

Driver Adaptive Warning Systems

Thesis Proposal

Parag H. Batavia

CMU-RI-TR-98-07

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

March, 1998

(C) 1998 Parag Batavia

This work was supported in part by the National Science Foundation (NSF) under a graduate research fellowship, by the National Highway Traffic Safety Administration (NHTSA) under contract DTNH22-93-C-07023, by the Office of Naval Research (ONR) under contract N00014-95-1-0591, and by the Federal Highway Administration (FHWA) under cooperative agreement DTFH61-94-X-00001 as part of the National Automated Highway System Consortium

Abstract

Each year, many preventable highway automobile accidents involving single vehicles are caused by inattention and distraction. These accidents are classified as single vehicle road departures. Lane departure and curve negotiation warning systems are an emerging technology to help prevent these types of accidents. **I plan to build a road departure warning system that learns individual driver behavior, and uses this knowledge to reduce false alarms and increase warning time.** Current warning systems are physics based -- they look at vehicle trajectory, but mainly ignore driver ability and characteristics. I propose to develop an adaptive lane departure and curve negotiation warning system. This system should learn individual traits of the driver -- both stationary and changing, and use this information to improve warning time and reduce false alarms. A number of research issues are involved in this work, as it has to improve upon the state of the art, yet not become so complicated to use that the average driver would feel uncomfortable using it. In this proposal, I will discuss these issues and describe preliminary results in using a connectionist approach to predict the driver's steering response given vehicle state information. This approach can successfully detect lane changes, which I treat as surrogate road departures.

Table of Contents

Introduction 5

Motivation 6
Definitions 6

Previous Work 9

ROR Collision Avoidance Using IVHS Countermeasures 9
Control Theoretic Models 10
HMM Based Intent Recognition 10
Human Control Strategy Modeling 11
Daisy 12
Crewman's Associate for Path Control (CAPC) 12
RALPH 13
Curve Negotiation Warning Work 14
Other Work 14
Discussion 14

The Rapidly Adaptive Lateral Position Handler (RALPH) and Navlab 8 15

Vehicle Description 15
RALPH 17

Pre-Proposal Work 17

Datasets 17
 Carnegie Mellon Research Institute Data 17
 Initial CMU Data Study 19
 Description 19
 Experimental Effects 19
Alarm Analysis 20
Driver Differences 23
Neural Network Based Warning System 25
 Neural Net Architecture 25
 Datasets 26
 Results 26
 Test Set Results 26
 Lane Change Results 28
 Normalized Inputs 32
 Discussion 33
Curve Negotiation Differences 34

Research Issues 34

Model Input 36
Learning Method 36
Learning History 37
Surrounding Vehicles 37
Curve Warning Adaptation 37
Predictability vs. Efficiency 37
Sensitivity to Perceptual Errors 38
System Evaluation 38

System Evaluation 38

Quantitative Analysis 39

Qualitative Analysis 39

Proposed Work 39

Data Collection 39

Navtech Map Evaluation 40

Curve Behavior Analysis and Model Development 40

Lateral Control Model Selection 40

Evaluation On CMRI Truck Driver Data 41

Accounting for Surrounding Vehicles 42

Final User Study 42

Expected Contributions 42

Schedule 43

References 43

Thesis Statement

I plan to build a road departure warning system that learns individual driver behavior, and uses this knowledge to reduce false alarms and increase warning time.

1. Introduction

This thesis has the goal of developing a driver adaptive warning system which is capable of both lane departure and curve negotiation warning, for use in preventing single vehicle road departure accidents, also known as Run-Off-Road (ROR) accidents. This problem has significance because of the number of traffic fatalities which result from ROR crashes. Preventing a portion of these accidents would have a real impact in the number of lives saved.

Current warning systems, such as Pomerleau's RALPH warning system [35] use physics based models, such as looking at the position of the vehicle in the lane, or looking at the direction the vehicle is pointing relative to the lane and calculating a time to lane crossing metric. While these systems work well, they do not capture individual driver traits which could be exploited to improve the efficiency of the system.

These traits vary between different drivers. They also exist within the same driver, over time. Examples of differences include mean lane position, lane position variance, steering wheel reversal rate, steering wheel reversal magnitude, etc. While most of the behaviors expressed through these differences are safe, drivers also engage in unsafe behaviors, such as straddling two lanes or excessive weaving.

This leads to issues which make this problem interesting from a machine learning standpoint. I am trying to model a plant (in this case, a driver) whose reactions to a given situation change over time. The time period over which the change occurs is variable. The domain is noisy, and I do not have any positive examples of the situation I am trying to avoid (ROR crashes). I also only want to learn behaviors which are safe expressions of individuality. If the driver is slowly falling asleep, I do not want to learn, and therefore allow, the sluggish driving which leads up to the eventual ROR. Furthermore, the algorithm has to operate in real time.

