

Driver Modeling, Velocity and Energy Consumption Prediction of Electric Vehicles

Carl Bredberg

John Stjernrup



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6035
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2017 by Carl Bredberg & John Stjernrup. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2017

Abstract

A driver model can be used to predict the vehicle velocity and the energy consumption. It can be modeled such that it benefits from historical data and can be further improved if a navigation system is available. A well implemented driver model is important since the car fleet seems to be more and more electrified. The role of the driver model would then be to increase the accuracy for when the driver needs to take a break to recharge the vehicle, and thereby decrease the driver's range anxiety.

Historical behaviour of different drivers has been measured and collected by Volvo Car Corporation. The information regarding these drivers has been used in four out of five implemented driver models. Three of the models use Markov chain theory to make the prediction while the fourth takes advantage of frequency analysis. Above the aim to increase the accuracy of the energy consumption prediction it is investigated to what extent a personal driver model can be created.

In addition to the driver models a primitive method to predict when a driver reacts to a new posted reference speed is proposed and four validation methods are suggested.

The results indicate that the driver models based on historical data perform better energy predictions than the one without any historical data. The driver model that uses Gaussian mixture model together with Markov chain theory makes the best energy prediction. The individual differences are especially shown at road segments with higher reference speed. In urban traffic it is more likely that the traffic pattern decides the energy consumption.

Acknowledgements

We would like to thank all the colleagues at Powertrain, Strategy and Concepts at Volvo Car Corporation. We would like to especially thank our supervisor Martin Sivertsson that has had unlimited with curiosity, energy and given us fantastic support. Thanks also to Christoffer Strömberg and Lee Chih Feng that have participated in a lot of discussion and helped us move forward. For all the happy moments together, we would like to thank Johan Almgren and Gustav Elingsbo.

We would also like to thank the department of Automatic Control at Lund University. A special thanks to our supervisor Anton Cervin for all your relevant feedback during the work with the thesis.

*Göteborg, June 2017
Carl Bredberg and John Stjernrup*

Dictionary & Abbreviations

Speed category	-	The driver's reference speed
TPM	-	Transition probability matrix
ODE	-	Ordinary differential equation
ODEwNoise	-	Ordinary differential equation with added noise
GMM	-	Gaussian mixture model
DD	-	Discrete distribution
PHEV	-	Plug-in hybrid electric vehicle
BEV	-	Battery electric vehicle
DTE	-	Distance to empty

Contents

1. Introduction	11
1.1 Problem formulation	11
1.2 Limitations	12
1.3 Approach	13
1.4 Thesis contribution	13
1.5 Thesis outline	14
2. Background	15
2.1 Energy prediction in vehicles today	15
2.2 Personal driving style	17
2.3 Inspiration from previous work	18
3. Theory	20
3.1 Physical relationships	20
3.2 Modeling	22
3.3 Validation	25
4. Modeling	27
4.1 Velocity profile generation	27
4.2 Structure of the transition probability matrix	30
4.3 Prediction based on the transition probability matrix	33
4.4 Prediction based on ordinary differential equations	36
4.5 Training routes	37
4.6 Individual reaction to changed speed category	38
4.7 Energy calculations	39
4.8 Tuning parameters	40
4.9 Validation	41
4.10 Summary modeling	46
5. Results	47
5.1 Driver models	47
5.2 Robustness of the driver models	52
5.3 Memory and computation requirements	54
5.4 Tuned parameters	55

6. Discussion	58
6.1 Validation methods	58
6.2 General error sources	60
6.3 Driver models	61
6.4 Robustness of the driver models	64
6.5 Memory and computation requirement	66
6.6 Tuned parameters	67
6.7 Further work	69
7. Conclusion	71
A. Appendix	72
A.1 Driver models	72
A.2 Robustness	74
A.3 Less training data	76
Bibliography	78

1

Introduction

Today, the society is battling an increasing average temperature on earth. Since the transport sector is one of the contributors to the exhaust of greenhouse gases, it is important to make the propulsion of the vehicles as efficient as possible. One of the keys to increase the efficiency of passenger vehicles is so called plug-in hybrid vehicles (PHEV) and battery electric vehicles (BEV). The PHEV is a hybrid with both a combustion engine and an electrical engine. From the perspective of the PHEV, the prediction of the velocity and torque during a trip is important to be able to optimize the usage of the energy sources. As for a BEV, it is important to offer the driver a correct prediction of the range of the battery, such that the driver can schedule when to recharge the battery.

One of the most used arguments to not buy a BEV is that the range is not as long as for a conventional car and that it takes long time to charge it. But if the energy used by the BEV can be better predicted, the driver hopefully loses the range anxiety and most of the time can self-better plan when he or she needs to stop and charge. In this way it gets more likely that the BEV gains trust from the driver.

1.1 Problem formulation

In this thesis it is investigated if it is possible to estimate a personalized energy consumption by predicting the velocity and torque. It is assumed that the route to drive is known and therefore also the signed speed on the road and road inclination.

With information regarding reference speed on a specific road segment the velocity can be predicted. For example only by following the signed speed of the road. A problem with this simple way to predict the velocity is that it not catches the fluctuations in velocity that naturally happen when driving a car. A measured velocity during a route is shown in Figure 1.1 demonstrating that fluctuations occur. A hypothesis in this thesis is that the fluctuations affect the energy consumption to such an extent that they needs to be taken into account.

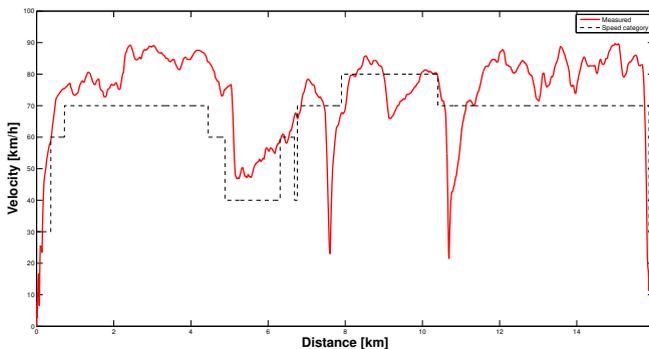


Figure 1.1: Velocity profile where the measured velocity is plotted together with the speed category, i.e. reference speed.

As shown in Figure 1.1, the driver seems to drive faster than the speed category unless something is preventing him or her from doing that. This leads into the question if every single driver has his or her own driving style or rather follows the traffic pattern. And, is it possible and beneficial to make a personalized driver model or is it more effective to use one standardized model due to the dominance of the traffic pattern, such that memory and computer power needed in the car can be minimized. To test these questions, four different history based driver models are implemented and evaluated.

1.2 Limitations

The data that is used in this thesis is collected by Volvo Car Corporation. The data is labelled as vehicle specific. This means that the data from one vehicle can have several different drivers. In this thesis it is assumed that one vehicle always is driven by the same driver. To minimize measurement errors, as few as possible measured inputs were used.

To predict the future velocity on a road, road specific information, such as road inclination and reference speed, needs to be available. This could be provided by a navigation system. The best reference speed would be the average velocity on the road. In this thesis though, that was not available. Instead the signed velocity, recorded by the camera on the car, was used as reference speed. The advantage with the average velocity would have been that it better reflects reality. It should also be time dependent. For example is the mean velocity downtown often much lower during rush hour. The road inclination was measured by the car. Furthermore,

information where different constantly recurrent occurrences occur, like stop signs and traffic lights, were not available.

1.3 Approach

The purpose of this thesis is to benchmark different ways to predict the behaviour of unique drivers. To do so, a literature study is performed to find basics for predictive models which could be of interest for this thesis.

Further on, data from Volvo Car Corporation are provided and the setup of the data environment is done in Matlab. The data contains files from many different vehicles and each file has historical information about the unique vehicle.

The historical data is divided into training and validation data. The training data is used to train four driver models while the validation data is used as comparison for the results from the predictions. Three of the models are based on Markov chains. The training data is used to train the transition probability matrix that is used in these models. The fourth model is trained by a frequency analysis of the training data. Also a fifth model is implemented without any connection to the historical behaviour.

To know if the models used to predict the energy consumption work, a method to validate against the validation data is implemented. Finally, the results of the validation are presented and analyzed. The drivers are tested both against their own validation data to be able to see which driver models that make the best prediction and also against other drivers' validation data. This is to test robustness of the different approaches and if it is possible to create individual driver models.

1.4 Thesis contribution

This thesis contributes with implementation and evaluation of five different driver models. Their purpose is to estimate the energy consumption during a known route by predicting the velocity and torque. Also a simple method to catch individual reactions to a new speed category and four different ways to validate the results from the prediction are suggested.

1.5 Thesis outline

Chapter 2 describes why this thesis is of interest, what has been done so far on this subject and also earlier theses and reports that have been an inspiration. Continuing into Chapter 3, the physical relationships to propel a vehicle are explained. Further on the theory behind the different models used to predict the velocity follows.

Chapter 4 is divided into parts explaining how the whole process is done from processing the collected measured data to obtaining the results. It describes which driver models that are chosen to be implemented, how they are implemented, how they are trained and finally validated.

In the next-coming two chapters, Chapter 5 and Chapter 6, the results are presented and a discussion is held. Finally a conclusion is made in Chapter 7.

2

Background

This chapter begins with a part which describes some techniques that are used today to estimate how far a person can drive with the remaining fuel in the car. Both in conventional cars and vehicles using a battery. A section follows with motivation why there is a need to perform this investigation. The chapter ends with a summary of the previous works that has been of inspiration for this thesis.

2.1 Energy prediction in vehicles today

One can say that the aim of this thesis is to do a better prediction of the "distance to empty (DTE) tank information". In other words, a better estimation of how far the vehicle can travel on the remaining energy storage. Today it is very common that cars can give the driver information about how far it is until the tank is empty. To do so primitive methods are often used. Different suppliers use different algorithms to estimate the DTE. Volvo V40 from 2015 [Volvo, 2017] uses an algorithm where it, based on the mean fuel consumption during the last 30 km and the remaining fuel in the tank, calculates the distance to empty tank as

$$DTE = \frac{V_{tank}}{\bar{\Psi}_{30km}} \left[\frac{l}{l/km} \right], \quad (2.1)$$

where V_{tank} is the volume of fuel left in the tank and Ψ_{30km} is the amount of fuel consumed per kilometer during the last 30 km.

More variables must be considered when it comes to an electrical vehicle. To calculate the energy left in the battery, variables as: state of charge (SOC), battery voltage, battery temperature and battery aging effect [Gao, 2014] need to be taken into account. If the battery energy is known the DTE can be calculated as in Eq. (2.2). $E_{battery}$ is the energy left in the battery. $P_{driving}$ is the power used to propel the vehicle, $P_{climate\ control}$ power for example the air condition and $P_{auxiliary}$ power for

functions in the car besides the propulsion and climate control.

$$DTE = \frac{E_{battery}}{P_{driving} + P_{climate\ control} + P_{auxiliary}} \bar{v} \quad \left[\frac{kWh}{kW} km/h \right] \quad (2.2)$$

Since the total power usage, can change dramatically in a BEV, the instantaneous value should not be the best value to use. According to [Gao, 2014] the rolling averaged value of the velocity, \bar{v} , is often used. The rolling average time differs from supplier to supplier and can be from a couple of seconds to several minutes. Because of this high complexity to calculate the DTE in a BEV it is a good idea to investigate how the total power usage can be predicted which indirectly is what this thesis will focus on.

Figure 2.1 shows how the SOC varies during a route. The figure shows that the calculations of the DTE from Eq. (2.1) not predicts the correct DTE. What is seen from the measured data is that it fluctuates and differs during a route. With some information about the route, these fluctuations should be able to be captured by the driver model.

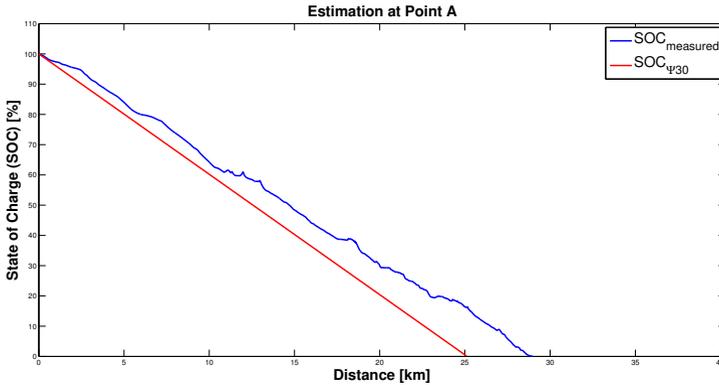


Figure 2.1: The plot shows an example were the SOC was measured and when it was predicted via Eq. (2.1) at point A. I.e. when the route start, 0 km.

A navigation system is often available in cars nowadays. It is easy for the driver to insert which route he or she will drive but the route can also be predicted. As example of a route prediction the vehicle can learn that the driver always drives to work at 8 a.m. Monday to Friday. Then the route is easy to predict and it is not necessary to insert the route to the GPS. Since there are many ways to find the route, this thesis assumes that the route always is known. Information that can be used to predict the driving power by the known route are: reference speed, road inclinations, distances, how much the road is turning and so on.

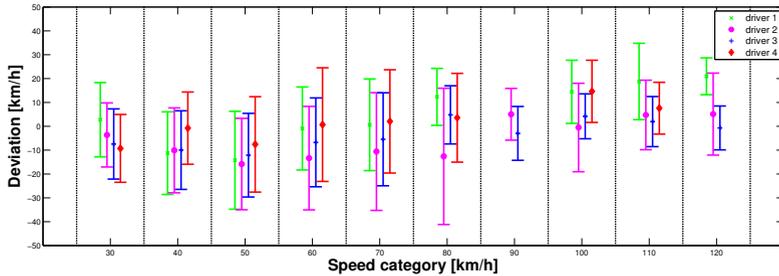


Figure 2.2: The plot shows four different drivers' average velocity deviation from the speed category and their standard deviation from the mean value. To be included in the plot, the driver was required to have driven a specific distance at each speed category. E.g. driver 1 has not driven the required distance, 80 km, at speed category 90 km/h.

2.2 Personal driving style

Above the energy estimation, this thesis investigates if it is possible to make a personal driver model. To do so the hypothesis is that it is of interest to predict the velocity profile. Looking at Figure 2.2 the mean and standard deviation of velocity relative to the velocity on the road is shown for four different drivers. The differences shown in mean and also spread in velocity indicate that there are driver unique effects.

Some important facts are shown in Figure 2.2. It can be seen that the drivers do not drive that similar to each other but they are following the same pattern for themselves. The figure shows that *driver 1* almost always drives faster than the speed category. What also can be seen in the figure is that *driver 2* has a large standard deviation. That is a pattern that is important to include in a power estimation since big variation in velocity may come from large accelerations and decelerations. In such a case it is important to catch that behaviour in the prediction since acceleration affects the energy consumption. What also can be seen is that *driver 3* is at higher speed categories more inclined to choose a specific velocity and remain there.

Regarding the standard deviation, it is important to remember that a large deviation does not necessarily mean a lot of changes in velocity. E.g. if a person drives 50 km with velocity 130 km/h and cruise controller, and then another 50 km with velocity 110 km/h and cruise controller, a relatively large standard deviation will be obtained even though the driver keeps a steady velocity.

Continuing on that reasoning, how often a driver changes velocity could be investigated by doing a frequency analysis on the velocity. Thereof Figure 2.3, which shows the mean frequency from the same four drivers driving on the same routes as

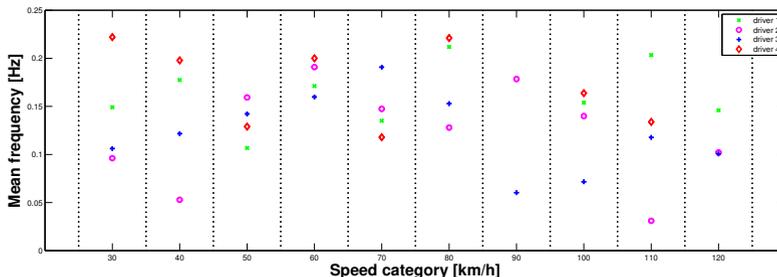


Figure 2.3: The plot shows the four drivers' mean frequency of their velocity signal at different speed categories. The frequency analysis is done on the same drivers and routes as in Figure 2.2. The same markers are also used.

in Figure 2.2.

