Take-Over Time Modeling and Prediction for Conditional Driving Automation

Steven Hwang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Linda Ng Boyle, Chair

Ashis Banerjee, Chair

Ji-Eun Kim

Program Authorized to Offer Degree:
Industrial & Systems Engineering

University of Washington

Abstract

Take-Over Time Modeling and Prediction for Conditional Driving Automation

Steven Hwang

Co-Chairs of the Supervisory Committee:
Professor Linda Ng Boyle
Department of Industrial & Systems Engineering and Civil & Environmental Engineering

Professor Ashis Banerjee
Department of Industrial & Systems Engineering and Mechanical Engineering

Autonomous vehicles are designed to enhance the overall driver safety by taking the driver out of the loop. However, the autonomous vehicles that are currently available on the market still require that the driver is available to take-over control in case the automation fails (ex. Tesla AutoPilot). If the driver is not able to take-over in time and resume control of the vehicle, unsafe consequences, such as crashes, could occur. Extensive research has been done in modeling the driver behavior around these take-over events to mitigate the crash risk. However, these models use complex or unrealistic data sources (eye gaze, external environment, heart rate, video data, etc.) and are unable to handle time series data for updated predictions as new data is acquired.

The main focus of this research is to develop a modeling framework for take-over events in conditional driving automation with the focus on predicting the remaining take-over time. This research also explores the impact of individual driver differences on model performance through the use of online-learning. A hidden semi-Markov model (HSMM) was proposed for modeling the take-over time and the modeling framework was assessed on data collected from a driving simulator study. The proposed model is able to accurately predict the remaining take-over time, capture the uncertainty in the model prediction, and handle variable

length time series data when compared to state-of-the-art regression prediction models. By developing a framework to model the take-over time of drivers, autonomous vehicle manufacturers can mitigate the risk of vehicle control handovers as the technology continues to be developed and changed over time.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

TAKE-OVER REQUEST: a system warning given when the automation needs to hand control back to the driver

TAKE-OVER TIME: the time between the take-over request and when the driver takes over control of the vehicle

TIME BUDGET: the time between the take-over request and automation system limit

BRAKING INTERVENTION: the time when the standardized brake pedal travels greater than 10%

STEERING INTERVENTION: the time when the steering wheel angle is moved more than 2°

ACCELERATION INTERVENTION: the time when the standardized accelerator pedal travels greater than 10%

DYNAMIC DRIVING TASK: includes the operational (steering, braking, accelerating, monitoring the vehicle and roadway) and tactical (responding to events, determining when to change lanes, turn, use signals, etc.) aspects of driving

AUTOMATED DRIVING SYSTEMS: is defined by the SAE to be driving automation systems that can perform the entire dynamic driving task, regardless of the limiting conditions. This refers to the SAE levels 3-5

ADVANCED DRIVER-ASSISTANCE SYSTEMS: are electronic systems that aid a user while driving, for example the Tesla Autopilot

Abbreviations

HSMM	**H**idden-**S**emi **M**arkov **M**odel
SAE	**S**ociety of **A**utomotive **E**ngineers

TOR	**T**ake-**O**ver **R**equest
TOT	**T**ake-**O**ver **T**ime
TB	**T**ime **B**udget
ADS	**A**utomated **D**riving **S**ystems
DDT	**D**ynamic **D**riving **T**ask
ACC	**A**daptive **C**ruise **C**ontrol
ADAS	**A**dvanced **D**river-**A**ssistance **S**ystems
DTW	**D**ynamic **T**ime **W**arping
DBA	**D**TW **B**arycenter **A**veraging
DBSCAN	**D**ensity-**B**ased Spatial **C**lustering of **A**pplications with **N**oise
EM	**E**xpectation-**M**aximization
MAE	**M**ean **A**bsolute **E**rror

ACKNOWLEDGMENTS

To my adviser Dr. Linda Ng Boyle, I wanted to express my sincere gratitude for all the opportunities you have given me to grow and for fostering a research environment that does not only focus on academics, but also on developing relationships with your students that go beyond our time at the University of Washington.

To my adviser Dr. Ashis Banerjee, I am grateful for the mentorship you have provided me throughout the years. I always knew that if I needed advice or guidance, whether it be about academics or life in general, you would be willing to take the time out of your day to sit down and talk with me.

To each member of my committe, I appreciate all the support and for taking the time to be on my committee.

To all my fellow peers and friends in the Human Factors and Statistical Modeling lab: Fiete Krutein, Tianshu Feng, Jundi Liu, Huizhong Guo, Haena Kim, Ning Li, Erika Miller, and Xingwei Wu. I wanted to thank everyone for the fun memories over the years and all the help you have provided. I wish each and everyone of you all the best.

Lastly, I wanted to thank my mom, dad, brother, and sister-in-law for being there for me in case I ever needed to talk to someone.

DEDICATION

to my dear wife, Renee

Chapter 1

INTRODUCTION

In the United States approximately 12 million vehicles were produced in 2015, with the total number of registered vehicles on the road exceeding 263 million [10]. When compared to 2014, this showed an increase in the number of vehicles sold by approximately 4%, with a total increase in the number of registered vehicles of 1.2% . The increasing trend in the number of vehicles on the road comes with many problems, such as traffic jams and an increasing number of accidents. In the United States alone, there were more than 7 million accidents in 2016, which is an increase of about 20% when compared to 2014 [50]. To address the issues caused by the increasing number of vehicles on the road, researchers and car companies have put their hopes in *Advanced Driver-Assistance Systems* (ADAS).

The most common cause of vehicular accidents and traffic jams are decisions and mistakes made by the driver [49, 11]. ADAS has the ability to prevent these types of mistakes by taking the driver out-of-the-loop for different aspects of the dynamic driving task (DDT). The most promising ADAS are Automated Driving Systems (ADS), that can assume full control of the DDT. Unfortunately, full driving automation is still many years away from being available, and in the meantime conditional driving automation, such as the Tesla Autopilot, is what is currently available on the market [14]. In conditional driving automation, the vehicle is able to take over the lateral and longitudinal responsibility for a limited amount of time. During these periods of automation, the driver can engage in secondary tasks, but must remain available to respond to a request to intervene, known as a take-over request (TOR), in cases where the automation has reached its system limit or failed. Ensuring that the driver is able to resume control in time poses one of the major safety challenges for the adoption of conditional driving automation by the public.

A critical safety lever that can be manipulated to affect the safe transitions from automated to manual driving is the amount of time available for taking over control of the vehicle, this is referred to as the time budget (TB) [23]. A research question then arises, what is the optimal TB? This question is a hot area of research within automated driving research, and has been examined often in literature [22, 45, 65, 41, 31]. However, a single optimal TB still has not been identified and agreed on upon.

Instead of identifying an optimal TB, predictive approaches of the TOT have been gaining popularity in literature [38, 23, 8, 55]. TOT prediction shows promise for being able to detect situations where the driver will not be able to take-over in time, regardless of the TB provided. These predictions can be incorporated with technology to develop safety features that can reduce the likelihood of crashes during transitions. However, there is still limited research in this area, as many of the prediction models are unable to adapt to the individual driver behavior or can not predict the TOT on a continuous scale. The research in this proposal intends to address these limitations of past prediction models.

1.1 Research Objective

There are two research objectives in this proposal, which focus on predicting the TOT of drivers on a continuous scale for system initiated take-overs in conditional driving automation, SAE level 3 [4], and how to adapt to within and between driver differences in driver behavior over time. In this proposal the take-over time is defined as the time between the take-over request (TOR) and either a brake or steering intervention. An overview of the thesis work and their relationship to the proposed research questions can be seen in Fig. 1.1. The research questions are:

Question 1: Can the take-over time be accurately predicted on a continuous scale? This research question aims to evaluate whether a hidden semi-Markov model can accurately predict the take-over time of a driver on a continuous scale. This approach is shown to be effective in modeling and predicting driver behavior after an alert. However, other popular regression methods can include linear regression, random forest, and XGBoost. As part of

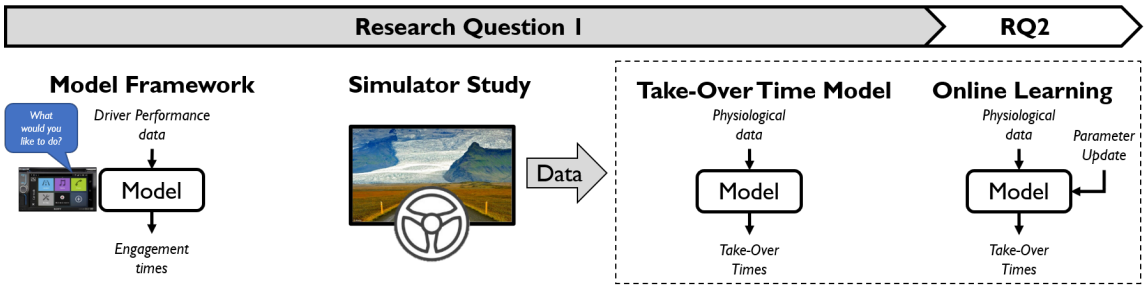


Figure 1.1: An overview of the proposed work and the relationship to the two research questions.

this dissertation, I will try all three, but will be able to identify which is best based on different loss metrics, such as mean absolute errors.

Question 2: Does accounting for the within and between driver differences improve the performance of the take-over time predictions? This research question aims to evaluate whether online learning can adapt to differences between and within drivers, and whether this will improve the model performance. An online learning algorithm will be developed based on research from graphical models. Finally, the performance will be compared to the TOT model learned using traditional learning methods.

Chapter 2

BACKGROUND

This chapter summarizes the current state of knowledge regarding the TOT in conditional automated driving, specifically the factors that can impact it and the state of the art prediction methods. This chapter highlights the importance of TOT predictions and the current gaps in research.

2.1 Levels of Automation

The term ADS refers to the hardware and software that are collectively capable of performing the entire dynamic driving task (DDT) on a sustained basis. Within ADS, there are various definitions of the different levels of automation, but the most widely adopted definitions, which are the definitions used in this proposal, are the ones defined by the Society of Automotive Engineers (SAE). The SAE defines six levels of driving automation that range from 0 to 5, these levels are summarized below [4].

- *Level 0 - No Driving Automation*: The driver performs the DDT at all time.
- *Level 1 - Driver Assistance*: The driving automation system can control either the lateral or longitudinal motion of the vehicle, but not both. The driver is expected to perform the remainder of the DDT.
- *Level 2 - Partial Driving Automation*: The driving automation system can perform lateral and longitudinal control, but the human must monitor the environment and supervise the system at all times.
- *Level 3 - Conditional Driving Automation*: The driving automation system can perform the entire DDT under limited conditions, with the expectation of the driver to respond to a request to intervene.

- *Level 4 - High Driving Automation*: The driving automation system can perform the entire DDT under limited conditions, with no expectation of the driver to respond to a request to intervene.
- *Level 5 - Full Driving Automation*: The driving automation system provides full control and there is no expectation of the driver to intervene.

In the definitions given by SAE, ADS refer to automation levels 3-5. In these levels of automation, the driver is removed from the control loop and the likelihood of human error is reduced. At this time, there are no level 4-5 ADS, and only conditional driving automation (level 3) is available to consumers, an example of this type of system is the Tesla Autopilot [1]. In conditional driving automation, the ADS is able to assume control, for a limited amount of time, of the DDT in specified traffic scenarios. During these periods of automation, the driver is able to engage in secondary tasks while driving, but the ADS will experience scenarios where the system can no longer handle the DDT. An example of this could be when there are no visible lane markings for the ADS to follow, and then the system will need to hand back control of the DDT to the driver (see Figure 2.1). In such situations, the ADS will issue a TOR for the driver to take-over control of the DDT. The driver will only have a limited amount of time, TB, to assume control of the DDT, and the time it takes the driver to resume control of the DDT is called the TOT. These situations pose a critical safety risk for conditional driving automation, and it is still necessary to consider the human in the control loop, more specifically how long it will take the driver to resume back control of vehicle. This proposal focuses on these take-over situations in conditional driving automation (Level-3).

2.2 Take-Over Time

Take-over situations are the key safety critical event that need to be accounted for in the safe adoption of conditional driving automation. In these take-over situations, the ADS has reached its system limit, and must hand over control of the DDT back to the driver. A critical design parameter in these situations is the available time for taking over control of

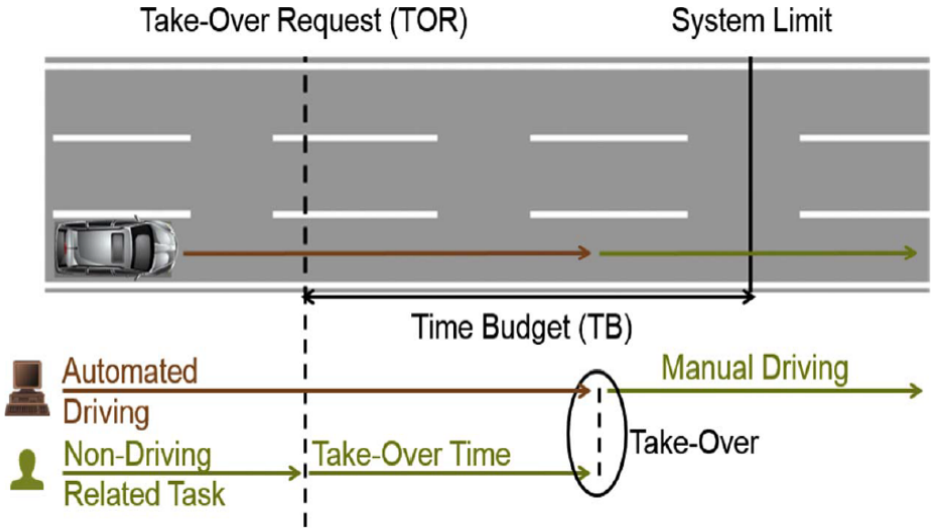


Figure 2.1: The different phases of a take-over event in conditional driving automation. The figure is taken from Gold et al. [23]

the DDT, this time is known as the the TB [41]. If the TB is too short, the driver will not have sufficient time to assume control of the DDT, and an accident may occur [45]. Conversely, if the TB is too long, the driver could lose trust in the system [5]. A natural question then arises, what is the optimal TB?

To answer this question, many studies have been conducted to understand how long it takes for a driver to resume control of the DDT. Radlmayr et al. examined how different traffic density effected the TOT using a driving simulator study and a within-subject design. They found that the TOT was highly dependent on the traffic situation, but could not determine the effect on the TOT [61]. Further studies were conducted by Mok et al. on understanding how secondary task engagement effected the TOT [46], and by Wandtner et al. on how different modalities of secondary task influenced the TOT [66]. These studies found secondary task engagement increased the TOT of driver's, and that different secondary task modalities had a significant effect on the TOT. The study found that secondary tasks that required both visual and manual engagement increased the TOT the most. Hergeth et al. found that prior familiarization with ADS had a significant impact on TOT [28], and a

study conducted by Clark et al. found differences across age on the TOT of driver's [13]. These studies suggest that there are many different factors that affect the TOT, and a single TB will not be optimal in all situations; let alone for each individual driver.

2.3 Take-Over Time Prediction

Modeling and predicting the TOT is a relatively new research area, where the research focuses on identifying periods of time where a transition can be done safely or on predicting the TOT. Prediction models for TOT can be separated into two categories: classification or regression. An exhaustive review of literature found that classification approaches are the only models proposed for predicting the TOT, but regression approaches have been used for modeling purposes.

Braunagel et al. proposed a method that classifies a driver into either a high take-over readiness or a low take-over readiness using gaze and environment features [8]. The take-over readiness was derived from threshold bounds for driver performance metrics, such as lane deviation and time to first brake, and subjective ratings from subject matter experts. With their method and derived take-over readiness, they were able to achieve an accuracy of 79% using linear SVM. Nilsson et al. proposed a model to identify when a driver could safely resume control using a driver controllability set [55]. Their method provided bounds (based on driver performance data from manual driving) on the minimum time headway, time to collision, and longitudinal acceleration for which the transition of autonomous to manual driving were identified to be safe. Their model assumed a fixed TOT for drivers of 1 second and no metrics for model validation were reported. The identification of safe transition periods provide an alternate approach to TOT predictions, but these methods did not consider cases in which crashes occurred; thus, it is not understood how the identified safe transition zones impact the crash likelihood.

Lotz et al. developed a hybrid classification method for predicting the TOT of truck drivers in conditional driving automation [38]. In their method, the TOT was categorized into 4 classes, which were then used with different classification methods for prediction.

Their offline estimation using SVM achieved a misclassification rate of 22.5%, while their online estimation had a misclassification rate of 38.8%. This research tried to bridge the gap between classification and regression, but the categorization of the TOT leads to a loss in information that negatively impact the performance of the model. Classification methods are a first step to predicting the TOT, however, these methods do not tackle the prediction problem directly, and must make assumptions that can negatively impact the model's performance.