Gathering appropriate data to test the system presents a challenge. Ultimately, the system has to set a threshold, beyond which the driver is in danger of an ROR. However, setting this threshold without any examples of RORs is difficult. To gather positive examples of ROR situations would require repeatedly driving a test vehicle off road by distracting or incapacitating the driver. Obviously, this is not a possibility. Therefore, I have to propose operational definitions of ROR and true and false alarms in terms of available data.

The system sensitivity needs to be tunable, and the system operation needs to be easily understood by the average driver. If the warning system uses a very complicated set of criteria by which it triggers alarms, the driver may not be able to develop a feel for why the system is doing what it is doing, and would not want to use it. While the internal complexity of the model is not necessarily an issue, how it manifests itself to the driver is. An (exaggerated) example of this would be if the system took the day of week into account when deciding whether or not to trigger an alarm, perhaps because more ROR crashes occur on Mondays. In this case, a situation which would not sound an alarm six days of the week would sound an alarm on Mondays. This would confuse the user, as he may not be aware that the system is more sensitive on Mondays. User testing, therefore, is a critical method of evaluation.

Addressing these issues will require contributions to 1) driver modeling, through the development and analysis of a low level driver model, and 2) machine learning, through the development of a learning method which properly learns safe variations in time-varying plant behavior and learns anomalous behavior without true positive examples.

1.1. Motivation

In 1996, there were over 37,000 automobile accidents involving fatalities, in which 42,000 people were killed. While there are many different causes of accidents, those that involve a single vehicle are frequently caused by inattention or incapacitation, leading to roadway departure. Of the 37,000 fatal accidents in 1996, over 21,000 were single vehicle accidents. These 21,000 accidents resulted in 22,500 fatalities, or 56% of the total [44]. The combined cost of *all* accidents is estimated to be over \$150 billion per year.

Intuitively, most people would agree that different people drive differently. We have all seen people weaving wildly on the road. For some, this may indicate an incapacitation, such as fatigue or drunkenness. For others, this is simply how they drive. Some people hug one side of the road, as if afraid of getting brushed by traffic on the other side. Others, it seems, sometimes straddle two lanes. There are different types of behaviors like this that drivers engage in. Some of these behaviors, such as the tendency to hug one side of the road and corner-cutting, are safe expressions of personal preference. Other behaviors, such as straddling two lanes or weaving wildly, can be unsafe. The difficulty in developing an adaptive lane departure warning system lies partly in adapting to safe changes in driver behavior, while not adapting to unsafe changes.

As we will see in Section 4, this type of behavior varies not only between drivers, but also within the same driver over time. The driving style of a person during the first hour of an eight hour trip can look different than during the last hour. Generally, these differences can have an impact on the alarm rate of a warning system, but do not affect overall driver safety. Therefore, it is useful to adapt to these changes. Because of this, a one time adaptation is not sufficient. Rather, the system should slowly adapt itself over time, being sensitive enough to learn large scale changes in lane keeping behavior, but not so responsive as to never issue a warning, because it has adapted to an unsafe situation.

Curve negotiation is another area where many accidents occur. Of the 21,000 single vehicle fatal accidents mentioned above, 4,800 occurred due to errors in curve negotiation. This is particularly a problem for trucks and other large vehicles in mountainous areas, where curves can be sharp, and shoulders are narrow. There is not a lot of room for mistakes when driving in this type of environment. Posted curve speed signs are sometimes missed or ignored. Driver differences also exist when negotiating curves. Certain drivers are aggressive, traverse curves at high speed, and brake fairly late. Other drivers are more cautious, and slow down well in advance of a curve. These differences are worth accounting for, to prevent false alarms. The success of this technology will require accurate GPS maps, which allow the vehicle to locate itself relative to oncoming curves, so that warnings can be issued with enough time to be useful.

1.2. Definitions

It would be difficult to accomplish the goal of my thesis, which is to minimize false run off road (ROR) alarms and improve response to true ROR situations, without defining ROR crashes, ROR situations and false and true alarms. The motivation for this goal is that I believe it will lead to fewer ROR situations, and consequently, fewer ROR crashes. I define an ROR situation as:

ROR Situation (ROR): Any vehicle state which leads to an ROR crash. However, not all ROR situations lead to ROR crash, as the driver may initiate corrective action.

ROR Crash: An event in which at least 1 tire crosses a lane boundary, resulting in a full lane departure and/or crash.

