1-1-2017

Toward Human-Like Automated Driving: Learning Spacing Profiles From Human Driving Data

Syed Ali
Wayne State University,

TOWARDS HUMAN-LIKE AUTOMATED DRIVING: LEARNING SPACING PROFILES FROM HUMAN DRIVING DATA

by

SYED ALI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2017

MAJOR: Computer Engineering

Approved By:

_____

Advisor Date

_____

_____

_____

DEDICATION

To my parents, wife and daughter

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my committee members; Dr. Abhilash Pandya, Dr. Le Yi Wang, Dr. Darin Ellis and Dr. Feng Lin for their guidance in this research. Special thanks to Dr. Abhilash Pandya for his support and guidance from the first day I walked into his lab through the past several years, and for his support in directing this research. Thanks to my colleagues at DENSO and at General Motors for all their support and encouragement. I wanted to acknowledge Dr. Shawn Hunt for his support and guidance in formulating the research problem. Special thanks to all the CARES team members for providing critical feedback on the research material.

Lastly, but most importantly, I wanted to thank my entire family for their unconditional support and encouragement through this journey. Every one of you had a positive impact and were a source of support throughout this journey.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION AND MOTIVATION

If the actions of an Automated Vehicle (AV) are not like human drivers, it can cause misinterpretation or confusion among other road users, which may lead to a hazardous situation. Consider for example Google's self-driving vehicle's accident that occurred with a municipal bus in 2016 [1]. Google's self-driving vehicle was planning to turn right at an upcoming intersection while travelling in the right turn lane when it sensed sandbags in its path ahead. To drive around the sandbags, it planned to merge into the lane to its left. While it was waiting to merge into the lane, it sensed a gap between a municipal bus and truck. It assumed that the municipal bus would slow down to let it merge in. With that assumption, the self-driving vehicle initiated moving towards the left lane. As an experienced human driver, we speed up and try to match the speed of the traffic in the lane we plan to merge in before we begin moving in. This self-driving vehicle on the other hand initiated its attempt to move into the lane where traffic was travelling at 15-20 mph while it was moving only at 2 mph. The bus driver assumed that since the self-driving vehicle was travelling at such slow speed, it was not going to attempt to merge in and the bus driver continued to drive without slowing down to create extra gap to let the self-driving vehicle merge in. The self-driving vehicle continued to drive towards the left lane at 2mph and ended up hitting the bus on the side. This was an example where the intention of the AV system as portrayed by its actions were not perceived as such by the human driver. The main reason for this confusion was self-driving actions were not human-like and because it made an incorrect assumption about the intention of the human driving the municipal bus. This example highlights

the need for self-driving vehicles to perform actions like human drivers to safely integrate with mixed traffic involving human drivers.

1.1 Motivation

Among the various challenges involved in integrating self-driving vehicles into our existing traffic, understanding the intentions and needs of human beings involved in this eco-system is arguably the most challenging aspect. The involvement of human beings in this eco-system ranges from being an occupant or driver of an automated vehicle, to being a driver or occupant of other vehicles that are around the Automated Vehicle or in the form of other road users such as pedestrians and cyclists. Realizing the intention and comfort zones/ranges of these humans entails understanding many interactions. Among some of the key interactions are (1) intentions of other human drivers around the automated vehicle, (2) an automated vehicle's ability to gauge how its intentions are being perceived by other human drivers and (3) to understand the expectations of the automated vehicle's driver and its passengers.

Consider the scenario shown in Figure 1 below of an automated vehicle planning to make a left turn. The driver of the AV would have his/her own perception of what a safe gap size for the oncoming traffic should be before initiating the lane change. In contrast to the driver's perception of safe gap size, the AV system may very well be able to make a left turn with a tight gap in between the two oncoming vehicles, but that would make the human driver in the AV uncomfortable. It may also cause discomfort and misjudgment by the drivers of other vehicles, leading to a hazardous situation.

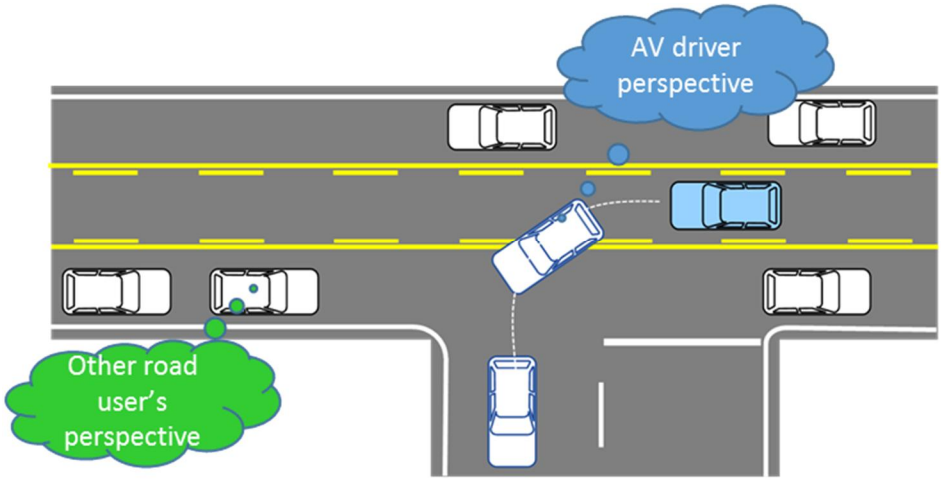


Figure 1: The automated vehicle (blue) intending to make a left turn. The driver of the automated vehicle has his own expectation of what gap size for oncoming traffic is safe before initiating left turn. Similarly, the driver of the vehicle on the oncoming lane has his own perception of what is a safe and acceptable gap for the AV to make a turn in front of it.

There are numerous other driving behaviors that human drivers demonstrate in their day-to-day driving which can directly be attributed to their perception of safe driving and what they find comfortable. Examples of human driving behavior that are closely related to a driver's perception of comfort and safety include, (1) increasing the distance from an oversized truck traveling in a neighboring lane (Figure 2, top), or (2) slowing down and increasing the space from stopped traffic in a neighboring lane during rush hour (Figure 2, bottom). With the tremendous onboard sensing capability inside a future AV [2, 3], it can sense the position and motion of the truck in the next lane to a high accuracy and drive with optimal (minimal) spacing from it. This optimality might be uncomfortable for the passengers of the AV or even the driver of the truck causing unnecessary stress or even accidents.

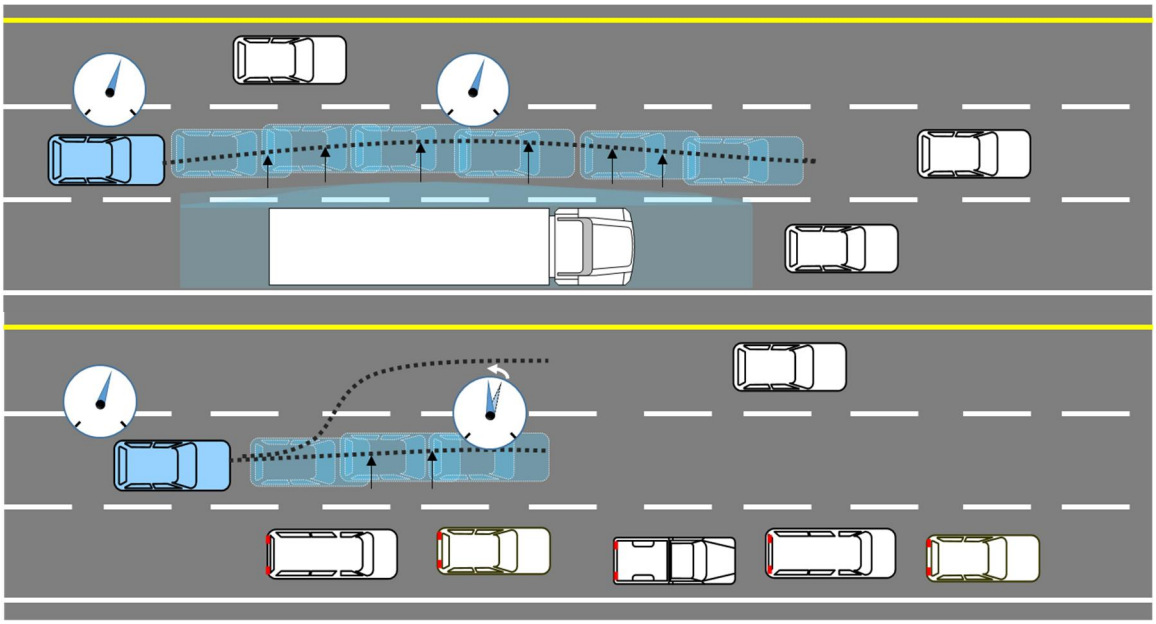


Figure 2: Vehicle increasing its space from an oversized truck traveling in neighboring lane (top). Stopped cars in the neighboring lane(bottom). Drivers in the neighboring lane typically tend to increase spacing and slow down or even switch lanes in anticipation of someone cutting in from the stopped lane.

Similarly driving at highway speeds next to a lane with stopped vehicles may be well within the capability limits of an AV, but that would not be perceived as safe and comfortable by the driver and occupants. Therefore, for AV technology to be accepted by its users and for these vehicles to safely integrate with traffic involving human drivers, it needs to drive within the context and understanding of human drivers. Learning these complex behaviors is challenging and there needs to be a methodology which allows these systems to learn from human demonstration. One of the main tenants of this thesis is to show a use case of such a methodology.

1.2 Research Aim

The driving use-case considered in this research is shown in Figure 3. Consider a two-lane road where a subject vehicle (referred to as the ego-vehicle, veh_{ego}) was travelling in the inner

lane. The ego-vehicle is shown in blue in this figure. A stationary vehicle (consider a broken-down vehicle) in the ego-vehicle's lane is shown in white. The driver of the ego-vehicle must then pass the stationary vehicle (hence forth referred to as the target vehicle, veh_{target}) by switching lanes temporarily to the outer lane. Can an AV system learn how a human driver performs this lane switch and the spacing he maintains while passing by the target vehicle?

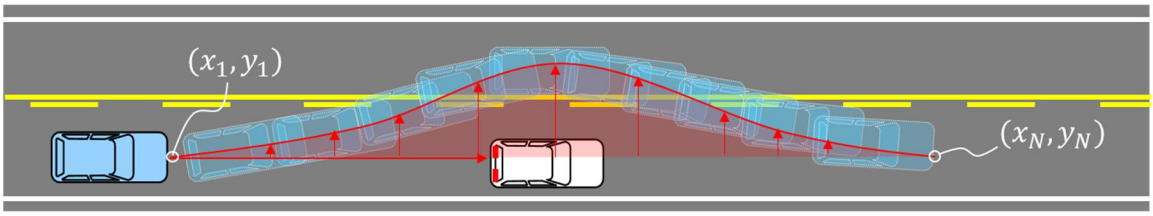


Figure 3: The driver of the ego-vehicle (blue) changing lane for a stationary vehicle(white) in lane. The overall spacing profile consisting of lateral spacing along the travel direction that the ego-vehicle kept from the target vehicle is shown in red. This research aimed to learn the spacing profile of the ego-vehicle driver from the stationary target vehicle.

As the driver of the ego-vehicle approaches this stationary vehicle and drives around it, he/she will keep a certain longitudinal and lateral spacing from this vehicle. This spacing profile that the driver of ego-vehicle would maintain from this target vehicle could vary depending on several factors such as the speed of approach to veh_{target}, road conditions (e.g. slippery or dry), road geometry (e.g. curved versus straight or undulated road) and the driver's perception among others.

The aim of this research was to learn the spacing profile that the ego-vehicle driver maintained from veh_{target}, while driving at highway speeds and traveling on straight as well as curved roads. This is a simple example for which a learning strategy/methodology is developed here. This methodology could potentially be used to study other variations for instance, non-

pass stationary target vehicles, different times of the day, different weather conditions, driving on roads with more than two lanes or on roads with different elevations.

The specific questions for this research were:

1. What are the factors that affect a human driver's spacing profile as he/she approaches and drives around a stationary target vehicle?
2. Can a model be learned from human driving data that represents his/her spacing profile from other vehicles?
3. What would be the difference between the spacing model that has been learned from human driving data as compared to computing the spacing profile as an optimization problem?

1.3 Specific Research Objectives

The first objective of this research was to develop the research platform that would facilitate studying a driver's spacing profile. This research platform consisted of hardware components, data capture components, data analysis software, and an extensive simulation environment. The hardware platform consisted of instrumenting vehicles to capture the pose and motion profile of each of the vehicles involved in this study. An extensive software toolchain was also prepared to process and analyze the simulated and real-world driving data. To better understand the driving environment and the type of data that would be helpful in studying this research problem, a simulation environment was created. This simulation environment consisted of a) a road track, b) ego vehicle and c) target vehicles. The simulated road track was created to match the actual road that was intended to be used in this research. The vehicle dynamics model

fidelity was chosen to represent the aspects of the driving dynamics that were of primary interest for this research such as vehicle lateral acceleration.

The second objective of this research was to formulate a Learning from Demonstration (LfD) methodology what would allow the identification of a set of features that represent an ego-vehicle driver's spacing profile. This involved capturing real-world test-track driving data, processing all the different data attributes and finding a small subset of features that showed strong correlation to driver's spacing profile. This was achieved by employing feature selection techniques in machine learning.

The third objective of this research was to teach a model that will be able to provide the spacing profile from a stationary target vehicle that matches an ego-vehicle's driver spacing profile within a pre-defined statistical tolerance. Developing the model involved exploring several different machine learning algorithms to learn this non-linear regression profile.

The fourth and final objective of this research was to compare the output from the learned model with the output from a constrained optimization formulation of the same problem, i.e. representing the driver's spacing profile. Researchers in the past attempted to formulate the representation of human-like driving as an optimization problem. In this fourth objective, an objective function was formulated using cost criteria that partially consisted of features that were learned in this research. Additional constraint criteria were defined to represent the driving environment similar to that used in the learning problem. Finally, a comparison of the spacing profile generated by the two approaches (i.e. spacing profile predicted by the learned model and the optimized spacing profile) resulting from the same driving scenarios are presented.

1.4 Dissertation Outline

The outline of the remainder of this dissertation is as following: Chapter 2: presents the background to automated driving with a focus on human-like automated driving. Three of the most relevant research results from the literature search are discussed in this chapter. Chapter 3: details the test platform, including the software and hardware platforms, along with information concerning the software tools to analyze the data. Chapter 4 discusses the driving task that was used in this research, formulation of the spacing profile as a learning problem, the data pre-processing pipeline and determining the models that represent the driver's spacing preferences. The spacing profile was also formulated as a constrained optimization problem and presented in this chapter. This formulation of spacing profile as an optimization problem was done to compare it with the learning approach. The results of spacing profile as a constrained optimization problem compared to spacing profile by the learned model are presented in Chapter 5. The performance of the learned models against unseen data is also presented in this chapter. Lastly, the summary of the research contributions of this thesis, along with the limitations of this research and future work are discussed in Chapter 6.

CHAPTER 2: BACKGROUND

This chapter presents an overview of the literature search that was conducted as part of the background research in order to better understand the gaps that existed in the learning of human-like driving. In the later part of this chapter, a review of some of the key literature related to this research is discussed.

2.1 Relevant Literature

The topic of human-like automated driving touches upon many subject areas within the field of automated driving or ground robotics in general. The figure below (Figure 4) shows some of the subject areas that were considered as part of the literature search to gain broader understanding regarding what had been accomplished in the development of automated driving systems that considered the human aspect, including human comfort needs and acting in human-like ways.

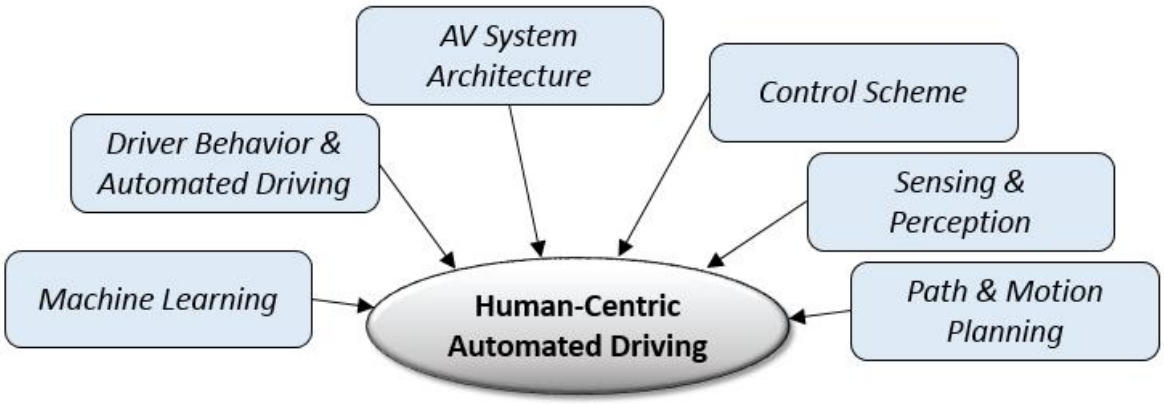


Figure 4: Relevant technical areas in understanding Human-Centric Automated Driving.

2.2 Automated Vehicle System Overview

The various systems within an Automated Vehicle can be broadly categorized based on their functionality into sensing and perception, planning behavior and actuator control (Figure 5).

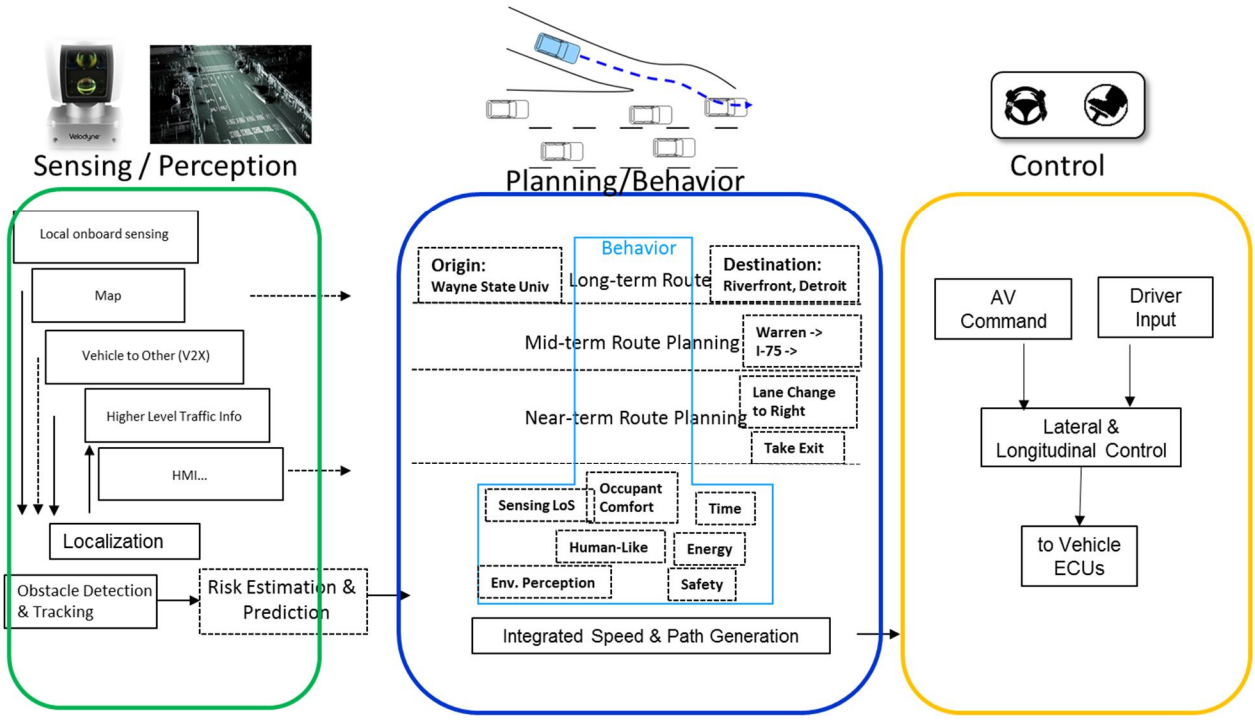


Figure 5: A simplified overview of Automated Vehicle Architecture. The Sensing and Perception system feeds in environmental information such as sensed objects, their perceived intended path, range of sensing and quality of sensed information among others. The Planning and Behavior system plans a path considering several aspects within its understanding such as human-like driving and passes on trajectory information to the control system. The control system provides commands for the lateral and longitudinal control of the vehicle by considering motion commands from the AV system as well as the inputs from the driver.

Sensing and Perception

The role of the sensing and perception system as the name suggests is to sense the environment internal to and outside of the AV. A large selection of sensing modalities is generally employed within the sensing and perception category including a) monocular and stereo-vision

camera systems, b) long-, medium- and short-range Radio Detection And Ranging (RADAR) sensors, c) ultrasonic sensors, d) Light Imaging Detection And Ranging (LIDAR) sensors e) Global Positioning System (GPS) receivers f) Inertial Measurement Units (IMU) of various accuracy and resolution g) Vehicle-to-other (V2X) communication devices, and h) High-Definition (HD) maps among others.

The sensing of the environment internal to an AV would typically consist of occupant monitoring systems that monitor the state and attention of the driver. The driver monitoring systems sense and perceive various actions of the driver using sensors such as IR cameras that are mounted on top of the steering wheel facing the driver [4, 5]. Some of the common applications of driver monitoring systems include detecting a drowsy or inattentive driver among others.

The sensing of the environment external to an AV can include a wide variety of sensors depending on the application. The long-range RADARS for example have been widely used in the Adaptive Cruise Control (ACC) feature [6, 7, 8] to sense other vehicles up to 180 meters in front of the AV [9]. The camera systems on the other hand have been widely applied to almost all aspects of external environment sensing; from sensing lane marks in front of and around the vehicle, to detecting traffic signs, road-side curbs, other vehicles, bicyclists and pedestrians among others [10]. The LIDAR sensors have been considered as a key enabler for automated driving [11]. A combination of camera with LIDARS and of RADAR with LIDARS have been used in applications ranging from vehicle and pedestrian detection to detecting drivable areas [12, 13, 14].

Another technology that complements all the other sensors discussed thus far and that facilitates functions within the AV such as localization and path planning among others, is HD-maps [15]. These highly detailed maps of the road and surrounding infrastructure allow AVs to see beyond their local onboard sensing. They also support other sensing functions such as traffic light identification and sign recognition in challenging urban scenarios. In [3], there is a detailed overview of sensing and perception; from different sensors, to multi-sensor data fusion and functions and applications that use these data.

Planning and Behavior

The role of the planning and behavior system within the AV framework is to provide a feasible trajectory towards the destination that is collision-free, meets a certain energy and/or time criteria and considers occupant comfort among other criteria. For a planning system to plan a path, the driving environment as perceived by the sensing and perception system needs to be represented in a way that a planner can search through to find the best path. Among the environment representation techniques is the cost map [16] (shown in Figure 6), where the environment is represented in terms of cost for each cell by considering the pose and motion information of the obstacles around the AV.

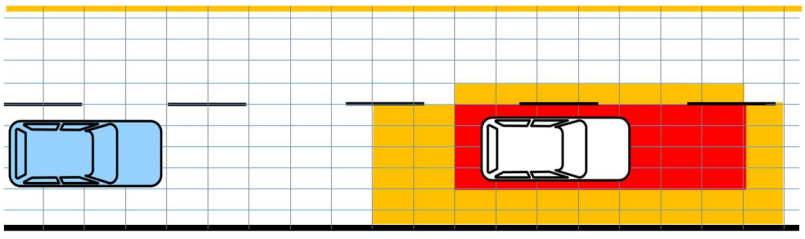


Figure 6: Driving space represented as a cost map. The cells that were predicted to be occupied based on the detection and motion estimation of preceding vehicle represented with a high cost (shown in red), whereas cells with low probability of being occupied by preceding vehicle based on its perceived motion represented with a lower cost (shown in orange).

Among the various planning techniques is separating the search space into near, mind and far areas, where lattice of varying resolution/dimension is employed in each of the areas [17]. In the lattice space near the AV for example, the AV is represented in a higher dimensional-space such as its position (x, y), velocities $(\dot{x}, \dot{y})$, accelerations $(\ddot{x}, \ddot{y})$ and heading (θ). As we move to mid-area lattice, the search space dimension can be reduced to position and velocities, while far area can be reduced further down to position only. This allows searching through a reduced-dimension search space and creating a long-term plan that fulfils short and long-term driving goals. A detailed overview of the state-of-the-art in planning techniques is provided in [16].

Actuation Control

The role of the actuation control system is to minimize the error to the desired trajectory provided by the planning system. The actuation control system is responsible for generating steering and acceleration commands to their respective Electronic Control Units (ECU) in a vehicle. Several control algorithms ranging from classical to modern control techniques have been applied to control the vehicle to a desired trajectory [18, 19].

2.3 Human-Like Path Formulation and Path Following

There are several systems within an automated vehicle that play an important role in mimicking human driving; from perception of the environment to control of the actuators. The human-like aspects within the perception of the environment includes being able to sense and estimate the intention of other road users in a similar manner to human drivers. The planning system takes the perceived environment, driver's preference, and energy and time criteria among other factors to formulate a safe trajectory that would allow the AV to traverse through the available free space around the automated vehicle to reach to its destination. The planning

system is a critical contributor and in some sense the primary responsible system towards human-like driving since it formulates the path and subsequently a trajectory that the AV should follow.

The trajectory formulated by the planning system is then used by the control system to determine its error at a certain look-ahead distance (assuming preview-based control scheme), which is used to determine what its next set of lateral and longitudinal control commands should be to minimize the error along the desired trajectory. The control system may additionally encompass factors of human-like control such as considering the lateral and longitudinal forces that the vehicle may experience while travelling on the desired path. The following sections, 2.3.1 and 2.3.2 provide an overview of some of the relevant literature where research is focused on human comfort and human-like driving. The research work presented in section 2.3.1 focuses on safety and comfort within the control system framework, whereas the work referenced in section 2.3.2 can be generalized as part of a planning system within an automated vehicle.

2.3.1 Trajectory as an Optimization Problem

In [20] Dongwook et al. focused on providing safe and comfortable automated driving when unexpected obstacles, such as a broken-down vehicle, were experienced in the lane while driving on inner-city streets. Their problem definition can be generalized as given a lead vehicle's path and other vehicles in the vicinity of the subject vehicle, what would be the series of steering control commands that would provide a safe and comfortable drive. A LIDAR sensor mounted around the front of the vehicle was used to detect the leading vehicle and other obstacles in the driving environment. The Extended Kalman Filter (EKF) was used to track the leading vehicle's position and motion data. The state vector data representing the lead vehicle data is given by:

$$x = \left[P_{x,rel}, P_{y,rel}, \theta_{rel}, v_{pre}, \gamma_{pre}, a_{pre}, \dot{\gamma_{pre}}\right] \quad [2\text{-}1]$$

And the measurement vector is given by

$$z = \left[P_{x,rel}, P_{y,rel}, \theta_{rel}\right]^T \quad [2\text{-}2]$$

In designing a steering controller, the authors had the objective of following the lead vehicle, while keeping a safe distance from obstacles and keeping lateral acceleration within certain pre-defined comfort and safety limits. They used Model Predictive Control (MPC) to design this steering controller. The state equation for the ego-vehicle was represented as,

$$\xi = \begin{bmatrix}\beta & \psi & \dot{\psi} & X & Y\end{bmatrix} \quad [2\text{-}3]$$

where, β represents vehicle side-slip, ψ heading, $\dot{\psi}$ yaw rate and (X, Y) the position of the ego-vehicle. The bicycle model (see APPENDIX B: Lateral Vehicle Dynamics Model) was used to represent the vehicle state as:

$$\frac{d}{dt}\begin{bmatrix}\beta\\ \psi\\ \dot{\psi}\\ X\\ Y\end{bmatrix} = \begin{bmatrix}\dfrac{-2(C_f + C_r)}{mv_x}\beta + \left(\dfrac{-2(l_f C_f - l_r C_r)}{m{v_x}^2} - 1\right)\dot{\psi} + \dfrac{2C_f}{mv_x}\delta_f\\ \dot{\psi}\\ \dfrac{-2(l_f C_f - l_r C_r)}{I_z}\beta + \dfrac{-2(l_f{}^2 C_f - l_r{}^2 C_r)}{I_z v_x}\psi + \dfrac{2l_f C_f}{I_z}\delta_f\\ v_x \cos\psi - v_x \tan\beta \sin\psi\\ v_x \sin\psi + v_x \tan\beta \cos\psi\end{bmatrix} \quad [2\text{-}4]$$

The vehicle trajectory based on control input (u) as the steering angle δ_f was generated from the output variables in the inertial reference frame was:

$$\eta = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}.\xi = \begin{bmatrix}X & Y\end{bmatrix}^T \quad [2\text{-}5]$$

Given current stateξ_0, MPC computes the optimal control sequence $[u_k]$ by solving the following optimization problem:

$$Minimize\, J_{mpc}\left(\left[\xi_k\right]_{k=0}^{N}, [u_k]\right)$$

for, $(i) \quad \xi_{k+1} - f^{dt}(\xi_k, u_k) = 0$

$$(ii) \qquad d_{min} - d_{safety} \geq 0$$

$$(iii) \qquad |\dot{\psi}| - \frac{a_{ymax}}{v_x} \leq 0$$

where N was the look-ahead horizon.

The cost function for the above optimization problem was given as;

$$J_a = \phi(\Delta\eta_N) + \sum_{k=0}^{N-1}\left[L(\Delta\eta_k, u_k) + \lambda_k^{\tau}\{\xi_{k+1} - f^{dt}(\xi_k, u_k)\}\right. \qquad [2\text{-}6]$$

$$\left. + \mu_{obs} G_{obs}(d_{min}) + \mu_{ay} G_{ay}(\dot{\psi})\right]$$

where $G_{obs}(d_{min})$ and $G_{ay}(\dot{\psi})$ are penalty functions (given below) on the control input and yaw rate and μ_{obs}, μ_{ay} are their weighting functions respectively.

$$G_{obs}(d_{min}) = \begin{bmatrix} \frac{1}{2}\left|\frac{1}{d_{min}} - \frac{1}{d_{safety}}\right|^2 & if\ \frac{1}{d_{min}} - \frac{1}{d_{safety}} \geq 0 \\ 0 & else \end{bmatrix}, \qquad [2\text{-}7]$$

$$G_{ay}(\dot{\psi}) = \begin{bmatrix} \frac{1}{2}\left|\dot{\psi}| - \frac{a_{y,sat}}{v_x}\right|^2 & if\ |\dot{\psi}| - \frac{a_{y,sat}}{v_x} > 0 \\ 0 & else \end{bmatrix}$$

The figure below (Figure 7) shows the lead vehicle (red) reference trajectory $\eta_{ref,i}{}_{i=0}^{N}$. The resulting optimized ego-vehicle trajectory ($\eta_i{}_{i=0}^{N}$) based on the above-defined criteria is shown as the black solid-line.

