The braking sequences created by the neural network exhibit the more gradual braking seen in the measurements, and the determined braking distance is accurately estimated.*



*(**a**) The route for this thesis, first section of rural road. At* 5.5 km *there is an upwards slope in conjunction to a curve which reduces the speed additionally for the measurements.*

*(**b**) The route for this thesis, second section of rural road. There is another slope at 10* km *which reduces the speed in the measurements.*

*(**c**) The old route. This road section includes some suburban driving where there are speed bumps and similar measures designed to lower the speed at certain points along with intersections with poor line of sight, all contributing to the larger speed variations seen in the measurements.*

Figure 6.3: *Here the effect of the curve speed model can be observed. The model manages to provide an approximation of the curve speed where the simulation previously kept a constant speed, but there are still improvements to be made. The models for braking distance and braking sequences perform well with results close to the measurements for most of the braking sequences.*

6.2 Discussion

The results show that the estimated models can replicate most of the speed adjustments that human drivers apply through curves and intersections. The neural network used for calculating braking sequences performs very well when compared to human drivers but lacks some of the random variations sometimes observed in the measurements, see Figure 6.3a where there are many shorter braking sequences. The braking distance model also performs well with most of the estimated braking sequences starting at positions close to what can be observed in the measurements. It should be noted that these two models are applied in both curves and places where the speed limit is lowered and that they seem to perform well in both cases. The speed variation function that's applied when driving on straight roads manages to reproduce a speed variation similar to human drivers. The result is primarily visible for highway driving but the function is applied any time there are no external disturbances visible to the vehicle.

Out of the models assembled in this thesis, the model that performs the worst is the curve speed model. It mainly comes down to not having enough data to model on but also the quality of the data itself. Being bound to use the map data for calculating the curvature limited how closely the results could match real life driving and despite efforts to make the data more usable by creating splines from the road geometry and also making use of the curvature integral it still doesn't fully represent reality.

While the results are generally good there are also other things that affect the driving speed than just curves and speed limits, as seen in Figure 6.3. Intersections, slopes, speed bumps, road surface and surroundings all play a role in affecting the speed to varying degrees. In order to properly model human driver behaviour and improve the models further more external factors will have to be considered when deciding the speed.

The methodology for the models that determine driver behaviour used in this thesis is based on the decisions and actions of human drivers when approaching curves or intersections. During the process of working on this thesis different approaches were investigated for the speed adjustment through curves such as a model that only used current road curvature and vehicle speed as input and produced a new vehicle speed as output. The problem with basing the speed directly on the curvature data is that drivers adjust their speed well before reaching a curve which means that reactions to the curvature occur before any change in the input has happened. This behaviour led to the conception of the curve speed model together with the look-ahead model where each model has a single purpose as opposed to one model that controls all driver behaviour. Creating multiple smaller models allowed the estimation process for each to be more straightforward since the process from input to output is simpler and easier to visualize. In comparison, the process of how a curve ahead of the vehicle affects the current speed is not as simple and it might be hard to decide on a model structure and constraints in the same way as was done for the curve speed model and the braking distance model.

An advantage with using multiple models with separate tasks is that the two models not explicitly attached to curves can be used individually in other situations. The look-ahead model and the braking sequence model are applied for all speed reductions, something that would not have been possible if a single model had been used for the entire process. The decision-making process that is used in the implementation would not have been possible had there only been one model governing the curve driving behaviour. Applying the models individually is what allows the implementation to work as well as it does.

A neural network might have been able to estimate the full driver behaviour given enough inputs and training data, though it would probably have had to be larger (more neurons and likely more layers) than the one currently used for braking sequences. With a more advanced network structure there's even the possibility that the decision-making process could be handled by the network as well. The negative side of this is while using a large neural network one would be giving up control of the estimation process. The look-ahead model and the curve speed model have constraints on them which can be handled well when using regression. With a neural network, especially a large and complex one, it's very difficult to control how the output is shaped. Additionally, the data recorded for this thesis would likely not have been enough to train a more sophisticated neural network. The neural network that was used for estimating braking sequences is a fairly small one with only two layers and six neurons, taking two inputs and producing one output essentially acting as a non-linear function.

7

Conclusions and Future Work

This chapter will go through what conclusions can be drawn from this thesis and what further work can be carried out in order to improve or expand it.

7.1 Conclusions

Based on comparisons between the simulation data and real world driving, the results from this thesis show that it is possible to model human driver behaviour to approximate the speed they drive at with respect to certain road features. Mainly, the braking distance used for a given speed reduction starting at a certain speed has been successfully modelled as a quadratic function while entire braking sequences have been emulated well with use of a neural network. The curve speed model presented in this work is inconclusive because of lack of data, but an approximation of curve speeds with regards to curvature has been achieved.

The models have also been implemented in MATLAB to work together with the SUMO traffic simulation environment where simulations can be carried out on imported maps from OpenStreetMap. The implementation makes use of a decision-making process that finds objects to brake for and applies the correct braking action. This creates the possibility to simulate arbitrary routes on any existing road network throughout the world with the implemented models producing speeds much more consistent with human drivers.

