Data-driven Steering-Torque Behaviour Modelling

A Hidden Markov Model Approach

Robert Jaap van Wijk

Master Thesis
Delft University of Technology

TU Delft

Data-driven Steering-Torque Behaviour Modelling

A Hidden Markov Model Approach

by

Robert Jaap van Wijk

to obtain the degree of Master of Science in Mechanical Engineering

at Delft University of Technology,

to be defended publicly on Wednesday December 8th, 2021 at 14:30 PM.

Student number:	4313968		
Project duration:	November 9th, 2020 – November 24th, 2021		
Thesis committee:	Dr. ir. R. Happee,	Chair	TU Delft, CoR-IV
	Dr. ir. B. Shyrokau.	Supervisor	TU Delft, CoR-IV
	Dr. ir. A. Zgonnikov,	Committee member	TU Delft, CoR-HRI

November 24th, 2021

The work in this master thesis was performed in collaboration with Toyota Motor Europe, Belgium.

This thesis is confidential and cannot be made public until December 31, 2024.

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

TU Delft Delft University of Technology

Acknowledgements

Without a doubt, we will all remember how the beginning of ”that” year changed our way of life. For me, it was the year in which I accepted, what can be described as, one of the biggest and exciting challenges in my life so far. On top of that came the difficulty of working from home, the exhilarating experience of moving to a different country, and the opportunity to work at a well renowned automotive company. I would like to sincerely thank all the individuals who supported, guided, and motivated me along the way.

First of all, I wish to thank my thesis supervisors. From Delft University of Technology I am grateful to have had Barys Shyrokau as my supervising professor. He is a genuine, kindhearted, and hard-working man whom I would like to thank for his belief in, patience with and source of intellectual motivation for this project. From Belgium, I want to express my gratitude to Andrea Lazcano, my daily supervisor at Toyota Motor Europe. I admire her passion and devotion to her work and has been an endless source of inspiration to my own work. Furthermore, I would like to thank my manager at Toyota, Xabier Akutain, for reminding me to stay vigilant about the bigger picture and the opportunity to perform this project at Toyota.

To my parents, on which I can always count whenever and wherever. Celebrating with me every proud moment and encouraging me to keep believing in myself in every though moment. In particular to my mom, who always tells me to just explain the complex problem I was facing, nodding her head like she understood. And to my dad, who would sit down with me until I found the solution. You are my inspiration to strive for the best.

To all my friends in and around Delft. Each and every one of them wonderful people without whom my student life would not have been the same.

Last but not least, to my Brussels flatmates. A group of heart-warming people who made working from home not so bad after all.

Robert Jaap van Wijk
Delft, November 2021

Abstract

Commercial Driver Steering Assistance Systems (DSAS) currently available in mid- to high-end vehicles focus on path-tracking performance without taking into account the steering intentions of the driver. When DSAS is conflicting with the intentions of the driver it risks being turned off, resulting in discomfort and reduced safety. Improved driver-automation interaction can be achieved by sharing vehicle lateral control through torques on the steering wheel. Moreover, integrating a driver steering-torque model in the DSAS controller allows the system to better match driver intentions, limitations, and capabilities.

The work in this thesis presents a data-driven approach to driver steering-torque modelling. An existing steering-angle-based approach is modified, parametrized, and evaluated for estimating driver steering-torque instead, making the driver model appropriate for the development of new torque- feedback based DSAS. Driver behaviour is modelled by learning the parameters of a Hidden Markov Model (HMM). Subsequently, steering-torque estimation is performed by inferring the HMM model with Gaussian Mixture Regression (GMR).

In addition, as the HMM approach has not yet been considered for estimating driver steering-torque, an extensive parameter selection framework is presented for the objective selection of the model's degrees-of-freedom. First, a feature selection step evaluates the relevance of features that describe the model's output behaviour. Thereafter, an iterative overfitting criteria is employed to appropriately select the model configuration. Final model behaviour is determined by adjusting the metric weights of a linear performance score with the aim to make a trade-off between model estimation accuracy and smoothness.

Naturalistic driver steering-torque data from seven participants was gathered in a fixed-base driving simulator, located at Toyota Motor Europe (TME) in Belgium. The data was used for the training, evaluation, and testing of the modified HMM approach. Results on a test set indicated that model performance is dependent on individual driver behaviour, rather than driver skill level. Furthermore, the results demonstrated that an average accuracy of 92% is achieved while estimations are 37% smoother and require 90% less data compared to a baseline model.

Keywords: Haptic Shared Control, Driver Model, Data-Driven, Hidden Markov Model, Feature Selection

Contents

List of Figures

List of Tables

1

Introduction

Vision Zero is the long-term goal of the European Commission towards nearly zero road related deaths by 2050. Although the number of deaths have decreased by 55% between 2001 and 2018, still in 2018 alone more than 25.000 people lost their lives on European roads [10]. To reach their goal, the European Commission (EC) revised the General Safety Regulation in 2019. Among others, the regulation mandates the implementation of a Lane Keeping Assist System (LKAS) in new road vehicles by 2022 [34].

LKAS belong to a subclass of Driver Assistance Systems (DAS) called Driver Steering Assistance Systems (DSAS). DSAS can (partially) take over the lateral control of the vehicle by applying a guidance signal on the steering column, assisting the driver to stay within lane boundaries. By doing so, driver and automation system share the lateral control of the vehicle simultaneously. However, a limitation of current implementations is that the main focus of DSAS is purely based on path-tracking performance, without taking into account the interaction with the driver. If the system is conflicting with the steering intentions of the driver, the system can be considered intrusive. This in turn leads to driver discomfort, reduced safety, and a high chance of being rejected by the driver who will switch the system off [26].

To improve Human-Machine Interaction (HMI), Abbink et al. summarized in the 2012 work [1] important guidelines for designing automation systems interacting with humans:

- *Human perspective*: the human operator should always interact and receive feedback with the automation system *continuously*.
- *Automation perspective*: knowledge about the human operator should be integrated in the automation system to better understand and match the intentions of the operator.

Continuous interaction between human operator and automation system was proposed by Abbink et al. in [1] to be achievable through the sharing of forces on a common interface, a concept named Haptic Shared Control (HSC) [2]. Better understanding the intentions and capabilities of the human operator can be achieved through the proper modeling and integration of human behaviour in automation systems. Relating this to DSAS, this entails that improved interaction between driver and DSAS can be achieved when lateral control is shared through torques on the steering wheel. Combining this concept with the integration of a driver steering-torque behavioural model, DSAS can take into account driver intentions, capabilities, and limitations. This gives DSAS the opportunity to gain driver trust, improve driving comfort, and perform efficient interventions while minimizing conflict [36, 15]. However, while HSC-based DSAS is an experimentally validated concept, accurately modelling driver steering behaviour is still a research challenge due to the highly complex, stochastic, and variable nature of human behaviour [20].

The goal of a driver steering behaviour model is to accurately describe the relation in which the driver translates perceived information about the driving scenario to a steering action on the steering-wheel [24]. This perceived information can be understood as,

- *Visual:* perception of vehicle surroundings through the drivers' eyes.
- *Sensory:* perception of ego-vehicle motion through vestibular and kinesthetic sensory systems.
- *Feel:* tactile and haptic feel through the steering wheel.

State-of-the-art literature on driver steering behaviour modelling can roughly be categorized in parametric, non-parametric and mixed approaches. Parametric approaches often follow a theoretical background in which the steering task is modelled after physical principles using a priori assumed analytical models. Examples of such parametric approaches are the Cybernetic Driver Model by L. Saleh et al. [35] and an optimal-control-based approach by T. Niu and D. Cole [29]. Based on experiments, both approaches define sub-models for perceptual, cognitive, and neuromuscular dynamics differently. Respective model parameters are tuned by hand or learned from data to accurately fit the driver behaviour. Advantages of parametric approaches is that they provide an intuitive understanding of the human steering task. Furthermore, the mentioned approaches have already found experimental implementations in haptic shared controllers, respectively in [25] and [21]. However, while the perceptual and neuromuscular sub-models are thoroughly researched, human cognitive functions is a highly non-linear process and still subject to active research and thus are only approximated by parametric models.

Non-parametric approaches do not define model structure a priori. Instead, the structure is inferred from data using machine-learning or statistical methods. An example of a non-parametric driver model is the neural network-based approach by S. Jugade [18]. A two-layer feed-forward neural network was proposed to predict driver steering actions as a function of the driving scenario. Depending on the quality of the training data, advantages of such non-parametric approaches include the ability to learn detailed and non-linear descriptions of the steering task, directly fitted to the driver. However, a disadvantage of the black-box model structure is that model interpretation and intuition is lost. Furthermore, such models are typically maneuver specific (e.g. lane keeping only) and require retraining for alternative maneuvers.

Mixed approaches aim to combine the advantages of parametric models, that is model intuition, with that of non-parametric approaches, being the ability to learn non-linear descriptions and directly fitting to personal driver behaviour. This is achieved by assuming part of the model structure to follow a theoretical background with a priori defined sub-models for upper-level dynamics. The model structure is then completed with a data-driven approach, often statistical- or machine-learning-based, to define the lower-level model functions. Examples of mixed approaches are the Motion Primitives (MP) driver model by M. Flad et al. [12] and the Hidden Markov Model (HMM) based driver model by S. Lefèvre et al. [22]. However, both the mentioned non-parametric and mixed model approaches are *steering-angle* based and have not yet been considered for modelling driver steering-torque.

Summarizing, to improve Human-Machine Interaction between driver and DSAS, literature suggests to adopt the HSC concept in combination with the integration of a driver steering-torque behaviour model in the DSAS. However, while HSC is a validated concept, accurately modelling driver behaviour is still a research challenge due to the complex, stochastic and variable nature of human behaviour. Mixed parametric / non-parametric modelling approaches are attractive methods due to their ability to capture this complex non-linear behaviour. However, current approaches have not yet been considered for modelling driver steering-torque directly.

1.1. Research Objectives

To address the lack of mixed-parametric approaches to driver steering-torque behaviour modelling, the work in this thesis presents a Hidden Markov Model based approach. The main goal of this thesis is to develop and evaluate the implementation and parameterization of an HMM approach to time-varying driver steering-torque behaviour modelling. The intended purpose is to make this model appropriate for the integration in new torque-feedback based DSAS. The main goal is broken down in the following research objectives:

- Investigate how the parameters of a Hidden Markov Model are learned and how the model can be inferred to respectively, capture and estimate driver steering-torque behaviour.
- Perform a virtually simulated parameter sensitivity study to investigate how the model parameters influence estimation quality.
- Develop an evaluation method to determine suitable features and model configurations that optimize the estimation quality.
- Design and perform a driver experiment in a fixed-base driving simulator to generate naturalistic driver steering-torque data for the training, evaluation and testing of the HMM model.

1.2. Thesis Structure

This thesis is structured as follows. Main findings and results of the proposed approach are presented in the form of an IEEE journal paper in Chapter 2. The Appendices A to E provide respectively, an in-depth explanations on model and inference methodology (Appendix A), details on data collection (Appendix B), extended results of the parameter sensitivity study (Appendix C), the parameter selection methodology (Appendix D), and supplementary results to the main findings (Appendix E).

2

Journal Paper

This chapter presents the main findings and results of the proposed approach in the form of an IEEE journal paper.

Data-driven Steering-Torque Behaviour Modelling: A Hidden Markov Model Approach

Current commercial Driver Steering Assistance Systems (DSAS) focus on path-tracking performance without taking into account driver intentions. Improved driver-automation interaction can be achieved by sharing vehicle lateral control through torques. Furthermore, integrating a driver steering-torque model allows to better match driver intentions. In this research, an existing driver model is adapted and parametrized for estimating driver steering-torque. Driver behaviour is modelled by learning the parameters of a Hidden Markov Model (HMM) and estimation is performed with Gaussian Mixture Regression (GMR). A parameter selection framework enables to select model hyper-parameters objectively. First, feature relevance is determined with an extensive feature selection step. Thereafter, an iterative overfitting criteria is employed to select the number of hidden states. Final model behaviour is determined by adjusting the metric weights of a linear cost-function with the aim to trade-off estimation accuracy and smoothness. Naturalistic driver steering-torque data from seven participants was gathered in a fixed-base driving simulator at Toyota Motor Europe for the training, evaluation, and testing of the proposed model. The results demonstrate that a 92% model accuracy can be achieved while the estimated steering-torque is 37% smoother and requires 90% less data compared to a baseline model.

Index Terms—**Haptic Shared Control, Driver Model, Data-Driven, Hidden Markov Model, Feature Selection**

I. Introduction

DRIVER Steering Assistance Systems (DSAS) can (partially) take over the lateral control of the vehicle, thus sharing control with the driver. However, current commercial DSAS mainly focus on path-tracking performance without taking into account interaction with the driver. When DSAS is conflicting with the steering intentions of the driver, it risks being turned off [1], resulting in discomfort and reduced safety.

To improve Human-Machine Interaction (HMI), Abbink et al. summarized in [2] important automation design guidelines. Firstly, the human operator should interact and receive feedback with the automation system continuously. Secondly, knowledge about the human operator should be integrated to match intentions and capabilities. The former can be achieved with interaction through forces on the steering-wheel, referred to as Haptic Shared Control (HSC) [3], while the latter is achieved through the integration of a driver behavioural model. When integrated, the combination of these two methods allow DSAS to gain driver trust, improve driving comfort, and perform efficient interventions while minimizing conflict [4], [5]. However, while HSC-based DSAS is an experimentally validated concept, accurately modelling driver steering behaviour is still a research challenge due to the highly complex, stochastic, and variable nature of human behaviour [6].

State-of-the-art literature on driver behaviour modelling can be categorized in parametric, non-parametric and mixed approaches. Parametric approaches model the steering task based on physical theoretic principles using a priori assumed analytical models. This provides an intuitive understanding of the human steering task. Parameterization is often done by hand or learned from data. Examples of parametric driver models are the Cybernetic Driver Model developed by L. Saleh et al. [7] and an optimal-control-based approach developed by T. Niu and D. Cole [8]. Both models have already found experimental implementations in haptic shared controllers ([9] and [10], respectively). Each model assumes different sub-models for driver cognitive, perceptual, and neuromuscular dynamics. However, while the perceptual and neuromuscular sub-models are thoroughly researched, human cognitive processing is highly non-linear and still subject to active research and thus only approximated by parametric models.

Non-parametric approaches infer model structure from data using machine-learning methods without a priori assumptions. Depending on the quality of the data, this enables to learn detailed and non-linear descriptions of the steering behaviour directly fitted to the driver. An example of a non-parametric model is the feed-forward neural-network-based approach by S. Jugade [11]. However, the black-box structure makes that model interpretation and intuition is lost. Furthermore, non-parametric approaches are often maneuver specific and require retraining for new maneuvers.

Mixed approaches aim to combine model intuition of parametric models with the ability of non-parametric approaches to learn non-linear descriptions. Part of the model structure follows a theoretical background, while the model is completed with data-driven design. Examples are the Motion Primitives (MP) driver model by M. Flad et al. [12] and the Hidden Markov Model (HMM) based driver model by S. Lefèvre et al. [13]. Both models assume upper-level dynamics of which the lower-level functions are learned with statistical methods. However, the mentioned non-parametric and mixed approaches model human *steering-angle* behaviour and have not yet been investigated to directly model steering-torque.

The current work proposes to adapt the HMM approach of [13] to model driver steering-torque instead. In the work by Lefèvre et al., an HMM-based driver model is implemented in an MPC-based Lane Departure Warning controller. The model provides the controller with a personalized (learned) steering-angle reference signal when the controller predicts an unintentional lane departure. This reference is taken into account in the optimization criterion to guide the vehicle back in the lane as close to what the driver would do. The focus of this work is on modifying the HMM model to provide a *continuous* steering-torque signal instead, making the model appropriate for the development of new torque-

feedback based driver assistance systems [10]. Furthermore, an extensive parameter selection framework is presented to design the preferred model behaviour.

This paper is structured as follows. The proposed driver model and the inference process for estimating driver steering-torque is explained in Section II and a preliminary parameter sensitivity study is included in Section III. Next, Section IV introduces the parameter selection framework, followed by a driving simulator experiment in Section V for generating naturalistic steering-torque data to validate the approach. The results are presented in Section VI and the conclusions are highlighted in Section VII.

II. Driver Model

The goal of a driver steering-torque model is to accurately describe the relation in which the driver translates perceived information about the driving scenario to a steering-torque action on the steering-wheel [14]. Modelling human steering behaviour with HMMs was introduced by A. Pentland and A. Liu in [15]. In the work, human behaviour is considered to consist of a set of discrete states, each with their own control behaviour, sequenced together with a Markov chain. Lower- and higher-level driving maneuvers can be described by hierarchically organizing the set of states. Lefèvre et al. proposed in [13] to use single Gaussian normal distributions as the state control behaviours to learn the relation between the driving scenario and the steering-angle. The current work adopts the normal distributions, but learns the relation with the steering-torque instead. This section first elaborates on the model structure in Section II-A and subsequently explains the model inference process for estimating driver steering-torque in Section II-B.

A. Model Structure

The steering behaviour of the driver is modelled with a fully connected HMM. The aim of the model is to learn the joint probability distribution between the current driving scenario, defined as a vector of measured features, $F \in \mathbb{R}^{N_f}$, and the driver steering-torque T_d. The model is described by the number of hidden states K, determining the model configuration, and three model parameters related to the model's behaviour, collectively denoted as $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}, B\}$. $\boldsymbol{\pi} \in \mathbb{R}^K$ is a vector of prior probabilities for starting in state $k = \{1, ..., K\}$ at time $t = 1$. $\boldsymbol{A} \in \mathbb{R}^{K \times K}$ is a matrix of probabilities, denoted as the (discrete) state transition matrix. Each element a_{jk} describes the probability of transitioning between state j at time $t-1$ to state k at time t, for $j, k = \{1, ..., K\}$. The parameter B is the set of state emission (control) behaviours, $b_k = P(x_t|k)$ for $k = \{1, ..., K\}$, and is defined as a probability distribution of being in state k and observing $x_t = [F_t, T_t^d]^\top$, the combined observation vector at time t. As the steering-torque T_d is a continuous variable, each state emission distribution is assumed to be a single multivariate Gaussian, $b_k \sim \mathcal{N}(\mu_k, \Sigma_k)$. The mean μ_k and covariance matrix Σ_k are defined by the mean and covariance of the input (F) and output (T_d) features, denoted as,

$$\mu_k = \begin{bmatrix} \mu_k^F \\ \mu_k^T \end{bmatrix} \text{ and } \Sigma_k = \begin{bmatrix} \Sigma_k^{FF} & \Sigma_k^{FT} \\ \Sigma_k^{TF} & \Sigma_k^{TT} \end{bmatrix} \tag{1}$$

Denoting s_t as the state at time t, the random variable $\boldsymbol{S} = \{s_1, ..., s_T\}$ denotes the hidden state sequence from time $t = 1$ to time $t = T$. Together with the random variable $\boldsymbol{X} = \{x_1, ..., x_T\}$, denoted as the observation sequence, the joint state-observation probability distribution $P(\boldsymbol{X}, \boldsymbol{S}|\boldsymbol{\theta})$ is defined in [16] as,

$$p(\boldsymbol{X}, \boldsymbol{S}|\boldsymbol{\theta}) = p(s_1|\boldsymbol{\pi}) \left[\prod_{t=2}^{T} P(s_t|s_{t-1}, \boldsymbol{A})\right] \prod_{t=1}^{T} p(x_t|s_t, B) \tag{2}$$

The model parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}, B\}$ are learned from recorded driving data with the Baum-Welch Algorithm [17], an adapted version of the Expectation-Maximization (EM) Algorithm [18]. The type of features F and the number of hidden states K are free model parameters and are determined with the parameter selection framework, explained in Section IV.

B. Model Inference

At each time instant, the state emission distributions represent a mixture of K multivariate Gaussians. Therefore, estimation of driver steering-torque T_d^{est} is performed with Gaussian Mixture Regression (GMR) [19] and is calculated as,

$$T_{d,t}^{est} = \sum_{k=1}^{K} \alpha_{t,k}^F \left[\mu_k^{FF} + \Sigma_k^{TF}(\Sigma_k^{FF})^{-1}(F_t - \mu_k^{FF})\right] \tag{3}$$

where $\alpha_{t,k}$ represents the mixture weights. Calinon et al. proposed in [20] to calculate the mixture weights recursively through the HMM representation, considering both spatial and temporal information contained in the HMM. This recursive calculation is known as the Forward Variable [17] and is defined as,

$$\alpha_{t,k}^F = \frac{\left(\sum_{j=1}^{K} \alpha_{t-1,j}^F a_{jk}\right) \mathcal{N}(F_t|\mu_k^F, \Sigma_k^{FF})}{\sum_{i=1}^{K} \left[\left(\sum_{j=1}^{K} \alpha_{t-1,j} a_{ji}\right) \mathcal{N}(F_t|\mu_k^F, \Sigma_k^{FF})\right]} \tag{4}$$

and corresponds to the probability of observing the partial observation sequence $\{F_1, F_2, ..., F_t\}$ until time t and being in state k, given the model parameters $\boldsymbol{\theta}$ [17]. The Forward Variable $\alpha_{1,k}^F$ is initialized with prior probability π_k as,

$$\alpha_{1,k}^F = \frac{\pi_k \mathcal{N}(F_1|\mu_k^F, \Sigma_k^{FF})}{\sum_{j=1}^{K} \left[\pi_j \mathcal{N}(F_1|\mu_k^F, \Sigma_k^{FF})\right]} \tag{5}$$

III. Parameter Sensitivity Study

An offline simulated pilot study was performed in IPG CarMaker with the aim to investigate model potential of capturing and estimating driver steering-torque and to investigate the impact of the free model parameters, the type of features and number of hidden states. Steering-torque data was generated by simulating two virtual IPGDriver behaviours (defensive and aggressive) for two hours each. Both drivers followed a continuous highway lane-keeping pursuit strategy at fixed longitudinal vehicle speed of 100 km/h. To ensure high-fidelity realistic simulations, vehicle dynamics were simulated with a non-linear vehicle model with proprietary steering system

TABLE I
BENCHMARK MODEL PERFORMANCE

Metric	Defensive	Generic	Aggressive	Generic
Accuracy [%]	89.9	90.1	90.9	90.8
$RMSE_T$ [Nm]	0.165	0.162	0.145	0.147

[21], parametrized to represent a Toyota production vehicle. Subsequently, driver-dependent and generic benchmark HMM models were trained and validated. The models were trained with eighteen candidate features (see Appendix, Table A.1) and five hidden states to ensure sufficient model complexity.

A. Results & Discussion

Both driver-dependent models are able to fit well on the simulated data, shown by Table I, with an accuracy around 90% and an ∼0.15 Nm RMSE. Furthermore, the generic driver model is able to match this performance for both drivers. Observing that the generic model marginally improves on the defensive driver while doing slightly worse on the aggressive driver, potentially indicates that both behaviours are being averaged.

However, although the steering-torque is estimated accurately, the signal oscillates around the true value, seen in Figure 1a. From a Human-Machine Interaction (HMI) perspective, using such signals as reference for lateral controllers might result in uncomfortable steering wheel vibrations and thus should be kept to a minimum. From a modelling perspective this might indicate that the model is overfitted on the data. This can be observed in Figure 1b, as these oscillations are mainly produced by individual states.