Regression approaches address the limitations where classification falls short, but require a higher level of accuracy, as the predictions are on a continuous scale. Currently, no regression models have been proposed for prediction. Instead, regression models have only been used for modeling purposes and to study the effect of different features on the TOT. Körber et al. used a regression approach to model the TOT through the use of Bayesian high density intervals [35]. Their method allowed for the mean TOT and a confidence interval, that can capture the uncertainty, to be estimated, but it required different models for each take-over situations (i.e. traffic density, secondary task engagement). Also, there model required the assumption that there is an underlying common TOT distribution for all drivers. Gold et al. used linear regression to study the contribution of different factors on the TOT [23]. They found that when accounting for individual driver characteristics (demographic and past driver experience), they could explain an additional 30-40% variance, but this approach was not pursued as this data is usually not available in a realistic setting. Their final regression model had a root mean square error of 0.81 seconds, but it was never evaluated on a withheld test set.

2.3.1 Eye Glance Behavior

Recently, researchers have looked to address the unrealistic data sources limitation through the use of physiological measures, such as gaze and head pose behavior. In two recent studies conducted by Louw et al., found that the gaze behavior of a driver is good indicator of whether a driver would crash during take-over situation [39, 40]. This is not surprising since driving

is a highly demanding visual task. They found that driver's who had an erratic pattern of eye fixations were more likely to crash, than those who had a more stable eye fixation during take-over situations. In another study by Körber et al., found that there were significant correlation between the TOT and gaze distribution of driver's [34]. With the advancements of technology, collection of gaze and head pose data can be done through lightweight sensors [2] or the use of machine learning algorithms and video data [26, 47]. Given the potential benefits of using gaze or head pose data as a realistic data source, this research intends to address the remaining two limitations of past research in TOT predictions.

2.4 Gaps in Literature

The current research in TOT predictions, as summarized in Section 2.3, have not been able to predict the TOT on a continuous scale or were unable to account for differences within and between drivers. This is important, as continuous predictions of the TOT can model the crash risk, whereas classification methods fall short. Also, research in driver behavior has shown there exist significant differences across drivers [74], and that prolonged exposure to automated systems can foster behavioral adaptations over time [44]. Not being able to model these individual differences in a driver or changes over time can lead to a decrease in prediction performance.

This dissertation proposal is novel for two reasons. First, it uses a time-series regression approach to modeling and predicting the TOT. This approach allows the model to not only predict the TOT on a continuous scale, but also make updated predictions as more data is obtained. Second, online learning will allow the model to adapt to differences in individual driver behavior and changes in behavior over time.

Chapter 3
MODEL FRAMEWORK

This chapter outlines the development of a driver behavior model (hidden semi-Markov model) framework for modeling a driver's response to an alert. Previous work from driver behavior modeling and human factors that influenced this model is described. Then the model formulation, model learning, and prediction algorithm is presented. Finally, a case study with data from a manual driving simulator study was used to demonstrate the validity of the model. This chapter provide a general framework for predicting a driver response time given that an alert has been issued. This framework incorporates time series data to model the evolution of a driver's decision over time. The objective of this chapter is to develop a framework for addressing the first research question:

- *Question 1: Can the take-over time be accurately predicted on a continuous scale?*

3.1 Model Description

A hidden semi-Markov model (HSMM) is used to capture driver behavior in response to an alert. The motivation for this model comes from two areas of research in human information processing models for warnings and driver behavior modeling.

Human information processing is an area of research that considers models that describe humans perception and use of information. Within this area of research, the Communication-Human Information Processing (C-HIP) model is used to examine how humans process warning information when making decisions [69]. The C-HIP model (Fig. 3.1) has several parts, but this paper highlights two parts that are advantageous for modeling with HSMMs. First, the C-HIP model shows that humans process warning information through a number of hidden states, which mirror the hidden states of the HSMMs. Second, information is

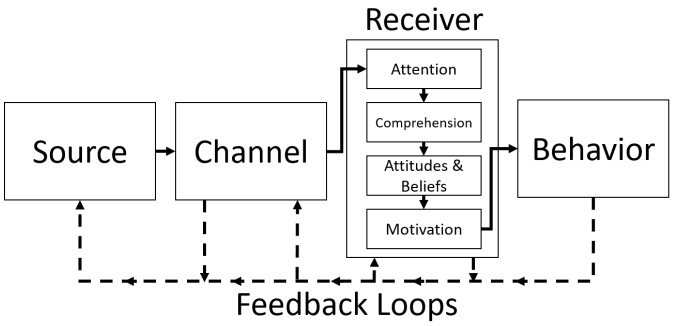


Figure 3.1: Communication-Human Information Processing (C-HIP) model (redrawn from Wogalter [69, p. 52]). The 'Source' and 'Channel' represent the alert, the 'Receiver' is a driver processing the cue, and the 'Behavior' is the response action of a driver.

processed sequentially such that information flows from one state to the next and never backwards. This type of information flow can be obtained through the use of a left-to-right structure for the HSMM.

In driver behavior modeling and prediction, hidden Markov models (HMMs) have been used in a wide range of applications including estimating driver decisions at intersections [21], driver fatigue detection [19], and modeling driver decisions at phase transitions [63]. A drawback of HMM is that the state transitions happen on a fixed interval and the probability of transitioning between the states does not change. This property of HMMs is not suitable for modeling dynamic systems where the underlying transition probabilities can change over time. HSMMs can handle these types of problems due to the semi-Markov property that allows for the state duration to be modeled as a random variable.

3.1.1 Hidden Semi-Markov Models

A HSMM is a doubly embedded stochastic model with an underlying unobservable (hidden) stochastic semi-Markov process that governs another observable stochastic process. The semi-Markov property states that the probability of transitioning between states only depends on the current state and the time spent in the current state, which we denote as $d_t(i)$.

This allows for the transition probabilities between states to dynamically change over time, which mimics how driver behavior can evolve.

We define $O_{1:T}$ to represent the observable driver-performance data sequence from time 1 to T, S_t is a driver's underlying hidden state at time t, N the state when a driver's behavior changes (i.e., response to an alert). A HSMM, ϕ, is then characterized by four parameters:

$$\phi = (A, \Theta, B, \pi) \tag{3.1}$$

where $A \in \mathbb{R}^{N \times N}$ is the nonrecurrent driver state transition matrix, π is the initial driver state distribution that represents the probability of starting at any of the states, B is the probability distribution of the observable data, and Θ is the probability distribution for the state duration times. The equations for these four parameters are:

$$A = P\left(S_{t+1} = j \mid S_t = i\right) \tag{3.2}$$

$$B_i(O_t) = P(O_t \mid S_t = i) \tag{3.3}$$

$$\Theta_i(d_t(i)) = P(d_t(i) \mid S_t = i) \tag{3.4}$$

$$\pi_i = P(S_1 = i) \quad . \tag{3.5}$$

3.1.2 Driver Transition Time Model

The proposed model is a left-to-right HSMM, as shown in Fig. 3.2, where the transitions flow in one direction from the current driver state to the next driver state. This structure has been shown to be effective in modeling driver behavior as a series of sub-sequences [58], and the structure mirrors how warning information is processed by humans, according to the C-HIP model (see Fig. 3.1). A left-to-right structure means that the HSMM always starts in the first state (π), and can only transition to the next driver state and never back (A), see Eq. 3.6. In this model, we defined the observed probability distribution as a mixture of multivariate Normal distribution due to its ability to approximate any finite continuous density function [36], and the duration time as a Normal distribution.

Figure 3.2: A left-to-right hidden semi-Markov model proposed to model a driver response to an alert. O_t represents the observed data at each time point, d_i is the time duration the HSMM spends in each hidden state, and the a_i, j is the transition probabilities between each state can only occur between adjacent states.

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}, \; B_i = \sum_{j}^{M} \delta_j^i \mathcal{N}_j^i(\mu_j, \, \mathbf{\Sigma}_j)$$

$$\pi = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}, \; \Theta_i = \mathcal{N}_i(\mu_i, \mathbf{\Sigma}_i) \tag{3.6}$$

3.1.3 Model Learning

For model learning, we use a modified Expectation Maximization (EM) algorithm in conjunction with the state duration approximation formula from Cartella *et al.* [12]. This improves the duration estimation accuracy and reduces the computational complexity as compared to traditional EM algorithms that discretize the state duration into a finite set of values [71]. In our implementation, the final model is selected based on the Huber loss function to reduce the effect of outliers [27, p.349]. The Python *numpy* package is used to implement the EM algorithm to learn the model parameters.

For the EM algorithm, we start with an initial guess for the model parameters, ϕ, and we continue iterating until the likelihood function, $P(O|\phi)$, no longer improves between consecutive iterations. The initial parameter initialization is crucial in obtaining a good fit for the model. To address this issue, 40 random initializations of ϕ were executed for training, and the model that achieved the smallest loss on the validation set was selected. The driver response model learning procedure is as follows:

1. The hyperparameter is tuned using 6-fold cross validation and grid search on the number of states (N) and the number of Gaussian mixtures (M).

2. The cross validation hyperparameter for each individual is used to train a HSMM with the EM algorithm.

For the modified EM-algorithm, we only update the observation and state duration parameters as we fix the state transition probability and initial state probability to have a left-to-right structure. Our data set includes multiple time series instead of a single stream of data, so the distribution parameters were only updated after one full pass through of the training data set. The update equations for observation distribution (mixture of multivariate Normal distributions) were the mixture probabilities (Eq. 3.7), mean vector (Eq. 3.8), and covariance matrix (Eq. 3.9).

$$c_{ik} = \frac{\sum_{t=1}^{T} \gamma_t(i,k) \cdot (\boldsymbol{x_t} - \boldsymbol{\mu_{ik}})(\boldsymbol{x_t} - \boldsymbol{\mu_{ik}})^T}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{3.7}$$

$$\boldsymbol{\mu_{ik}} = \frac{\sum_{t=1}^{T} \gamma_t(i,k)}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{3.8}$$

$$\boldsymbol{\Sigma_{ik}} = \frac{\sum_{t=1}^{T} \gamma_t(i,k) \cdot \boldsymbol{x_t}}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{3.9}$$

Where $\gamma_t(i,k)$ is the probability of being in a particular state at time t given all the data. This probability is calculated using the forward-backward equations in the EM-algorithm.

The duration parameters were updated using the update equations duration mean (Eq. 3.10) and duration variance (Eq. 3.11). Where $\alpha_t(i)$ and $\beta_t(i)$ are the forward and backward probabilities, $b_j(\boldsymbol{x_t})$ are the observation probabilities, and $a_{ij}(\hat{d}_t(i))$ is the dynamic transition matrix. For a more detailed description of the EM-algorithm and the update equations, please refer to Cartella *et al.* [12].

$$\mu_i = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \left(\sum_{j=1, j\neq i}^{N} a_{ij} (\hat{d}_t(i)) b_j(\boldsymbol{x_{t+1}}) \beta_{t+1}(j) \right) \hat{d}_t(i)}{\sum_{t=1}^{T-1} \alpha_t(i) \left(\sum_{j=1, j\neq i}^{N} a_{ij} (\hat{d}_t(i)) b_j(\boldsymbol{x_{t+1}}) \beta_{t+1}(j) \right)} \tag{3.10}$$

$$\sigma_i^2 = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \left(\sum_{j=1, j\neq i}^{N} a_{ij} (\hat{d}_t(i)) b_j(\boldsymbol{x_{t+1}}) \beta_{t+1}(j) \right) (\hat{d}_t(i) - \mu_i)^2}{\sum_{t=1}^{T-1} \alpha_t(i) \left(\sum_{j=1, j\neq i}^{N} a_{ij} (\hat{d}_t(i)) b_j(\boldsymbol{x_{t+1}}) \beta_{t+1}(j) \right)} \tag{3.11}$$

The update equations presented above are for a single time series, so to update the model parameters for multiple time series we first calculate the updated model parameters for every time series in the data set. Next, we average the model parameters for each time series together to get the final parameter value for updating at each EM-iteration. For the observation model parameters, we use a weighted average (Eq. 3.12) that accounts for the variable length of each time series. Where S_j is the j^{th} EM-iteration model parameter, S_n are the model parameters from each time series for the j^{th} EM-iteration, and L_n is the length of the time series. For the duration model parameters, we use a normal average as each time series example only produces a single realization of the state duration.

$$S_{j+1} = \frac{\sum_{n=1}^{N} S_n * L_n}{\sum_{n=1}^{N} L_n} \tag{3.12}$$

3.1.4 Remaining Behavior Change Time

One of the most important advantages of HSMM is its ability to effectively model the dependencies in time series data due to the semi-Markov property [**?**]. This enables estimation of the remaining time for any given state and the prediction of the expected time until we enter another state.

The last state, S_N,is reached when a driver engages in the secondary task (see Sec.) 3.2.1 for a description of the secondary task. We can then estimate the remaining driver response time as the expected time to reach the last state. When a new observation is acquired, the procedure to estimate the remaining time is:

1. Use the Viterbi algorithm to estimate the current state [18]

2. Estimate the expected remaining state duration time

3. Project the future state transitions until the behavior change state S_N is reached

4. Add up the remaining state and the future state duration times

The current state estimation is performed with the Viterbi algorithm. This allows us to obtain an estimate of the probability of being in state S_i at time t, which we denote as $\delta_t(i)$.

$$\delta_t(i) = \max_{S_1,\cdots,S_{t-1}} P(S_t = i \mid S_1, \cdots, S_{t-1}, O_1, \cdots, O_t) \tag{3.13}$$

Having calculated δ_t, the maximum a posteriori (MAP) estimate of the current state, $\hat{s}_t$, is obtained by (3.14).

$$\hat{s}_t = _{1\leq i\leq N} \ \delta_t(i) \tag{3.14}$$

The expected remaining duration time of the current state is estimated as the difference between the estimated state duration and the expected state duration.

$$d_{avg}^{current} = \sum_{i=1}^{N-1} (\mu_i^{\Theta} - d_t(i)) \odot \delta_t(i) \tag{3.15}$$

In Eq. 3.15, $\odot$ is element wise multiplication and μ_i^{Θ} is the expected value of the duration distribution for state i.

The current state duration time is approximated by Eq. 3.16.

$$\hat{d}_{t+1}(i) \approx P(S_t = i | S_{t+1} = i, O_{1:t+1})(\hat{d}_t(i) + \triangle t) \tag{3.16}$$

This is an iterative equation, with $\hat{d}_0 = 0$ and $\triangle t$ is the time between the current and previous observation.

The projection of the future state transition is obtained from Eq. 3.17:

$$\delta_{t+d} = A^T \cdot \delta_t \tag{3.17}$$

Then the estimate of the new current state is obtained by the taking MAP estimate, Eq. (3.14). The future state projections are continued until the current state estimate is equal to a driver response state, S_N.

The remaining driver behavior change time for the future states is estimated by Eq. 3.18.

$$d_{avg}^{future} = \sum_{i=1}^{N-1} (\mu_i^{\Theta}) \odot \delta_t(i) \tag{3.18}$$

Finally a driver behavior change time estimate is the sum of all duration state time estimates for the current and future states, Eq. 3.19.

$$D_{avg} = d_{avg}^{current} + \sum d_{avg}^{future} \tag{3.19}$$

An estimate for a confidence interval is obtained by following the same steps, but substituting the current and future duration with Eq. 3.20 and 3.21.

$$d_{CI}^{current} = \sum_{i=1}^{N-1} (\mu_i^{\Theta} \pm \sigma_i^{\Theta} - d_t(i)) \odot \delta_t(i) \tag{3.20}$$

$$d_{CI}^{future} = \sum_{i=1}^{N-1} (\mu_i^{\Theta} \pm \sigma_i^{\Theta}) \odot \delta_t(i) \tag{3.21}$$

For more details, please refer to the original paper concerning state duration approximation formula by Cartella *et al.* [12].

3.2 Case Study

To show the validity of this model, data from a previous study using a fixed-base NADS miniSim driving simulator (Fig. 3.3) was used to predict driver response time [33]. The original study was designed to examine cognitive workload as drivers interacted with an in-vehicle voice control system (VCS). The data were used to train and validate the driver behavior change model for this paper.

3.2.1 Simulator Data

Participants

The study was conducted with 24 research participants with an equal number of males and females in each age group. Due to issues with some of the video and simulator data, only

Figure 3.3: The NADS miniSim driving simulator setup used for data collection. The simulator is located on the University of Washington campus in Settle, WA.

16 out of the 24 participants were used for the forthcoming analysis (Table 3.1). The video data were used to manually extract the task initiation time for each participant (Sec. 3.2.1). All participants were required to hold a valid US driver's license and had driven in the US for at least three years. Participants were also compensated for their time and participation in this study.

Table 3.1: Summary statistics of the drivers in the data set used for modeling.

Age Group	Mean Age	SD (Age)	Sample Size	No. of Females
18-24	22.67	1.25	3	1
25-39	29.63	2.43	4	2
40-54	46.6	3.89	5	2
55+	60.0	3.94	4	3

Voice Control System Tasks

A voice control system (VCS) was used in this study to examine driver's engagement in a secondary task. The VCS task required the participant to verbally select a preferred restaurant out of five choices. The participants were always asked to choose an American restaurant based on three criteria: distance, rating, and price. The correct choice would be an American restaurant with the highest rating, lowest price, and closest distance. There was always one selection that was best given these criteria. The criteria information for the restaurants were presented to the participant in three different modes: audio, visual, and hybrid (visual and audio). An example of the IVIS display in the visual and hybrid modes is shown in Fig. 3.4).