In the datasets which I currently possess, there were no ROR crashes. In fact, encountering a true ROR crash is very rare, and not something I can wait for. One possibility is to treat lane changes as true alarms, as vehicle state during lane changes can be similar to vehicle state during ROR situations due to unintended steering input or inattention. This is justified in Section 2.1.

There are two levels to the definitions issue. The first is an abstract definition of what vehicle states lead to roadway departures for a given driver (true alarms), and what states are normal, yet trigger current alarm systems (false alarms). The second issue is to develop operational definitions which can be used to evaluate a warning system on data in currently existing datasets.

The first issue, regarding what leads to an ROR crash, is very complicated. It depends on vehicle state, driver state, road state, and surrounding environment. Unfortunately, there is no useful boundary in vehicle state space beyond which an ROR crash will definitely occur. It would be nice to be able to say that if $vehicle_yaw > x$, $lateral_position < y$, and $road\ curvature > z$, then an ROR will occur. While it may be possible, using physics, to say that beyond a point no recovery is possible, triggering a warning would be useless. There is also no single driver behavior which should *always* trigger an alarm, as even rapid swerving may have a legitimate cause, such as obstacle avoidance. Without taking the surrounding environment (such as the presence of other vehicles), road conditions (such as friction and shoulder width), and human factors into account, it is impossible to determine whether or not an alarm should trigger. While I am interested in driver behavior and how it is affected by surrounding vehicles, there are issues which I am not going to deeply investigate. These issues include roadway conditions, human factors, and road configuration (i.e., the presence or absence of shoulders). As I will show in Section 4.4, it is possible to improve upon the state of the art without fully addressing these issues. Therefore, conclusive abstract definitions of true and false alarms is outside the scope of my thesis.

However, I do present a working set of definitions, which I feel are reasonable. If the alarm triggers in a situation which the driver deems normal, he will feel it is a false alarm. Driver perception is therefore very important in classifying an alarm. The definitions of a system designer are irrelevant if the end user does not agree with them.

Hadden et al. [19] have done a simulation study of lane departure countermeasure effectiveness. The particulars of the study are discussed in Section 2.1. However, during analysis of the results, they put forth 6 possible outcomes of a countermeasure intervention. I use this framework to present the following definitions (Note, when I refer to the driver, I mean a normal driver who is not incapacitated and is paying attention to the road):

Safe True Alarm: An alarm triggered in time to prevent a situation where the driver, in hindsight, recognizes that his actions could have resulted in an ROR crash.

Late True Alarm: An alarm triggered in a situation where the driver, in hindsight, recognizes that his actions could have resulted in an ROR crash. This alarm, however, comes too late to fully prevent an ROR situation, and an ROR crash may have occurred.

Safe False Alarm: An alarm which triggers in a situation where the driver, in hindsight, does not believe that his actions could have resulted in an ROR crash. Besides the alarm itself, there is no other consequence.

Unsafe False Alarm: An alarm which triggers in a situation where the driver, in hindsight, does not believe that his actions could have resulted in an ROR crash. In this case, the alarm causes a reaction in the driver which could lead to an unsafe situation.

False Negative: A situation in which the driver, in hindsight, recognizes that his actions could have resulted in an ROR crash, yet no alarm triggered.

True Negative: A situation in which the driver, in hindsight, does not believe his actions would result in an ROR crash, and no alarm triggered. This is by far the most common outcome.

Burgett [6] presents definitions of system efficiency, which is discussed in Section 6.2 of this proposal. He also mentions another category of false alarms, although I have altered his definition to make it more applicable to this domain:

Nuisance Alarm: A safe false alarm caused by either poor system design or perceptual error.

For the 2nd issue, regarding design and preliminary evaluation, it may be fair to say that lane changes are similar to true alarms. The validity of this requires an analysis of vehicle state directly before true ROR situations, and a comparison against vehicle state before and during voluntary lane changes. The IVHS Countermeasures work, described in Section 2.1 implies that vehicle state during ROR crashes due to inattention or incapacitation is grossly similar to a lane change. The goal of the system in this case would be to trigger warnings when approaching the alarm state, while not triggering during other, similar states.