Results from a preliminary parameter sensitivity study observed that the estimation quality is highly influenced by the number and type of features. Using fewer features improved estimation smoothness while similar accuracies to the benchmark model were achieved. Furthermore, the quality is also determined by the number of hidden states. Fewer states increased estimation smoothness, while more states increased estimation accuracy. From these observations it was concluded that final model behaviour is susceptible to the specific features the number of hidden states selected. Furthermore, as similar performance is achieved with less information, it strengthens the suspicion of model overfitting.

Following these conclusions, Section IV introduces a framework for the objective selection of the free model parameters with the aim to prevent overfitting. Furthermore, a model evaluation function is introduced that allows to balance model estimation accuracy with the smoothness of the estimation in order to keep the oscillations to a minimum.

IV. PARAMETER SELECTION FRAMEWORK

Initially, features and states were selected with the aim to obtain an accurate steering-torque estimation without taking into account model limitations. The sensitivity analysis showed that using too many features or states causes oscillations in the estimated signal, indicating possible overfitting. This section proposes a feature selection and overfit prevention framework to objectively find an optimal feature subset and

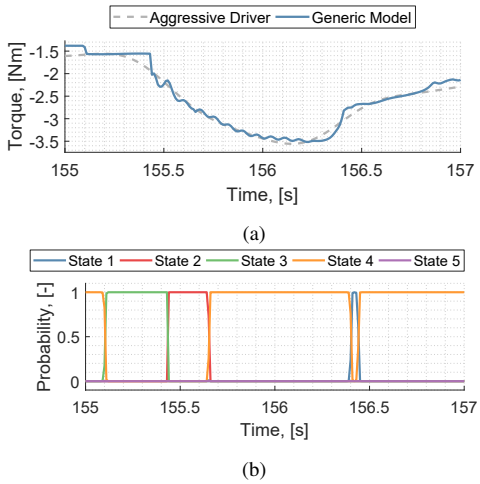


Fig. 1: Virtual Model Estimation Performance. (a) Accurate but oscillating estimations around 156s. (b) State probabilities α_k indicate that oscillations are generated by single states.

number of states K that maximizes estimation accuracy and generalisation capabilities, while minimizing oscillations. The framework consists of three parts, shown in Figure 2. Step one involves a feature selection method that generates candidate subsets. In step two, feature relevance is determined based on an evaluation function and one optimal subset is selected. The last step involves selecting the number of states K by fixing the feature subset. After the parameters are determined, final model performance is evaluated on a test set in Section VI. This section is organized as follows. Feature selection is introduced in Section IV-A, followed by the proposed method in Section IV-B. Section IV-C describes the optimal subset selection and determining the number of states K is explained in Section IV-D.

A. Feature Selection Background

Relevant features are strongly related to the regression target and their contribution cannot be described by other features [22]. Removing redundant and irrelevant features with feature selection improves learning speed and model performance, while reducing risk of overfitting and required data storage [23]. As opposed to feature *extraction*, where data dimensionality is reduced by constructing new relevant features, feature *selection* is preferred in applications where the original features have physical interpretations (e.g., the steering wheel angle δ_{SWA}). Feature selection reduces data dimensionality by directly selecting a relevant subset, making the final model more easily interpretable. Methods are categorized as supervised or unsupervised based on the availability of a regression target in the learning phase. Supervised methods determine feature relevance via correlation with the regression target, while unsupervised methods determine feature relevance by alternative criteria, as labeled data is missing [24]. The current work proposes a supervised method since the driver steering-torque is treated as the regression target. However, literature on supervised methods for HMMs is limited and often research specific [23].

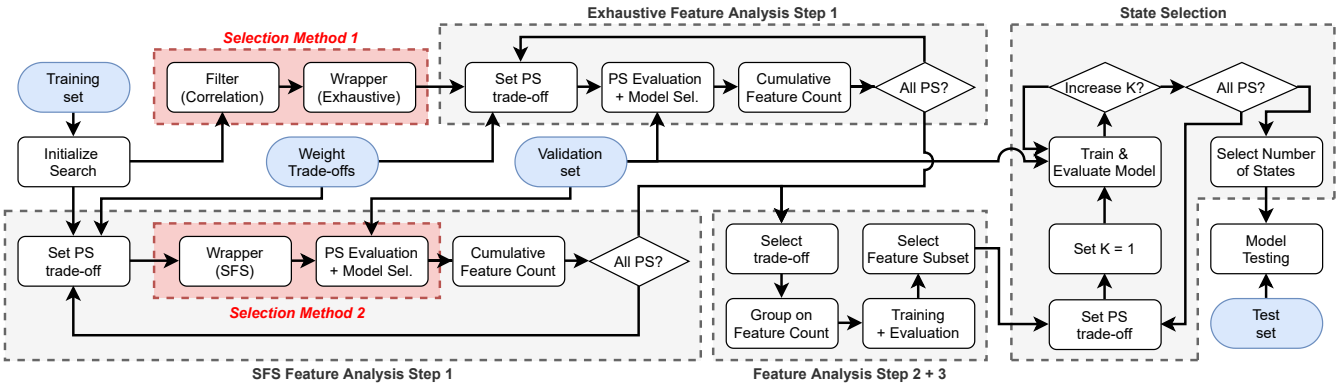


Fig. 2: The proposed parameter selection framework as explained in Sections IV-B to IV-D.

Therefore, the current proposed method is an extension of the work [25] by Fabian Faller, where an extensive feature selection analysis was performed for the HMM-based prediction of unintentional lane changes. The analysis consists of two wrapper-based selection methods, one performing an exhaustive search and the other performing a sequential forward search (SFS) [26], respectively. Wrapper methods determine inter-feature dependencies and redundancies by utilizing the training algorithm as a performance measure [27]. This makes wrappers computationally expensive but allow them to obtain better performing subsets. Since the number of possible candidate subsets in an exhaustive search grows exponentially with the number of features, Faller manually selected 11 of 24 total features, resulting in the evaluation of 2047 candidate subsets. Contrary to the exhaustive method, the SFS-wrapper only generates a fraction of the possible candidate subsets, allowing for a broader search over all candidate features. Faller uses the SFS-wrapper to validate the candidate subsets generated by the exhaustive wrapper. With both methods, Faller was able to successfully identify several optimal feature subsets. Even though the HMM was used as a classifier, due to the promising results and lack of alternative regression-based strategies for HMMs [23], a similar approach is adopted.

The current work adopts and extends the method of Faller by replacing the manual pre-selection with a filter method, see Section IV-B. Filter methods are independent of the learning algorithm and rank features based on data performance metrics [27]. They are computationally efficient but do not always remove all redundant features [22]. By feeding the filter output to the exhaustive wrapper, this problem is alleviated. Such filter-wrapper combinations were proposed as hybrid methods to combine the best properties of filters and wrappers [27]. Additionally, the current work implements a weighted performance score, see Section IV-C, allowing the evaluation of the learning algorithm to make a trade-off between estimation accuracy and estimation smoothness.

B. Feature Selection Methodology

The proposed method consists of two methods in parallel.

Method 1: is the hybrid method. First, a filter will rank the candidate features according to their univariate relevance with the steering-torque. The absolute Spearman correlation coefficient is chosen as the ranking criteria, evaluating the monotonic relationship[1], defined as,

$$|r_s| = \left| 1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N(N^2 - 1)} \right| \tag{6}$$

where N is the number elements and d_i is the difference in rank of element i. The ten strongest correlated features are selected. These features then form the new candidate subset for the exhaustive wrapper, determining inter-dependencies and redundancies in this subset. For ten features, the wrapper generates (trains) 1023 candidate subsets (models). The chosen training algorithm is a two-state ($K = 2$) HMM to balance model complexity with computation time. Model evaluation with the performance score is performed subsequently, see Section IV-C.

Method 2: is a wrapper running in parallel to the hybrid method. A sequential forward search (SFS) is selected, performing a search over all 18 candidate features. Starting with an empty "best feature set", new candidate subsets are generated by combining the "best set" with each remaining candidate feature, separately. Only the feature which combination with the "best set" maximizes the performance score is added to the "best set", and the process is repeated. The process is terminated either when selecting the next feature does not increase the performance score by more than 1% or all features are selected. The same two-state HMM is chosen as the training algorithm. Since this selection process is dependent on the performance score, model evaluation is performed simultaneously.

C. Feature Selection Analysis

Selecting the optimal feature subset consists of three steps. The first step determines feature relevance based on feature occurrence. Based on the observed occurrences, the second step groups these features into new subsets. The last step selects the optimal subset from the grouped subsets.

Step 1: Preliminary results showed that no single best feature subset exists for all validation recordings. Therefore, feature relevance is determined by counting feature occurrences. First, for each validation recording, the best performing

[1]relationship at non-constant rate

model is determined by evaluating the Performance Score (PS). The PS enables to balance estimation accuracy with estimation smoothness, defined as,

$$PS = \omega_1 * \|(100 - A_{T,est})\| + \omega_2 * \|SM_{T,est}\| \quad (7)$$

where $A_{T,est}$ and $SM_{T,est}$ are respectively the model accuracy and estimation smoothness metrics, defined in the Appendix, and ω_1 and ω_2 are the metric weights. Models with lower PS are considered better. Once evaluated for a specific weight trade-off, the individual features in the best selected models are counted cumulatively. The higher the count, the more relevant the feature is considered. This process is repeated for both wrapper methods, over a range of weight settings, to map feature relevance for different accuracy and smoothness trade-offs. The goal is to select the weight trade-off that results in the simplest model with an optimal balance between estimation accuracy and smoothness.

Step 2: For the selected weight trade-off, new feature subsets are generated by combining the results of both wrapper methods and grouping on feature relevance. To limit the number of new possible feature combinations, three selection strategies are defined. The new subsets contain...

- Strict Selection: ...features with at least 66% relevance.
- Mild Selection: ...features with at least 33% relevance.
- Liberal Selection: ...features counted at least once.

For each generated subset, a new two-state HMM is trained. Fixing the weight trade-off, selected in Step 1, allows to directly compare the trained models by their PS. Additionally, to put model performance into perspective, a baseline model was trained with all eighteen candidate features and two hidden states.

Step 3: The optimal feature subset is selected based on the model with the lowest PS.

D. State Selection

Final model complexity is determined by fixing the feature subset and selecting the number of states K accordingly. Increasing the states allows to capture driver steering-torque in greater detail. However, this also increases model dependency on the training data, known as overfitting. A common method to prevent overfitting is to evaluate the Bayesian Information Criteria (BIC) [28] on a large variety of models. The model with the lowest BIC score is considered to generalize best. However, as Faller noted in [25], the BIC is exclusively based on the models likelihood for a given training set. Therefore, it provides an uncertain estimation of the models' ability to generalize. Alternatively, Faller proposed that by evaluating model performance on the (unseen) validation set, generalisation capabilities and thus model complexity is ensured implicitly. Extending the method of Faller in the current work, the PS (7) provides a means to balance the selection of the number of hidden states between accurate and smooth estimation.

The state selection process works as follows: New models are trained for a range of states, while the feature subset remains fixed. Subsequently, the performance score for each model is evaluated on each validation recording for different weight trade-offs. For each trade-off, state selection is performed in two ways. The first method selects a general number of states based on the average PS over all recordings. The second method selects a number of states per recording. The general number of states is then selected as the average over all recordings. For both methods, the search is terminated if the addition of another state does not improve the PS by $>1\%$. Final state selection is determined by the minimum of the two methods.

V. Driving Simulator Experiment

Human driving contains variability in behaviour due to the stochastic nature of human movements. When presented with identical scenarios, humans seldom reproduce an identical action [6]. Therefore, to validate HMM driver modelling for naturalistic driving, a driving experiment was performed in a fixed-base simulator at Toyota Motor Europe (TME), Belgium. The goal of the experiment was to gather sufficient naturalistic driving data for training, validation, and testing of a generic driver HMM steering-torque model.

A. Driving Scenario

The designed scenario consists of 200km of randomly generated three-lane highway, each lane 3.5 meter in width. Road coordinates were defined in MATLAB and interpolated using the IPG CarMaker Scenario Editor. The height profile of a real-world highway scene was added to reduce average look-ahead distance and increase participant immersion. The scenario is split into 24 sections of five minutes (approximately eight kilometers) to avoid degradation of participant concentration per trial. Twelve sections were specifically designed to maximize scenario variability and are used for generating training data. The remaining twelve sections were designed to provide new driving scenarios for model validation and testing. Participants were tasked to manually follow the center lane as close to their preferred driving style without secondary tasks. No other driver assist was implemented except for a cruise controller, keeping vehicle speed at 100 km/h for the duration of each trial.

B. Experimental Procedure

A total of 7 participants took part in the experiment. All participants were staff members of TME, averaging 32 years of age (SD = 6.6 years), with knowledge on vehicle dynamics. Participants, except one, were in possession of a driver's license for an average of 15 years (SD = 6.5). Among the participants there were two expert test drivers, two advanced, two intermediate and one novice driver. The experiment took place in a fixed-base driving simulator with a mock-up Toyota production vehicle in front of a 210° projection screen, see Figure 3. The CarMaker scenarios were rendered with rFpro software. Participants performed two sessions, split over separate days, each in which twelve sections were driven. The order in which the sections were presented was randomized for every session and every participant. At the start of the first session, participants were able to familiarize themselves with the conditions during a test trial that was not recorded.

Fig. 3: Driving Simulator at Toyota Motor Europe.

C. Dataset Allocation

A total of 168 runs were recorded, resulting in 840 minutes (14 hours) of driving time. The recordings on the training sections for all participants, 84 in total (420 min.), are used for model training. Half of the remaining 84 recordings is used for model validation (42 recordings, 210 min.) and the other half is used for model testing (42 recordings, 210 min.).

VI. Results & Discussion

A. Feature Correlation

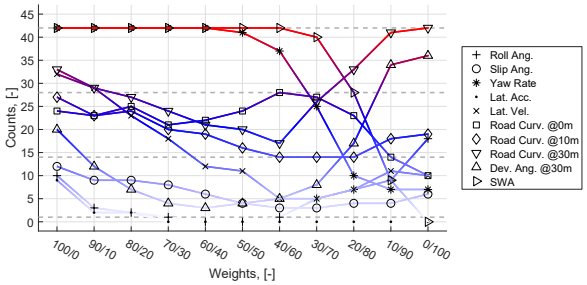
Features were ranked based on the average correlation over all training recordings. The ten strongest correlated features included in the exhaustive wrapper of sub-method 1 are summarized in Table II. The deviation angle at 10 meters is the single remaining feature that is strongly correlated ($r_s = 0.88$), but is disregarded due to the exhaustive wrapper feature limit. Feature relevance of the SFS-wrapper should determine if this is justified. The remaining features did not show a stronger correlation than $r_s = 0.4$ and are therefore disregarded based on weak correlation.

TABLE II
Feature Correlation

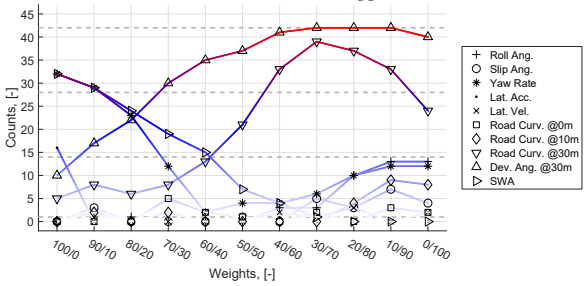
	Features	$\lvert r_s \rvert$	p
1	Steering Wheel Angle	0.9557	0
2	Deviation Angle @30m	0.9484	0
3	Road Curvature @10m	0.9321	0
4	Road Curvature @30m	0.9251	0
5	Yaw Rate	0.9249	0
6	Lateral Acceleration	0.9143	0
7	Roll Angle	0.9123	0
8	Road Curvature @0m	0.9099	0
9	Slip Angle	0.9062	0
10	Lateral Velocity	0.9058	0
11	Deviation Angle @10m	0.8848	0

B. Determining Feature Relevance

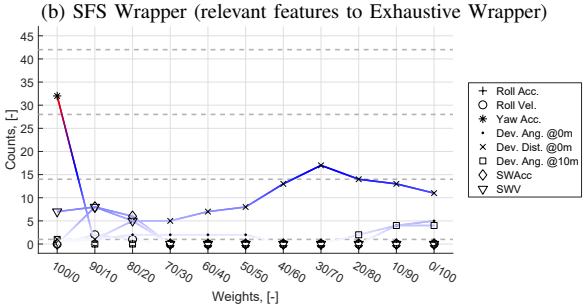
The influence of different weight trade-offs was mapped in ten percent intervals. The maximum feature count possible is equal to the number of validation recordings, 42. From the mapping of the exhaustive wrapper in Figure 4a it is observed that for increased weight on estimation smoothness, overall feature relevance decreases up to a balanced trade-off. This also means that the average number of features contained in the subsets decreases, indicating simpler models. Furthermore, while vehicle dynamics and driver input have increased relevance for more accurate estimations, road preview features becomes more relevant when smoother signals are preferred, which makes sense intuitively.



(a) Exhaustive Wrapper



(b) SFS Wrapper (relevant features to Exhaustive Wrapper)

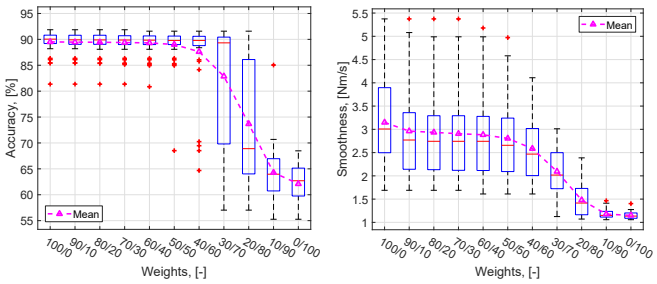


(c) SFS Wrapper (disregarded features)

Fig. 4: Mapping influence of metric weights on feature counts. Weights are denoted as accuracy/smoothness.

From the feature count of the SFS wrapper method, seen in Figure 4b, it is observed that the SWA and Yaw Rate are still important for model accuracy and road preview features for estimation smoothness. However, the results show a more conservative count. A possible explanation for this difference is that the SFS method does not explore all possible feature combinations. It was observed that model performance increases rapidly for small feature subset and that the search is terminated pre-maturely. Results from the exhaustive method showed however, that unexplored feature combinations performed better on the same validation recordings. Furthermore, as the disregarded features are barely selected (seen in Figure 4c), it is assumed that feature pre-selection based on correlation with the steering-torque is justified.

Selecting the weight trade-off is based on a combination of estimation accuracy and smoothness scores and feature relevance. It is observed from Figure 5 that accuracy and smoothness scores remain approximately constant over the



(a) Model accuracy (b) Estimation smoothness

Fig. 5: Influence of metric weights on model performance for the exhaustive wrapper method.

range 100/0 to 40/60, averaging 89% and 2.94Nm/s, respectively, while for the same range overall feature relevance decreases. Basically, similar accuracy and smoother estimation is achieved with fewer features and thus simpler models. However, given the outliers, equal model performance is not guaranteed for all recordings. The outliers correspond to the recordings of the "Novice" participant and potentially indicates decreased model performance for novice drivers.

To visualize model performance, Figure 6 compares the estimated signals of models with different weight trade-offs. The figure shows that a smooth model is only able to approximate the true steering-torque and lacks accuracy. As more weight is given to accurate estimations, approximation quality increases rapidly. For further analysis, a 50/50 weight trade-off is preferred as it objectively provides a balance between estimation accuracy and estimation smoothness, while reducing model complexity. Increasing the weight on smoothness significantly degrades model accuracy, while increasing the weight on accuracy further degrades estimation smoothness.

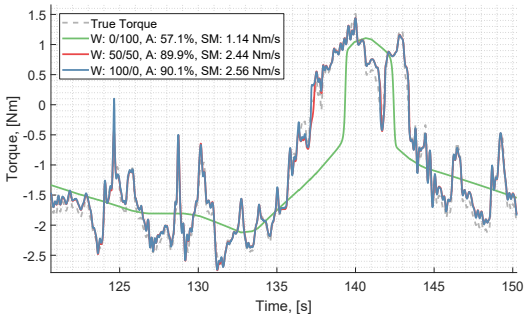


Fig. 6: Steering-torque estimations of the optimal models for different weight trade-offs.

C. Optimal Feature Subset

Final performance still depends on a specific feature subset. As only partial feature relevance overlap between methods was observed, the new subsets are based on the exhaustive (1), SFS (2), and combined (3) feature relevance. All combinations are found in Appendix, Table A.2. For each subset, new two-state HMMs were trained and performance on the validation recordings is compared for a 50/50 weight trade-off in Figure 7. The baseline model, defined in Section IV-C, is the most accurate (avg. 91%) but also the least smooth (avg. 5.82 Nm/s). Except for "Strict 2" and "Mild 2" (producing smooth estimations, avg. 1.22 Nm/s, but inaccurate, avg. 62%), all remaining selection strategies improved upon the baseline model based on their respective performance scores. As each selection was able to achieve similar accuracies (89±0.08%), sorting is dictated by the respective smoothness scores. Noticeable is that the selection with the least amount of features, "Strict 1", is able to achieve the smoothest estimations (2.86 Nm/s) and therefore the best performance score. For this reason, "Strict 1" is selected as the preferred feature subset, consisting of the steering wheel angle, θ_{SWA}, and the yaw rate, $\dot{\psi}$.

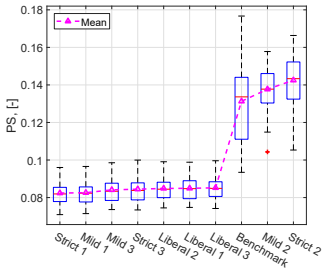


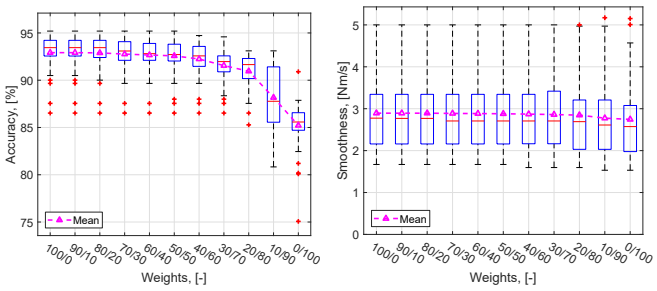
Fig. 7: Performance score of subset strategies

D. State Selection

New models were trained for $K \in \{1, ..., 20\}$. The influence of metric weights on state selection was mapped in ten percent intervals. From Figure 8 it is observed that more hidden states are preferred for more accurate models, while fewer hidden states are in favor of smoother estimations. Furthermore, selecting the number of hidden states based on the average K over individual recordings shows a more conservative state selection for a majority of the settings and should thus be preferred over the average PS selection. However, what the performance metrics in Figures 8a and 8b show is that selecting fewer hidden states significantly reduces model accuracy while only marginally improving the smoothness. As both metrics stay virtually constant for weights over 50% (resp.), a balanced weight trade-off is selected. Looking at Figure 8c, this results in a selection of five hidden states.