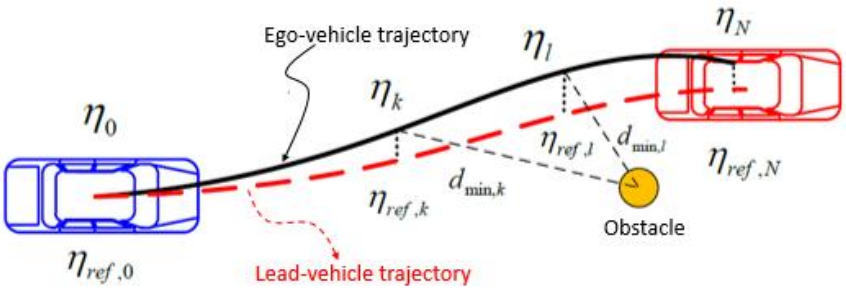


Figure 7: Lead vehicle reference path (red-dashed line) and optimized ego-vehicle path (black-solid line). The distance from obstacle and lateral acceleration were costs considered while trying to follow the lead-vehicle's path.

The authors discussed the simulated results for inner-city slow-speed driving with the optimized trajectory, but did not discuss the feasibility of this approach for high-speed driving. Among other challenges at higher speeds, it would be difficult to extend this technique in deriving an optimized trajectory over a longer preview distance in real-time. A longer preview distance is essential for safely and comfortably controlling a vehicle travelling at higher speeds. Additionally, a more detailed model would be needed to appropriately represent the motion and distribution of forces at higher speeds and in highly dynamic situations. That is possibly why a real-time implementation of such a strategy that uses detailed vehicle dynamics has not been seen in the real-world environment [16]. The authors also used fixed parameters such as distance to the obstacle for the optimization criteria. At slower speeds, the human occupants may be comfortable in driving closer to an obstacle, but using the same distance from an obstacle can be unsafe and discomforting at higher speeds. Some other metrics such as headway and time-to-collision (3.2.4) are more appropriate for such use as they account for distance with respect to the speed difference between a host and target vehicle.

2.3.2 Learning Techniques Applied to Driving and Path Selection

Learning from Demonstration (LfD) [21], also known as robot programming by demonstration or apprenticeship learning, is a class within machine learning that has been gaining attention in recent years in respect of developing automated vehicle systems. In control systems, such learning-based systems can be referred to as Adaptive Optimal Control systems [22]. The idea in such a learning-based system is that the system tries to learn necessary actions given a state by observing a teacher perform the action, shown as teacher demonstrations (*D*) in figure below. The learning system derives policies (π) from the observed state of the system and the action demonstrated by the teacher. This learned policy set is then used to perform actions (*a'*) based on world observations (*z'*).

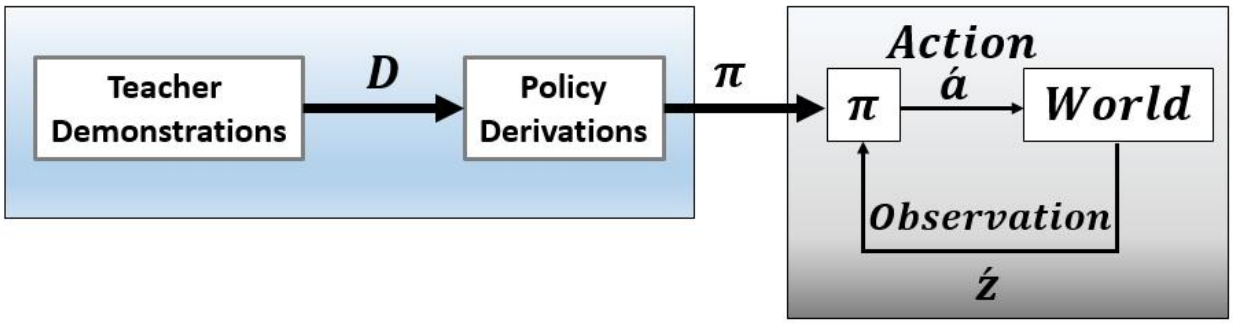


Figure 8:Learning from demonstration framework [21]. The policy set π is derived by observing teacher demonstrations for a state set. The learned policy is applied to perform actions (a') by observing state pair (z') of the world.

Such learning-based systems can prove to be very effective, especially in cases where a system is represented by a complex set of equations that have many parameters. In contrast to the typical hand tuning of these parameters by an expert, the system can learn these parameters based on observations. Learning system parameters, in contrast to hand tuning has been proven to provide better results compared to hand tuning [23].

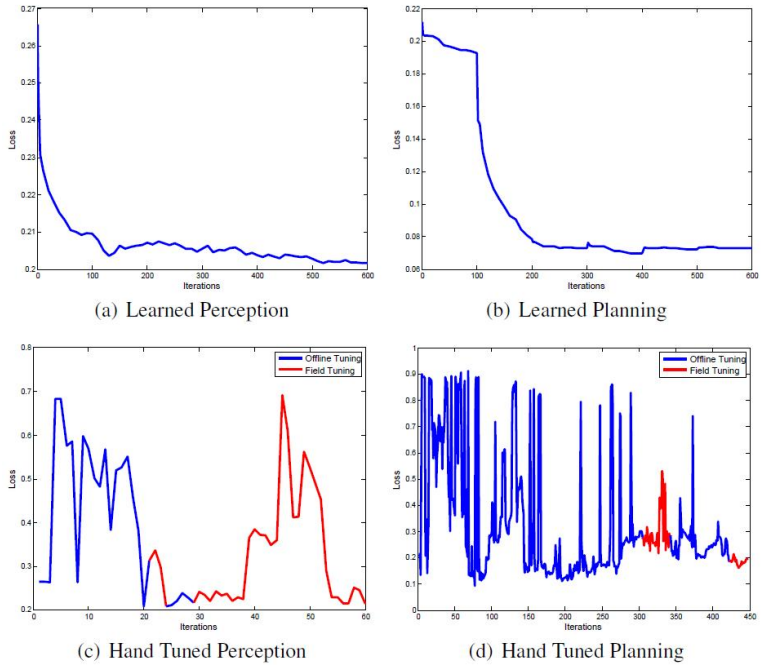


Figure 9: Hand-tuning versus learning system. System performance comparison for perception and planning using hand tuning vs. learning [23]

Tianyu et al. [24]investigated the creation of a human-like motion profile using LfD techniques. They collected driving data from two drivers while driving through an increasing level of tight turns; highway, urban and parking lot driving in a traffic-free environment. The block diagram in Figure 10 shows the process of generating a human-like reference trajectory. The path was derived using smoothness and curvature reduction criteria, whereas the speed model was parametrized and the parameters were learned from human driving data.

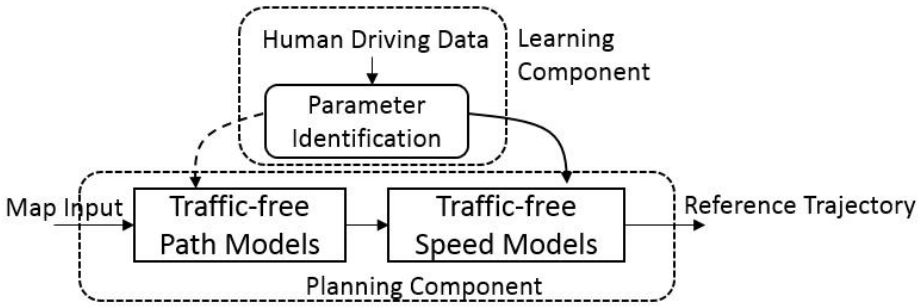


Figure 10: Human-like driving in a traffic-free environment consisting of a learning component that learns parameters from human driving data and the planning component that generates a trajectory based on learned parameters [24]

The path model was derived as following the center line with the following form:

$$\{s_i, [x_i, y_i], \theta_i, \kappa_i\},$$

Where $s_i, [x_i, y_i], \theta_i, \kappa_i$ represented the longitudinal station along the lane, global position, heading and path curvature respectively. The bumps created between two waypoints were smoothened independently for $x_i \; and \; y_i$ yielding

$$\{s_i, [x_i{}^*, y_i{}^*], \theta_i{}^*, \kappa_i{}^*\}$$

While taking sharp turns (high curvature), drivers tend to reduce the maximum curvature by using the available part of then lane, which was represented as o_i, lateral nudge (offset) to the centerline given by:

$$\{o_i\} = \underset{\{o_i\}}{argmin} \sum \left| \frac{p_i - p_{i-1}}{|p_i - p_{i-1}|} - \frac{p_{i+1} - p_i}{|p_{i+1} - p_i|} \right|, \; where \; p_i = \begin{bmatrix} x_i^* - o_i . \sin\theta_i^* \\ y_i^* + o_i . \cos\theta_i^* \end{bmatrix}$$

The Lavenberg-Marquardt algorithm was used to solve the optimization problem giving the reference path.

In terms of speed model, Tianyu et al. observed that human drivers tend to slow down ahead of approaching a tight curve. To model this cautious human behavior of slowing down ahead of the tight curve, maintaining the low speed while crossing the maximum curvature on

the turn and then increasing speed again, the authors proposed following speed model for tight curves:

$$\{v_i\} = M^{tight}(\{\kappa_i\}, \boldsymbol{Q})$$

$$\text{where, } \boldsymbol{Q} = \left[v_{min}, s_\Delta, l_v, \tilde{a}_{lon}, \tilde{d}_{lon}\right]^T$$

The following figure shows the speed profile model for tight curve and the parameters used in the model shape profile $\boldsymbol{Q}$*; where* v_{min} represents the minimum velocity that drivers reach before reaching the maximum curvature and maintain that over path length l_v. The parameters $\tilde{a}_{lon}$and $\tilde{d}_{lon}$ represent longitudinal acceleration and deceleration respectively. The parameters of this model were learned from human driving data with the following least-square formulation:

$$\widehat{\boldsymbol{Q}} = \underset{Q}{argmin} \left\| \{v_i^{human}\} - M^{tight}(\{\kappa_i\}, \boldsymbol{Q}) \right\|$$

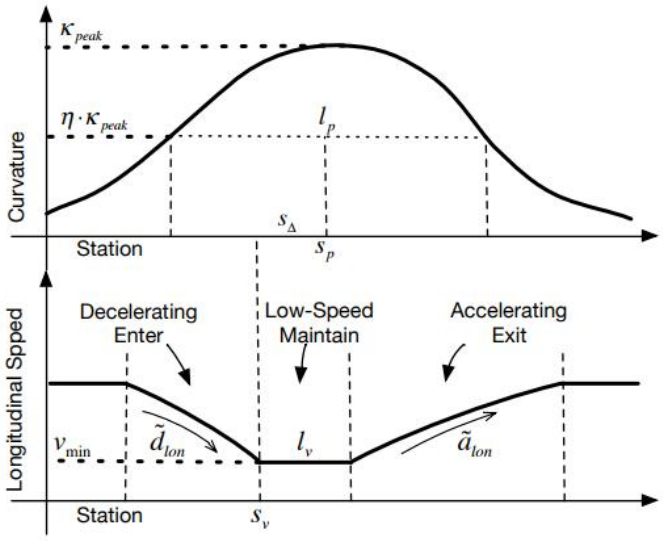


Figure 11: Speed model for a tight curve M^{tight} [24]

In order to improve smoothness of speed profile generated by the above model, the interpreted jerk $\dot{v}$ was constrained below j_{lon}iteratively as; $|\dot{v}| \leq j_{lon}$.

The comparison of the final path and speed profile generated by the path and speed model is shown in Figure 12. Although the results show good fit to the driver data, these models were created for traffic-free environment and no consideration to driving style in environment with other road users. Additionally, this study did not consider any influence of surrounding road structure either.

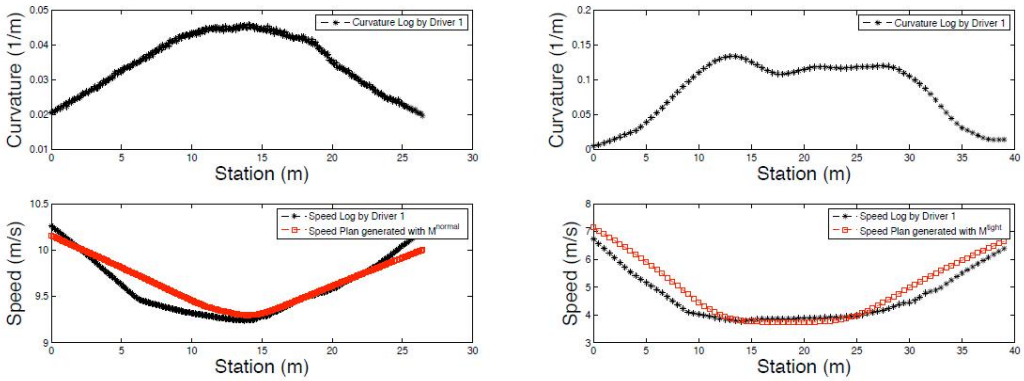


Figure 12: Driver speed profile versus learned speed profile while traveling path of a given curvature [24]

Markus et al. [25] employed the LfD approach to learn a driving style using human driving data. The authors modeled lane change driving style as a cost function consisting of longitudinal and lateral accelerations, longitudinal and lateral jerk, curvature, desired speed, center of lane, distance to obstacle center and following distance. A feature-based inverse reinforcement learning was used to find the model parameters that reflected the human's driving style the best. The learned model was then used to compute the trajectories while driving in automated mode. Figure 13 shows the generated trajectory from the learned policy (red) plotted against the observed trajectory (blue). Although any statistical measure for the quality of fit of the generated trajectory versus the observed trajectory was not provided in this study, the authors concluded that the trajectories generated from the model did not fit the observed trajectory perfectly.

Another conclusion presented by the authors was that the observations did not meet the defined optimality criteria for any cost function that was a linear combination of the feature set.

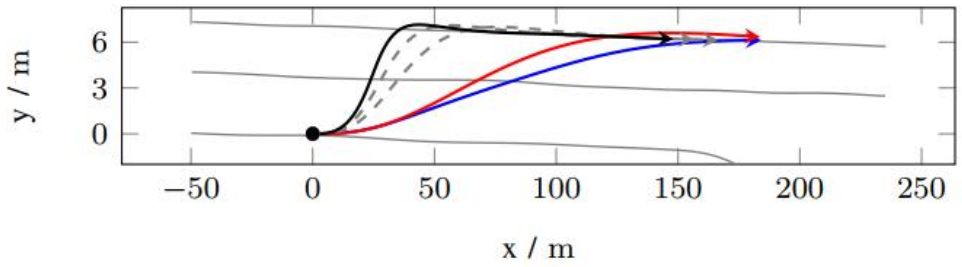


Figure 13:Observed (blue) versus generated trajectory (red) from learning policy [25].

Although the resulting trajectory did not fit the demonstrated trajectory well, the authors noted that they could learn the magnitude of the features that contribute to the driver's comfort. The feature set employed by the authors was quite extensive but lacked two components that were identified to be important in our research; 1) lateral distance of the host vehicle from the obstacle along the travel direction and 2) relative velocity/acceleration with respect to the obstacle. The relative velocity could have also been captured if the authors had considered time-to-collision information. It is our assumption that perhaps the lack of these two features in their model resulted in the performance deficiencies in the generated trajectory.

2.4 Chapter Summary

This chapter provided an overview of the automated vehicle system architecture. The automated vehicle system was presented as a combination of sensing and perception, planning and actuation control systems. Some key technologies and methodologies within each of these systems were discussed. Some of the literature relevant to this research was also discussed. Two primary areas of research that focused on human comfort or human-like driving were detailed. At first, a review of the trajectory optimization technique was presented where the authors

considered safety and comfort criteria to find an optimized path given a lead vehicle path. After that a review of some of the recent work that employed learning from demonstration techniques in learning a human-like driving style was given. While other researchers have attempted to derive a human-like path preference using learning from demonstration techniques, these studies were lacking in respect of a) providing human-like selection in a driving environment that included traffic or b) they were not able to fit the observed trajectories.

In the next chapter, the research platform is presented, consisting of both the hardware and software components. The physical test environment, consisting of the test track and the instrumentation that was used to gather vehicle data is explained. This is followed by an overview of the simulation platform and the software environment prepared to process and analyze the captured data.

CHAPTER 3: RESEARCH ENABLING TEST PLATFORM: TOOLS AND TECHNIQUES

The research hardware and software test platforms, along with software tools created to support the various stages of this research are discussed here. The discussion includes the, test environment, instrumentation used, data gathering techniques, pre-processing and the simulation environment. These tools allow the gathering and processing of data and are discussed in detail within this chapter.

3.1 Physical Test Platform

3.1.1 Test Track

For this study, we wanted to use a road that would simulate areal-world driving environment, while being able to park target vehicles in a lane without creating a safety risk. An automotive proving ground is ideal for such a requirement. Fowlerville proving grounds by FT Techno of America, FTTA [26] is one such facility that has tracks which have been built to city roads and highway specifications. This facility has therefore been extensively used within the automotive industry for the development and testing of active safety such as ACC, Lane Keep Assist (LKA) and Collision Mitigation Braking (CMB) features among others.

From the several tracks that this facility offers, the “Oval Track” was used for this study (Figure 14). This was a 4.827km (approximately 3 miles) long track, consisting of two lanes each with a width of 3.6m. As the name suggests, this track was a closed oval shaped course, consisting of straightway on the East and West side of the track. Each of the straightways were roughly 1370m in length. The road curve on the South end of the track had a radius of 300m, whereas the North end of the curve had a decreasing radius from 457m to 300m. These radii are similar

to those that can be encountered in highway driving, thereby allowing comfortable continuous driving at highway speeds of 112 kph (70mph).

Figure 14: Two lane oval test track at Fowlerville Proving grounds of FT Techno of America

3.1.2 Test Track Map

Real-Time Kinematic (explained in section 3.1.3.2) quality positional information for points along the track was used to extract the road geometry. Consider a point $p_{(1,1)}$which represents the first point, while point $p_{(1,n)}$represents the last point *n* on the inner lane line as show in Figure 15. Similarly points $p_{(2,1)}$, $p_{(3,1)}$ and points $p_{(2,n)}$, $p_{(3,n)}$represent the first and last points of the center lane line and outer lane line, respectively. The lane line mark point $p_{(i,j)}{}_{i=1:3}^{j=1:n}$ consisted of latitude (ϕ), longitude (λ) and radius (R) along the road travel direction.

$$p_{(i,j)}{}_{i=1:3}^{j=1:n} := \{\phi_i^j, \lambda_i^j, R_i^j\}_{i=1:3}^{j=1:n}$$

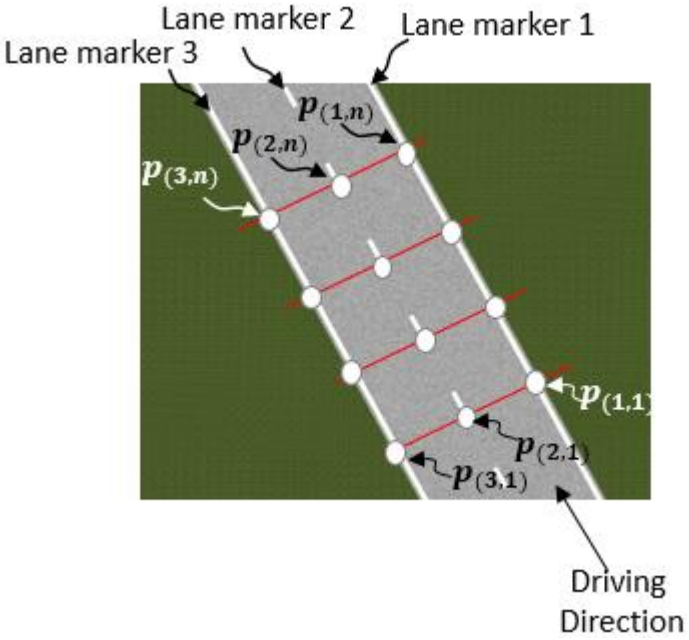


Figure 15: Lane attribute definition along the road. The inner solid line of the right lane (along the travel direction) – labelled line 1, center dashed line – lane 2 and the outer solid line of the left lane – line 3. Attributes measured along the road cross section at each of the lane line marks were latitude, longitude, altitude and radius.

The position measurements were converted from geodetic coordinates to Universal Transverse Mercator (UTM) coordinates. UTM is an international location reference system that provides an approximated two-dimensional representation of the three-dimensional surface of the Earth. This coordinate system converts the Earth's surface into a two-dimensional plane by dividing the Earth's surface into 60 equally spaced vertical planes or zones. The UTM coordinate system consists of vertically running parallel lines, referred to as Easting and horizontally running parallel lines, referred to as Northing, with each parallel line equally spaced 1000m apart forming $1000m^2$ squares.

The planar approximation formula to convert from geodetic coordinates to UTM is as follows:

$$x = (\phi - \phi_{origin}) * r * \cos(\phi) * {}^{pi}/_{180} \qquad [3\text{-}1]$$

$$y = (\lambda - \lambda_{origin}) * r * {}^{pi}/_{180}$$

where, $\phi_{origin}, \lambda_{origin}$ and r are the latitude, longitude in degrees, minutes and seconds and the earth's radius in meters. ϕ and λ are the measurement latitude and longitude points that are to be converted to *x* and *y* respectively.

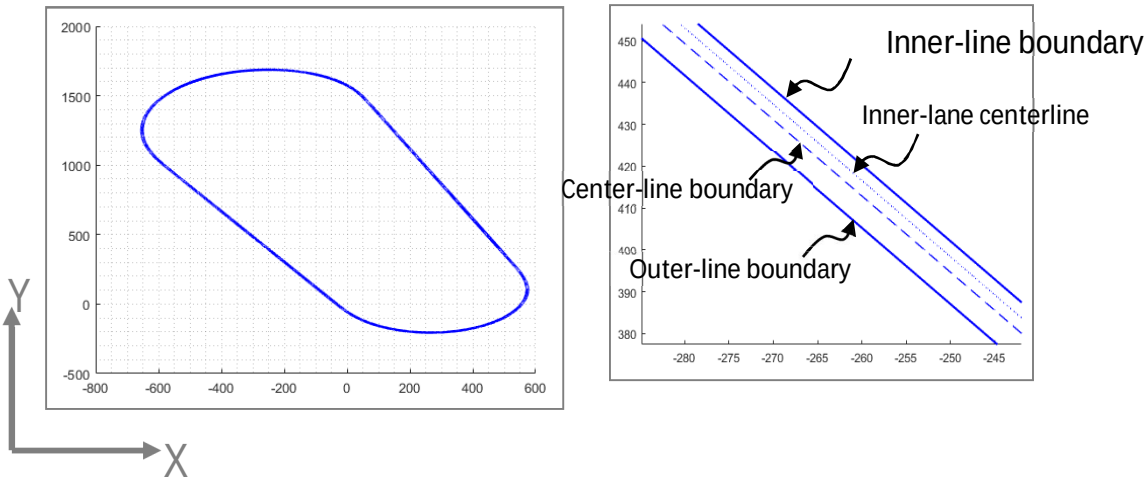


Figure 16: GPS map of the test track lanes shown in UTM coordinates, with zoomed in view showing the lane lines (right)

The plot above (*Figure 16*, left) shows test track lane lines plotted using UTM coordinates of the test track points $p_{(i,j)}{}_{i=1:3}^{j=1:n}$. The image on the right shows a zoomed-in view of a segment of the North-West part of the straightway. In addition to the three lane line boundaries, an additional line, labelled the inner-lane centerline is shown. The points for this line were derived by calculating the mid-point between the respective points on the inner-line and center-line. This line was selected as the lateral offset origin, with the inner-line, center-line and outer-line boundaries at -1.8 m, 1.8 m and 5.4 m respectively.

$$Inner\ lane\ Centerline\ (ICL) = \left[\frac{p_{(1x,i)} + p_{(2x,i)}}{2}, \frac{p_{(1y,i)} + p_{(2y,i)}}{2}\right]_{i=1}^{n} \quad [3\text{-}2]$$

where $n = \frac{trackLength}{incrementDistance}$, with *trackLength* of 4827m and *incrementDistance* of 0.5 m.

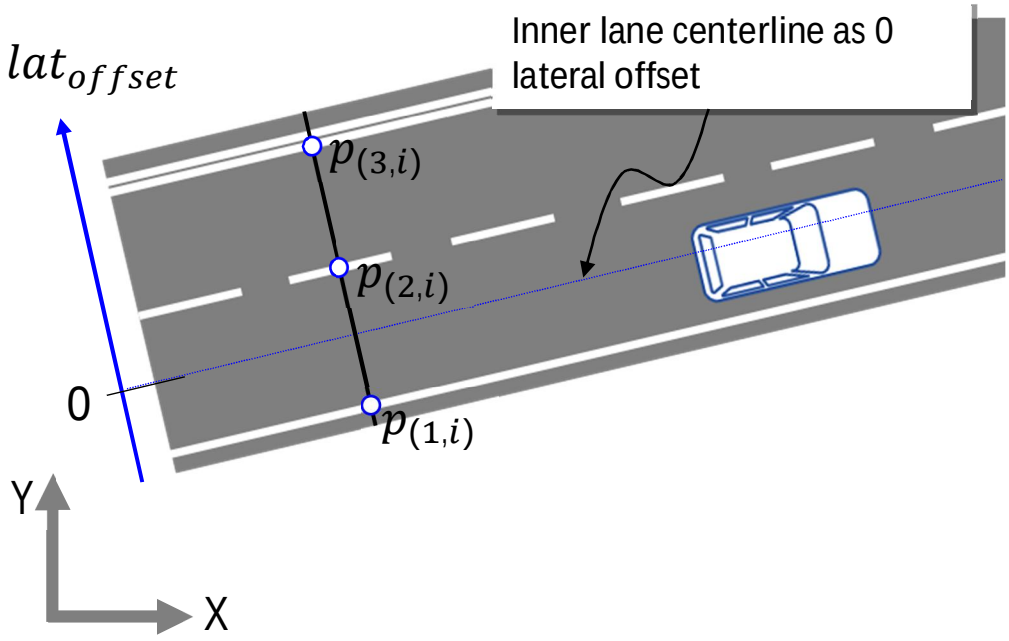


Figure 17: Center point between p(1,i) and p(2,i), also defined as the inner-lane centerline used as the zero lateral offset.

3.1.3 Acquiring Vehicle Pose and Motion Information

To minimize the errors in pose and motion information, as well as to have data periodicity at a high rate, Oxford RT 3003 [27], an off-the-shelf integrated GPS/INS system was selected. Table 1 lists the specifications for the RT-3000 family. There are several modes listed under position specification namely SPS, SPBAS, DGPS and RTK. Among these, Real Time Kinematic (RTK) can provide centimeter-level positional accuracy. The following section briefly explains how the

position information is resolved based on the information received from satellites. The RTK system helps to enhance the position measurement accuracy.

Models	Position (m)	Velocity (km/h)	Pitch / Roll (deg)	Heading (deg)	Slip angle at 50 km/h (deg)	Dual antenna
RT3100	1.8/SPS 0.6/SBAS 0.4/DGPS	0.1	0.05	0.1	0.2	No
RT3102	1.8/SPS 0.6/SBAS 0.4/DGPS	0.1	0.05	0.1	0.2	Yes
RT3002	1.5/SPS 0.6/SBAS 0.4/DGPS 0.01/RTK	0.05	0.03	0.1	0.15	No
RT3003	1.5/SPS 0.6/SBAS 0.4/DGPS 0.01/RTK	0.05	0.03	0.1	0.15	Yes

Table 1: *RT3000 Family Specification. RT3003 was used for pose and motion information in this study*

3.1.3.1 Global Navigation Satellite System (GNSS)

As a general concept, GNSS-based systems calculate the distance between a receiver and satellite by calculating the time it takes for the signal to travel from the satellites to the receiver. With distance measurement from a single satellite, we know that the receiver antenna is somewhere on a sphere with a radius equal to the range measurement, ρ_r^1. If we now also know our distance measurement from a second satellite ρ_r^2, our receiver is at either of the two locations, *x* or *y*, where the two radii are ρ_r^1 and ρ_r^2. If we know that we are at a distance ρ_r^3 from a third satellite, we can only be at one position *x*, where each of the three spheres meet (see Figure 18). This process of resolving the position is known as “triangulation”. This simple explanation facilitates our discussion regarding how RTK operates. In understanding GNSS based

position information, it is critical to understand that in actual practice, we need at least four satellites to be able to resolve our position accurately. Details of how this information is derived is given in [28, 29]. The references also provide an in-depth understanding of GNS systems.

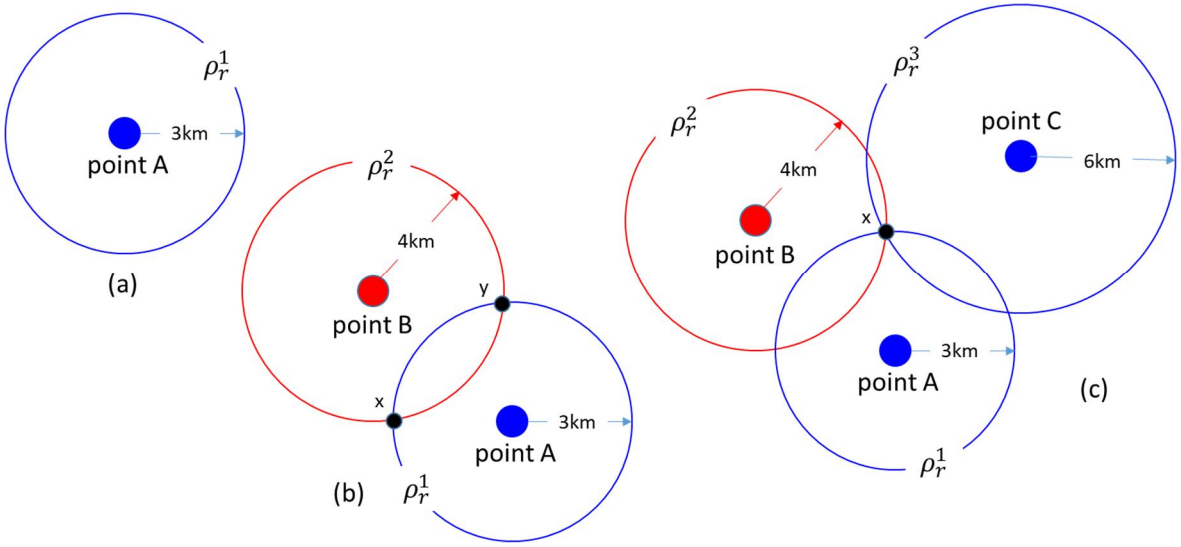


Figure 18: A simple illustration to understand GNSS-based position resolution [28]. (a) Knowing our range from point A, we can only know that we are somewhere around this point along the sphere of radius equal to the range. (b,c) We can resolve our position ambiguity by knowing our range from points B and C.

The accuracy when calculating the range from the satellites is influenced by several factors including the electronics on the receiver-side, the ionospheric delays and the multi-path caused by surrounding infrastructure around the receiver's location among others. Figure 19 shows an example of some of the challenges that must be accounted for during GNSS signal propagation from a satellite to a receiver. The receiver in this example is situated next to a tall building. While the receiver has a direct line of sight to satellite A, the signal from this satellite can be refracted due to ionospheric delays. This same satellite's signal may additionally be received by the receiver due to reflection from the building (multi-path). The signal from a second satellite, satellite B, may be blocked and reflected away from the receiver due to the building

structure.