If regarding *driver 1* and *2* at speed category 110 km/h in Figure 2.2, it can be seen that they are getting almost the same, quite big standard deviation for the velocity. Continuing by looking at Figure 2.3 it can be seen that *driver 2* drives at very low mean frequency and *driver 1* drives at high mean frequency. It is likely that *driver 2* in this case very often uses the cruise control at freeways. This is an argument why also the frequency needs to be investigated.

With this information, the conclusion that people tend to drive differently compared to each other can be drawn. And it is of interest to investigate if it is possible to make a personal driver model.

2.3 Inspiration from previous work

There are a couple of theses and papers that investigate how to predict the future velocity of a vehicle. Several methods that this thesis uses to predict the future velocity are based on Markov models. The idea to do so was inspired by the thesis [Torp and Önnegren, 2013]. The objective of their thesis was to generate stochastic driving cycles based on real-world data. To generate the driving cycles they used Markov chain theory.

One can think that the driver only chooses the next velocity depending on what he or she experiences at that specific moment. By taking this into consideration Markov chains are a powerful method to use since it only uses the current state to predict the future.

Another report, [Müller et al., 2004] looks into the possibility to predict the fu-

ture velocity taking the road conditions and individual driving styles into account with the purpose to optimize the fuel consumption. They compare three cases: "Speed limit", "velocity regarding curves on the road" and "velocity given vision". The case that restricts the velocity the most is declared as the target speed the driver wants to reach. To adapt the prediction algorithm to the specific driver, personally tuned coefficients are used. For example, the magnitude of one coefficient determines how fast a driver responds to a new speed limit sign. Also this report uses Markov chains, though in this case, to handle stochastic events in the traffic.

The report [Karbowski et al., 2012] also considers the impact from external circumstances. They use a tool that provide them with data like stop signs, traffic lights and traffic pattern speeds. To make use of the information about possible stops, there is an algorithm determining the likelihood whether the traffic light is red or green. The choice to use traffic pattern speed is interesting, since then variations due to traffic, curves and vision is taken into consideration unlike the case to only use signposted speed limits.

Furthermore, in a report from [Schori et al., 2015], three different ordinary differential equations are used to model the speed profile, having information from the navigation system available. One state each to represent acceleration, deceleration to a lower speed and deceleration to stop the vehicle respectively. A method to add driver individual noise is also used in the report by [Schori et al., 2015]. Via frequency analysis of the velocity a personalized noise is found and added to the velocity from the ordinary differential equation model.

3

Theory

This chapter is divided into three different parts. The first contains the theory regarding physical properties to calculate the energy consumed by a car in motion. Secondly, the theory necessary to create and understand the driver models are described. The last section explains the mathematical tools relative error and confidence interval.

3.1 Physical relationships

In order to calculate the estimated energy during a route the predicted velocity and resulting force need to be known. The first subsection contains the equations for the energy calculation. The equations to calculate the resulting force are presented in the following subsection. Finally the theory regarding power regeneration is explained.

3.1.1 Energy calculations

The energy needed is calculated through

$$E = \int P dt, \quad (3.1)$$

where E is the total energy when the power P is integrated over time. The power though is

$$P = T \frac{v}{r}, \quad (3.2)$$

where T represents the total torque at the four wheels, v is the velocity and r is the radius of the wheel. Total torque is

$$T = \sum_{n=1}^4 T_n. \quad (3.3)$$

Torque is also a product of resulting force $F_{resulting}$ applied perpendicularly to the wheel axis at the distance r from the axis.

$$T_n = F_{resulting} \cdot r \quad (3.4)$$

3.1.2 Road load polynomial

The force $F_{resulting}$ is the result of adding the force due to acceleration, F_a , and $f(v)$ together. $f(v)$ is the force it takes to keep the vehicle at constant speed.

$$F_{resulting} = f(v) + F_a, \quad (3.5)$$

where according to [Guzzella and Sciarretta, 2007] the road load polynomial $f(v)$ is the sum of the aerodynamic friction losses F_{aero} , the rolling friction losses F_r , the force caused by gravity when driving on non-horizontal roads F_g and a disturbance force F_d . In this thesis the disturbance force has been neglected.

$$f(v) = F_{aero} + F_r + F_g = \frac{1}{2} \rho c_d A_f v^2 + mg(c_r \cos \theta + \sin \theta) \quad (3.6)$$

where ρ is the air density, c_d is the drag coefficient, A_f is the front area of the car, v is the velocity of the car, m is vehicle mass, g is the gravitational acceleration, c_r is the rolling friction constant and θ is the degree of which the road deviates from the horizontal plane.

The accelerating force is calculated through *Newton's second law*,

$$F_a = m \frac{dv}{dt} = ma, \quad (3.7)$$

where m is the vehicle mass and a acceleration.

3.1.3 Power regeneration

Well known fact is that PHEVs and BEVs offer the possibility to propel the vehicle with renewable energy. Another advantage with the vehicles propelled partially or fully with a battery as energy source is that it is possible to regenerate power if the circumstances allow. For example if the driver brakes or drives downhill. How the power regeneration works will not be explained in this thesis. But it is important to remember that all energy, from example a braking, can not be regenerated due to the efficiency in the components between the wheel and battery.

If the power P demanded to propel the vehicle due to a specific torque is calculated according to Eq. (3.2), the power sent from the battery is, in this thesis, calculated as follows.

$$P_{battery} = \frac{P}{\eta}, \quad (3.8)$$

where η is the product of the efficiency in the battery, electrical motor and gearbox.

$$\eta = \eta_{battery} \cdot \eta_{EM} \cdot \eta_{GB} \quad (3.9)$$

The additional losses mentioned before are a consequence during the regeneration when $P_{regeneration}$, from the kinetic energy in the wheels, is used to charge the battery. Then the power to the battery is

$$P_{battery} = \eta P_{regeneration}. \quad (3.10)$$

100 % regeneration means that all of the power generated by the braking force could be taken care of except for the losses (difference between $P_{regeneration}$ and $P_{battery}$). An option to this could be that only a specific amount of power, P_{level} , could be recovered. Then, if $P_{regeneration} > P_{level}$, a certain amount of power would go to waste.

$$P_{waste} = P_{regeneration} - P_{level} \quad (3.11)$$

3.2 Modeling

This section contains the necessary theory about Markov based statistical tools, ordinary differential equations and frequency analysis to be able to create the driver models.

3.2.1 Markov chains

To predict the future, a possible tool to use is Markov chains. Since the time in this thesis is regarded as discrete, this is a case for discrete Markov chains. The idea with Markov chains is to predict a future sequence of events, only using the current state, X , as information, according to the following equation [Rydén and Lindgren, 2000].

$$\mathbf{P}(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_0 = x_0) = \mathbf{P}(X_n = x_n | X_{n-1} = x_{n-1}) \quad (3.12)$$

To do so a column vector, x_s , that contains the probabilities of different states at time s is assumed. The sum of the elements in x_s is one.

$$x_s = [x_1 \quad x_2 \quad \dots \quad x_n]^T \quad (3.13)$$

To calculate what probability of the state the process will be in the next time step, a quadratic transition probability matrix is needed.

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \quad (3.14)$$

where n denotes the number of states in the matrix. Each state is represented by a specific row and the sum of each row is equal to one. The number p_{kj} at an arbitrary

position (k, j) in the matrix, indicates the probability to go from state k to state j . The two above equations multiplied with each other, results in information of what state the process will be in at a specific time d .

$$x_{s+d} = P^d x_s \quad (3.15)$$

3.2.2 Gaussian mixture model

To fit a vector of latent variables a Gaussian mixture model can be used. The latent variables is in this thesis the changes in acceleration that have not occurred during the training phase. The model is based on that the latent variables can be clustered to K -number of clusters. These clusters are assumed to be Gaussian distributed. The K -number of Gaussian distributions are then mixed together in a multivariate Gaussian distribution

$$p(\theta) = \sum_{i=1}^K \mathbf{w}_i \mathcal{N}(\mathbf{x} | \mu_i, \Sigma_i) \quad (3.16)$$

where \mathbf{w} is a vector with the mixing coefficients of the Gaussian distributions. If there are several observations \mathbf{x} is a vector $[x_1, x_2, \dots, x_N]$. μ are the mean values and Σ are the covariance matrices. The parameters \mathbf{w} , μ and Σ are found using the Expectation-maximization algorithm.

[Bishop, 2007]

3.2.2.1 Expectations maximization algorithm

The expectation-maximization algorithm is an iterative algorithm to find the parameters that maximizes the likelihood of a function. In this case the algorithm is used such that it maximizes the likelihood function with respect to the Gaussian Mixture Model parameters i.e. the mixing coefficients w , the means μ and the covariances Σ . The algorithm is described as in [Bishop, 2007].

The algorithm is divided into two steps, E-step and M-step. The initialization of w_k , μ_k and Σ_k is done by the *K-means-algorithm*. When the initialization is set, the initial value of the log likelihood is evaluated. Then the algorithm does the E-step.

E-step: Evaluate the posterior probabilities using the current parameter values.

$$\gamma(z_{nk}) = \frac{w_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K w_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}. \quad (3.17)$$

M-step: Re-estimate the parameters according to the new posterior probability

$$\mu_k^* = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n, \quad (3.18)$$

$$\Sigma_k^* = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k^*) (x_n - \mu_k^*)^T, \quad (3.19)$$

$$w_k^* = \frac{N_k}{N}, \quad (3.20)$$

where

$$N_k = \sum_{n=1}^N \gamma(z_{nk}). \quad (3.21)$$

Evaluation: Evaluate the log likelihood for the new parameters

$$\ln p(X|\mu, \Sigma, w) = \sum_{n=1}^N \ln \left[\sum_{k=1}^K w_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right]. \quad (3.22)$$

If neither the convergence criterion for the parameters or the log likelihood is satisfied, return to the E-step.

3.2.3 Acceleration calculations with ordinary differential equations

To reach a desired velocity one option is to use ordinary differential equations (ODE). In [Schori et al., 2015] three different equations are proposed to predict a velocity profile. In this thesis two of them are used and they are representing acceleration and deceleration respectively to a desired velocity.

$$a = \dot{v} = e \left(1 - (v/v_{cat})^\delta \right) \Big|_{v \leq v_{cat}} \quad (3.23)$$

$$a = \dot{v} = -e \left(1 - (v_{cat}/v)^\delta \right) \Big|_{v > v_{cat}} \quad (3.24)$$

where a is acceleration, v is current vehicle velocity and v_{cat} is desired velocity. e and δ are constants. Typically $e \in [1 \ 2] [m/s^2]$ which represents typical accelerations and decelerations and $\delta \in [2 \ 6]$ which determines when the acceleration is reduced while approaching v_{cat} . The first equation will result in a positive acceleration while the second results in a negative acceleration.

3.2.4 Frequency analysis

According to [Folland, 1992], Fourier analysis can be described as "*a collection of related techniques for resolving general functions into sums of integrals of simple functions or functions with certain properties*".

The idea is to represent a time dependent signal $s(t)$ in the frequency domain instead of in the time domain. To do so, the standard definition for an integrable signal is, [Folland, 1992],

$$S(\Omega) = \int s(t) \cdot e^{-i\Omega t} dt, \quad (3.25)$$

where $S(\Omega)$ is the signal in frequency domain and $\Omega = 2\pi f$. f is frequency. In the case for a finite discrete signal $x(n)$ with N samples the relationship is, according to [Folland, 1992], described as

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi kn}{N}}, \quad (3.26)$$

where X_k is a sequence of size N with complex numbers where each number contains information about the amplitude and the phase for a certain frequency component of x_n .

To calculate the amplitude and phase at sample k , the following relationship

$$A_k = \frac{\sqrt{\Re(X_k)^2 + \Im(X_k)^2}}{N}, \quad (3.27)$$

holds for amplitude and

$$\phi_k = \tan^{-1} \left(\frac{\Im(X_k)}{\Re(X_k)} \right), \quad (3.28)$$

is used to find the phase. $\Re(X)$ means the real part of X and $\Im(X)$ is the imaginary part of X .

The interval of different frequencies that could be found is derived assuming sampling period T_s . Then considering *Nyquist-Shannon sampling theorem*, the frequency interval is

$$f \in \left[0 \quad \frac{1}{2T_s} \right] \quad (3.29)$$

The theory regarding frequency analysis will be used both as an attempt to make the differential equations in Subsection 3.2.3 individual but also as a way to validate the results obtained later on in the thesis.

By taking advantage of the Eq. (3.27), (3.28) and (3.29) a time dependent noise Π can be calculated for N number of amplitudes, phase shifts and frequencies.

$$\Pi = \sum_{i=1}^N A_i \cos(2\pi f_i t + \phi_i) \quad (3.30)$$

3.3 Validation

The theory necessary for validation of the prediction is presented below.

3.3.1 Relative error

If a measured value x is compared with the estimated value x^* the relative error δ between these values are

$$\delta = \frac{|x - x^*|}{x} \quad (3.31)$$

This equation is later used to validate the results from the predictions.

3.3.2 Confidence interval

A confidence interval can be used to find an interval where the mean value, μ will occur with a specific certainty, $1 - \alpha$. Eq. (3.32) shows how a confidence interval with known standard deviation σ is calculated.

$$I_{\mu} = \left[\mu - \lambda_{\alpha} \frac{\sigma}{\sqrt{N}}, \mu + \lambda_{\alpha} \frac{\sigma}{\sqrt{N}} \right], \quad (3.32)$$

where λ_{α} is the quantile and N the number of samples. When this statistical tool is used in this thesis, $\alpha = 0.05$ is chosen, hence, $\lambda_{\alpha} = 1.6449$.

4

Modeling

This chapter describes the steps in the process to obtain the results. In other words, how the measured data is processed to give an energy estimation.

The chapter begins with a section where the algorithm that is used to create a velocity profile is presented together with an overview of the five driver models. The structure of the transition probability matrix is described in Section 4.2. That is followed by a description of how the TPM obtains its content during the training phase.

How the three methods based on Markov chains use the TPM to predict the velocity is described in Section 4.3. Further on the fourth and fifth model that are based on the ordinary differential equations are explained in Section 4.4.

The training routes are visualized in Section 4.5 followed by Section 4.6 which describes a primitive method that is used to predict when a driver reacts to a new speed category. Next Section 4.7 explains the energy calculations after that a velocity profile has been estimated.

The five models open up possibilities to tune parameters such that they impact the results. The tuning parameters are shown in Table 4.5 in Section 4.8. Section 4.9 presents the validation routes and proposes the four different validation methods that will be used to validate the five driver models. The validation methods are based on velocity, power, change in energy with respect to distance and frequency.

4.1 Velocity profile generation

Five driver models will be investigated in this thesis. All models use the same kind of algorithm to generate a velocity profile. The predicted velocity profiles are to be

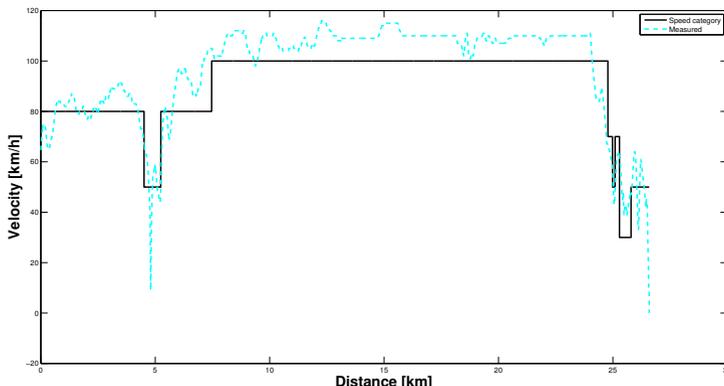


Figure 4.1: An example of a velocity profile over 27 km. Black solid line is speed category and the dashed line is measured velocity.

compared to a measured velocity profile. For example the one shown in Figure 4.1.

The algorithm to predict a velocity profile is described in Figure 4.2. The driver models are constructed such that a change in acceleration, Δa is generated in every time step. The time between two samples is fixed.

4.1.1 Algorithm

Assume that the velocity profile in Figure 4.1 should be generated. The generation of an estimated velocity profile works as in Figure 4.2, where $d_{AB} = 27 \text{ km}$, $a_0 = 0.55 \text{ m/s}^2$, $v_0 = 62 \text{ km/h}$ and $\mathbf{v}_{cat}(d_0^*) = 80 \text{ km/h}$. \mathbf{v}_{cat} is a vector with all speed categories during the route. The speed category vector has a corresponding distance vector, such that the speed category at a specific distance can be picked from the position in the speed category vector that corresponds to the position of the closest distance in the distance vector.

The block "Model" in Figure 4.2 is replaced by one of the five driver models.