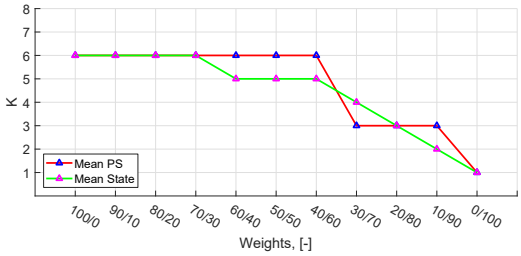
E. Model Testing

To put final model performance into perspective, a Baseline (BL) model was trained with all eighteen candidate features and five hidden states. Both the Generic Driver (GD) model and BL model were evaluated on the test set. Comparing average model performance in Figures 9a and 9b, shows that the GD model is able to match BL model accuracy while simultaneously producing 37% smoother estimations on average. This is best observed in Figure 9c, where the GD model significantly reduces noise in the estimated steering-torque. However, model performance did not improve equally for each participant. Steering behaviour of the "Novice" participant was the hardest to estimate both accurately and smoothly, averaging 90% and 4.22 Nm/s respectively, and is best observed in Figure 9d. The reduced model performance



(a) Model accuracy

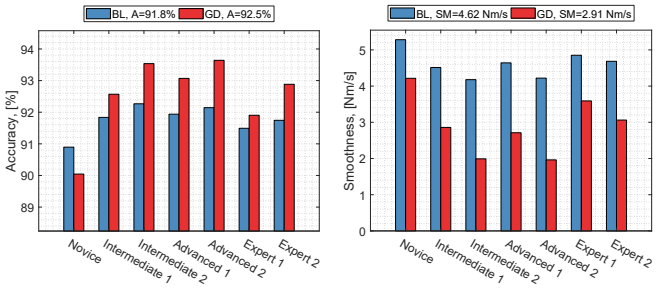
(b) Estimation smoothness



(c) State Selection

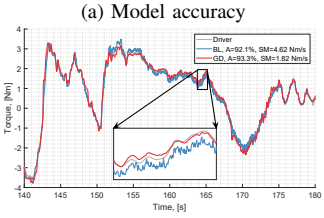
Fig. 8: Influence of metric weights on state selection for an HMM model with "Strict 1" feature selection.

can be explained by observing the more abrupt steering behaviour of the participant itself, which causes the model to adjust the estimations accordingly, reducing the metric scores. Among the remaining participants, performance appears to be dependent on individual steering behaviour rather than skill level. However, current participant sample size is insufficient to draw general conclusions and this observation will have to serve as an hypothesis for further research.

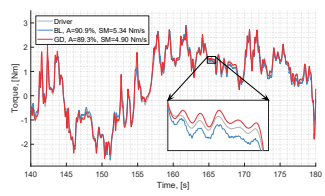


(a) Model accuracy

(b) Estimation smoothness



(c) Best performing test-recording



(d) Worst performing test-recording

Fig. 9: Performance Comparison between Baseline- (BL) and Generic Driver (GD) model.

VII. Conclusion

The HMM-based method presented in this work addressed the lack of data-driven approaches to driver steering-torque behaviour modelling by modifying the state emission probabilities. With the aid of the parameter selection framework, a generic driver model was able to accurately estimate driver steering-torque for multiple drivers while keeping the estimations as smooth as possible. Results on a separate test set indicated that model performance is dependent on individual driver behaviour, rather than driver skill level.
The implementation of the weighted performance score in the parameter selection framework, allowing a trade-off between accuracy and estimation smoothness, provided the insight to appropriately select the free model parameters while preventing model overfitting. By first selecting the feature subset based on feature relevance, preferred basic model behaviour can be designed. Subsequent state selection enables to further fine-tune this behaviour.

VIII. Future Work

While the results show the potential of the proposed approach for modelling driver steering-torque behaviour, parameter choices were based on objective metrics without taking into account driver steering feel. In order to validate the selected parameters, directions of future work include the implementation of the proposed driver model in a DSAS. For example, a Lane Keeping Assist controller similar to [13] and conduct subjective simulator experiments comparing different weight trade-off settings. Once validated, other directions include model robustness to noise, online learning approaches (e.g. [29]) for adaptation to individual and time-varying driver behaviour, and a wider scope of driving scenarios (varying speed, traffic, off-highway, etc.). Furthermore, as was proposed by the authors of [15], the hidden states can be organized hierarchically to represent not only lower- but also higher-level behaviour, such as lane-changing or overtaking. To the best of the author's knowledge, combining multiple HMM to explain more complex steering behaviour has not yet been investigated in combination with GMR estimation.

Appendix

A. Model Performance Metrics

- Root-mean-square-error (RMSE) of estimated steering-torque, $RMSE_{T,est}$,

$$RMSE_{T,est} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(T_{i,est} - T_{i,d}\right)^2} \tag{8}$$

- Model estimation accuracy, $A_{T,est}$,

$$A_{T,est} = \left[1 - \frac{1}{SD(T_d)} RMSE_{T,est}\right] \times 100 \tag{9}$$

- Smoothness of estimated steering-torque, $SM_{T,est}$,

$$SM_{T,est} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\dot{T}_{i,est} - \bar{\dot{T}}_{est}\right)} \tag{10}$$

B. List of investigated Candidate Features

TABLE A.1
CANDIDATE FEATURES.

Feature Name	Symbol	Unit
Driver Input		
Steering Wheel Angle	θ_{SWA}	rad
Steering Wheel Velocity	$\dot{\theta}_{SWA}$	rad/s
Steering Wheel Acceleration	$\ddot{\theta}_{SWA}$	rad/s^2
Vehicle Dynamics		
Lateral Velocity	v_y	m/s
Lateral Acceleration	a_y	m/s^2
Slip Angle	β	rad
Yaw Rate	$\dot{\psi}$	rad/s
Yaw Acceleration	$\ddot{\psi}$	rad/s^2
Roll Angle	ϕ	rad
Roll Velocity	$\dot{\phi}$	rad/s
Roll Acceleration	$\ddot{\phi}$	rad/s^2
Road Preview		
Deviation Distance @0m	e_{y0}	m
Deviation Angle @0m	$e_{\psi 0}$	rad
Road Curvature @0m	ρ_0	1/m
Deviation Angle @10m	$e_{\psi 10}$	rad
Road Curvature @10m	ρ_{10}	1/m
Deviation Angle @30m	$e_{\psi 30}$	rad
Road Curvature @30m	ρ_{30}	1/m

C. List of Selection Strategies based on feature relevance

TABLE A.2
SELECTION STRATEGIES

	θ_{SWA}	$\dot{\psi}$	ρ_0	ρ_{30}	ρ_{10}	v_y	$e_{\psi 30}$	β	e_{y0}	$e_{\psi 0}$
Counts										
Exhaustive	42	41	24	20	16	11	4	4	0	0
SFS	7	4	1	21	0	0	37	1	8	2
Strategies										
Strict 1	X	X								
Strict 2							X			
Strict 3	X	X					X			
Mild 1	X	X	X	X	X					
Mild 2				X			X			
Mild 3	X	X	X	X	X		X			
Liberal 1	X	X	X	X	X	X	X	X		
Liberal 2	X	X	X	X			X	X	X	X
Liberal 3	X	X	X	X	X	X	X	X	X	X

REFERENCES

[1] M. Martens and G. Jenssen, *Behavioral Adaptation and Acceptance*, pp. 117–138. London: Springer London, 2012.

[2] D. Abbink, M. Mulder, and E. Boer, "Sharing control with haptics: Seamless driver support from manual to automatic control," *Human Factors*, vol. 54, no. 5, pp. 786–798, 2012.

[3] A. D.A. and M. M., "Neuromuscular analysis as a guideline in designing shared control," in *Advances in Haptics* (M. H. Zadeh, ed.), ch. 27, Rijeka: IntechOpen, 2010.

[4] J. Sjöberg, E. Coelingh, M. Ali, M. Brännström, and P. Falcone, "Driver models to increase the potential of automotive active safety functions," in *2010 18th European Signal Processing Conference*, pp. 204–208, 2010.

[5] M. Hasenjäger and H. Wersing, "Personalization in advanced driver assistance systems and autonomous vehicles: A review," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7, 2017.

[6] S. Kolekar, W. Mugge, and D. Abbink, "Modeling intradriver steering variability based on sensorimotor control theories," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 3, pp. 291–303, 2018.

[7] L. Saleh, P. Chevrel, F. Mars, J. Lafay, and F. Claveau, "Human-like cybernetic driver model for lane keeping," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 4368–4373, 2011. 18th IFAC World Congress.

[8] T. Niu and D. Cole, "A model of driver steering control incorporating steering torque feedback and state estimation," 2020.

[9] F. Mars and P. Chevrel, "Modelling human control of steering for the design of advanced driver assistance systems," *Annual Reviews in Control*, vol. 44, pp. 292–302, 2017.

[10] A. Lazcano, T. Niu, X. Akutain, D. Cole, and B. Shyrokau, "Mpc-based haptic shared steering system: A driver modelling approach for symbiotic driving," *IEEE/ASME Transactions on Mechatronics*, pp. 1–1, 2021.

[11] S. Jugade, *Shared control authority between human and autonomous driving system for intelligent vehicles*. Theses, Université de Technologie de Compiègne, Sept. 2019.

[12] M. Flad, C. Trautmann, G. Diehm, and S. Hohmann, "Individual driver modeling via optimal selection of steering primitives," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 6276–6282, 2014. 19th IFAC World Congress.

[13] S. Lefèvre, Y. Gao, D. Vasquez, H. Tseng, R. Bajcsy, and F. Borrelli, "Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control," in *12th International Symposium on Advanced Vehicle Control*, (Tokyo, Japan), 2014.

[14] C. Macadam, "Understanding and modeling the human driver," *Vehicle System Dynamics*, vol. 40, no. 1-3, pp. 101–134, 2003.

[15] A. Pentland and A. Liu, "Modeling and Prediction of Human Behavior," *Neural Computation*, vol. 11, pp. 229–242, 01 1999.

[16] C. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York: Springer-Verlag, 2006.

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

[18] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[19] Y. Tian, L. Sigal, H. Badino, F. de la Torre, and Y. Liu, "Latent gaussian mixture regression for human pose estimation," in *Computer Vision – ACCV 2010*, (Berlin, Heidelberg), pp. 679–690, Springer Berlin Heidelberg, 2011.

[20] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[21] D. Mircea, B. Shyrokau, A. Ocariz, and X. Akutain, "Torque control for more realistic hand-wheel haptics in a driving simulator," in *Proceedings of the Driving Simulation Conference 2019 Europe VR* (A. Kemeny, F. Colombet, F. Merienne, and S. Espié, eds.), (Strasbourg, France), pp. 35–42, Driving Simulation Association, 2019.

[22] M. Karagiannopoulos, D. Anyfantis, S. Kotsiantis, and P. Pintelas, "Feature selection for regression problems," *Proceedings of the 8th Hellenic European Research on Computer Mathematics & its Applications, Athens, Greece*, vol. 2022, 2007.

[23] S. Adams and P. Beling, "A survey of feature selection methods for gaussian mixture models and hidden markov models," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1739–1779, 2019.

[24] J. Li, K. Cheng, S. Wang, F. Morstatter, R. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, Dec. 2017.

[25] F. Faller, *Utilization of a hidden-markov-model for the prediction of lane change maneuvers*. PhD thesis, Master's thesis, TU Darmstadt, Knowledge Engineering Group, 2016.

[26] M. Liwicki and H. Bunke, "Feature selection for hmm and blstm based handwriting recognition of whiteboard notes," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 05, pp. 907–923, 2009.

[27] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1200–1205, 2015.

[28] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, 2004.

[29] J. Geukes, *Personalized Driving Assistance to Predict Lane Change Manoeuvres*. PhD thesis, Master's thesis, TU Darmstadt, Knowledge Engineering Group, 2016.

A

Hidden Markov Models & Gaussian Mixture Regression

The goal of a driver steering-torque model is to accurately describe the relation in which the driver translates perceived information about the driving scenario to a steering-torque action on the steering-wheel [24]. In the current work, the behaviour of the driver is assumed to consist of a set of discrete internal states, each with their own control behaviour, sequenced together with a Markov chain [31], known as a Hidden Markov Model (HMM). The control behaviour is assumed to consist of single multivariate Gaussian normal distributions, one for each state. This allows the driver steering-torque to be estimated with Gaussian Mixture Regression (GMR), a multivariate regression method.

This chapter elaborates on the model structure that is an HMM in Appendix A.1. The section discusses what parameters define the model and how these parameters are learned from the data. The discussed theory is based on Chapter 13 of the book *"Pattern Recognition and Machine Learning"* by Christopher M. Bishop [5] and the paper *"A Tutorial on Hidden Markov Models"* by Lawrence R. Rabiner [32]. Furthermore, Appendix A.2 explains the concept of model inference with GMR and how it is applied to HMMs. The discussed theory is based on the paper *"Latent Gaussian Mixture Regression for Human Pose Estimation"* by Yan Tian et al. [37] and the paper *"Learning and Reproduction of Gestures by Imitation"* by Sylvain Calinon et al. [6]. This chapter is concluded with the software implementation in Appendix A.3.

A.1. Hidden Markov Models

The non-holonomic properties of road vehicles make that successive observations on the driving scenario cannot be assumed to be independent. Therefore, estimating the next value of driver steering-torque can be considered as a problem where we wish to estimate the next value in a time-series of observations. A suitable model to explain the dependencies between successive observations in a time-series is the Hidden Markov Model (HMM).

An HMM is a probabilistic modelling method that models a double stochastic process of which one is unobservable, commonly referred to as process states, and as a function of these process states produces another, this time observable, stochastic process. A simple example to get a feeling for this concept suggested by Rabiner in [32] is a set of N urns that each contain a set of M colored balls. A person will pick according to some random hidden process one specific urn and from that urn will pick one ball. Subsequently, the person only shows you the ball color without telling you from which urn the ball came. The person puts the ball back in the urn it came from, and repeats the process. By doing so, you end up with a finite observable sequence of ball colors that was generated according to the random hidden process. The observed sequence can then be used as the observable output of the HMM.

Coupling this idea to modelling human behaviour, Pentland first proposed in [31] that the human driver can be considered as the unobservable stochastic process, consisting of an unknown number of states, controlling the lateral movement of the vehicle. In the 2014 work [22], Lefevre et al. proposed this method for modelling driver steering-angle behaviour. The model is used for detecting unintentional lane changes and subsequently generates a reference signal for a Model Predictive Controller (MPC) to steer the vehicle back in the lane, mirroring the observed drivers' behaviour as close as possible. In the present work it is proposed to use this method for modelling driver steering-torque behaviour with the intended application of integrating the model in a continuous Driver Steering Assistance System (DSAS). The HMM aims to model a joint probability density function between the driving scenario (e.g. steering wheel angle, lane center deviation, and road curvature), denoted as a feature vector $F \in \mathbb{R}^{N_f}$ of length N_f, and the driver steering-torque, denoted as T_d. These observations, denoted as $x = [F, T_d]^\mathsf{T}$, will serve as the observable output for the HMM.

A.1.1. Structure

A Hidden Markov Model consists of a set of discrete latent model states s. Each state is responsible for a different part of the observed output x, through a conditional probability distribution $p(x|s)$ (e.g. a vector of ball color probabilities). It is then assumed that the state sequence $S = \{s_1, ..., s_T\}$ forms a first-order Markov chain, meaning the current state s_t depends solely on the previous attained state s_{t-1} through a conditional probability distribution $p(s_t|s_{t-1})$. By doing so it can be shown with graph theory ([5], p607) that the current observation x_t is dependent on all previous observations $\{x_1, ..., x_{t-1}\}$, while limiting the number of model parameters needed to describe the model. The corresponding graph network of the joint state-observation sequence is illustrated in Figure A.1.

The conditional probability distributions $p(s_t|s_{t-1})$ and $p(x|s)$ are respectively called the *state transition distribution* and the *state emission distribution*. Since an HMM consists of discrete latent states, the state transition distribution $p(s_t|s_{t-1})$ corresponds to a matrix where each element denotes the probability of transitioning from one state to another. This matrix will be denoted as **A**. In this work, a fully connected HMM is used, meaning all states can be reached from any state including the same state, as this resembles the ability of the driver to switch between all types of behaviours including keeping the same behaviour. Other HMM types exist and can be found in [32]. To initialize the state sequence S at time $t = 1$, a marginal probability distribution, denoted as $\pi = p(s_1)$, is introduced that describes the probability of starting in a particular state k.

In contrast to the urn-ball example, where the state emission distribution $p(x|s)$ is a vector of probabilities because there is a finite number of colors, modelling the continuous driving observations

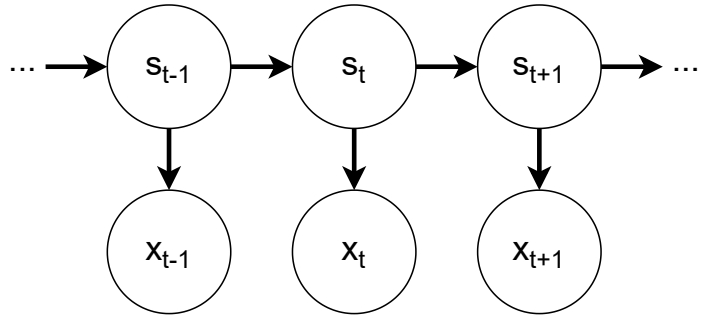


Figure A.1: *Graphical interpretation of a Hidden Markov Model [5]. The arrows indicate the conditional probability distributions between states s and observations x. As depicted, the current state s_t is dependent only on on the previous attained state s_{t-1} through the conditional probability distribution $p\,(s_t|s_{t-1})$ and the current observation x_t is dependent only on the current state s_t through the conditional probability distribution $p\,(x_t|s_t)$.*

$x = [F, T_d]^\mathsf{T}$ requires a continuous distribution. Therefore, each state emission distribution $p(x|s)$ is assumed to be a single multivariate Gaussian distribution, denoted as $b_k \sim \mathcal{N}(\mu_k, \Sigma_k)$ for state k. The set of all state emission distributions will be denoted as B. The mean μ_k and covariance matrix Σ_k of each state distribution consists of the mean and covariance of each measured feature that is described by state k, concatenated as follows,

$$\mu_k = \begin{bmatrix} \mu_k^F \\ \mu_k^T \end{bmatrix} \text{ and } \Sigma_k = \begin{bmatrix} \Sigma_k^{FF} & \Sigma_k^{FT} \\ \Sigma_k^{TF} & \Sigma_k^{TT} \end{bmatrix} \tag{A.1}$$

where $\mu_k^F \in \mathbb{R}^{N_f}$ is the mean vector of the measured features, $\mu_k^T \in \mathbb{R}$ is the mean value of the steering-torque, $\Sigma_k^{FF} \in \mathbb{R}^{N_f \times N_f}$ is the auto-covariance matrix for the measured features, $\Sigma_k^{TT} \in \mathbb{R}$ is the auto-covariance for the steering-torque, and $\Sigma_k^{TF} \in \mathbb{R}^{1 \times N_f}$ and $\Sigma_k^{FT} \in \mathbb{R}^{N_f \times 1}$ are the cross-covariance matrices of the measured features and the steering-torque.

Summarizing, to fully describe an HMM, the parameter set $\theta = \{K, \pi, A, B\}$ needs to be defined where,

1. K is the number of model states.
2. $\pi = p\,(s_1)$ is the marginal probability distribution of initializing the state sequence in state k at time $t = 1$.
3. $A = p\,(s_t|s_{t-1})$ is the conditional state transition matrix.
4. $B = p\,(x_t|s_t)$ is the set of conditional state emission distributions.

Given the state sequence $S = \{s_1, \ldots, s_T\}$, and the observation sequence $X = \{x_1, \ldots, x_T\}$, both of length T, and the parameter set θ, the joint probability distribution over the states and the observations [5] can be written as,

$$p(X, S|\theta) = p(s_1|\pi)\left[\prod_{t=2}^{T} p(s_t|s_{t-1}, A)\right]\prod_{t=1}^{T} p(x_t|s_t, B) \tag{A.2}$$

How the parameter set θ is defined will be explained in the following section.

A.1.2. Training an HMM

The next step is to find the set of model parameters θ that maximizes the likelihood of observing the observation sequence X. The likelihood function is obtained by marginalizing over the latent variables

S in the joint probability density function of Equation (A.2) as,

$$p(X|\theta) = \sum_{S} p\,(X, S|\theta) \tag{A.3}$$

However, since the latent variables are unobservable, Equation (A.3) is equivalent to summing over all possible state sequences that the stochastic process could go through. There exists no way of solving this problem analytically [32]. Therefore, iterative procedures like the Expectation-Maximization (EM) algorithm [8] are used to locally maximize the likelihood function of incomplete (or latent) data. The method iterates over two steps.

1. In the Expectation (E)-step a posterior distribution over the latent variables $p(S|X, \theta^{old})$ is computed, in which θ^{old} is the current parameter setting initialized with K-means clustering, to evaluate the expectation of the logarithm of the complete-data likelihood function Equation (A.3) [5] defined as the Q-function $Q(\theta, \theta^{old})$ in [8],

$$Q\left(\theta, \theta^{old}\right) = \sum_{Q} p\left(S|X, \theta^{old}\right) \ln p\,(X, S|\theta) \tag{A.4}$$

   which is a function of the model parameters θ.

2. The Maximization (M)-step then maximizes the Q-function Equation (A.4) w.r.t. the model parameters θ as,

$$\theta^{new} = \arg\max_{\theta} Q\left(\theta|\theta^{old}\right) \tag{A.5}$$

   until a local optimum is found or an iteration limit is reached.

To give an intuitive explanation on the workings of the EM-algorithm, Rabiner gives a derivation of the re-estimation equations for updating the model parameters θ in [32]. First, two variables are introduced as,

1. The *joint posterior distribution of two successive states* $\xi_{t,kj}$ is defined as the probability of being in state k at time t, and state j at time $t+1$, given the observation sequence X and the model parameters θ,

$$\xi_{t,kj} = p(s_t = k, s_{t+1} = j|X, \theta) \tag{A.6}$$

2. The *marginal posterior distribution of the states* $\gamma_{t,k}$ is defined as the probability of being in state k at time t, given the observation sequence X and the model parameters θ. For a model of K states this can be expressed in terms of $\xi_{t,kj}$ as,

$$\gamma_{t,k} = p(s_t = k|X, \theta) = \sum_{j=1}^{K} \xi_{t,kj} \tag{A.7}$$

Summing $\xi_{t,kj}$ over the observation time $t = \{1, ..., T-1\}$ results in a quantity representing the *expected number of transitions from state k to state j during the observation sequence X*. Similarly, summing $\gamma_{t,k}$ over time represents the *expected number of times state k is visited*, Equations (A.8a) and (A.8b) respectively.

$$\sum_{t=1}^{T-1} \xi_{t,kj} = E[\text{transition from state } k \text{ to state } j] \tag{A.8a}$$

$$\sum_{t=1}^{T-1} \gamma_{t,k} = E[\text{visit state } k] \tag{A.8b}$$

To determine the quantities $\xi_{t,kj}$ and $\gamma_{t,k}$ efficiently, Baum et al. introduced the Forward-Backward Algorithm [32], which is discussed in Appendix A.1.3. Both quantities serve as the basis for updating the model parameters θ. The probability of starting in a particular state π_k is updated as the frequency of being in state k at time instance $t = 1$,

$$\pi_k^{new} = \gamma_{1,k} \tag{A.9}$$

The conditional state transition probabilities A are updated as the ratio of the frequency of transitioning from state k to state j, Equation (A.8a), over the frequency of visiting state k, Equation (A.8b),

$$A_{kj}^{new} = \frac{\sum_{t=1}^{T-1} \xi_{t,kj}}{\sum_{t=1}^{T-1} \gamma_{t,k}} \tag{A.10}$$

Since the state emission probabilities B are assumed multivariate gaussians, the parameters μ_k and Σ_k defining the conditional state emission distribution b_k of state k are updated as,

$$\mu_k^{new} = \frac{\sum_{t=1}^{T} \gamma_{t,k} x_t}{\sum_{t=1}^{T} \gamma_{t,k}} \tag{A.11a}$$

$$\Sigma_k^{new} = \frac{\sum_{t=1}^{T} \gamma_{t,k} \, (x_t - \mu_k) \, (x_t - \mu_k)^\top}{\sum_{t=1}^{T} \gamma_{t,k}} \tag{A.11b}$$

A.1.3. Forward-Backward Algorithm

Since direct computation over all possible state sequences Equation (A.3) is computationally infeasible, Baum et al. introduced an efficient algorithm to calculate the variables $\xi_{t,kj}$ and $\gamma_{t,k}$, called the Forward-Backward Algorithm [32]. As the name suggests the algorithm consists of two parts, the forward and backward procedure.