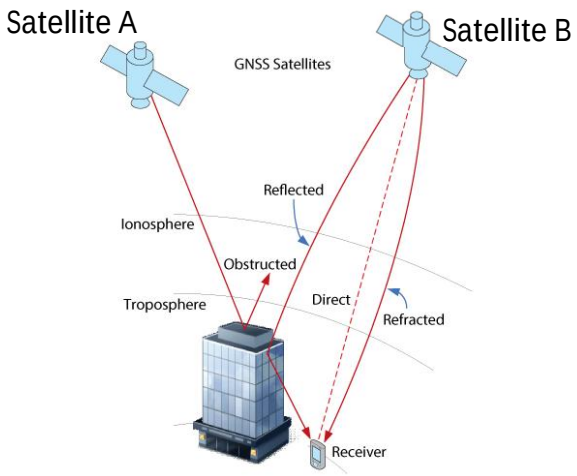


Figure 19: GNSS Signal Propagation [28]. Satellite A's signal is blocked by the infrastructure surrounding the receiver, whereas multipath signal propagation is being seen by the signal from Satellite B.

3.1.3.2 Real-Time Kinematic (RTK)

Real-Time Kinematic (RTK) is a technique that employs carrier-based ranges that are orders of magnitude more precise than code-based ranges. The basic concept in RTK is to minimize or remove errors that are common between the vehicle (also referred to as rover in the context of RTK) and a fixed GNSS reference station, also known as the base station. A rover enhances its position accuracy by using algorithms that incorporate corrections received from the base station in combination with its own measurements from available satellites, thereby providing centimeter-level position accuracy.

3.1.3.3 Oxford RT 3003

The RT 3003 is an inertial and GPS navigation system, which was developed by Oxford [27]. This GPS/INS system is heavily used within the automotive industry for applications that spread across many different systems within a vehicle – from vehicle dynamics related work to

testing Advanced Driver Assistance Systems. The RT unit consists of three angular rate sensors, three accelerometers, a GPS receiver and required processing all within a compact box as shown in Figure 20. Internal to this system are the algorithms that allow Kalman filtering and other alignment algorithms. A more precise measurement is achieved with Kalman filter monitoring and updating of these measurements using GPS. Under good driving conditions, defined by a combination of factors including number of available satellites and GPS fix among others, the RT 3003 can account for biases from its inertial sensors. Similarly, under challenging GPS conditions, it can use measurements from its inertial sensors to correct the GPS measurements.

Figure 20: RT 3003 by OxTS (left). Vehicle frame definition (right) with X facing forward, positive Y to the right of vehicle (passenger-side) and positive Z as downwards into the ground.

In addition to the coupling of GPS and INS measurements, this system also supports a dual-antenna configuration. In this system, the RT measures the difference in position to further solve GPS ambiguities. This dual antenna configuration also supports the provision of high-accuracy heading information in low dynamic driving conditions. The FTTA test facility has a base station location on the facility, not far from the oval track, thereby allowing the RT to operate in RTK mode on the track. This allows absolute positional accuracy to within a few centimeters to be obtained while driving on the track.

3.1.3.4 RT Setup in Vehicle

This section discusses RT installation and setup in a vehicle. OxTS provides a mounting strut that can be quickly installed in the rear passenger seating area supported by locking a lever in the strut after extending it to push against the vehicle floor and ceiling panel. Figure 21 shows the placement of the dual antennas on the roof. The dual antennas provided with the RT system typically have magnetic mounts, making them easy to install on the roof. It is best to maximize the distance between the antennas, for example, by placing the primary antenna towards the

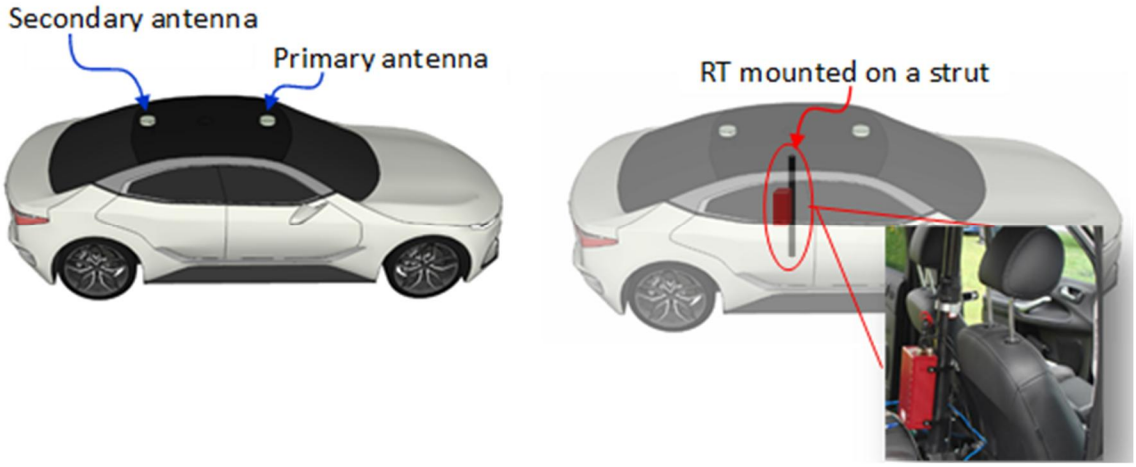


Figure 21: RT 3003 vehicle installation. Primary and secondary antennas can be installed on the roof using magnetic mounts (left), whereas the RT can be installed in the rear passenger area using a strut provided by OxTS for temporary installations.

front of the roof, while placing the secondary antenna near the rear-end of the roof. At the same time, care should be taken to ensure that the two antennas are on same flat plane which requires that the extreme front and rear end of the roof where the roof shape typically starts to bend in a passenger car be avoided. Finally, power can be applied to the RT unit through the cigarette lighter port available in the vehicle.

The next step after installing the antennas and the RT unit is to enter the extrinsic parameters related to the antenna and RT unit using the software provided by OxTS. This requires several different measurements, including the separation distance between the antennas, their height from the RT and the RT unit's height above the ground and from the rear axle. The final step after entering this information is to calibrate the system. A calibration routine involves powering the system up and driving the vehicle in a figure eight pattern as well as at higher speeds which include high dynamic turns that would allow the RT system to warm up and to calculate its IMU biases.

3.1.4 Data logging

The RT data logging setup is shown in Figure 22 .The RT system processes and transmits data using Controller Area Network (CAN) [30] protocol with an update in measurement every 10 ms. This translates into a maximum spacing interval between two data samples of approximately 0.3m while driving at highway speeds (assuming 31 m/s). There are several off-the-shelf tools available that can be used to log CAN data. CANalyzer, a software tool by Vector [31] is among the most widely used CAN tool within the automotive industry and was also used to log RT data in this research.

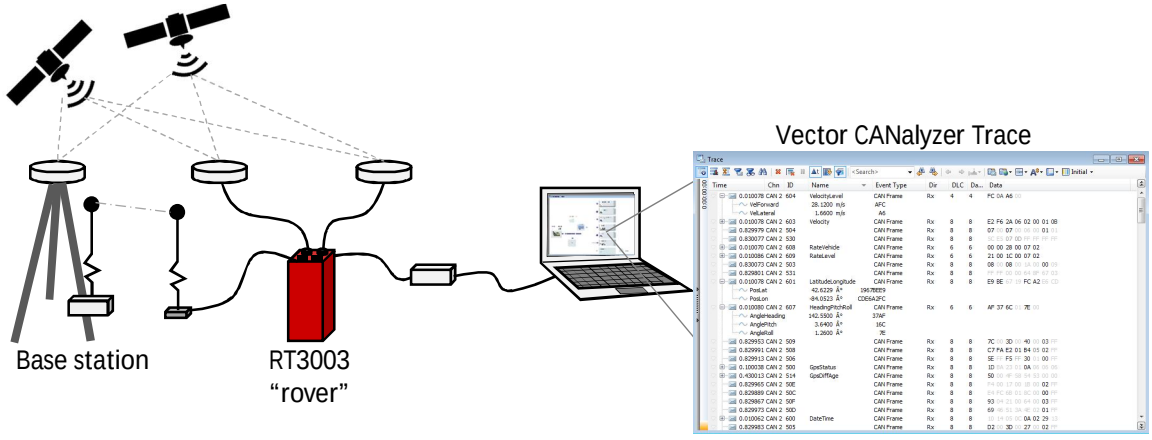


Figure 22: RT 3003 setup. The RT unit operating in RTK mode with a local base station corrections received over a wireless link. The system processes and outputs motion and pose data at 100Hz via CAN. The CAN data can be logged on a PC using Vector CANtech hardware and software.

A database configuration (dbc) file is a standard file format that contains definition of the CAN network, the ECUs connected on the CAN network and the messages and signals on a CAN bus. This information is necessary to decode raw data received on the CAN bus into physical data. Figure 23 shows a sample dbc file with a subset of the messages that are relevant for this research

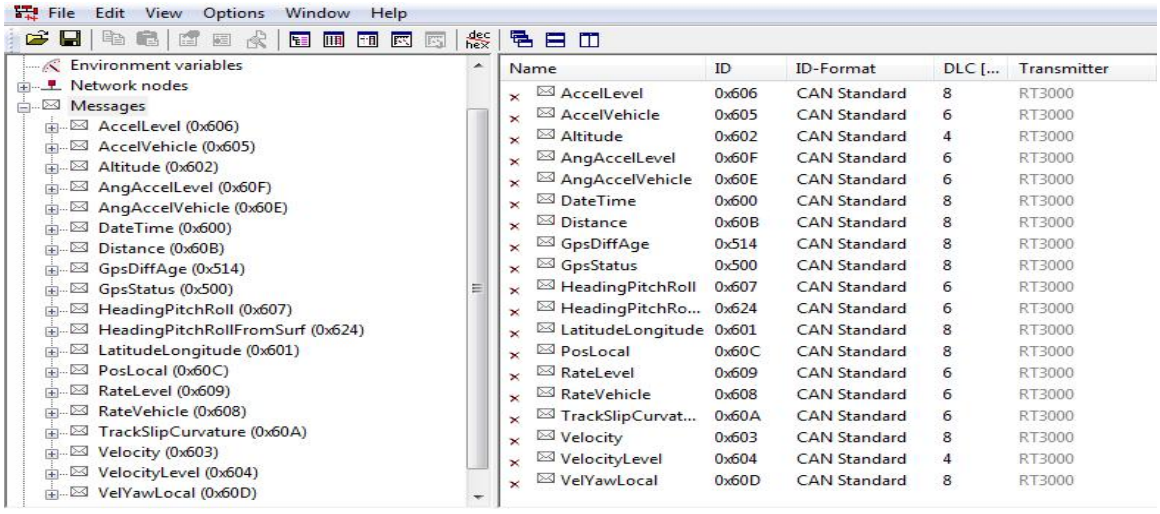


Figure 23: Database configuration (dbc) file for a subset of messages provided by RT 3003

from among all of the available messages that RT 3003 provides. The signal information within a logged data sample can be decoded using its dbc file and data can be extracted from CANalyzer into several formats including comma separated values (CSV) or a mat file [32]. Either format can be easily imported into MATLAB for further processing. There are several modes with which data can be logged in CANalyzer – continuous logging was done in the ego-vehicle throughout the test (details of the driving scenario are discussed in 4.1.1). On the other hand, data were only logged in the target vehicles as they moved to a new location. Since RT data in each vehicle were logged independently, a common reference or label in the logged data was needed to align the data samples from different vehicles. The GPS time, which is acquired and reported within the data stream transmitted by RT over CAN, was used as a common reference to sync data from different vehicles. The following list of signals was exported for each data sample and later used in MATLAB to align data between the logs from the ego and target vehicles.

Data Content	Units
Date and Time	year, month, day, s, min, hr
Latitude, Longitude	degrees, degrees
Velocity (forward, lateral)	m/s, m/s
Accelerations (forward, lateral, down)	m/s^2 , m/s^2 ,m/s^2
Heading, Pitch, Roll	degrees, degrees, degrees
Angular accelerations (forward, pitch, yaw)	deg/s, deg/s, deg/s
Track Curvature	1/m

Table 2: List of signals extracted from CAN data logs and exported into MATLAB for further processing

3.2 Data Pre-processing

A highly important step before inferring information from data is to thoroughly analyze the data and perform a pre-processing step. Depending on the characteristics of the data collected, this step may include data correlation in cases where multiple samples were collected from various participating entities. In addition, different mathematical operations are needed to (1) cleanse data, (2) to create newer relations with the data (which usually requires subject matter knowledge) that may help us to gain more insight into the data and (3) to identify and remove any erroneous data that can occur due to equipment malfunction or external interference in a controlled test. This section discusses the data pre-processing steps that were performed to prepare the data for further analysis.

3.2.1 Correlating Data Samples

Since data samples from each of the participating vehicles were recorded with independent logging systems, data samples from these vehicles had to be correlated. This was done by iterating through the ego-vehicle data samples and using the included GPS time-stamps as part of each of the vehicle data trace to find time aligning pairs between the target vehicle and ego-vehicle samples. A MATLAB script (m-script) was developed to search through the target vehicle data to find samples that time aligned with the GPS time in the ego-vehicle data trace. Since both target vehicles may have data samples that were captured in the same GPS time range as the ego-vehicle data sample, final selection of the matching pair of target vehicles for any given ego-vehicle path trace was done based on the shortest Euclidean distance to the ego-vehicle path. The matching pairs were then plotted and manually reviewed as a last step to confirm that correct ego and target vehicle pairs were selected. The following figure () shows one such sample

trace with the ego-vehicle path trace and its time-aligned pair of target vehicle data plotted to obtain confirmation on whether the pairing was done correctly. A database file was created as an output from this step that contained the ego-vehicle data sample start and end index, with its matching target vehicle pair for each of the trails performed on the test track. A CSV export of this database is shown in Table 3.

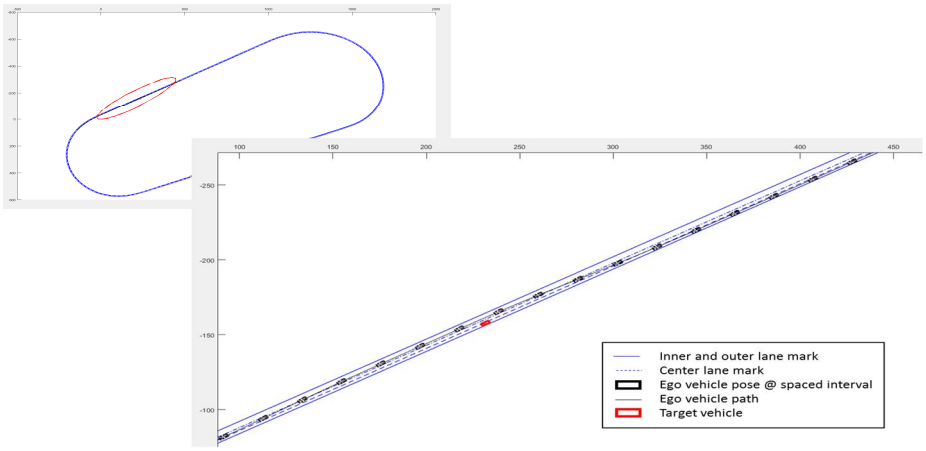


Figure 24: Data sample of an RT log from two different vehicles plotted on a test track map. The data samples from the two vehicles were time aligned using GPS time.

datasetNo.	startFileName	endFileName	startIdx	endIdx	matching TgtVeh	EgoVehVel
1	3	3	18000	21000	2	86.437
2	3	3	28000	30001	2	80.024
3	4	4	11000	13000	5	89.503
4	4	4	19000	22000	4	79.55
5	4	4	28000	30001	6	88.182
6	5	5	10000	14000	5	83.179
7	5	5	18000	22000	10	89.781
8	5	6	29000	2500	6	84.775
9	6	6	11000	13000	11	85.456
10	6	6	20000	23000	8	84.393
11	6	6	26500	29000	13	90.083
12	7	7	6000	8500	9	86.878
13	7	7	16000	18000	14	90.716
14	7	7	25500	28000	10	85.608
15	8	8	7000	9000	15	85.236

Table 3: Sample excerpt from a CSV file export of the database that was created after aligning start and end samples of ego-vehicle data for each test trial and its time-aligned pair of target vehicle indexes.

3.2.2 Calculating Lateral Offset and Road Radius

The next step after aligning the data samples between the ego and target vehicles was to calculate the lateral offset from the inner lane centerline and to correlate the radius of the road against each sampled position of the ego and target vehicles. For either of these calculations, centerline data sample that correlates with the sampled ego or target vehicle position had to be determined. Assuming the ego-vehicle position for the jth sample to be $p_{ego}(j)$. From $p_{ego}(j)$, a line perpendicular to the lane was drawn. The point where this line intersected the centerline was referred to as $p_{intersect}(j)$. The Euclidean distance between $p_{eog}(j)$and $p_{intersect}(j)$ was used to calculate the lateral offset, lat_{offset} (see Figure 25). Similarly, where this line intersected the inner lane line (assuming the ith point of line $p_{1,}\, p_{(1,i)}$), the radius of the road associated with that point ($R_1^i \subset p_{(1,i)}$) was used as the radius of the road the ego-vehicle was traveling on for the sample

$p_{ego}(j)$. The intersection to the inner lane line was also needed to calculate the radius at a certain distance ahead of vehicle. At this stage of data pre-processing, the road radius at headways of 1 through 5 secs was extracted for every sampled ego-vehicle point. Since the target vehicles in this study were stationary, only the road radius associated with their sampled positions was extracted and stored in a structure for later use.

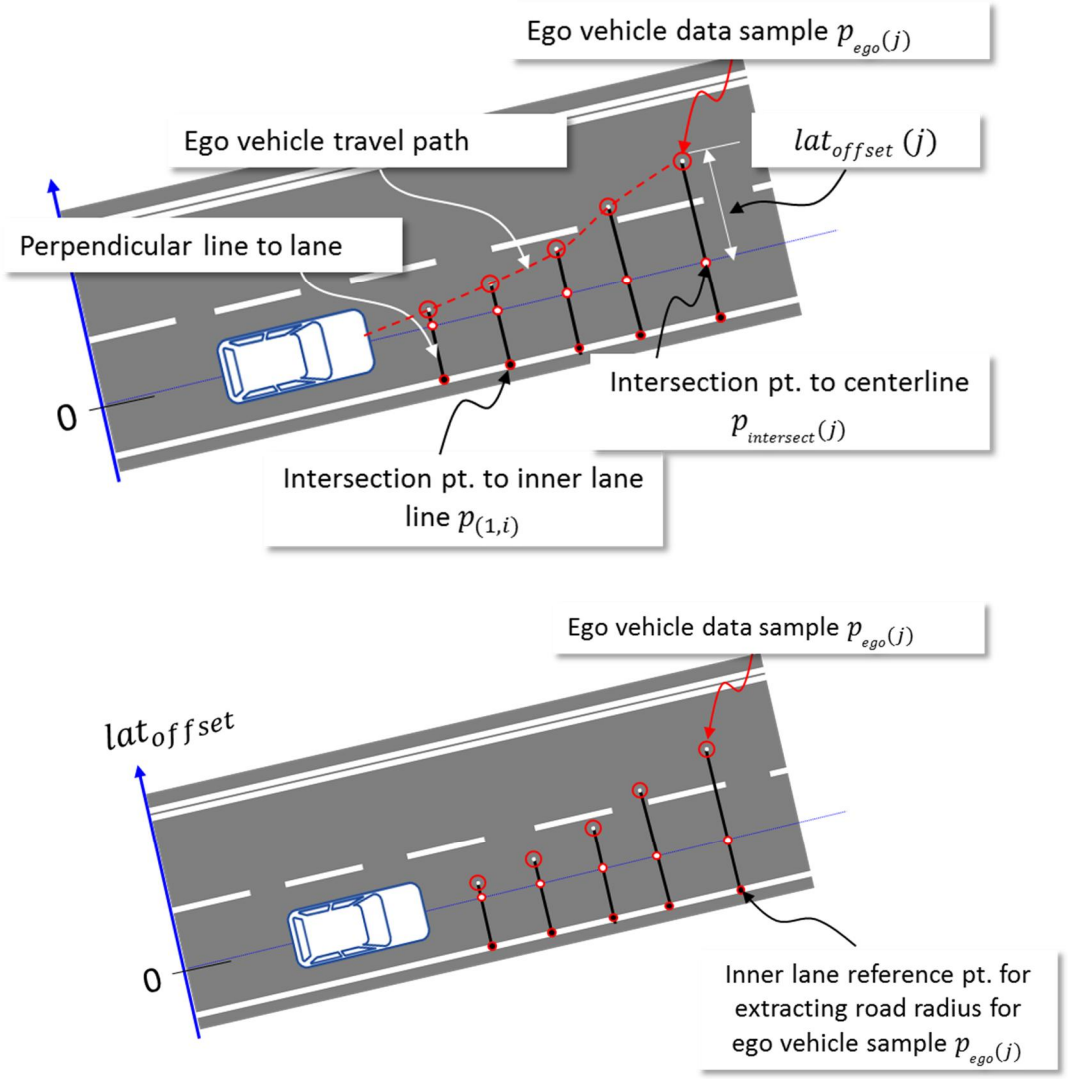


Figure 25: Calculating the distance to inner lane center and extracting the radius of the road.

3.2.3 Lane Change Active Flag

Lane change active (*laneChangeActive*) flag was created to observe its relationship with the other labels/factors in the driving environment. This flag was created based on the logic presented in code segment 1.

```
laneChangeEndIdx := 0

for i := laneChnageStartIdx:length( distanceToInnerLane)
    if lateralOffset_i > 0
        if laneChangeStartIdx ≠ 0
            laneChangeStartIdx := i
        else
            if lateralOffset_i > centerLaneLinePosition
                // successful lane change detected
                break from for loop
    else
        if laneChangeStartIdx ≠ 0
            laneChangeStartIdx := 0
```

Code 1: Algorithm logic for creating lane change start label

The pseudo code above describes the detection and creation of a lane change start label. The lane change end label was created with a similar logic flow, with the exception that this time we had to look for the *lateralOffset* to either get to 0 or close to it (in driving samples where the ego-vehicle did not return to a *lateralOffset* of 0 for a long distance after finishing a lane change). For data samples from the lane change start label to the sample before the lane change end label, the *laneChangeActive* flag was set (1). For all other samples, the flag was reset to 0.

3.2.4 Time-To -Collision

Time to Collision (TTC) refers to the time to impact if the current driving conditions continue [33]. TTC has been widely used in active safety applications, from use in Forward Collision Warning (FCW) to active braking in Automatic Emergency Braking (AEB) [34]. Time to

collision is a ratio of the distance to the target vehicle and the relative velocity between the ego and target vehicle:

$$TTC = {d(egoVeh, target)}/{\Delta v(egoVeh, target)} \quad [3\text{-}3]$$

where, d(egoVeh, target) is the Euclidean distance between the measurement origin point of the ego-vehicle and that of the target vehicle (Figure 26).

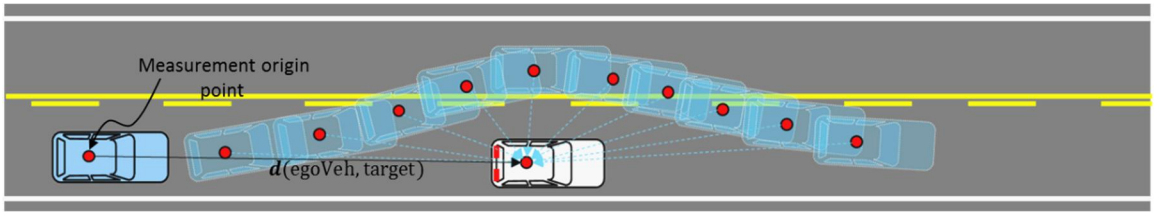


Figure 26: Distance calculation to target vehicle. Distance between the two vehicles is calculated based on the Euclidean distance between the origin of RT units in each vehicle.

3.2.5 Tying it all Together

As a final step in the data pre-processing phase, all the processed information from the previous steps for each ego-vehicle and associated target vehicle data sample was plotted. A sample plot is shown in Figure 27. The plot in Figure 27 (a) shows the lateral offset of the ego-vehicle plotted against the longitudinal (travel) distance to the target vehicle. The red circle marker in this plot as well as the remaining plots in this figure shows the same point in time where the TTC with the target vehicle was zero, i.e. the ego-vehicle was positioned next to the target vehicle in an adjacent lane. This helped in visualizing the data trends among different plots. For example, the speed of the ego-vehicle at TTC equal to zero was 23.67 m/s, while the lateral acceleration was 0.44 ${m}/{s^2}$. The small black markers shown around the longitudinal distance markers 100 and -200 in the plot in Figure 27 (c) show the lane change start and end points. The

logic to determining the data samples that qualify for an active lane change session is discussed in detail in section 3.2.3 . The data samples starting from the lane change start sample to the lane change end sample were assigned lane change active (1) state, as shown in Figure 27 (i).

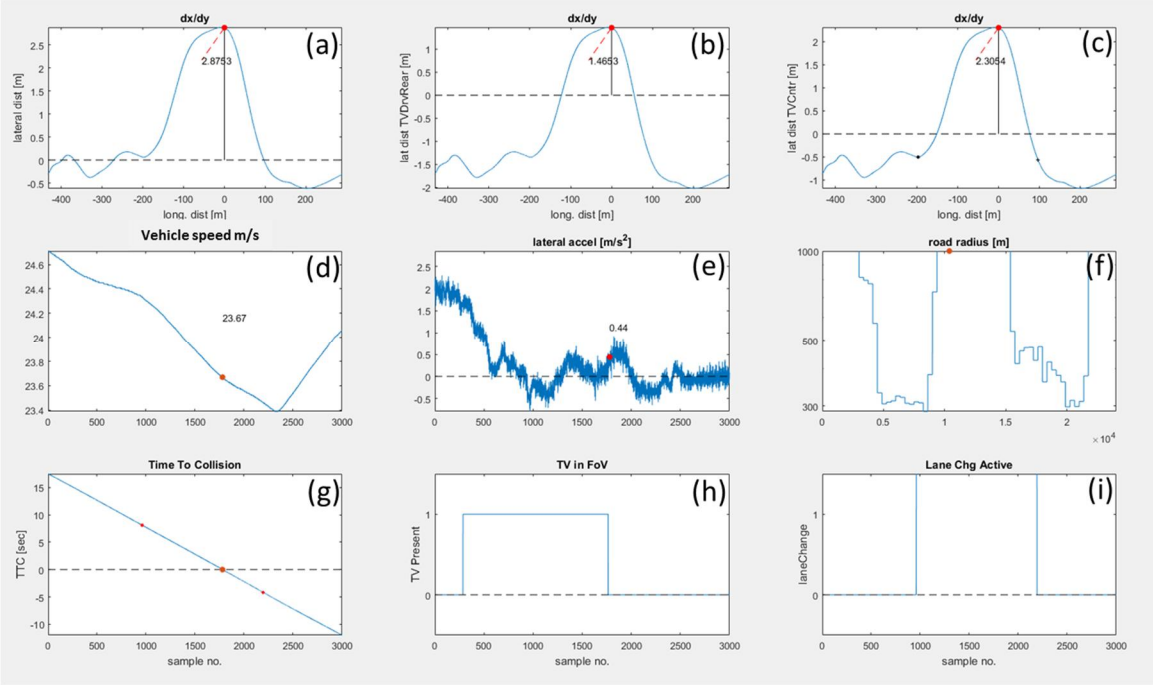


Figure 27: Plot of various data attributes to help analyze the data more effectively. (a) Ego-vehicle lateral offset(m), with data point labelled where the ego vehicle TTC from the target vehicle was 0. (b) Ego-vehicle lateral offset(m) from the target vehicle rear. (c) Ego-vehicle lateral offset from target vehicle center. (d) Ego-vehicle speed (mps). (e) Ego-vehicle lateral acceleration (mps^2). (f) Ego-vehicle radius while crossing target vehicle. (g)Ego-vehicle TTC, with lane change start and end points highlighted. (h) Simulated ideal radar sensor showing when the target was in Field of View, (i) Lane change active flag

Similarly, the plot in Figure 27 (g) shows the TTC for the ego-vehicle for this entire data sample. The two red circles in this plot around sample numbers 2200 and 900 are to highlight the lane change start end samples. These lane change start and end points can be correlated between the plots in (c) and (g) to visually analyze whether any relationship exists between the lane change start point, lateral distance of the vehicle and TTC.

Lastly, the plot in Figure 27 (i) was created by simulating a sensor with a horizontal Field of View (H-FoV) of 90 degrees. This plot shows whether the target vehicle was in FoV (1) or not (0). The motivation behind creating this plot was to determine whether the target vehicle was within the viewing range of a typical in-vehicle obstacle detection sensor from the point that the lane change start was defined [35]. This is important because we may be able to show a strong correlation of certain features to a desired output using specialized instrumentation, but we may not be able to extract the same features from a production sensor. That in turn would make it difficult to extend the research to a larger platform of production vehicles.

3.3 Simulation Test Platform

Although the final model in this study was developed using the driving data captured from the physical test platform discussed in previous section, a simulated environment was also created at the initial stages of this research. Typical to any real-world development, it is an effective strategy both cost and effort-wise to develop the initial concept in simulation, and to prepare the development environment for handling various aspects of real-world data. In addition, a validated model can also become a source of data as collecting actual data in every different driving configuration and speed can be expensive and potentially dangerous.

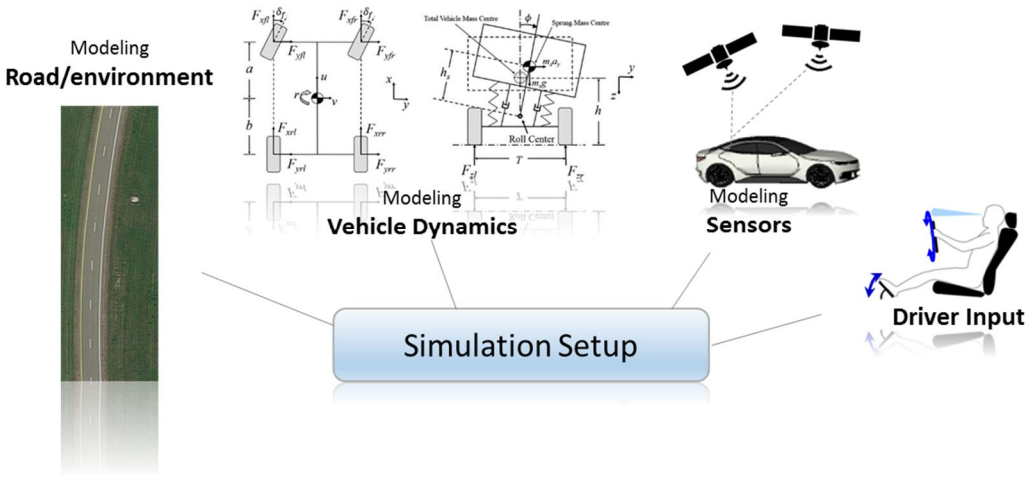


Figure 28: Key aspects to consider when simulating a driving environment

The extent to which a real-world driving environment should be modelled is dependent on the application and research objective. If the intent is to develop a control scheme framework for vehicle platooning [36], a longitudinal vehicle dynamics model [37] that allows the study of the impact of longitudinal spacing would be sufficient. Even when the intent is to study longitudinal and lateral spacing of the vehicle from other obstacles, varying road geometries, a detailed vehicle dynamics representation is useful. A model that reflects longitudinal as well as lateral driving dynamics, real-world sensor models as well as appropriate means to take driver input, all become relevant. When studying how someone drives through a given environment and what factors influence their choices, a simulated driving environment does not provide the driver the same feel as what they would experience while driving in the real-world. Even if all the other aspects of real-world driving conditions can be simulated, the sense of severity of the actions performed in a simulated environment would be completely different than how a person would behave or react in a real-world environment. Similarly, the forces acting on the vehicle and perceived by the driver while driving, which have a great impact in our driving styles, are

tremendously expensive to re-create in simulation and often impractical. For these reasons, driver input data in simulated environments were only used as for the visualization of testing parameters and as a proof-of-concept towards creating an accurate simulation.