Another question which arises, given my objective, is how much of an increase in warning time and decrease in false alarm rate should I attempt to achieve? These two issues are very closely related, and improving one generally has a negative effect on the other. System efficiency can be described as being positively correlated with warning time, and negatively correlated with the number of false alarms. Most current systems use a time to lane crossing (TLC) method, in which the lateral position and lateral velocity of the vehicle is used to determine the time for a tire to cross a lane boundary. Warning systems which use TLC set a threshold. When the time to cross a lane boundary drops below this threshold, an alarm is triggered. Since TLC based systems can have user defined thresholds, a warning can be given as early as desired. However, the higher the TLC threshold, the more false alarms are generated. Setting the TLC threshold such that an alarm is only generated when crossing a lane boundary has zero warning time, yet has a low false alarm rate, as passenger car drivers rarely deviate from their lane on straight roads. Because of this relationship, the increase in warning time has to be compared to the reduction in false alarm rate. A large decrease in false alarm rate, and greater accuracy in modeling the driver would allow for a more warning for a true alarm.

Current research shows that drivers are possibly willing to tolerate one false alarm per hour. Anything more than that annoys the driver so much that he turns the system off. However, the true number of acceptable false alarms awaits the completion of a thesis on human factors in driving. Until that day, my approach will be tunable to increase or decrease the false alarm rate (with a corresponding change in warning time). While I do not believe I can escape the warning time/false alarm trade off, I believe I can build a system which has greater efficiency than TLC by adapting to the driver.

Given that I am interested in lane keeping performance, which is lateral behavior, and curve speed warning, which is longitudinal, I should also be interested in longitudinal behavior along straight roads as well. However, I am explicitly not modeling speed keeping, headway maintenance, or car following behavior. The reason for this is that a failure in lateral lane keeping behavior or longitudinal curve negotiation behavior are both direct causes of ROR crashes. A causality link between the other aspects of driver modeling mentioned above and ROR crashes has not been strongly demonstrated. As the goal of my thesis is to reduce ROR crashes, ignoring longitudinal behavior (except for curve negotiation) does not directly impact my work.

2. Previous Work

While there is a large amount of literature in driver modeling, both at the path and tactical level, there has not been a lot of work done in driver adaptive warning systems. I begin with a brief review of some work done to categorize and characterize single vehicle accidents, then general driver modeling work, followed by detailed descriptions of two efforts in control strategy modeling (which has applications to warning systems), and end with examples of actual driver warning systems. For a more extensive literature survey in the form of an annotated bibliography, see [2].

2.1. ROR Collision Avoidance Using IVHS Countermeasures

The goal of this work, done at CMU and CALSPAN, was to develop a taxonomy of roadway departures, and design functional measures which could ameliorate the effects of these crashes. The taxonomy broadly classifies crashes into different causal factors, such as inattention, relinquished steering control, evasive maneuver, lost directional control, vehicle failure, and vehicle speed. 102 accidents were selected from a database of approximately 200, and categorized into these causes. The accidents were further broken down into type of deviation. The two types of deviation were long and short, where a long deviation included crossing a full lane before lane departure, and a short deviation meant a roadway departure on the side of the road closest to the vehicle. These accidents were also classified by pre-existing event/conditions (such as road geometry, road state, presence of obstacles, etc.) and on road and off road action by the driver. In the majority of the analysis, the accidents due to evasive maneuvers and vehicle failure were not used, as it was decided that resolving those causes was outside the scope of the program.

The results from this work, while encouraging, have to be properly weighed given the methods used. The trajectory for the accidents was computed by analysis of the crash scene and intersected with a nominal trajectory to follow the road. This included looking at the final position of the vehicle, along with any skid marks that might be present. For accidents caused by inattention or loss of steering control, which tend to have larger times from deviation (from nominal trajectory) beginning to roadway departure, a circular arc was fitted to points determined by skid marks. Then, vehicle velocity information was either gotten from the driver, witnesses, or through various assumptions about the crash scene. This was used to generate times from deviation beginning to roadway departure. This analysis determined that on average, there is about 2.12s during this period. After taking driver and vehicle response time into account, about 1 second is left to determine that a situation is abnormal, and sound an alarm. The presence of a typical shoulder adds about 0.5 seconds.

Taken into context, this work starts to show that departures due to inattention or incapacitation tend to be gentler than those caused by active maneuvers. This implies that grossly, these departures are similar to lane changes. However, the actual numbers derived have to be taken with a grain of salt, as the methodology was very inexact, due to the limited information available characterizing the crash.

In a later phase of this work [19] the authors performed a simulation study to look at the effectiveness of TLC in preventing ROR crashes. They split up an ROR crashes into two possible cases: the first is a 1-tire ROR, which means at least one front tire has crossed a lane boundary. The second cases is a 2-tire ROR, which is when both front tires have crossed a lane boundary. Using a dynamic vehicle model and driver steering model, they simulated inattention (by deactivating the driver model) over sets of curves. Using a Monte Carlo simulation to vary parameters such as velocity, incapacitation time, TLC

threshold, and driver reaction time, they performed over 500 runs both with and without a TLC based warning system in place. The results show that increasing TLC threshold prevents RORs, at the expense of false alarms. These results show the inherent trade-off which must occur when balancing warning time against false alarms.