The sequence of actions for engagement in the secondary task is described in [37] and shown in Table 3.2. The VCS task would prompt the participant using the auditory statement: "What would you like to do?". After the auditory cue, the participants were required to say the following phrase to start the next task: "Find an American restaurant". The participants were instructed to prioritize driving and only engage in the VCS task when they felt comfortable doing so. Since the goal of this study is to predict a driver's transition time to the secondary task once an alert is activated, we focus on the time duration between steps 1 and 2.

Procedure

The test run consisted of driving on a straight four-lane roadway while following a lead vehicle at a speed of approximately 55 mph. During the drive, the participants were instructed to engage in the secondary VCS tasks after the auditory VCS task prompt and only when they felt comfortable doing so. Each mode of VCS task (auditory, visual, and hybrid) was presented with two levels of cognitive load (high and low). During the drive, the participants performed 6 tasks of each modality (3 high and 3 low), for a total of 18 task initiations. Before starting the experiment, the participants had the chance to familiarize themselves with the

Table 3.2: Modified from Li *et. al* [37]. The sequence of actions from the VCS task; the time between steps 1 and 2 are examined in this paper.

Step	Actor	Mode	Dialog/Action
1	VCS	All	(Task begins) "What would you like to do"
2	Participant	All	"Find an American restaurant"*
3	WOZ** (experimenter)	All	Initiates restaurant list
4	VCS	All	Displays (and/or speaks) restaurant information
3*5	Participant	Audio only	Listens for key information as the system sequentially reads off items for each restaurant, at system's pace
	Participant	Visual only	Visually scans a list of restaurants shown on screen and searches for key information
	Participant	Hybrid	Switches between visually scanning and listening for the list of restaurants
6	Participant	All	Retrieves information from working memory
7	Participant	All	"Select [restaurant name]"
8	WOZ (experimenter)	All	Selects restaurant stated by participant. This entry completes the task (Task ends) and initiates the next task.
9	VCS	All	"Starting navigation"
10	VCS	All	Five seconds pause, then go to Step 1

Note: * The dialog is corrected here from what was presented in Li et al [37];

**WOZ=wizard of oz protocol

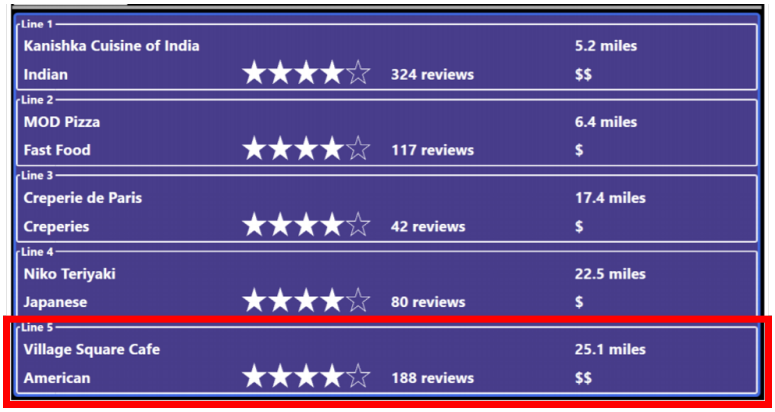


Figure 3.4: User interface for the VCS tasks under visual-only and hybrid modes. The screen would stay blank during auditory-only mode. The correct answer is shown in the red outlined box.

driving simulator and the VCS tasks.

Data Extraction

The task initiation time was defined from the initiation of the alert (Step 1 in Table 3.2) to the time the participant verbally engaged the system to start the VCS task (Step 2). A plot of the task initiation time for each participant is shown in Fig. 3.5. The vehicle's angular velocity and longitudinal acceleration were used as the inputs to the models. The simulator recorded data at 60 Hz.

3.3 Results

Each of the 16 participants performed 18 tasks during the simulator study, resulting in 18 task initiation times per participant. The task initiation times were split into a training set ($n = 12$) and a test set ($n = 6$) for every participant. This split led to a final data set of 192 and 96 task initiations for training and testing, respectively.

The train-test split was stratified based on task modality as participants can observe the difference in modality at the start of the task. Task difficulty was not considered because

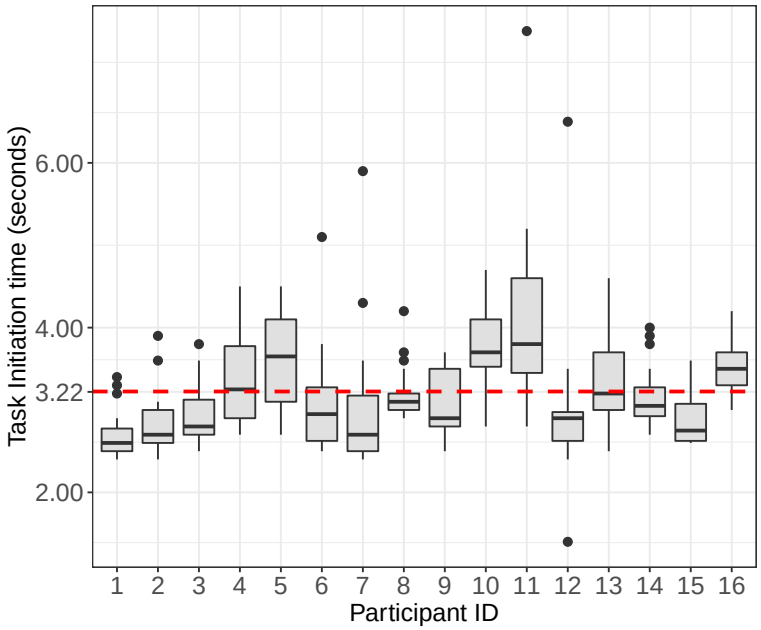


Figure 3.5: Task initiation time is defined as the time between the VCS starting prompt (Step 1, Table 3.2) and the participant's verbal initiation of the VCS task (Step 2, Table 3.2). This is explained in Sec. 3.2.1. The mean task initiation time is shown as a red dashed line and equals 3.22 seconds.

participants had no knowledge of the difference in cognitive demand prior to task engagement and therefore, it had not impact on the time between steps 1 and 2. The training data were standardized with a mean of zero and a standard deviation of 1. The values obtained from the training data were applied to the test data.

Three models were trained and compared for each participant: individually tuned hyperparameters, a low complexity model ($N = 3$ and $M = 3$) and a high complexity model ($N = 5$ and $M = 5$). For model evaluation, 11 predictions were made at equally spaced time intervals between the time the alert was provided until the time a driver engaged the IVIS. The performance results of only one model are presented in greater details.

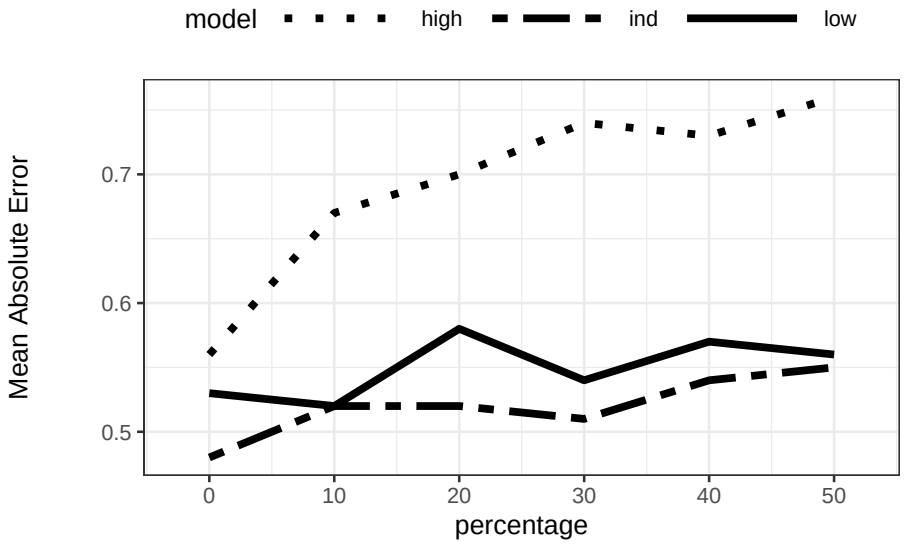
3.3.1 Comparison Among Models

Fig. 3.6 shows a subset of the predictions on the test data set from the time the alert was provided (0%) until one-half of the participant's response time had passed (50%). At the 50% mark, more data are acquired and the model prediction gets updated. For example, if a participant had a total response time of 4 seconds, 0% would be the first performance data point at the time of the alert. At 50% (or 2 seconds into the response time), the prediction is updated with the 120 additional data points (2(sec)x 60(Hz)) that are acquired at this point in time. For the *high complexity* model, the prediction errors increased as more data were acquired. For the *individual* and *low complexity* models, the prediction error remained relatively constant over time. This observation can be explained by the fact that 11 of the 16 individually tuned models had the same parameters as the low complexity models, implying that the high complexity models were overfitted to the training data.

When comparing the performance of the models, we use the mean absolute error (MAE) instead of mean absolute percentage error (MAPE) since it is easy to interpret as its unit is in seconds. Moreover, very large changes in the MAPE can occur even due to small prediction errors when the remaining task initiation times are small. The individually tuned hyperparameter had the best overall prediction performance (MAE: $\bar{X} = 0.51, SD = 0.51$) and the low complexity model performed slightly worse (mean absolute error: $\bar{X} = 0.55, SD = 0.52$). The high complexity model performed the worst (MAE: $\bar{X} = 0.72, SD = 0.68$). In the real world, it is not feasible to tune the individual hyperparameters for each participant. Given that the low complexity model performed only marginally worse with respect to MAE and the hyperparameters are fixed beforehand (i.e., they do not require tuning), we move forward with analyzing the performance of this model in greater details.

3.3.2 Task Modality

To analyze whether the task modality had an effect on the model prediction performance (absolute error), an analysis of variance (ANOVA) was performed on the test data set. For



Model	0	10	20	30	40	50
Low	0.53 (0.49)	0.52 (0.53)	0.58 (0.54)	0.54 (0.49)	0.57 (0.48)	0.56 (0.47)
Individual	0.48 (0.41)	0.52 (0.53)	0.52 (0.49)	0.51 (0.49)	0.54 (0.52)	0.55 (0.52)
High	0.56 (0.50)	0.67 (0.59)	0.70 (0.58)	0.74 (0.57)	0.73 (0.59)	0.76 (0.63)

Figure 3.6: Mean absolute error of each model with the standard errors shown in parentheses.

the individual models, only modality was considered in the ANOVA. For the low and high complexity model, the analysis included a driver variable to account for individual differences.

There were no significant differences in modality (assessed at α=0.05) for the individual models (F(2,1028) = 0.20, p = 0.8214) or low complexity models (F(2,1028) = 2.07, p = 0.1273). There was a significant difference in modality for the high complexity models (F(2,1028) = 4.49, p = 0.0114). More specifically, the combined visual/audio (hybrid) mode (mean absolute error= 0.90) was significantly higher when compared to the audio mode (mean absolute error=0.74).

Further investigation on the high complexity model shows that the training error and standard error (MAE=0.59$\pm$0.57) were much smaller than the test error (MAE=0.72$\pm$0.68). This suggests that there may be overfitting in the high complexity model, which would lead to poor prediction performance in the test set, This overfitting may also lead to differences observed in modality. Given that the task initiation prompts were identical regardless of task modality (Sec. 3.2.1), the remaining analysis focuses on the low complexity model and does not include task modality.

3.3.3 Driver Response Time Predictions

The low complexity model has a maximum absolute error of 3.34 seconds, but 85% of the predictions fell within an absolute error of 1 second, see Fig. 3.7. These results show that the model of driver response time is able to make predictions with a high degree of accuracy. Fig. 3.8 shows the model prediction for two different change times for the same participant. Each time a new observation is acquired, the model can update the prediction while also providing the uncertainty of the prediction through a confidence interval. The confidence intervals were calculated using Eq. 3.20 and 3.18 at a 95% confidence level.

As we examine the predictions over time in Fig. 3.8, there are two advantages of the model. First, the model can update predictions as new data are acquired. Initially, when the alert is given, the predictions from the model are exactly the same. As new data are acquired, the model can take advantage of the time series aspect of the data to provide an

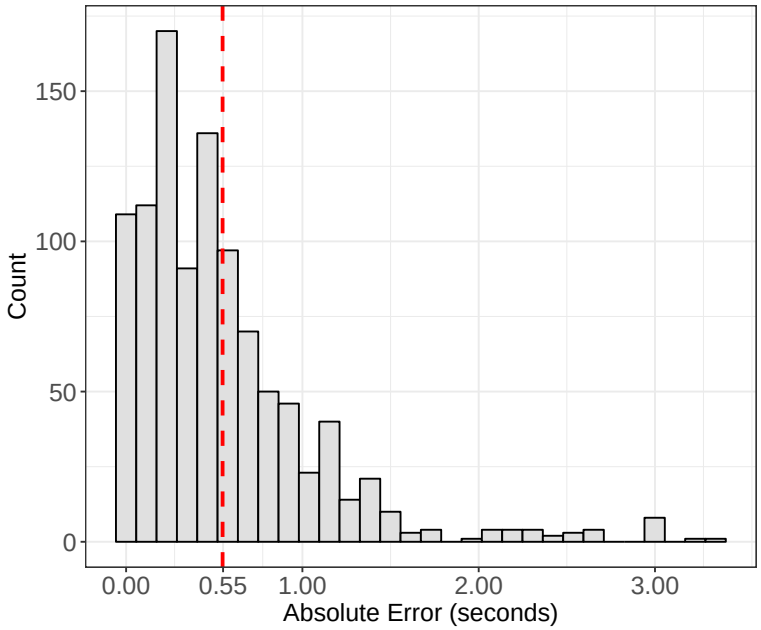


Figure 3.7: The absolute error of predictions on the test data for the low complexity model. The mean absolute error is 0.55 (red dashed line).

updated prediction that tend to converge to the true driver response time. This allows the model to adapt to drivers' responses over time. This can be observed in Example 1 of Fig. 3.8, where the model detects a change at time index=2.5 seconds and updates the prediction accordingly. This is one of the advantages of HSMMs when compared to other regression based methods that just predict a fixed value given the same input.

The second advantage of the model is its ability to quantify uncertainty in the form of a confidence interval. As more data are included in the model, the confidence interval of the model will shrink.

3.4 Discussion

This chapter describes a framework for predicting a driver transition time to a secondary task after an alert is issued. The predictions are achieved using time series data and a left-to-right HSMM. We demonstrate the performance of the framework using data from a

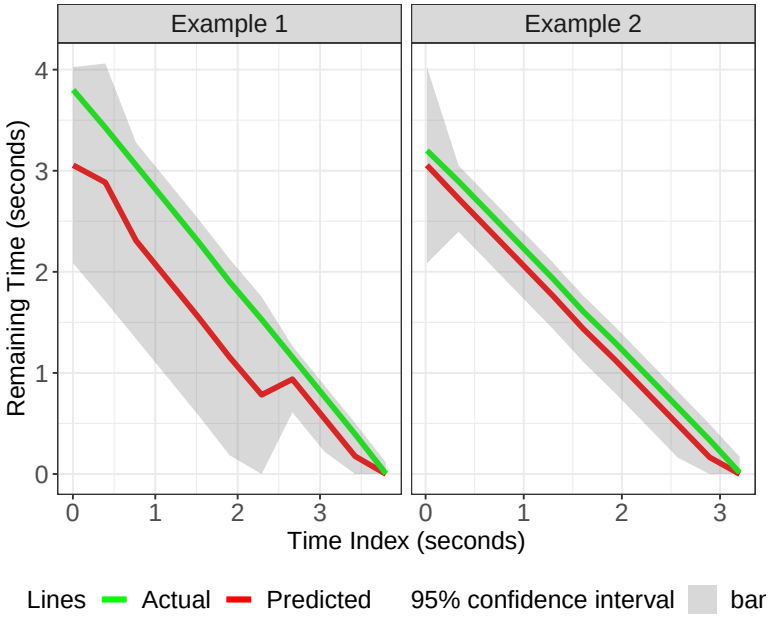


Figure 3.8: Two example predictions from the low complexity model. The prediction intervals converge to the actual driver response time and its uncertainty decreases as the true time approaches.

driving simulator that explored drivers' interaction with an IVIS. A low complexity HSMM performs almost as well as the individually tuned models in terms of mean absolute error. The prediction error for the low complexity and individual models stabilizes after 20% of the data are acquired. This shows that only a small amount of data needs to be collected after an alert is provided for the model to adapt to the specific driving situation and driver behavior. We also show that this method accurately predicts a driver's transition time within 0.55 seconds, on average.

The transition times predicted by this model are points of high risk driving behavior, where a driver is changing their attention from the road to some task engagement and vice versa. Identifying these crucial time points can help us better design in-vehicle safety features to mitigate these risks. This model can, therefore, be used within collision warning systems or conditional autonomous vehicles to help facilitate timely system interventions or handover of vehicle control.

This study has two limitations that impact the interpretability and application of the model. The model was only used for one type of in-vehicle system. While it worked reasonably well for the system we examined after acquiring only 20% of the data, that may not be the case for other in-vehicle systems and advanced technologies. The system used in this study was also not a collision warning system, meaning that additional investigation needs to be done. Second, the model was validated only for a straight road in a simulated rural environment. There are also varying complexities of on-road environments and driver performance data depends highly on the driving situation [**?**]. Thus, the model performance still needs to be studied for different environmental conditions.