7.2 Future Work

Since the curve speed model lacks sufficient data, one possibility for future work is to perform a more rigorous data collection with a more varied selection of drivers on different roads to get a large and varied data set of curve speeds and general driving. The curvature data received from the road geometry has not been sufficiently detailed and has often not been accurate to real life. Using road data from some other more accurate source could be the solution.

At the moment a driver only reduces their speed at intersections if they are turning, encounter other vehicles, or if there's a stop sign or a traffic light present. Determining behaviour through intersections requires information about the intersections themselves and the connecting roads. SUMO has many intersection types which can be useful for predicting behaviour and each road also has an associated priority which helps the simulation determine right-of-way scenarios and yielding in intersections. However, there is no parameter for different road surfaces which would be useful in separating insignificant gravel roads from paved roads in intersections. The data is available from OpenStreetMap which SUMO can import maps from so perhaps it could be possible to include the additional data when importing.

Line of sight in intersections and curves in part determine how fast one drives through them and also if objects are visible to the driver. Implementing line of sight would only be possible if the surroundings are well defined. The imported data from OpenStreetMap does include forest areas and buildings, but there is no way to access any of this data from within the simulation at the moment.

The version of SUMO that has been used only makes use of 2d map data so slopes in the terrain are disregarded completely. With a 3d map and a model changing the speed for slopes it should be possible to get more accurate results in places with significant elevation differences.

Only a single speed factor for speed limits for each driver has been used, which has been observed to not be entirely representative of how drivers treat speed limits and can certainly be improved on in the future. The driver's perception of the road plays a role in determining the speed they drive at. Most drivers will stay well below the speed limit when driving on the smaller rural roads with speed limits at 70 km/h (see Figure 6.3 as an example, a and b in particular) because of curves, poor line of sight and also the road surface being damaged. But on other roads with a speed limit that's deemed too low the driver might drive much closer to or even above the speed limit (can be seen in Figure 6.3c).

Bibliography

M. Brackstone and M. McDonald. Car-following: A historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2(4):181–196, 1999. Cited on pages 3, 4, and 5.

Robert E Chandler, Robert Herman, and Elliott W Montroll. Traffic dynamics: Studies in car following. *Operations research*, 6(2):165–184, 1958. Cited on page 5.

Environmental Protection Agency (EPA). EPA Federal Test Procedure (FTP), 2017. URL `https://www.epa.gov/emission-standards-reference-guide/epa-federal-test-procedure-ftp`. Accessed 2017-08. Cited on page 2.

Peter G Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105–111, 1981. Cited on pages 5 and 7.

Qiuming Gong, Shawn Midlam-Mohler, Vincenzo Marano, and Giorgio Rizzoni. An iterative Markov chain approach for generating vehicle driving cycles. *SAE International Journal of Engines*, 4(2011-01-0880):1035–1045, 2011. Cited on page 3.

Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1994. Cited on page 34.

Takeyuki Hida. Brownian motion. In *Brownian Motion*, pages 44–113. Springer, 1980. Cited on page 28.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. Cited on page 34.

S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, Mathematisch–Naturwissenschaftlichen Fakultät der Universität zu Köln, Köln, 1998. Cited on pages 3 and 7.

Per Kågesson. Cycle-Beating and the EU Test Cycle for Cars. European Federation for Transport and Environment, 1998. Cited on page 2.

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. Cited on page 34.

Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11 (2):431–441, 1963. Cited on page 34.

Peter Mock, John German, Anup Bandivadekar, and Iddo Riemersma. Discrepancies between type-approval and “real-world” fuel-consumption and CO_2 values. International Council on Clean Transportation, 2012. Cited on page 2.

Douglas C Montgomery and George C Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2003. Cited on page 23.

P. Nyberg, E. Frisk, and L. Nielsen. Using Real-World Driving Databases to Generate Driving Cycles with Equivalence Properties. *IEEE Transactions on Vehicular Technology*, 65(6):4095–4105, 2016. Cited on page 3.

Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999. Cited on page 29.

G. Souffran, L. Miègeville, and P. Guérin. Simulation of real-world vehicle missions using a stochastic Markov model for optimal design purposes. In *2011 IEEE Vehicle Power and Propulsion Conference, VPPC 2011*, 2011. Cited on page 3.

H.Y. Tong, W.T. Hung, and C.S. Cheung. Development of a driving cycle for Hong Kong. *Atmospheric Environment*, 33(15):2323–2335, 1999. Cited on page 3.

M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 62(2 B):1805–1824, 2000. Cited on page 5.

Martin Treiber, Arne Kesting, and Dirk Helbing. Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A: Statistical Mechanics and its Applications*, 360(1):71–88, 2006. Cited on page 5.

United Nations. Addendum 100: Regulation No. 101, Agreement Concerning the Adoption of Uniform Technical Prescriptions for Wheeled Vehicles, Equipment and Parts which can be fitted and/or be used on Wheeled Vehicles and the Conditions for Reciprocal Recognition of Approvals Granted on the Basis of these Prescriptions, 2013. Cited on page 2.

Peter M Williams. Bayesian regularization and pruning using a Laplace prior. *Neural computation*, 7(1):117–143, 1995. Cited on page 35.