2.2. Control Theoretic Models

Investigation of control theoretic approaches to driver modeling began as early as the 1950s, when Pipes [33] modeled the driver as a gain and a time delay, and modeled the vehicle lateral position as an integration of steering wheel angle. Over the next few decades, that work was expanded upon, as the model of the driver became more complicated and attempted to take into account evidence provided by studies of driver behavior.

Wierwille [46], who has been active in this field for many years, presented an early model which took into account past lateral displacement, future roadway curvature, and driver vantage point. This work showed that information on the upcoming road curvature helps to eliminate the effects of perceptual and reaction lag.

Crossman and Szostak [12] proposed a three level model which combined open loop control of vehicle curvature given upcoming road information, with closed loops around lateral position and lateral velocity. McRuer et al. [26] added a “precognitive” open loop control module, which was used to establish the driver on an appropriate trajectory for lane changes and obstacle avoidance maneuvers.

Baxter and Harrison [3] take a previous linear control model, and add a non-linear hysteresis element, in an attempt to model the oscillations of drivers driving on straight roads. Rather than raw vehicle state, they use aim-point error, which is the angle between the vehicle heading and the lane centerline at a certain lookahead distance. Their results indicate a 10% improvement in modeling accuracy over the standard linear model they tested against.

The main assumption in control theoretic approaches to driver modeling is that humans, and the vehicles that they control, can be adequately simulated using 2nd order systems. Stochastic and non-linear effects, such as crosswind response, cannot be modeled well using these approaches. Furthermore, it becomes very difficult to take into account environmental effects such as the presence of other vehicles. One area where these approaches have worked well is in car following, as shown by Chandler [10], Bekey [4], Ioannou [21], and Naab [27]

2.3. HMM Based Intent Recognition

Liu and Pentland [24] at the Nissan-Cambridge Research Labs in Boston have developed a model of driver intention using Hidden Markov Models (HMM)[37]. Their motivation for recognizing driver intent is to aid in selecting a proper dynamical driver model, given the current situation. For instance, different models would apply during an overtaking scenario, such as lane changing and acceleration phases.

Their data was collected using a fixed based Nissan 240SX simulator. The simulator is capable of logging steering position, steering velocity, and vehicle velocity and acceleration. The cab of the simulator is the front half of a real 240SX, and the driver’s view is projected on large screens in front of the windshield, with a 60x40 degree field of view. Eight male subjects were asked to drive the simulator around a city, while they were randomly given instructions (presented on the screen), such as “change

to the left lane,” “overtake this vehicle,” etc. This data was used to train separate HMMs for each maneuver. During real-time operation, the observations of the vehicle state are run through each HMM, and the one with the highest likelihood of generating the presented observation determines the current action.

The results show that correct recognition rate is around 85%, within 1-1.5 seconds of beginning the maneuver. The rates vary for the different maneuvers, but are generally in the mid to high 80% range. However, these results are for detection after the maneuver has begun. It is still unclear (as lane change maneuvers tend to be 2-4 seconds long) whether or not predicting driver intent using this method is feasible. Certainly, for longer maneuvers such as passing, a 1-1.5 second classification time is a good result.

While this is not a driver warning system per se, the ideas explored in this work can have an impact in the design of a warning system. Particularly, their idea of using an HMM to generate the probability of being involved in a certain maneuver could be useful for suppressing a warning system during the maneuver.

2.4. Human Control Strategy Modeling

Michael Nechyba has been doing work in Human Control Strategy Modeling. His approach is to build a hybrid model consisting of a neural architecture [29] for modeling of continuous systems, along with an HMM for discontinuous systems [28]. Models are validated using an HMM based similarity metric [30] that looks at the cross-probability of a sequence of observations generated by both training data, and the model, fed back upon itself. The domain which he has concentrated on is driving.

Nechyba used a driving simulator to collect data on 6 people, where the state information recorded is lateral, longitudinal, and angular velocity. The control outputs are steering and brake/throttle. While the steering control is nearly linear and modeled with a neural controller, the discreteness of the brake and throttle commands were better modeled using the HMM approach mentioned above.