3.3.1 Modeling Road Environment

There are several off-the-shelf environment modeling tools available for Advanced Driver Assistance Systems (ADAS) and automated driving related environment development. Some of the well-known names in the automotive industry include PreScan [38], IPG Automotive [39], Eelektrobit [40], dSPACE [41] and CarSim [42] among others. More recently, the Robot Operating System (ROS) [43] has also become popular for creating simulators focused on automated driving.

For this research, PreScan was chosen as the simulator for modeling the road environment as it supports nearly all aspects of simulated environment creation – from modeling the environment, to sensors, vehicle dynamics and driver input. PreScan provides built-in models for different types of road geometries and provides an intuitive interface to adapt these road models to match the real driving environment. Another useful utility built into PreScan is the ability to import a GPS track to create a road model. The test track GPS map was imported using this utility to create the oval track. The road created from the imported GPS track was then further edited to adjust the road attributes such as the number of lanes, lane width, lane mark type etc. (Figure 29).

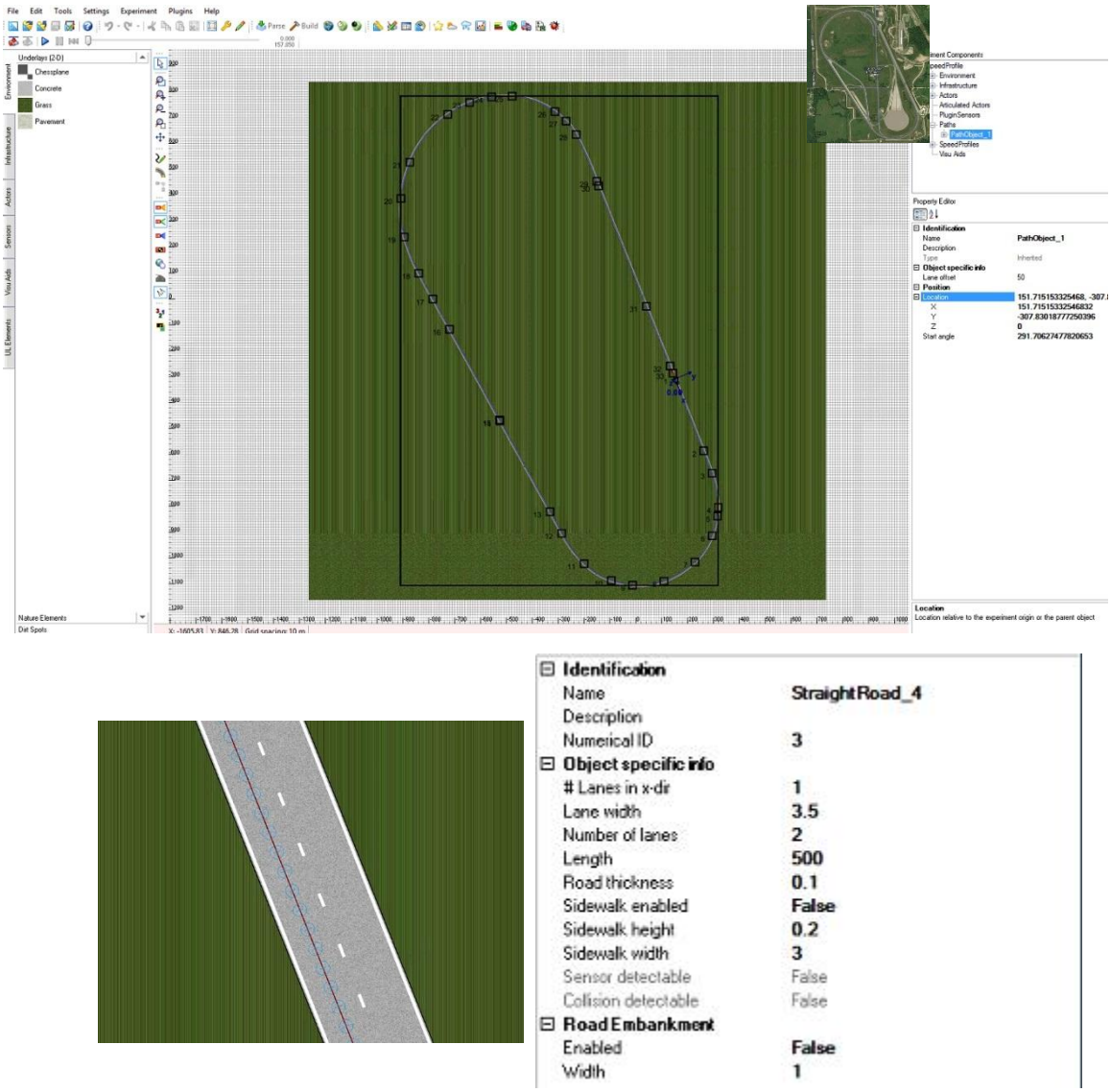


Figure 29: Simulated Test Track in PreScan (Top). Zoomed-in view of the road model of the East section of the track (bottom-left). Also shown are some of the editable properties of the road, such as number of lanes, lane width, lane mark type etc. (bottom-right).

3.3.2 Modeling Vehicle Dynamics

The path taken by a vehicle and the lateral forces experienced by it while traveling on a given path were important factors in this study. The performance of several different vehicle models was therefore studied and compared to data from the actual vehicle to ensure that the longitudinal and lateral forces produced within the simulated vehicle model were closely

representative of real-world driving dynamics. Figure 30 shows a 2D and a 3D version of the vehicle dynamics model implemented in PreScan [38].

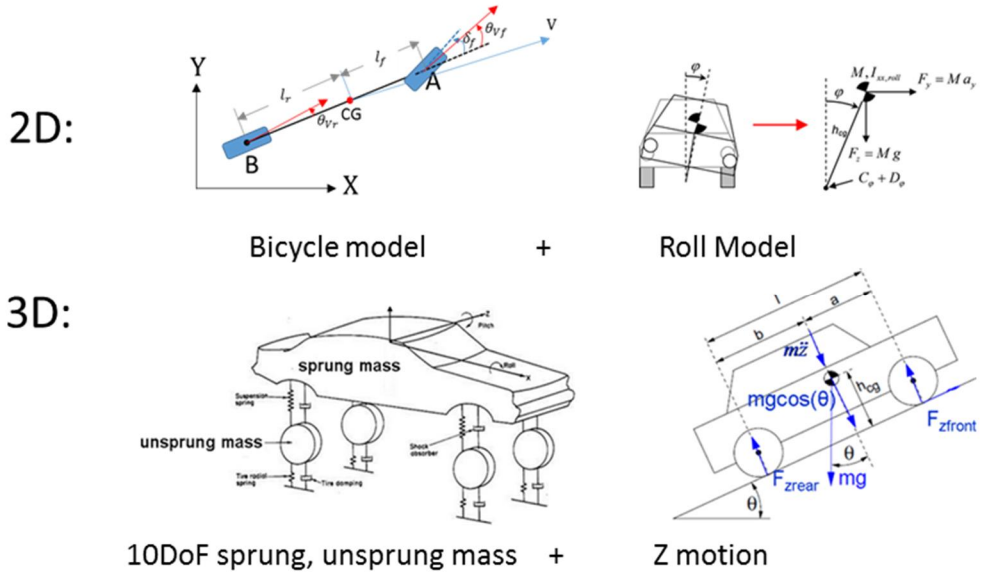


Figure 30: Vehicle dynamics models in PreScan [12]. The 2D model uses a bicycle model combined with a vehicle roll model. The 3D model on the other hand had a more detailed vehicle model representation with 10DoF between spring and unsprung masses.

The 2D vehicle dynamics model in PreScan was a combination of bicycle model and model for vehicle roll motion. There are several assumptions made in the bicycle model of the vehicle such as a constant turning radius and a zero slip angle at both front and rear wheels [37]. These assumptions hold true for turns of constant radius and slow speed driving (< 5 m/s), but are not valid for high-speed driving. The 2D model was therefore not considered further. The 3D vehicle dynamics model in PreScan has a much more elaborate vehicle dynamics model. It has for instance 10 Degrees of Freedom (DoF) between sprung and unsprung masses. Although our study did not consider vehicle travel in the Z-direction, this 3D model also incorporated a dynamics model for Z-motion of the vehicle (Figure 30). Reference [37] describes a detailed overview of vehicle longitudinal dynamics, including Z-motion. This 3D vehicle dynamics model was used and

several model parameters of this model were tuned to match the test vehicle such as vehicle mass, its track width, moment of inertia etc. The result of the simulated vehicle travelling through the scenario used in this study is discussed in the next section.

3.3.3 Simulating a Driving Scenario with Obstacle

After the creation of the road environment, and selecting and tuning the vehicle dynamics model, the next step was to simulate a driving scenario as intended in this research. Figure 31 shows the simulated drive scene of the ego-vehicle traveling in the inner lane on a straight road, changing lane for the stationary vehicle in the lane and then returning to the inner lane. The reference path for this simulated scene was created using the *path definition* tool in PreScan, whereas the path following (steering control) was done by the *path follower* model in PreScan. Figure 32 shows the longitudinal and lateral accelerations experienced by this vehicle.

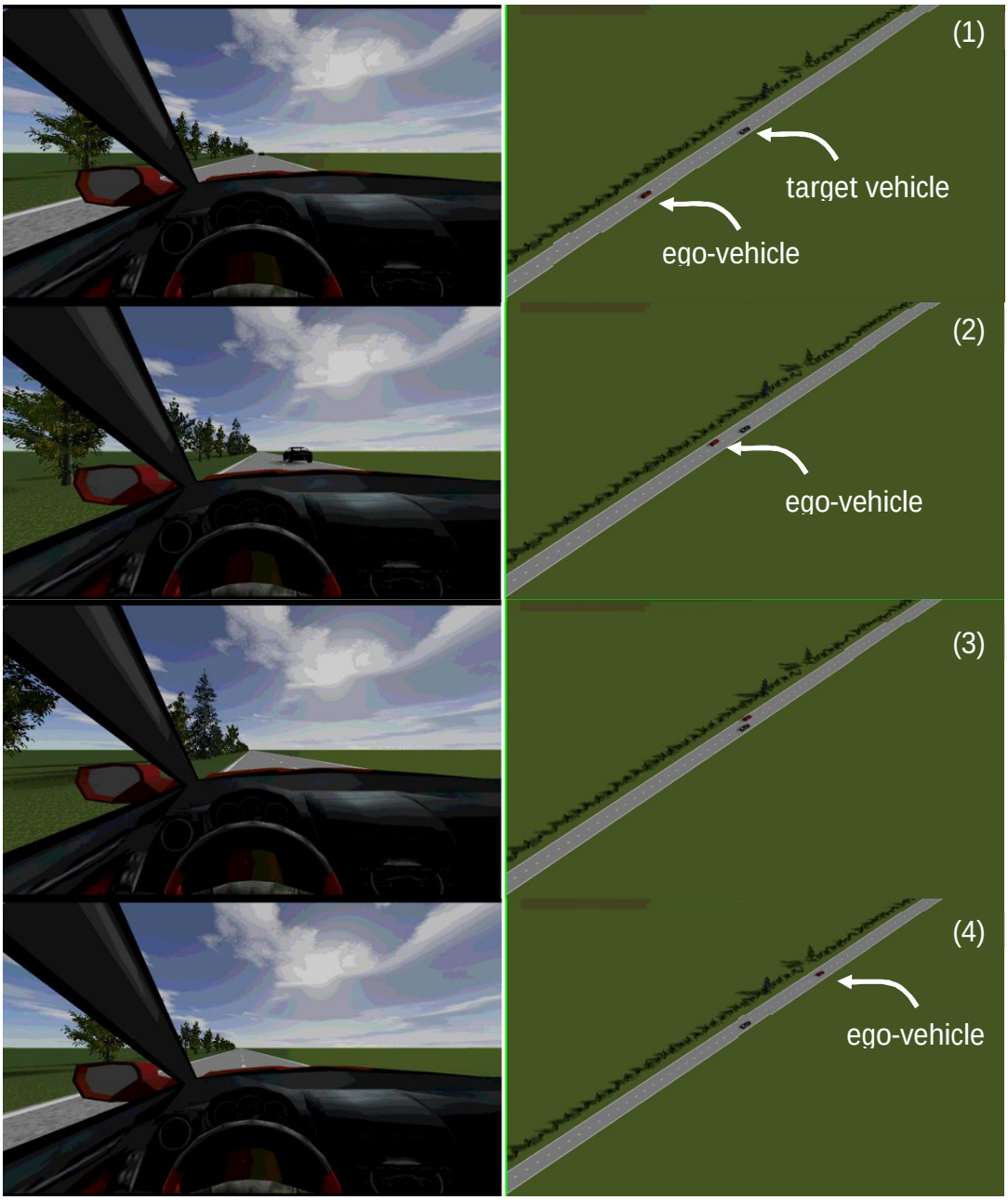


Figure 31: Simulated driving scene in PreScan to capture initial sample data to review the driving dynamics under various scenarios and to prepare the software processing environment for actual test data. (1) Ego-vehicle approaching target vehicle while traveling in the inner lane. (2)-(3) Ego-vehicle changing lane to pass the target vehicle. (4) Ego-vehicle returned to inner lane.

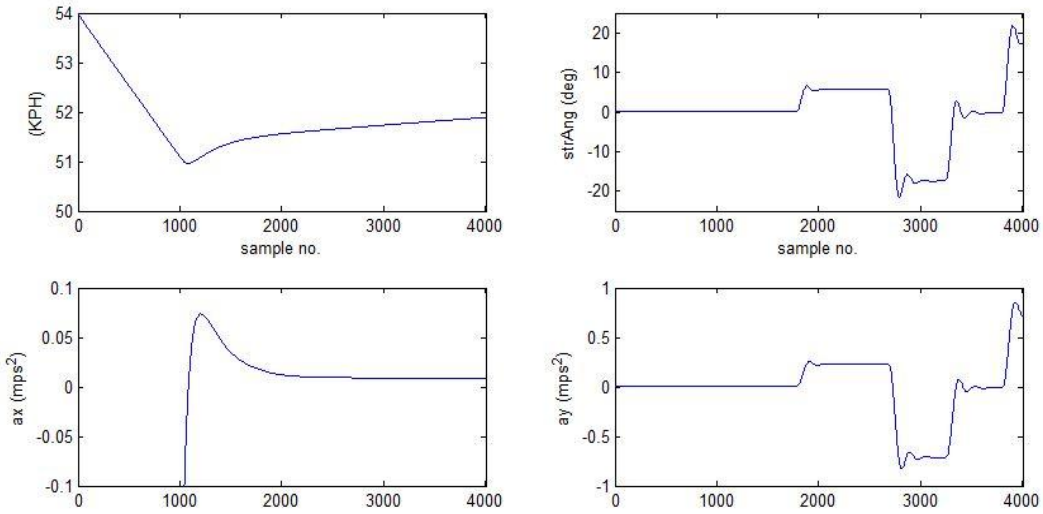


Figure 32: Simulated vehicle data. Input velocity (top left) and steering angle (top right), vehicle longitudinal (bottom left) and lateral accelerations (bottom right) while following the commanded steering.

3.4 Chapter Summary

This chapter provided a detailed review of the research platform developed and employed in this research. As part of the simulation platform, a road model was created in PreScan's modeling environment. This road model was similar to the physical test track that the data were later captured from. Two different variants of vehicle dynamics models, i.e. 2D and 3D dynamics models were explored in PreScan. The simulated vehicle dynamics response was compared against the actual vehicle drive data. The simulated vehicle dynamics model was tuned based on the parameters of the test vehicle including vehicle weight, wheel base etc. It was observed that, with same velocity input as was recorded in the actual test vehicle, the lateral acceleration of the simulated vehicle closely followed that of the actual vehicle. This simulated model was used to capture the simulated ego-vehicle's pose and motion data to be able to analyze and process the data from the physical test track.

As part of the physical platform, three test vehicles were instrumented with Oxford RT to acquire pose and motion data from each vehicle. The driving data were captured on a test track in Fowlerville Proving Grounds. The data set from the ego and test vehicles was aligned using a GPS time stamp. Data pre-processing was done to transform the data and to create a number of attributes such as the lateral offset, Time-To-Collision and Lane Change Active Flag. Finally, a data visualization interface was created in MATLAB where various data attributes were plotted to observe and to further analyze the trends and to find any possible relationships between the different data attributes.

In the next chapter, details of the driving task are presented. The formulation of the learning problem is then discussed, followed by details of the various data pre-processing steps prior to feature selection. The feature selection process is then discussed and a feature set comprising of four features that showed strong correlation in being able to represent the spacing profile was identified. Finally, numerous learning algorithms were explored and a model was learned to predict a driver's spacing profile.

CHAPTER 4: LEARNING FROM HUMAN DRIVING DATA

The next step after preparing the test platform was to capture human driving data on the physical test platform and to learn the spacing profile from these data. This chapter discusses the detailed steps involved in collecting the data and then learning user preferences from that driving dataset. The chapter is broken down into (1) the explanation of the test scenario, (2) collection of the drive data, followed by (3) analyzing the driving data, (4) performing feature selection and lastly (5) learning the driver's spacing profile using the selected features. In the last section of this chapter, the spacing profile is formulated as a constrained optimization problem. This was done to discuss and contrast the spacing profile as a learning problem versus a constrained optimization problem. The two approaches are compared and discussed in the next chapter.

4.1 Collecting Human Driving Data

4.1.1 The Driving Task

The driver selected for this study was a licensed driver, who also had the training required by the test track to drive on the course. The following is a description of the instructions given to the driver. More information on the overview of the task along with the IRB that was given to the driver prior to this study can be found in the research information sheet (see Appendix C).

1. The driver was asked to drive in the inner lane of the two-lane track and instructed to drive at a certain velocity $v_{des} = \{55, 70\}\, mph$ defined at the beginning of each test. This velocity was to be manually maintained to the best of the driver's ability.

2. Two target vehicles were parked at different locations on the inner lane at the beginning of each test session. The positions of the target vehicles were not disclosed to the driver (and were randomly distributed). The driving line of sight allowed the driver to see any given target vehicles at head$way \geq 5secs$. After the ego-vehicle passed a target vehicle, that target vehicle was either moved to a new location or kept parked at the same location.

3. The driver was instructed to change the lane when the target vehicle was encountered in the inner lane. When it was deemed safe and comfortable the driver was to change lanes and return to the inner lane after passing the ego-vehicle and continue driving at v_{des}.

Figure 33 shows the driving task, with the ego-vehicle (blue) making a lane change around a stationary target vehicle (white) and returning to the inner lane.

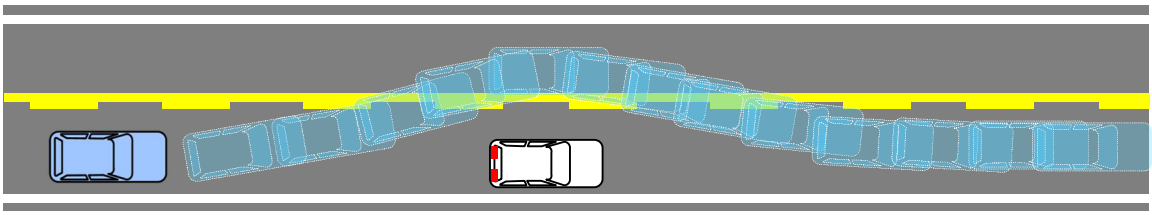


Figure 33: The driving scenario used in this study. The driver was asked to the drive ego-vehicle in the inner lane at a pre-defined velocity, change lane as deemed safe and comfortable when the target vehicle was encountered in the lane and return to the inner lane when deemed safe after passing the target vehicle.

4.1.2 Positioning of Target Vehicles on the Test Track

The test track was divided into four sections, each on the East and West sides of the track, with a total of eight sections based on the road radius trend for determining where the target vehicles should be positioned. The four segments on each of the East and the West sides of the

track were; (1) beginning of the curve where the road curve was finishing and turning into a straight road, (2) middle of the curve, where the road was straight, (3) start of the curve from the straight road and (4) the curved part of the road. These were referred to as follows: curve-to-straight, straight, straight-to-curve and in-curve respectively, as shown in the Figure 34 below.

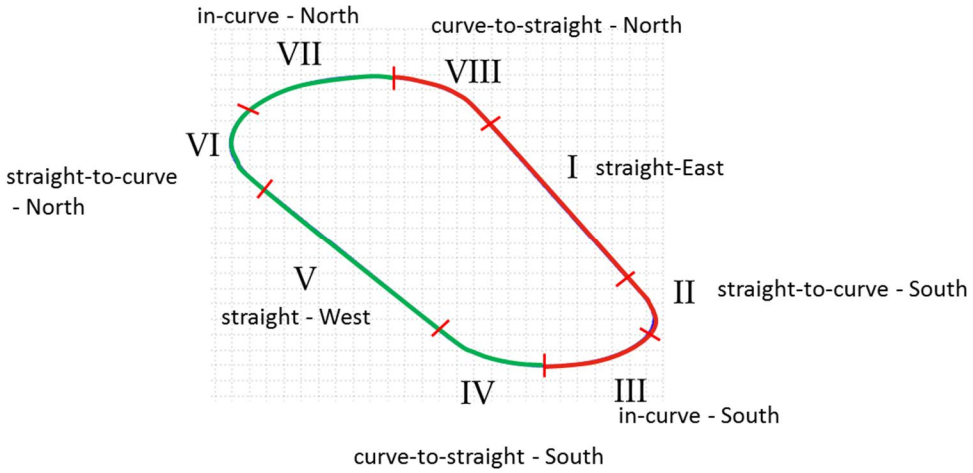


Figure 34: Test track sections created based on road radius trend. Target vehicle 1 was positioned at locations within sections I, II, III and VIII, while target vehicle 2 was positioned at locations within sections IV, V, VI and VII.

The original intent was to have the vehicles positioned among all of the locations equally, but due to limited track time (and the curved sections being much shorter than anticipated), we collected fewer samples on the curved section of the road as compared to the straightways. We recorded 67 sample of the ego-vehicle driving by the target vehicles on straight (East, West) sections of the road, while 28 samples were captured on the curved sections, as shown in Table 4.

Track Segment	#	% of Total Samples
Straight (East, West)	67	70.53%
Curved (North, South)	28	29.47%

Table 4: Number of samples in straight versus curved sections of the track

In regard to speed, the following plot shows the average velocity for each drive sample. Although the driver was instructed to drive at v_{des} of 55 and 70 MPH, since the speed was manually controlled, the speed range during a given drive sample could be seen to be in the range of $v_{des} \pm 5\ mph$. A total of 54 drive laps of passing the target vehicle (57%) were conducted at 55 mph, while 39 (41%) drive runs were conducted at 70 mph. Two drive runs (samples 89 and 90) were conducted at around 40 mph, but since completing the full lap around the course at this speed took roughly 10 minutes between setting up and driving, it was not feasible to capture a significant number of samples at slower speed for this test. The primary focus was therefore kept at highway speeds of 55 and 70 mph.

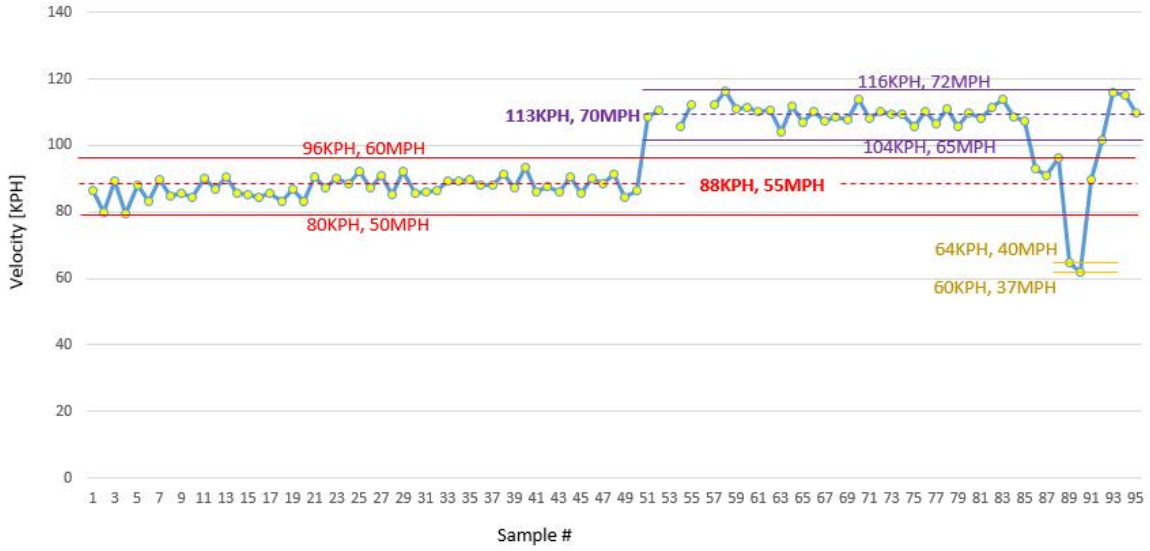


Figure 35: The arithmetic mean of the velocities for the drive samples.

Figure 36 shows the location of the target vehicles on the track. Target vehicle 1 is shown in blue(+), whereas target vehicle 2 is shown in red(o). The cyan solid line shows the path trail of the ego-vehicle from all the drives. Due to the zoom scale used to show all of the target vehicles on this plot, multiple drive runs almost appear to take the same path, which was not the case.

Figure 37 shows samples of the ego-vehicle during the lane change maneuver while passing the target vehicle at four different sections of the track.

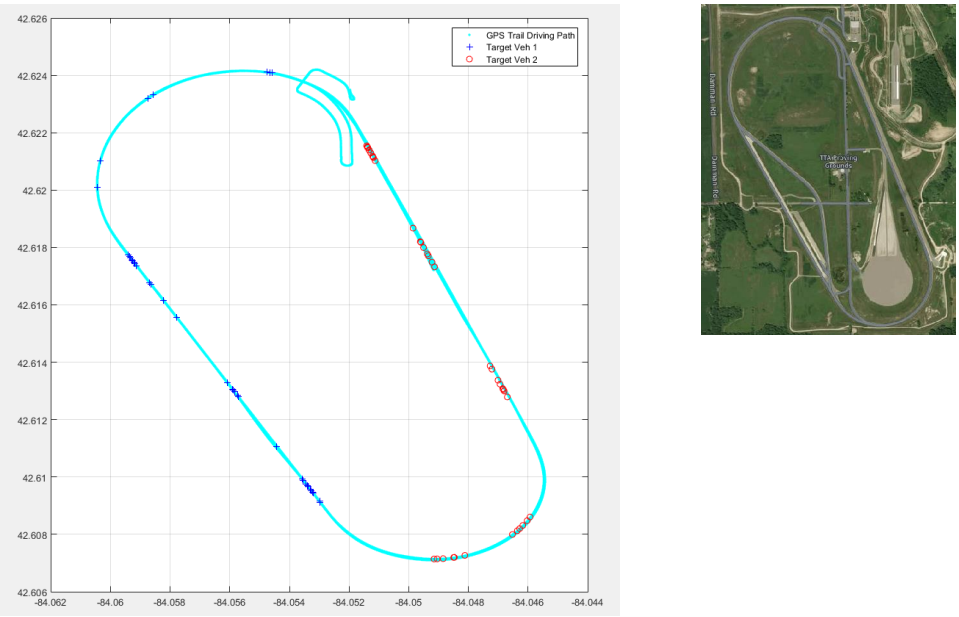


Figure 36: Target vehicle (1 and 2) positions on the track, with an underlay of samples of the ego-vehicle driving path in cyan color.

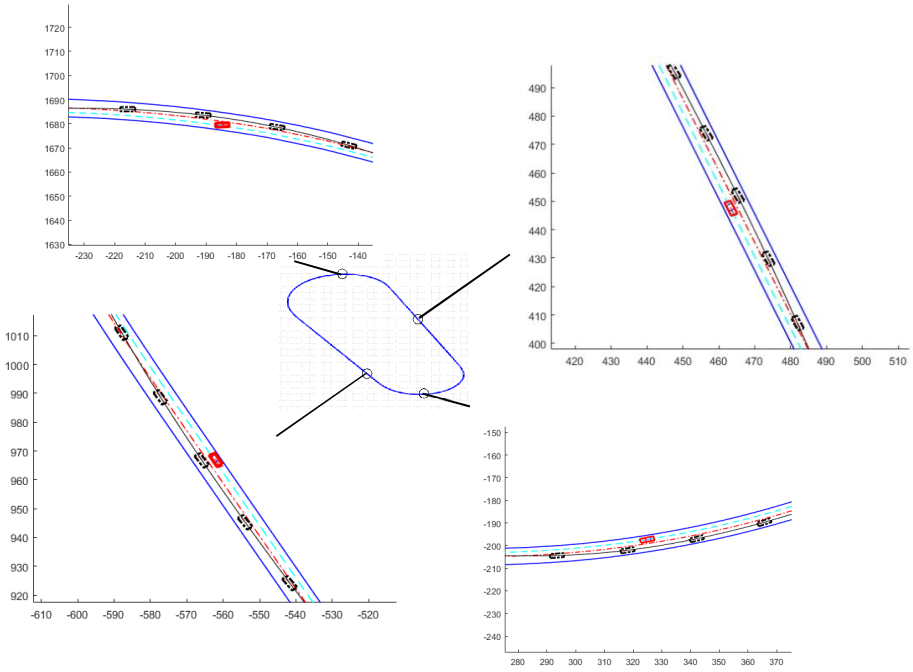


Figure 37: Selected samples of the drive data with the ego-vehicle making a lane change and passing by the target vehicle shown from various sections of the track.

4.2 Data Attribute Reduction

There are instances where an attribute does not reflect the conditions being studied. Careful consideration therefore needs to be given in removing data attributes that could potentially confound the modeling of that data and effect the model complexity and training time. Another benefit of data reduction prior to model training is in cases where the data may have a degenerative distribution [44]. Depending on the learning algorithm being used, removing such predictors can provide significant improvement in model performance and stability.

It is generally good practice to review the data in detail before moving on to modeling with it. Some of the aspects to consider are looking for outliers or abnormal trends in the data, perhaps due to equipment malfunction, recording errors or an abnormal event during data collection [8]. Such data samples can adversely impact a model's inferring ability. Similarly, there may be chunks of data that were gathered as part of the overall test, but that may not have any impact on the learning problem at hand and are therefore unnecessary to include as part of the training data.