1. In the forward procedure, the algorithm inductively calculates the probability of being in state k after seeing the partial observation sequence $\{x_1, \ldots, x_t\}$ until time t, given the model parameters θ, denoted as the *forward variable* $\alpha_{t,k}$,

$$\alpha_{t,k} = p(x_1, \ldots, x_t, s_t = k|\theta) \tag{A.12}$$

   The induction algorithm works its way *forward* along the observation sequence X starting at $t = 1$ as follows,

   Step 1: Initialization, $\alpha_{1,k} = \pi_k b_k(x_1) \qquad 1 \le k \le K$ (A.13a)

   Step 2: Induction, $\alpha_{t+1,k} = \left[\sum_{j=1}^{K} \alpha_{t,j} a_{kj}\right] b_k(x_{t+1}) \qquad 1 \le t \le T-1, \; 1 \le k \le K$ (A.13b)

   Step 3: Termination, $p\,(X|\theta) = \sum_{k=1}^{K} \alpha_{T,k}$ (A.13c)

2. In the backward procedure, the algorithm inductively calculates the probability of seeing the partial observation sequence starting from $t + 1$ until time T, given of being in state k at time t and the model parameters θ, denoted as the *backward variable* $\beta_{t,k}$,

$$\beta_{t,k} = p(x_{t+1}, x_{t+2}, \ldots, T|s_t = k, \theta) \tag{A.14}$$

Now the induction algorithm works its way *backwards* along the observation sequence X starting at time T as follows,

Step 1: Initialization, $\beta_{T,k} = 1 \qquad 1 \leq k \leq K$ (A.15a)

Step 2: Induction, $\beta_{t,k} = \sum_{j=1}^{K} a_{kj} b_j(x_{t+1}) \beta_{t+1,k} \qquad T-1 \leq t \leq 1,\ 1 \leq k \leq K$ (A.15b)

Step 3: Termination, $p(X|\theta) = \sum_{k=1}^{K} \beta_{1,k} \pi_{1,k} b_k(x_1)$ (A.15c)

The forward variable $\alpha_{t,k}$ and backward variable $\beta_{t,k}$ can now be used to define the variables $\xi_{t,kj}$ and $\gamma_{t,k}$ ([5], Section 13.2.2) introduced in Appendix A.1.2. The definition of $\xi_{t,kj}$ and $\gamma_{t,k}$ are as follows [32],

$$\xi_{t,kj} = p(s_t = k, s_{t+1} = j|X, \theta) = \frac{\alpha_{t,k} a_{kj} b_j(x_{t+1}) \beta_{t+1,j}}{\sum_{k=1}^{K} \sum_{j=1}^{K} \alpha_{t,k} a_{kj} b_j(x_{t+1}) \beta_{t+1,j}} \tag{A.16}$$

$$\gamma_{t,k} = p(s_t = k|X, \theta) = \sum_{j=1}^{K} \xi_{t,kj} = \frac{\alpha_{t,k} \beta_{t,k}}{\sum_{k=1}^{K} \alpha_{t,k} \beta_{t,k}} \tag{A.17}$$

Now the update equations Equations (A.9) to (A.11) can be used to update the parameter set θ.

A.2. Gaussian Mixture Regression

Using Gaussian distributions as the HMM state emissions, a probabilistic distribution between input and output features is learned directly in the form of a Gaussian Mixture Model (GMM). This allows for the derivation of explicit conditional distributions for model inference with Gaussian Mixture Regression (GMR) as employed in [37].

A.2.1. General Concept

GMR is a parametric regression method where it is assumed that the joint variables (x, y), where $x \in \mathbb{R}^{N_x}$ is some input vector of length N_x and $y \in \mathbb{R}^{N_y}$ is the target output vector of length N_y, follow a Gaussian mixture distribution with K components ([37], section 2) denoted as,

$$p(x, y) = \sum_{k=1}^{K} \pi_k p(x, y; \mu_k, \Sigma_k) \tag{A.18}$$

where $p(x, y; \mu_k, \Sigma_k)$ is the multivariate Gaussian distribution for mixture component k. The parameters π_k, $\mu_k = [\mu_k^x, \mu_k^y]^T$, $\Sigma_k = [\Sigma_k^{xx}, \Sigma_k^{xy}; \Sigma_k^{yx}, \Sigma_k^{yy}]$ indicate respectively, the mixture weights, means, and covariances of each mixture.

The global regression function can then be formulated using Bayes rule. The result is a mixture of distributions conditioned on the input variable and is defined as [37],

$$p(y|x) = \frac{p(x, y)}{p(x)} = \sum_{k=1}^{K} \omega_k p(y|x; \bar{\mu}_k, \bar{\Sigma}_k^2) \tag{A.19}$$

where $p(y|x; \bar{\mu}_k, \bar{\Sigma}_k^2)$ is the conditional distribution of mixture component k and ω_k is the mixing

weight defined as,

$$\omega_k = \frac{\pi_k p(x; \mu_k^x, \Sigma_k^{xx})}{\sum_{j=1}^{K} \pi_j p(x; \mu_j^x, \Sigma_j^{xx})} \tag{A.20}$$

Since the conditional distribution of a partitioned normal random variable is also a normal distribution [9], the mean $\bar{\mu}_k$ and covariance $\bar{\Sigma}_k$ of each mixture component k is defined as,

$$\bar{\mu}_k = \mu_k^{xx} + \Sigma_k^{yx}(\Sigma_k^{xx})^{-1}(x - \mu_k^{xx}) \tag{A.21a}$$
$$\bar{\Sigma}_k = \Sigma_k^{yy} - \Sigma_k^{yx}(\Sigma_k^{xx})^{-1}\Sigma_k^{xy} \tag{A.21b}$$

Given a new input signal x^{new}, the estimate of the output signal y^{est} is obtained by computing the expectation over the regression function Equation (A.19), which can be expressed as the weighted sum of the means $\bar{\mu}_k$ [37],

$$y^{est} = E\left[p(y|x)\right] = \sum_{k=1}^{K} \omega_k \bar{\mu}_k \tag{A.22}$$

From Equation (A.22) it follows that, since the parameters for the Gaussian emission distributions are learned during the training process, the regression step is performed with a limited number of computations and can therefore be considered a computationally efficient approach. Furthermore, GMR is able to handle several missing variables since it considers any input-to-output variable mapping, simply by using different partitioning of the variables [14].

A.2.2. GMR inference on HMM

In the case of an HMM, the Gaussian mixture distributions $p(x, y; \mu_k, \Sigma_k)$ are the respective Gaussian state emission distributions b_k for $k = 1, \ldots, K$. For the proposed driver steering-torque model, the steering-torque T_d is considered as the output variable y and the features F used in the model, are considered as the input variables x. Adapting the GMR method for inference on an HMM, the input-output variables are partitioned as was proposed in Equation (A.1) in Appendix A.1.1,

$$\mu_k = \begin{bmatrix} \mu_k^F \\ \mu_k^T \end{bmatrix} \text{ and } \Sigma_k = \begin{bmatrix} \Sigma_k^{FF} & \Sigma_k^{FT} \\ \Sigma_k^{TF} & \Sigma_k^{TT} \end{bmatrix} \tag{A.23}$$

where $\{\mu_k, \Sigma_k\}$ are the Gaussian distribution parameters for state k and $k = \{1, \ldots, K\}$ are the index of the individual K states. The distribution of each mixture component is now denoted as $p(T|F; \bar{\mu}_k, \bar{\Sigma}_k)$, where $\bar{\mu}_k$ and $\bar{\Sigma}_k$ are defined as,

$$\bar{\mu}_{t,k} = \mu_k^{FF} + \Sigma_k^{TF}(\Sigma_k^{FF})^{-1}(F_t - \mu_k^{FF}) \tag{A.24a}$$
$$\bar{\Sigma}_k = \Sigma_k^{TT} - \Sigma_k^{TF}(\Sigma_k^{FF})^{-1}\Sigma_k^{FT} \tag{A.24b}$$

where F_t is the vector of input feature values at the estimated time-step t. The estimation of the drivers' steering-torque $T_{d,t}^{est}$ can then be computed as,

$$T_{d,t}^{est} = \sum_{k=1}^{K} \omega_k \bar{\mu}_{t,k} \tag{A.25}$$

One issue remains, namely how to determine the mixture weights ω_k in Equation (A.25). Until now, the set of state emission distributions B was considered as a Gaussian Mixture Model as was done in the original GMR framework. The weight for mixture k was defined as the probability of the observation x belonging to mixture component k. However, to consider both spatial and temporal

information contained in the HMM, Calinon et al. proposed in [6] to represent the mixture weights ω_k with the HMM Forward Variable $\alpha_{t,k}$ (see Appendix A.1.3), representing the probability of observing the partial observation sequence $\{x_1, ..., x_t\}$ and being in state k at time t. To use the forward variable in Equation (A.25), the induction algorithm of Equation (A.13) is adapted by only involving the values of the input variables F as follows,

- The Adapted Forward Procedure

$$\text{Step 1: Initialization,} \quad \alpha_{1,k}^{F} = \pi_k b_k^{F}(x_1) \qquad 1 \leq k \leq K \tag{A.26a}$$

$$\text{Step 2: Induction,} \quad \alpha_{t+1,k}^{F} = \left[\sum_{j=1}^{K} \alpha_{t,j}^{F} a_{kj}\right] b_k^{F}(x_{t+1}) \qquad 1 \leq t \leq T-1,\ 1 \leq k \leq K \tag{A.26b}$$

$$\text{Step 3: Termination,} \quad p\,(X|\theta) = \sum_{k=1}^{K} \alpha_{T,k} \tag{A.26c}$$

where the emission distributions b_k^F is defined by the partitioned parameters $\mathcal{N}(\mu_k^F, \Sigma_k^{FF})$. With the newly defined mixture weights, the current driver steering-torque estimate $T_d^{est}(t)$ is calculated with the current feature values F_t as,

$$T_{d,t}^{est} = \sum_{k=1}^{K} \alpha_{t,k} \left(\mu_k^{FF} + \Sigma_k^{TF} \left(\Sigma_k^{FF}\right)^{-1} \left(F_t - \mu_k^{FF}\right)\right) \tag{A.27}$$

A.3. Software Implementation

The implementation of the proposed HMM+GMR approach is done in MATLAB 2017b, the program shared by all parties involved in this project. Additionally, the program used for data gathering, IPG CarMaker, provides a MATLAB tool to convert CarMaker-data to be used in the MATLAB format. The implementation of the HMM method is done with the open-source Probabilistic Modelling Toolkit (PMTK3 [1]) for MATLAB. This toolkit is designed by former professor Kevin P. Murphy at the University of British Columbia as additional source material for his book *"Machine Learning: A Probabilistic Perspective"* [28]. The included HMM implementation in the toolkit made the modelling process in this thesis work significantly easier, reducing the effort for training and validating models and opened up time for analysing the steering-torque estimation quality.

[1] https://github.com/probml/pmtk3

B

Data Acquisition

Learning the parameters of the state-transition and -emission probabilities with the Expectation- Maximization algorithm requires a sufficient amount of data. This chapter elaborates on the process of collecting driver steering-torque data for two data-sets, sharing common aspects. The first data-set was collected virtually in IPG CarMaker for a preliminary parameter sensitivity study, see Appendix C. The second data-set was collected experimentally in a fixed-base simulator for the analysis of model performance on naturalistic driving data, see Appendix E. The common aspects shared by the data-sets are the driving scenario (a three-lane highway designed from scratch), a high-fidelity non-linear vehicle model with proprietary steering system and additional sensors, and the collected data features.

This chapter is structured as follows. Appendix B.1 elaborates upon the driving scenario design process, followed by a description of the vehicle model in Appendix B.2. A description of the type of data features is given in Appendix B.3, after which this chapter is concluded with the allocation of the collected data in Appendix B.4.

B.1. Driving Scenario Design

In order to gain control on how the naturalistic driving data was generated, new virtual highway scenarios were designed. Twelve out of the 24 sections were specifically designed to maximize scenario variability and are used for generating training data. The twelve remaining sections were designed to provide new driving scenarios for model validation and testing. This section elaborates on the design choices made.

B.1.1. Pseudo-Randomized Road Generation

The generated highway sections consist of straight and curved road elements. Straight elements were defined by their length L [m]. Element lengths were sampled ranging from 25 to 200m with 25m intervals. Each curved element is defined by three parameters, being (1) the direction (left or right), (2) the radius R [m] (defining element curvature), and (3) the angle θ [deg] (defining element length by $S = R\theta \times \pi/180$). To ensure a wide variety of cornering situations similar to real-world highways, element radii were sampled ranging from R = 300 to 1200 meters with 100 meter intervals. The minimum radius (maximum curvature) provides sharp cornering up to a maximum lateral acceleration of 3 m/s^2 @ 100 km/h. The purpose of the scenarios were controlled by using specific settings for element radii, summarized in Table B.1. The idea behind the sharp (S), medium (M), and light (L) sections were to generate, respectively higher, medium, and lower amplitude and frequency steering motions. Combining each of the settings (SM, ML, SL) enabled to generate combinations of those frequencies, as sharper corners transition into shallower corners and vice versa. The angles θ ranged from 10 to 20 degrees with 5 degree interval. For model training purposes, two sections for each of the first six settings were generated. The remaining twelve sections for validation and testing purposes were generated with the seventh setting, combining all types of cornering situations.

Table B.1: *The 7 settings that define the virtual highway sections. For each setting, the radii R that defines the curvature were sampled at 100m interval, the angles θ that define the curve length were sampled at 5 degrees interval ranging from 10 and 20 degrees.*

Setting	Name	Radii [m]	Comments
1	S	300 - 600	Only sharp curved corners
2	M	600 - 900	Only medium curved corners
3	L	900 - 1200	Only lightly curved corners
4	SM	300 - 900	Combination of sharp and medium curved corners
5	ML	600 - 1200	Combination of medium and lightly curved corners
6	SL	300 - 600 & 900 - 1200	Combination of sharp and lightly curved corners
7	SML	300 - 1200	Combination of sharp, medium and lightly curved corners

Sections were generated by first calculating the relative coordinates of the road elements, and concatenating the elements randomly to obtain the absolute coordinates. However, to ensure a balanced dataset constraints were set, defined as follows. For each section, the following settings needed to be distributed as uniformly as possible,

- Element directions (including straight elements).
- Element radii (curved only).
- Element angles (curved only).

Finally, an elevation profile was added to the sections to enhance simulator immersion. Furthermore, it would ensure that driver road preview would occasionally be blocked by road crests, making the road less predictable. The profile was taken from an existing scenario in the simulator and by inverting both horizontally and vertically, four variations of this profile were created. Each section got assigned one at random.

To illustrate the complete road generation process, a flowchart is shown in Figure B.1. After coordinate interpolation in the IPG CarMaker Scenario Editor, road lines, hills and trees were added to the environment, providing reference points for the driver.

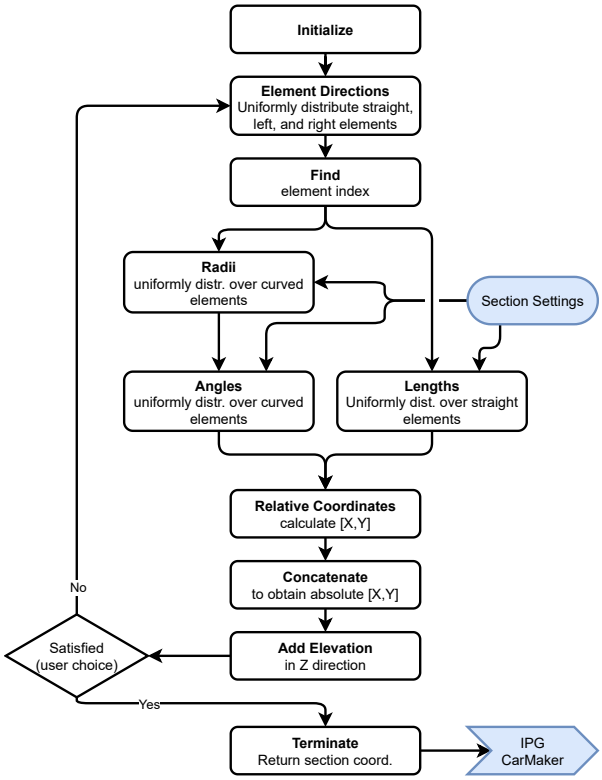


Figure B.1: *Flowchart illustrating the generation of new highway section coordinates.*

B.2. Vehicle Model

The vehicle dynamics were simulated with a custom non-linear vehicle model and a proprietary non-linear steering system [27] developed in Simulink by Toyota Motor Europe [1]. The parameters of the model are tuned to represent a Toyota production vehicle. The implementation of this model allows for the realistic simulation of real-world scenarios with high fidelity. The local coordinate system used in the model is the native right-handed system that is used in CarMaker. This system is located on the ground, at the rearmost point of the vehicle, and oriented such that the positive x-axis points in the driving direction, the positive y-axis points to the left of the vehicle, and the positive z-direction is up, see Figure B.2. In order to collect the necessary features, the vehicle model is equipped with several sensors. An Inertial Measurement Unit (IMU) is located at the vehicle Center of Mass (COM) and is used to measure inertial body movements (Vehicle Dynamics in Table B.2). Furthermore, three ideal road sensors are placed at respectively, the front axle of the vehicle, 10 meters, and 30 meters ahead of the vehicle to measure the road preview (Road Preview section in Table B.2. The road sensors simulate the use of a forward facing camera measuring the respective quantities. The non-linear steering system outputs the remaining driver input features, including driver steering-torque, steering-angle, and derivatives.

[1] https://www.toyota-europe.com/

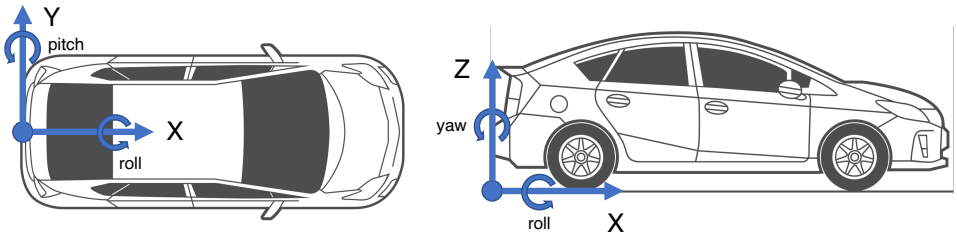


Figure B.2: *Vehicle coordinate system. Prius Illustration from* `https://www.vecteezy.com/free-vector/toyota`).

B.3. Candidate Features

The variables describing the driving scenario F_t at a particular time instant t and defining the dimensions of the Gaussian emission distribution B, are denoted as *features*. For the current study, an pre-selection of 18 features was made. The details on the selected feature set are presented in Table B.2. In the sensitivity study, all features were used to test the proposed modelling approach. With the data collected from the simulator experiment, all features were recorded and a parameter selection framework is implemented to select the optimal model parameters (see Appendix D).

Table B.2: *List of pre-selected candidate features that describe the driving scenario.*

Feature Name	Symbol	Unit	Real-world feasability
Driver Input			
Steering Wheel Angle	θ_{SWA}	rad	Steer Angle Sensor
Steering Wheel Velocity	$\dot{\theta}_{SWA}$	rad/s	SAS
Steering Wheel Acceleration	$\ddot{\theta}_{SWA}$	rad/s^2	SAS
Vehicle Dynamics			
Lateral Velocity	v_y	m/s	IMU
Lateral Acceleration	a_y	m/s^2	IMU
Slip Angle	β	rad	Estimation
Yaw Rate	$\dot{\psi}$	rad/s	IMU
Yaw Acceleration	$\ddot{\psi}$	rad/s^2	IMU
Roll Angle	ϕ	rad	IMU
Roll Velocity	$\dot{\phi}$	rad/s	IMU
Roll Acceleration	$\ddot{\phi}$	rad/s^2	IMU
Road Preview			
Deviation Distance @0m	e_{y0}	m	Camera
Deviation Angle @0m	$e_{\psi 0}$	rad	Camera
Road Curvature @0m	ρ_0	1/m	Camera
Deviation Angle @10m	$e_{\psi 10}$	rad	Camera
Road Curvature @10m	ρ_{10}	1/m	Camera
Deviation Angle @30m	$e_{\psi 30}$	rad	Camera
Road Curvature @30m	ρ_{30}	1/m	Camera

B.4. Scenario Allocation

For the proper validation and testing of performance and generalisation capabilities of the proposed modelling approach, the dataset is split into a training, validation and test set. As the purpose of the first twelve generated sections was to present the model with an as wide as possible range of cornering situations, these sections are used for the training of the HMM model. This comes down to approximately one hour of training data per participant. The remaining twelve sections were designed to present the model with new scenarios. For the sensitivity study, all twelve remaining sections are

used for generating validation data, totalling one hour of data. For the analysis on naturalistic driving data, the first six sections are used for model validation, totalling 30 minutes per participant, and the other six sections for model testing, another 30 minutes per participant.

C

Parameter Sensitivity Study

The parameters defining the state transition matrix and the state emission distributions of the Hidden Markov Model are learned from data by applying the Expectation-Maximization algorithm. However, not all parameters can be determined by this algorithm. The remaining free parameters to select are the type of features in the state emission distributions and the number of hidden states. To investigate how these model parameters influence the estimation behaviour, a preliminary parameter sensitivity study has been performed. The aim of the study was twofold. Firstly, an offline simulated experiment was performed in IPG CarMaker to gather virtual driving data. This data is then used for the training and evaluation of benchmark HMM models. Secondly, this data is used to investigate how varying of the type of features and number of model states impact the benchmark performance.

This chapter is structured as follows. Appendix C.1 explains how the virtual driver behaviours are generated, followed by an analysis of this behaviour in Appendix C.2. Evaluating the performance of the benchmark models is presented in Appendix C.3, followed by the results of varying the model parameters in Appendix C.4. The conclusions are highlighted in Appendix C.5.