In summary, this chapter provides a prediction framework for capturing a driver's transition time given an in-vehicle alert. As technology continues to advance, it becomes increasingly important to understand if a driver can respond in an appropriate time frame. While the study focused on transition time to a secondary task, there are implications for driver take-over time that we will explore in a later chapter.

Chapter 4

SIMULATOR STUDY

This chapter outline the simulator study designed for data collection. This study utilizes a driving simulator study that was conducted at the University of Washington in Seattle, WA from January-March 2020. A conditional driving automation (adaptive cruise control and lane keeping) was used to capture the drivers take-over time during various take-over events. A visual-manual secondary task was used to induce a distraction, a inertial measurement unit sensor was used to capture the head position of the driver, and a driving simulator was used to capture the take-over time. The study was approved by the University of Washington IRB (STUDY00008915). Informed written consent was obtained from each participant at the beginning of the study. This study was a 3^1 (Time Budget: 1, 2, and 3 seconds) repeated measure (4) partial factorial experiment.

4.1 Participants

Thirty-four licensed drivers (21 male, 11 female, and 2 no responses) between the ages of 19 and 56 (mean=30.4, SD=11.2) were recruited for this study. All participants had a valid US driver's license, drove at least 3,000 miles annually, and were fluent in English. Participants were compensated $35 for the study with up to an additional $5 depending on their performance on the secondary task. This compensation tied to the secondary task was added to incentivize the participants to focus their attention on the secondary task throughout the drive.

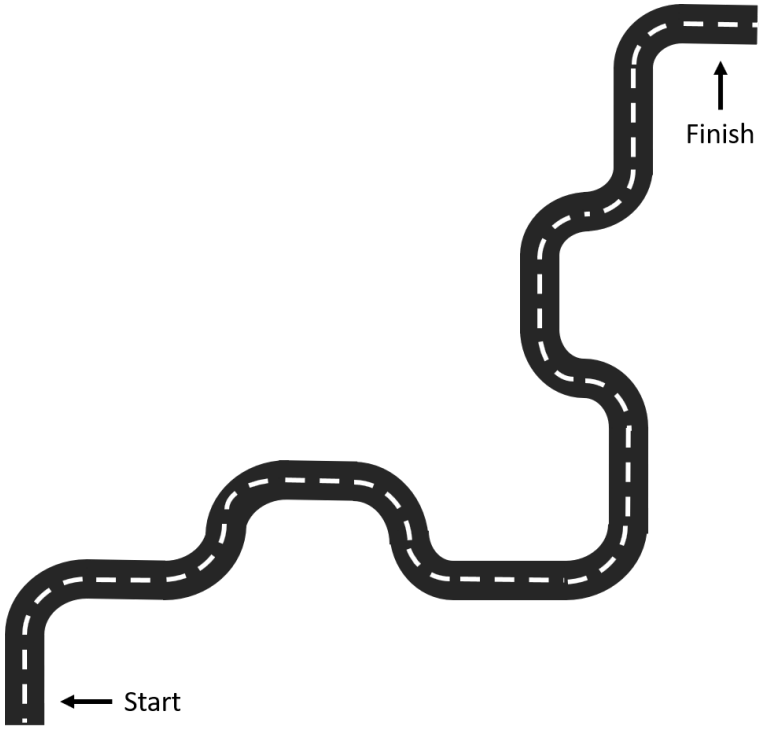


Figure 4.1: Scenario roadway configuration (not to scale).

4.2 Apparatus and Sensors

4.2.1 Driving Simulator

A fixed-based National Advanced Driving Simulator (NADS) miniSim quarter cab driving simulator was used in this study (see Fig. 3.3). The display of the simulator contains 3 widescreen monitors that emulate a 140 degree horizontal field of vision and a 30 degree vertical field of vision. The data was collected at 60 *Hz* from the simulator.

The driving scenario used in this study was specifically developed for this dissertation. The driving scenario was a 2-lane highway in each direction, that was separated by a grass median strip. The road configuration included straight segments and wide curve segments, see Fig. 4.1. Each drive took approximately 25 minutes to complete. The posted speed limit was 65 miles per hour (mph) and the lead vehicle traveled at a fixed speed of 65 mph.

During the drive, the participant's vehicle was engaged in a level-3 conditional driving automation (adaptive cruise control and lane keeping). This meant that the driver did not have control the steering or acceleration/braking of the vehicle. The vehicles speed was set at 75 mph and the conditional driving automation had 3 different time budget (follow distance) settings: 1, 2, and 3 seconds. Throughout the drive, there were 12 unannounced take-over events that were simulated at random with a uniform 2 to 3 minutes between events. The participants were given a random order of the 3 different TBs that were each repeated 4 times each. An auditory TOR was given to the participants for a fixed time of 10-seconds during the take-over events. During the take-over events the lead vehicle would slow down from 65 mph to 55 mph over 4 seconds. The participants were given instructions and adequate training to familiarize themselves with the driving automation and the TOR auditory cue before the main drive.

4.2.2 Secondary Task

While the vehicle was engaged in the conditional driving automation, the participant was required to perform a visual-manual secondary task using a touch screen monitor (see Fig. 4.2). A visual-manual task was chosen to simulate a distraction that is similar to using a smartphone. The secondary task was a web application that was developed using R Shiny. The task required the participant to type the word shown on the screen into a text input box before pressing the 'Submit' button (see Fig. 4.3). The secondary task continuously displayed a new word after submission, with a second delay between words. Participants were instructed to prioritize the secondary task when the vehicle was engaged in the conditional driving automation, and to not perform any secondary tasks when they had to take-over control and manually drive.

4.2.3 Head Pose Sensor

A head position measurement device was built for this thesis using a headband and an IMU sensor from SparkFun, see Fig. 4.4 [3]. The IMU sensor combines a microprocessor

Figure 4.2: The location of the secondary task used for the study.

Figure 4.3: User interface for the secondary task, which required the participant to type a 6 letter word before pressing submit.

with three, 3-axis sensors: an accelerometer, gyroscope and magnetometer. This allowed the sensor to measure its own orientation, which we used to record the head position of the participant. The code was developed using Arduino and built in packages developed by

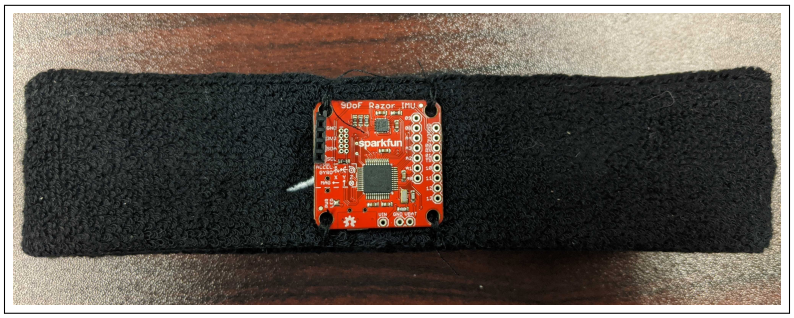


Figure 4.4: The head pose sensor and headband worn by participants. The headband was worn with the sensor located in the middle of the participants forehead.

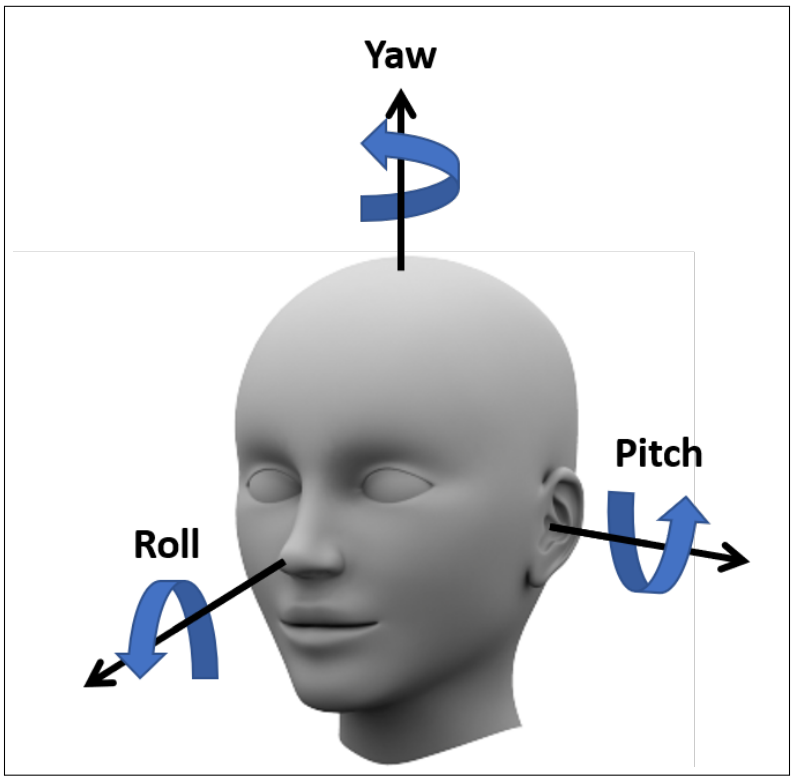


Figure 4.5: The 3 angles measures by the head sensor.

SparkFun. The head pose data was recorded at 60 Hz and Fig. 4.5 show the three angles (roll, pitch, and yaw) that are measured by the sensor.

4.3 Procedure

When participants arrived to the study, they were instructed to review the consent form and ask any questions they had before signing. After signing the consent form, participants were given two forms of training before the main drive. The participants were given instructions and familiarization of each component of the study. Afterwards, the participants had a warm-up drive where they were allowed to practice the take-over events and familiarize themselves with the auditory TOR. Once the participant was comfortable, the head pose sensor was attached and the main drive was started.

The main drive took approximately 25-minutes to complete. After the drive, the participants took a post-driving survey consisting of 6 questions asking about past exposure to autonomous vehicles and their trust in the autonomous system during the scenario. The trust and likelihood to purchase question were a 7-point Likert scale question, where 1 indicated very low and 7 indicated very high. The questions asked were:

1. What is your age?
2. What gender do you identify as?
3. Have you ever driven in a semi-autonomous or a self-driving vehicle? (Ex: Tesla, BMW, etc.)
4. If yes, how many times?
5. How much did you trust the self-driving system?
6. How likely are you to buy a self-driving car?

4.4 Dependent Variable: Take-Over Time

The TOT was defined as the time between the TOR and the start of the driver's maneuver. The threshold values that determined the start of the driver's maneuver are a 2° steering

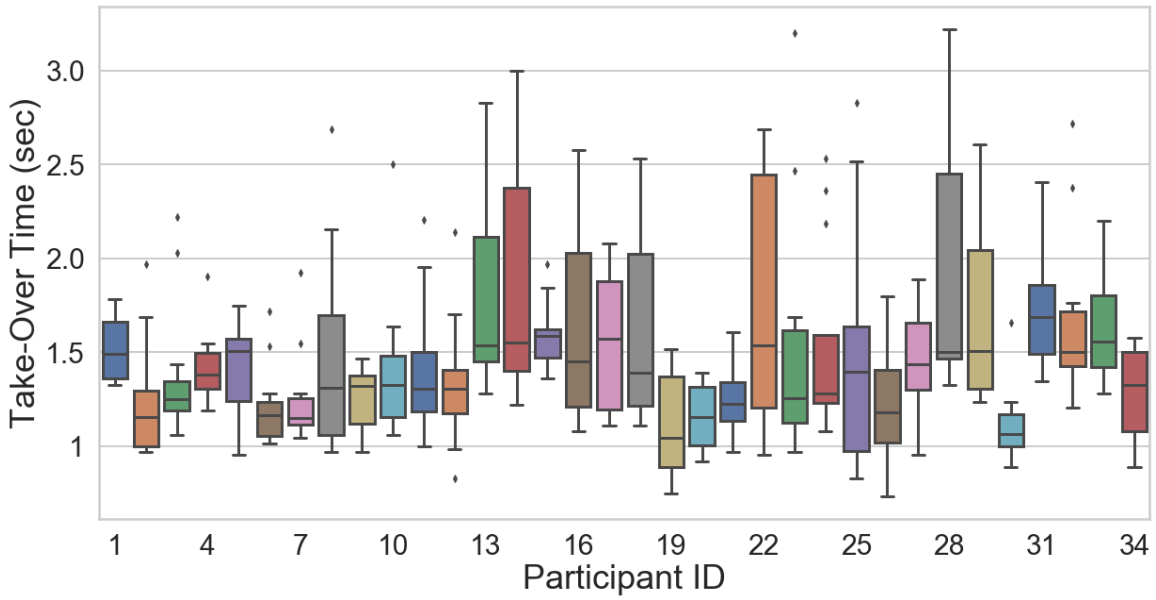


Figure 4.6: A boxplot of the 12 take-over times for each of the 34 participants. The mean TOT was 1.48 seconds with a standard deviation of 0.45 seconds.

wheel angle change or a 10% brake or acceleration pedal actuation [23]. The final data set consisted of 404 take-over events, instead 408, because 4 take-over events were removed due to a 0 second TOT. These were deemed to be outliers and removed because the participant was anticipating a take-over event. The mean TOT was 1.48 seconds with a standard deviation of 0.45 seconds, see Fig. 4.6, with a maximum TOT of 3.22 seconds and a minimum TOT of 0.74 seconds.

Chapter 5
TAKE-OVER TIME MODEL

In conditional driving automation, the vehicle is able to take over the lateral and longitudinal responsibility for a limited amount of time [4]. During these periods of automation, the driver can engage in secondary tasks, but must remain available to respond to a request to intervene in cases where the automation has reached its system limit or failed. Ensuring that the driver is able to resume control in time during vehicle control hand-off poses one of the major safety challenges for the adoption of conditional driving automation by the public [22]. This chapter presents a driver TOT model, then evaluates the model's performance using a driving simulator study (see Ch. 4), and finally compares the performance against state-of-the-art machine learning regression models. The objective of this chapter is to address the first research question of this dissertation:

- *Question 1: Can the take-over time be accurately predicted on a continuous scale?*

5.1 Analytical Methods

An important step in any machine learning method is finding a representation of the original measures that correlates with the target of interest. So, first the head pose data was pre-processed to create an input feature for TOT modeling. Then, the HSMM developed in Sec. 3.1.1 was modified to model the TOT of drivers.

5.1.1 Data Pre-Processing

Before we could generate features, we had to correct the sensor drift of the raw head pose data (see Fig. 5.1 & 5.2). This was due to the fact that the head position sensor was free floating and drift can occur. To fix this issue, we fixed the head facing forward at the road

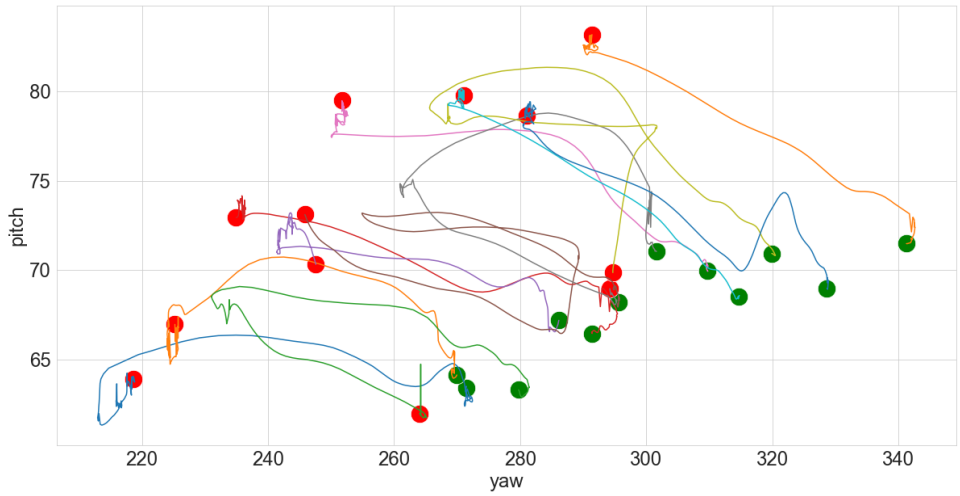


Figure 5.1: An example of the raw head movement (yaw and pitch) before data cleaning of a single participant for the 12 take-over events. The green points are the starting head position of the drivers at the time of the TOR and the red points are the end position 2-seconds after the take-over. We can see that the sensor had some drift on the measurements that needed to be corrected.

to be 0. This was done by using a small portion of each time series to shift the values where the participant was driving the vehicle and not working on secondary.

For feature generation, we had three issues we wanted to address when pre-processing the data. Those three issues were the variable head starting location, the head movement direction, and the exact head path. Only the yaw (left and right head movement) and the pitch (up and down head movement) Euler angles were used and pre-processed prior to model training, see Fig. 4.5. First, short time series segments were extracted for each take-over event from the time of the TOR to 2-seconds after the participant had taken over control of the vehicle.

To account for the differences in the starting head position when the participants head was off the road, the data was shifted to have the starting head location at the time of the TOR to be 0. This was done by shifting the data by using the first data point at the time of the TOR. This shift is independent of future data points and the train-test split.