This section discusses several aspects of the reduction including (1) reduction of the predictor set, (2) identifying and removing outliers and (3) selecting segments from the data samples that are of relevance to the training algorithms. The driving data collected from the test track can be divided into three main categories:

1. Ego-vehicle pose and motion data

This category contains the driving data pertaining to the ego-vehicle. These data include for instance the *x*, *y* position in UTM coordinates defining also its accelerations and orientation

information. The complete list of data captured of the ego-vehicle can be represented as follows:

Position and Motion Data: $\{x_i^j, y_i^j, z_i^j, \dot{x}_i^j, \dot{y}_i^j, \dot{z}_i^j, \ddot{x}_i^j, \ddot{y}_i^j, \ddot{z}_i^j\}$

Orientation Data: $\{\varphi_i^j, \theta_i^j, \psi_i^j, \dot{\varphi}_i^j, \dot{\theta}_i^j, \dot{\psi}_i^j, \dot{\delta}_i^j\}$

$$where, i := 1{:}\, length(dataSample), and\ j := 1{:}\, 95$$

2. Target vehicle pose and motion data

The data captured for the target vehicles were identical to the ego-vehicle, which can be represented as following:

Position and Motion Data: $\{x_j^\lambda, y_j^\lambda, z_j^\lambda, \dot{x}_j^\lambda, \dot{y}_j^\lambda, \dot{z}_j^\lambda, \ddot{x}_j^\lambda, \ddot{y}_j^\lambda, \ddot{z}_j^\lambda\}$

Orientation Data: $\{\varphi_j^\lambda, \theta_j^\lambda, \psi_j^\lambda, \dot{\varphi}_j^\lambda, \dot{\theta}_j^\lambda, \dot{\psi}_j^\lambda, \dot{\delta}_j^\lambda\}$

$$where, j := 1{:}\, 95\ for\ the\ total\ samples\ ,$$

$$and\ \lambda \epsilon \{1,2\}\ representing\ target\ vehicle\ 1\ and\ 2\ respectively$$

3. Lane Mark Information

For each of the lane marks, the position in terms of UTM coordinates *x* and *y* was calculated, along with the road radius at each sampled point, which can be represented as:

$$\{x_k^1, y_k^1, k_k^1\}_{k=1}^n, \{x_k^2, y_k^2, k_k^2\}_{k=1}^n \text{and } \{x_k^3, y_k^3, k_k^3\}_{k=1}^n$$

for the inner, center and outer lanes respectively, where *k* is the sampled point at 0.5m intervals along the road travel direction. For more details on how the lane data are represented, see section 3.1.2 .

The captured attributes from the ego-vehicle, target vehicles and lane mark information, along with the transformed data set (discussed in the Data Pre-processing, section 3.2) were

further analyzed to determine if this attribute set can be reduced to a smaller subset. Figure 38 shows the complete data set that was originally captured. It was determined that for the process of learning from these data, the driving environment in this study could be appropriately represented with a data subset of 13 attributes. These 13 attributes include features that were captured directly from RT as well as some of the transformed attributes from this dataset such as the TTC. The following is a list of the selected attributes at this stage of the feature reduction:

$$selectAttr = \{\dot{x}_{ego}, \ddot{y}_{ego}, \dot{\psi}_{ego}, distTV, TTC, latPos_{ego}, r_{ego}, r_{ego_{1s}}, r_{ego\_2s}, r_{ego\_3s}, r_{ego\_4s}, latPos_{TV}, k_{TV}\} \quad [4\text{-}1]$$

For some of the attributes that were excluded during this selection process, the information they provided was retained by the transformed attributes. Consider TTC and lateralOffset as an example. TTC captures the longitudinal positioning of the ego-vehicle from the target vehicle in its range measurement, thereby eliminating the need to consider the longitudinal position of the ego-vehicle and that of the target vehicle individually. The TTC measurement also takes into consideration the ego and target vehicle velocities as part of its relative velocity calculation, thereby eliminating the need to separately bring the target vehicle velocity into the predictor set.

Similarly, the lateralOffset predictor considers the *x* and *y* positions of both the inner lane line and the center lane line, as both are used in calculating the inner lane center line, which is what the lateral offset was calculated from. As for the remaining attributes that were not selected at this stage, they were not used because they were not relevant in providing key information related to the driving scenario being studied. For example, the ego-vehicle *z*-direction motion and *z*-acceleration were not considered as we knew that the driving

environment did not have too much variation in the *z*-direction. Similarly, although initially captured and analyzed, the longitudinal acceleration and longitudinal jerk were not selected as the ego-vehicle was driving at a steady speed in all the tests. Similarly, since the target vehicles were only being studied while they were stationary, their velocities and accelerations were not relevant and were therefore removed from the selected attribute set.

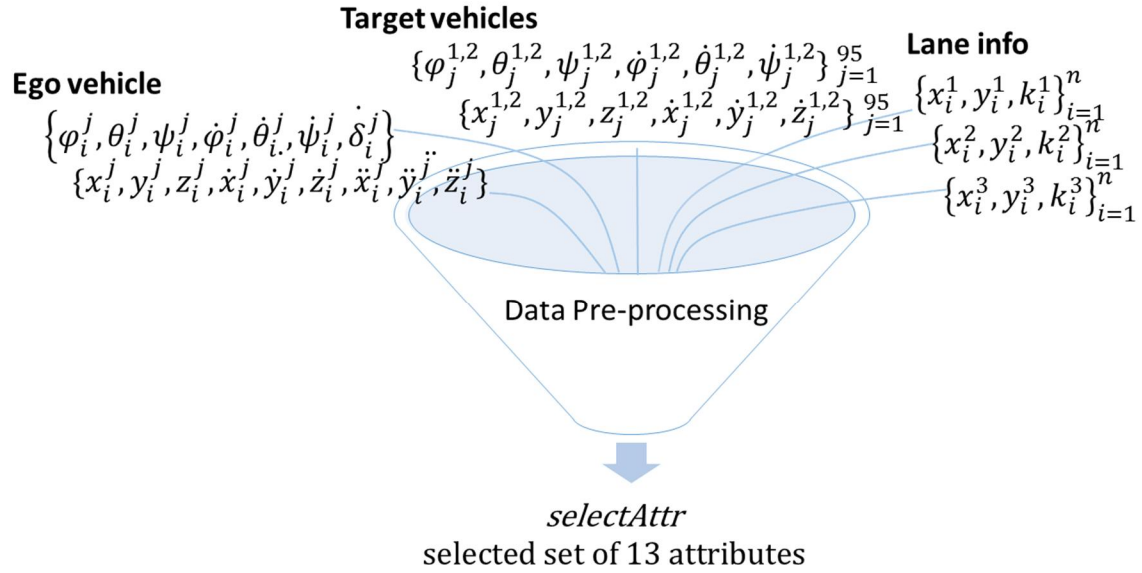


Figure 38: As part of the data attribute reduction, the driving data set was reduced from the original captured set of 37 attributes to a subset of 13. The information provided by some of the data attributes that were initially captured and did not make it to this list, were retained by the transformed predictors. Other attributes that did not make it to this set were deemed not significant in providing information for the driving case being studied.

Another important aspect while preparing the data for training is to look for data samples that may not be properly represented by the predictor set and may adversely impact the learning process. Consider the example shown in Figure 39. The ego-vehicle encounters an unplanned vehicle on the inner lane just before approaching target vehicle 1 and makes a lane change to pass this “intruder”.

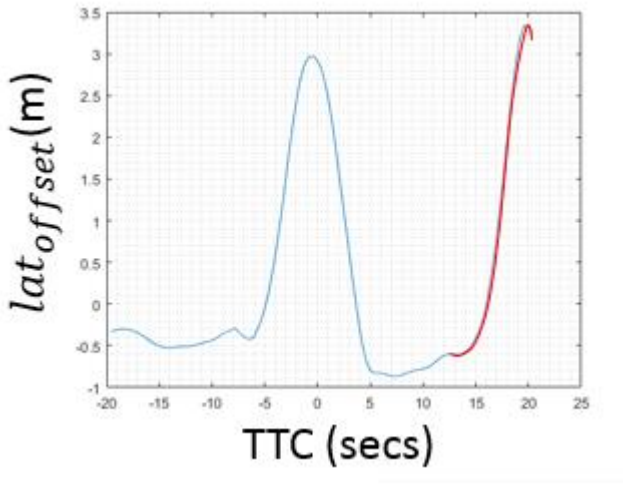


Figure 39: Ego-vehicle making a lane change for an unplanned obstacle (shown in red) on the test track prior to coming back to the inner lane and making a lane change for the target vehicle. Since pose and motion data for this unplanned vehicle were not represented in the predictor set, using this segment of data for training can adversely affect the learned model's performance as there is no predictor supporting the reasoning as to why the ego-vehicle made this lane change.

It is to be kept in mind that no information regarding this intruder vehicle was represented by the predictors. Therefore, using this segment of lane change data where there was no target vehicle present as represented by the predictors, would adversely affect the model learning. Similarly, using the remainder of the segment, shown on the left in this plot, may also be problematic as the path and motion profile of the driver in this case may be very different than in the normal/planned driving scenario. A few of such data samples were noted in the data log as erroneous data and manually eliminated from the training data set, shown in Figure 40.

Data Idx	Start Filename	Start Idx	End Idx	TV #	TV Data Idx	TV Position on Track	Ego Vel	Comments
4	CANWIN_L4_wTime	19000	22000	1	6	Mid	79.55	Intruder vehicle before target vehicle
17	CANWIN_L8_wTime	27000	28500	2	16	Mid	85.621	Intruder vehicle before target vehicle
63	CANWIN_L29_wTime	28000	30000	1	32	Start	104.081	Intruder vehicle after target vehicle
72	CANWIN_L33_wTime	19000	21000	2	58	end	110.022	Intruder vehicle after target vehicle
77	CANWIN_L35_wTime	1000	2200	1	40	end	106.378	Intruder vehicle after target vehicle
82	CANWIN_L36_wTime	28000	30000	2	67	end	111.387	Intruder vehicle before target vehicle

Figure 40: An excerpt from the data log highlighting data samples that were removed from the training set as part of pre-processing step due to erroneous data.

4.3 Feature Scaling

For problems involving multiple features, ensuring that the features or predictors are on the same scale, i.e. have same range of values, can be helpful for models that use gradient descent to converge. Examples of some of the learning algorithms that benefit from features being on the same scale include Partial Least Squares and Neural Networks [46]. Among some of the well-known standardization techniques are Z-score normalization and min-max scaling [44, 47, 48].

Standardization or Z-score normalization rescales features to have zero mean ($\mu := 0$) and standard normal distribution ($\sigma := 1$). This is achieved by subtracting the average predictor value from all the values. This results in all the values having a zero mean. The resulting values are then divided by their standard deviations, resulting in all values having a standard deviation of one. Z-score calculation for feature vector $\boldsymbol{x}$ can be represented as:

$$\boldsymbol{z} = \frac{\boldsymbol{x} - \mu}{\sigma} \qquad [4\text{-}2]$$

The Z-score standardization was applied to all the feature vectors in this study before moving on to the next step of feature selection.

4.4 Feature Selection

For learning problems that have high-dimensional data, selecting a smaller subset of features for use in the model construction can be helpful in (1) increasing model accuracy, (2) in reducing the model complexity and (3) in reducing overfitting by enhanced generalization [45, 44]. The process of selecting a subset of features is referred to as Feature Selection or Subset Selection.

There are two main approaches that can be employed for subset selection: wrapper methods, also known as best subset selection methods attempt to the find the predictor set that provides the best model performance by exhausting 2^ppredictor combinations, where p is the total number of predictors. In other words, this method uses all the predictor sets individually as the learning algorithm inputs and evaluates the performance of the model derived using each.

This method usually works well for problems that do not have large sets of predictors to begin with. For problems containing high-dimensional predictor set, this technique suffers from computational limitations [46]. As an example, for a problem containing 20 predictors, there would be over a million combinations of predictor sets to develop the model for and then compare performance of those models. Root Mean Squared Error (RMSE) is generally used as a performance measure in evaluating model performance among others [44].

Unlike wrapper methods, filter methods are evaluated with predictive models and only the predictors that pass certain criteria are then used in developing the final model. A thorough review of the filter methods is provided in [49].

Since the number of predictor set being considered for the learning problem in this research was comparatively small (a total of 13 predictors selected before feature selection, with

5 of them being the road radius at different headways, leaving 8 unique predictors), the wrapper method was employed for feature selection, with a decision tree approach [44] used as the learning algorithm.

	Predictors														
Iter	#Pred	YR	longD	Rad	Rad1S	Rad2S	Rad3S	Rad4S	Rad5S	TTC	Vel	LatAcc	RadTV	TVLatD	RMSE
1	13	x	x	x	x	x	x	x	x	x	x	x	x	x	0.04
2	8		x	x			x			x	x	x	x	x	0.05
3	6		x	x			x			x	x			x	0.04
4	6		x	x	x		x				x			x	0.06
5	5			x			x			x	x			x	0.04
6	5		x	x			x				x			x	0.04
7	5		x	x				x			x			x	0.04
8	5		x	x					x		x			x	0.05
9	5		x	x		x					x			x	0.05
10	5		x	x	x						x			x	0.05
11	5						x			x	x	x		x	0.05
12	5			x		x				x	x			x	0.05
13	4			x			x			x				x	0.13
14	4						x			x	x			x	0.03
15	4					x				x	x			x	0.03
16	4				x					x	x			x	0.03
17	4			x						x	x			x	0.03
18	3			x						x				x	0.20
19	3						x			x	x				0.12

Worst

Best

Figure 41: Evaluating model performance against various combinations of predictors. root mean square error was used to evaluate model performance. A selected combination of 4 predictors (green) provided the best model performance.

A sample of the predictor set combination and RMSE value for their respective models are shown in Figure 41. The search space explored for feature selection started with 13 predictors. The minimum number of predictors considered were 3, as the driving environment was not appropriately represented for any predictor combination below 3 based on subject matter knowledge. One of the goals in the overall learning process was to use the predictors that are least expensive in terms of acquiring them in real-world vehicles, ones that are less noisy and that can be measured relatively accurately.

Looking at the model performance, the highest number of predictors, 13, provided reasonable performance, but that could also have been due to model overfitting. As mentioned before, some of the predictors could be meta-predictors that intrinsically include other predictors. The overfitting with 13 predictors becomes obvious as we move to a lower number of predictor combinations. The predictor sets of 4 and 5 provided similar performance as that of the 13-predictor combination. Finally, reviewing the performance with combinations of 3 predictors resulted in erratic RMSE values. Based on these results, a selected group of 4 predictors provided the best RMSE and was selected as the feature set. For combination of 4 predictors, radius of the road at 1, 2 and 3 seconds provided the same RMSE value. Since, like the road radius of public highways, the road radius of test track changed gradually, this may have caused the radius values at headways of 1 through 3 secs ahead of the ego-vehicle to be close enough to each other to have the same significance in the model performance. The authors of [50] studied driver gaze to determine where drivers look when they drive and concluded that we look at the road 1 to 2 secs ahead of us in determining our steering correction. Based on this study, in combination with the observation that the radius at headways of 1 through 3 secs was providing the same model performance, the radius of the road at 1 secs of headway was selected. The final predictor set from the feature selection step was as follows:

$$\boldsymbol{p} = \{\dot{\boldsymbol{x}}_{\boldsymbol{ego}}, \boldsymbol{TTC}, \boldsymbol{latPos}_{\boldsymbol{TV}}, \boldsymbol{R}_{\boldsymbol{ego}_{1s}}\} \qquad [4\text{-}3]$$

4.5 Model Selection and Training

4.5.1 Selecting an Appropriate Learning Algorithm for a Problem

To select an appropriate model for any prediction problem, it is important to understand the solution that is being learned with respect to the ability of the chosen learning algorithm. At one end of the spectrum are the algorithms that are restrictive in their predictive ability, but are more interpretable. Figure 42 provides a selected list of algorithms, plotted on a scale showing how flexible versus interpretable each of the listed algorithms is. An example of such a learning algorithm would be linear regression, where one can easily understand the relationship of the predicted value Y, to each of the predictors, $X_1, X_2 \ldots X_p$, but this learning algorithm, as its name suggests, can only provide accurate models for problems where the predictors have a linear relationship to the predicted value.

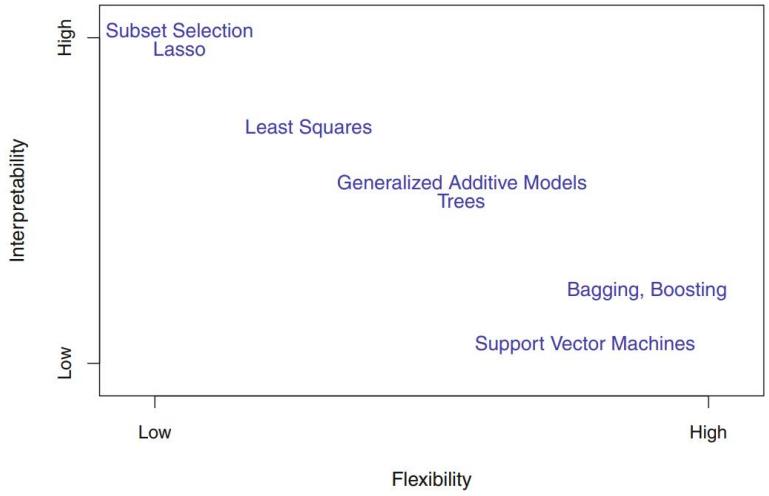


Figure 42: Tradeoff between a learning algorithm's flexibility and its interpretability [46].

On the other hand, a highly flexible model such as bagging trees that can represent a complex, non-linear relationship between the predictors and the output can be very difficult to interpret. Figure 42 shows a plot of flexibility versus interpretability and lists several algorithms.

Towards the left in this plot are algorithms that are highly interpretable, but low in their flexibility to be able to model complex and non-linear relationships among the predictors and the observed value. As we move towards the right of this plot, we increase the leaning algorithm's flexibility and ability to model complex relationships, but these models become harder to interpret. Therefore, depending on the complexity of problem at hand, the relationship between the predictors and the output, and the desire to be able to infer the relationship between the input and output, can determine the appropriate learning algorithm for a given problem.

There is a large selection of learning algorithms available for regression problems (see Figure 43 for a summary of algorithms). For a given learning problem, if we have an intuition of the trends of our data and what type of relationship exists between the predictors and the observed value, we can be more effective in selecting an appropriate learning scheme. For example, if we have intuition based on the data we have reviewed that the observed value would be a linear cost function of some combination of the predictors, we can try some of the less flexible, but more interpretable, algorithms such as linear regression. In the case of a learning problem that involves a very large set of predictors, identifying such relations can be challenging or unfeasible. In such cases, a slew of learning algorithms can be explored by developing models using various algorithms and the performance of these models can be studied further to narrow them down to an algorithm that would work best for that problem. Both MATLAB and Weka [25] have such exploration features available where several learning algorithms can be selected to train against a given dataset and model performances can then be further evaluated using metrics that are discussed in the next section. A list of learning algorithms is provided in Figure

43. Among other aspects, a learning algorithm's ability to perform feature selection and its data pre-processing capabilities are all listed in this table.

Model	Allows $n < p$	Pre-processing	Interpretable	Automatic feature selection	# Tuning parameters	Robust to predictor noise	Computation time
Linear regression†	×	CS, NZV, Corr	✓	×	0	×	✓
Partial least squares	✓	CS	✓	∘	1	×	✓
Ridge regression	×	CS, NZV	✓	×	1	×	✓
Elastic net/lasso	✓	CS, NZV	✓	✓	1–2	×	✓
Neural networks	✓	CS, NZV, Corr	×	×	2	×	×
Support vector machines	✓	CS	×	×	1–3	×	×
MARS/FDA	✓		∘	✓	1–2	∘	∘
K-nearest neighbors	✓	CS, NZV	×	×	1	∘	✓
Single trees	✓		∘	✓	1	✓	✓
Model trees/rules†	✓		∘	✓	1–2	✓	✓
Bagged trees	✓		×	✓	0	✓	∘
Random forest	✓		×	∘	0–1	✓	×
Boosted trees	✓		×	✓	3	✓	×
Cubist†	✓		×	∘	2	✓	×
Logistic regression*	×	CS, NZV, Corr	✓	×	0	×	✓
{LQRM}DA*	×	NZV	∘	×	0–2	×	✓
Nearest shrunken centroids*	✓	NZV	∘	✓	1	×	✓
Naïve Bayes*	✓	NZV	×	×	0–1	∘	∘
C5.0*	✓		∘	✓	0–3	✓	×

†regression only *classification only

Symbols represent affirmative (✓), negative (×), and somewhere in between (∘)

- CS = centering and scaling
- NZV = remove near-zero predictors
- Corr = remove highly correlated predictors

Figure 43: Summary of Learning Algorithms and some of their characteristics [44].

During the data pre-processing stage, it was observed that the predictors do not have a linear relationship with the lateral offset. Knowing that our learning problem was that of a non-linear regression, bagged trees and neural networks [19] were at first used to build the model. To see if a slightly less flexible algorithm might also provide acceptable performance, a few such algorithms were used to develop the model. The performance of each of the developed models was then compared with the performance of bagged trees and neural networks. Among the various learning algorithms attempted were; 1) Linear Regression, 2) Quadratic Support Vector Machines (SVM), 3) Cubic SVM, 4) Fine Gaussian SVM, 5) Simple Tree, 6) Boosted Trees, 7) Bagged Trees, 8) Neural Networks.

An important point to be noted here is that each of the above algorithms were used to develop a model while using their default parameters for training in MATLAB. The model performances discussed in the next section therefore do not reflect a given algorithm's overall learning capability. Instead, the model performance at this stage should only be treated as a rough indication of how well a given learning algorithm would perform for the given data. At this stage in the learning process, the intent is to select an algorithm that provides the right combination of flexibility, interpretability and predictability. Once an algorithm is selected based on these requirements, model performance can be further improved by tuning of the various parameters. The intent of this research was not to find the best tuned model, but rather an approach to how such a learning problem would be solved, along with a model that represents a driver's lateral offset that was acceptable for our use case. Therefore, further effort was not invested in improving the performance of the models that did not performed well based on the performance metrics used.

4.5.2 Metrics to Evaluate Model Performance

The discussion on model performance can be separated into *test error rate* and *training error rate* [8]. If the available data have been separated into test and training data sets prior to training, a model's predictability for unseen data can be estimated by evaluating its performance against the test set. This method of estimating model performance is referred to as the *test error rate*. Since data collected in the real world are usually limited in terms of variability, we can only gain an estimate of how the model may perform with future unseen data through the test error rate. The performance of the model estimated through the test error rate is not guaranteed for all future data because of the limited data set that was introduced to train the model with.

To evaluate a model's performance during the training phase, the available data can be divided into a training set and validation set. During training, the learned model is evaluated against the validation set that was held out prior to training to estimate the model's predictability. This method is referred to as holdout validation. If the validation data set always has the same fixed set of samples from the data, validating model's predictability using it would give a limited insight into how well the model would perform. To counter this problem, k-fold Cross Validation (CV) is widely used for model validation. In this approach, the available data are divided into k groups or folds. In each training iteration, $k - 1$ folds are used to train and the k^{Th} fold is used as the validation set to evaluate model performance. This process therefore provides k estimates of model performance, and the mean of these estimates is used as the overall model performance.

To quantify a model's predictive capability, we compare the predictions against the observations and use a metric or a variety of metrics to evaluate the performance. Some of the common metrics for quantifying a model's predictive capability are Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Coefficient of Determination (R^2) [19, 26]. These metrics are applicable in either calculating the test or training error rate. The MSE is calculated by summing the squares of the residuals and dividing them by the total number of samples. RMSE is then calculated by taking the square root of the MSE. RMSE is generally interpreted as how far the residuals are from zero. We generally would want MSE and RMSE to be low (ideally zero), which would indicate a high quality of fit between the predicted and observed values. In the case of k-fold cross validation, we would have k values for $RMSE$, i.e. $RMSE_1, RMSE_2, \ \ldots \ RMSE_k$. The overall performance estimate is then computed as the average of these values;

$$CV_k = \frac{1}{k}\sum_{i=1}^{k} RMSE_i \quad [4\text{-}4]$$

To calculate the coefficient of correlation (R^2) [53], we find the coefficient of correlation ($R = \rho_{(X,Y)} = {cov(X,Y)}/{\sigma_X \sigma_Y}$) between the actual and the predicted values and square it. $R^2$is interpreted as how well the observed values are represented by the model, based on the total variation of the predicted output of the model [54]. All three of these metrics, namely MSE, $RMSE$ and R^2were used in evaluating model performance in this research.

4.5.3 Comparing Model Performance

The prediction performance of various models was compared using the metrics discussed in the previous section. Figure 44 shows the performance of the various models that were built using 5-fold cross validation. The linear regression, as was anticipated, did not perform well, with a RMSE of 1.23 and R^2 of 0.1. The Fine Gaussian SVM model performed well, with RMSE of 0.09 and R^2 0.96. The models trained using Bagged Trees and Neural Nets stood out amongst the attempted regression algorithms, with RMSE of 0.04 and 0.26 respectively. These two trained models were tested against the

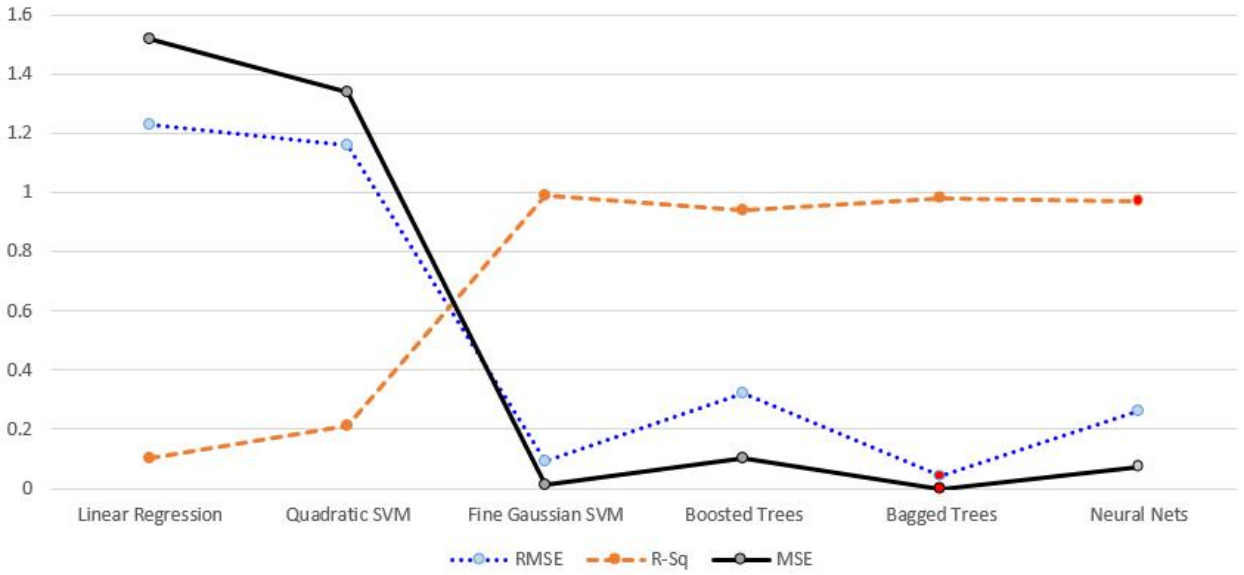


Figure 44: Performance of various models measured by comparing the predicted to the observed values, measured in terms of MSE, RMSE and R^2 using 5-fold cross validation. The model built using bagged trees provided the best performance.

4.5.4 Model Training

To train the final model, a hold-n-out scheme was used, where 6 of the data samples were retained as test data and the model was trained against the remainder of the data set. As shown in the figure below, one sample was kept out from each, straight-to-curve – South, curve-to-straight – South, straight – West, straight-to-curve – North, in-curve – North and curve-to-straight – North sections. The performance of the model against retained samples is discussed in the next chapter.

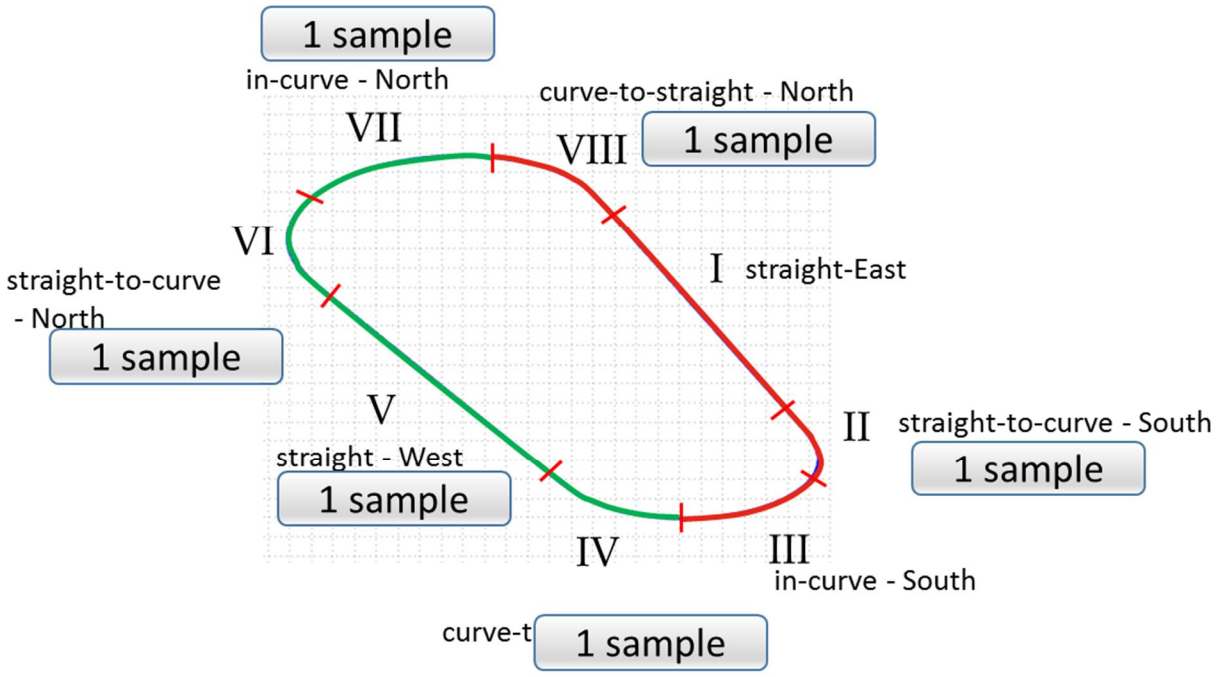


Figure 45: Six of the data samples were held back as test set, one from each of the labelled sections

4.6 Spacing Profile as a Constrained Optimization Problem

Calculating the driving path has been treated as a parameter optimization problem by several researchers [55, 56, 57]. The authors of [20] treated the ego-vehicle driving path as a cost minimization problem given a lead-vehicle's path profile. The costs defined in this research were

(1) deviation from reference path, (2) distance from obstacle and (3) lateral acceleration. See section (2.3.1) for a detailed overview of the problem definition. The minimization problem defined by these authors was re-formulated as follows:

$$minimize\ J(\ \left[\xi_{ref}{}_{k+1}\right]_{k=0}^{N}, [u_k]\ ) \tag{4-5}$$

subject to:

$$\xi_{ref}{}_{k+1} - f^{dt}\left(\xi_{ref}{}_{k}, u_k\right) = 0$$

$$d_k - d_{safety} \geq 0$$

$$|\dot{\psi}| - \frac{a_{y,max}}{V} \leq 0$$

$$\xi_{y_k} - P_{min} \geq 0$$

$$\xi_{y_k} - P_{max} \leq 0$$

Where ξ_{ref} is the reference path, $u_k$is the steering angle input, $d_k$is the distance from obstacle in path, d_{safety} is the safety distance defined from the obstacle, $\dot{\psi}$ is the vehicle yaw rate, $\frac{a_{y,max}}{V}$ is the ratio of maximum defined lateral acceleration and vehicle forward velocity, $\xi_{y_k}$is the lateral portion of the path, P_{min} is the minimum lateral position and $P_{max}$is the maximum lateral position. To solve this minimization problem, the cost function was structured as:

$$\sum_{k=0}^{N}[\xi_{path}{}_{k} + \mu_{obs}G_{obs}(d_k) + \mu_{a_y}G_{a_y}(\dot{\psi}_k) + \mu_{p_{min}}G_{p_{min}}(\xi_{y_k}) + \mu_{p_{max}}G_{p_{max}}(\xi_{y_k})] \tag{4-6}$$

where,

$$G_{obs}(d_k) = \begin{cases} d_k - d_{safety}, & if\ d_k - d_{safety} < 0 \\ 0, & otherwise \end{cases}$$

$$G_{a_y}(\dot{\psi}_k) = \begin{cases} |\dot{\psi}| - \frac{a_{y_{max}}}{V}, & if\ |\dot{\psi}| - \frac{a_{y_{max}}}{V} > 0 \\ 0, & otherwise \end{cases}$$

$$G_{p_{min}}(\xi_{y_k}) = \begin{cases} \xi_{y_k} - P_{min}, & if\ \xi_{y_k} - P_{min} < 0 \\ 0, & otherwise \end{cases}$$

$$G_{p_{max}}(\xi_{y_k}) = \begin{cases} \xi_{y_k} - P_{max}, & if\ \xi_{y_k} - P_{max} > 0 \\ 0, & otherwise \end{cases}$$

and $\mu_{obs}, \mu_{a_y}, \mu_{p_{min}}$ and $\mu_{p_{max}}$ were the weights for each of the costs.