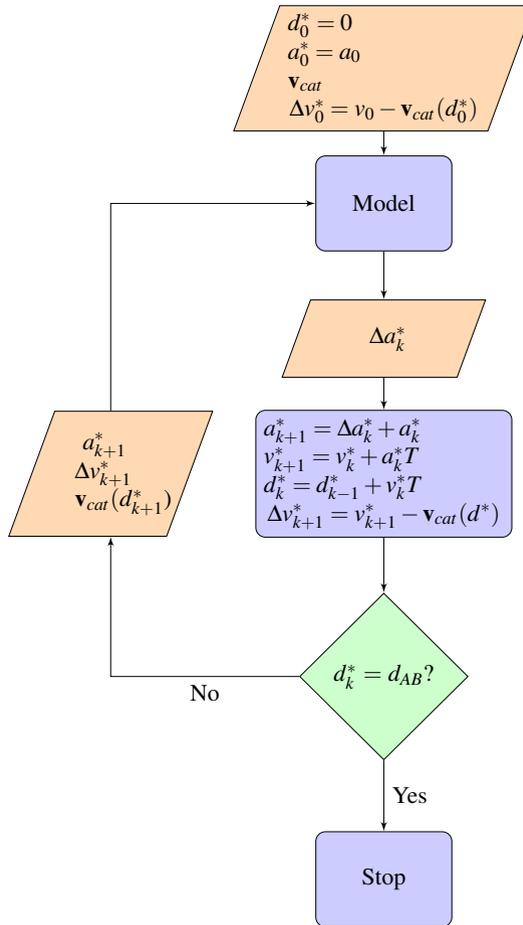


Figure 4.2: Schematic view of how a velocity profile is generated.

4.1.2 Driver Models

Three models are developed from Markov models. They use the same transition probability matrix (TPM) in different ways. The TPM is trained and then the models have different solutions for how they should use the TPM.

The first model chooses its next Δa by considering the information in each state as discrete distributions. The second model uses Gaussian mixture model to find latent values and build a mixture of Gaussian distributions in each state from which it then predicts Δa . The third model uses the Gaussian mixture distributions and developing them such that it considered both the state the prediction already visited

and estimates future states it may visit. The distribution from these states are concatenated such that next Δa can be predicted.

The other structure is based on the ordinary differential equations in Eq. (3.23) and (3.24). From these equations two driver models are implemented. One model which only uses the ordinary equations to predict Δa . The second model is trained such that it imitates the drivers frequency when driving at different speed categories. The frequencies are added to the ODE.

4.2 Structure of the transition probability matrix

The following section describes the structure of the TPM and how it is trained. The concept of the TPM is shown in Table 4.1. All parts in the table will be described in the upcoming subsections.

Table 4.1: The structure of the TPM. S_i is an unique state with respect to acceleration, a , difference between current velocity and speed category, Δv , and speed category, v_{cat} . \mathbb{O}_i is what this thesis calls an observation matrix of size $u \times 2$ where u is the number of unique changes in accelerations, Δa , that has occurred during the training data in that state.

State	Observations
$S_1(a_1, \Delta v_1, v_{cat_1})$	\mathbb{O}_1
$S_2(a_1, \Delta v_1, v_{cat_2})$	\mathbb{O}_2
\vdots	\vdots
$S_n(a_x, \Delta v_y, v_{cat_z})$	\mathbb{O}_n

4.2.1 Choice of state variables

There are many factors taken into account when a driver determines his or her target velocity. As described in Section 2.3, both [Müller et al., 2004] and [Karbowski et al., 2012] write about outer circumstances limiting the velocity. For example physical properties on the road, stop signs, traffic pattern speed and traffic lights are all factors which a driver should take into account. Implementing these variables though, would require access to a tool providing this data. Furthermore, [Torp and Önnegren, 2013] are using velocity and acceleration as variables.

The state variables chosen in this thesis will be influenced from [Torp and Önnegren, 2013], i.e. velocity and acceleration, but also the speed category, which will represent the reference from outer circumstances, will be taken into consideration.

To make a decision of what target speed the driver wants to have, he or she needs to have some inputs. These inputs will be the state variables. In this thesis it is

assumed that the three inputs the driver considers the most when driving a vehicle is: the speed category, current acceleration and current velocity. The input current velocity will be changed to Δv , which is the difference between current velocity and speed category. This is done to reduce the size of the TPM.

$$\Delta v = v_{act} - v_{cat} \quad (4.1)$$

To summarize, the state variables chosen are acceleration a , speed category v_{cat} and Δv .

4.2.2 Grid

The state variables presented in Subsection 4.2.1 are implemented as arrays and together they will be referred to as the grid. The values that the speed category can adopt are given in kilometers per hour and declared in the vector v_{cat} where $v_{cat} = 0$ is used when there is no speed category registered in the data set. This happens when the vehicle has not registered any speed limit, typically in the beginning of a route.

$$v_{cat} \in [0, 120], \quad (4.2)$$

where the step between two speed categories is 10 km/h , resulting in 13 possible speed categories that the process can adopt.

The array to keep track of the acceleration is the following.

$$a \in [-3, 3], \quad (4.3)$$

where the resolution is highest around 0 m/s^2 . I.e, the difference between two elements next to each other is larger in the beginning and the end of the array, than it is in the center. The chosen array for acceleration means that it is possible to have 47 different accelerations.

For Δv the array is

$$\Delta v \in [-102, 64]. \quad (4.4)$$

Also here the step length between two elements varies. The number of possible Δv is 54 and the smallest steps between adjacent values is around 6 km/h . Values, during training or prediction, that are not present in the arrays will be rounded to the closest value available in an array.

Regarding the chosen grid some states can never be reached since the velocity can't be negative. The fact that Δv can adopt large negative values, down to -102 km/h , becomes more important the higher the speed category gets. The model needs to find a difference between when driving a bit slower than speed category and much slower than speed category. This because it is assumed there is a very significant difference in how you drive if you follow a truck, normally: $\Delta v = -40 km/h$

at $v_{cat} = 120 \text{ km/h}$ or are in a queue due to roadwork, normally: $\Delta v = -100 \text{ km/h}$
 at $v_{cat} = 120 \text{ km/h}$.

The sizes of the arrays, a , Δv and v_{cat} , implies that there will be

$$47 \cdot 54 \cdot 13 = 32994 \tag{4.5}$$

different states. If the states which can't be reached due to only positive velocities are removed, there are 26837 possible states for the process to visit.

4.2.3 Training

The TPM obtains its content when the training data is stored into a specific spot with respect to the inputs. As reported earlier, the inputs to enter a specific state is the acceleration, difference between velocity and speed category and speed category.

Compared to [Torp and Önnegren, 2013] who store both the difference in velocity, difference in acceleration and the probability at each spot in the TPM, only the difference in acceleration to the next time step, Δa_k , and number of observations are stored in this thesis. Δa_k is calculated by Eq. (4.6).

$$\Delta a_k = a_{k+1} - a_k \tag{4.6}$$

Furthermore, Table 4.2 is an example of an observation matrix \mathbb{O}_x that was introduced in Table 4.1.

Table 4.2: Illustration of the content in a specific state. Δa_1 and Δa_2 represents two different Δa and p and q is the number of observations of the specific Δa .

Δa	Occurrences
Δa_1	p
Δa_2	q
\vdots	\vdots

As mentioned in Subsection 4.2.2, the speed category sometimes was registered as 0 km/h. To store the training data at the correct position in the TPM in this case two assumptions were made. Firstly, for velocities above 60 km/h, the road segment was assumed to be a road with unlimited speed limit. When this occurred, the data was stored in 120 km/h. Secondly, if the velocity was below 60 km/h the data was stored in 0 km/h.

Furthermore, speed category equal to 5 km/h and 15 km/h occurred, but rarely. When it happened the speed category was treated as 10 km/h and 20 km/h respectively.

4.3 Prediction based on the transition probability matrix

By using the TPM a discrete distribution model and a Gaussian mixture model is implemented. The Gaussian mixture model is further developed to a multi-step prediction which could take the historical and future distributions into account. All these three methods take advantage of the TPM to find the change in acceleration, Δa , to then calculate the acceleration and velocity.

4.3.1 Discrete distributions

When using the TPM to predict the velocity profile using Markov chains with the discrete distributions, each state in the TPM contains discrete values of Δa , see Table 4.2. Further on Δa is selected probability based i.e. with respect to the number of occurrences in the training data. In other words that means that Δa , could never adopt a value which has not occurred during the training session. An example of a discrete distribution is shown in Figure 4.3. If a state in the TPM doesn't contain any Δa , the next Δa is calculated by the ordinary differential equations from Eq. (3.23) and (3.24).

With respect to the upcoming discussion about memory and computational requirements it is important to remember that the information in the discrete TPM is stored as described in Table 4.2.

4.3.2 Gaussian mixture model

Another model to find Δa is implemented by a Gaussian mixture model (GMM). The Δa is picked from a three clusters Gaussian mixture distribution. The mixture distributions in each state is generated as in subsections 3.2.2 and 3.2.2.1 from the discrete values in each state in Table 4.2. This means that a new TPM is generated where every state holds a Gaussian mixture distribution instead of a discrete distribution. The number of clusters K , could be changed to optimize the model. Δa is chosen from the probability distribution which is shown in Figure 4.3. It is a three cluster Gaussian mixture distribution created from the discrete distribution in the same figure. As shown in the figure, Δa that have not occurred during training can be picked by this method. The GMM is implemented with three clusters. This model is constructed in the same way as DD such that if a state is empty, the next Δa is calculated by the ordinary differential equations from Eq. (3.23) and (3.24).

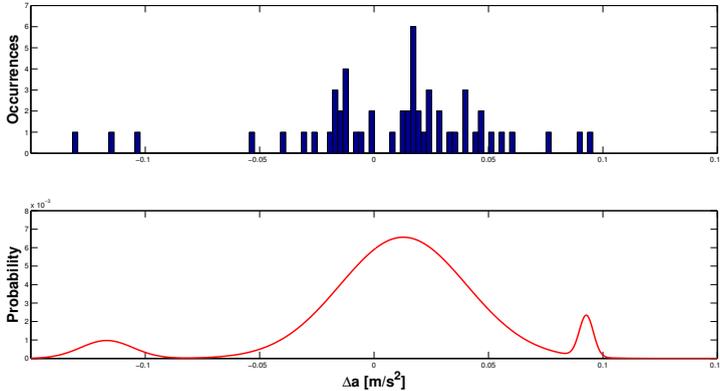


Figure 4.3: The upper subfigure shows a discrete distribution, while the lower one shows the Gaussian mixture distribution that is created from the discrete distribution.

The way the information is stored in the TPM for GMM, is in a different way than for DD. Each state holds enough variables to create, for example, the distribution showed in Figure 4.3. Basically expected values and covariance matrices for the distributions.

4.3.3 Multi-step prediction

The Gaussian mixture model is further implemented to do a multi-step prediction. This method will question the idea that a driver only acts depending on the present state by making it possible to take both historical states and estimated future states into the equation.

The model is implemented such that states that have been visited before, current state and future states are merged together. The merge of the Gaussian mixture distributions in each state is done by weights such that current state is the most contributing to the new distribution. The future states are estimated by the visited states.

Regarding time and computational requirements for the Multi-step prediction it is a trade-off between, choosing longer time for the prediction to be done and have less information saved, or faster computations and more information in every state. In this thesis faster computations is chosen. That means that the information about the distributions in the states in the TPM for Gaussian mixture model, see Subsection 4.3.2, are replaced by a more memory consuming array. The array is later referred to as D and contains information about the probability to choose a specific Δa from a Gaussian mixture distribution.

4.3 Prediction based on the transition probability matrix

The number of backward steps n_b and forward steps n_f decides how many states backward and forward this multi-step method should use respectively.

The backward distributions are found by storing n_b distributions that the simu-

lation has visited as a vector $D_b = \begin{bmatrix} D_{k-n_b} \\ \vdots \\ D_{k-1} \\ D_k \end{bmatrix}$, where k represents the current time.

From this vector every backward state distribution can be found. To find the future step distribution an estimated Δa_k^* needs to be found. To find Δa_k^* a fused distribution D^* is calculated, to simplify the code implementation, the vector D_b is extended

with n_b zero distributions such that $D_b = \begin{bmatrix} D_{k-n_b} \\ \vdots \\ D_{k-1} \\ D_k \\ \mathbb{0}_{k+1} \\ \vdots \\ \mathbb{0}_{k+n_b} \end{bmatrix}$.

To be able to choose how much each distribution should influence the fused distribution a weight vector \mathbf{w} is implemented. The weights are set such that the current state k is the most important and then the latest state $[k-1]$ is the next to most important and so on. To make the weight vector to a parameter that can be varied through different simulations \mathbf{w} is implemented as following:

$$\mathbf{w} = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \quad (4.7)$$

where $\mathbf{x} = [-n_b : n_b]$. Then \mathbf{w} is normalized as in Eq. 4.8.

$$\hat{\mathbf{w}} = \frac{\mathbf{w}}{\sum_{i=0}^{n_b} w_i} \quad (4.8)$$

This gives a weight vector that is normal distributed around $\boldsymbol{\mu}$. $\boldsymbol{\mu}$ is set to 0 which implies that the largest weight is placed on the current state distribution. $\boldsymbol{\sigma}$ is the standard deviation which can be varied through different simulations. $\boldsymbol{\sigma}$ decides how much the backward distributions further from the current state can impact the fused distribution. The smaller $\boldsymbol{\sigma}$ is, the less important are distributions further away in time from the current distribution. The fused distribution is calculated as in Eq. 4.9.

$$D^* = \sum_{i=-n_b}^{n_b} w_i D_{k+i} \quad (4.9)$$

From D^* a Δa_k^* is chosen. With help of Δa_k^* a new velocity v_{k+1}^* , acceleration a_{k+1}^* , and speed category $v_{cat_{k+1}}^*$ is found. I.e, a new state is found. From these the estimated distribution $D(a_{k+1}^*, \Delta v_{k+1}^*, v_{cat_{k+1}}^*)$ is found in the TPM.

The same procedure continues n_f times. To clarify:

$$\text{Next step, the } D_b = \begin{bmatrix} D_{k-(n_b-1)} \\ \vdots \\ D_k \\ \mathbf{0}_{k+1} \\ \vdots \\ \mathbf{0}_{k+n_b+1} \end{bmatrix}.$$

When the procedure has done n_f iterations a new distribution vector D_{tot} is built with all the calculated distributions as following.

$$D_{tot} = \begin{bmatrix} D_{k-n_b} \\ \vdots \\ D_{k-1} \\ D_k \\ D(a_{k+1}^*, \Delta v_{k+1}^*, v_{cat_{k+1}}^*) \\ \vdots \\ D(a_{k+n_f}^*, \Delta v_{k+n_f}^*, v_{cat_{k+n_f}}^*) \end{bmatrix}$$

If $n_b \neq n_f$, D_{tot} is filled up with zeros as in D_b to make $[\max(n_b, n_f) * 2 + 1]$ number of rows in D_{tot} , such that the code is simplified.

The weight vector $\tilde{\mathbf{w}}$ is implemented almost in the same way as before:

$\tilde{\mathbf{w}} = \mathcal{N}(\mathbf{x}|\mu, \sigma^2)$ where $\mathbf{x} = [-\max(n_b, n_f) : \max(n_b, n_f)]$. $\tilde{\mathbf{w}}$ is normalized with the sum of the weights for the distributions that is not zero. Then Δa_k is found in the same way as before from the fused total distribution Eq. (4.9).

The three parameters that can be changed for this method is the width of the weight vector. I.e, how much the different forward and backward states should impact the choice of the next Δa . The second and third parameters are the number of backward and forward steps that should impact the choice of Δa . Since this method is built on the Gaussian mixture model the benefits from that method will be utilized in the multi-step method.

4.4 Prediction based on ordinary differential equations

Two models use the ordinary differential equations described in Subsection 3.2.3 in different ways to calculate the acceleration. The first model is a simple model that is independent of the historical data. It is implemented such that it only uses the ODE. The second model is further developed to be more personalized by a frequency analysis of the driver.

4.4.1 Acceleration via ordinary differential equations

The fourth method to determine the acceleration uses Eq. (3.23) and (3.24) depending on if the current velocity is below or above the speed category. I.e. the TPM is not used in this case. Neither does it have any connection to training data, except the individual reaction to speed category, making the velocity profile non-individual.