C.1. Virtual Drivers

Driving data was generated by simulating two virtual behaviours following a continuous highway lane-keeping pursuit strategy at a fixed longitudinal vehicle speed of 100 km/h. To simulate driving behaviours, IPG CarMaker includes a build-in IPGDriver which allows to add human control actions (e.g. steering, throttle, braking, gear shifting, etc.) and limitations (e.g. longitudinal-, lateral accelerations) to the vehicle simulation. The IPGDriver model is based on an advanced PID controller using preview points and real driver measurements to increase behaviour realism [30]. The two driving styles used in the sensitivity study are based on the ”defensive” and ”aggressive” parameter presets that come with the IPGDriver, see Table C.1. The defensive preset is interpreted as a careful driver that limits corner cutting and absolute accelerations to a specific direction (e.g. first brake, then steer). The aggressive preset is interpreted as a sportier driver, allowing for increased corner cutting (thus increased lateral deviation from the lane center) and increased absolute accelerations (e.g. brake and steer simultaneously). Both presets are modified by adding a tolerated lateral lane center deviation of 0.5 and 1.0 meters, respectively, to increase (robotic) behavioural variability.

Table C.1: *Predefined IPGDriver parameter presets with modification in last column.*

Preset	**Max. long. acc.** [m/s^2]	**Max. long. dec.** [m/s^2]	**Max. lat. acc.** [m/s^2]	**Corner cut coeff.** [-]	**Tol. lat. dev.** [m]
Defensive	2	2	3	0.2	0.5
Aggressive	4	6	5	0.8	1.0

C.2. Virtual Steering Behaviour Analysis

Figure C.1 shows a section of the dataset where the difference in driving style between the two drivers can be observed. The example shows both drivers transitioning from a left hand curve (from 150 to 155s) to a right hand curve (from 155 to 165s), indicated by a positive to negative transition of both steering angle Figure C.1b and steering torque Figure C.1c at around 155s. It can be observed that the defensive driver (in blue) tries to catch up with each corner as the lateral deviation of the vehicle has the opposite sign related to the steering angle and steering torque in both curves. On the other hand, the aggressive driver (in red) can be seen to actively cut the corners as the lateral deviation remains the same sign as the steering angle and torque in both curves. Looking at these figures, the difference in behaviour is small but it can be observed that the aggressive driver requires slightly less steering-wheel-angle and -effort (torque) to negotiate the curves compared to the defensive driver, indicated by the lower peaks in Figure C.1b and Figure C.1c, respectively.

To quantify the behaviour of both drivers over the whole dataset, Table C.2 presents the root-mean-

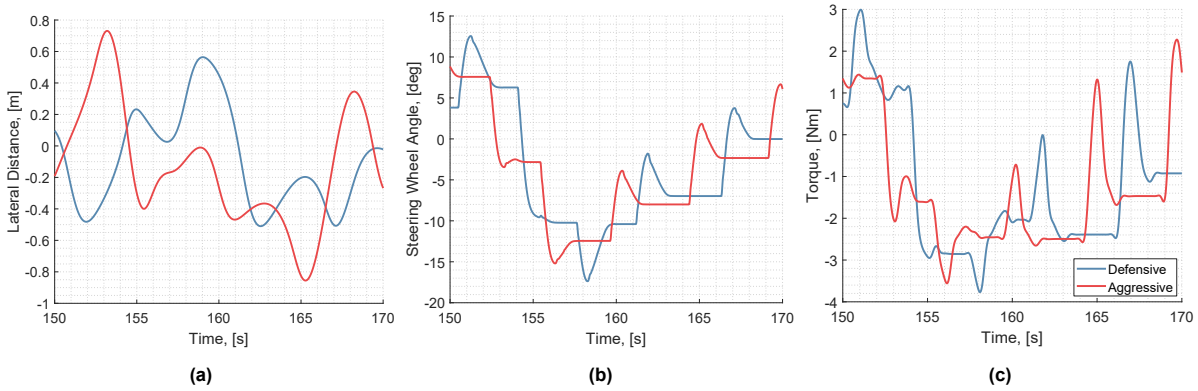


Figure C.1: *Comparing the driving behaviour of the defensive driver (in blue) and the aggressive driver (in red) on a section where the road transitions from a left-hand curve into a right-hand curve. (a) Lateral distance from the vehicle to the lane center. (b) Steering wheel angle. (c) Driver steering-torque input.*

square (RMS) and maximum value of the driver steering-torque, lateral deviation, velocity, and acceleration, steering wheel angle and velocity on the training dataset. The table shows that, on average, the aggressive driver applies slightly less steering torque T_d on the steering wheel compared to the defensive driver while allowing for an increased lateral deviation e_{y0} from the lane center. As a result of this behaviour a reduction in vehicle lateral response (v_y, a_y) is observed. The maximum value of the steering wheel velocity $\dot{\theta}_{SWA}$ further indicate that the aggressive driver is able to react with higher speed compared to the defensive driver, while limiting the steering range and effort, indicated by the lower maximum value of the steering wheel angle θ_{SWA} and steering torque T_d.

Table C.2: *Quantified Driver Characteristics over the whole dataset: RMS and maximum values of various measured variables.*

KPI	Unit	Defensive	Aggressive
RMS T_d	[Nm]	1.6705	1.6409
Max T_d	[Nm]	4.2637	4.0869
RMS e_y	[m]	0.4131	0.4665
Max e_y	[m]	1.1285	1.4975
RMS v_y	[m/s]	0.0626	0.0607
RMS a_y	[m/s^2]	0.9998	0.9653
RMS θ_{SWA}	[deg]	7.1144	6.8679
Max θ_{SWA}	[deg]	24.7497	22.7767
RMS $\dot{\theta}_{SWA}$	[deg/s]	6.5095	6.1365
Max $\dot{\theta}_{SWA}$	[deg/s]	68.0953	170.1870

Although the difference in behaviour is minimal, the main subjective conclusion on comparing both driving styles from Figure C.1 is that the aggressive driver shows a more anticipating steering behaviour compared to the more passive behaviour of the defensive driver. This is objectively summarized in Table C.2 as it shows that the aggressive driver requires less effort to follow the lane center, giving the impression of a more advanced driving style, compared to a more novice driving style of the defensive driver.

C.3. Model Performance

For the analysis of model performance, driver-dependent and generic HMM models were trained. The models were designed using the 18 candidate features from Table B.2 and five hidden states to ensure model complexity. To evaluate generalisation capabilities, the individual driver models are also cross-validated. From the results in Table C.3 it is observed that the difference in model performance is negligible. Both individual driver models perform equally, where the behaviour of the defensive driver is harder to estimate accurately. The generic driver model appears to perform as an average of both individual driver models, improving upon the defensive model, while performing marginally worse than the aggressive model. This means one of two things. Either each model has good generalisation capabilities, or the difference in behaviour is negligible and are variations of the same behaviour. In that case, intra-driver behaviour variability is captured well.

Table C.3: *Model performance on driver validation data. Metric Scores: $RMSE_{est}$ (Accuracy)*

Model	Defensive Driver	Aggressive Driver
Defensive Model	0.1650 Nm (89.87%)	0.1538 Nm (90.37%)
Aggressive Model	0.1657 Nm (89.82%)	0.1450 Nm (90.92%)
Generic Model	0.1621 Nm (90.05%)	0.1471 Nm (90.79%)

Even though both models fit the virtual driving data well, taking a closer look at the estimations in Figure C.2a reveals that the estimated signal fluctuates around the true steering-torque. As this can lead to undesirable steering wheel vibrations, a preliminary parameter sensitivity study investigated how these oscillations can be kept to a minimum.

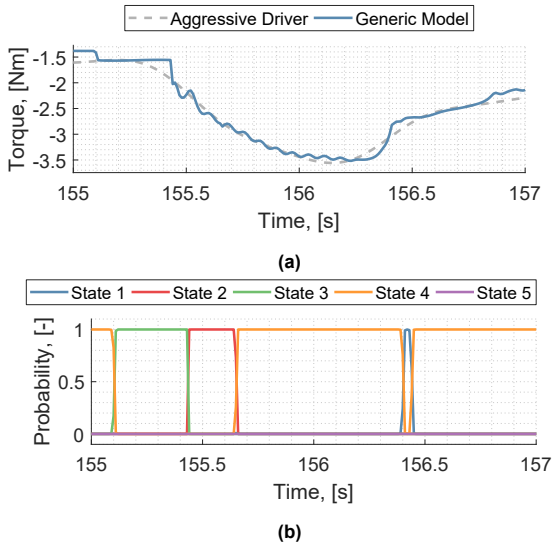


Figure C.2: *Virtual Model Estimation Performance. (a) Accurate but oscillating estimations around 156s. (b) State probabilities α_k indicate that oscillations are generated by single states.*

C.4. Parameter Sensitivity Study

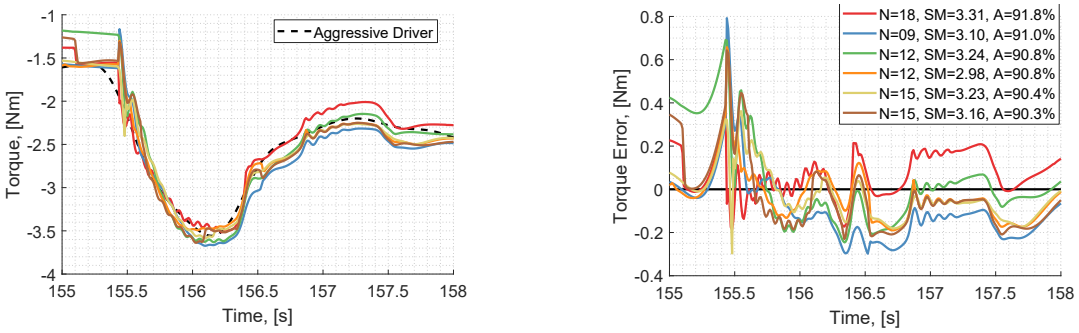
The forward variables $\alpha_{t,k}$ determine what state is dominant depending on the observation F_t at time t (see Appendix A.2.2). Therefore, they determine final model behaviour. From the side-by-side comparison in Figure C.2, it is observed that the fluctuating estimations are generated by individual states, while it was expected that the model would rapidly switch between different states to create such behaviours. The parameter that defines the behaviour of a single state is the chosen feature set F. As mentioned by Li et al. in [23], having too many features might result in overfitting. Therefore, the fluctuating estimations might be an indication that this has occurred.

To alleviate this, new *generic driver* models were trained with reduced feature sets, counting 15, 12, and 9 features. For each set, five models were trained with randomly selected feature combinations. To quantify the amount of oscillations in the estimations, a smoothness metric is defined as the standard deviation of the estimated steering-torque rate,

$$SM = SD\left(\dot{T}_{est}\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\dot{T}_{i,est} - \mu_{\dot{T}_{est}}\right)} \tag{C.1}$$

The smaller the value of SM becomes, the smoother the estimation is considered. Figure C.3 plots the estimated steering-torque of the original generic driver model (in red) containing all 18 features compared to the five best performing new models, sorted on the model accuracy. Two effects are observed. First, both reducing the number of features (N < 18) as well as the choice of features (for equal N) influence the smoothness of the estimated steering-torque. For example, two models with 12 features each achieve an accuracy of 90.8% but one produces a smoother estimation compared to the other (SM = 2.98 vs. 3.24). Remarkable is that reducing the number of features even further (N=9) makes the smoothness worse (3.10 vs. 2.98) but increases the accuracy (91.0% vs. 90.8%). The second effect is that the difference in accuracy of the new models compared to the original model is almost negligible. This means that a similar accuracy can be achieved with less information by properly selecting the set of features.

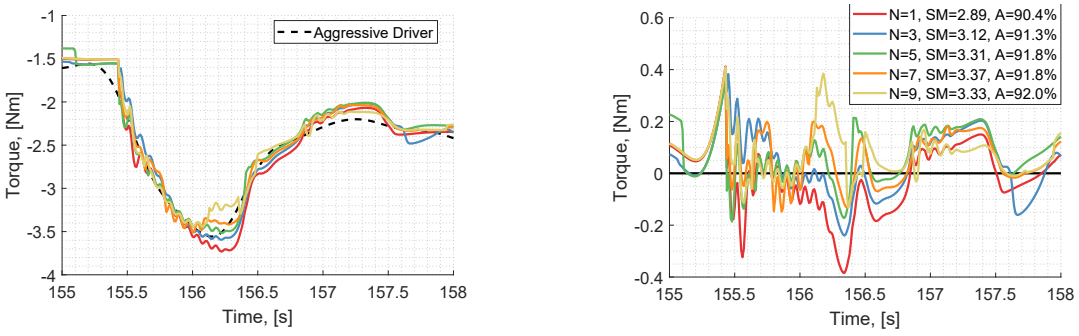
All previous presented models are trained with a fixed number of five model states to purely look at the influence of varying the features. As it has become clear how the behaviour of a single state is influenced, the number of model states K determines the overall model configuration. By fixing the



(a) *Estimation* **(b)** *Estimation Error*

Figure C.3: *Influence of varying the number of and specific selected features. (a) Instantaneous steering-torque estimation. (b) Estimation error w.r.t. the aggressive driver steering-torque.*

feature set, the influence of the number of hidden states can be investigated. A range of new driver individual models were trained with respectively 1, 3, 7, and 9 states. Figure C.4 plots the estimated steering-torque of the original model (with 5 states) compared to the newly trained models. Observed is that increasing the number of states results in a higher accuracy. However, the estimated steering-torque becomes less smooth as well (increased SM). Furthermore, with increased model complexity the risk of overfitting also increases ([11], Section 5.4.1, p33). Therefore, an overfitting criterion needs to be introduced to find the optimal number of model states.



(a) *Estimation* **(b)** *Estimation Error*

Figure C.4: *Influence of varying the number of states. (a) Instantaneous steering-torque estimation. (b) Estimation error w.r.t. the aggressive driver steering-torque.*

C.5. Conclusion

The proposed approach is able to capture and estimate the steering-torque behaviour of both drivers with high accuracy, validating the model approach on virtual driving data. Although marginal, the results showed that a defensive driving style is harder to estimate compared to an aggressive driving style. The small performance difference could be the cause of similar driving behaviours, meaning that behaviour variability is captured as well. Furthermore, the model is able to reproduce non-linear steering-torque by combining different state behaviours through the forward variable $\alpha_{t,k}$. However, using many features results in oscillations in the estimated signal, possibly indicating overfitting. By properly selecting the number of and specific set of features, followed by selecting the number of states K, can significantly reduce these oscillations. It is concluded that further investigation into the selection of model parameters is recommended. Therefore, Appendix D introduces a framework for selecting the optimal feature set and number of model states to balance model accuracy with estimation smoothness.

D

Parameter Selection Framework

In the parameter sensitivity study two things became apparent. Firstly, the 18 candidate features were selected without taking into account model limitations. Each feature was assumed to contribute towards accurately estimating driver steering-torque behaviour. However, it was observed that a similar accuracy can be achieved with less features, which simultaneously smoothened the estimation signal. Secondly, increasing the number of hidden states increased the estimation accuracy as the data structure can be captured in greater detail. However, more states also increased the oscillations. Both observations are an indication of the model overfitting on the training data. This motivated the search for the combination of features, called the feature set, and number of hidden states that improved the estimation quality.

This chapter provides additional background on the implemented parameter selection framework. The framework consists of a dimensionality reduction method (to systematically find an optimal feature set) and an overfitting prevention procedure (that selects the number of hidden states) to find the model parameters that optimize the estimation quality. The chapter is organized as follows. First an introduction to dimensionality reduction for HMMs is given in Appendix D.1. In order to evaluate model performance, a model performance score is introduced in Appendix D.2. Next, Appendices D.3 and D.4 provide additional material on, respectively, the implemented feature and state selection methods, as discussed in Chapter 2. Appendix D.5 summarizes the complete parameter selection process.

D.1. Feature Selection Background

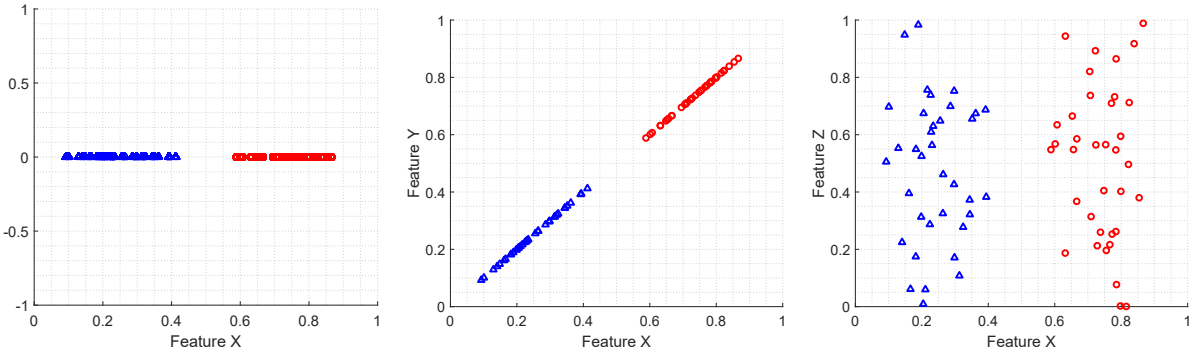
Including all 18 potential features in the model make that the Gaussian state emissions b_k become 18 dimensional joint probability distributions. Two issues can arise. Either, with a limited number of data points the data space quickly becomes sparse. This is called the curse of dimensionality [4] and decreases generalisation performance on new data. A possible solution is to record more data but in many applications the exact required data size is unknown and data recording can be a time consuming and/or expensive process. The second issue that can arise is that in the case of sufficient data, having too many features increases the risk of overfitting [23]. This also decreases generalisation capabilities and irrelevant or redundant features introduce noise that further degrades model performance. A possible solution for this problem is Dimensionality Reduction. The aim is to reduce the dimensionality of the data by constructing new lower dimensional features transformed from the original feature set, or filter out the most relevant existing features to optimize the model performance. Given that the model estimation is highly accurate but noisy, possibly due to the many included features, the current work proposes a dimensionality reduction method in order to find a feature subset with high accuracy but limited noise.

D.1.1. Dimensionality Reduction

Features can be categorized as either relevant, redundant or irrelevant, see Figure D.1. Relevant features are strongly related to the regression target and their contribution cannot be described by other features [19]. If it can be described by other features, it is considered redundant. In the case features are weakly correlated, they are considered irrelevant. Removing redundant, irrelevant (or noisy) features through dimensionality reduction improves learning speed, performance, generalization capabilities and model interpretation while simultaneously reducing the risk of overfitting and data storage [3, 23]. Dimensionality reduction methods can be broadly categorized as *feature extraction* or *feature selection* methods.

Feature extraction methods reduce the dimensionality of the data by transforming the original high dimensional data into a lower dimensional set by constructing new relevant features, eliminating the redundant or irrelevant part of the data. An example is Principal Component Analysis (PCA), projecting the original data onto the principal components while preserving as much variance as possible. The components are considered as the newly constructed feature set. Feature extraction is often applied if the original features lack physical meaning, e.g. the individual pixels in an image. However, interpreting the new features is a challenge. Furthermore, all original features still need to be collected in order to extract the new features.

In applications with features of physical meaning, feature selection methods are preferred. These



(a) *Relevant Feature X.* **(b)** *Redundant Feature Y.* **(c)** *Irrelevant Feature Z.*

Figure D.1: *An example of feature relevance [23]. (a) Feature X is able to separate samples into classes and is therefore relevant. (b) Feature Y separates the samples in the same way as Feature X, is therefore redundant and can be removed. (c) Feature Z is not able to separate the samples, is therefore irrelevant and should be removed.*

directly select a subset from the original features according to an evaluation algorithm, retaining the original meaning, thus increasing the interpretability of the final model. Furthermore, the disregarded features excluded from the subset no longer have to be stored. Since the current work directly measures features of the driving scenario with physical meaning to them (e.g. steering wheel angle, lateral velocity, road curvature, etc.), feature selection is the preferred dimension reduction method.

D.1.2. Feature Selection

Feature selection methods are further categorized into unsupervised and supervised methods. The former is commonly implemented for *classification* problems to cluster unlabeled data. Supervised methods are commonly used for both classification and *regression* problems since the correlation of a feature with respectively, class label or regression target is used as a measure for feature relevance [23]. In this work supervised methods are preferred since the steering-torque is considered as the regression target. Supervised feature selection methods are sub-categorized as filter, wrapper, embedded or hybrid methods, based on whether the model training algorithm is included in the process [17]. Each sub-category consists of a *selection-* and an *evaluation* algorithm. The latter evaluates the performance of candidate feature subsets generated by the former that searches to find the best performing subset.

- *Filter* selection methods determine univariate or multivariate feature relevance by respectively ranking single features or evaluating feature subsets, using an evaluation function. This function is based on feature measures and relation with the regression target, independent of the training algorithm. By being independent, filter methods are computationally cheap, but do not always remove all redundant features [19].
- *Wrapper* selection methods include the model training algorithm as a method of evaluating the subset performance. This allows such methods to determine dependency between multiple features within the selected subset, obtaining better performing subsets compared to filter methods. However, due to including the training algorithm they are computationally slow. Furthermore, they are biased towards the training algorithm, needing a separate validation dataset for validating the final subset.
- *Hybrid* selection methods are combinations of filter and wrapper methods with the aim to combine the advantages and alleviate the disadvantages of both methods (e.g. proposed in [16] for short-term load forecasting). First, a filter method reduces the high dimensional feature space by selecting one or several potential feature subsets. Subsequently, a wrapper method aims to find the best among them. Hybrid selection methods achieve the high accuracy paired with wrappers and efficiency paired with filters.
- *embedded* selection methods are feature selection methods embedded in the learning algorithm as part of the algorithms' functionalities.

Jović et al. concluded in [17] that there exists no single best method and the choice for a particular method is highly dependent on the application. They recommend future research to focus on optimizing *hybrid selection methods*, as they offer the best results.

D.1.3. Feature Selection for HMMs

Literature provides plenty of reviews on feature selection methods. The majority focuses on classification problems [7, 23, 38, 33], to a lesser extent on regression problems [19], or a combination of both [17]. However, supervised feature selection methods specifically designed for HMMs are scarce and research specific, as was concluded by Adams & Beling in [3].

An example of driver modelling specific feature selection is found in the 2016 thesis work [11] by Fabian Faller, where an extensive feature selection analysis was performed on 24 features for predicting

unintentional lane changes. Two wrapper-based selection methods were employed. The first method performs an *exhaustive search* through all possible combinations of 11 manually pre-selected features. To validate the resulting best feature subset, the second method performs a *greedy search* on all 24 features, starting with an empty feature set and adding the next feature that maximizes the model performance. With the implemented strategy Faller was able to successfully identify several optimal feature subsets.

The work by Faller deals with an HMM classification problem while the current work presents a regression problem. However, due to the positive obtained result and a lack of alternative supervised regression-based strategies specifically for driver modelling with HMMs, a similar approach is adopted.

D.2. Performance Score

Throughout the entire parameter selection process a single evaluation function is used to evaluate the performance of all trained models. The aim of the function is to balance estimation accuracy with signal smoothness. It is denoted as the Performance Score (PS) and expressed as,

$$PS = \omega_1 * \|(100 - A_{T,est})\| + \omega_2 * \|SM_{T,est}\| \tag{D.1}$$

where ω_1 and ω_2 are metric weights, $A_{T,est}$ is the accuracy metric, and $SM_{T,est}$ is the smoothness metric. A model with a lower PS is considered as a better model. The accuracy metric $A_{T,est}$ is defined as,

$$A_{T,est} = \left[1 - \frac{1}{SD(T_d)} RMSE_{T,est}\right] \times 100 \tag{D.2}$$

where $SD(T_d)$ is the standard deviation of the *observed* driver steering-torque, and $RMSE_{T,est}$ is the root-mean-square-error (RMSE) of the *estimated* steering-torque. The latter is defined as,

$$RMSE_{T,est} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(T_{i,est} - T_{i,d}\right)^2} \tag{D.3}$$

where N is the length of the used datarecording. Subsequently, the smoothness metric is defined as the standard deviation of the estimated steering-torque *rate* and expressed as,

$$SM_{T,est} = SD\left(\dot{T}_{est}\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\dot{T}_{i,est} - \bar{\dot{T}}_{est}\right)} \tag{D.4}$$

where $\bar{\dot{T}}_{est}$ is the mean of the estimated steering-torque rate, $\dot{T}_{est}$.