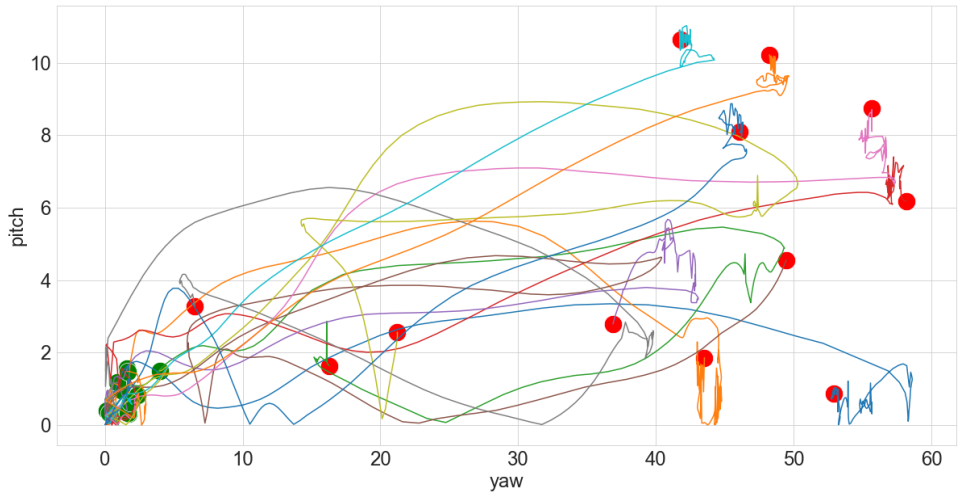


Figure 5.2: An example of the cleaned head movement (yaw and pitch) of a single participant (same as Fig. 5.1) for the 12 take-over events. The green points are the starting head position of the drivers at the time of the TOR and the red points are the end position 2-seconds after the take-over. We can see that the drift has been corrected and the starting positions have all been standardized.

Then the range of the data was transformed to contain the values between $[-90, 90]$. All values outside of this range were truncated to be 90 or -90. Next the absolute value was taken to constrain the values between $[0, 90]$. This was done to have the features be independent of the direction the driver was looking (i.e. left or right, up or down). Then the values were normalized to be between $[0, 1]$ by dividing everything by 90.

Finally, the two features were combined into a single features by using $max(yaw, roll)$. Past research indicates that a stable gaze on the center of road after take-over event correlates with a low crash risk [39]. So the exact path of the drivers head back to the center of the road is not as important as are they looking down the road or not. Finally, the data was then downsampled to 20 Hz using a fast Fourier transform to reduce the noise of the raw measurements. Two examples of the final pre-processed timeseries can be seen in Fig. 5.3.

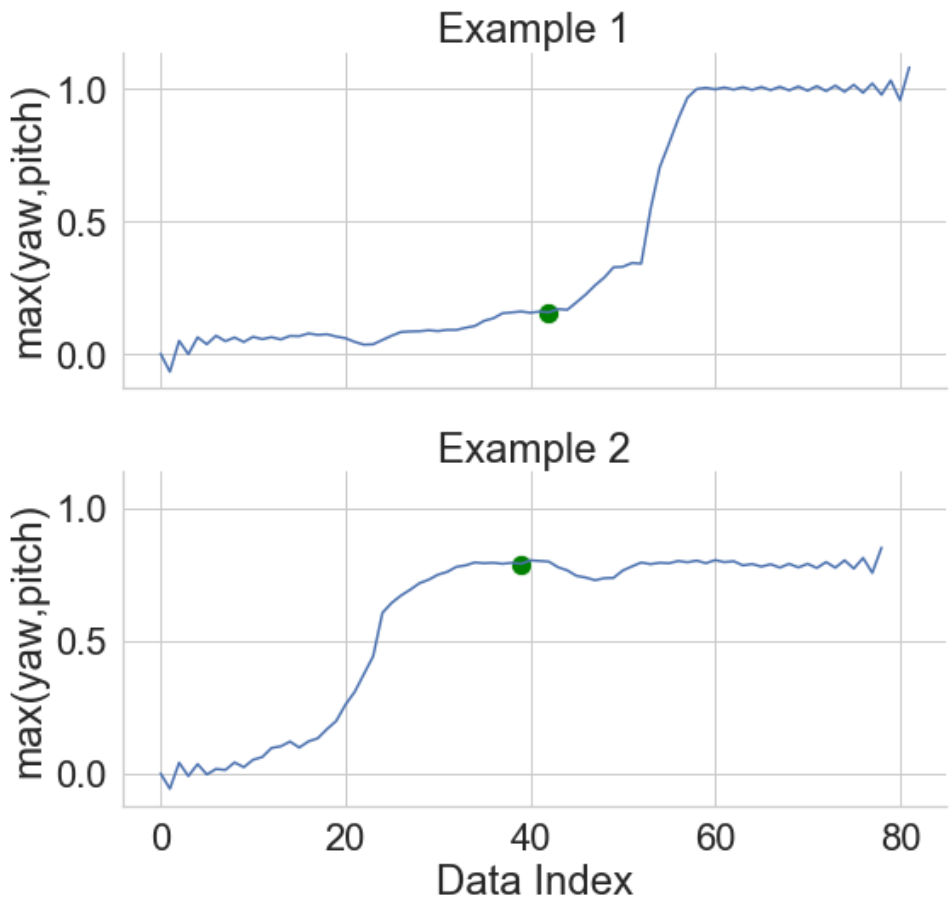


Figure 5.3: Two examples of the final pre-processed data for two different take-over events for the same participant. The green dot indicates the time when the participant has taken-over control of the vehicle.

5.1.2 HSMM

A HSMM was used to model the TOT of drivers. The state transition structure, model learning, and TOT predictions were the same as described in Sec. 3.1.1. The differences between the model described in Sec. 3.1.1 and this one is the distribution used for modeling the observations and the criteria used for model selection.

For model selection, the log likelihood was used for determining the final model, and the model parameters (the number of hidden states, N) were tuned using the training error. A

normal distribution was used for modeling the observation distribution and thus requires a new update equation for calculating the sufficient statistics (mean and variance). The new update equations for the observation distribution parameters are:

$$\mu_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot \boldsymbol{x_t}}{\sum_{t=1}^{T} \gamma_t(i)} \tag{5.1}$$

$$\sigma_i^2 = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot (\boldsymbol{x_t} - \mu_i)(\boldsymbol{x_t} - \mu_i)^T}{\sum_{t=1}^{T} \gamma_t(i)} \tag{5.2}$$

5.2 Results

The final data set consisted of 404 take-over events, instead 408, because 4 take-over events were removed due to a 0 second TOT. The data were then stratified on the time budget and split into a training set (302 take-over events) and test set (102 take-over events). Each participant had approximately 9 take-over events for training and 3 for testing.

For model training, the initial parameter values can greatly affect the final model learned. We used 20 parameter initializations, where we randomly selected 5% of the data to use for the starting parameters. The model with the largest log likelihood out of the 20 initializations was kept as the final model. For parameter tuning (i.e. the number of hidden states), we used the training error to determine the final model to assess on the test data set.

For model prediction, 11 predictions were made at evenly spaced percentiles for each time series from 0% (time of the TOR) to 100% (the time the driver had taken-over control), with 10% between predictions. The predictions were made at [0%, 10%, ... 90%, 100%] with the prediction target (y) being the remaining take-over time. For example, if a participant had a take-over time of 2 seconds, 0% would be the first data point at the time of the TOR with a target of 2 seconds. At 50% (or 1 seconds into the TOT), the prediction is updated with the 20 additional data points and the target is 1 second (2 (sec) TOT - 1 (sec) elapsed time).

When assessing the prediction performance of the models, we use the mean absolute error (MAE) instead of mean absolute percentage error (MAPE) since it is easy to interpret as

its unit is in seconds. Moreover, very large changes in the MAPE can occur due to small prediction errors when the remaining TOT are small. The mean TOT was 1.48 seconds with a standard deviation of 0.45 seconds. The maximum TOT was 3.22 seconds and the minimum TOT was 0.74 seconds.

5.2.1 Driver Head Behavior During Take-Over Events

Past research has shown that gaze reactions were on average to be later than the time it took to deactivate the automation [65]. We wanted to investigate whether our data set contained a similar driver behavior. We split the data into two sets, one for head facing forward ($n = 142$) before the take-over and the other for head facing off of the road ($n = 262$) at the time of the take-over. For the purpose of this analysis, any value below 0.2 (18 degrees) was considered to be head off the road.

After the split, we found that participants took control of the vehicle before their head was facing the road in 262 of the take-over events, while only 142 take-over events had the head facing the road before the participant resumed control. This finding is the same as Vogelphol *et. al*, where a majority of the participants reacted first before assessing their driving environment [65].

Next we wanted to analyze if the TOT for the two corresponding actions were significantly different. Fig. 5.4 shows a boxplot of the TOT for the two data sets (on and off road), we can visibily see that when the participants head is facing the road before the take-over, they take longer to take-over control of the vehicle. The mean TOT for head off the road is 1.33 seconds and the mean TOT for head on the road is 1.74 seconds. We conducted a Welch's t-test for unequal variance and sample size and found the mean difference in the take-over times to be statistically significant (t-value = -8.086, p-value = $1.023e^{-13}$). A statsitical power analysis for a unpaired two-sample t-test was performed for unequal variance and unequal sample size. The significance-level was set to 0.05 and the effect size was 0.41. Finally, the power was calculated to be 0.9473, which is significantly larger than a commonly used value of 0.80.

Lastly to visually see the difference in head pose behavior, we calculated the dynamic

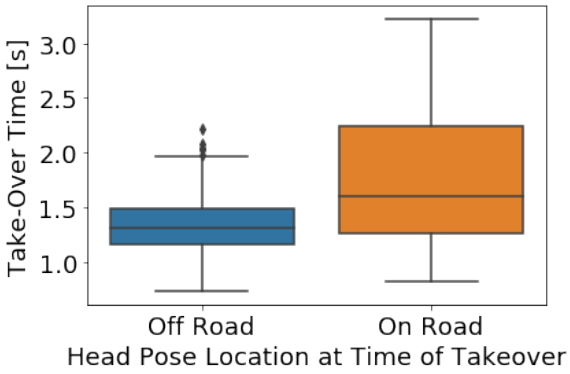


Figure 5.4: A boxplot of the take-over times for the training data sets of the head to road before and after at the time of take-over. The mean TOT for head off the road is 1.33 seconds, and the mean TOT for head on the road is 1.74 seconds.

time warping (DTW) barycenter averages for each of the driver behaviors and plotted them in Fig. 5.5 [60]. This is a method of averaging time series of different lengths. We can see that both head positions start at 0, where the participant is focused on the secondary task. The participants who had their head position facing the road, start moving their head right away, but it takes some time for them to assess the driving situation and their head position to stabilize. While, the participants who had their head position off the road seem to be focused on the secondary task and only start to move their head position as they are taking over. Given the differences in the TOT, we will be training a separate model for each of the behaviors outlined above.

5.2.2 Take-Over Time Model Performance

The HSMM TOT model had an overall MAE of 0.269 seconds with a standard error of 0.259 seconds. If we look at the performance of the two individual models, the head off the road (3 hidden states) had an $MAE = 0.246 \pm 0.200$ and the head on the road (5 hidden states) had an $MAE = 0.360 \pm 0.352$.

Along with point predictions, we can produce confidence intervals (CI) to represent the level of uncertainty of the prediction as we are able to model the state duration with a

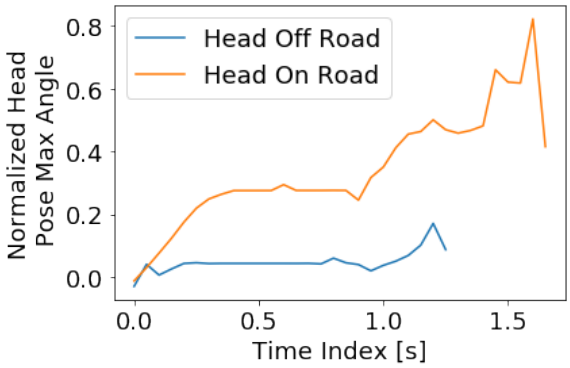


Figure 5.5: The dynamic time warping barycenter average of the head on road and head off road examples for the train data set [60]

distribution (normal distribution). For a 95% CI, the mean width for the head off road model is 0.974 seconds and 86.7% of the predictions fall within the CI. For the head on the road model the mean width is 0.903 seconds and 59.7% of the predictions fall within the CI.

5.2.3 Model Comparison

To assess our model performance, we trained 3 other regression models for comparison. These models were a linear regression, random forest, and XGBoost. These models can not be trained on time series of variable length, so we used 1 seconds worth of time lags as features for models training on the already pre-processed data, see Sec. 5.1.1. This equated to 20 input features for each of the 3 regression models. Two models were trained just like the HSMM for each of the driver behaviors identified in Sec. 5.2.1.

The models test error, the average of both models (head on and head off the road), can be seen in Table 5.1. We can see that the HSMM performed slighlty better on average than the XGBoost, but their prediction performances weren't significantly different due to the overlapping standard error.

Although the overall prediction performance (MAE) of the HSMM and XGBoost are approximately the same, we decided to analyze both models in detail to see how they differed.

Table 5.1: Model mean absolute error on the test data set.

Regression Model	**Mean Absolute Error (s)**	**Standard Error (s)**
Linear Regression	0.341	0.254
Random Forest	0.306	0.242
XGBoost	0.282	0.247
HSMM	0.269	0.259

First we looked at how the models performed at the different prediction percentiles (i.e. as more time series data is acquired), see Fig. 5.6. The HSMM has better initial prediction performance (0-40%) and end prediction performance (80-100%), while the XGBoost performs better in the middle of the time series data. Both models tend to have worse performance as more data is acquired, this will be elaborated on in more detail in the discussion. But this still does not give us a clear picture of how the models differ, so next we analyze some examples of each model's predictions.

Fig. 5.7 shows two example predictions for the same input time series for both the HSMM and XGBoost to see how the different model properties affect the model performance. The main property we are going to focus on is how each model handles variable input sizes. The HSMM can handle time series of variable length, while the XGBoost has a fixed input size that has to be predetermined before hand (e.g. 1-second). In the top row we have example of poor model predictions and in the bottom row are examples of good model predictions. We can see that the HSMM usually has a monotonically decreasing TOT prediction and better convergence even when the model is performing poorly. On the other hand, the XGBoost TOT predictions can increase from the last prediction when the model is not performing well.

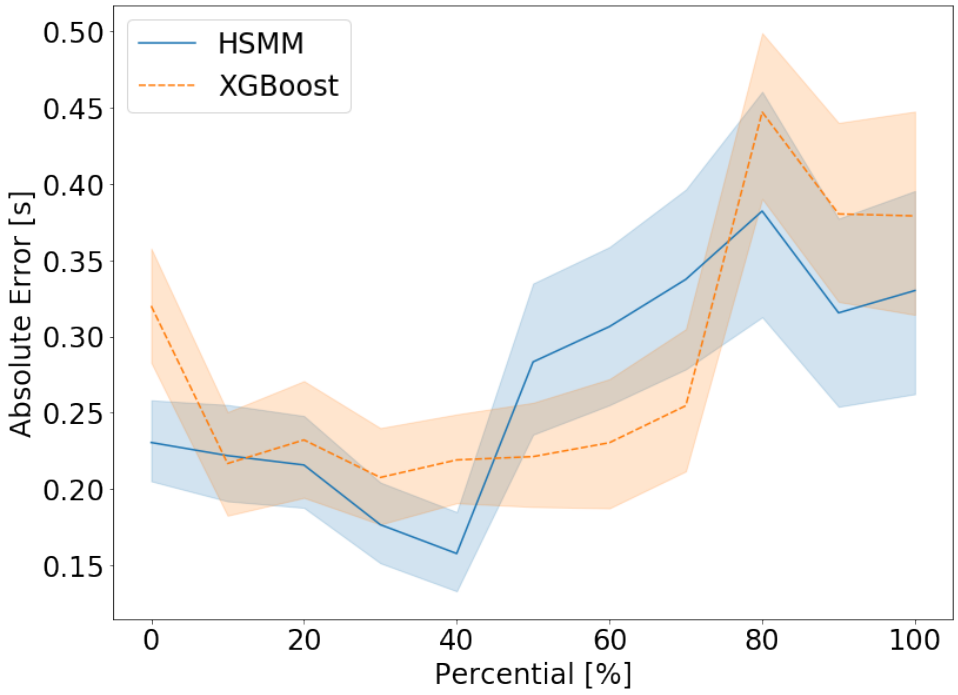


Figure 5.6: Model prediction performance at each prediction percentile, see Sec. **??**, for the HSMM and XGBoost. Both models had similar prediction performance with HSMM performing better at the end points (0% and 100%) and XGBoost performing better in middle (20% to 60%).

5.3 Discussion

This chapter evaluated this first research question, which focused on whether the TOT can be accurately predicted on a continuous scale. The head pose data was first separated into two different data sets based on the driver behavior at the time of the take-over. Then two models were trained and the performance was assessed using MAE. Finally, the model was compared to other commonly used machine learning regression models to quantify its performance. The results showed that the modeling framework was able to accurately predict the TOT to within a MAE of 0.269 seconds using a simple data source of the head position of the driver.