The values used for e and δ in the equations are shown below.

$$\begin{cases} e = 1.2 \text{ m/s}^2 \\ \delta = 5 \end{cases}$$

4.4.2 Acceleration via ordinary differential equations with added noise

As a way to make the velocity profile in the section above individual Fourier analysis is used as described in the discrete case in Subsection 3.2.4. The signal in time domain, x_n , is in this case the velocity as a function of time at a specific speed category and the output X_k , velocity as a function of frequency. Using Eq. (3.27) and (3.28) the amplitude and phase in the frequency interval described by Eq. (3.29) is calculated. The original velocity signal used is a part of the velocity from the training data.

For a specific time t the noise is calculated according to Eq. (3.30) The number of frequencies, N , used could be varied but is in this thesis the same as the amount of X_k .

Furthermore, in an attempt to make the model even more individualized v_{cat} , in Eq. (3.23) and (3.24) is set to the historical mean velocity for the driver at that speed category.

e and δ for the equations to calculate the new acceleration is the same as in the case for clean ODE. This method will be referred to as ODEwNoise, abbreviation of ordinary differential equation with noise.

4.5 Training routes

Three different drivers, D_A , D_B and D_C are used in this thesis. Their training routes were found by taking the first 5000 km they have driven and illustrated in Figure 4.4. This means that the training data for the different drivers may be very different from each other if time traveled on the different speed categories are compared. The thought though, is to investigate if a car needs to put any computer power to filter and store some specific data at different speed categories. If this solution works, the car just needs the memory capacity that is necessary for 5000 km.

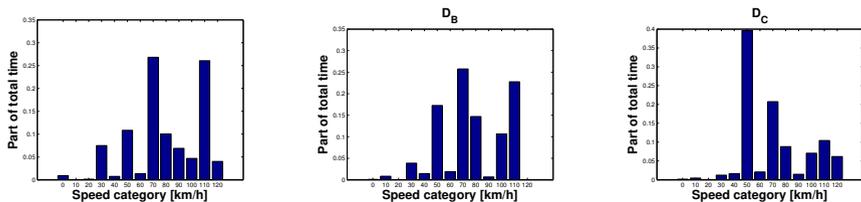


Figure 4.4: The subfigures show the three drivers’ 5000 km training routes as a histogram over the part time spent at each speed category.

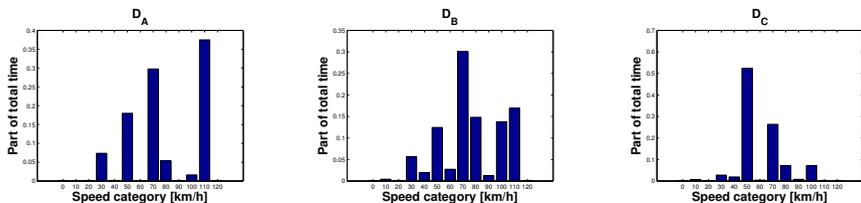


Figure 4.5: The subfigures show the three drivers’ 500 km training routes as a histogram over the part time spent at each speed category.

To investigate if it is possible to get good result after training at a small amount of data, training routes at a tenth of the distance of the training routes in Figure 4.5, 500 km are implemented. If the results of these routes are as good as for the longer training routes, the memory capacity in the vehicle can be minimized.

4.6 Individual reaction to changed speed category

To further make the driver models more individualized an algorithm is implemented to find when a driver is reacting to a new reference speed. The distance in meter is saved with respect to both old and new speed category. The individual distances for a driver is located during the training phase of the TPM.

For the algorithm to find the distance when a driver reacts to the new reference speed, a limited search area is defined around the occurrence of the step in speed category. The outer limits of this area are decided with respect to old and new reference speed. For example, a low speed category results in a smaller area for the algorithm to investigate. Then the measured acceleration within this limited area is investigated to find where the reaction occurred. The first time the sign of the acceleration changes within this area a reaction spot is registered. An example of this is shown in Figure 4.6.

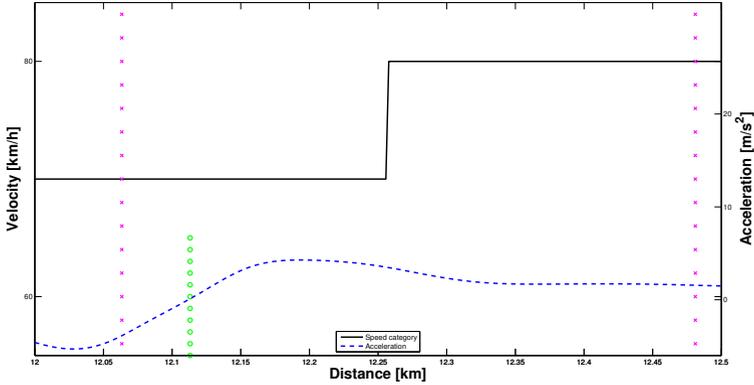


Figure 4.6: The figure shows a change in speed category, black solid line, from 70 km/h to 80 km/h. The pink "x-shaped" lines limit the area in which it is possible for the algorithm to find a reaction. The blue dotted line is acceleration. The reaction spot is found when acceleration changes sign for the first time within the specific region bounded by the outer limits, going from left to right. In this case from negative to positive. The reaction spot is marked with the green vertical "o-line". Left y-axis is velocity and right y-axis is acceleration.

4.7 Energy calculations

When the velocity profile generation has finished information regarding velocity, acceleration, road inclination and time is available. Using this information the resulting force, Eq. (3.5) is calculated assuming the following numbers for the road load coefficients.

Table 4.3: Coefficients used for the energy calculations

g	=	9.82	m/s^2
ρ	=	1.2041	kg/m^3
c_r	=	0.02	
m	=	2042	kg
A_f	=	2.3	m^2
c_d	=	0.315	
r	=	0.32	m

Following the remaining equations in Section 3.1 the energy used during the route is obtained.

The consequence of fixed coefficients is investigated by taking re-writing Eq. (3.5) and consider it as in Eq. (4.10).

$$F_{resulting} = Av^2 + B(g(C\cos\theta + \sin\theta) + a) \quad (4.10)$$

where $A = \frac{1}{2}A_f c_d \rho$, $B = m$ and $C = c_r$. To investigate how different coefficients would impact the final energy consumption A , B and C are varied one by one.

Finally, it is in this thesis assumed 100 % regeneration. The efficiencies used are

Table 4.4: Efficiencies used

$\eta_{battery}$	=	0.98
η_{EM}	=	0.85
η_{GB}	=	0.95

4.8 Tuning parameters

All the models above open up possibilities for a range of parameters that can be tuned to change the performance of the model. Table 4.5 contains a summary of all the different parameters that for each method can be changed. As described in Section 4.2 the TPM also includes parameters that can be varied to change the result of the three methods that uses the TPM.

Table 4.5: The table shows the parameters that can be changed to impact the result of the model.

Parameter \ Model	DD	GMM	Multi	ODE	ODEwNoise
Resolution on grid	x	x	x		
Sample in time/distance	x	x	x	x	x
Number of clusters		x	x		
Number of historical and future states			x		
Weights on historical and future states			x		
e and δ				x	x
Number of highest amplitudes					x
Amount training data	x	x	x		x

4.8.1 Parameters chosen

Regarding sampling in time relative distance, time sampling is chosen in this thesis. The time base, i.e. time between two samples, is 0.2 seconds. Furthermore, the amount training data in the TPMs is 5000 km. The remaining parameters that are presented in Table 4.5 and regarded as constants are declared earlier in the chapter. If a parameter is varied through out experiments, it is presented in Chapter 5.

4.9 Validation

The validation will be done on three different drivers. Each of the three drivers will be simulated on three different routes ten times each. From each route is four different validation values calculated: Relative error for velocity, δ_v , power, δ_p , change in energy with respect to distance, δ_Φ and frequency, δ_f .

To investigate the velocity prediction and energy estimations from the modeling methods and to be able to draw conclusions, four different values are produced, examined and discussed. The purpose of the four values are to separately give a measure on how well the driver model imitated the real driver. Three of the four final validation values are calculated by the same method but validates different inputs. The fourth validation values is slightly different. First in this section the calculations made to reach to the final value are presented. This is followed by a description of each validation input separately. The last two subsections contains which routes that are chosen for validation and how the robustness of the driver models are evaluated.

4.9.1 Theory

The four values are all calculated by the following calculation steps where x is the input that is changed depending on which validation value that is of interest. x^* is the estimated value. The relative error for the mean of the estimation δ_{μ_i} at each speed category i is calculated.

$$\delta_{\mu_i} = \frac{|x_{\mu_i} - x_{\mu_i}^*|}{x_{\mu_i}}. \quad (4.11)$$

The δ_{μ_i} are concatenated together with respect to the time t_i spent at each speed category. The result is then divided by the total time T to get a mean relative error δ_μ for the average value.

$$\delta_\mu = \frac{1}{T} \sum_{i=1}^{13} t_i \delta_{\mu_i}, \quad (4.12)$$

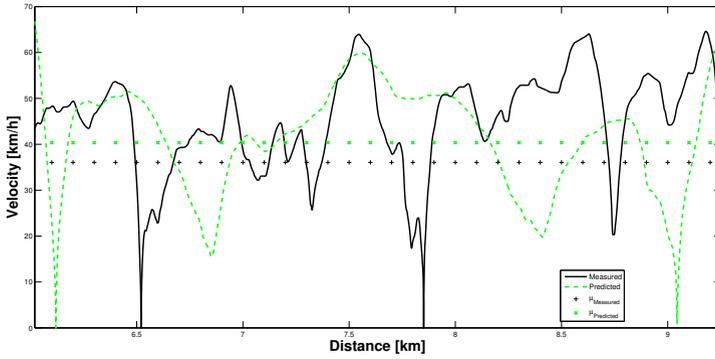


Figure 4.7: The figure shows measured velocity and the predicted velocity during a segment with speed category 40 km/h of a validation route.

The same procedure is done to find the mean relative error for the standard deviation δ_σ . These two values are then merged together by

$$\delta_x = \frac{\delta_\mu}{2} + \frac{\delta_\sigma}{2}. \quad (4.13)$$

The aim is to minimize δ_x .

In three of the four different validation methods δ_x will be replaced by δ_v , δ_P and δ_Φ respectively as shown in Eq. (4.13). In the fourth case, for δ_f , only the relative error for the average value is considered. That is, Eq. (4.13) is not included in the calculations.

The reason why both the average value and standard deviation are taken into account have originated from that the average value itself does not tell if the method manage to create an individual driver model. Instead a combination of the two values will give a hint if the predicted velocity will result in the same velocity, power and change in energy distribution as the measured data.

The reason why the average value and the mean of the standard deviation for each speed category are used instead of instantaneous values is due to that the aim is not to make an accurate velocity prediction at a specific moment or after a specific distance. An example of this is shown in Figure 4.7.

The figure shows a relatively good velocity prediction. Over the segment the prediction captures the driver behaviour when he or she brakes down to zero two times. If the validation had been done on the instantaneous relative errors the results have been very bad when the driver brakes down to zero. It is shown over the segment

that it is a correct behaviour to predict. But since when it happens not is of interest, the mean velocity and the mean standard deviation over the speed category are instead used for the relative error calculations.

A hypothesis is that the traffic pattern is not as dominant at freeways as in urban environments. The more the traffic pattern is dominant, the harder it is to do a personal driver model. To test this, two easy understandable values will be calculated for each route. The values are the mean value of the relative error for the change in energy δ_{Φ_μ} and velocity δ_{v_μ} for all simulations n as in

$$\delta_{x_\mu} = \frac{1}{n} \sum_{i=1}^n \delta_x^i, \quad (4.14)$$

such that each route gets two values that can be used as comparison.

4.9.2 Velocity

To be able to find a value to compare the velocity prediction, the predicted velocity and measured velocity is processed as described in Eq. (4.11), (4.12) and (4.13).

The purpose of investigating the velocity is due to three things. Firstly, to see how well the prediction imitates the measured velocity and since it is an easy interpreted physical quantity. Finally, how much the velocity influences the energy consumption. That is why it is interesting to see if the result from the velocity validation follows the same pattern as the result from the derivative of the energy with respect to time respectively distance.

4.9.3 Power

The second value to compare is the relative error of the power used during the route. To do so the same calculations are made as in the case above when the relative error for the velocity was calculated.

The average value of the power can by itself tell if the velocity prediction will give a good energy estimation or not, assuming that the difference between time traveled in the measured and predicted case is not to large. So the purpose of using the standard deviation in this case is for it to indicate if the aim to create an individual driver model is possible.

4.9.4 Change in energy with respect to distance

As a third value to examine the result from velocity prediction, this thesis investigate the change in energy Φ , which is the difference in energy between each time step, with respect to the distance traveled during the time step. To clarify, this method gives a measure of how many kilowatt-hours (kWh) that is consumed per kilometer

(km).

$$\Phi = \frac{\Delta E}{\Delta x} \quad \left[\frac{kWh}{km} \right] \quad (4.15)$$

This is done for every unique speed category resulting in an average and standard deviation value for the change in energy for every speed category. In the same way as for the velocity and power, a final relative error is then calculated for the energy.

As for the power in the subsection above, the average value for Φ can tell if a good energy estimation has been performed or not. The standard deviation can be used as a measurement of how well the driver profile is done. The scope is to minimize the standard deviation error $\delta_{\sigma_{\Phi}}$ as that means that the driver model captures the same fluctuations in energy.

4.9.5 Frequency

Finally, the fourth value to measure the methods ability to create a personal velocity profile is the average frequency in the power spectrum of the velocity signal. In other words this is a measure of how good the prediction methods manage to imitate the frequency of which the driver changes the velocity with in the training data. Following the procedure for finite discrete signal in Subsection 3.2.4 in the amplitude for N frequencies within a specific range is obtained. An example of a frequency spectrum is shown in Figure 4.8. For every speed category an average frequency is calculated with

$$\bar{f} = \frac{1}{A_{tot}} \sum_{i=1}^N A_i \cdot f_i \quad [Hz], \quad (4.16)$$

where A_i is the amplitude for the frequency f_i .

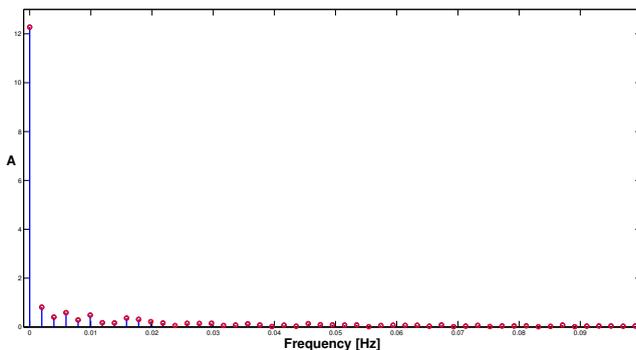


Figure 4.8: The figure shows the frequency spectrum up to 0.1 Hz. For frequencies above 0.1 Hz the intensity is almost non-existent. In this example 0 Hz is the most intense frequency.

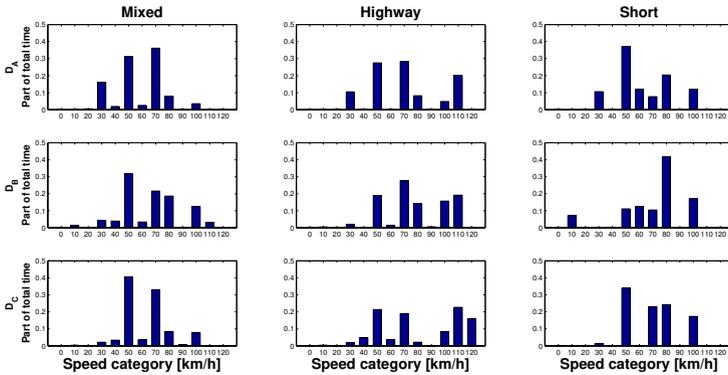


Figure 4.9: The figure shows the three drivers' validation routes as a histogram over the part of time spent at each speed category.

Following the same procedure as for the three previous values to validate the prediction, aside that no standard deviation is calculated for the frequency, a final value is obtained of how good the frequency prediction is.

The reason to investigate the relative error in mean frequency has its origin in the discussion regarding the results from Figure 2.2 and 2.3. The final value will not indicate anything about the energy estimation. Instead, together with the other three values from the subsections above, it can tell how well the methods manage to create an individual driver model.

4.9.6 Validation routes

As mentioned in the beginning of this section, three different drivers are to be used to validate the predictions. The drivers are referred to as D_A , D_B and D_C . The different validation routes are presented in Figure 4.9. To be able to do a good validation three different mixes of speed categories and distances were found for the three drivers.