Lastly, the two metrics in the PS are normalized for two reasons. Firstly, in order to be able to sum the metrics up in the PS. Secondly, to be able to compare the PS of different models under consideration. The normalization constant is calculated as follows. For each data recording x, the normalization constant for metric n, c_n^x, is defined as the reciprocal of the sum of metric scores over M evaluated models, expressed as,

$$c_n^x = \frac{1}{\sum_{m=1}^{M} \text{Score}_{n,m}} \tag{D.5}$$

where n is either the estimation accuracy or estimation smoothness score and $\text{Score}_{n,m}$ is the metric score for model m.

D.3. Feature Selection Analysis

As no single best feature set exists for the entire validation dataset (see Appendix E.1.2, ”Feature Relevance Assumption”), feature relevance was assumed as a measure of feature importance. This section provides flowcharts for illustrating how feature relevance was obtained for the exhaustive- and the SFS feature selection methods, respectively shown in Figure D.2 and Figure D.3, as visual aid to the explanation in Section IV-C of Chapter 2.

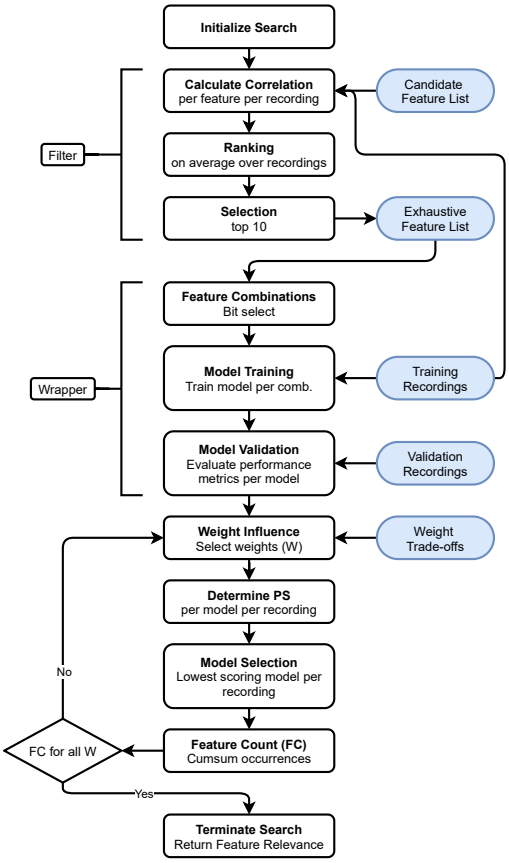


Figure D.2: *Flowchart illustrating the determination of feature relevance for the exhaustive feature selection method. First, the candidate features are filtered by ranking based on correlation with the steering-torque, forming the ”Exhaustive Feature List”. The exhaustive wrapper then trains an HMM for every combination (1023), and evaluates the performance metrics of all models separately. Then, for each weight trade-off and each recording, the performance score is calculated for each model. Lastly, for each recording the model with the lowest PS is determined and the features are counted cumulatively.*

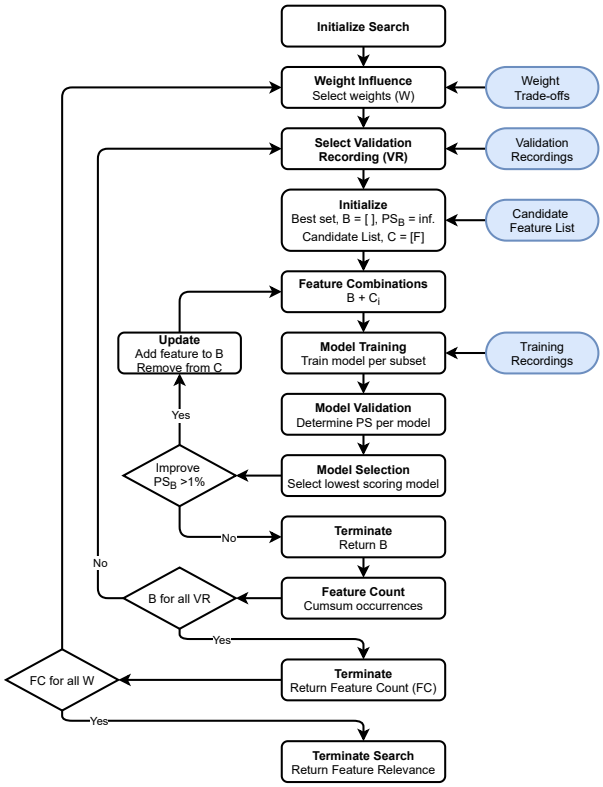
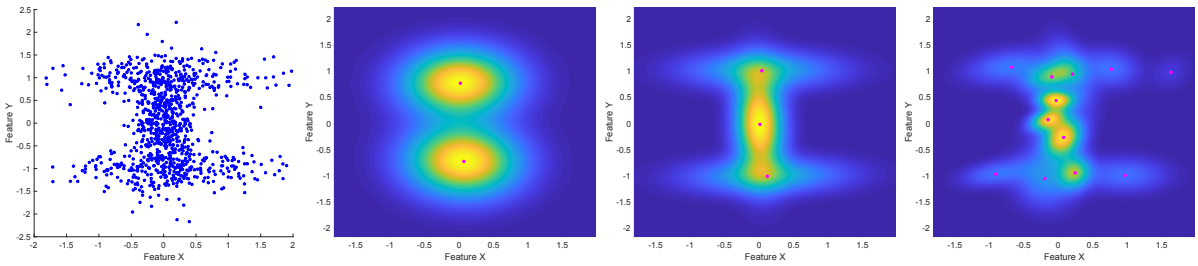


Figure D.3: *Flowchart illustrating the determination of feature relevance for the SFS feature selection method. As this method requires the performance score, first the weight trade-off is selected. For each recording and the selected weight trade-off, an empty "best set" is initialized. Then, for every combination of the "best set" and the remaining candidate features, a model is trained, metric scores evaluated, and performance score calculated. The best set is updated with the feature of the combination that maximizes the PS for that recording. This process is repeated until the PS is not improved. Then, features in the best set are counted cumulatively, and this process is repeated for all recordings and all weight trade-offs.*

D.4. State Selection

Increasing the number of hidden states allows to capture the driver steering-torque in greater detail. However, this also increases the model dependency on the training data, known as overfitting. For example, Figure D.4 shows the modelling of a two-feature dataset with a Gaussian mixture model (equivalent to the HMM Gaussian state emissions without the temporal relations). Increasing the number of hidden states, respectively shown in Figures D.4b to D.4d, clearly shows that the data structure is captured in greater detail with more states. However, by making the model too complex (Figure D.4d)



(a) *Dataset* **(b)** *Two-state model* **(c)** *Three-state model* **(d)** *Twelve-state model*

Figure D.4: *Model configuration example from [11]. Figures (b) to (d) show the probability density function (pdf) of the models. Each magenta dot represents the mean of a mixture component. With increasing number of components, data structure is captured in greater detail (b and c). However, a model with too many components starts picking up on local data structures (d) which is undesired for model generalizability.*

it becomes too dependent on the specific structure of the training data. When presented with new data samples the model is likely unable to generalize.

To prevent overfitting, generalisation capabilities are evaluated on a separate validation set, as suggested by Faller in [11]. Extending the method, the current work utilizes the PS, Equation (D.1), to trade-off the state selection between accurate and smooth estimations. Illustrating the state selection methods, Figure D.5 and Figure D.6 present flowcharts for the individual and average state selection methods, respectively.

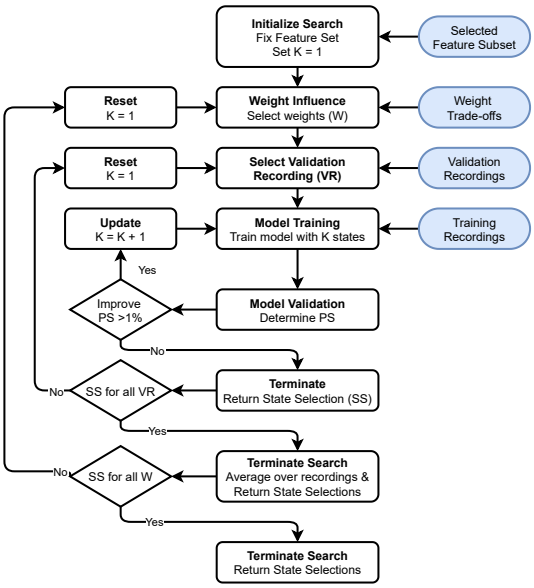


Figure D.5: *Flowchart illustrating the Individual State Selection method. The method is initialized with the fixed selected feature set and one hidden state. As the increase of the number of states is dependent on the model PS, first the weight trade-off is selected. Then for each validation recording individually, the number of states is increased as long as the model PS for that recording improves by more than 1%. When this criteria is not met, the number of hidden states that last improved the PS for that recording is returned. This is repeated for all recordings, after which the number of hidden states is averaged over all recordings before a new weight trade-off is selected. This process is repeated for all weight trade-offs.*

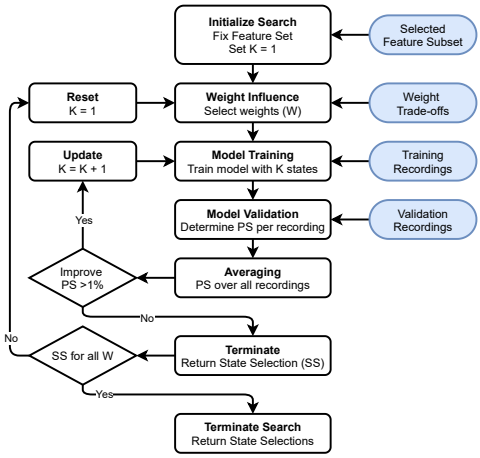


Figure D.6: *Flowchart illustrating the Average State Selection method. The method is initialized with the fixed selected feature set and one hidden state. As the increase of the number of states is dependent on the model PS, first the weight trade-off is selected. Then for all recordings simultaneously, the number of states is increased as long as the average model PS over all recordings improves by more than 1%. When this criteria is not met, the number of hidden states that last improved the PS is returned. This process is repeated for all weight trade-offs.*

D.5. Complete Parameter Selection Process

Figure D.7 illustrates the complete parameter selection process. Summarizing the process, first the proposed feature selection method performs a limited exhaustive and complete sequential forward search over the features in the training dataset. Subsequently, in the selection analysis, all models are validated and feature occurrences are counted to determine a selection strategy. A performance score sensitivity analysis will determine the final feature subset. Thereafter, the model configuration is determined by fixing the feature subset and varying the number of hidden states. The model with the best average performance on all recordings in the validation set, determines the number of hidden states. Finally, model performance for the selected parameters is analysed by training the final model and evaluate it on the test set.

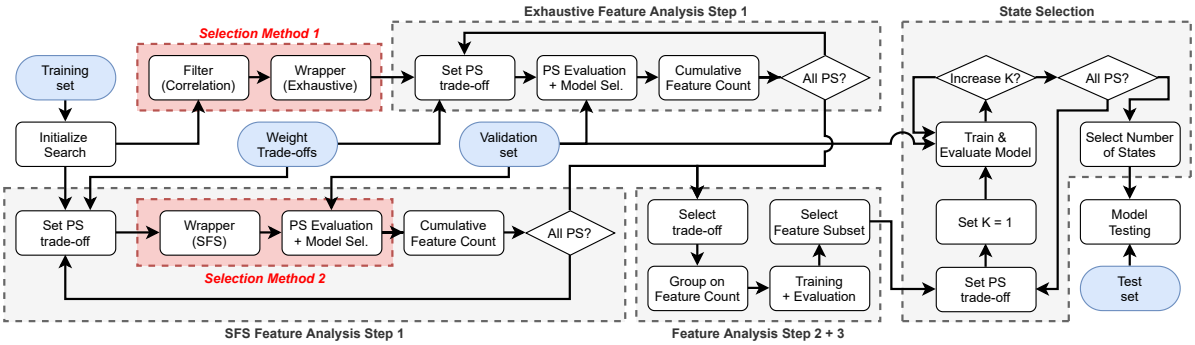


Figure D.7: *The complete parameter selection framework.*

E

Results & Discussion

The purpose of the parameter selection framework is to analyse what type of features and number of hidden states should be selected to achieve preferred steering-torque estimation behaviour. By selecting different weights for the Performance Score, a trade-off can be made to balance the model's estimation accuracy with the smoothness of the estimation.

This chapter provides supplementary results in Appendix E.1 to supporting the findings in the main body Chapter 2 and is structured in the same order. Furthermore, limitations of the work and recommendations for future work are discussed in Appendices E.2 and E.3, respectively. Lastly, Appendix E.4 explores the boundaries of the current presented model in a preliminary study in preparation for future work.

E.1. Supplementary Results

The aim of the driving simulator experiment was to collect sufficient naturalistic driving data for the training, validation and testing of the proposed HMM modelling approach. This section provides supplementary results to support the main findings. Each subsection corresponds to the subsections of Section VI. in Chapter 2.

E.1.1. Feature Correlation

Correlation coefficients per feature over all training recordings, presented visually in boxplots in Figure E.1, shows a clear distinction between stronger and weaker correlated features. Feature sorting is based on the average correlation coefficient, without taking into account the spread.

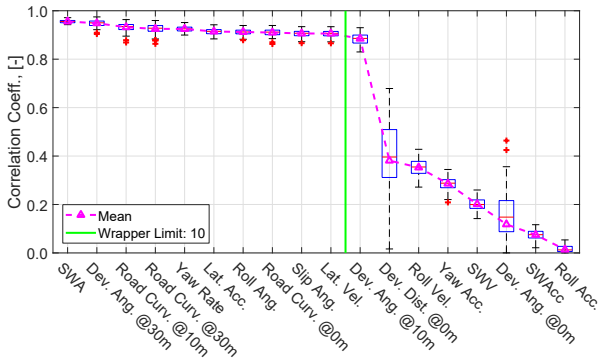


Figure E.1: *Feature Correlation coefficients per feature over all training recordings, sorted on their average.*

The average correlation coefficients of the disregarded features are summarized in Table E.1. Apart from the deviation angle @10m, the remaining features were disregarded due to weak correlation with the steering-torque.

Table E.1: *Correlation coefficients corresponding to the disregarded features.*

Disregarded Features	$\lvert r_s \rvert$	p
Deviation Angle @10m	0.8848	0
Deviation Distance @0m	0.3812	< 0.01
Roll Velocity	0.3529	0
Yaw Acceleration	0.2863	0
Steering Wheel Velocity	0.2016	0
Deviation Angle @0m	0.1180	0.01
Steering Wheel Acceleration	0.0748	0
Roll Acceleration	0.0138	0.20

E.1.2. Determining Feature Relevance

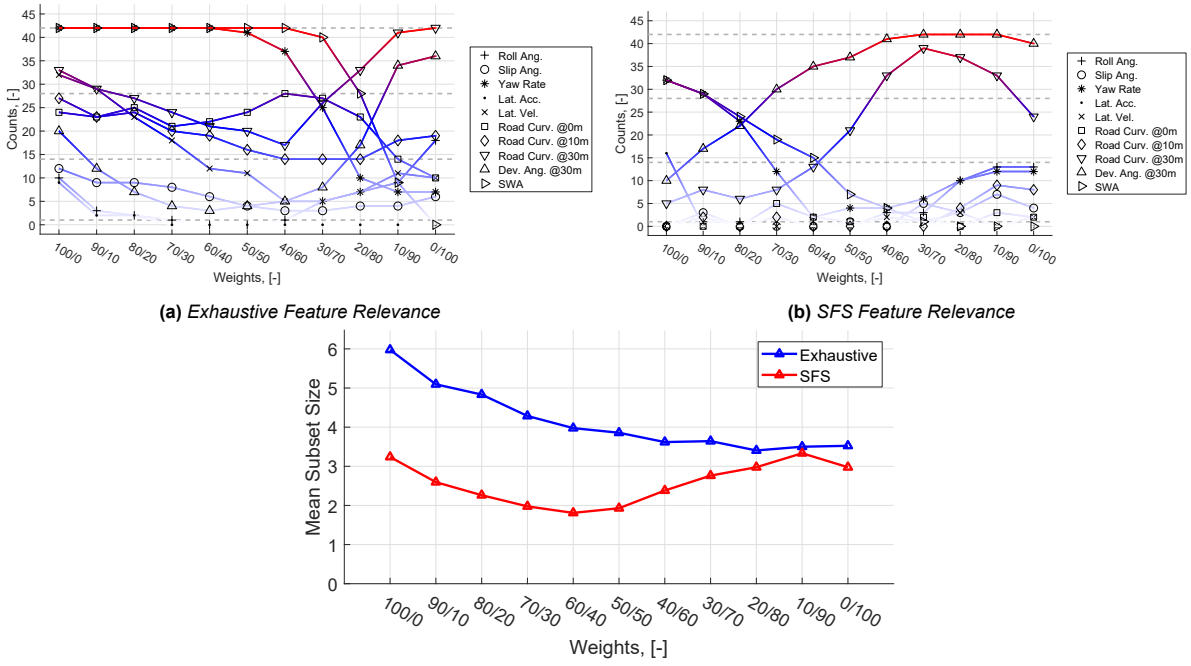
Feature relevance assumption: Initial investigation on feature relevance is performed by only focusing on model accuracy. The PS weights $\omega_{1,2}$ are set to a 100/0 trade-off and the model achieving the lowest PS for a particular validation recording is selected. The results in Table E.2 show that there exists a wide variety of feature combinations in the selected models and there exist no single best performing model. For example, while the selected model for recording 19 only contains three features, achieving an accuracy of 91%, the model for recording 6 requires all 10 candidate features while only achieving 86% accuracy. However, looking at the individual features it is observed that some features are selected more often than others. For example, the steering wheel angle (SWA, column 10) and yaw rate (Yaw Rate, column 3) are included in every model, while the lateral acceleration (Lat. Acc., column 4) and roll angle (Roll Ang., column 1) are selected only for respectively 9 and 10 out of 42 recordings. Therefore, obtaining feature relevance by counting the occurrences of features in the individual models is assumed to be a viable method to determine what features should be included in the final model.

Table E.2: *Results from preliminary feature occurrence. PS weights were 100/0, meaning focus was purely on accuracy. For each validation recording (rows) an X marks a feature occurring in the best performing model for that recording. The table shows that there exists a variety of models that perform better on specific validation recordings.*

Val. Rec.	Roll Ang.	Slip Ang.	Yaw Rate	Lat. Acc.	Lat. Vel.	Road Curv. @0m	Road Curv. @10m	Road Curv. @30m	Dev. Ang. @30m	SWA	Accuracy [%]
1		X	X			X	X	X	X	X	85.43
2			X		X	X	X	X		X	86.34
3		X	X			X	X	X		X	81.36
4		X	X			X	X	X		X	86.08
5			X					X	X	X	85.44
6	X	X	X	X	X	X	X	X	X	X	86.19
7			X		X			X	X	X	90.65
8		X	X					X	X	X	89.97
9			X		X		X	X	X	X	89.04
10	X	X	X	X	X	X	X	X	X	X	89.24
11			X		X			X	X	X	91.38
12	X	X	X	X	X	X	X	X	X	X	89.71
13			X		X	X		X		X	88.19
14			X		X	X		X		X	89.20
15			X		X			X	X	X	89.50
16			X		X	X	X	X		X	88.51
17			X			X	X			X	90.64
18			X	X	X		X			X	88.99
19			X		X					X	91.02
20			X		X	X	X			X	90.90
21		X	X			X	X	X	X	X	89.21
22			X		X		X	X		X	90.12
23			X		X	X	X	X		X	89.88
24			X	X	X					X	90.11
25			X		X	X	X	X		X	90.89
26			X		X	X	X	X		X	90.56
27			X		X		X	X	X	X	89.68
28	X	X	X	X	X	X	X	X	X	X	89.63
29			X		X	X	X	X		X	90.81
30			X		X	X	X	X		X	89.52
31	X		X				X	X	X	X	90.68
32	X		X			X		X	X	X	91.04
33			X		X			X	X	X	89.52
34	X	X	X	X	X			X	X	X	90.75
35			X		X		X	X	X	X	91.68
36	X	X	X	X	X	X	X	X	X	X	90.82
37			X		X					X	91.49
38			X		X	X				X	91.22
39	X		X			X	X	X		X	90.72
40			X		X		X			X	90.81
41			X		X					X	91.86
42	X	X	X	X	X	X	X	X	X	X	90.67
Mean	10	12	42	9	32	24	27	33	20	42	89.51

End of table

Average feature subset size per trade-off: As weight on estimation smoothness increases, overall feature relevance decreases up to balanced metric trade-off. From Figure E.2c it can be observed that the average feature subset size of the best selected models decreases. This means that the selected models for the validation recordings become simpler. This in turn decreases the risk of overfitting.



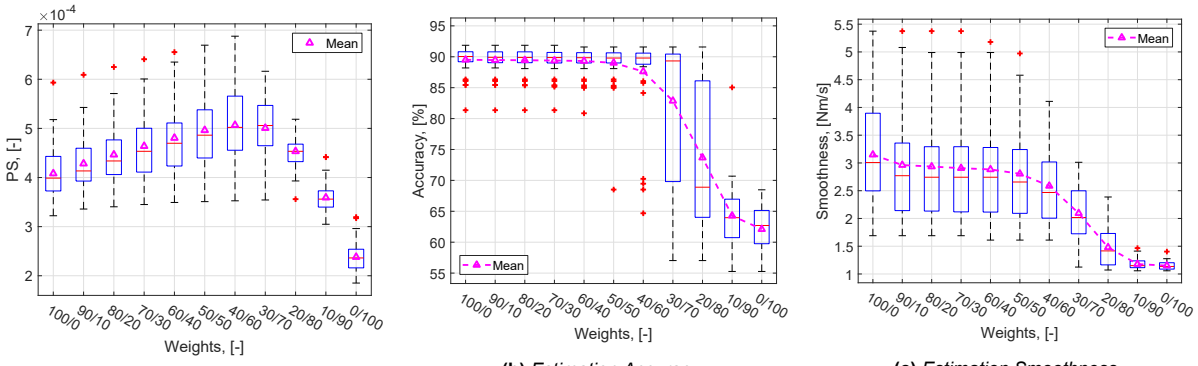
(a) *Exhaustive Feature Relevance*

(b) *SFS Feature Relevance*

(c) *Average feature subset size.*

Figure E.2: *Influence of varying PS weights $\omega_{1,2}$ on the average feature subset size of the best selected models. Smaller average sizes indicate overall simpler models.*

Objective balance between performance metrics: In order to determine a weight trade-off, the performance metrics are compared, instead of the performance score. This has the following reason. The performance score for each model is defined by two parameters, the selected weights and the metric (accuracy and smoothness) normalization factor over all models considered. However, even for the same model, the performance score of a particular model changes for different weight trade-offs. Meaning, if for example the same model is considered optimal for multiple weight trade-offs, their performance scores cannot directly be compared, such as in Figure E.3a. However, what the figure does show, is an indication of where the objective balance between the two performance metrics lies. Based on the figure this is around a 40/60 trade-off. In favor of increased model accuracy for all participants, the current work considers a 50/50 trade-off as this increases average model accuracy and thus deviates slightly from the objective balance.