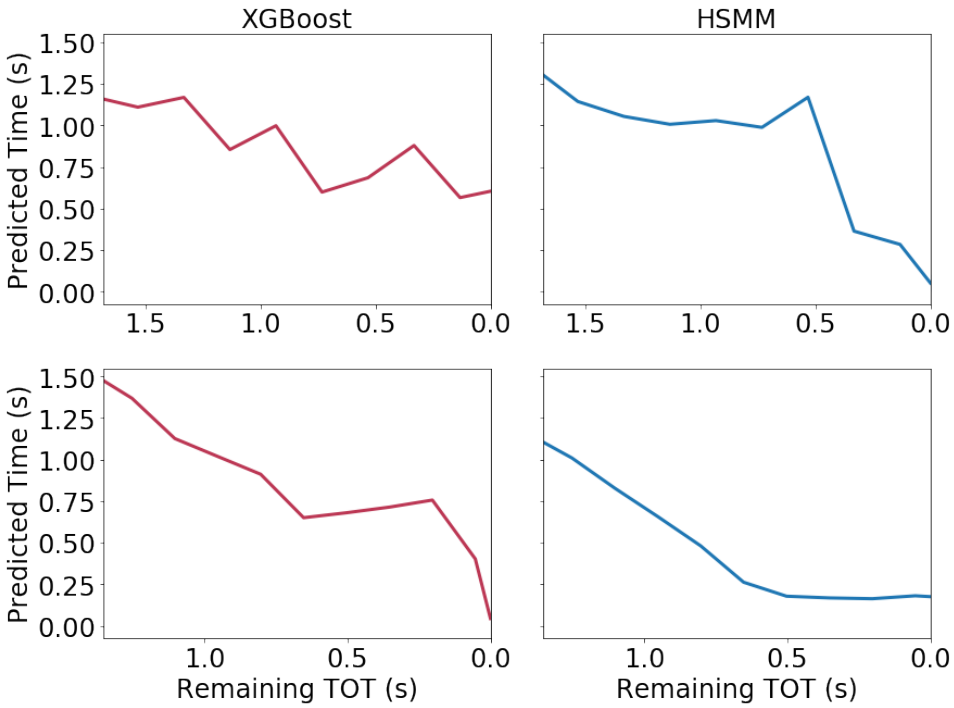


Figure 5.7: Two example predictions of XGboost and HSMM for the same input data (rows). In the top row, we have two examples of when the model perform poorly on prediction. Even though the initial predictions are quite poor, the HSMM is able to converge to the true TOT as more data is acquired. In the bottom row, we have two examples of where both models performed well by tracking the target well (slope = -1) and converging to the true TOT.

5.3.1 Driver Head Behavior and Take-Over Time

Analyzing the head pose data revealed two distinct driver behaviors at the time of the take-over: head facing the road before the take-over (automation deactivation) and head facing off the road during the take-over. This finding was similar to Vogelpohl *et al.*, who analyzed gaze behavior during take-over events and found that the majority of drivers took control before their gaze looked down the road. However, their study did not analyze if there was a difference in TOT between these two groups of driver behavior.

As expected, the head facing on the road before the take-over had a longer mean TOT of 1.74 seconds, when compared to the head facing off the road, which had a mean TOT of 1.33

seconds. This finding suggests that when the drivers head is facing the road before taking control of the vehicle, they can assess the situation and delay their take-over response to be appropriate to the perceived risk.

5.3.2 Predicting the Take-Over Time

Two take-over time models were trained for each of the driver behavior groups using the pre-processed head pose data. The combined models had an MAE 0.269 ± 0.259 seconds. When compared to the mean driver reaction time in emergency braking situations, which is 0.25 seconds, the prediction error of the model is on the same magnitude [25]. This shows that the model is able to predict the TOT to a reasonable accuracy on a continuous scale.

When analyzing the individual model performance, we found that the head facing on the road model ($MAE = 0.360 \pm 0.352$) had a worse prediction error than the head facing off the road model ($MAE = 0.246 \pm 0.200$). This suggests that there is a loss of information once the head is facing the road and the model performance degrades. Past research has shown that gaze data contains vital information on when a driver will resume control of the vehicle [40]. The head pose data is an approximate of gaze, so we lose detailed information at what exactly the driver is looking at (i.e. the dashboard, the road, the mirror, etc.) once the head is facing the road. Thus, it is possible to improve the performance of these models by integrating the gaze of the drive, which is a harder to capture data source in the driving environment. The model did has some limitations due to the driving simulator study size. In this study only a single take-over event, driving scenario, take-over alert, and secondary task was analyzed. Due to this, how the model generalizes to a wider variety of situations still needs to be explored. Finally, when compared to state-of-the-art machine learning regression models, our proposed TOT model was able to outperform or perform the same in prediction performance.

Chapter 6
ONLINE LEARNING

Past driving experience, individual driver differences, and automation trust have been shown to have an effect on the takeover of vehicle control in conditional driving automation [13, 28]. As a result, a driver's response to conditional driving automation are likely to evolve from continued use and as the technology changes [44]. Thus, it is important to adapt the model to account for these individual differences and changes in behavior over time to improve the model's performance. This chapter explores the effect of individual driver differences on the model prediction performance. The objective of this chapter is to address the second research question of this dissertation:

- *Question 2: Does accounting for the within and between driver differences improve the performance of the take-over time predictions?*

6.1 Analytical Methods

Two different time series clustering, using dynamic time warping (DTW) as the distance metric, and a online EM-algorithm were used to train individual models to evaluate the effect of individual driver differences on model performance.

6.1.1 Time Series Clustering

Due to the small data set size for each individual driver (12 take-over events) and how online algorithms require a large amount of data to accurately estimate model parameters [59], clustering was done to increase the data set size by grouping together similar drivers based on their head movement behavior. We used two different clustering methods that allowed for different assumptions about the drivers in our data set. The first clustering method

we used was hierarchical clustering, which allowed us to assume that every driver had other drivers that were similar to them and thus belonged to a cluster. Other clustering algorithms that used the same assumptions were considered, such as K-means. In the end hierarchical clustering was chosen for its ability to visually analyze the structure of the data using a dendrogram and for not having to specify the number of clusters before hand.

We decided to also use a second clustering method because the assumption that every driver in our data set had other drivers similar to them was too restrictive given our driver population size of 34. We used Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which allowed us to relax that assumption and assume that only some of the drivers were similar to each other and not every driver belonged to a cluster. DBSCAN accomplishes this by allowing for some of the drivers to be labled as outliers or noise and thus only clusters a subset of the drivers into groups.

Prior to clustering, we had two issues we had to address. First, we wanted to cluster the participants so all of their data ended up in one cluster and not split between multiple clusters. Then we had to deal with the variable length of each time series and how to cluster them together. We took the pre-processed head pose training data for each of the 34 participants and created an average time series using DTW Barycenter Averaging (DBA). This method averages multiple time series of varying length by iteratively updating an average time series that minimizes the squared distance [60]. The distance metric was computed using the DTW distance which is based on the Levenshtein distance (also called the edit distance), see Eq. 6.1 [51, 62].

$$D(A_i, B_j) = \delta(a_i, b_j) + min \begin{bmatrix} D(A_{i-1}, B_{j-1}) \\ D(A_i, B_{j-1}) \\ D(A_{i-1}, B_j) \end{bmatrix} \tag{6.1}$$

Where $A =< a_1, a_2, ..., a_{T_A} >$ and $B =< b_1, b_2, ..., b_{T_B} >$ are two time series of varying length, A_i is the subsequence ($< a_1, a_2, ..., a_i >$) of time series A, and let δ be the Euclidean distance between the elements of each time series. At a high level, DTW allows for a one-to-

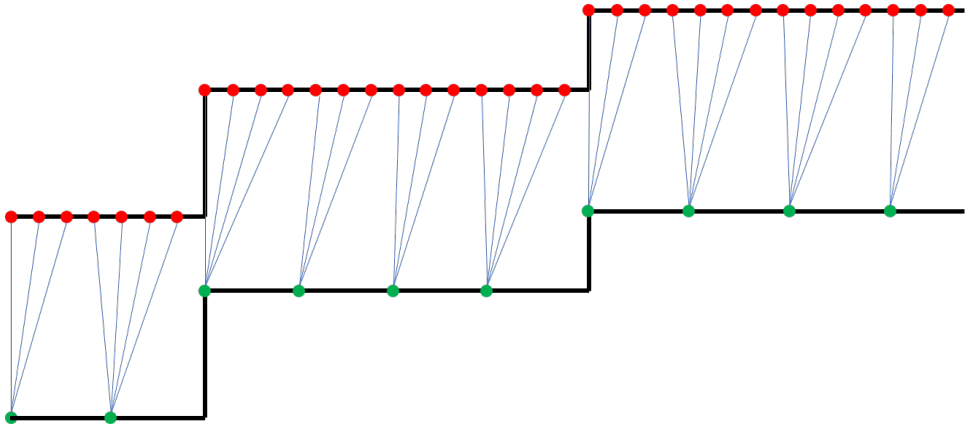


Figure 6.1: An example of how DTW is able to calculate distance between two time series of variable length. It is able to match elements of each time series together in a one-to-many relationship.

many matching of the elements of each time series to each other, see Fig. 6.1. This average time series was then used for clustering, which ensured that all of a participants data ends up in a single cluster.

Hierarchical clustering builds clusters from the bottom up by merging nearby clusters. The method starts with each observation in its own cluster and successively merges clusters together until all observations are in a single cluster at the top. The number of clusters can be determined beforehand by specifying the number of clusters you want or it can be determined by the algorithm through the use of a distance threshold. In our case we used the distance threshold and dendrogram plot to determine the number of clusters.

DBSCAN, unlike hierarchical clustering, does not require the number of clusters to be specified before hand. Instead it will discover the number of clusters based on two parameters: the neighborhood size (ϵ) and the minimum number of data points to create a cluster [15]. These 2 parameters control, in our case, the minimum number of drivers needed to create a cluster and how similar drivers need to be to join a cluster. The parameters for DBSCAN were tuned so that at least 50% of the participants were put into a cluster instead of identified as noise or outliers.

6.1.2 Mini Batch Online-Expectation Maximization Algorithm

For online learning, we implemented a mini-batch algorithm for online-EM proposed by Nguyen *et al.* [52]. The mini-batch algorithm processes a mini-batch (m) of time series observations and calculates the model parameters ($\bar{s}(x_i)$) that will be updated. Then the model parameters are updated for the i^{th} iteration of the online learning using Eq. 6.2.

$$s^{(i)} = (1 - \lambda_i)s^{(i-1)} + \lambda_i \bar{s}(x_i) \tag{6.2}$$

Where $s^{(i-1)}$ is the model parameter from the $i-1$ iteration and $lambda_i$ is the current iteration learning rate. The learning rate for the i^{th} iteration is calculated by raising the previous $(i-1)$ learning rate to the power of the learning decay (α), see Eq. 6.3. The update equations for the model parameters are the same as described in Ch. 5.1.2.

$$\lambda_i = \lambda_{i-1}^{\alpha} \tag{6.3}$$

There were 4 parameters for the online learning EM-algorithm: the learning rate (λ), the learning rate decay (α), the mini-batch size (m), and the mini-batch overlap ($m_{overlap}$). These parameters were tuned using a random search and cross validation (80/20). After tuning the parameters, the final models were trained on the full training data set.

6.2 Results

The procedure we followed to cluster the data was to first average the time series data on the training set of each participant to produce a single time series using DBA [60]. This produces what the average head movement during the take-over event would look like for each participant and allowed us to cluster all of the participants data into the same cluster. These average time series for each participant were then fed into an unsupervised clustering method (DBSCAN or hierarchical clustering with DTW as the distance metric) to produce the clusters [15]. For DBSCAN only the drivers that were identified in a cluster were used for model training and assessment. The clusters were then used to train individual models

using the online EM-algorithm. The prediction procedure on the test data set was the same procedure described in Ch. 5.2.

6.2.1 Hierarchical Clustering

For hierarchical clustering, the distance thresholds were determined by examining the dendrogram to analyze the linkage structure of the participants, see Fig. 6.2 and 6.5. The dendrogram showed many choices of distance thresholds that would lead to clusters consisting of 1 or 2 participants. These small clusters would have been counter productive to our original goal of increasing the data set size for training. Thus the distance threshold was determined to have the clusters as large as possible (the blue lines in Fig 6.2 6.5).

For the head off the road data set, the hierarchical clustering identified three clusters of size 22, 7, and 5 (clusters 1,2 and 3). The clusters were analyzed by plotting the average head movement to see if any patterns in driver responses could be identified, see Fig. 6.3. The two small clusters, clusters 2 and 3, had similar head movement behavior where drivers showed no head movement until the final moment when the driver started taking-over. The difference between the two clusters was that cluster 3 seemed to have an erratic head movement at the time of the take-over. Cluster 1 was the largest with 22 of the participants. The drivers in this cluster showed a behavior of some initial head movement like they are reacting to the TOR, but the drivers seemed to be distracted in trying to finish the secondary task before finally take-over control of the vehicle. The TOT of each cluster were then compared, see Fig. 6.4. Cluster 1 had the largest variation in TOT with a mean of 1.34 seconds and a standard deviation of 0.26. Cluster 2 had a mean TOT of 1.36 and standard deviation of .20. Cluster 3 had the smallest mean TOT of 1.25 seconds and standard deviation of 0.25.

For the head on the road data set, the hierarchical clustering identified two clusters of size 26 (cluster 2) and 7 (cluster 1). Cluster 2, the larger of the two clusters, exhibited a behavior of an initial head movement to the road followed by a second head movement to assess the take-over situation (see Fig. 6.6). This group had the longest mean TOT of 1.79 (SD=0.61) seconds, which indicates that the drivers took their time to take-over control and

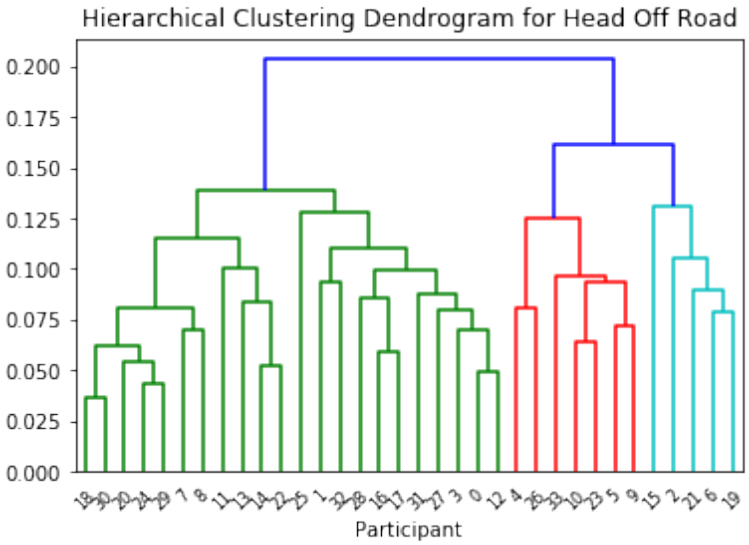


Figure 6.2: dendrogram of the identified clusters for the head off the road data set. We split the data using a distance threshold of 0.15 to divide the data into three clusters.

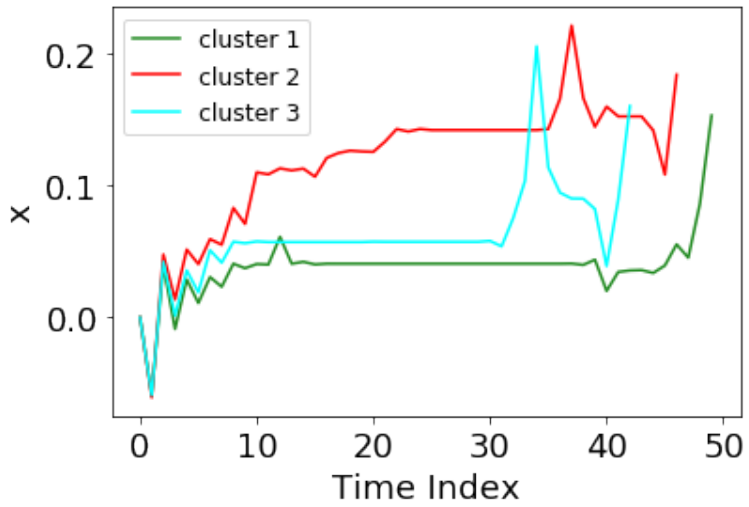


Figure 6.3: Using hierarchical clustering, three clusters were identified for the head off the road data set. The average head movement for each of these clusters was plotted.

felt safe to delay their response, see Fig. 6.7. The other cluster identified had a quicker head movement to the road and a shorter TOT mean TOT of 1.43 (SD=0.36) seconds. This group

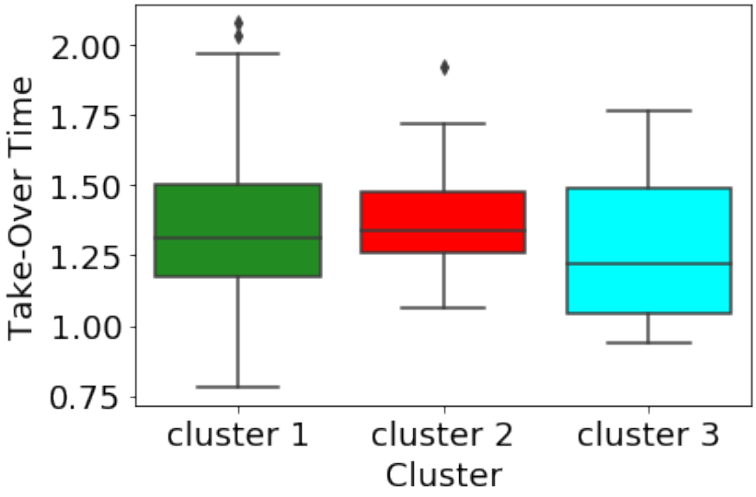


Figure 6.4: The take-over time for each cluster in the head off data set with hierarchical clustering.

of drivers seemed to assess the outside driving situation as a high crash risk and decided to take-over control of the vehicle immediately.