The results, which demonstrate that his models do a better job at modeling drivers than an optimal bayes classifier, are impressive. However, the model is quite complex; perhaps more so than needed for a driver warning system. Some of the complexity was induced by the limitations of the simulator, which is unrealistic. His work also concentrates on longer term control strategies. Shorter-term variations due to local changes in driver state and driver environment (which can be on the order of minutes) are not accounted for, and this is a limitation. Furthermore, his use of a cascade architecture to learn steering output prevents it from being used in an on-line system. This is because cascade architectures are not amenable to on-line learning, as once a hidden unit is added, the input weights to that unit are frozen. Therefore, it can't forget what it has previously learned. In a domain where the proper response to a situation changes over time, this limitation prevents this approach from being deployable.

2.5. Daisy

The Driver Assisting System (DAISY) [15],[31] is a comprehensive driver adaptive warning system, geared to give warnings based on a time reserve, which is a combination of time to lane crossing (TLC) [17], along with time to collision (with other vehicles or obstacles). The system consists of a situation analysis monitor, which uses petri-nets to classify the current situation given environmental inputs, such as the pose of surrounding vehicles. Car following and lane keeping are provided as examples of tactical situations. An “average” driver model is ascribed to the surrounding vehicles, and used to determine limitations in action selection.

The actual driver model is multi-level, and consists of a rule based model for intent recognition, along with a neural architecture for skill level control. The intent recognition module attempts to predict what the driver will do given the current tactical situation. This information is then used to select a skill model, which predicts the actual control inputs the driver is likely to produce to realize his predicted intention.

The skill model is implemented using a FuzzyART [9] (Adaptive Resonance Theory) network in an ARTMAP [8] architecture. FuzzyART is a modification of ART1 [7] for dealing with analog patterns. ARTMAP is an associative memory, which trains two ART1 nets, one to cluster input, and one to cluster output. The two ART nets are connected via an associative network. Essentially, a two level FuzzyART/ARTMAP network is used to cluster feature vectors describing the pose of surrounding vehicles. The clustering at the second level is done at a finer resolution than the first, and is then associated with a time series of expected control outputs. There is a set of these 2-level networks, each one corresponding to a different tactical situation.

There are 17 different FuzzyART networks for longitudinal control (describing situations such as car following or car approaching), and 24 for lateral control (including lane keeping and overtaking). Currently, the training is all done off-line, using recorded data, and a simulated model of the test vehicle. This is because a Genetic Algorithm [18] is used to optimize network specific parameters, such as the relative weights of each feature in the tactical feature vector, which changes given different tactical situations. The authors believe that once the optimal feature weights are found for each different network, they can be used during on-line training of different drivers.

The main limitations of this work as follows: First, the large number of models requires extensive training and large amounts of data. Second, the system hasn't been deployed on a real vehicle, so there is no data as to its effectiveness -- even the simulation results are sketchy and hard to interpret. Third, their dependence on a situational analysis model requires them to manually define all the situations a driver may encounter. Finally, the system makes no allowance for changes in driver behavior over time.

2.6. Crewman's Associate for Path Control (CAPC)

The CAPC system [13], developed at the University of Michigan Transportation Research Institute, is a prototype vehicle which implements a TLC based lane departure warning. While the goal of the project was to build a driver-adaptive system, the current implementation uses hardcoded TLC thresholds. CAPC uses a sophisticated model of road geometry [23] along with vehicle performance [25] to push forward the vehicle in time, until a point at which a lane departure occurs. Heuristics are used to determine when to sound an alarm, and the TLC thresholds are empirically determined in simulation. There are two TLC thresholds, one for warning in the form of an audible buzzer, and another for intervention via differential braking. The main contribution of this system is a refinement in how TLC is calculated. While this improves upon the performance of TLC based systems, it still contains the inherent limitations of TLC, such as lack of driver adaptation and inability to deal with the effects of surrounding vehicles. A small user study was also performed. The qualitative results of this evaluation showed that it behaved as users expected it to. There is not much detail on how many people participated, or even if they were researchers or picked from a random population.

A preliminary result in driver adaptation is also presented [32]. The ARX [11] algorithm is used to develop a transfer function from vehicle state (lateral deviation and heading angle) to steering wheel position. This transfer function is repeatedly computed, using slices of the data, which is from a simulator. A third order function is used, although two of the poles and zeros were related to quantization error in the state sampling. However, the dominant pole location changes as a function of time, indicat-

ing a larger effective time constant of control. This correlates with plots of how the standard deviation of lateral position changes over time. While this may be an indication of fatigue, numerically, the differences are very small, and they authors haven't shown if it is repeatable. The only difference that they seem to be learning is in lane position variance. The use of a simulator is also a large disadvantage, as driving simulators (especially unsophisticated ones) are very unrealistic and it is not clear that control strategies for simulators and real vehicles are similar. However, this result is an indication that driver behavior does change over time. Further evidence of this is presented in Section 4.3.