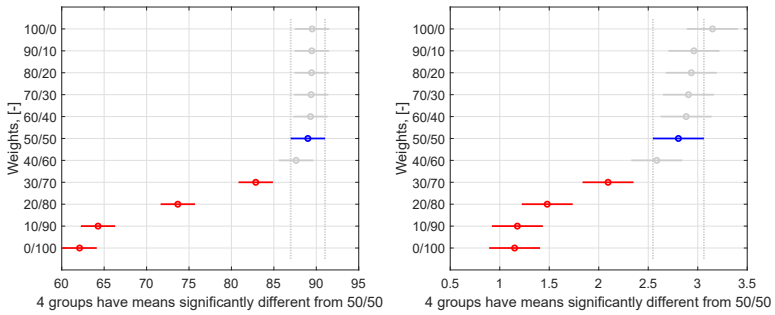
(a) *Performance Score*

(b) *Estimation Accuracy*

(c) *Estimation Smoothness*

Figure E.3: *Influence of the PS weights $\omega_{1,2}$ on the distribution of performance scores.*

Significance of average model performance metrics: The metric scores can be directly compared, as was done in Chapter 2. As the metric scores are close together for the trade-off range 100/0 to 40/60, a one-way ANOVA test was performed for both metrics to investigate if there is a significant difference between average metric scores for different trade-offs. The results are summarized in Figure E.4. For both metrics the null hypothesis is rejected (p > 0.05), meaning that there is no significant difference between the trade-offs, and thus a trade-off can be selected.

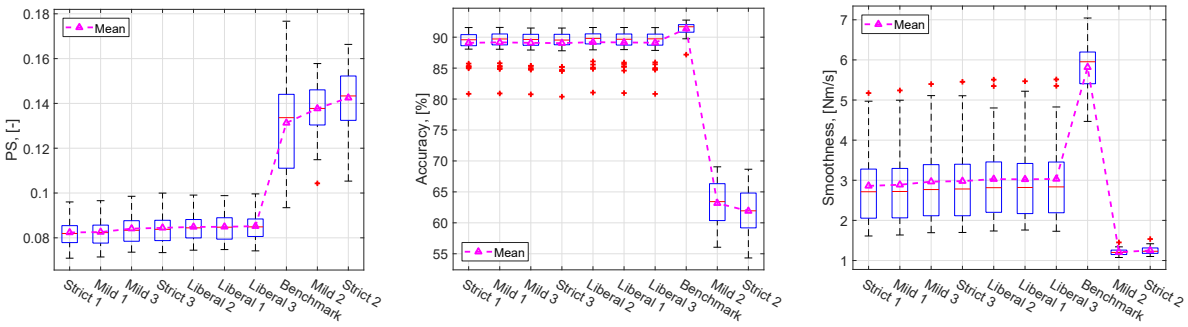


(a) *Estimation Accuracy* **(b)** *Estimation Smoothness*

Figure E.4: *The results from a one-way ANOVA test to investigate significant difference between average model performance metrics for feature relevance.*

E.1.3. Optimal Feature Subset

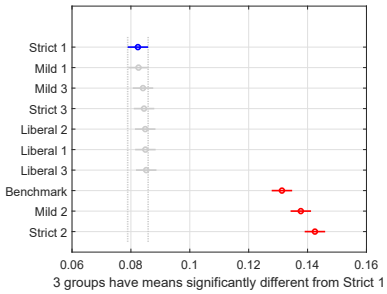
Performance Metrics of subset strategies: Figures E.5b and E.5c show the model accuracies and estimation smoothness scores corresponding to the subset strategies performance scores, Figure E.5a that was presented in Chapter 2. Improved results are obtained when subset selection is based solely on feature relevance of the exhaustive wrapper (strategies ”X1”) or where relevance of the SFS wrapper complements the exhaustive wrapper (strategies ”X3”), with the exception of ”Liberal 2”, as this subset contains many similar features. Furthermore, the figures show the superior accuracy of the baseline model, but also the worse estimation smoothness.



(a) *Performance Score* **(b)** *Estimation Accuracy* **(c)** *Estimation Smoothness*

Figure E.5: *Metric results of the models corresponding to the final subset selections, sorted on the average performance score.*

Significance of average strategy performance: As the performance scores for the selection strategies are close together for the first seven strategies, a one-way ANOVA test was performed to investigate if there is a significant difference between average performance scores for different strategies. The results are summarized in Figure E.4. The null hypothesis is rejected (p > 0.05), meaning that there is no significant difference between the strategies, and thus a strategy can be selected.

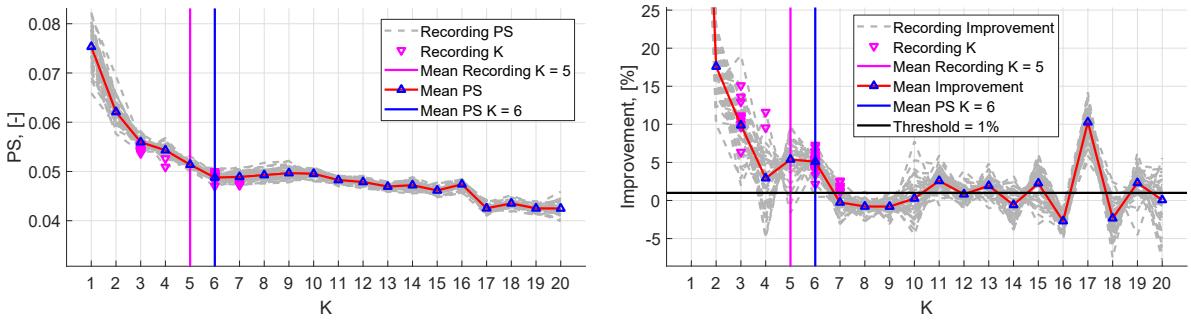


(a) *Performance Score*

Figure E.6: *The results from a one-way ANOVA test to investigate significant difference between average performance scores of several strategies.*

E.1.4. State Selection

Visualizing the state selection methods: State selection was performed with two methods. Figure E.7 visualizes both methods for a 50/50 metric trade-off. The figures show that the performance score keeps improving as K increases. However, with exponentially decreasing rate. As the average PS, indicated in red, does not improve more than 1% for $K = 7$, state selection is terminated. The last state that did improve ($K = 6$) is then considered as the optimal state.
Selection based on the average number of states per recording is shown with magenta inverted triangles. Observed is that some recordings do not prefer as many states as the ”average PS” selection suggests. Therefore, the average over the recordings provides a middle ground between all recordings and leads to a more conservative selection.

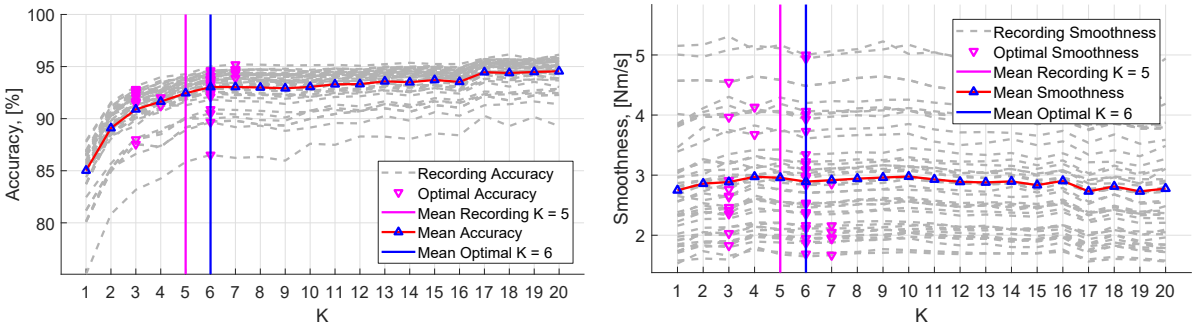


(a) *Performance Score* **(b)** *Performance Score Rate*

Figure E.7: *Score gradients for increasing K.*

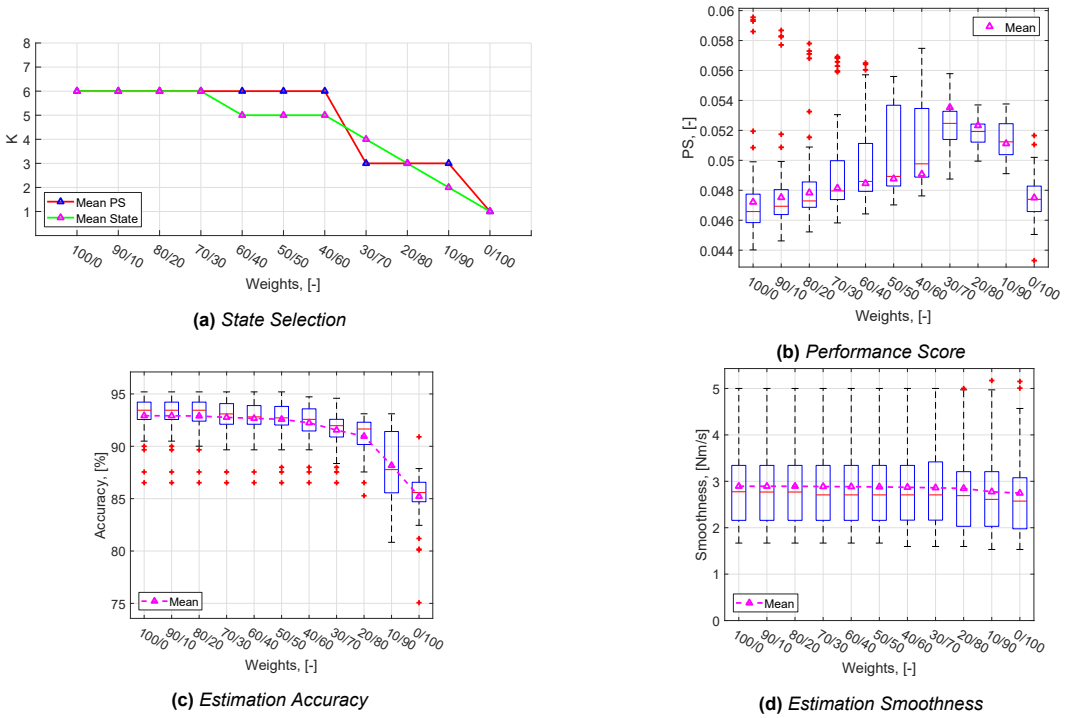
Corresponding performance metrics: Plotted in Figure E.8 are the performance metrics corresponding to the performance score of Figure E.7a. From the figure it is observed that the increasing model accuracy has the most influence on the performance score, as smoothness scores stay relatively constant over all validation recordings. The drop in performance improvement at K = 4 (Figure E.7b) is caused by a reduction in estimation smoothness (Figure E.8b). However, as model accuracy keeps increasing, the marginal decrease in smoothness is worth the increase in model states to K = 6 from a performance score perspective. Similar figures can be plotted for other metric trade-offs.

Objective balance between performance metrics: Regarding the comparison between metric trade-offs, the same reasoning holds as for feature relevance (see Appendix E.1.2). However, a similar figure can be plotted for the state selection, seen in Figure E.9b. It is observed that a 30/70 metric trade-off provides the best objective balance between model accuracy and estimation smoothness. Referring to Figure E.9a, this results in a selection of three model states. However, looking at the metrics individually, in Figures E.9c and E.9d, selecting fewer states significantly reduces estimation accuracy while only marginally improving the smoothness. Therefore, (similar to feature relevance) in favor of increased estimation accuracy, weight on smoothness is decreased and from Figure E.9a is concluded that more states are preferred.



(a) *Model Accuracy* **(b)** *Estimation Smoothness*

Figure E.8: *Gradient of performance metrics for increasing K.*



(a) *State Selection* **(b)** *Performance Score*

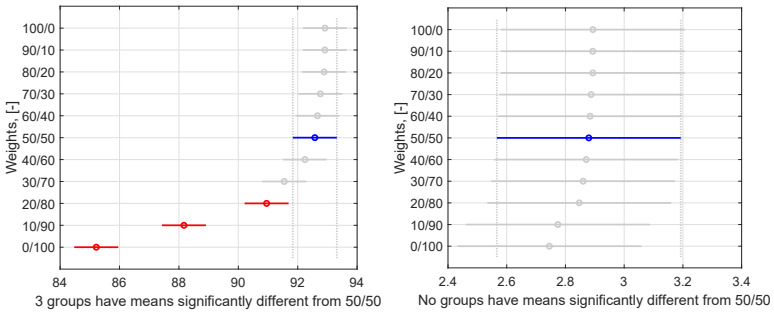
(c) *Estimation Accuracy* **(d)** *Estimation Smoothness*

Figure E.9: *Weight influence on the performance score of the optimal model states.*

Significance of average model metric scores: Similar to the feature relevance, the metric scores can be directly compared. As the accuracy scores are close together for the trade-off range 100/0 to 40/60 and all trade-offs for the smoothness, a one-way ANOVA test was performed for both metrics to investigate if there is a significant difference between average metric scores for different trade-offs. The results are summarized in Figure E.10. For both the accuracy in the mentioned range and the smoothness scores, the null hypothesis is rejected ($p > 0.05$), meaning that there is no significant difference between the trade-offs, and thus a trade-off can be selected.

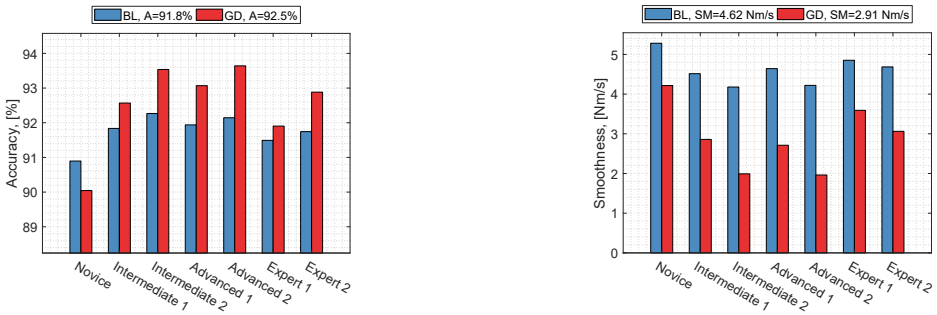
E.1.5. Model Testing

Personal behaviour versus skill level: Among the remaining participants, model performance appears to be dependent on the participant personal steering behaviour rather than skill level. For example, the generic driver model achieves the highest accuracy with the smoothest estimation for both participants ”Intermediate 2” (A=93.5%, SM=1.99Nm/s) and ”Advanced 2” (A=93.6%, SM=1.96Nm/s). Similar observations are seen for the participants of similar skill levels, i.e. ”Intermediate 1” (A=92.5%, SM=2.86Nm/s) and ”Advanced 1” (A=93.1%, SM=2.71Nm/s).



(a) *Estimation Accuracy* **(b)** *Estimation Smoothness*

Figure E.10: *The results from a one-way ANOVA test to investigate significant difference between average model performance metrics on state selection.*



(a) *Model accuracy* **(b)** *Estimation smoothness*

Figure E.11: *Baseline Performance Comparison.*

As was stated in Chapter 2, the difference in model performance between participants was mainly due to the behaviour of the participant itself. If the behaviour is more abrupt, the model also has to be more abrupt, ”negatively” impacting performance metrics. This is best observed in Figure E.12, comparing the steering-torques of the best (”Advanced 1”) and worst (”Novice”) modelled participants, negotiating the same corner. As the ”Novice” participant makes more abrupt steering corrections compared to the ”Advanced” participant, the generic driver model mimics this behaviour, inherently decreasing the smoothness score. Furthermore, this also influences the overall accuracy for this participant as the model sometimes overshoots (i.e. around 275s).

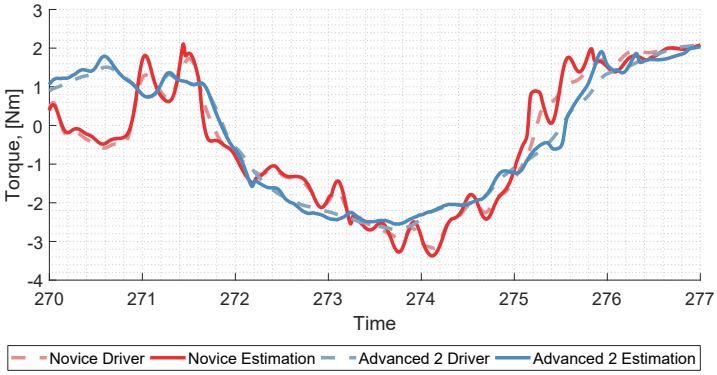


Figure E.12: *Comparing the estimated steering-torque for the novice and an advanced participant.*

Model Reproducibility: in the E-step of the EM-algorithm, see Appendix A.1.2, the parameters of the prior probabilities π_k are initialized at random (constrained between zero and one). The parameters of the Gaussian normal distributions (mean and covariance) are initialized with the K-means algorithm ([5], Chapter 9). The K-means algorithm itself initializes the cluster means by randomly selecting K

data points. To investigate the robustness of the approach to the random initialization process, the training algorithm is repeated ten times and results are summarized in Table E.3. From the table it is concluded that the 95% confidence interval (CI) for a sample size of ten models is sufficiently small to assume that the correct model parameters are found consistently.

Table E.3: *Model reproducibility results for a model sample size of ten.*

(a) *Estimation Accuracy, %*

Participant	Mean	95% CI		Min.	Max.
Novice	89.7	89.3	90.1	88.5	90.1
Intermediate 1	92.5	92.3	92.6	92.0	92.6
Intermediate 2	93.4	93.3	93.6	93.0	93.6
Advanced 1	92.9	92.6	93.1	92.1	93.1
Advanced 2	93.6	93.4	93.7	93.2	93.6
Expert 1	91.8	91.6	91.9	91.3	92.0
Expert 2	92.8	92.7	92.9	92.4	92.9

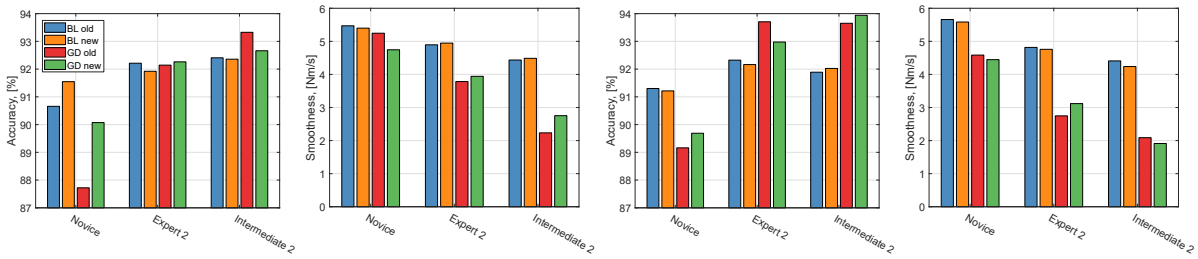
(b) *Estimation Smoothness, Nm/s*

Participant	Mean	95% CI		Min.	Max.
Novice	4.22	4.21	4.24	4.20	4.26
Intermediate 1	2.86	2.86	2.86	2.85	2.87
Intermediate 2	2.01	1.99	2.02	1.99	2.06
Advanced 1	2.73	2.71	2.76	2.71	2.81
Advanced 2	1.98	1.96	2.00	1.96	2.03
Expert 1	3.60	3.59	3.61	3.58	3.64
Expert 2	3.07	3.06	3.07	3.06	3.09

E.2. Limitations

Simulator Immersion: Unfortunately, due to Covid-19 measures taken by the Belgian government, availability of the TME simulator was limited during the span of the research. This has resulted in insufficient time to explore simulator performance for the presented scenario lengths. A cascading effect of this was that the designed scenarios were functional but had reduced visual immersion representing realistic highways. Feedback from participants indicated being distracted during the experiment due to a lack of visual cues. This was assumed to be advantageous from a behaviour perspective as it is thought of that driver steering variability would increase due to inattention. However, this does limit reflection of true participant steering behaviour. Therefore, for future experiments, more time needs to be allocated to make scenarios of this length more visually engaging. Adding land- and road attributes (e.g. overhead signs, highway exits, lane barriers, distant housing, etc.) will provide the necessary visual cues for increased participant immersion.

Hardware Upgrades: During pilot experiments, large oscillating steering-torques were observed in situations where participants drove in a straight line. This in turn got picked up by the HMM, resulting in a false reflection of participant steering behaviour. The opted short-term solution involved tuning steering model parameters such to minimize this behaviour while retaining natural steering feel, albeit with increased steering effort. These settings were validated by a TME expert technician (representing steering feel of older generation vehicles), and experiments were started due to limited simulator availability (also Covid-19 related). The long-term solution was found during the experiment and involved replacing parts of the steering-column. It was decided that this would be fixed after experiments had finished to preserve uniform experimental conditions. Once replaced, additional experiments on both training and validation data were performed with several participants. Steering model parameters were not changed. From Figure E.13, it is observed that the new steering column dynamics do not have significant impact on model performance. However, further influence of different steering model parameters on model performance has not been investigated.



(a) *Training Data Accuracy* **(b)** *Training Data Smoothness* **(c)** *Validation Data Accuracy* **(d)** *Validation Data Smoothness*

Figure E.13: *Baseline and Generic Driver Model Performance before and after steering column replacement.*

Feature Selection Methods: The scarcity of literature on feature selection for HMM [3] and the fact that the aim of this work was not to find the best method, has led to the adoption of simpler methods as performed in [11]. The presented results in this work show that the hybrid (filter and exhaustive wrapper) and sequential forward selection methods are also effective for parameterizing HMM-based driver steering-torque models. However, especially the selected wrapper methods in combination with the size of the presented data-set and the analysis of different weight trade-offs, makes the current method computationally expensive. As the current work was performed offline, this did not have any immediate impact on the results. However, this can lead to limitations for online scenarios. This motivates the potential for exploring and comparing efficient selection methods for HMM-based regression problems.

E.3. Recommendations for Future Work

Steering-Torque Predictions: While the presented results show the potential of the proposed approach to model and estimate driver steering-torque behaviour, parameter choices were based on objective metrics without taking into account driver steering feel. In order to validate the selected parameters *subjectively*, it is proposed to implement the HMM driver model in a driver steering assistance system. As the current work focused on lane-keeping behaviour, a Lane Keeping Assist controller similar to the work [22] by S. Lefevre et al. is a potential application. Steering-torque can be predicted recursively over a prediction horizon by alternating between a bicycle model (to predict future vehicle heading) and steering-torque estimation (providing bicycle model input). The obtained prediction will serve as a reference signal to a non-linear model predictive (NMPC) controller. Once implemented, different weight trade-offs can be compared in a subjective simulator experiment. Furthermore, this provides the opportunity to compare subjective feel between the steering-angle based HMM of [22] and the proposed steering-torque based HMM from the current work.