The above reformulation of the problem was implemented in MATLAB, where *fmincon* was used to solve this multi-variable optimization problem. The simplified 2D vehicle dynamics model [20] was used to represent the vehicle plant model. The resulting spacing profile from this optimization model is discussed in the next chapter.

4.7 Chapter Summary

This chapter presented the methodology and steps performed to learn the spacing profile from human driving data. The driving task performed by the ego-vehicle driver is explained at first, followed by explanation of the various sections of the test track, where each of the two target vehicles were positioned. The several steps taken as part of the data pre-processing are then discussed including feature reduction, feature scaling and selection. A set of four features (selectattr) consisting of the ego-vehicle velocity, radius of the road at 2 secs of headway ahead of the ego-vehicle, TTC with veh_{target} and lateral position of veh_{target} in the lane were identified as the features that strongly correlated with being able to represent the ego-vehicle driver's spacing profile from veh_{target}. Several machine learning algorithms were explored to learn the

non-linear regression problem and the performance of the models developed using them was discussed.

Before training the models, six data samples were retained from the training set. Two models were developed using bagged trees and neural networks. Towards the end of this chapter, a human-like spacing profile was formulated as a constrained optimization problem. In the next chapter, the spacing profile provided by the optimization function is compared to the output of the learned model. The performance of the learned model is also discussed in the next chapter by comparing the model's predictions to observed data for the six data samples that were retained during training.

CHAPTER 5: RESULTS AND DISCUSSION

This chapter discusses how well the predicted lateral offset matched the observed values for the six data samples that were retained during training. Since RMSE provides the quality of fit by taking the difference between each of the predicted sample against the associated observed sample, time offsets between the predicted and observed values cannot be interpreted using RMSE. Two other metrics therefore are introduced before critically reviewing the prediction results. In the last section of this chapter, the optimization criteria that authors in [20] used are applied in our driving use case and the results are discussed.

5.1 Spacing Profile as a Constrained Optimization Problem versus a Learning Problem

The plots in Figure 46 show a comparison of the spacing profile derived using the formulated optimization problem (problem formulation discussed in section 4.6) shown as an *Optimized* Path, while the spacing profile model developed from learning from human driving is shown as *Predicted* Path. The reference path given to the optimization problem in this example was the lane center. In Figure 46 (top), although the optimized version navigates around the obstacle, the profile is not like the human spacing profile, shown in the figure as *Driver Path*. Recall that TTC was identified as an important predictor of the driver's spacing profile in this research (see section 4.4). The bottom plot in Figure 46 shows the optimized spacing profile with the TTC added as a cost to the optimization problem. Although the optimized response does not completely follow the human profile while returning to the lane, it follows a human-like trend closely while approaching the target vehicle. Based on the improvement in the optimized profile due to adding one feature that was identified as a signification predictor for the `human spacing

profile, it can be assumed that by updating the cost structure of the optimization problem by further applying the feature set and trends learned from human driving, we may be able to formulate the constrained optimization problem to reflect the human-like spacing more closely. Further study is needed to consider other practical factors such as the optimization model's ability to return an optimized path within a pre-defined application time.

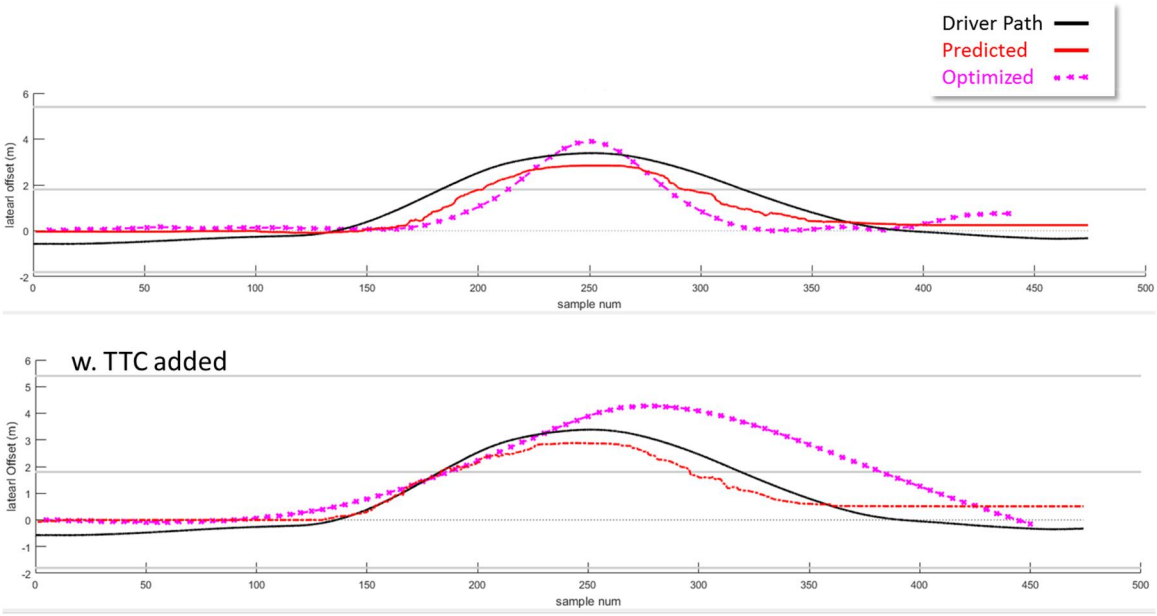


Figure 46:Comparison of optimized versus predicted path. (Top)Optimized path with center of lane as desired path, without TTC as a cost, (bottom) optimized path with center of lane as desired path with TTC included as cost.

5.2 Metrics to Measure Similarity between Predicted and Observed Data

The model performance metrics discussed thus far are evaluated by the squared error between a predicted point and its respective observed point to estimate the quality of a model's fit. To measure the similarity between two signals that may have a time lag between them, two other metrics, the cross-correlation and coefficient of variation, are introduced in this section.

These metrics will be used along with the other model performance metrics discussed in previous sections, to discuss model performance against the six data samples that were retained.

5.2.1 Cross-Correlation

Cross-correlation provides a measure of similarity between two random sequences as a function of a time lag applied to one of them. Mathematically, cross correlation is defined as [58]:

$$(f * g)(\tau) \stackrel{\text{def}}{=} f^*(t)g(t+\tau)dt \qquad [5\text{-}1]$$

where, $f^*$is the complex conjugate of f, and τ is the lag.

Cross-correlation is used in signal processing applications across many fields, from automotive applications to interpreting medical scans, from pattern matching to finding the time delay between two series [30, 31, 32]. An example of its use is in radar systems to find the time delay between the transmitted and received signal [33]. The motivation of using cross-correlation in this research is to quantify the similarity in the predicted and observed values in terms of the cross-correlation magnitude, and to determine the time delay at the maximum correlation point between the two series.

Consider the lateral offset data from one of the drive samples shown in Figure 47 (top). This reference sample $f(t)$ was shifted to the right by 300 samples, $f(t + 300)$, labelled as *ref shifted by 300*. By performing cross-correlation between these two signals, we can expect to get maximum correlation at a delay of 300, which was the delay that was added to the reference signal to create the test signal. The cross-correlation plot between these two signals using the *xcorr* command in MATLAB is shown in the bottom plot of Figure 47, where the time lag at maximum correlation can be seen to be 300.

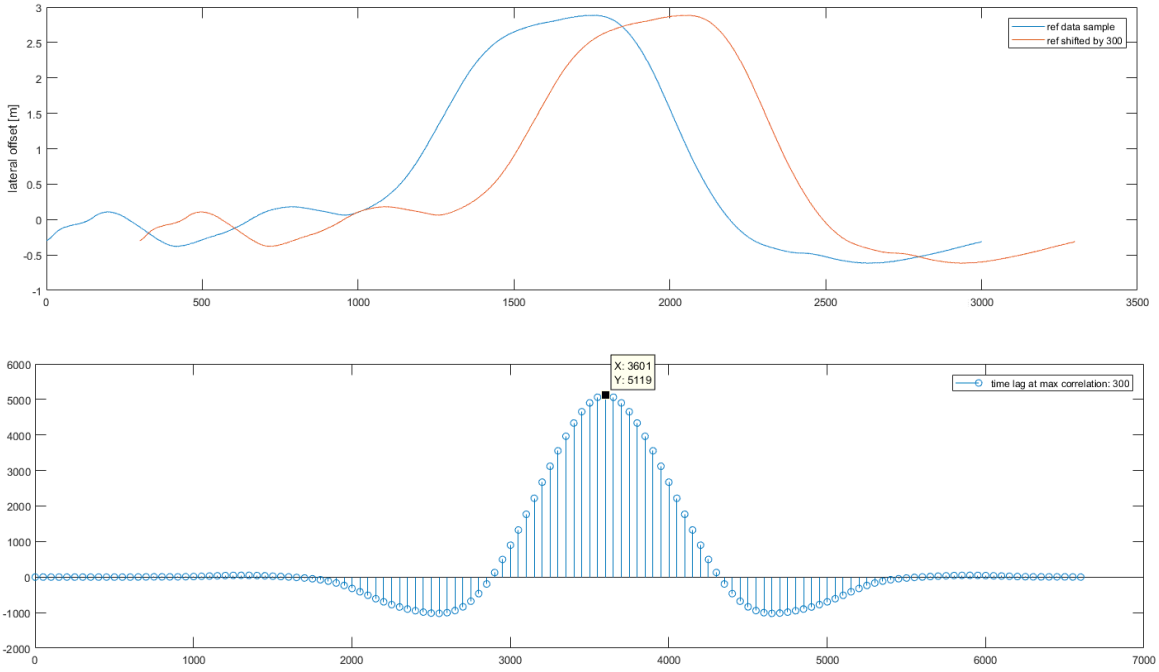


Figure 47: (top) Reference driving data sample and the same sample shifted to the right by 300 samples, (bottom) max correlation observed at time delay of 300 samples, whcih was the shift applied to the reference data.

5.2.2 Coefficient of Variation

Another estimate that provides us some insight into the similarities between two data series is the coefficient of variation (C_V). The C_V provides us with a measure of dispersion of a given data set and is the ratio of the standard deviation of the data set (s) and its mean ($\bar{x}$) [63]:

$$C_V = {}^{s}/_{\bar{x}} \qquad [5\text{-}2]$$

An important aspect to keep in mind while using C_V is that it only provides a meaningful assessment of the dispersion of data when the values of the variables being measured are positive, such that $\bar{x} > 0$ [34]. A TTC versus lateral offset plot is shown for three different data samples (6,7 and 23) in Figure 48. Each of the data samples are slightly different from the other in terms of the trend of lateral offset as it approaches and crosses the TTC of zero. It can also be seen from the plot that the dispersion of the data among all three samples is slightly different.

The C_V for the datasets 6, 7 and 23 are 0.54, 0.47 and 0.52 respectively. This tells us that the data sets 6 and 23 are relatively similar in terms of their dispersion around the mean.

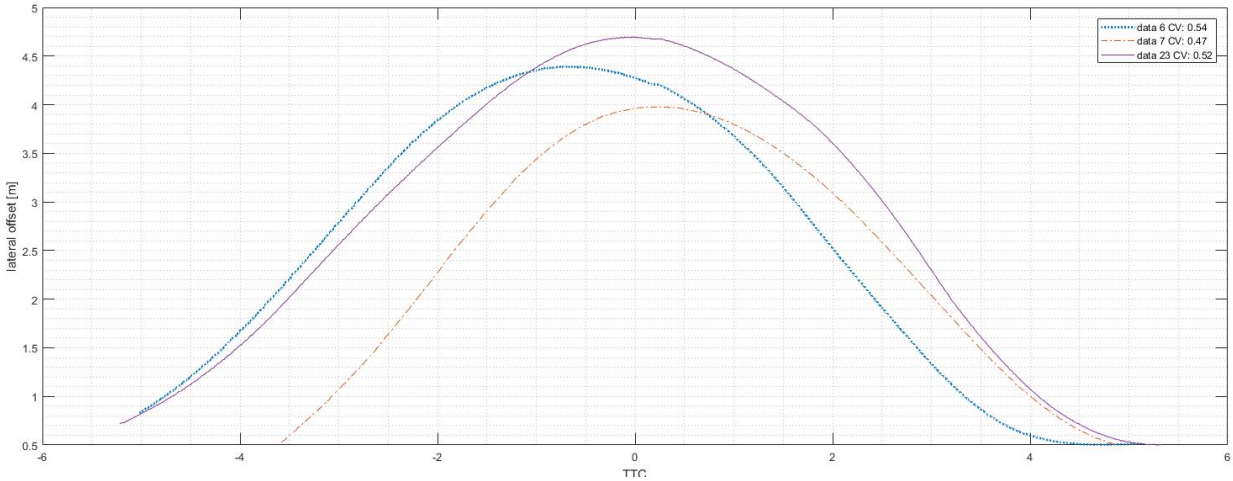


Figure 48: TTC versus lateral offset for four different drive samples. Data samples 6 and 23 have very similar trends and dispersion, hence close CV values of 0.54 and 0.52 respectively.

5.3 Performance for Retained Data Samples

The following figure (Figure 49) shows a sample plot and highlights the numerous details to help understand the information that will be presented in the plots discussed later in this section. The *x*-axis is the Time-To-Collision to the target vehicle. TTC has a positive value as the ego-vehicle approaches the target vehicle, is zero when the ego-vehicle is next to the target vehicle and has a negative value once the ego-vehicle passes the target vehicle. Mathematically, this can be written as:

$$TTC := \begin{cases} d(p_{ego}, p_{TV})/v_{rel}, & if\ \theta_{rel} < {}^{\pi}/_{2} \\ 0 \quad , & if\ \theta_{rel} == {}^{\pi}/_{2} \\ -d(p_{ego}, p_{TV})/v_{rel}, & if\ \theta_{rel} > {}^{\pi}/_{2} \end{cases} \qquad [5\text{-}3]$$

The *Y*-axis represents the lateral offset from the center of inner lane, with positive lateral offset as the vehicle moves towards the outer lane. The three horizontal lines going across the plot are the borders (lane markers) for the inner and outer lanes. The black line represents the

retained test sample, while the red and blue lines represent the predicted lateral offset of the models built using Neural Networks and Bagged Trees respectively. Recall that the test track was divided into eight sections (section 4.1.2) - the gray lines in the plot are for the lateral offset data of the remaining data samples from the road section that the retained sample was selected from.

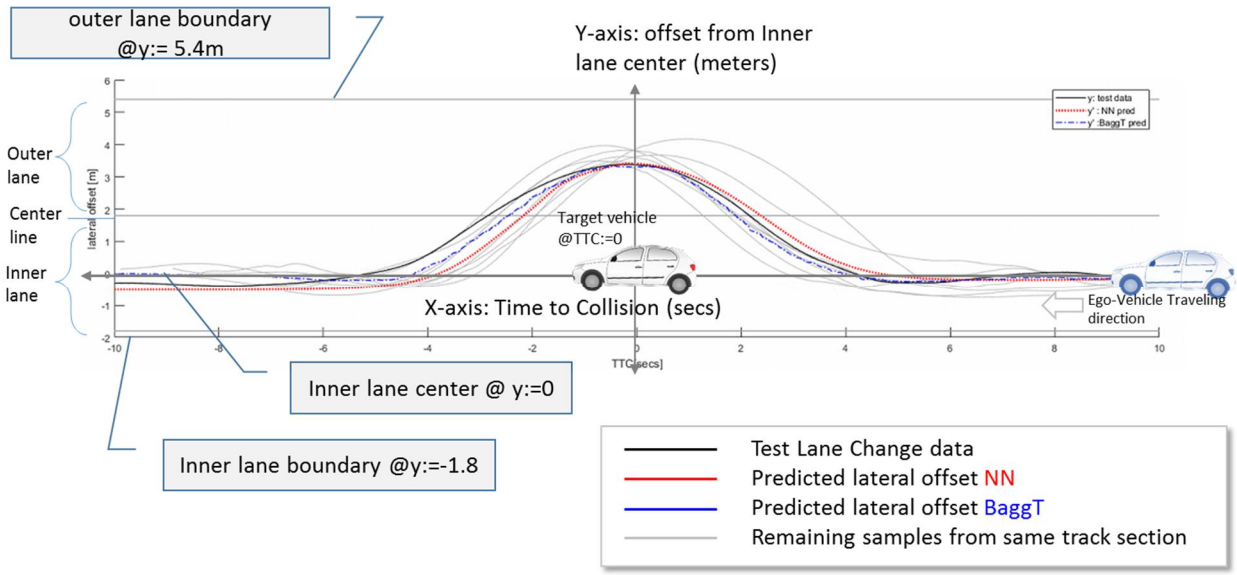


Figure 49: A detailed illustration of how to interpret the plots that will be presented later in this chapter.

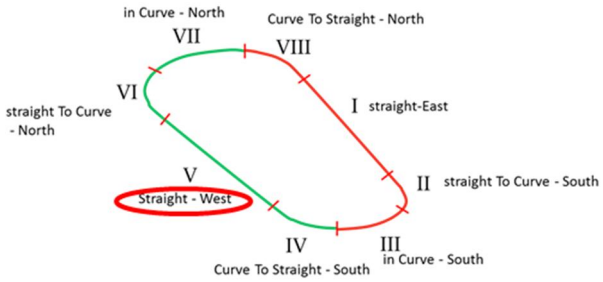
The remainder of this section discusses model performance of the six data samples that were retained during training. Under each sub-section of the retained data samples, the road section from which the sample was derived will be presented first. Two plots will be presented - the first plot, as discussed in the previous section, will show the predicted performance with the observed values, along with the observed values of all the other samples from the retained sample's road section group. The second plot will show the same predicted and observed values, but this time with *all* the data samples captured during data collection.

Due to the stochastic nature of human driving, slight variations can be observed even in the data samples from a given group. Therefore, it is not expected that the predictions would

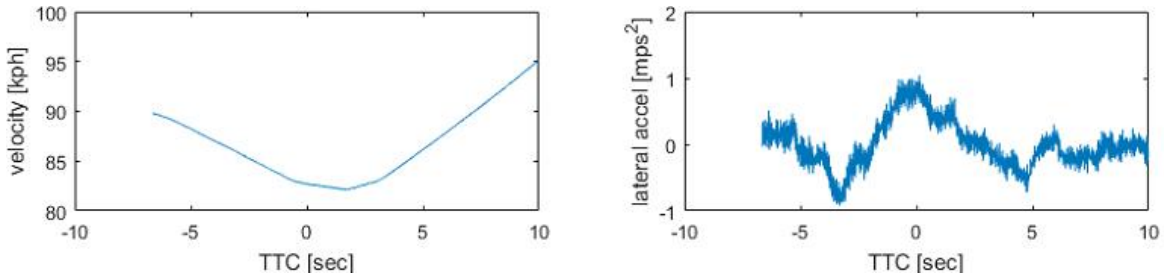
perfectly match any given observation. A reasonable approach instead would be to compare how the observed data compare against their own group and then to compare that with each of the two model predictions.

5.3.1 Test Data 1: Straight-West

(a)



(b)



(c)

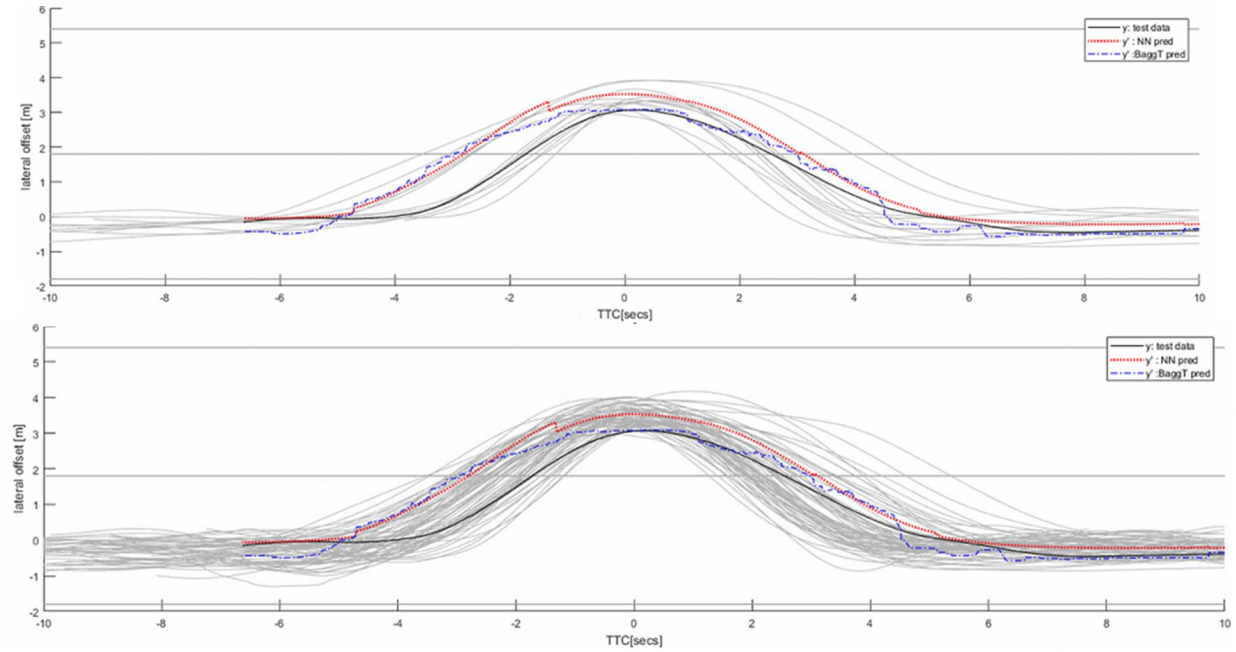


Figure 50: Test data 1 -(a)This sample was selected from straight-West, section V of the road. The ego-vehicle was driving straight while making the lane change for the target vehicle. (b) Average velocity for this sample was 86.81 KPH. Ego-vehicle velocity(left) and lateral acceleration profile (right). (c) (top) Observed versus predicted lateral offset compared with sample's own group in gray and (bottom) compared with all the data samples from all groups

5.3.2 Test Data 2: Straight-to-curve - South

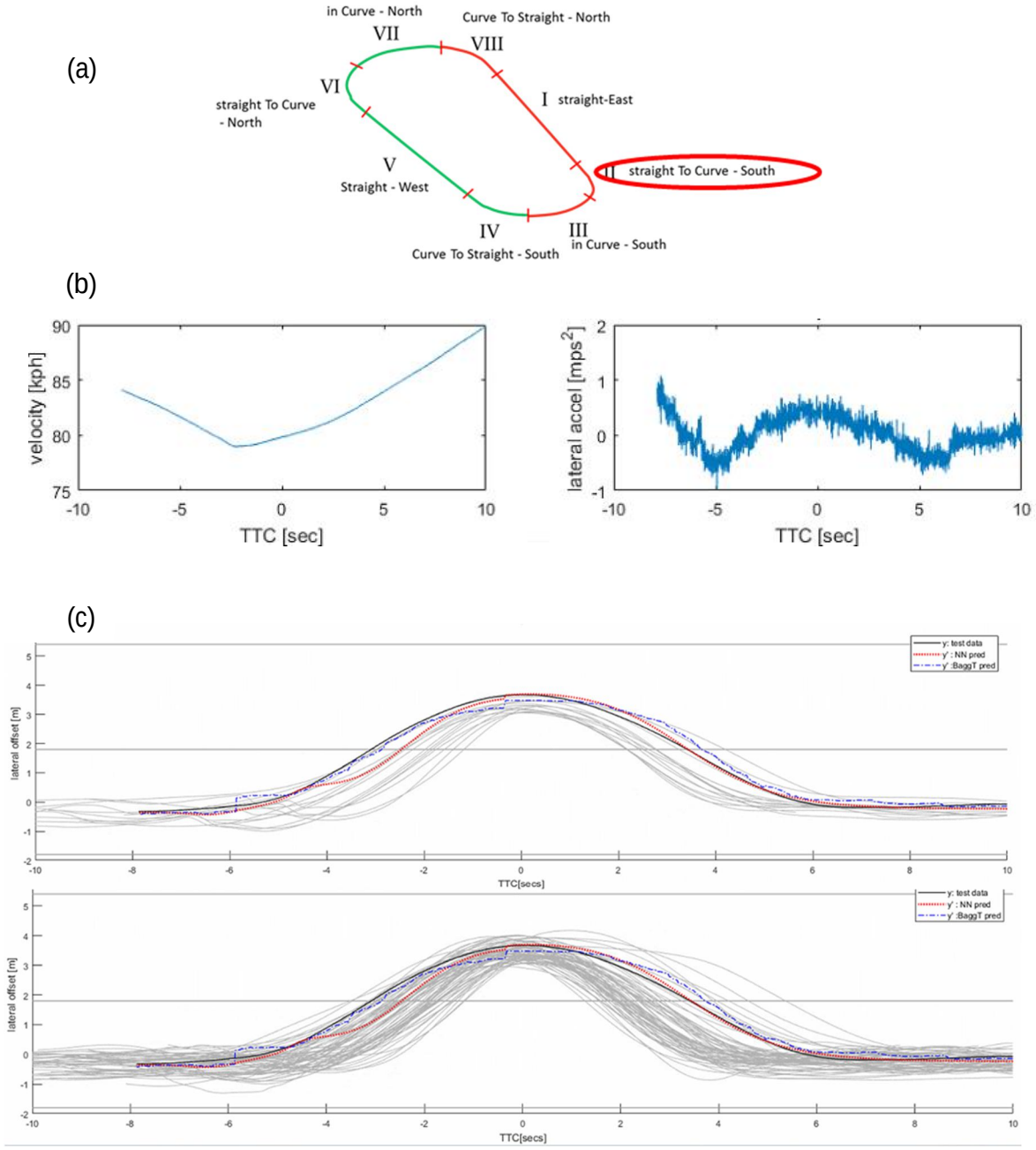


Figure 51: Test data 2 -(a)This sample was selected from straight-To-Curve, section II of the road. The ego-vehicle was getting into the curve while making the lane change to the target vehicle. (b) Average velocity for this sample was 82.89 KPH. Ego-vehicle velocity(left) and lateral acceleration profile (right). (c) (top) Observed versus predicted lateral offset compared with sample's own group in gray and (bottom) compared with all the data samples from all groups.

5.3.3 Test Data 3: curve-to-straight-North

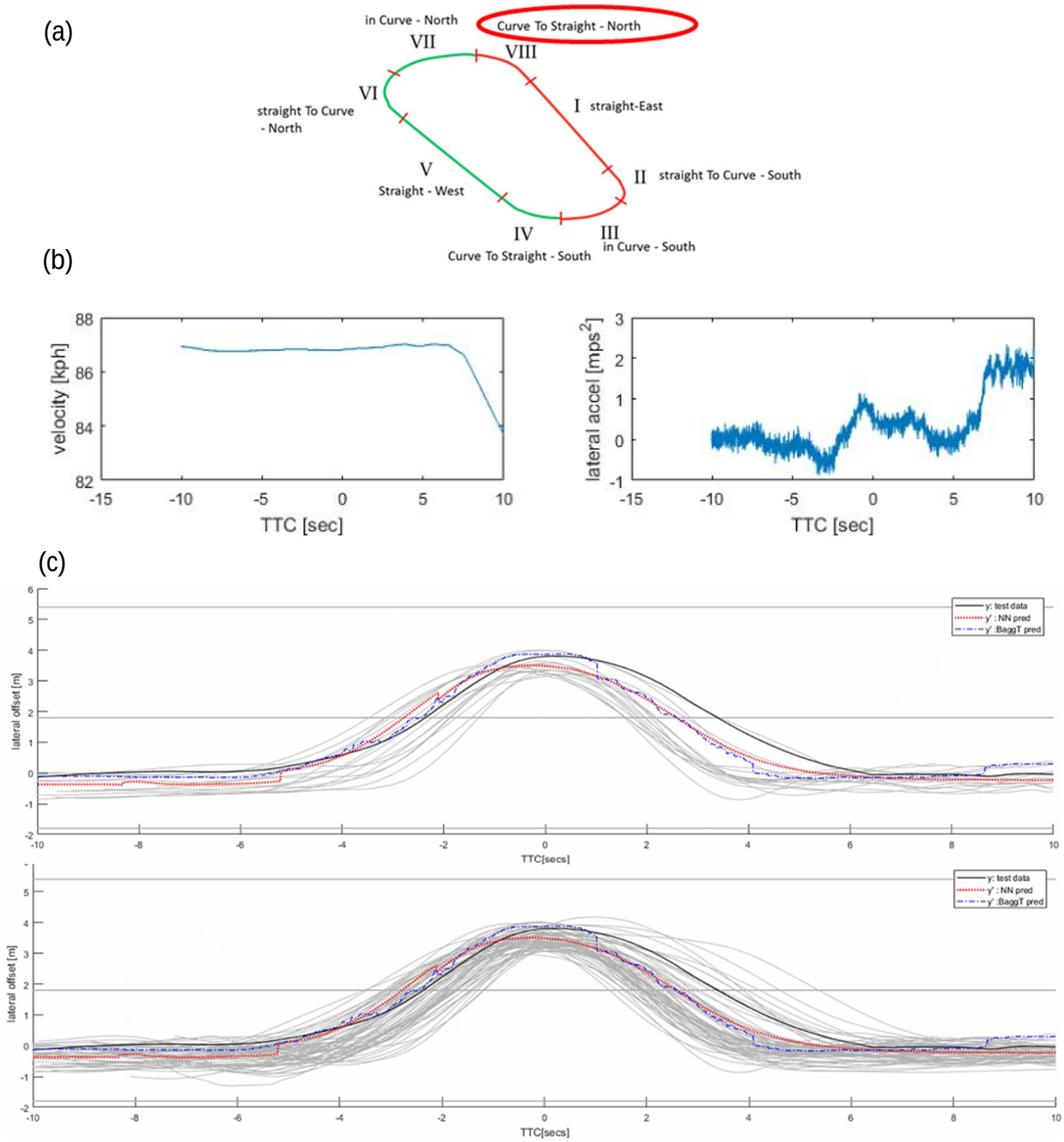


Figure 52: Test data 3 -(a)This sample was selected from Curve-To-straight, section VIII of the road. The ego-vehicle was driving out of the curve before making the lane change for the target vehicle. (b) Average velocity for this sample was 86.66 KPH. Ego-vehicle velocity(left) and lateral acceleration profile (right). (c) (top) Observed versus predicted lateral offset compared with sample's own group in gray and (bottom) compared with all the data samples from all groups.