2.7. RALPH

The RALPH lane tracking system [35], described in Section 3, includes a lane departure warning system (which from now on will be referred to as RALPH-WS, for RALPH Warning System) which depends on TLC. The TLC is calculated by using estimates of lateral velocity and current lane position. The upcoming geometry of the road is not accounted for. A TLC threshold is selected, and if the current lateral velocity and lane position indicate that the driver will exceed the lane boundary in a time less than the threshold, an alarm is sounded. There are a number of situations in which perceptual limitations prevent proper operation of the system. Approximately 16 heuristics are applied to determine whether or not the warning system output is reliable. These heuristics include:

- The presence of a nearby obstacle or obstruction.
- Low confidence in the lane tracking system.
- If the vehicle is tailgating, or very close to the vehicle in front of it.
- If there have been too many warnings in a given time period.

Furthermore, the warning system is also disabled in situations where it looks as if the lane departure is intentional, or corrective measures are being taken, such as:

- A high steering wheel rate, indicating a correction.
- A turn signal being active.
- Brake being applied.

While the system performs well, the false alarm rate is dependent on the style of the driver. Someone who normally drives along the center of the road and does not deviate much has a low alarm rate. Other users can have higher rates, as they may normally drive closer to the side of the road. The false alarm rate is tied to the TLC threshold used. A TLC threshold of 0.0 seconds will cause an alarm to sound only when a tire is already over the lane boundary. In area with wide shoulders, this is appropriate. However, on narrow stretches of road, a higher threshold is needed.

2.8. Curve Negotiation Warning Work

There has not been much work done in vehicle based curve warning systems, and none on adaptive curve warning systems. Tamura [42] describes heuristics to determine whether or not a given speed is appropriate for an upcoming curve. They also use a custom GPS map to localize themselves and provide warning of upcoming curves. However, their system is not adaptive as it does not take advantage of differences in braking onset or speed through curves. There has been some work on infrastructure based warning systems. Bergan et al. [5] determine weight, type, speed and deceleration to determine if a truck is in danger of tipping over. They use infrastructure mounted sensors to determine these variables, and use a sign to alert the driver. Fukuda [16] uses a road mounted microwave doppler radar to measure vehicle speed along curves. The purpose of this is to reduce the number of oncoming lane

encroachments by vehicles, which is apparently a major problem in Japan. Fukuda demonstrates that using the radar to warn the driver of excessive speed (via a message board) reduced both average vehicle speed and number of encroachments. While infrastructure based methods have an advantage in cost and ease of deployment, their reliance on message boards leaves open the possibility that a driver may miss a warning. I believe that on board systems, particularly adaptive ones which alert the driver that he is doing something not normal for him, will ultimately prove more effective at reducing accidents during curve negotiation.

2.9. Other Work

Takahashi and Kuroda [41] used ID3 [36], a decision tree induction algorithm, to design a controller which anticipates the intention of a driver going downhill to downshift for engine braking. The results showed that their ID3 derived rules, which looked at vehicle speed and acceleration, were able to trigger downshifts when the driver expected them.

The University of Michigan Transportation Research Institute has conducted a large study of how driver characteristics influence headway maintenance [14], [39]. They loaned vehicles equipped with a prototype adaptive cruise control (ACC) system to over 100 drivers who were going on long trips. The ACC system allowed the user to set a desired headway, and would decelerate the vehicle if the constraint was violated. Independent variables included driver age, sex, and cruise control usage, along with road type and environmental factors. While the system is not adaptive, it allowed the user a choice of headway settings. The researchers analyzed headway maintenance with ACC active and inactive. They showed that younger drivers tend to maintain a closer headway, and selected the lowest setting when in ACC mode. This study is one of the only large scale user studies ever conducted, and the volume of data collected will be very useful for longitudinal driver modeling.

Zhao [47] is using multiple Kalman filters to track vehicles using vision. She uses separate filters tuned to in-lane driving and lane changes and is therefore able to track low level in-lane motion as well as tactical level motion. This work has promise for use in a very unobtrusive data collection system.

2.10. Discussion

The previous work begins to show that there is an interest in developing driver adaptive warning systems. This is evident in the ARX based adaptation by the CAPC group, Nechyba's results, and the DAISY system. However, the current state of the art in driver warning systems does not include a system which both adapts to the driver, and demonstrates a quantitative improvement over non-adaptive systems. Furthermore, none of the work mentioned above exists in a form that allows it to be used in an actual vehicle on a daily basis by an untrained user.