Since the drivers have not driven on the same roads the routes can not be exactly the same. The goal was to find three similar routes for driver A, B and C. The routes are chosen as one mix of lower speed categories referred to as *Mixed* and one mix of higher speed categories referred to as *Highway*. The length of the routes are around 280 km. Since the drivers have not driven that far during one route, different routes have been concatenated and are referred to as one route of length 280 km. The route referred to as *Short* is one single route per driver at around 25 km that have a wide range of different speed categories.

4.9.7 Robustness of the driver models

To validate the robustness of the driver models, the personalized trained models are validated against another driver's validation routes. I.e. it is a test if the prediction goes wrong if a friend drives your car. The models are considered robust if the prediction goes wrong when doing this test.

The test will be performed on the three drivers' three validation routes. This means that a total of 18 simulations or six cross validations will be performed. For each simulation are the confidence intervals for the relative errors calculated and compared with the individual results.

To find out the answer if it is easier to model a driver with increased speed category, the same values as in the section above $\delta_{\phi_{\mu}}$ and $\delta_{v_{\mu}}$ are calculated from Eq. (4.14) where $n = 6$. These values are compared with the individual results.

4.10 Summary modeling

To summarize the chapter there is now five driver models to predict the energy consumption for a vehicle. The models predict the next change in acceleration to generate a velocity profile through the algorithm in Figure 4.2. Three of the driver models are based on a TPM that is trained in the same way for all three models. One of the models stores the values in the TPM as discrete distributions (DD). The second model utilizes Gaussian mixture models (GMM) to find out next the change in acceleration from the TPM. The third model is a development of the GMM and tries to predict the future by taking both the past and the consequences of a future step into account.

The fourth model is based on ordinary differential equations where a personal noise have been found by a frequency analysis (ODEwNoise) and the fifth method uses only ordinary differential equations (ODE) to find next change in acceleration. The input that is needed to predict the velocity is the speed categories and its corresponding distance vector.

The models will be validated with four different validation methods that all calculates the relative errors based on velocity $[v]$, power $[P]$, change in energy based on distance $[\Phi]$ or frequency $[f]$. To test the influence from the traffic pattern a fifth validation method will be used.

5

Results

In this thesis five different models to predict the velocity have been proposed with the purpose to be able to estimate the energy consumption during a known route. Different ways to validate the predictions have also been proposed and in this chapter the results from simulations will be presented when the predictions are compared with measured data as described in Section 4.9.

The first section in this chapter presents the results when the results from the driver models are compared to each other. Secondly, to evaluate robustness and if the aim to create a personal driver model is reached, the driver models are evaluated when drivers are paired with a TPM belonging to another driver. This is followed by the required memory capacity and computation requirements for the driver models. Finally what happens when some of the parameters presented in Table 4.5 are changed is presented.

The three drivers used for validation will be, referred to as D_A , D_B and D_C .

To be able to look at the exact values for the figures and tables, tables are provided in Appendix A.1. The confidence intervals are created by ten simulations, unless something else is declared, using Eq. (3.32).

5.1 Driver models

In this part of the chapter it is presented how the driver models perform compared to each other. The results in this section are from the training at 5000 km with sample time 0.2 s. In Subsection 5.1.1 the results from comparing DD, GMM, ODEwNoise and ODE are presented. It is followed by a separate investigation done on the Multi-step model due to its higher computational requirements.

5.1.1 Accuracy

Four different driver models are compared in this part of the result. The results for each validation method is visualized in Figure 5.1. Figure 5.1 shows that the Markov based methods are generally generating better results than the ODE based methods when using Velocity, Power and Change in energy with respect to distance as the validation methods. It is hard to draw any conclusion from the validation method Frequency since a majority of the intervals overlap.

To get a clearer view of which model that performs the best energy and velocity prediction, the mean value of δ_{Φ_μ} and δ_{v_μ} for each route is calculated and presented in Table 5.1. The table is then followed by Figure 5.2 to 5.5 which will be helpful in the discussion.

Table 5.1: The mean of the relative error for the change in energy per unit distance, δ_{Φ_μ} and velocity δ_{v_μ} concatenated for the three drivers at each route.

δ_{Φ_μ}	Mixed	Highway	Short	δ_{v_μ}	Mixed	Highway	Short
DD	0.1071	0.1122	0.2278	DD	0.1651	0.2163	0.6675
GMM	0.1000	0.0913	0.2775	GMM	0.1478	0.2066	0.5760
ODEwNoise	0.2278	0.1929	0.5839	ODEwNoise	0.4184	0.3584	0.3429
ODE	0.2962	0.2725	0.2573	ODE	0.4960	0.4541	0.4016

Regarding Table 5.1 both the change in energy and velocity validation tell that DD and GMM perform the best prediction if looking at the routes *Mixed* and *Highway*. By the two driver models, GMM gives the best prediction according to this table.

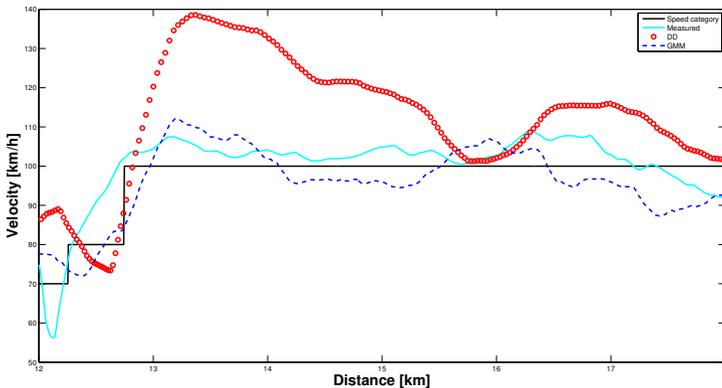
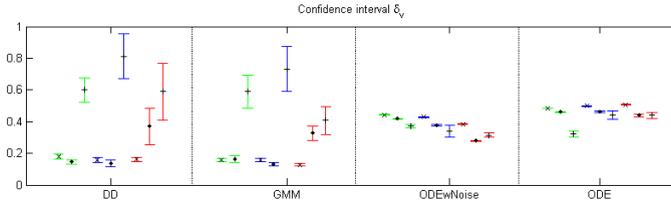
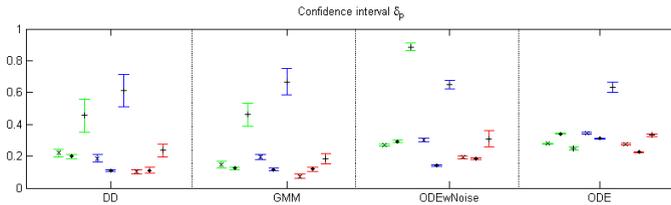


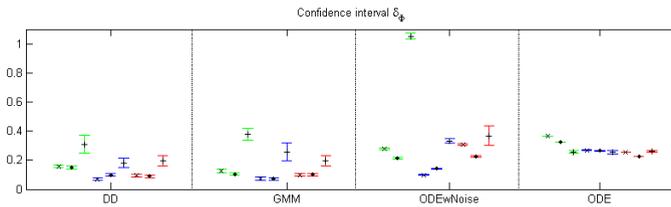
Figure 5.2: The figure shows the measured velocity for D_A and the predicted velocities by the driver models DD and GMM. The example is a part from the first out of ten simulations for D_A on the validation route *Short*.



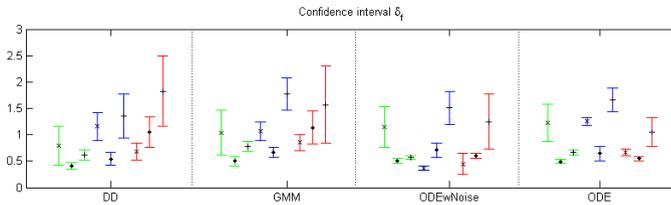
(a) Velocity



(b) Power



(c) Change in energy with respect to distance



(d) Frequency

Figure 5.1: The figures show the confidence interval of the relative error after ten simulations with three drivers on three routes per driver. Each figure is structured as follows: The figure is divided into squares which represents the model that has been used. In each square the drivers are color coded such that green is D_A , blue is D_B and red is D_C . The colors from left to right is [green green green blue blue blue red red red]. The symbol in the middle represents the routes which are described in Section 4.9: x is *Mixed*, \cdot is *Highway* and $+$ is *Short*. The exact values for the confidence intervals are found in Appendix A.1.

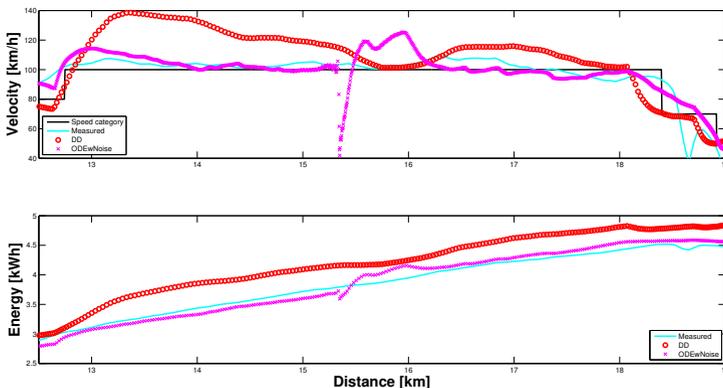


Figure 5.3: The figure shows the measured data and predicted results from DD and ODEwNoise. The upper subfigure shows the velocity profiles and the lower one the energy consumption so far after a specific distance. The example is from the third out of ten simulations on D_A Short.

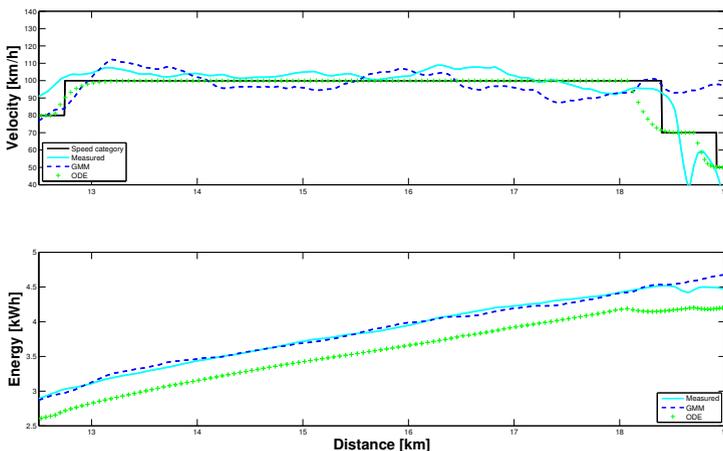


Figure 5.4: The figure shows the measured data and predicted results from GMM and ODE. The upper subfigure shows the velocity profiles and the lower one the energy consumption so far after a specific distance. The example is from the third out of ten simulations on D_A Short.

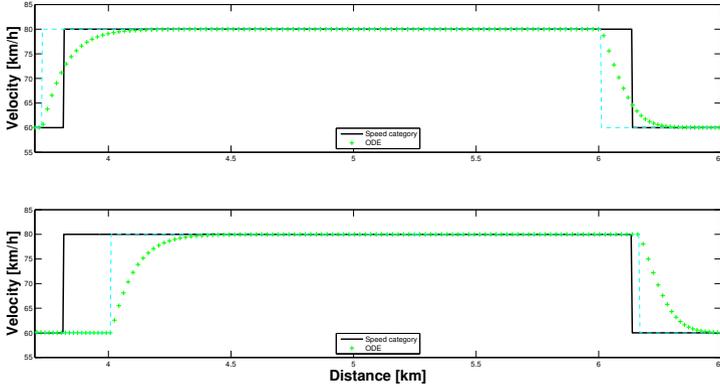


Figure 5.5: The figure shows the same road segment for two different simulations. It can be seen that different spots for reaction can happen due to the probability dimension in individual reaction to speed category. In this figure the dashed line marks the reaction spot which entails that the predicted velocity in the upper subfigure reach the target velocity before the lower one. Why there in fact are confidence intervals for ODEwNoise and ODE, for example in Figure 5.1 is explained by different reaction spots.

5.1.2 Investigation of Multi-step prediction

The investigation of the Multi-step model was due to its high computational requirements made separate. It was tested which combination of how many steps forward and backward and weights, that is the best to use. The simulation were done on D_A when the TPM had been trained on 5000 kilometers and the validation route was *Highway*.

The best number of backward and forward steps are decided by finding the combination that renders the minimum value of

$$\frac{\delta_v}{2} + \frac{\delta_\Phi}{2}. \quad (5.1)$$

Fifteen different combinations of backward, forward and weights were simulated and the five combinations that was best were further simulated five times each. The mean value and its 95 % confidence interval, I_δ , was calculated as $\bar{\delta} = \frac{1}{5} \sum_{i=1}^5 \left(\frac{\delta_v^i}{2} + \frac{\delta_\Phi^i}{2} \right)$ and by Eq. (3.32). The five best combinations are presented in Table 5.2.

Table 5.2: Result from the investigation of the Multi-step model.

Backward steps	Forward steps	Weight/Std	$\bar{\delta}$	$I_{\bar{\delta}}$
0	0	-	0.092	[0.087 0.097]
1	0	2	0.108	[0.102 0.113]
2	0	2	0.110	[0.103 0.116]
1	0	1	0.111	[0.099 0.128]
0	2	2	0.116	[0.106 0.126]

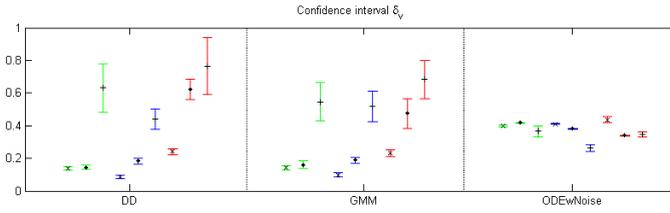
These results implies that it is better to use a model were neither the future or backward information are taken into account. Consequently, no further simulations will be done on the Multi-step prediction.

5.2 Robustness of the driver models

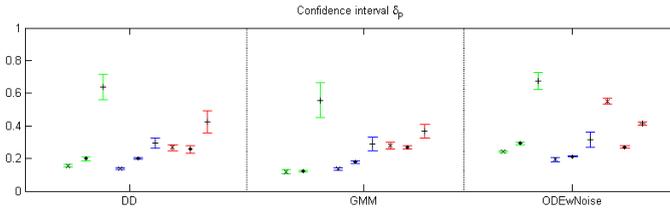
This part of the chapter will present what happens when a driver is validated against data collected by another driver. In other words the wrong TPM and wrong frequency analysis will be used on a set of validation routes from another driver. This can be interpreted as if a friend drives your car. The purpose is to further investigate if it is possible with the modeling methods chosen in this thesis to create a personal driver model.

The three drivers have been validated on each other such that total six cross validations. Since it would be difficult to visualize all the results in a figure three validations were chosen. The results in Figure 5.6 shows the result from when D_A has been driven on the validation routes of D_B and D_C and when D_B has been driven on the validation routes of D_C . Since the model ODE not is personalized it will not be included in these results.

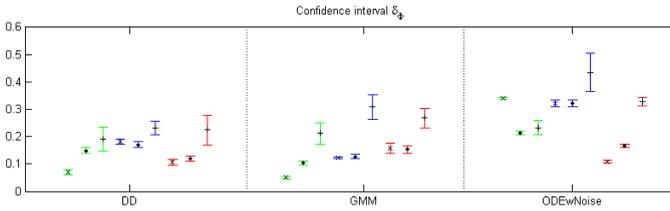
All the six cross validations are included in Table 5.4 where the result have been calculated in the same way as in Table 5.1. Table 5.4 is appropriate to use as a comparison to Table 5.1.



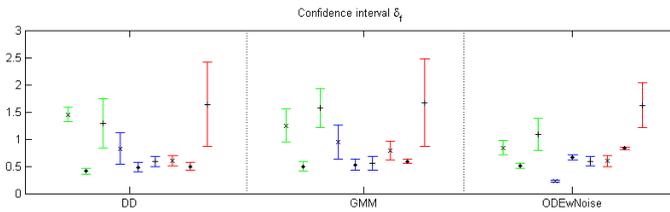
(a) Velocity



(b) Power



(c) Change in energy with respect to distance



(d) Frequency

Figure 5.6: The figures show the confidence interval of the relative error after ten simulations where the TPM and frequency analysis of D_A is driven on the validation routes of D_B (green) and D_C (blue). The red are the trained D_B driven on the validation of D_C . Each figure is structured as in Figure 5.1. The symbol in the middle represents the routes: x is *Mixed* · is *Short*.