5.3.4 Test Data 4: straight-to-curve-North

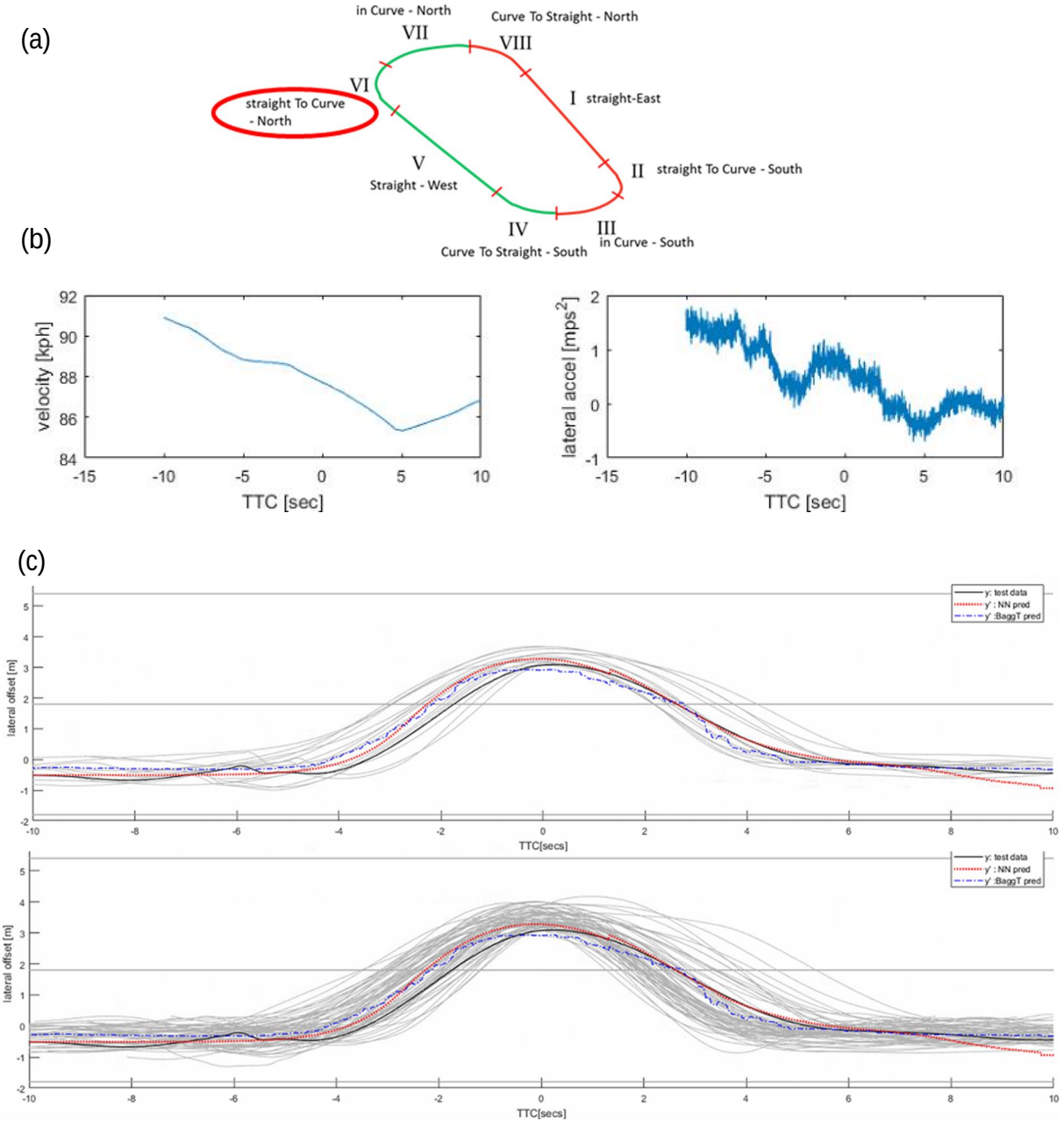


Figure 53: Test data 4 -(a)This sample was selected from the Straight-To-Curve, section VI of the road. The ego-vehicle was driving into the curve while making the lane change for the target vehicle. (b) Average velocity for this sample was 87.75 KPH. Ego-vehicle velocity(left) and lateral acceleration profile (right). (c) (top) Observed versus predicted lateral offset compared with sample's own group in gray and (bottom) compared with all the data samples from all groups.

5.3.5 Test Data 5: in-curve-North

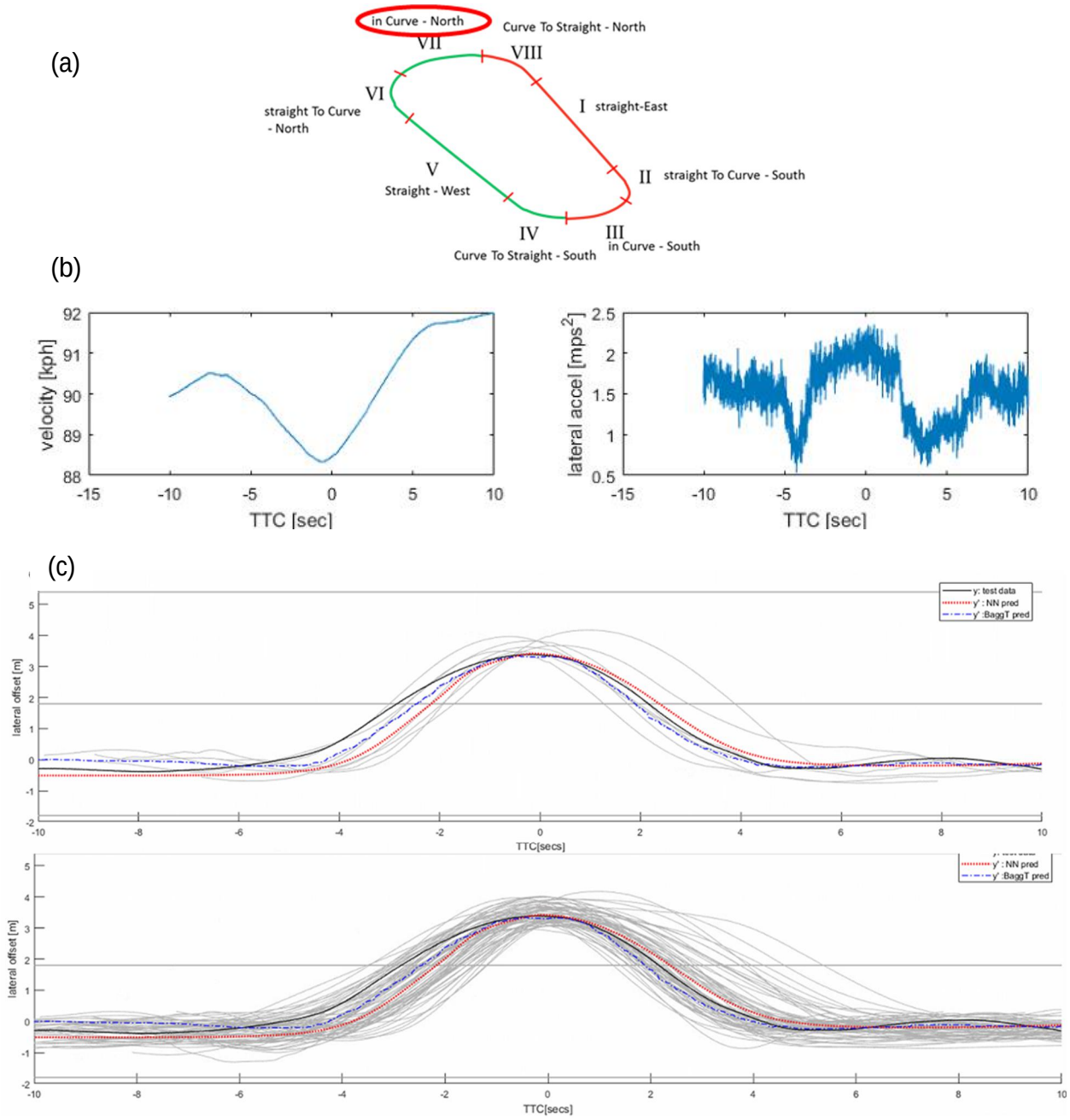


Figure 54: Test data 5 -(a)This sample was selected from the in-curve, section VII of the road. The ego-vehicle was driving into the curve while making the lane change for the target vehicle. (b) Average velocity for this sample was 90.24 KPH. Ego-vehicle velocity(left) and lateral acceleration profile (right). (c) (top) Observed versus predicted lateral offset compared with sample's own group in gray and (bottom) compared with all the data samples from all groups.

5.3.6 Test Data 6: curve-to-straight-South

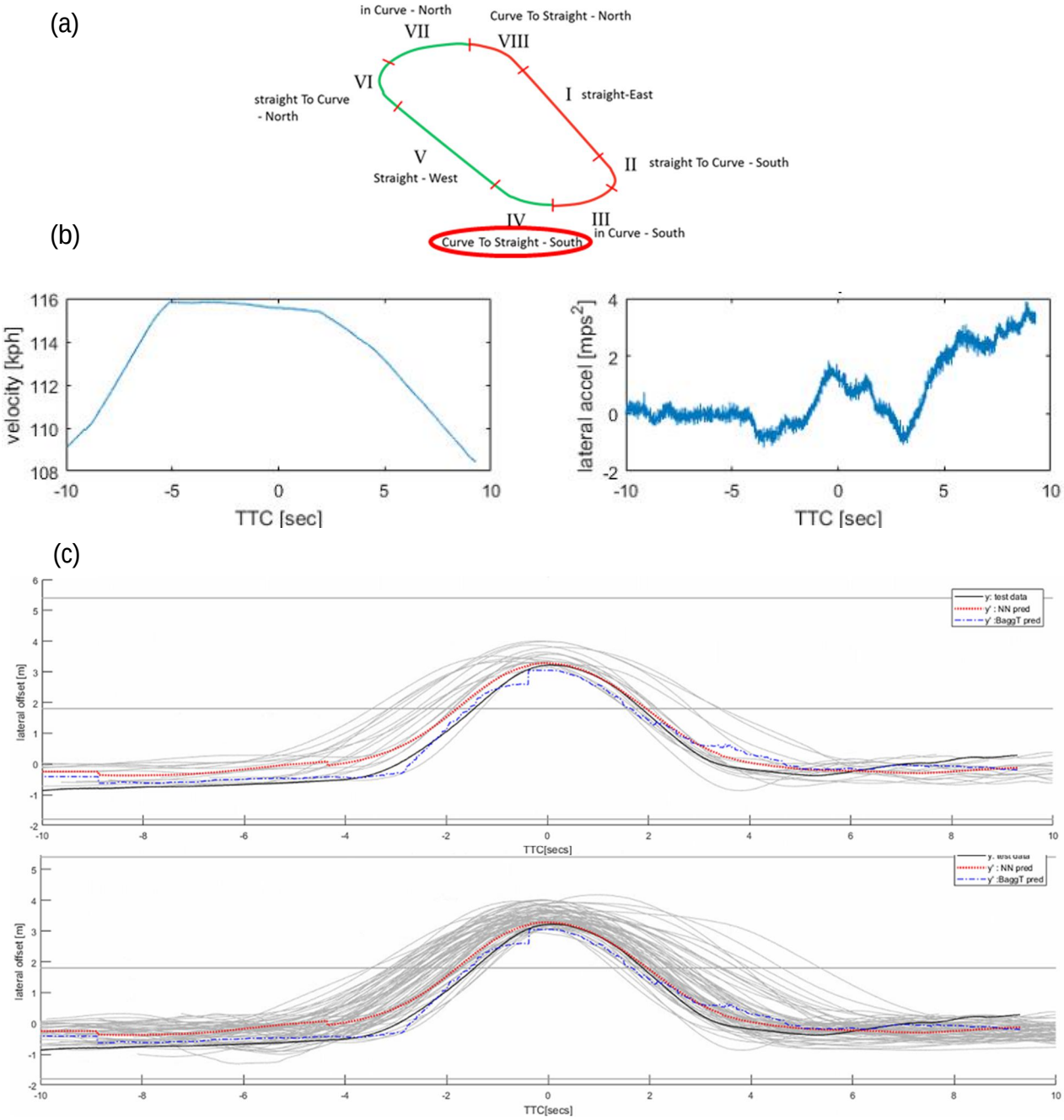


Figure 55: Test data 6 -(a)This sample was selected from the curve-to-straight, section IV of the road. The ego-vehicle was driving along the curve while making the lane change for the target vehicle. (b) Average velocity for this sample was 113.58 KPH. Ego-vehicle velocity(left) and lateral acceleration profile (right). (c) (top) Observed versus predicted lateral offset compared with sample's own group in gray and (bottom) compared with all the data samples from all groups.

5.4 Discussion on Model Performance for Test Data

The previous section illustrated the model performance against the six test data samples. The test data were within the speed range of approximately 82 KPH to 113 KPH and included driving on straight as well as curved sections of the road. As shown in Figure 50 through Figure 55, both the Neural Network and Bagged Trees model performed very well in predicting the driver's lateral offset. Figure 56 provides a glimpse into the quantitative analysis of the quality of fit in terms of RMSE. Most of the RMSE values are well under 50cm except for sample number 1, where RMSE for the Neural Network-based model was 54cm.

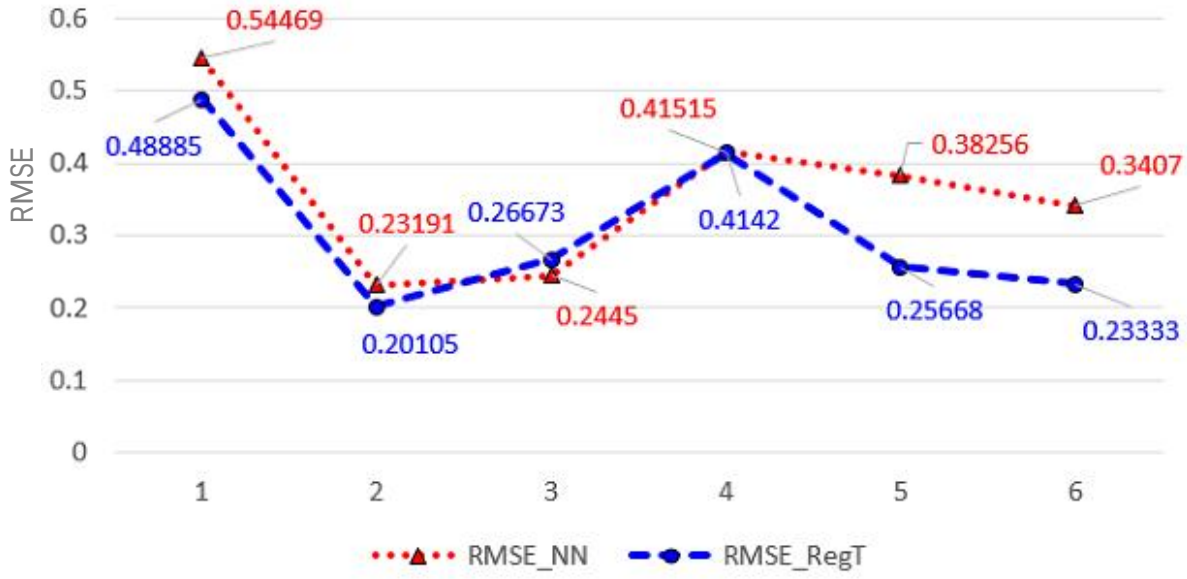


Figure 56: Model performance against the six test samples in terms of RMSE. For sample 1, although the RMSE is slightly higher than the RMSE for the remaining samples, the predictions are still well within the bounds of the lateral offset of remaining data sets within that same group.

As discussed earlier, there was variation in the lane change maneuver performed by the driver even when driving through the same area multiple times. It therefore is more appropriate to consider the performance with the remaining of lane change samples within the group of

sample 1 in addition to the RMSE values. Figure 50 (c, top) shows the predicted lateral offset along with the lateral offset for the remainder of that group. The predictions can be seen to be well within the bounds of the remaining data samples with this group.

In some cases, the predicted lane change started slightly earlier than the test sample. It is therefore important to consider the cross correlation along with the time delay between the predicted and observed lateral offset trend. The cross correlation between the predicted and observed values for the test samples is presented in Figure 57.

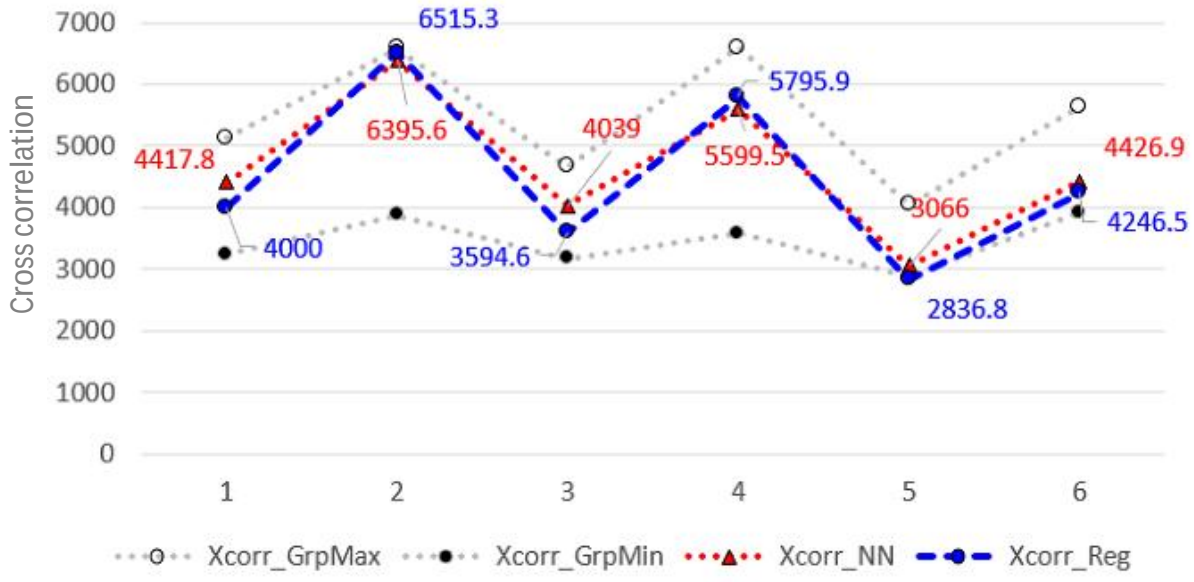


Figure 57: Cross correlation of the predicted lateral offset and the test sample shown as Xcorr_NN and Xcorr_Reg for neural network and bagged trees models respectively. Additionally, the maximum and minimum values for the cross-correlation of the test sample is presented with its own group data set given as Xcorr_GrpMax and Xcorr_GrpMin respectively.

To determine whether these cross-correlation values are within acceptable ranges, the maximum and minimum cross-correlation values between the test sample and its group is also presented in Figure 57 as Xcorr_GrpMax and Xcorr_GrpMin respectively. All the cross-correlation values from both prediction models were observed to be within the bounds of the cross-correlation of the test sample with its own group.

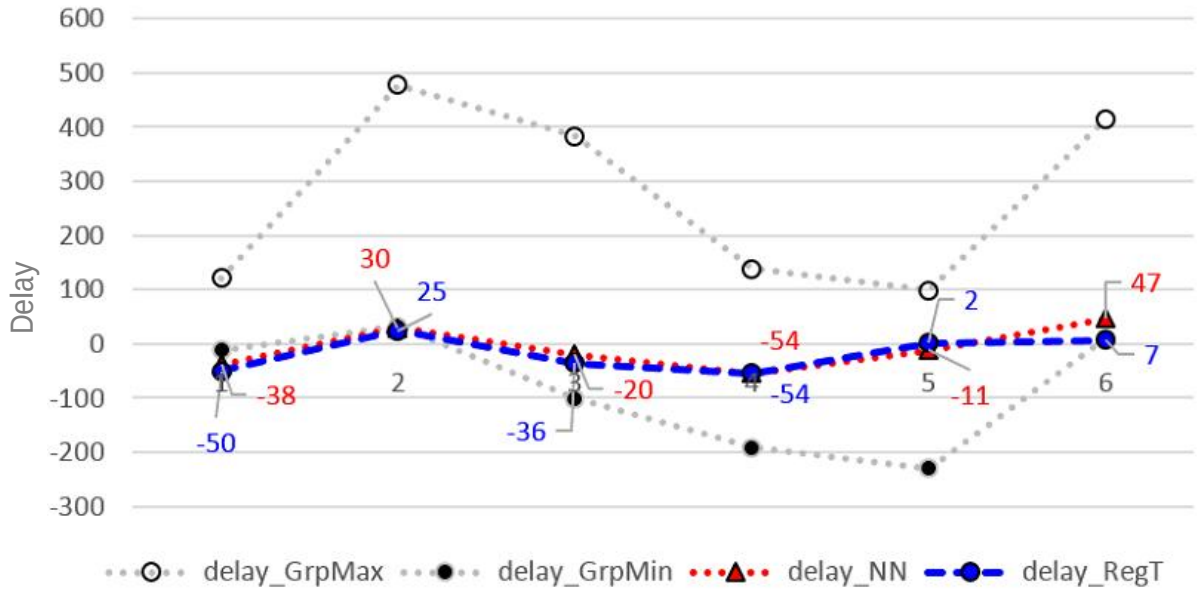


Figure 58: Delay in samples, calculated based on maximum cross correlation between the predicted and observed data for models by Neural Network and Bagged Trees shown as delay_NN and delay_Reg respectively. In comparison, the delay measured between the test sample and the rest of its group members is presented as delay_GrpMax and delay_GrpMin for the maximum and minimum delays respectively.

The delay in data trends between the predicted and observed values can be measured in terms of offset of the maximum cross-correlation values. These delays are presented for each of the two models (as delay_NN and delay_RegT) along with the maximum and minimum delays of the test sample observed within its own group (as delay_Grpmax and delay_GrpMin) in Figure 58. The predicted responses can be seen to be closer to zero much better than the delay comparison of the test sample within its own group. It can therefore be deduced that the predicted spacing profile trends were similar to those of the test samples.

5.5 Conclusion

In this research, the spacing profile of human drivers in terms of distance from other vehicles was formulated as a learning problem. The spacing profile was defined in terms of lateral offset from the lane center and the longitudinal spacing from target vehicles was defined in terms of Time-To-Collision. Due to the stochastic nature of human driving and possibly a large variety of factors that can influence human driving decisions, this study focused on a specific use case of passing a stationary vehicle in the ego-vehicle's lane.

The ego and target vehicles were instrumented to measure the position and motion profiles of each of the vehicles. A total of 95 samples of human driving data were captured, each of which included driving in the inner lane, making a lane change as a stationary target vehicle was encountered in the inner lane and returning to the inner lane of a test track. Machine learning techniques were used to identify a feature set that strongly represented the driver's spacing profile while passing target vehicles. This feature set consisted of the following four features: 1) radius of the road two seconds ahead of the ego-vehicle, 2) ego-vehicle's velocity, 3) Time-To-Collision with the target vehicle and 4) target vehicle's lateral position in the lane.

Two models were developed, one using Neural Networks and the other using Bagged Trees. The predicted values of the models were within statistical significance defined in terms of RMSE and cross-correlation. A comparison was done by formulating the spacing profile as an optimization problem, where the objective was to minimize the distance to lane center, while keeping a safe distance from the target vehicle and keeping the lateral acceleration within a pre-defined maximum lateral acceleration. Although the optimized path went around the obstacle, the profile trend did not follow the human driving data well.

Since Time-To-Collision was found to be a strong predictor in this study, it was added as a cost to the earlier defined optimization problem. The optimized spacing profile with Time-To-Collision added as a cost to the optimization problem represented the human spacing profile much better than the optimized profile that did not include the TTC. The optimized profile output with TTC was more like the human spacing profile prior to crossing the target vehicle. After passing the target vehicle, while the optimized spacing profile was within the defined criteria, it deviated more from the human sample. Although we might be able to achieve reasonable results in this driving use case by further tuning the optimization model parameters, it becomes impractical to continue to refine the cost functions and associated weights as we expand the driving scenarios. On the other hand, learning from the demonstration framework allows us to continue to expand the model's capability by re-training it with the expanded driving scenario data set.

This research provided an end-to-end methodology concerning how to learn a human-like driving style using learning from demonstration techniques. Although the use case focused on in this research is a small percentage of overall driving use cases, the methodology presented here can be extended to learning a human driving style in a large variety of use cases. The predictor set identified in this research consists of data that can be gathered in current production vehicles that are equipped with ADAS features, providing a rather inexpensive way to capture data and implement human-like spacing profiles in today's production vehicles.

CHAPTER 6: RESEARCH CONTRIBUTIONS AND FUTURE WORK

6.1 Summary of Contributions

Learning from demonstration techniques is becoming more prevalent in the automated driving industry. For human drivers to yield control to automation, for the occupants to feel safe and comfortable and for other human drivers to drive with mixed automated traffic, the automated vehicles will need to think and act like human drivers. As drivers, we each have a different driving style and comfort preference in many different aspects of driving. This research focused on learning how human drivers space themselves while approaching and passing other stationary vehicles.

There are three main novel and key technical contributions of this research: First, the methodology and framework laid out in this research using learning from demonstration techniques to learn a driver's spacing profile in the presence of other obstacles is the first to the best of the author's knowledge. Several researchers have attempted to learn how human drivers position themselves in a lane, but in an obstacle-free environment. This research lays out the methodology from problem definition, to vehicle instrumentation and data collection, followed by using machine learning techniques to develop models that can present a human driving style in the presence of other vehicles. Although this research focused on driving around stationary vehicles, the methodology and framework can be extended to include many other driving scenarios.

The second technical contribution of this research is in identifying a combination of features that represents a driver's spacing profile from other stationary vehicles. This is a

significant contribution as this is the first research that ties the identified feature set to human driving data in terms of spacing profile. While formulating the problem, a special focus was given to information that could be made readily available with the sensing technology available in today's vehicles. Each of the identified features can be captured using in-vehicle cameras, radars, a GPS system and maps, all of which are either available in production vehicles as of this writing or are expected to be available in the very near future.

The third contribution of this research was the development of a spacing profile model that can represent a driver's spacing profile. Although other researchers have attempted to learn driving style from human driving in a traffic-free environment, to the best of the author's knowledge, this is the first model that can provide a human-like spacing profile in the presence of other vehicles. The learned model is able to predict a human-like spacing profile, while driving at highway speeds, on straight as well as curved roads and in the presence of other stationary vehicles.

6.2 Limitations of this Research and Future Work

While this research laid out the ground work in presenting a human-like profile selection in the presence of other obstacles, it had a few limitations which will be the next focus of this research. Some of the limitations of this research included driving on a test track only, using driving data from a single driver only and in a limited driving scenario, i.e. only considering stationary target vehicles. The use of a test track for this study was primarily safety driven. A test track provides an ideal environment for such studies that cannot be conducted safely on a public road. The ability to park target vehicles in the middle of a lane on a public road is impractical and unsafe. This safety concern is no longer relevant as we start considering the use cases that involve

some of the typical non-stationary target vehicle scenarios. Another motivation for selecting the test track for this study was instrumentation; the RT system that was used in this study can provide positional information accurate to within a few centimeters in the presence of a local base station (see section 3.1.3.2 for background on RTK). The FTTA test track had a local base station installed near the test track. Using this track therefore allowed us to obtain RTK-level positional accuracy.

To expand this study with other use cases on public roads, the instrumentation to acquire the ego and target vehicle pose and motion information must be revisited as it is not practical to equip every target vehicle with an RT system. Many auto manufacturers that are releasing vehicles with active safety features such as adaptive cruise control and lane keep assist have camera systems [64, 65]. These camera systems are capable of estimating ego-vehicle position in a lane along with estimating the relative position and motion of other vehicles. Information from such a camera system with an on-board navigation system will provide comparable information to the features gathered using the RT system in this study. By replacing a specialized instrumentation like the RT system used in this study with production sensors readily available in a vehicle, this study can be expanded to gathering data from multiple drivers from different vehicles. That would allow us to develop a more generalizable spacing profile model.

While this study looked at stationary target vehicles only, it allowed for the formulation of a methodology that demonstrates the entire process from problem definition to vehicle instrumentation, data collection and feature identification, finally leading towards developing a model that reflects a human-like spacing profile. This methodology can be extended to learning

many other aspects of human driving, for example a human driver's ability to estimate risk based on the motion trend of other vehicles observed over time, among others.

APPENDIX A: VEHICLE KINEMATIC MODEL

Kinematic model of a vehicle is a mathematical representation derived solely based on geometric relation of vehicle's motion, without any considerations to the forces that affect it. Consider the model of vehicle known as bicycle model shown in figure below.

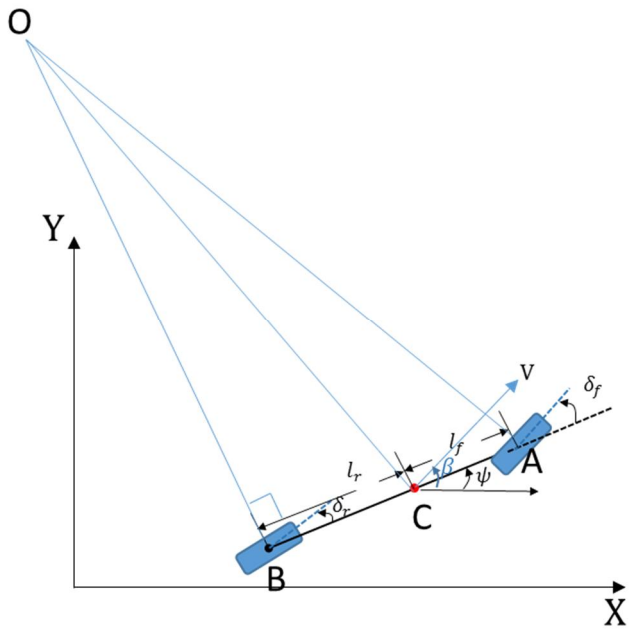


Figure 59: Bicycle model used to derive kinematics of vehicle motion

This is a single-track vehicle model, where two front wheels are combined and represented as one at A and similarly the two rear wheels are combined and represented at B. This assumes that both front wheels turn at the same angle, which is a simplification of how actual wheels are designed as these angles are not exactly the same and path travelled by the two is of different radius as well. in an actual vehicle. Assume that Center of Gravity (CoG) of vehicle is at C. Distance to the front and rear axle from CoG is represented as l_f and l_r respectively. Global 2D coordinates are represented as (X,Y). Vehicle longitudinal axis makes an angle ψ with X-axis. The resultant velocity at vehicle CoG V makes an angle β with vehicle

longitudinal axis, known as vehicle slip angle. Front and rear wheel angles are represented as δ_f and δ_r respectively. Assume that vehicle is taking a turn of radius R (OC), with instantaneous rolling center O.