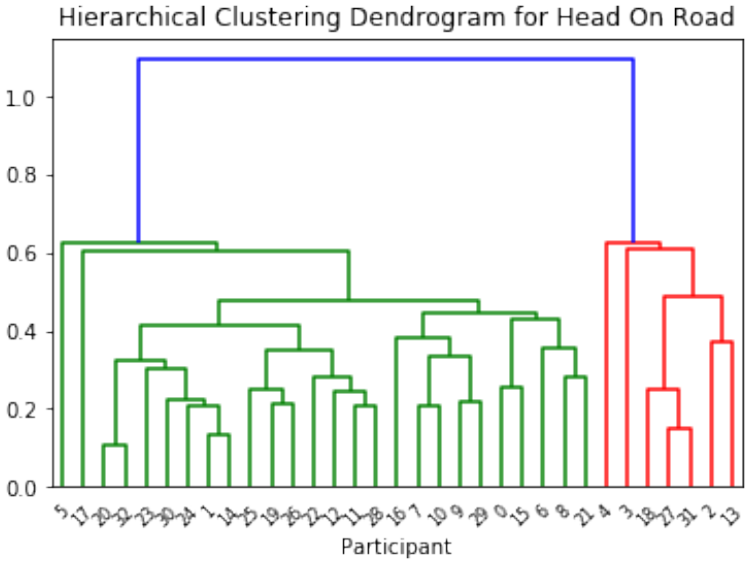


Figure 6.5: dendrogram of the identified clusters for the head on the road data set. We split the data using a distance threshold of 0.8 to divide the data into two clusters.

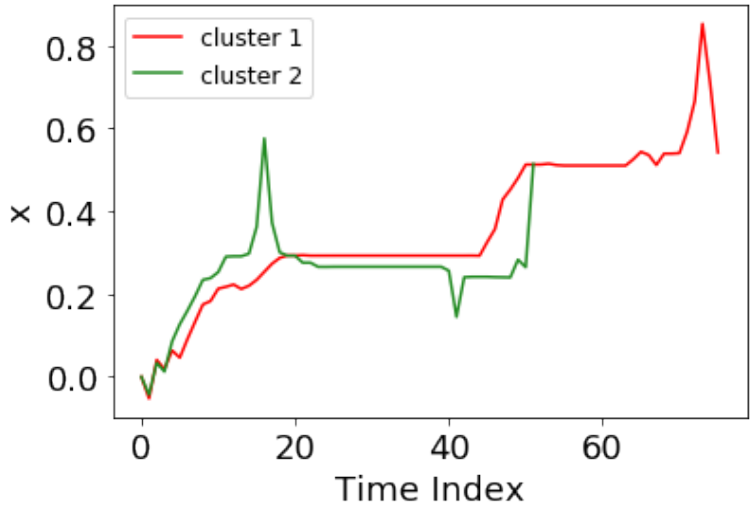


Figure 6.6: Using hierarchical clustering, two clusters were identified for the head on the road data set. The average head movement for each of these clusters was plotted.

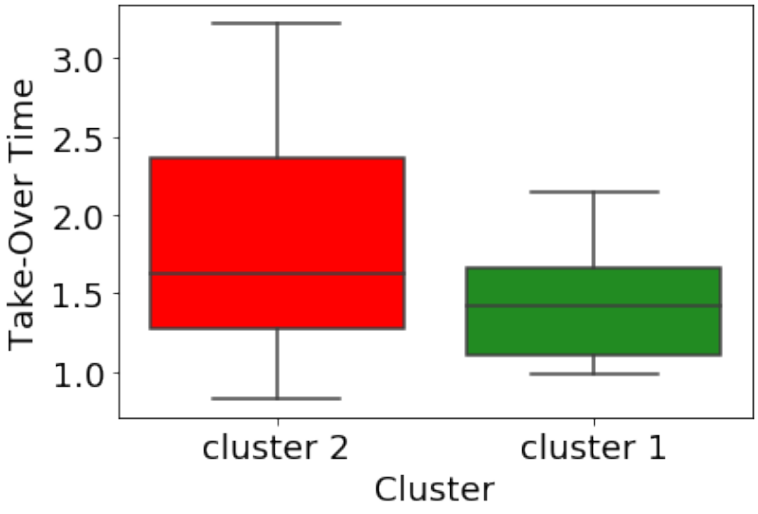


Figure 6.7: The take-over time for each cluster in the head on data set with hierarchical clustering.

6.2.2 DBSCAN

For DBSCAN, only a subset of participants were clustered together and the rest were identified as noise or outliers. Under the assumption that not all the drivers were alike, we took a conservative approach and tuned the parameters to cluster about 50% of the participants into groups.

For the head off the road data set, DBSCAN identified one clusters of size 18. The average head movement for this cluster was plotted in Fig. 6.10. This cluster was similar to the cluster 1 identified by hierarchical clustering in head movement behavior and the take-over time. The drivers were distracted by the secondary task for the majority of the take-over and only start to move their head to the road as they are taking-over control of the vehicle. When compared to cluster 1 of the hierarchical clustering, the mean TOT was similar at a mean of 1.30 seconds and a standard deviation of 0.24, see Fig 6.9.

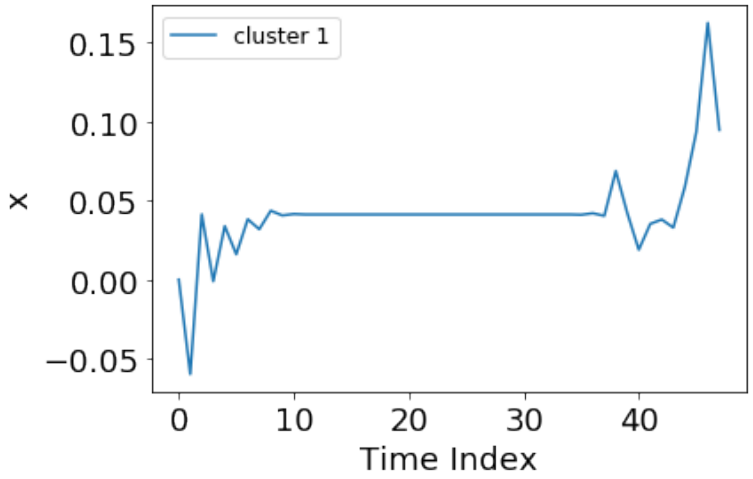


Figure 6.8: Using unsupervised clustering (DBSCAN with DTW), a single cluster was identified for the head off the road data set. The time series labeled as noise are not plotted.

For the head on the road data set, DBSCAN identified two clusters of size 9 and 7 just like hierarchical clustering. The average head movement for this cluster was plotted in Fig. 6.8. The two clusters had similar head movement behavior as the clusters identified in the

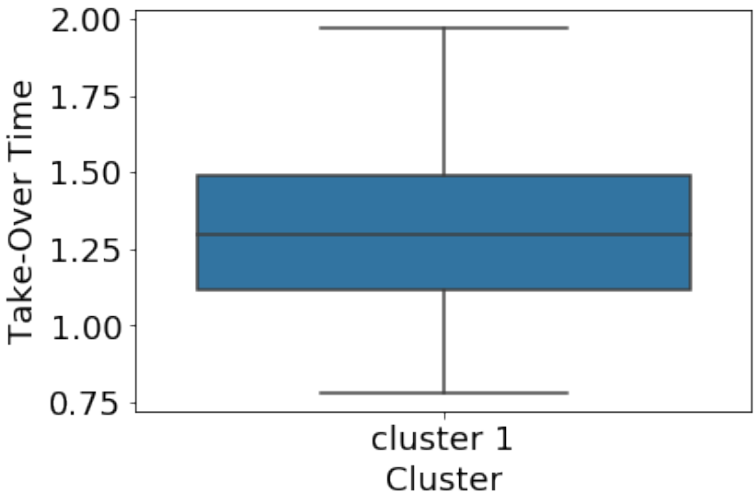


Figure 6.9: The take-over time for each cluster in the head off data set with DBSCAN.

hierarchical clustering. With cluster 1 exhibiting a low crash risk scenario where drivers have time to make two distinct head movements and delay their take-over (mean = 1.76 seconds and standard deviation = 0.52). Cluster 2 can be interpreted as a higher crash risk scenario where drivers move their head to the road and take-over immediately. The participants in this cluster had shorter TOT of mean of 1.60 seconds with a standard deviation of 0.54. The TOT were plotted in Fig. 6.11.

6.2.3 Online Learning

The 4 parameters for the online learning model were tuned using the training loss. The final parameter values used were: a mini-batch size of 8, a mini-batch overlap of 4, a learning rate of 0.01, and a learning rate decay of 0.99. For training, the data was shuffled then processed sequentially for mini-batch training. For model performance assessment, we are examining the difference between the offline models performance and the online models performance in MAE, see Eq. 6.4. This allows us to see if there is any improvement in the model performance of the online models when compared to the offline models.

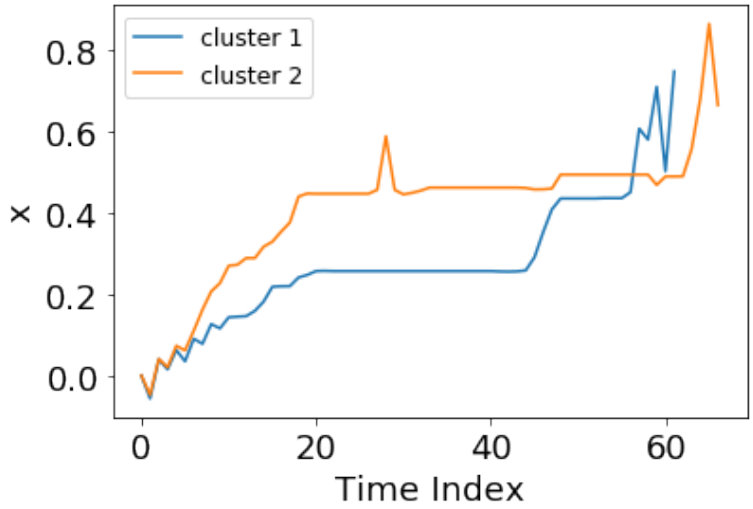


Figure 6.10: Using unsupervised clustering (DBSCAN with DTW), two clusters were identified for the head on the road data set. The time series labeled as noise are not plotted.

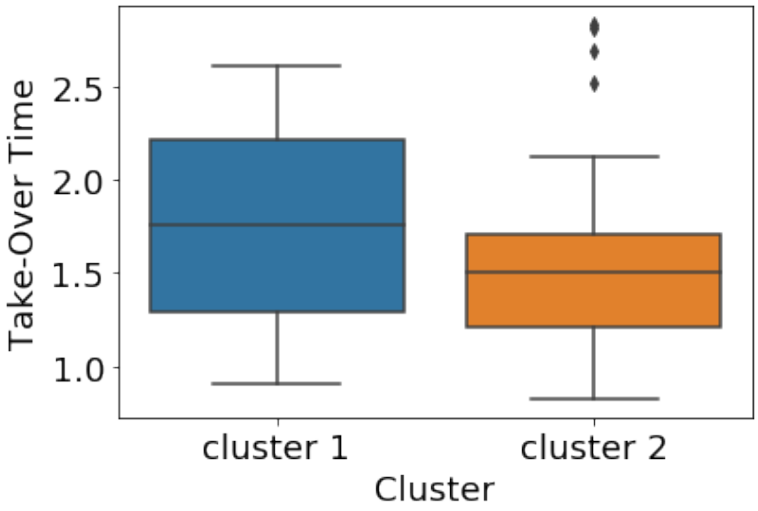


Figure 6.11: The take-over time for each cluster in the head on data set with DBSCAN.

$$l(x) = MAE_{offline}(x) - MAE_{online}(x) \tag{6.4}$$

For the hierarchical clustering, a total of 5 online models were trained (3 for head off

and 2 for head on). The model error (offline-online) for the head off the road model was $MAE : 0.0009 \pm 0.0285$ for the train data set and $MAE : -0.0030 \pm 0.0223$ for the test data set. For the head on the road model, the train error was $MAE : 0.0005 \pm 0.0176$ and the test error was $MAE : 0.0002 \pm 0.0223$. A one sample t-test was performed on the difference in prediction performance on the train data set and not the test data set due to the sample size. For the head off the road model there was no significant difference between the offline and online models (p-value: 0.1868, t-value: 1.3202). For the head on the road model there was no significant difference between the offline and online models (p-value: 0.1848, t-value: 1.3265).

For DBSCAN, a total of 3 online models were trained (1 for head off and 2 for head on). The model error (offline-online) for the head off the road model was $MAE : 0.0029 \pm 0.0774$ for the train data set and $MAE : -0.0118 \pm 0.0647$ for the test data set. For the head on the road model, the train error was $MAE : 0.0009 \pm 0.0175$ and the test error was $MAE : -0.0001 \pm 0.0180$. Both models were not significantly different when compared to the offline models (head off models: p-value: 0.1382, t-value: 1.4826; head on models: p-value: 0.3540, t-value: 0.9269).

6.3 Discussion

This chapter explored the effects of individual driver differences on model performance. The drivers were first clustered into groups using two different clustering methods with different assumptions about the drivers. Next, online models were trained for each of the clusters identified and their prediction performance was compared to the offline models. The research question this chapter set out to answer is inconclusive the effect of individual driver differences was not able to be analyzed due to the data set size. Instead, the results show that accounting for differences between groups of similar drivers, such as families, does not improve model performance. The effect of individual driver differences still needs to be explored.

The results indicate that there was no significant difference in model performance when

accounting for groups of similar individual drivers. This is an important finding because it shows that using pooled data from multiple participants will not improve the model performance. So manufacturers will have to account for differences at the individual level and can not assume that a family of drivers will all be the same. This analysis does have some limitations. The analysis was not able to train individual driver models, thus the findings do not provide insight on the effect of individual driver differences. By expanding the study to have more data for each individual driver, the effect of individual driver differences can be analyze and different online-learning methods can be applied. A second limitation is that only a single online learning algorithm was explored and there are many different algorithms, such as gradient descent or another online EM-algorithm, that can be applied to this framework. These alternate algorithms need to be analyzed to identify the one that is best suited for use in the vehicle environment.

Chapter 7
GENERAL CONCLUSIONS

This chapter provides an overall summary of the findings of this dissertation, possible application areas, and future research.

7.1 Overall Findings

The objective of this dissertation was to model the TOT of drivers in conditional automated driving, as well as examine the effect of individual driver differences on model performance. A driving simulator study using conditional driving automation was used to capture the take-over responses of drivers. Each driver was exposed to 12 take-over events over a 25 minutes drive. The TOT was modeled using a HSMM and head pose data. Finally, online-learning was used to explore the effect individual driver differences on model performance. The key findings from this dissertation are summarized below:

1. *Take-Over Time and Head Behavior.* Drivers whose head was facing the road before taking-over control of the vehicle tended to have a longer TOT, relative to participants whose head was facing off the road.

2. *Take-Over Time Modeling.* The TOT model was able to predict the true TOT to within an error of 0.269 seconds on average. The head off the road model tended to have better prediction performance than the head off the road model. When compared to other state of the art machine learning regression models, the HSMM was able to outperform or perform the same. The HSMM provided other beneficial model properties such as prediction uncertainty (confidence intervals) and the ability to handle variable input size.

3. *Head Pose Data.* The head position of the driver was shown to be highly correlated with TOT. However, the prediction performance of the head on the road model suggest that there is a loss of information once the driver's head is facing the road. As an unobtrusive and easy to measure data source, the head pose provides enough information to accurately model the TOT of drivers in conditional automated driving.

4. *Individual Driver Differences.* There were no statistical differences between the offline and online models with regard to MAE. The results suggest that updating the model parameters at a grouped level, for example a family of drivers or rental cars, does not improve the model. However, the data set size for each participant was too small to train individual models so the effect of individual driver differences still needs to be explored.

7.2 Contributions

As autonomous vehicles become increasingly available in the market, gaps in the technology will require the driver to remain in the loop in case they need to take-over. These transitions of vehicle control are periods of high crash risk where our model can be applied to increase driver safety and technology adoption. One major limitation in the literature is that there is a lack of regression prediction models that focus on the TOT using a simple data source. The majority of the research evaluates how much time is needed for a driver to take-over control, but depending on the situation and driver, the TB that is optimal will not always be the same. Hence, there is a gap in the knowledge on predictions frameworks that can model the TOT of a driver. Therefore, this dissertation developed a framework for predicting the TOT of a driver in conditional automated driving and explored the effects of individual driver differences.