An investigation of the confidence intervals that are attached in Appendix A.2 can be done as in Table 5.3. The table shows if the result from the individual simulation gave a better result than the cross validation [I], if the cross validation gave a better result [C] or if no conclusion can be drawn since to that the confidence interval overlaps [-]. Table 5.3 is the result of the trained D_A on the validation route of D_C for *Mixed* and *Highway*.

Table 5.3: Comparison of the confidence intervals of the cross validation of $D_A - D_C$ with the individual validation $D_A - D_A$.

<i>Mixed</i>	I_v	I_p	I_Φ	I_f	<i>Highway</i>	I_v	I_p	I_Φ	I_f	<i>Short</i>	I_v	I_p	I_Φ	I_f
DD	I	-	C	-	DD	I	-	C	I	DD	-	C	C	I
GMM	I	I	-	-	GMM	I	I	I	I	GMM	-	C	C	C
ODEwNoise	-	I	I	C	ODEwNoise	C	I	I	I	ODEwNoise	I	C	C	C

Table 5.4: The mean of the relative error of the energy, δ_{Φ_μ} and velocity, δ_{v_μ} concatenated for the three drivers at each route when simulated on wrong driver, total 18 simulations.

δ_{Φ_μ}	Mixed	Highway	Short	δ_{v_μ}	Mixed	Highway	Short
DD	0.1099	0.1292	0.2115	DD	0.1702	0.3049	0.5515
GMM	0.1153	0.1283	0.2704	GMM	0.1652	0.3015	0.5591
ODEwNoise	0.2695	0.2318	0.2960	ODEwNoise	0.4225	0.3608	0.3371

5.3 Memory and computation requirements

When making the prediction of the velocity profile the driver model take a different amount of time to perform the computations. This will be presented as the computational requirements. The models do also demand a different amount of memory space to make the predictions, implying that it takes an individual memory capacity depending on the modeling method.

To compare computation requirements for the different driver models, timing were made during simulation of the velocity profiles. The time for ODE is declared as one time unit and the other ones will be presented in a ratio to the results from ODE.

Table 5.5: Result from the computation requirements investigation.

Model	DD	GMM	ODEwNoise	ODE
Time ratio	2.22	8.05	1.10	1

The computation time for driver model Multi-step prediction varies depending on how many steps forward and backwards that is taken into account. In Table 5.6 zero step backwards and zero step forward is taking 1 time unit.

Table 5.6: Ratio for the time it takes to make the velocity prediction depending on how many steps forward and backwards that are used in Multi-step prediction.

Backward steps	Forward steps	Time ratio
0	0	1
1	0	1.28
0	1	1.76
1	1	2.63
5	0	2.72
0	5	5.51
5	5	10.31

Further on, the memory requirements were investigated. ODE is once again used as the method for comparison. The time step used to create the models is 0.2 s. The distributions in the Multi-step prediction is created with arrays of size $[1 \times 2200]$.

Table 5.7: Ratio for the memory required for the different driver models. DD_{500} and GMM_{500} depicts the models with less training data in subsection 5.4.1.

Model	DD	GMM	Multi-step	ODEwNoise	ODE	DD_{500}	GMM_{500}
Memory ratio	18.14	9.28	553.56	1.23	1	3.73	6.03

5.4 Tuned parameters

This section contains the results from when some of the parameters in Table 4.5 were changed. First is the amount of training data changed. That is followed by results from when different sample times were used. Finally, the result from how different road load coefficients impact the final energy consumption.

5.4.1 Amount training data

The investigation of what happens if the amount of training data is the tenth of the training data in Section 5.1, is presented in Figure 5.7 and Table 5.9. The validation route *Short* is not included in this test because of that it didn't give a good result on the large amount of training data. The confidence intervals for all the models can be found in Appendix A.3.

Figure 5.7 shows the confidence intervals for the models that use the TPM and are structured such that it should be easy to compare with the result of the larger amount of training data. ODEwNoise is not included in the figure since it would misfit the figure. The result of ODEwNoise is included in Table 5.8.

Table 5.8 is appropriate to use as a comparison to Table 5.1. The comparison

shows that the mean relative error value of the energy consumption and velocity is worse when a smaller amount of training data is used.

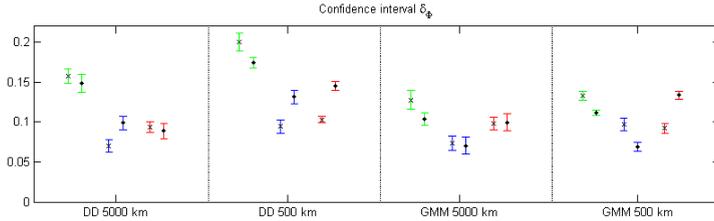


Figure 5.7: The figure shows the confidence intervals of the change in energy with respect to distance. The squares are structured as in Figure 5.1, and as mentioned the route *Short* is not included.

Table 5.8: The mean of the relative error for the energy, δ_{Φ_μ} and velocity δ_{v_μ} . Where the three drivers results have been concatenated at each route with a small amount of training data.

δ_{Φ_μ}	Mixed	Highway	δ_{v_μ}	Mixed	Highway
DD	0.1327	0.1503	DD	0.1786	0.2199
GMM	0.1075	0.1048	GMM	0.1655	0.2085
ODEwNoise	0.2341	0.2777	ODEwNoise	0.4351	0.3901

5.4.2 Sample time

The sample time was changed and D_B was simulated 10 times. The remaining parameters remained as in Subsection 5.1. The mean value of δ_v and δ_Φ was calculated for each route and method. ODE is independent of the sample time so it is not included in Table 5.9.

Table 5.9: The table shows the relative error of energy with different sample time at validation routes *Mixed* and *Highway*.

δ_Φ	Mixed				Highway			
	0.1	0.2	0.5	1	0.1	0.2	0.5	1
Sample time [s]								
DD	0.064	0.071	0.070	0.091	0.096	0.099	0.109	0.130
GMM	0.121	0.080	0.073	0.082	0.068	0.071	0.066	0.078
ODEwNoise	0.185	0.093	0.098	0.169	0.154	0.141	0.110	0.203

Table 5.10: The table shows the relative error of velocity with different sample time at validation routes *Mixed* and *Highway*.

δ_v	Mixed				Highway			
Sample time [s]	0.1	0.2	0.5	1	0.1	0.2	0.5	1
DD	0.186	0.156	0.131	0.129	0.147	0.135	0.123	0.139
GMM	0.153	0.157	0.187	0.207	0.156	0.132	0.118	0.159
ODEwNoise	0.389	0.428	0.466	0.489	0.312	0.376	0.391	0.403

5.4.3 Road load polynomial

As mentioned earlier in the thesis the coefficients in the road load polynomial are fixed even though they may vary during a route in reality.

To investigate how the fixed coefficients impacts the result of the energy prediction a route at 25 km was simulated. The coefficients A , B and C were one by one multiplied with a scale factor $\in [0.5, 1.5]$ and the final energy consumption of the route was calculated. The result is shown in Figure 5.8. To refresh the reader's memory, $A = \frac{1}{2}A_f c_d \rho$ [kg/m], $B = m$ [kg] and $C = c_r$.

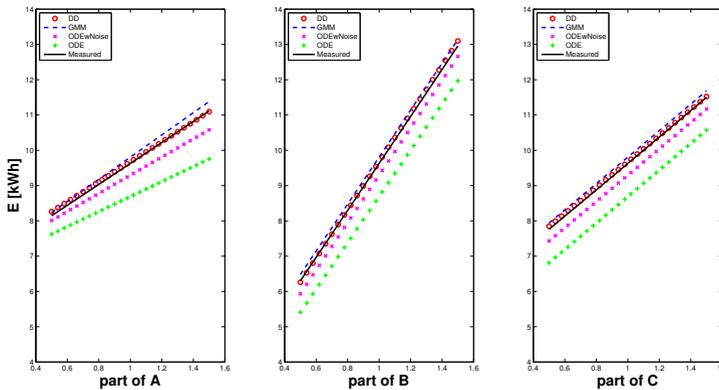


Figure 5.8: The figure shows the final energy consumption with respect to the scale factor for A , B and C respectively. How the final energy consumption varies for different A is shown in the left figure, for different B in the central figure and for different C to the right. The dotted line is the energy for the discrete distribution, "x-shaped" line the result for Gaussian mixture model, "o-shaped" line shows the ordinary differential equations with added noise, "+-shaped" line the energy for the pure ordinary differential equation and finally the measured energy is illustrated with a solid line.

6

Discussion

The chapter begins with a discussion about the results from the chosen validation methods and what impact general error sources may have. That is followed by a section about how the different driver models perform and then a discussion if they managed to become personalized.

The chapter continues with some sentences about the memory usage and the computational complexity. Then comes a section with thoughts about how different parameters chosen affect the achieved results.

The final section holds a discussion about ideas and thoughts for future work on this subject.

6.1 Validation methods

In Figure 5.1 the results from the prediction simulations for the drivers individually are illustrated in four different sub-figures. Each figure shows the results from a specific validation method. The following section will hold the discussion with the advantages and disadvantages with the different validation methods. The consequence of the discussion is that the result from the validation methods velocity and change in energy with respect to distance will be in focus for the remaining part of the discussion.

6.1.1 Velocity

In Figure 5.6a, the results on how well the driver models imitate a driver's actual velocity distribution during a route is shown. A disadvantage that can be seen with this validation method is that a good or bad result doesn't guarantee a good or bad energy estimation. More about that in Subsection 6.1.2. Even though this weakness exists, the validation method is of interest for the remaining part of the discussion.

From a perspective if it is possible to make a personalized driver model the velocity is the easiest one to check on. The result shows that it is a big difference between different drivers velocities. The velocity is also easy to understand and relate to.

6.1.2 Power

Secondly in Figure 5.6b, the results from the power estimation follows. Interesting here is that the bad velocity prediction by DD for D_C on the *Highway* doesn't seem to affect the quality of the power estimation. Even though the velocity prediction is wrong, the mean power during the speed category could still be correct due to accelerations or even road inclination. Let's say the prediction is at constant velocity above the mean of the measured velocity. Then the relative velocity error is weak. But if the measured velocity is fluctuating at different speeds which cost power due to the accelerations the relative power error can still be good.

Another circumstance that affects the power estimation is the road inclination. If the road inclination increases it gets more dominant in the road load polynomial than the velocity or acceleration. The same discussion can be held around the relatively good velocity prediction by D_B for ODEwNoise at the short route even though it results in a weak power estimation.

Furthermore, a good power estimation doesn't necessarily mean a correct estimation of the energy consumed. For that to happen, the time traveled would have to be around the same value for both prediction and validation on each speed category. I.e, a good velocity prediction together with a good power estimation would result in a good energy estimation. Another aspect that is worth taking into account is that the power can be seen as it holds the information from both the velocity prediction and change in energy per unit distance. See the following equation where the units for the two validation methods results in power.

$$\frac{kWh}{km} \cdot \frac{km}{h} = kW \quad (6.1)$$

Investigate Figure 5.1 together with the tables in Appendix A.1. Then it can be seen that the intervals for power and change in energy with respect to distance rank the modeling methods in the same way on *Mixed* and *Highway* routes unless, in the few cases, where the intervals overlap and no conclusion can be drawn. This together with that the power itself doesn't tell if the prediction results in the correct energy estimation leads to that the validation method below, "Change in energy with respect to distance", will be used as the measure of how good the energy estimation is.

6.1.3 Change in energy with respect to distance

In the same way as for the power validation, the results from change in energy with respect to distance, doesn't always follow the same pattern as the result when the velocity was evaluated. This can be seen in Figure 5.6c. That is why a separate investigation of the results from the velocity and this validation method is interesting. It will give a measure on how individual the velocity profile created is and in the same time how good the energy estimation is.

As mentioned earlier in this thesis the distance to travel is known, implying that an integration of the average value for change in energy with respect to distance will be the total energy consumption. The fact that, in the end, it is the energy consumption that is of the greatest interest, the result from change in energy alone will be discussed in some cases. This is also the reason why this measurement suits best regarding computing distance to empty.

6.1.4 Frequency

Regarding the fourth method used to validate the results from the prediction it is hard to draw any conclusions from the results in Figure 5.6d. For every comparison to point out what modeling method that is the best, the intervals overlap. The only exception is for when D_B drives on the *Mixed* route. Then ODEwNoise perform best but no ranking can be made among the remaining driver models. Since the validation method doesn't contribute to be able to rank the driver models, nor estimate the energy consumption, the validation method won't be regarded in the following discussions.

A comment regarding that ODEwNoise doesn't perform the best results would be suitable though. Firstly, it can depend on that the differential equations and the noise counteract each other in some time samples. This since the differential equation will calculate an acceleration taking the velocity closer to the drivers average velocity on that specific speed category while the noise is calculated with respect to time and speed category. Secondly, the chosen sequence from the training data may not look similar to the validation data.

6.2 General error sources

Error sources that consider all of the predicted values are measurement and numerical errors. Also the risk that more than one driver has been present in the training and validation data is possible. Since the route *Short* was chosen only with respect to its distance and speed categories it is possible that it has been driven by a driver that is not responsible for the majority of the training data. This is not the case for the other two validation routes since they are several shorter routes merged together. The consequence from this is that the route *Short* is more sensitive to this

error source than routes *Mixed* and *Highway*.

Considering the overall weak results from the Markov based models on the validation route *Short* it is likely due to that the distance validated on is too short. The distance traveled on the other two validation routes is ten times longer making it possible for the models to compensate for the weak estimations that happens due to stochastic events.

Regarding measurement errors it shouldn't affect our results since it should be spread over both the training and validation data. However, it should affect if this was to be implemented in reality. This since the actual energy consumption will differ from the one predicted in the driver model. When it comes to numerical errors they are present. For example are both the distance and power computed using the numerical method *Cumulative trapezoidal numerical integration*, implying that the value calculated will not be the same as in reality. Neither is the acceleration the same as in reality since it is assumed to be linear between two samples. These numerical errors should be worse, the larger the time base gets.

6.3 Driver models

This section presents separate discussions regarding the five different driver models. The results that are discussed are from Section 5.1.

6.3.1 Discrete distribution

The model based on discrete distribution performs very well at the longer routes. It is almost as good as the GMM and better than the two models based on ODE in all the validations besides frequencies where they are similar. The reason why it performs a bit worse than GMM may be because DD can only adopt a change in acceleration that has been present during the training routes. Since the training routes only are 5000 km some states will not have enough information. This problem could be solved for the GMM since a distribution is created and makes it possible to adopt changes in acceleration that has not occurred during training. For example that may be the reason why the predicted velocity by DD in Figure 5.2 accelerates up to almost 140 km/h in the beginning of the road with speed category 100 km/h.

The distance interval shown in Figure 5.2 catches almost all of the road segment of 100 km/h for this prediction. The behaviour by the predictions for DD and GMM is reflected in their relative mean errors for the velocity, δ_μ . DD has a fault of around 10.4 % and the fault for GMM is around 3.7 %.

Also when comparing the velocity confidence interval for D_A when using DD

compared to ODEwNoise the velocity prediction is weak for the *Short* route. Even though that is the case, DD performs a better estimation of the change in energy per unit distance. In Figure 5.3 this is shown. In the lower figure of the two, the predicted energy consumption for DD is more parallel with the measured energy consumption. The relative mean error for change in energy is for DD 45,5 % and for ODEwNoise 52.8 %. This may be due to two things. First that DD predicts the acceleration better. Secondly that the velocities and the accelerations occurred at the same road inclination as the velocities and the accelerations did for the measured data which result in a more similar power demand.

6.3.2 Gaussian mixture model

The Gaussian mixture model performs very similar results to the DD. Since both methods are based on the same TPM it is no surprise that the results are similar. Table 5.1 shows that the GMM predicts the energy consumption best during the longer validation routes. But at the short validation route, for D_A regarding change in energy per unit distance, both DD and ODE gets better results. That the ODE would perform better on the short route wasn't expected but could be explained by the behaviours in Figure 5.4. On speed category 100 km/h, GMM predicts that the velocity would be higher and fluctuate more than the measured velocity does. The higher velocity and accelerations means that more energy will be consumed.

In the same case when D_A drives on the short route, the velocity prediction is worse for GMM than it is for ODE. If comparing the relative mean error for the two driver models, for the ten simulations, it can be seen that they have almost the same expected velocity value. But GMM has a larger standard deviation telling us that ODE on the short route is better at imitating the velocity behaviour from the measured velocity. Since the velocity predicted by the ODE is only following the speed category, except during speed category steps, GMM is probably having too large changes in velocity during the short route. Another obvious option could be that the driver in the chosen short route is not driving as he or she has done during the training routes.