This bicycle model assumes that the velocity vector at front and rear wheel is in the direction of wheel orientation, which implies that front and rear wheel slip angle is zero. This assumption only holds true when lateral forces generated at tires is small, which is valid when driving at low speed ($< 5\, m/s$). Assuming a constant turning radius, the lateral force acting on front and rear tires can be expressed as:

$$F_{lat} = ma = m\frac{V^2}{r}$$

The angle that vehicle makes traversing the course, γ can be represented as sum of vehicle slip angle and heading;

$$\gamma = \beta + \psi$$

Using law of sines on triangle ΔOCB;

$$\frac{\sin(\beta - \delta_r)}{l_r} = \frac{\sin\left(\frac{\pi}{2} + \delta_r\right)}{R} \qquad [1]$$

where, $\sin(\beta - \delta_r) = \sin(\beta)\cos(\delta_r) - \cos(\beta)\sin(\delta_r)$

and, $\sin\left(\frac{\pi}{2} + \delta_r\right) = \cos(\delta_r)$

Replacing the above in equation 1 would yield;

$$\frac{sin(\beta)\, cos(\delta_r) - cos(\beta)\, sin(\delta_r)}{l_r} = \frac{cos(\delta_r)}{R}$$

$$\Leftrightarrow \frac{sin(\beta)\, cos(\delta_r) - cos(\beta)\, sin(\delta_r)}{l_r} \cdot \frac{l_r}{cos(\delta_r)} = \frac{cos(\delta_r)}{R} \cdot \frac{l_r}{cos(\delta_r)}$$

By expanding and simplifying the above equation, we get,

$$\frac{\sin(\beta)\cos(\delta_r)}{\cos(\delta_r)} - \frac{\cos(\beta)\sin(\delta_r)}{\cos(\delta_r)} = \frac{l_r}{R}$$

$$\Leftrightarrow \sin(\beta) - \tan(\delta_r)\cos(\beta) = \frac{l_r}{R} \quad [02]$$

Similarly, by using law of sines for ΔOCA;

$$\frac{\sin\left(\frac{\pi}{2} - \delta_f\right)}{R} = \frac{\sin(\delta_f - \beta)}{l_f} \quad [3]$$

where, $\sin(\delta_f - \beta) = \sin(\delta_f)\cos(\beta) - \sin(\beta)\cos(\delta_f)$

$$\sin\left(\frac{\pi}{2} - \delta_f\right) = \cos(\delta_f)$$

Replacing them in equation 3,

$$\frac{\cos(\delta_f)}{R} = \frac{\sin(\delta_f)\cos(\beta) - \sin(\beta)\cos(\delta_f)}{l_f}$$

$$\Leftrightarrow \frac{\sin(\delta_f)\cos(\beta) - \sin(\beta)\cos(\delta_f)}{l_f} \cdot \frac{l_f}{\cos(\delta_f)} = \frac{\cos(\delta_f)}{R} \cdot \frac{l_f}{\cos(\delta_f)}$$

By expanding and simplifying the above equation, we get,

$$\frac{\sin(\delta_f)\cos(\beta)}{\cos(\delta_f)} - \frac{\sin(\beta)\cos(\delta_f)}{\cos(\delta_f)} = \frac{l_f}{R}$$

$$\Leftrightarrow \tan(\delta_f)\cos(\beta) - \sin(\beta) = \frac{l_f}{R} \quad [4]$$

By adding equations 2 and 4, we get

$$\sin(\beta) - \tan(\boldsymbol{\delta_r})\cos(\beta) = \frac{l_r}{R}$$

$$-\sin(\beta) + \tan(\delta_f)\cos(\beta) = \frac{l_f}{R}$$

$$\Rightarrow \tan(\delta_f)\cos(\beta) - \tan(\delta_r)\cos(\beta) = \frac{l_f + l_r}{R} \quad [5]$$

$$\Leftrightarrow \{\tan(\delta_f) - \tan(\delta_r)\}\cos(\beta) = \frac{l_f + l_r}{R}$$

$$\Leftrightarrow R = \frac{l_f + l_r}{\{\tan(\delta_f) - \tan(\delta_r)\}\cos(\beta)} \quad [6]$$

Rate of change of vehicle orientation on a curve can be given by,

$$\dot{\psi} = \frac{V}{R}$$

Replacing R by equation 1-5 would yield,

$$\dot{\psi} = \frac{V\{\tan(\delta_f) - \tan(\delta_r)\}\cos(\beta)}{l_f + l_r} \quad [7]$$

In order to derive a relation for side slip angle, β, multiplying equations [2] and [4] by l_f and l_r respectively, followed by subtracting resulting equation [4] from [2] as following:

$$\{\sin(\beta) - \tan(\boldsymbol{\delta_r})\cos(\beta)\}.\, l_f = \frac{l_r}{R}.\, l_f$$

$$\underset{-}{\underbrace{+}}\left\{\tan(\delta_f)\cos(\beta) \underset{+}{\underbrace{-}} \sin(\beta)\right\}.\, l_r = \underset{-}{\underbrace{+}}\frac{l_f}{R}.\, l_r$$

$$(l_f + l_r)\sin(\beta) - (\tan(\delta_r)\cos(\beta) + \tan(\delta_f)\cos(\beta)) = 0$$

$$\Leftrightarrow (l_f + l_r)\sin(\beta) = (\tan(\delta_r)\cos(\beta) + \tan(\delta_f)\cos(\beta))$$

$$\Leftrightarrow (l_f + l_r)\sin(\beta) = \{\tan(\delta_r) + \tan(\delta_f)\}\cos(\beta)$$

$$\Leftrightarrow (l_f + l_r)\tan(\beta) = \{\tan(\delta_r) + \tan(\delta_f)\}$$

$$\Leftrightarrow \tan(\beta) = \frac{\{\tan(\delta_r) + \tan(\delta_f)\}}{(l_f + l_r)}$$

$$\Leftrightarrow \beta = \tan^{-1} \frac{\{\tan(\delta_r) + \tan(\delta_f)\}}{(l_f + l_r)} \quad [8]$$

As noted earlier, the relationships for rate of change of heading (also referred to as '*yaw rate*') $\dot{\psi}$ and side –slip angle β derived in equations [6] and [7] are solely based on geometric relations and therefore are beneficial in representing these two terms in situations where additional frictional forces and disturbances are at minimum. This is true only at low driving speeds. Similarly, the assumption that velocity at each wheel is in the direction of each wheel is no longer valid at higher speeds. Therefore, a vehicle model based on the dynamics of vehicle motion thus has to be derived to represent driving at higher speeds.

APPENDIX B: LATERAL VEHICLE DYNAMICS MODEL

A bicycle model shown in Figure 59 is the basis for deriving lateral dynamics model in this section [37]. The two degrees of freedom considered for deriving this model are the lateral position of vehicle (Y) and heading(ψ). Applying Newton's second law of motion along vehicle Y-axis would yield,

$$ma_y = F_{yf} + F_{yr} \quad [0\text{-}1]$$

Where m is vehicle mass in kg, a_y is the inertial acceleration along y-axis in m/s^2 acting at vehicle CoG, F_{yf} and F_{yf} represent lateral forces in N acting on front and rear tires respectively. Solving for moment balance about z-axis would yield,

$$I_z\ddot{\psi} = l_f F_{yf} - l_r F_{yr} \quad [0\text{-}2]$$

Where I_z is the moment of inertia along z-axis, l_f and l_r are distances in m from CoG to the front and rear axle respectively. The forces acting on front and rear tires are found to be proportional to the tire slip angles for low slip-angles. Consider front wheel angle δ_f and θ_{Vf}, the angle that front wheel makes with vehicle longitudinal axis as. Front wheel side slip angle α_f can be written as,

$$\alpha_f = \delta_f - \theta_{Vf} \quad [0\text{-}3]$$

The lateral tire forces on the front wheel can therefore be written as,

$$F_{yf} = 2C_{\alpha f}\alpha_f \quad [0\text{-}4]$$

$$\Leftrightarrow F_{yf} = 2C_{\alpha f}(\delta_f - \theta_{Vf}) \quad [0\text{-}5]$$

Where, $C_{\alpha f}$ is the cornering stiffness of front tire. The factor of 2 is to account for the two front tires.

Assuming a non-steering rear wheel, $\delta_r \approx 0$. Rear tire side-slip can similarly be represented as,

$$\alpha_r = -\theta_{Vr} \quad [0\text{-}6]$$

The lateral tire forces on rear tires can be given as,

$$F_{yr} = 2C_{\alpha r}\alpha_r \quad [0\text{-}7]$$

$$F_{yr} = 2C_{\alpha r}(-\theta_{Vr}) \quad [0\text{-}8]$$

where,$C_{\alpha r}$ is the cornering stiffness of rear tires. Factor of 2 is to account for the two rear tires.

θ_{Vf} and θ_{Vr}can be calculated by the following relationship,

$$tan(\theta_{Vf}) = \frac{V_y + l_f\dot{\psi}}{V_x}$$

using small angle approximation,

$$\Rightarrow \theta_{Vf} = \frac{V_y + l_f\dot{\psi}}{V_x} \quad [0\text{-}9]$$

$$tan(\theta_{Vr}) = \frac{V_y - l_r\dot{\psi}}{V_x}$$

using small angle approximation,

$$\Rightarrow \theta_{Vr} = \frac{V_y - l_r\dot{\psi}}{V_x} \quad [0\text{-}10]$$

Substituting from [0-9] and [0-10] into [0-8] and [0-5] respectively would yield,

From eq. [0-5] we get, $F_{yf} = 2C_{\alpha f}(\delta_f - \theta_{Vf})$

$$\Longrightarrow F_{yf} = 2C_{\alpha f}\left(\delta_f - \frac{V_y + l_f\dot{\psi}}{V_x}\right)$$

$$\Leftrightarrow F_{yf} = 2C_{\alpha f}\delta_f - \frac{2C_{\alpha f}}{V_x}V_y - \frac{2C_{\alpha f}l_f}{V_x}\dot{\psi} \quad [0\text{-}11]$$

Similarly from eq. $F_{yr} = 2C_{\alpha r}(-\theta_{Vr})$ [0-8],

$$\Longrightarrow F_{yr} = 2C_{\alpha r}\left(-\frac{V_y - l_r\dot{\psi}}{V_x}\right)$$

$$\Leftrightarrow F_{yr} = -\frac{2C_{\alpha r}}{V_x}V_y + \frac{2C_{\alpha r}l_r}{V_x}\dot{\psi} \qquad [0\text{-}12]$$

From equation [0-1], we have the following relation,

$$ma_y = F_{yf} + F_{yr}$$

Substituting [0-11] and [0-12] in the above equation,

$$ma_y = 2C_{\alpha f}\delta_f - \frac{2C_{\alpha f}}{V_x}V_y - \frac{2C_{\alpha f}l_f}{V_x}\dot{\psi} - \frac{2C_{\alpha r}}{V_x}V_y + \frac{2C_{\alpha r}l_r}{V_x}\dot{\psi}$$

$$ma_y = 2C_{\alpha f}\delta_f - \frac{2C_{\alpha f} + 2C_{\alpha r}}{V_x}V_y - \frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{V_x}\dot{\psi}$$

Inertial acceleration a_y is influenced by lateral motion acceleration $\ddot{y}$ and centripetal acceleration $V_x\psi$, i.e. $a_y = \ddot{y} + V_x\psi$. Replacing $a_y$in the above equation with this expression and changing V_y to $\dot{y}$;

$$m(\ddot{y} + V_x\psi) = 2C_{\alpha f}\delta_f - \frac{2C_{\alpha f} + 2C_{\alpha r}}{V_x}\dot{y} - \frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{V_x}\dot{\psi}$$

$$\ddot{y} + V_x\psi = \frac{2C_{\alpha f}\delta_f}{m} - \frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x}\dot{y} - \frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{mV_x}\dot{\psi}$$

$$\ddot{y} = \frac{2C_{\alpha f}\delta_f}{m} - \frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x}\dot{y} - \left(V_x + \frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{mV_x}\right)\dot{\psi} \qquad [0\text{-}13]$$

Similarly, from equation [0-2]. We have the following expression,

$$I_z\ddot{\psi} = l_fF_{yf} - l_rF_{yr}$$

Substituting [0-11] and [0-12] in the above equation,

$$I_z\ddot{\psi} = l_f\left(2C_{\alpha f}\delta_f - \frac{2C_{\alpha f}}{V_x}V_y - \frac{2C_{\alpha f}l_f}{V_x}\dot{\psi}\right) - l_r\left(-\frac{2C_{\alpha r}}{V_x}V_y + \frac{2C_{\alpha r}l_r}{V_x}\dot{\psi}\right)$$

$$I_z\ddot{\psi} = 2l_fC_{\alpha f}\delta_f - \frac{2l_fC_{\alpha f}}{V_x}V_y - \frac{2l_f^2C_{\alpha f}}{V_x}\dot{\psi} + \frac{2l_rC_{\alpha r}}{V_x}V_y - \frac{2l_r^2C_{\alpha r}}{V_x}\dot{\psi}$$

$$I_z\ddot{\psi} = 2l_fC_{\alpha f}\delta_f - \frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{V_x}\dot{y} - \frac{2l_f^2C_{\alpha f} + 2l_r^2C_{\alpha r}}{V_x}\dot{\psi}$$

$$\ddot{\psi} = \frac{2l_fC_{\alpha f}\delta_f}{I_z} - \frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{I_zV_x}\dot{y} - \frac{2l_f^2C_{\alpha f} + 2l_r^2C_{\alpha r}}{I_zV_x}\dot{\psi} \qquad \text{[0-14]}$$

The above derive relations for lateral vehicle dynamics model can be represented in state space form as:

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{I_zV_x} & 0 & -\frac{2l_f^2C_{\alpha f} + 2l_r^2C_{\alpha r}}{I_zV_x} \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}\delta_f}{m} \\ 0 \\ \frac{2l_fC_{\alpha f}\delta_f}{I_z} \end{bmatrix}\delta \qquad \text{[0-15]}$$

As mentioned earlier, the vehicle model represented above is based on 2D bicycle model. It is to be kept in mind that this is a simplified representation that considers vehicle as a single point mass, is valid under linear operation range of tires and does not consider any vertical forces

APPENDIX C: RESEARCH INFORMATION SHEET

Title of Study: Vehicle Driving Data Collection on a Test Track at Normal Driving Speeds

Principal Investigator (PI): Syed Ali
Electrical and Computer Engineering
248-506-6892

Purpose:

You are being asked to be in a research study to help understand driver's spacing preference from other vehicles on the road because you meet the study criteria of having a valid driver's license, 5 years of driving experience and are familiar with Fowlerville proving ground test track. This study is being conducted at Wayne State University and data will be collected at Fowlerville Proving Grounds, which is one of the few test tracks used on daily rental basis for drive tests by the automotive industry.

Study Procedures:

If you take part in the study, you will be asked to drive vehicle on Fowlerville test track. While you will be driving, vehicle position and motion data will be automatically recorded on a device (GPS/IMU integrated system) that is installed in the vehicle.

You will be asked to drive on the inside lane of the two-lane test track, while maintaining driving speeds of 45, 55 or 65 MPH to the best of your ability. You will be advised on what speed to drive on before the beginning of every test drive.

There will be a second stationary vehicle on the inside lane. You will be able to see this vehicle well before 500 meters. You should change your lane to the outside lane as approaching the other vehicle at a distance you deem safe and comfortable. Once you pass the vehicle, you should return to the inner lane at a point you consider safe and comfortable.

You have the option of terminating any or all of the driving task requested above if you deem it unsafe or is uncomfortable for you at any point.

You are expected to visit Fowlerville proving ground once for half a day period.

Benefits

As a participant in this research study, there will be no direct benefit for you; however, information from this study may benefit other people now or in the future.

Risks

There are no known risks at this time to participation in this study.

Costs

There will be no costs to you for participation in this research study.

Compensation

You will not be paid for taking part in this study.

Research Related Injuries

In the event that this research related activity results in an injury, treatment will be made available including first aid, emergency treatment, and follow-up care as needed. Care for such will be billed in the ordinary manner to you or your insurance company. No reimbursement, compensation, or free medical care is offered by Wayne State University. If you think that you have suffered a research related injury, contact the PI right away at 248-506-6892.

Confidentiality:

All information collected about you during the course of this study will be kept without any identifiers. You will be identified in the research records by a code name driver A, B etc or number. There will be no list that links your identity with this code.

Voluntary Participation /Withdrawal:

Taking part in this study is voluntary. You are free to not answer any questions or withdraw at any time. Your decision will not change any present or future relationships with Wayne State University or its affiliates.

Questions

If you have any questions about this study now or in the future, you may contact Syed Ali or one of his research team members at the following phone number 248-506-6892. If you have questions or concerns about your rights as a research participant, the Chair of the Institutional Review Board can be contacted at (313) 577-1628. If you are unable to contact the research staff, or if you want to talk to someone other than the research staff, you may also call the Wayne State Research Subject Advocate at (313) 577-1628 to discuss problems, obtain information, or offer input.

Participation

By conducting the test driving, you are agreeing to participate in this study.

REFERENCES

[1] "A Google self-driving car caused a crash for the first time," [Online]. Available: https://www.theverge.com/2016/2/29/11134344/google-self-driving-car-crash-report. [Accessed 2017].

[2] C. Hong, Autonomous intelligent vehicles: theory, algorithms, and implementation, Springer Science & Business Media, 2011.

[3] O. Khatib and B. Siciliano, Springer handbook of robotics, Springer, 2016.

[4] M.-H. Sigari, M. Fathy and M. Soryani, "A driver face monitoring system for fatigue and distraction detection," *International journal of vehicular technology,* 2013.

[5] R. Chhabra, S. Verma and C. R. Krishna, "A survey on driver behavior detection techniques for intelligent transportation systems," in *7th International IEEE Conference on Cloud Computing, Data Science & Engineering-Confluence*, 2017.

[6] G. Bifulco, L. Pariota, F. Simonelli and R. Di Pace, "Development and testing of a fully adaptive cruise control system," *Transportation Research Part C: Emerging Technologies,* vol. 23, pp. 156-170, 2013.

[7] E. de Gelder, I. Cara, J. Uittenbogaard, L. Kroon, S. van Iersel and J. Hogema, "Towards personalised automated driving: Prediction of preferred ACC behaviour based on manual driving," in *IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[8] R. C. Zhao, P. K. Wong, Z. C. Xie and J. Zhao, "Real-time weighted multi-objective model predictive controller for adaptive cruise control systems," *International Journal of Automotive Technology,* vol. 18, no. 2, pp. 279-292, 2017.

[9] A. Ziebinski, R. Cupek, H. Erdogan and S. Waechter, "A survey of ADAS technologies for the future perspective of sensor fusion," in *International Conference on Computational Collective Intelligence*, 2016.

[10] A. Bar, R. Lerner, D. Levi and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications,* vol. 25, no. 3, pp. 727-745, 2014.

[11] V. Magnier, D. Gruyer and J. Godelle, "Automotive LIDAR objects detection and classification algorithm using the belief theory," *IEEE Intelligent Vehicles Symposium (IV),* pp. 746-751, 2017.

[12] Z. Liu, S. Yu, X. Wang and N. Zheng, "Detecting Drivable Area for Self-driving Cars: An Unsupervised Approach," *arXiv preprint,* vol. 1705.00451, 2017.

[13] J. Schlosser, C. Chow and Z. Kira, "Fusing lidar and images for pedestrian detection using convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[14] W. Xiao, B. Vallet, K. Schindler and N. Paparo, "Street-side vehicle detection, classification and change detection using mobile laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing,* pp. 166-178, 2016.

[15] H. G. Seif and X. Hu, "Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry," *Engineering,* vol. 2, no. 2, pp. 159-162, 2016.

[16] C. Katrakazas, M. Quddus, W.-H. Chen and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies,* vol. 60, pp. 416-442, 2015.

[17] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research,* vol. 28, no. 8, pp. 933-945, 2009.

[18] H.-S. Tan and J. Huang, "Design of a high-performance automatic steering controller for bus revenue service based on how drivers steer," *IEEE Transactions on Robotics ,* vol. 30, no. 5, pp. 1137-1147, 2014.

[19] J. Guo, P. Hu, L. Li and R. Wang, "Design of automatic steering controller for trajectory tracking of unmanned vehicles using genetic algorithms," *IEEE Transactions on Vehicular Technology,* vol. 61, no. 7, pp. 2913-2924, 2012.

[20] D. Kim, J. Lee, B. Kim, K. Lee and K. Yi, "Integrated risk management based automated vehicle following system on inner-city streets," in *2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*.

[21] B. D. Argall, S. Chernova, M. Veloso and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems,* vol. 57, no. 5, pp. 469-483, 2009.

[22] R. S. Sutton, A. G. Barto and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems,* vol. 12, no. 2, pp. 19-22, 1992.

[23] D. J. Silver, A. Bagnell and A. Stentz, "Learning autonomous driving styles and maneuvers from expert demonstration," in *Experimental Robotics*, Springer International Publishing, 2013, pp. 371-386.

[24] T. Gu and J. M. Dolan, "Toward human-like motion planning in urban environments," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014.

[25] M. Kuderer, S. Gulati and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[26] "Test Track and Facilities," [Online]. Available: http://ftt-a.com/test_tracks_d.aspx. [Accessed 2017].

[27] "RT3000 v2 Family," OxTS, [Online]. Available: http://www.oxts.com/products/rt3000-family/. [Accessed 2017].

[28] "GNSS System Overview," NovAtel, [Online]. Available: https://www.novatel.com/an-introduction-to-gnss/chapter-1-gnss-overview/. [Accessed 2017].

[29] L. B. Richard , P. J. Teunissen and O. Montenbruck, "Introduction to GNSS," *Springer Handbook of Global Navigation Satellite Systems,* pp. 3-23, 2017.

[30] M. Farsi, K. Ratcliff and M. Barbosa, "An overview of controller area network," *Computing & Control Engineering Journal,* vol. 10.3, pp. 113-120, 1999.

[31] "Vector CANalyzer," Vector, [Online]. Available: https://vector.com/vi_canalyzer_en.html. [Accessed 2017].

[32] "Mat file format," Matlab, [Online]. Available: https://www.mathworks.com/help/pdf_doc/matlab/matfile_format.pdf. [Accessed 2017].

[33] R. J. Kiefer, C. A. Flannagan and C. J. Jerome, "Time-to-collision judgments under realistic driving conditions," *Human Factors,* vol. 48, no. 2, pp. 334-345, 2006.

[34] E. Coelingh, A. Eidehall and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking," in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010.

[35] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems ,* vol. 14, no. 4, pp. 1773-1795, 2013.

[36] L. Y. Wang, A. Syed, G. Yin, A. Pandya and H. Zhang, "Control of vehicle platoons for highway safety and efficient utility: Consensus with communications and vehicle dynamics," *Journal of systems science and complexity 27,* vol. 4, pp. 605-631, 2014.

[37] R. Rajamani, "Vehicle Dynamics," in *Vehicle Dynamics and Control*, Springer, 2012, pp. 87-112.

[38] "Prescan," [Online]. Available: https://www.tassinternational.com/prescan. [Accessed 2017].

[39] "IPG Automotive," [Online]. Available: https://ipg-automotive.com/. [Accessed 2017].

[40] "Elektrobit Assist," [Online]. Available: https://www.elektrobit.com/products/eb-assist/. [Accessed 2017].

[41] "dSPACE ASM," [Online]. Available: https://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models.cfm. [Accessed 2017].

[42] "Carsim," [Online]. Available: https://www.carsim.com/. [Accessed 2017].

[43] "Robot Operating System," [Online]. Available: http://wiki.ros.org/. [Accessed 2017].

[44] M. Kuhn and K. Johnson, in *Applied Predictive Modeling*, Springer, 2013.

[45] V. Cherkassky and F. M. Mulier, Learning from data: concepts, theory, and methods, John Wiley & Sons, 2007.

[46] T. Hastie, R. Tibshirani and J. Friedman, in *The Elements of Statistical Learning*, Springer, 2017, pp. 80, 389-415.

[47] "Feature Scaling," [Online]. Available: https://en.wikipedia.org/wiki/Feature_scaling. [Accessed 2017].

[48] S. Raschka, "About Feature Scaling and Normalization," [Online]. Available: http://sebastianraschka.com/Articles/2014_about_feature_scaling.html. [Accessed 2017].

[49] S. Yvan, I. Inza and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics ,* vol. 23, no. 19, pp. 2507-2517, 2007.

[50] M. F. Land and D. N. Lee, "Where we look when we steer," *Nature,* vol. 369.6483, pp. 742-744, 1994.

[51] M. Hall, F. Eribe, G. Holmes, B. Pfahringer, P. Reutemann and I. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter 11,* vol. 1, pp. 10-18, 2009.

[52] A. G. Asuero, A. Sayago and A. G. Gonzalez, "The correlation coefficient: An overview," *Critical reviews in analytical chemistry,* vol. 36, no. 1, pp. 41-59, 2006.

[53] K. O. Tarald , "Cautionary note about R 2," *The American Statistician,* vol. 39, no. 4, pp. 279-285, 1985.

[54] S. A. Glantz and B. K. Slinker, Primer of applied regression and analysis of variance, McGraw-Hill, 1990.

[55] T. Gu, J. Snider, J. M. Dolan and J.-W. Lee, "Focused trajectory planning for autonomous on-road driving," in *2013 IEEE Intelligent Vehicles Symposium (IV)*.

[56] C. Sprunk, B. Lau, P. Pfaffz and W. Burgard, "Online generation of kinodynamic trajectories for non-circular omnidirectional robots," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*.

[57] W. Xu, J. Wei, J. M. Dolan and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*.

[58] P. A, in *The Fourier Integral and Its Applications*, McGraw-Hill, New York, 1962, pp. 244-245,252-253.

[59] C. Knapp and C. Glifford , "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 24, no. 4, pp. 320-327, 1976.

[60] J. Ianniello, "Time delay estimation via cross-correlation in the presence of large estimation errors," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 30, no. 6, pp. 998-1003, 1982.

[61] R. Taylor, "Interpretation of the correlation coefficient: a basic review," *Journal of diagnostic medical sonography,* vol. 6, no. 1, pp. 35-39, 1990.

[62] K. Kulpa, Signal processing in noise waveform radar, Artech House, 2013.

[63] R. Peck, C. Olsen and J. L. Devore, Introduction to statistics and data analysis, Cengage Learning, 2015.

[64] R. Bishop, Intelligent vehicle technology and trends, 2005.

[65] O. Eytan, A. Eyal and G. Stein, "Camera focus for adas". U.S. Patent 15/001,880, 20 January 2016.

[66] T. Howley, M. Madden, M.-L. O'Connell and A. Ryder, "The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data," in *Proceedings of AI-2005*, 2006.

[67] R. Azuma, "A Survey of Augmented Reality," *Presence: Teleoperators and Virtual Environments,* vol. 6, no. 4, pp. 355-385, 1997.

[68] P. I. Corke, "A Robotics Toolbox for MATLAB," *IEEE Robotics And Automation Magazine,* vol. 3, no. 1, pp. 24-32, March 1996.

[69] S. Ali, L. Reisner, B. King, A. Cao, G. Auner, M. Klein and A. Pandya, "Eye Gaze Tracking for Endoscopic Camera Positioning: An Application of a Hardware/Software Interface Developed to Automate Aesop," *Medicine Meets Virtual Reality 16 (MMVR16),* pp. 4-7, 29 January 2008.

[70] M. G. Wing, A. Eklund and L. D. Kellogg, "ConsumerGrade Grade Global Positioning System (GPS) Accuracy and Reliability," *J. Forestry,* p. 169–73, 2005.

[71] H. Landau, U. Vollath and X. Chen, "Virtual reference station systems," *Positioning,* p. 0, 2009.

[72] J. Wei, J. Snider, J. Kim, J. Dolan, R. Rajkumar and B. Litkouhi, "Towards a viable autonomous driving research platform," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[73] U. Ozguner and K. A. Unyelioglu, "An analytical study of vehicle steering control," in *Proceedings of the 4th IEEE Conference on Control Applications*, 1995.

[74] K. Jalali, S. Lambert and J. McPhee, "Development of a path-following and a speed control driver model for an electric vehicle," *SAE Technical Paper,* 2012.

[75] "Development of an advanced fuzzy active steering controller and a novel method to tune the fuzzy controller," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems,* vol. 6, no. 2013-01-0688, pp. 241-254, 2013.

[76] G. James, D. Witten and R. Tibshirani , An Introduction to Statistical Learning, Springer, 2013.

[77] J.-W. Lee and B. Litkouhi, "A unified framework of the automated lane centering/changing control for motion smoothness adaptation.," in *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*.

[78] H. Cheng, in *Autonomous intelligent vehicles: theory, algorithms, and implementation*, Springer Science & Business Media, 2011.

[79] B. Siciliano and O. Khatib, Springer handbook of robotics, Springer, 2016.

ABSTRACT

TOWARDS HUMAN-LIKE AUTOMATED DRIVING: LEARNING SPACING PROFILES FROM HUMAN DRIVING DATA

by

SYED ALI

December 2017

Advisor: Dr. Abhilash Pandya

Major: Computer Engineering

Degree: Doctor of Philosophy

For automated driving vehicles to be accepted by their users and safely integrate with traffic involving human drivers, they need to act and behave like human drivers. This not only involves understanding how the human driver or occupant in the automated vehicle expects their vehicle to operate, but also involves how other road users perceive the automated vehicle's intentions. This research aimed at learning how drivers space themselves while driving around other vehicles. It is shown that an optimized lane change maneuver does create a solution that is much different than what a human would do. There is a need to learn complex driving preferences from studying human drivers.

This research fills the gap in terms of learning human driving styles by providing an example of learned behavior (vehicle spacing) and the needed framework for encapsulating the learned data. A complete framework from problem formulation to data gathering and learning from human driving data was formulated as part of this research. On-road vehicle data were gathered while a human driver drove a vehicle. The driver was asked to make lane changes for stationary vehicles in his path with various road curvature conditions and speeds. The gathered

data, as well as Learning from Demonstration techniques, were used in formulating the spacing profile as a lane change maneuver. A concise feature set from captured data was identified to strongly represent a driver's spacing profile and a model was developed. The learned model represented the driver's spacing profile from stationary vehicles within acceptable statistical tolerance. This work provides a methodology for many other scenarios from which human-like driving style and related parameters can be learned and applied to automated vehicles.

AUTOBIOGRAPHICAL STATEMENT

Mr. Ali has a Bachelor in Science in Electrical in Computer Engineering and a Master in Science in Computer Engineering, both from Wayne State University. He is currently working as a Feature Owner for Lane Departure Warning and Control at General Motors, where he is responsible for designing driver assist features such as Lane Departure Warning and Lane Keep Assist, among others. Prior to joining General Motors, Mr. Ali lead several technical projects at Tier-1 suppliers, including leading a research and development team in automated driving at DENSO. He has 3 conference and one journal publications. He also has two issued and one pending U.S. patent.