The first research question focused on if the TOT can be predicted accurately on a continuous scale. The scientific contributions from this research question is the developed framework. This framework is flexible as the model can be adapted by changing the dis-

tributions used for modeling the observations and the state duration, the algorithm used for offline model learning, the input data used, and the algorithm used for online learning. So as the technology continues to change, the model can be adapted to accommodate these changes. This framework can also be applied to modeling a drivers behavior to any alert/warning/system cue and is not just applicable to TOT modeling.

The second research question focused on individual driver differences and their impact on the model performance. The scientific contributions from this research question is that even though individual driver differences has been shown to be correlated to the TOT, how to adapt prediction model to account for individual or even grouped differences is an area of research that still needs to be explored. The findings from this study have been disseminated into two academic journal articles and a conference preceding.

7.2.1 Publications

The results relating to development of a driver behavior model for a drivers response to an alert was evaluated on a manual driving data set. The results showed that the model was able to accurately predict when a driver would respond to an IVIS. These findings are in press to a special issue of *IEEE Transactions on Intelligent Transportation Systems* on *technologies for risk mitigation and support of impaired drivers* (Hwang, Boyle, & Banerjee, accepted). This corresponds to the *Model Framework* in Fig. 7.1.

The results relating to TOT predictions using the previously developed model showed that the model was able to accurately predict the TOT to within 0.269 seconds on average. The draft for this manuscript is completed and being prepared for submission to *IEEE Transactions on Intelligent Transportation Systems* (Hwang, Boyle, & Banerjee, in preparation). This corresponds to the *Simulator Study* and *Take-Over Time Model* in Fig. 7.1.

The results relating to the online learning of the TOT model show that there may be no benefits for updating the model at a grouped driver level. The results show there was no difference between then offline model and the online model performance, but the data set size was small and further investigation needs to be done. This manuscript is currently being

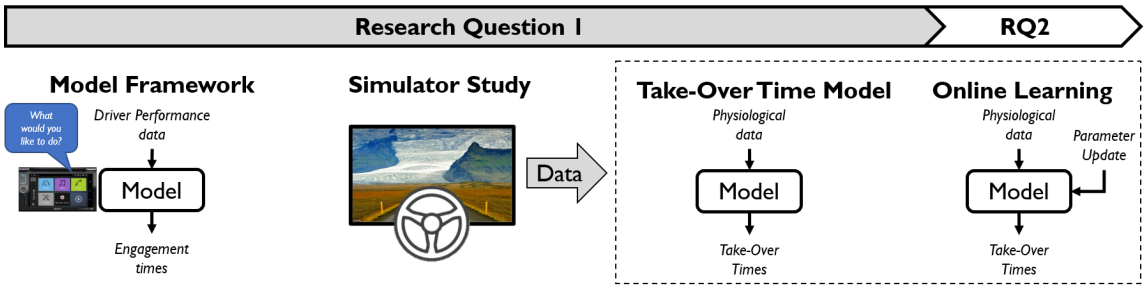


Figure 7.1: An overview of the proposed work and the relationship to the two research questions.

prepared for submission to the *64th Annual Meeting of the Human Factors and Ergonomics Society* (Hwang, Boyle, & Banerjee, in preparation). This corresponds to the *Simulator Study* and *Online Learning* in Fig. 7.1.

7.3 Limitations

A study limitation was the given time frame the data was collected (1 drive). Given more resources and time available, a larger study may show clear benefits of online learning for individual model performance and how the model performs in a variety of driving scenarios. The TOT model was shown to be able to accurately predict the TOT, but it is possible that the model is only effective in the scenario designed for this study. Thus, a larger study could provide more insight on how individual driver differences effect model performance and how well the model generalizes to different scenarios and real-world scenarios.

The data collected for this study came from a driving simulator. While driving simulator studies are beneficial in being able to control the independent variables, the risks drivers are willing to take are likely different when compared to an on-road study. That being said, driving simulator studies have been used numerous times to model driver behavior and this is a good first start to demonstrate the efficacy of our proposed framework.

The online learning results suggests that there is no difference to little difference in model performance when accounting for individual differences. This study exposed drivers to 12

take-over events. Even with this small data set, some model improvements were shown with online learning. By investigating the effect of individual differences, more significant improvements in model performance could be obtained. This will help to improve the model to adapt to the driver and increase the drivers trust in the system.

7.4 Future Research

The results of this study suggest that the TOT of a driver can be modeled accurately using time series data and a HSMM. This dissertation developed a framework for modeling and predicting the TOT of drivers in conditional driving automation and explored the effect of individual driver differences. However, a larger driving simulator study, various scenarios, or other data sources could help to improve the modeling framework.

More specifically, future research should consider a larger data set size (driving simulator or naturalistic) to analyze how the model generalizes to a variety of scenarios, real world data, or to conduct further analysis on the effects of individual driver differences. For example, sudden break events would have increased the crash risk and thus could effect the model performance. With a larger data set size different more factors can be manipulated to understand how the model generalizes to a wider variety of scenarios. Finally, other data sources should be investigated even though head pose data was shown to be highly correlated to the TOT. Recall the head pose data finding in Ch. 7.1, where once the head stops moving there is a loss of information about when the driver will take-over. Thus, there is still room for model improvement by investigating other data sources and how they can be integrated with the framework presented in this thesis. The last area of future work is taking this framework and implementing it in a conditional autonomous vehicle, such as the Tesla. This is important to understand how the information from this framework can be used to help improve driver safety. But before this framework can be applied in a real vehicle, an adequate online learning algorithm needs to be identified and implemented. As the driver continues to be use the technology and the manufacturers continues to update the technology, a method on updating the model to reflect these changes needs to be implemented for the framework

to be used in the applied setting.

BIBLIOGRAPHY

[1] Autopilot. https://www.tesla.com/autopilot. Accessed: 2019-04-25.

[2] Hardware and software for behavioral research eye-tracking. https://www.ergoneers.com/. Accessed: 2019-04-25.

[3] Sparkfun 9dof razor imu m0.

[4] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, jun 2018.

[5] G. Abe and J. Richardson. Alarm timing, trust and driver expectation for forward collision warning systems. *Applied Ergonomics*, 37(5):577 – 586, 2006.

[6] M. Azimi, P. Nasiopoulos, and R. Ward. Offline and online identification of hidden semi-markov models. *Signal Processing, IEEE Transactions on*, 53:2658 – 2663, 09 2005.

[7] A. Bietti, F. Bach, and A. Cont. An online em algorithm in hidden (semi-)markov models for audio segmentation and clustering. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1881–1885, April 2015.

[8] C. Braunagel, W. Rosenstiel, and E. Kasneci. Ready for take-over? a new driver assistance system for an automated classification of driver take-over readiness. *IEEE Intelligent Transportation Systems Magazine*, 9(4):10–22, winter 2017.

[9] M. Bueno, E. Dogan, F. Hadj Selem, E. Monacelli, S. Boverie, and A. Guillaume. How different mental workload levels affect the take-over control after automated driving. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2040–2045, Nov 2016.

[10] Bureau of Transportation Statistics. National transportation statistics 2018. Technical report, U.S. Department of Transportation, April 2018.

[11] Cambridge Systematics, Inc. Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. Technical report, Federal Highway Administration, September 2005.

[12] F. Cartella, J. Lemeire, L. Dimiccoli, and H. Sahli. Hidden semi-markov models for predictive maintenance. *Mathematical Problems in Engineering*, 2015(81):1–23, 12 2014.

[13] H. Clark and J. Feng. Age differences in the takeover of vehicle control and engagement in non-driving-related activities in simulated driving with conditional automation. *Accident Analysis Prevention*, 106:468 – 479, 2017.

[14] J. Dokic, B. Müller, and G. Meyer. European roadmap smart systems for automated driving. Technical report, European Technology Platform on Smart Systems Integration, April 2015.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[16] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *CVPR 2011*, pages 617–624, 2011.

[17] C. Flannagan, D. LeBlanc, S. Bogard, K. Nobukawa, P. Narayanaswamy, A. Leslie, R. Kiefer, M. Marchione, C. Beck, and K. Lobes. Large-scale field test of forward collision alert and lane departure warning systems. Technical report, feb 2016.

[18] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[19] R. Fu, H. Wang, and W. Zhao. Dynamic driver fatigue detection using hidden markov model in real driving condition. *Expert Systems With Applications*, 63:397–411, Jun 2016.

[20] T. Fuse and K. Kamiya. Statistical anomaly detection in human dynamics monitoring using a hierarchical dirichlet process hidden markov model. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3083–3092, Nov 2017.

[21] V. Gadepally, A. Krishnamurthy, and Ü. Özgüner. A framework for estimating driver decisions near intersections. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):637–646, Apr 2014.

[22] C. Gold, D. Damböck, L. Lorenz, and K.Bengler. "take over!" how long does it take to get the driver back into the loop? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 57(1):1938–1942, 2013.

[23] C. Gold, R.r Happee, and K. Bengler. Modeling take-over performance in level 3 conditionally automated vehicles. *Accident Analysis Prevention*, 116:3 – 13, 2018. Simulation of Traffic Safety in the Era of Advances in Technologies.

[24] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. Shinagawa, H. Ito, and T. Mikami. An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):942–953, Dec 2011.

[25] Marek Guzek, Marek Jaskiewicz, Rafał Jurecki, Zbigniew Lozia, and Piotr Zdanowicz. Driver reaction time under emergency breaking a car- research in the driving simulator. *Eksploatacja i Niezawodnosc - Maintenance and Reliability*, 14:295–301, 01 2012.

[26] D. W. Hansen and J. P. Hansen. Eye typing with common cameras. In *ETRA*, 2006.

[27] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[28] S. Hergeth, L. Lorenz, and J. F. Krems. Prior familiarization with takeover requests affects drivers' takeover performance and automation trust. *Human Factors*, 59(3):457–470, 2017. PMID: 27923886.

[29] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 07 2001.

[30] S. Iranmanesh, H. Mahjoub, H. Kazemi, and Y. Fallah. An adaptive forward collision warning framework design based on driver distraction. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–10, Dec In Press.

[31] T. Ito, A. Takata, and K. Oosawa. Time required for take-over from automated to manual driving. In *SAE Technical Paper*. SAE International, 04 2016.

[32] J.-A. Jang, K. Choi, and H. Cho. A fixed sensor-based intersection collision warning system in vulnerable line-of-sight and/or traffic-violation-prone environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1880–1890, Dec 2012.

[33] J. W. Jenness, L. N. Boyle, J. D. Lee, C.-C. Chang, V. Venkatraman, M. Gibson, K. E. Riegler, and D. Kellman. In-vehicle voice control interface performance evaluation final report. Technical report, 10 2016.

[34] M. KÃ, T. WeiÃÿgerber, L. Kalb, C. Blaschke, and M. Farid. Prediction of take-over time in highly automated driving by two psychometric tests. *DYNA*, 82:195 – 201, 10 2015.

[35] M. Körber, J. Radlmayr, and K. Bengler. Bayesian highest density intervals of take-over times for highly automated driving in different traffic densities. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 60(1):2009–2013, 2016.

[36] J. Q. Li and A. R. Barron. Mixture density estimation. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 279–285. MIT Press, 2000.

[37] Ning Li and Linda Ng Boyle. Allocation of driver attention for varying in-vehicle system modalities. *Human Factors*, 0(0):0018720819879585, 2019. PMID: 31887066.

[38] A. Lotz and S. Weissenberger. Predicting take-over times of truck drivers in conditional autonomous driving. In Neville Stanton, editor, *Advances in Human Aspects of Transportation*, pages 329–338, Cham, 2019. Springer International Publishing.

[39] T. Louw, R. Madigan, O. Carsten, and N. Merat. Were they in the loop during automated driving? links between visual attention and crash potential. *Injury Prevention*, 23:281–286, 09 2017.

[40] T. Louw and N. Merat. Are you in the loop? using gaze dispersion to understand driver visual attention during vehicle automation. *Transportation Research Part C: Emerging Technologies*, 76:35 – 50, 2017.

[41] Z. Lu, X. Coster, and J. de Winter. How much time do drivers need to obtain situation awareness? a laboratory-based study of automated driving. *Applied Ergonomics*, 60:293 – 304, 2017.

[42] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, Mar 2006.

[43] R. McCall, F. McGee, A. Mirnig, A. Meschtscherjakov, N. Louveton, T. Engel, and M. Tscheligi. A taxonomy of autonomous vehicle handover situations. *Transportation Research Part A: Policy and Practice*, 2018.

[44] E. Miller. *Behavioral Adaptions of Drivers to Autonomous Systems: Evaluating Intermediate and Carryover Effects*. PhD thesis, University of Washington, 2018. An optional note.

[45] B. Mok, M. Johns, K. J. Lee, David Miller, D. Sirkin, P. Ive, and W. Ju. Emergency, automation off: Unstructured transition timing for distracted drivers of automated vehicles. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2458–2464, Sep. 2015.

[46] B. Mok, M. Johns, D. Miller, and W. Ju. Tunneled in: Drivers with active secondary tasks need more time to transition from automation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2840–2844, New York, NY, USA, 2017. ACM.

[47] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626, April 2009.

[48] NADS. minisim, 2014.

[49] National Center for Statistics and Analysis. Traffic safety facts research notes. Technical Report DOT HS 812 456, National Highway Traffic Safety Administration, October 2017.

[50] National Center for Statistics and Analysis. Traffic safety facts 2016 data. Technical Report DOT HS 812 580, National Highway Traffic Safety Administration, September 2018.

[51] H. Ney and S. Ortmanns. Dynamic programming search for continuous speech recognition. *IEEE Signal Processing Magazine*, 16(5):64–83, 1999.

[52] Hieu Duy Nguyen, Florence Forbes, and Geoffrey J. McLachlan. Mini-batch learning of exponential family finite mixture models. *Stat. Comput.*, 30:731–748, 2020.

[53] NHTSA. Driver assistance technologies.

[54] NHTSA. Visual-manual nhtsa driver distraction guidelines for in-vehicle electronic devices. *Federal Register*, 78(81):24818–24890, 7 2013.

[55] J. Nilsson, P. Falcone, and J. Vinter. Safe transitions from automated to manual driving using driver controllability estimation. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1806–1816, Aug 2015.

[56] Society of Automotive Engineers. *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, jan 2014.

[57] C. Oh, J.-S. Oh, and S. G. Ritchie. Real-time hazardous traffic condition warning system: Framework and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 6(3):265–272, Sep 2005.

[58] N. Oliver and A. P. Pentland. Graphical models for driver behavior recognition in a smartcar. *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 7–12, 2000.

[59] Jimmy Olsson and Johan Westerborn. An efficient particle-based online em algorithm for general state-space models**this work is supported by the swedish research council, grant 2011-5577. *IFAC-PapersOnLine*, 48(28):963 – 968, 2015. 17th IFAC Symposium on System Identification SYSID 2015.

[60] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678 – 693, 2011.

[61] J. Radlmayr, C. Gold, L. Lorenz, M. Farid, and K. Bengler. How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 58(1):2063–2067, 2014.

[62] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[63] K. Tang, S. Zhu, Y. Xu, and F. Wang. Modeling drivers' dynamic decision-making behavior during the phase transition period: An analytical approach based on hidden markov model theory. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):206–214, Jan 2016.

[64] Marco Venturelli, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Deep head pose estimation from depth data for in-car automotive applications. In Hazem Wannous, Pietro Pala, Mohamed Daoudi, and Francisco Flórez-Revuelta, editors, *Understanding Human Activities Through 3D Sensors*, pages 74–85, Cham, 2018. Springer International Publishing.

[65] T. Vogelpohl, M. Kühn, T. Hummel, T. Gehlert, and M. Vollrath. Transitioning to manual driving requires additional time after automation deactivation. *Transportation Research Part F: Traffic Psychology and Behaviour*, 55:464 – 482, 2018.

[66] B. Wandtner, N. Schömig, and G. Schmidt. Effects of non-driving related task modalities on takeover performance in highly automated driving. *Human Factors*, 60(6):870–881, 2018. PMID: 29617161.

[67] X. Wang, M. Chen, M. Zhu, and P. Tremont. Development of a kinematic-based forward collision warning algorithm using an advanced driving simulator. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2583–2591, Sep 2016.

[68] M. S. Wogalter. Communication-human information processing (c-hip) model in forensic warning analysis. In S. Bagnara, R. Tartaglia, S. Albolino, T. Alexander, and Y. Fujita, editors, *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*, pages 761–769, Cham, 2019. Springer International Publishing.

[69] Michael S. Wogalter. *Handbook of warnings*. Lawrence Erlbaum, 2006.

[70] F. X. F. Ye, Y. Ma, and H. Qian. Estimate exponential memory decay in Hidden Markov Model and its applications. *arXiv e-prints*, page arXiv:1710.06078, Oct 2017.

[71] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215 – 243, 2010. Special Review Issue.

[72] R. B. Zadeh, M.i Ghatee, and H. R. Eftekhari. Three-phases smartphone-based warning system to protect vulnerable road users under fuzzy conditions. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2086–2098, Jul 2018.

[73] B. Zhang, J. de Winter, S. F. Varotto, R. Happee, and M. Martens. Determinants of take-over time from automated driving: A meta-analysis of 129 studies, 01 2019.

[74] L. Zhang, J. Wang, F. Yang, and K. Li. A quantification method of driver characteristics based on driver behavior questionnaire. In *2009 IEEE Intelligent Vehicles Symposium*, pages 616–620, June 2009.