When investigating the change in energy per unit distance for D_C on the short route, GMM gets better results. For the same route the intervals for the velocity prediction overlap but the interval is wider for GMM meaning there is a greater insecurity for GMM. That is obvious since GMM is probability based and ODE has the same pre-conditions for every simulation, see the discussion held in Section 6.1.

Regarding the discussion held in the above Subsection 6.3.1, about that DD accelerates to almost 140 km/h in Figure 5.2, it seems like a good choice to implement a 3-cluster Gaussian mixture distribution instead of discrete distributions in the TPM.

The GMM is the best model to use if the route is long according to the results in Table 5.1. The result shows that it is better both when the route has lower speed categories, typically city driving and when the route has higher speed categories. This indicates that the GMM is a good model to use to predict energy. It can be wrong for a shorter route but in the long run is it the most effective energy prediction method.

6.3.3 Multi-step prediction

That the backward step prediction does not make the model better suits well into the hypothesis that every decision is taken from where the driver is right now and that the history doesn't make any difference for the driver's next decision. The forward step prediction does not either improve the model. Since it adds a "future thinking" where the driver evaluates the consequences, the choice of acceleration he or she is about to choose the expectations was that the forward step prediction should make the model better. E.g. can the driver see if he or she chooses a high acceleration some samples before the speed category will change down and then instead chose a lower acceleration. But this is not what the results shows. The future steps does a worse performance than the backward steps.

It is not obvious to say why the prediction does not perform better when using the forward prediction. The main reason though may be that every forward step is estimated compared to the backward steps which have occurred earlier in the simulation. The fact is that for every forward step taken, there is one more error source added to the equation. A possible solution to the minimize the error which occurs due to estimations could be to tune the parameters for the future step better.

6.3.4 ODE with noise

If the results from the velocity validation is investigated it can be seen that ODEwNoise gets best result on the route called *Highway*. This is probably due to a reason mentioned earlier in this thesis. I.e., on roads with higher speed category you are more likely to drive the way you desire. Since the method used to implement this driver model makes a frequency analysis of the training data it is likelier, with the argument above, that the frequencies on roads with higher speed category is more periodic recurrent than the ones in urban traffic. Worth noticing also is that ODEwNoise generally is a better method to use to predict the velocity than ODE but worse than DD and GMM. Regarding the change in energy with respect to distance DD and GMM still makes better predictions.

6.3.5 ODE

The ordinary differential equation model is performing worst of all models at the longer routes. According to the result in Table 5.1 the mean relative error of the

change in energy over the change in distance, $\delta_{\Phi_{\mu}}$ is for city driving almost three times the relative error for the Markov based models. The GMM is 10.0 % wrong where ODE is 29.6 % wrong. The result on *Highway* is better but still very bad for the ODE, 27.25 % wrong.

The main reason why ODE performs bad in city driving is that it misses all the stops and starts, that happens very often when driving in city traffic. Starts and stops are, as described earlier, very crucial for the energy consumption. It should be interesting though to see how the ODE perform in a simulation where the location of stop signs and traffic lights is known.

The ODE model is a model that is good to use if the driver often uses the cruise control, i.e. very small accelerations, as long as the driver drives in a velocity close to the speed category. A behaviour similar to this could be seen in Figure 5.4, where the ODE predicts a better energy consumption than GMM: The cruise control is most common to use when there is no other traffic that disturb, typically on free-ways. According to the results the ODE is not performing well on the validation route, *Highway*. But it did very well compared to the other driver models on the validation route *Short* when investigating the velocity results. The reason for that is likely common to the discussion held in Subsection 6.3.2. In other words, the other driver models has a tendency to fluctuate too much on this chosen validation route where it looks like that the driver has been driving relatively stable.

If investigating the results from ODEwNoise and ODE their confidence intervals is much more narrow than the corresponding ones for DD and GMM. That is because those methods will predict almost the same velocities every time. The only changed condition in the ten simulations is the individual reaction to speed category. An illustration of the difference can be seen in Figure 5.5. This will result in that the velocity in some cases will have reached the target velocity already passing the speed category sign while it in other cases start accelerating in the same moment as driving by the new speed category sign.

6.4 Robustness of the driver models

In Chapter 5 there are results from the cross validation. The cross validation can be explained as there are two different drivers, D_B and D_C , driving the car belonging to D_A . D_C is also trying out D_B 's car. That means that the results from when using the wrong driver should be compared to the results from D_A and D_B in Figure 5.1.

To investigate this carefully, the tables presented in Appendix A.1 and Appendix A.2 that correspond to each other will be compared. For example, the confidence interval created when D_A was evaluated on his or her own *Mixed* validation route

is compared with when D_B drove the car belonging to D_A on the *Mixed* validation route for D_B .

When this is done for when D_B drove D_A 's car it is hard to say that a personal driver model is created for D_A . This since, if looking at *Mixed*, *Highway* and *Short* route separately, only the confidence interval for the frequency validation for ODEwNoise performs better when using the correct driver for the three different routes. And according to the discussion held in Section 6.1 regarding frequency validation, those results are not of interest.

If continuing by comparison of the results when D_C drove the car belonging to D_A the goal to create an individual driver model seems a bit more realistic. The result of this comparison is shown in Table 5.3.

In this table the intervals for velocity show better results for DD and GMM when driving on both the *Mixed* and *Highway* route with the correct driver. In fact, GMM with the correct driver performs better with all of the four validation methods. But still there are some intervals that overlap each other, implying that no conclusion can be drawn. Even some validation methods indicate that the "wrong" driver suits that driver model better. For example, when using DD and looking at energy per unit distance on both the *Mixed* and *Highway* route.

The third comparison made were when D_C drove D_B 's car. If referring to the discussion in Section 6.1 and thereby only looking at the interval for velocity and energy per unit distance the results from *Mixed* and *Highway* indicate that a personalized driver model has been created using DD and GMM. The same conclusion can't be drawn on the short route since the interval overlap during the same comparison.

The results from the three confidence interval comparisons indicate that it is possible to create a personalized driver model for at least one driver by using the driver models DD and GMM.

To further test the hypothesis that the driver models are more robust on free-ways than city driving, the mean value of the relative velocity error for all the six cross validations at both *Mixed* and *Highway* can be compared with the individual corresponding as in Table 6.1.

Table 6.1: Comparison of the velocity error from table 5.1 and table 5.4. [I] is from the individual validation and [C] is the cross validation

$\delta_{v,\mu}$	Mixed [I]	Mixed [C]	Highway[I]	Highway[C]
DD	0.1651	0.1702	0.2163	0.3049
GMM	0.1478	0.1652	0.2066	0.3015
ODEwNoise	0.4184	0.4225	0.3584	0.3608

The comparison shows that both DD and GMM are robust for drives at higher speed categories. On the other hand both models are less robust in city traffic but the relative errors are better. This shows that the traffic pattern is more dominant than the personal drive style in city traffic because the models manage to do pretty good predictions when wrong driver drove in lower velocities. It is logical that the traffic pattern is more dominant in city traffic due to traffic light, stop signs and zebra crossings etc. From this a conclusion can be drawn: It is more important that a personal trained model is used when driving at high speed category than on low speed categories. The model ODEwNoise is not a robust method due to this results.

6.5 Memory and computation requirement

In Section 5.3 the results from computation requirements and investigation about the amount of memory that the models need individually are presented. In Table 5.5 it can be seen that GMM takes the longest time to perform the prediction. The reason for that is that the distributions are saved in a few variables, meaning that calculations need to be made to first create a probability function and then choose a change in acceleration, Δa , probability based. In the case for DD, that is the second slowest, the choice of Δa can be made directly. Worth noticing is that ODEwNoise is almost as fast as ODE even though noise calculations are made in every time step.

Regarding the time result from the Multi-step prediction in Table 5.6, it can be seen that the forward step is much more time consuming than the backwards steps. The reason for that is that the distribution backwards already exists and the forward distributions are created in an iterative process for every step forward. As seen, this is time consuming.

In Table 5.7 the ratio between the memory requirements are presented. ODE is the driver model that needs least memory. The only information required is e and δ from Eq. (3.23) and information where an acceleration should start due to a change in speed limit. The Multi-step function is by far the most memory consuming method. It is due to the large arrays saved in every state. The memory required by GMM is much less than the Multi-step function. This is due to the variables mentioned in Subsection 4.3.2 which are used to save the large arrays for Multi-step in

a smaller format. This could also be applied in the Multi-step prediction if one is ready to sacrifice time. Another option could be to make an optimization about the size required for the array in every state.

Worth mentioning in the discussion is also that the driver models that takes advantage of the TPM are more sensitive, regarding memory requirements, to changed sampling time. At a shorter sampling time it is likelier that more states gets some information implying that the memory requirements will increase. The opposite applies for a larger sampling time.

6.6 Tuned parameters

In addition to the main simulations made to find what driver model that is the best and also if they become individual, some tuning parameters were changed. The discussion regarding the results presented in Section 5.4 are presented in the following sections.

6.6.1 Amount training data

An investigation where the amount of training data was decimated by a tenth, i.e. it was trained on 500 km instead of 5000 km was done. The results for the Markov based models were better than expected. The expectations were that the result should go more against the same result as the ODE, i.e. less individual. Recall, if a state in the TPM not has any information, which is more probable if less training is done, the ordinary equations are used.

Figure 5.7 compare DD and GMM on the *Mixed* and *Highway* validation routes. The comparison is based on the change in energy per distance. The figure shows that the DD is more sensitive to the amount of training data. The confidence interval indicates that the prediction gets better using the batch with more training data. This is true for all the three drivers at both routes. The GMM is not at all that sensitive, it seems like that model faster finds the correct prediction. Only two of the six confidence intervals show that it is better to use the big amount of training data.

These results are interesting in some different ways. From the earlier discussion it is said that the GMM is the best model to use but the DD is often very close in the results. The result from this investigation points on another maybe more important factor why to choose the GMM. Storage is expensive and if it can be decimated it is very beneficial. The results from the memory investigation in Table 5.7, shows that the size of the GMM is decreased by 35 % when a tenth of the historical data is used. The DD is decreased by a factor of three. DD is not as memory consuming as GMM when the training data is reduced by a tenth. This is because of if a state in the GMM is trained it can not grow larger than the number of parameters that are

needed to make a n-cluster distribution. On the opposite DD grows larger for every unique change in acceleration.

If the vehicle is used by a family such that say five different persons uses the car, the storage is of course more important such that every driver can have its own driver model. If the storage is unlimited it is still a benefit if the GMM can be used since if a lot of persons uses the car it is more likely that some persons don't drive the car that often and they should also have a well performing driver model.

6.6.2 Sample time

The sampling time used is important. This can be explained with the following two extremes. And remember that what is stored in a specific state in the TPM is the change in acceleration, Δa .

Assume that the sample time is very short. Then will the Δa be very small, close to zero, since between the samples no bigger change in acceleration will occur. This could be okay since during simulation, the same short sample time is used. The problem with too small sample times are that disturbances will be very dominant. E.g. if you drive on a patchy road the driver model will catch the small velocity differences that occur. These small velocity differences should be seen as disturbances and filtered by the sample time.

On other hand if the sample time is big, there is one main problem. It is that the training may miss changes in velocity. During one sample it is possible that the vehicle have both decelerated and accelerated. An example, if driving at 30 m/s at time 0 with sample time 2 s. The driver decelerates to 29 m/s during the first second and then accelerates to 30 m/s during the second second. Then due to the sample time, no change in acceleration has occurred when it in reality have both regenerated and generated energy. This behaviour is important to catch in a driver model. It is also important to remember that a larger time base will result in fewer samples in each state, which will deteriorate the results but it consumes less memory.

The results from the investigation of the sample time in Subsection 5.4.2, shows that the best results are found when the sample time is between 0.2 s and 0.5 s. This is reasonable due to the discussion above and since the reaction time for a healthy person is around 0.5 s. It seems like the ODEwNoise is a little bit more sensitive to a higher sample time than the other models according to its energy prediction.

6.6.3 Road load polynomial

In the region that is illustrated in the Figure 5.8 it can be seen that for every modeling method it is a linear behaviour and that the lines do not intersect. This is important since that means that even if an estimation of the coefficients would be

wrong, the ranking between the methods would be the same.

Worth noticing in the figure mentioned, is that the interval chosen to scale coefficients A , B and C is wide enough. For example for the mass $\pm 50\%$ means that the mass is varied around 1000 kg. For the rolling resistance, $\pm 50\%$ correspond to covering the range from new asphalt to driving on cobblestone.

6.7 Further work

During the course of this thesis there are many questions that have appeared and never been answered. A majority of these are presented in the following section.

To begin with it would have been interesting to see if the Multi-step model can be better implemented. One option that we think would be beneficial for the model would be if the distributions were merged together as in our implementation. Then, after the merge a new Gaussian mixture distribution can be found by the EM-algorithm as we did when we found the TPM for the GMM. By this method a probably better distribution can be found than by add the distribution as we did. But with this new model the computer complexity increases which is a disadvantage. If this works though, the TPM had been as memory consuming as the GMM.

Another way to predict the velocity could have been done by using Hidden Markov models. Then the question is how this would have been done and if it would have been a good modeling method. Our thoughts about doing a Hidden Markov Model comes from that it can build up the TPM and the states it needs by itself. We suggest that the states are hidden and that they are built up by time when it is trained with a sequence of acceleration inputs. This needs a lot of more work on but we think it is an interesting thought.

Regarding a discussion about which way to sample, it would have been interesting to compare the results from this thesis with results when measurements were taken after a specific distance. Also a further investigation on how much training data that would be the best could also be of interest for future work.

Another approach to estimate the energy consumption could be to identify what modeling method that gives the best result on a specific speed category. That way, during prediction, a model to predict the velocity could be chosen with respect to the speed category. E.g, if the driver always uses the cruise control when driving on freeway, the ODE at the drivers mean velocity would be a very cheap implementation to do.

If continuing on the reasoning about predicting the velocity, another approach

could have been to use different TPMs depending on if the process is in an acceleration mode around a speed category change or in a "steady state", between two steps in speed category. A final proposal how to predict the velocity using the TPM is that for an arbitrary part of the samples, the ordinary differential equation gets to choose the new acceleration independently of the TPM. This can be seen as a new type of noise for the ODE.

Finally, it is mentioned that the algorithm to predict the reaction to a new signed speed category may be a primitive way to find the reaction spot. The method has its shortcomings since it is for example is not always able to find the spot. The question is if a more advanced algorithm to find were a specific driver react to a new speed category would improve the results.

7

Conclusion

It is beneficial to use a driver model based on historical data. Generally the driver models based on Markov models: Discrete distributions (DD) and Gaussian mixture model (GMM) and Multi-prediction give the best velocity predictions and energy estimations. A comparison between these three driver models suggest that GMM is the best driver model to implement in the aspect of energy consumption estimation. The downside with DD and GMM is that they demand more memory to store historical data and it takes longer time to create the prediction.

The driver model based on frequency analysis did not give a good result compared to the models based on Markov models.

The Markov based models showed that an individual driver model can be trained such that they do a clearly worse result when they are driven by another driver at higher speed categories. The traffic pattern was showed to be more dominant in city traffic and so it was hard to make a personal driver model in city traffic.

A

Appendix

A.1 Driver models

Tables that contain the confidence intervals in Figure 5.1.