Data Quality: Steering wheel related features (e.g. steering-torque, position, velocity, acceleration) were measured with actual sensors in the steering column. Since in the current work model training was performed offline, these features could be pre-processed with a zero-phase low pass filter to filter out measurement noise. The remaining features were recorded directly through CarMaker, resulting in a high quality dataset. This benefited model training and have shown to perform well for offline estimation. In order to utilize the proposed modelling approach for real-world applications, additional pre-processing steps need to be applied before model training. On the one hand due to noisier real-world measurements. On the other hand that some features need to be extracted first, e.g. road preview features need to be extracted from a front facing cameras. In that case, also model robustness to noise can be investigated.

Personalized Driver Models: an interesting results was that model performance appeared to be related to personal driver behaviour. It was shown throughout intermediate results that there was no single best model to capture the behavioural variability of all participants equally. This motivates the concept of personalized driver steering-torque models. Possible scenarios include training models directly for one specific driver, or, adapting the generic driver model to the driver with online-learning methods. The latter has been investigated in [13] by J. Geukes, using adaptable HMMs for predicting lane change maneuvers and provides a valuable starting point for adaptable steering-torque models.

Model Potential: The longitudinal speed of the vehicle was fixed with a cruise controller to 100km/h in an attempt to avoid the influence of variable speed on driver steering behaviour. Adapting the model to different scenarios, such as variable speed or including traffic, is easily achievable by including the necessary features (e.g. longitudinal speed or distance to surrounding vehicles, respectively) in the analysis. It would be interesting to see how much these features contribute to steering-torque estimation. Furthermore, as was proposed by A. Pentland and A. Liu in [31], the hidden states of the model can be organized hierarchically to represent not only lower- (e.g. following a target lane) but also higher-level behaviour, such as lane-changing or overtaking. By modelling each maneuver with a different HMM (each with hidden states representing the lower-level behaviour), and sequencing these maneuver models with a Markov Chain, one will be able to obtain more complex driver steering-torque behaviour models. To the best of my knowledge, these more complex behaviour models have not yet been investigated in combination with GMR estimation.

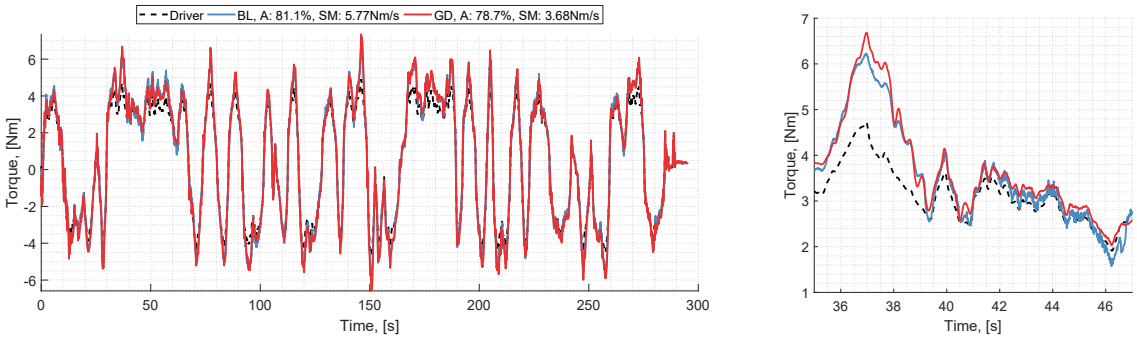
E.4. Preliminary Results

Even though the sections used for model validation and testing were generated randomly, fundamentally they consist of the same elements as the training set. Just in a different order. To investigate model generalisation boundaries, a preliminary study with alternative scenarios was performed. Evaluated scenarios include:

- **Sharper Corners:** for increased steering-effort and higher lateral accelerations.
- **Wider Lanes:** for increased steering variability.
- **Lane Changes:** for deviation from the target lane.
- **Evasive Maneuvers:** for increased and rapid changes in driver steering-torque.

For each scenario, performance of the Generic Driver (GD) Model is compared to the Baseline (BL) Model. Furthermore, vehicle speed remained fixed to 100km/h.

Sharper Corners: Three new scenarios of five minutes were generated similar to the experiment scenarios but with corner radii sampled between 175 and 300 meters with 25 meter intervals. All scenarios were driven by one participant.
Result: The results show a reduced performance for both models. The BL model averaged 80.0% accuracy with a smoothness of 6.25Nm/s. The GD model averaged 78.2% accuracy with a smoothness of 3.81Nm/s. This again validates that the GD model achieves similar performance with around 39% smoother estimations, which can be clearly observed from Figure E.14b. The reduction in performance of both models can be explained in situations with increased steering effort (larger torques), see Figure E.14a. Zooming in on one of such situations, Figure E.14b show the mismatch between driver torque and model estimation at 37s. Due to the Gaussian distribution, the models are able to make an estimation. However, the worse performance is thought to originate from that these situations occur at the tails of the emission distributions, where the likelihood is minimal, and thus estimation is uncertain. Therefore, both models generalize bad to such situations and require additional training.

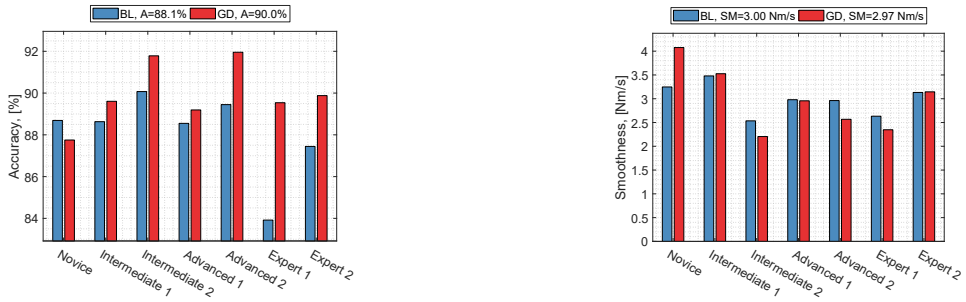


(a) *Global Estimation* **(b)** *Zoomed*

Figure E.14: *Baseline and Generic Driver Model Performance on randomly generated sections with sharper corners. Observed again is the reduced estimation noise and worse performance for larger torques.*

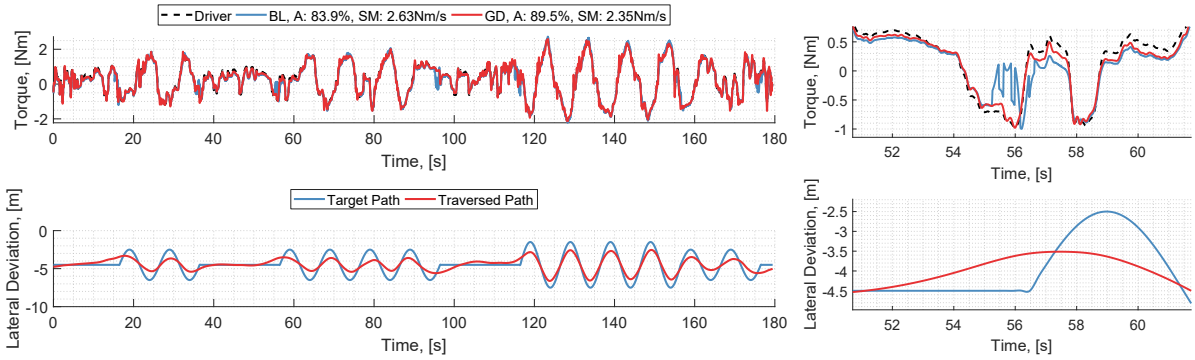
Wider Lanes: A new 5-meter wide single lane scenario of 5km long was designed. The road consists of four straight segments, alternated with four sinusoidal segments of varying amplitudes, see Figure E.16b. All participants performed the experiment and were only instructed to follow their preferred driving style without clarifying the lane-following strategy.
Result: Compared to the test results in Figure E.11, a wider road does not significantly impact model performance, see Figure E.15. Both models achieve similar results for all participants, except the BL model for participant ”Expert 1”. From the recording, shown in Figure E.16, it is observed that this is caused by poor estimations around straight and sinusoidal segment transitions, which are discontinuous. The poor estimations are thought to originate from that both the inclusion of road preview features

in the BL model and the significant corner cutting behaviour of the participant (therefore deviating from training data), the BL model cannot handle these discontinuities. Since the GD model does not contain road preview features, it does not suffer the same problem.



(a) *Wider Lane Estimation Accuracy* **(b)** *Wider Lane Estimation Smoothness*

Figure E.15: *Baseline and Generic Driver Model Performance metrics on a scenario with increased lane-width.*



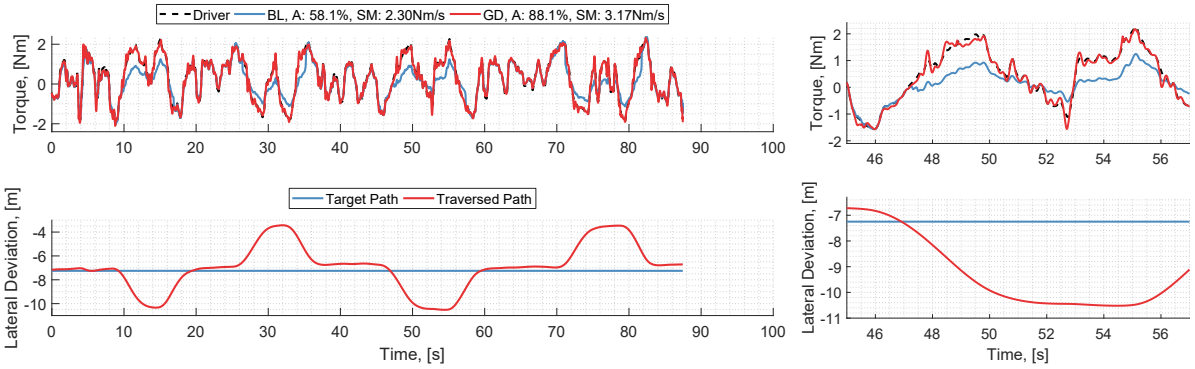
(a) *Global Estimation (top) and vehicle path data (bottom)* **(b)** *Zoomed*

Figure E.16: *Baseline and Generic Driver Model estimation performance on a scenario with increased lane-width.*

Lane Changes (LC): Two new simple scenarios were designed: A straight-line scenario and a curved scenario with one left- and one right-hand turn. During each scenario, two lane changes were performed in both directions (left and right). Each scenario was repeated three times and driven by one participant.
Result Sraight-line LC: In a straight line, the BL model seems to suffer significantly in accuracy during lane-changes compared to the GD model, see Figure E.17. BL model performance averages 63.1% accuracy with a smoothness of 2.53 Nm/s over three runs. The GD model averages 87.3% accuracy with a smoothness of 3.56 Nm/s. Similar to corner cutting of a wide lane, this is thought to originate from the fact that the BL model has road preview features, which are not correctly trained for large deviations from the center lane. To solve this issue, the target lane should switch during a lane change. This was not investigated in this study. However, this does not seem a problem for the GD model, again due to the fact that the GD model does not include road preview features.

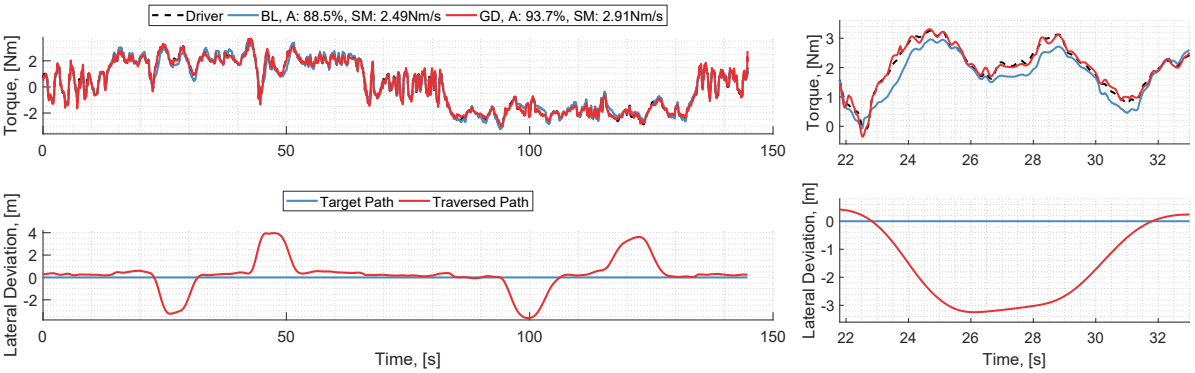
Result Curved LC: During both curvature negotiation and lane changing, the BL model does not seem to suffer as much in accuracy compared to the straight line scenario, see Figure E.18. Furthermore, also the GD model performance is improved. BL model performance averages 87.4% accuracy with a smoothness of 2.50 Nm/s over three runs. The GD model averages 91.8% accuracy with a smoothness of 3.19 Nm/s. The improved performance is thought of to be explained by the fact that the lateral deviation of a lane change during curvature negotiation can be compared to cutting the corner.

Evasive Maneuvers: The maneuvers were simulated by rapidly turning the steering wheel at a random moment during a straight-line driving scenario. Both left and right maneuvers were repeated two times and driven by one participant.
Result: Both models were unable to properly estimate the driver steering-torque, see Figure E.19.



(a) *Global Estimation (top) and vehicle path data (bottom)* **(b)** *Zoomed*

Figure E.17: *Baseline and Generic Driver Model Performance during a straight-line lane change scenario.*



(a) *Global Estimation (top) and vehicle path data (bottom)* **(b)** *Zoomed*

Figure E.18: *Baseline and Generic Driver Model Performance during a curved lane change scenario.*

Similar to the previous scenarios, this is thought of to be explained by that the increased steering-torque has a low likelihood of occurring for the current trained Gaussian distributions, and thus the model not being able to handle the situation. As generating training data for evasive maneuvers is not very practical, steering-torque should be estimated by evasive controllers instead.

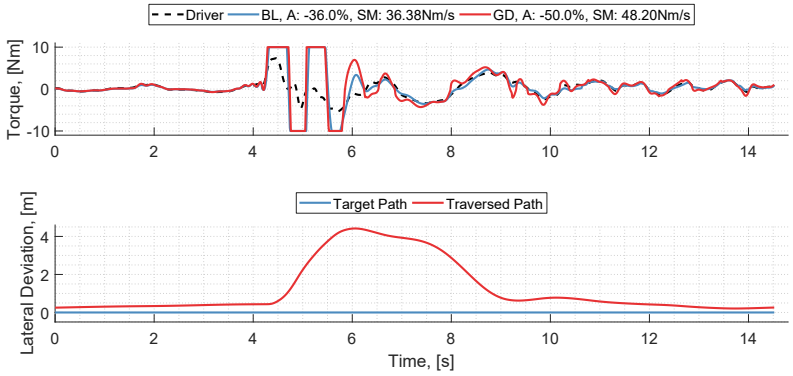


Figure E.19: *Baseline and Generic Driver Model Performance during an evasive maneuver. Global Estimation (top) and vehicle path data (bottom).*

Conclusion: For the majority of the presented tests, both the BL model as well as the GD model were still able to accurately estimate driver steering-torque. However, model limitations were discovered when newly presented data points lie in the tails of the state emission distributions. This appeared to be input-feature (e.g. road preview), and also output-feature dependable (steering-torque). It is thought of that the low likelihood of the data-point is causing the increase in estimation error.

Additionally to the presented test scenarios, it would be interesting to be able to train and validate HMM driver models with more realistic, real-world scenarios. Towards the end of this thesis, an interesting workflow for virtual road generation in CarMaker, based on real-world geographic data, was found in Chapter 2.3.2 of the work [30] by M. Olofsson and J. Pettersson and provides a valuable starting point.

Bibliography

[1] D.A Abbink, M. Mulder, and E.R. Boer. “Haptic shared control: smoothly shifting control authority?” In: *Cognition, Technology & Work* 14.1 (2012), pp. 19–28.

[2] D.A. Abbink and M. Mulder. “Neuromuscular Analysis as a Guideline in designing Shared Control”. In: *Advances in Haptics*. Ed. by M.H. Zadeh. Rijeka: IntechOpen, 2010. Chap. 27. doi: https://doi.org/https://doi.org/10.5772/8696.

[3] S. Adams and P.A. Beling. “A survey of feature selection methods for Gaussian mixture models and hidden Markov models”. In: *Artificial Intelligence Review* 52.3 (2019), pp. 1739–1779.

[4] R.E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961. isbn: 9781400874668. doi: https://doi.org/10.1515/9781400874668.

[5] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York: Springer-Verlag, 2006. isbn: 978-0-387-31073-2.

[6] S. Calinon et al. “Learning and Reproduction of Gestures by Imitation”. In: *IEEE Robotics Automation Magazine* 17.2 (2010), pp. 44–54. doi: https://doi.org/10.1109/MRA.2010.936947.

[7] G. Chandrashekar and F. Sahin. “A survey on feature selection methods”. In: *Computers & Electrical Engineering* 40.1 (2014). 40th-year commemorative issue, pp. 16–28. issn: 0045-7906. doi: https://doi.org/https://doi.org/10.1016/j.compeleceng.2013.11.024. url: https://www.sciencedirect.com/science/article/pii/S0045790613003066.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22. doi: https://doi.org/https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

[9] M.L. Eaton. “The normal distribution on a vector space”. In: *Multivariate Statistics*. Institute of Mathematical Statistics, 2007, pp. 103–131.

[10] *EU Road Safety Policy Framework 2021-2030 - Next steps towards ”Vision Zero”*. Aug. 2020. url: https://ec.europa.eu/transport/road_safety/what-we-do_en.

[11] F. Faller. “Utilization of a hidden-markov-model for the prediction of lane change maneuvers”. PhD thesis. Master’s thesis, TU Darmstadt, Knowledge Engineering Group, 2016.

[12] M. Flad et al. “Individual Driver Modeling via Optimal Selection of Steering Primitives”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 6276–6282. issn: 1474-6670. doi: https://doi.org/10.3182/20140824-6-ZA-1003.00223.

[13] J. Geukes. “Personalized Driving Assistance to Predict Lane Change Manoeuvres”. MA thesis. Technische Universität Darmstadt, 2016. url: http://www.ke.tu-darmstadt.de/m/lehre/arbeiten/master/2016/Geukes_Jan.pdf.

[14] Z. Ghahramani and M.I. Jordan. “Supervised learning from incomplete data via an EM approach”. In: *Advances in neural information processing systems*. 1994, pp. 120–127.

[15] M. Hasenjäger and H. Wersing. “Personalization in advanced driver assistance systems and autonomous vehicles: A review”. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. 2017, pp. 1–7. doi: https://doi.org/10.1109/ITSC.2017.8317803.

[16] Z. Hu et al. “Hybrid filter–wrapper feature selection for short-term load forecasting”. In: *Engineering Applications of Artificial Intelligence* 40 (2015), pp. 17–27. issn: 0952-1976. doi: https://doi.org/10.1016/j.engappai.2014.12.014. url: https://www.sciencedirect.com/science/article/pii/S0952197614003066.

[17] A. Jović, K. Brkić, and N. Bogunović. “A review of feature selection methods with applications”. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015, pp. 1200–1205. doi: https://doi.org/10.1109/MIPRO.2015.7160458.

[18] S. Jugade. “Shared control authority between human and autonomous driving system for intelligent vehicles”. Theses. Université de Technologie de Compiègne, Sept. 2019. url: https://tel.archives-ouvertes.fr/tel-02497365.

[19] M. Karagiannopoulos et al. “Feature selection for regression problems”. In: *Proceedings of the 8th Hellenic European Research on Computer Mathematics & its Applications, Athens, Greece* 2022 (2007).

[20] S. Kolekar, W. Mugge, and D. Abbink. “Modeling Intradriver Steering Variability Based on Sensorimotor Control Theories”. In: *IEEE Transactions on Human-Machine Systems* 48.3 (2018), pp. 291–303. doi: https://doi.org/10.1109/THMS.2018.2812620.

[21] A.M.R. Lazcano et al. “MPC-based Haptic Shared Steering System: A Driver Modelling Approach for Symbiotic Driving”. In: *IEEE/ASME Transactions on Mechatronics* (2021), pp. 1–1. doi: https://doi.org/10.1109/TMECH.2021.3063902.

[22] S. Lefèvre et al. “Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control”. In: *12th International Symposium on Advanced Vehicle Control*. Tokyo, Japan, 2014. url: https://hal.inria.fr/hal-01104458.

[23] J. Li et al. “Feature Selection: A Data Perspective”. In: *ACM Comput. Surv.* 50.6 (Dec. 2017). issn: 0360-0300. doi: https://doi.org/10.1145/3136625.

[24] C.C. Macadam. “Understanding and Modeling the Human Driver”. In: *Vehicle System Dynamics* 40.1-3 (2003), pp. 101–134. doi: https://doi.org/10.1076/vesd.40.1.101.15875.

[25] F. Mars and P. Chevrel. “Modelling human control of steering for the design of advanced driver assistance systems”. In: *Annual Reviews in Control* 44 (2017), pp. 292–302. issn: 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2017.09.011.

[26] M.H. Martens and G.D. Jenssen. “Behavioral Adaptation and Acceptance”. In: *Handbook of Intelligent Vehicles*. Ed. by A. Eskandarian. London: Springer London, 2012, pp. 117–138. isbn: 978-0-85729-085-4. doi: https://doi.org/10.1007/978-0-85729-085-4_6.

[27] D. Mircea et al. “Torque control for more realistic hand-wheel haptics in a driving simulator”. In: *Proceedings of the Driving Simulation Conference 2019 Europe VR*. Ed. by A. Kemeny et al. Driving Simulation Association. Strasbourg, France, Sept. 4, 2019, pp. 35–42. isbn: 978-2-85782-749-8.

[28] K.P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[29] T. Niu and D. Cole. “A Model of Driver Steering Control Incorporating Steering Torque Feedback and State Estimation”. In: (2020).

[30] M. Olofsson and J. Pettersson. “Parameterization and validation of road and driver behavior models for carmaker simulations and transmission HIL-rig”. MA thesis. 2015.

[31] A. Pentland and A. Liu. “Modeling and Prediction of Human Behavior”. In: *Neural Computation* 11.1 (Jan. 1999), pp. 229–242. issn: 0899-7667. doi: https://doi.org/10.1162/089976699300016890.

[32] L.R. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286. doi: https://doi.org/10.1109/5.18626.

[33] P. Ray, S.S. Reddy, and T. Banerjee. “Various dimension reduction techniques for high dimensional data analysis: a review”. In: *Artificial Intelligence Review* 54.5 (June 2021), pp. 3473–3515. issn: 1573-7462. doi: https://doi.org/10.1007/s10462-020-09928-0.

[34] *Regulation (EU) 2019/2144 of the European Parliament and of the Council*. Nov. 2019. url: https://ec.europa.eu/growth/sectors/automotive/safety_en.

[35] L. Saleh et al. “Human-like cybernetic driver model for lane keeping”. In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 4368–4373. issn: 1474-6670. doi: https://doi.org/10.3182/20110828-6-IT-1002.02349.

[36] J. Sjöberg et al. “Driver models to increase the potential of automotive active safety functions”. In: *2010 18th European Signal Processing Conference*. 2010, pp. 204–208.

[37] Y. Tian et al. “Latent Gaussian Mixture Regression for Human Pose Estimation”. In: *Computer Vision – ACCV 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 679–690. isbn: 978-3-642-19318-7.

[38] R. Zebari et al. “A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction”. In: *Journal of Applied Science and Technology Trends* 1.2 (2020), pp. 56–70.