Table A.1: Confidence intervals for the different modeling methods when validating D_A on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1617, 0.1948]	[0.1951, 0.2450]	[0.1485, 0.1663]	[0.4320, 1.1569]
GMM	[0.1477, 0.1695]	[0.1274, 0.1661]	[0.1163, 0.1396]	[0.6159, 1.4687]
ODEwNoise	[0.4398, 0.4456]	[0.2663, 0.2778]	[0.2719, 0.2841]	[0.7655, 0.5344]
ODE	[0.4815, 0.4858]	[0.2775, 0.2823]	[0.3647, 0.3684]	[0.8764, 1.5815]

Table A.2: Confidence intervals for the different modeling methods when validating D_A on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1306, 0.1583]	[0.1854, 0.2127]	[0.1379, 0.1598]	[0.3517, 0.4674]
GMM	[0.1398, 0.1826]	[0.1169, 0.1301]	[0.0956, 0.1113]	[0.4064, 0.5899]
ODEwNoise	[0.4164, 0.4216]	[0.2850, 0.3001]	[0.2055, 0.2177]	[0.4637, 0.5477]
ODE	[0.4578, 0.4638]	[0.3389, 0.3439]	[0.3248, 0.3274]	[0.4508, 0.5384]

Table A.3: Confidence intervals for the different modeling methods when validating D_A on the *Short* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.5230, 0.6778]	[0.3494, 0.5603]	[0.2474, 0.3687]	[0.5158, 0.7109]
GMM	[0.4821, 0.6947]	[0.3889, 0.5334]	[0.3388, 0.4200]	[0.6888, 0.8809]
ODEwNoise	[0.3616, 0.3842]	[0.8647, 0.9097]	[1.0320, 1.0708]	[0.5370, 0.6001]
ODE	[0.3036, 0.3418]	[0.2393, 0.2594]	[0.2490, 0.2657]	[0.6109, 0.7135]

Table A.4: Confidence intervals for the different modeling methods when validating D_B on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1396, 0.1717]	[0.1637, 0.2097]	[0.0625, 0.0777]	[0.8890, 1.4261]
GMM	[0.1469, 0.1673]	[0.1799, 0.2095]	[0.0642, 0.0824]	[0.8951, 1.2417]
ODEwNoise	[0.4251, 0.4314]	[0.2937, 0.3135]	[0.0939, 0.1014]	[0.3264, 0.4080]
ODE	[0.4940, 0.5016]	[0.3400, 0.3490]	[0.2660, 0.2717]	[1.1798, 1.3314]

Table A.5: Confidence intervals for the different modeling methods when validating D_B on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1147, 0.1557]	[0.1035, 0.1136]	[0.0907, 0.1071]	[0.4300, 0.6607]
GMM	[0.1217, 0.1422]	[0.1089, 0.1260]	[0.0597, 0.0816]	[0.5672, 0.7694]
ODEwNoise	[0.3713, 0.3800]	[0.1366, 0.1483]	[0.1360, 0.1452]	[0.5747, 0.8417]
ODE	[0.4589, 0.4665]	[0.3093, 0.3149]	[0.2632, 0.2667]	[0.5109, 0.7785]

Table A.6: Confidence intervals for the different modeling methods when validating D_B on the *Short* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.6716, 0.9530]	[0.5123, 0.7157]	[0.1470, 0.2122]	[0.9424, 1.7735]
GMM	[0.5914, 0.8729]	[0.5831, 0.7505]	[0.1940, 0.3182]	[1.4703, 2.0724]
ODEwNoise	[0.3030, 0.3781]	[0.6236, 0.6786]	[0.3207, 0.3455]	[1.1982, 1.8283]
ODE	[0.4136, 0.4705]	[0.5995, 0.6649]	[0.2451, 0.2645]	[1.4311, 1.8859]

Table A.7: Confidence intervals for the different modeling methods when validating D_C on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1497, 0.1733]	[0.0884, 0.1174]	[0.0872, 0.1006]	[0.5229, 0.8404]
GMM	[0.1188, 0.1367]	[0.0595, 0.0897]	[0.0907, 0.1066]	[0.6996, 1.0097]
ODEwNoise	[0.3808, 0.3879]	[0.1853, 0.2025]	[0.3028, 0.3126]	[0.2525, 0.6432]
ODE	[0.5030, 0.5099]	[0.2687, 0.2801]	[0.2522, 0.2544]	[0.6054, 0.7256]

Table A.8: Confidence intervals for the different modeling methods when validating D_C on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.2538, 0.4849]	[0.0953, 0.1284]	[0.0795, 0.0981]	[0.7672, 1.3332]
GMM	[0.2801, 0.3734]	[0.1063, 0.1300]	[0.0892, 0.1100]	[0.8272, 1.4496]
ODEwNoise	[0.2780, 0.2832]	[0.1781, 0.1904]	[0.2209, 0.2319]	[0.5578, 0.6565]
ODE	[0.4327, 0.4448]	[0.2244, 0.2296]	[0.2258, 0.2272]	[0.5022, 0.5893]

Table A.9: Confidence intervals for the different modeling methods when validating D_C on the *Short* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.4108, 0.7687]	[0.1961, 0.2756]	[0.1590, 0.2328]	[1.1562, 2.4996]
GMM	[0.3208, 0.4943]	[0.1550, 0.2143]	[0.1636, 0.2301]	[0.8381, 2.2946]
ODEwNoise	[0.3029, 0.3277]	[0.2575, 0.3618]	[0.3003, 0.4341]	[0.7379, 1.7648]
ODE	[0.4212, 0.4591]	[0.3234, 0.3411]	[0.2526, 0.2668]	[0.7841, 1.3200]

A.2 Robustness

The tables provided to support the confidence intervals in Figure 5.6.

Table A.10: Confidence intervals for the different modeling methods when D_B drove D_A 's car on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1559, 0.1785]	[0.2053, 0.2505]	[0.0610, 0.0748]	[1.2093, 1.6165]
GMM	[0.1484, 0.1647]	[0.1617, 0.2133]	[0.0749, 0.1042]	[0.9557, 1.7116]
ODEwNoise	[0.4189, 0.4228]	[0.4441, 0.4641]	[0.3368, 0.3520]	[1.3682, 1.4329]
ODE	[0.4862, 0.4962]	[0.3325, 0.3356]	[0.2585, 0.2617]	[1.1641, 1.2356]

Table A.11: Confidence intervals for the different modeling methods when D_B drove D_A 's car on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1292, 0.1576]	[0.1590, 0.1787]	[0.0881, 0.1131]	[0.4024, 0.6845]
GMM	[0.1418, 0.1723]	[0.1650, 0.1794]	[0.0766, 0.0957]	[0.6770, 0.8696]
ODEwNoise	[0.3560, 0.3603]	[0.2913, 0.3019]	[0.3027, 0.3151]	[0.9189, 0.9857]
ODE	[0.4453, 0.4579]	[0.3111, 0.3130]	[0.2593, 0.2614]	[0.8211, 0.9155]

Table A.12: Confidence intervals for the different modeling methods when D_B drove D_A 's car on the *Short* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.4546, 0.5767]	[0.5320, 0.7113]	[0.1864, 0.2324]	[1.2242, 2.2823]
GMM	[0.5212, 0.7534]	[0.5084, 0.6544]	[0.1934, 0.2974]	[1.3760, 2.4901]
ODEwNoise	[0.4139, 0.4467]	[0.4613, 0.5519]	[0.1613, 0.2051]	[0.8165, 1.3000]
ODE	[0.4997, 0.5205]	[0.6572, 0.7027]	[0.2543, 0.2699]	[0.8640, 1.5610]

Table A.13: Confidence intervals for the different modeling methods when D_C drove D_A 's car on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.2194, 0.2492]	[0.2101, 0.2436]	[0.0799, 0.0915]	[0.6151, 0.9137]
GMM	[0.1798, 0.2038]	[0.1850, 0.2374]	[0.1133, 0.1372]	[0.6654, 1.0119]
ODEwNoise	[0.4384, 0.4445]	[0.5403, 0.5569]	[0.3258, 0.3383]	[0.4137, 0.4513]
ODE	[0.4997, 0.5019]	[0.2705, 0.2803]	[0.2536, 0.2561]	[0.7020, 0.7897]

Table A.14: Confidence intervals for the different modeling methods when D_C drove D_A 's car on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.4914, 0.6134]	[0.1676, 0.1948]	[0.0750, 0.0865]	[0.5704, 0.9755]
GMM	[0.5327, 0.6893]	[0.2008, 0.2511]	[0.1595, 0.1972]	[1.1277, 1.7581]
ODEwNoise	[0.2782, 0.2803]	[0.3516, 0.3569]	[0.2388, 0.2447]	[0.6785, 0.7728]
ODE	[0.4394, 0.4523]	[0.2293, 0.2352]	[0.2254, 0.2270]	[0.5146, 0.6048]

Table A.15: Confidence intervals for the different modeling methods when D_C drove D_A 's car on the *Short* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.4159, 0.5917]	[0.2403, 0.3260]	[0.1049, 0.1698]	[1.1021, 2.0193]
GMM	[0.4462, 0.5777]	[0.2913, 0.3876]	[0.2323, 0.2957]	[1.1205, 2.6721]
ODEwNoise	[0.3111, 0.3558]	[0.4266, 0.4627]	[0.3028, 0.3331]	[0.5597, 1.0564]
ODE	[0.4346, 0.4586]	[0.3329, 0.3473]	[0.2555, 0.2680]	[0.3596, 0.9294]

Table A.16: Confidence intervals for the different modeling methods when D_C drove D_B 's car on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.2234, 0.2572]	[0.2477, 0.2857]	[0.0963, 0.1172]	[0.5037, 0.6974]
GMM	[0.2122, 0.2513]	[0.2586, 0.2993]	[0.1378, 0.1783]	[0.6217, 0.9633]
ODEwNoise	[0.4179, 0.4545]	[0.5313, 0.5678]	[0.1028, 0.1144]	[0.4995, 0.6891]
ODE	[0.4694, 0.5018]	[0.2423, 0.2738]	[0.2513, 0.2559]	[0.6040, 0.7546]

Table A.17: Confidence intervals for the different modeling methods when D_C drove D_B 's car on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.5591, 0.6873]	[0.2343, 0.2762]	[0.1114, 0.1279]	[0.4248, 0.5738]
GMM	[0.3837, 0.5659]	[0.2555, 0.2773]	[0.1374, 0.1675]	[0.5568, 0.6296]
ODEwNoise	[0.3384, 0.3431]	[0.2628, 0.2765]	[0.1596, 0.1735]	[0.8009, 0.8554]
ODE	[0.4409, 0.4458]	[0.2285, 0.2316]	[0.2286, 0.2302]	[0.5098, 0.6285]

Table A.18: Confidence intervals for the different modeling methods when D_C drove D_B 's car on the *Short* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.5888, 0.9424]	[0.3564, 0.4897]	[0.1708, 0.2796]	[0.8704, 2.4100]
GMM	[0.5639, 0.8020]	[0.3276, 0.4083]	[0.2326, 0.3035]	[0.8625, 2.4742]
ODEwNoise	[0.3289, 0.3608]	[0.4038, 0.4220]	[0.3116, 0.3423]	[1.2146, 2.0343]
ODE	[0.4295, 0.4617]	[0.3254, 0.3428]	[0.2513, 0.2694]	[0.8010, 1.3685]

A.3 Less training data

Data from the simulations on training data of 500 km. The tables contains the confidence interval that Figure 5.7 is built on.

Table A.19: Confidence intervals for the different modeling methods when validating D_A on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1788,0.1994]	[0.133, 0.1553]	[0.1896, 0.2113]	[0.7073, 1.9133]
GMM	[0.1832,0.1951]	[0.1089,0.1206]	[0.1278,0.1387]	[1.1779,2.4474]
ODEwNoise	[0.4663,0.4689]	[0.2261,0.2330]	[0.2584,0.2632]	[0.4011,0.4302]

Table A.20: Confidence intervals for the different modeling methods when validating D_A on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1740,0.1917]	[0.1905,0.2036]	[0.1673,0.1814]	[0.3886,0.6013]
GMM	[0.1713,0.1816]	[0.1297,0.1475]	[0.1082,0.1154]	[0.4709,0.6328]
ODEwNoise	[0.4391,0.4416]	[0.2517,0.2557]	[0.2217,0.2239]	[0.5982,0.6550]

Table A.21: Confidence intervals for the different modeling methods when validating D_B on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1733,0.1901]	[0.1164,0.1445]	[0.0858,0.1031]	[0.9529,1.5116]
GMM	[0.1741,0.1940]	[0.1326,0.1599]	[0.0890,0.1047]	[0.9393,1.5692]
ODEwNoise	[0.4588,0.4634]	[0.3866,0.4066]	[0.2155,0.2306]	[1.1279,1.1824]

Table A.22: Confidence intervals for the different modeling methods when validating D_B on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1723,0.1965]	[0.1455,0.1598]	[0.1234,0.1394]	[0.4256,0.7533]
GMM	[0.1442,0.1739]	[0.1112,0.1309]	[0.0632,0.0746]	[0.4412,0.7430]
ODEwNoise	[0.3853,0.3889]	[0.1661,0.1736]	[0.1335,0.1457]	[0.4917,0.6476]

Table A.23: Confidence intervals for the different modeling methods when validating D_C on the *Mixed* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.1499,0.1801]	[0.1113,0.1311]	[0.0990,0.1071]	[0.5278,0.7182]
GMM	[0.1140,0.1327]	[0.0832,0.0971]	[0.0862,0.0983]	[0.5077,0.08827]
ODEwNoise	[0.3739,0.3791]	[0.2017,0.2111]	[0.2149,0.2218]	[0.9217,1.0330]

Table A.24: Confidence intervals for the different modeling methods when validating D_C on the *Highway* route.

Model	I_{δ_v}	I_{δ_p}	I_{δ_Φ}	I_{δ_f}
DD	[0.2846,0.3005]	[0.1411,0.1518]	[0.1396,0.1506]	[0.4285,0.5204]
GMM	[0.2810,0.2987]	[0.1287,0.1395]	[0.1290,0.1384]	[0.5396,0.6526]
ODEwNoise	[0.3413,0.3443]	[0.4685,0.4813]	[0.4681,0.4731]	[1.1371,1.1935]

Bibliography

- Bishop, C (2007). “Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn”. *Springer, New York*.
- Folland, G. B. (1992). *Fourier analysis and its applications*. Vol. 4. American Mathematical Soc.
- Gao, G. G. (2014). *Investigation of climate control power consumption in DTE estimation for electric vehicles*. Tech. rep. SAE Technical Paper.
- Guzzella, L. and A. Sciarretta (2007). “Vehicle propulsion systems”. In: Springer. Chap. 2, pp. 13–21.
- Karbowski, D., S. Pagerit, and A. Calkins (2012). “Energy consumption prediction of a vehicle along a user-specified real-world trip”. In: *Proc. from Electric Vehicle Symposium (EVS 26), Los Angeles*.
- Müller, M., M. Reif, M. Pandit, W. Staiger, and B. Martin (2004). *Vehicle Speed Prediction for Driver Assistance Systems*. Tech. rep. University of Kaiserslautern, Germany and DaimlerChrysler Corporation.
- Rydén, T and G Lindgren (2000). *Markovprocesser. lund. kfs ab. 2. uppl.*
- Schori, M., T. J. Boehme, B. Frank, and B. P. Lampe (2015). “Optimal calibration of map-based energy management for plug-in parallel hybrid configurations: a hybrid optimal control approach”. *IEEE Transactions on Vehicular Technology* **64**:9, pp. 3897–3907.
- Torp, E. and P. Önnegren (2013). *Driving cycle generation using statistical analysis and markov chains*.
- Volvo, C. C. (2017). 2. URL: <http://support.volvocars.com/uk/cars/Pages/owners-manual.aspx?mc=y555&my=2015&sw=14w20&article=68bde8fd2c211105c0a801e8013b10da>.

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS	
		<i>Date of issue</i> June 2017	
		<i>Document Number</i> ISRN LUTFD2/TFRT--6035--SE	
<i>Author(s)</i> Carl Bredberg John Stjernrup		<i>Supervisor</i> Martin Sivertsson, Volvo Car Corporation Anton Cervin, Dept. of Automatic Control, Lund University, Sweden Giacomo Como, Dept. of Automatic Control, Lund University, Sweden (examiner)	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Driver Modeling, Velocity and Energy Consumption Prediction of Electric Vehicles			
<i>Abstract</i> <p>A driver model can be used to predict the vehicle velocity and the energy consumption. It can be modeled such that it benefits from historical data and can be further improved if a navigation system is available. A well implemented driver model is important since the car fleet seems to be more and more electrified. The role of the driver model would then be to increase the accuracy for when the driver needs to take a break to recharge the vehicle, and thereby decrease the driver's range anxiety. Historical behaviour of different drivers has been measured and collected by Volvo Car Corporation. The information regarding these drivers has been used in four out of five implemented driver models. Three of the models use Markov chain theory to make the prediction while the fourth takes advantage of frequency analysis. Above the aim to increase the accuracy of the energy consumption prediction it is investigated to what extent a personal driver model can be created.</p> <p>In addition to the driver models a primitive method to predict when a driver reacts to a new posted reference speed is proposed and four validation methods are suggested.</p> <p>The results indicate that the driver models based on historical data perform better energy predictions than the one without any historical data. The driver model that uses Gaussian mixture model together with Markov chain theory makes the best energy prediction. The individual differences are especially shown at road segments with higher reference speed. In urban traffic it is more likely that the traffic pattern decides the energy consumption.</p>			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 1-78	<i>Recipient's notes</i>	
<i>Security classification</i>			