While some of the previous work does recognize the need for driver adaptation, it is regarded as a one time procedure. There is the implicit assumption that a canonical model for an individual can be learned, and that further modification of the model is never necessary. I believe that this is not the case, and will present some experimental evidence in Section 4.3.



Figure 1: Navlab 8 sensor position.

Something else that is ignored by all work except DAISY, is the effect of other vehicles on the roadway. For instance, it is normal for drivers in the left hand lane to ride the left boundary when a truck or bus is passing them on the right. While DAISY does make an attempt at handling this issue, it is done by listing all the possible high level situations (being tailgated, being passed on the right, etc.), and training separate FuzzyART nets for them. This leads to a large set of models, whose proper use is predicated on a valid situational analysis module to select among the models. I believe a better solution is to design one model which accounts for the effects of surrounding vehicles on the driver's behavior.

Furthermore, the previous work shows a lack of attention to curve negotiation, and focuses on time to lane crossing, which may not be a good estimate of danger while negotiating curves, due to curve cutting.

Finally, there is the issue of user acceptance and deployability, which is related to the predictability of the system. The above systems, particularly Daisy, use complicated models. These complicated models may produce reactions which are not easily predictable by a user. Unfortunately, it is difficult to confirm or deny this as there have been no real world trials with Daisy (which is another weakness). The issue of driver acceptance is discussed further in Section 5.6

3. The Rapidly Adaptive Lateral Position Handler (RALPH) and Navlab 8

3.1. Vehicle Description

Our primary testbed is Navlab 8, an Oldsmobile Silhouette mini-van, which is depicted in Figure 1. The mini-van has been modified by the addition of actuators on the steering column and throttle pedal. A 180 MHz. Pentium Pro is located in the back, and is used for all processing.

A CCD camera is mounted on the windshield, underneath the rear-view mirror. This camera is used by RALPH for lane tracking and vision based obstacle detection. A radar obstacle sensor made by DELCO Electronics is mounted behind the front license plate, and is used for detecting vehicles directly ahead and to the front-left/right. Two side sensors are mounted on the sides of the vehicle, near

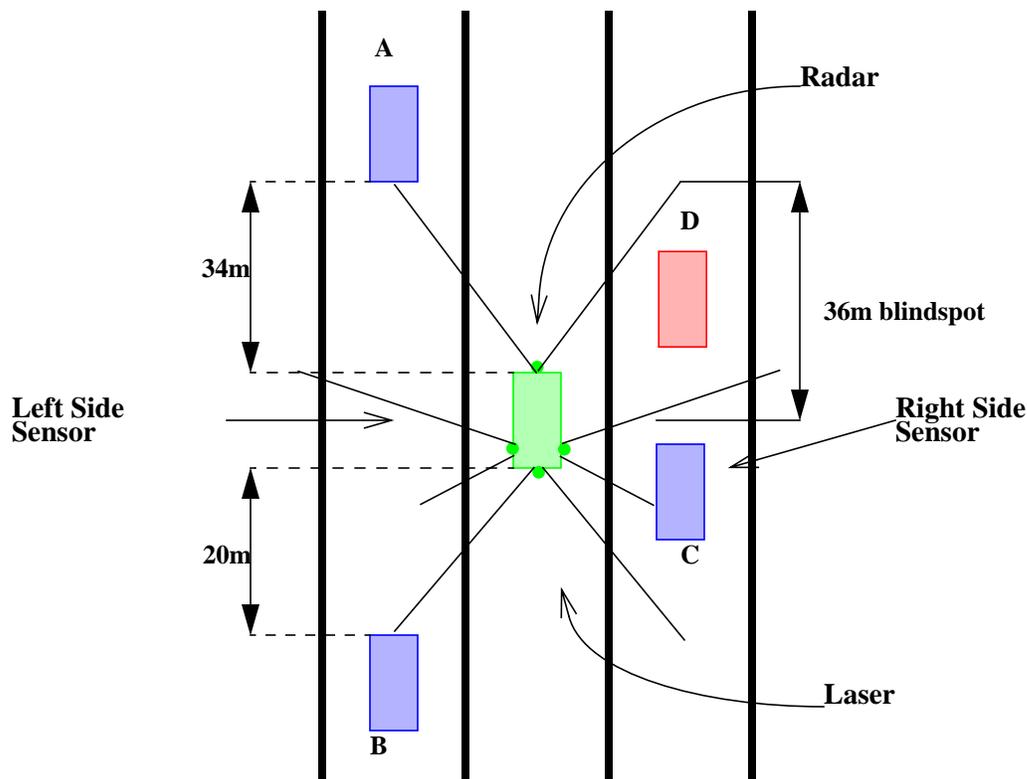


Figure 2: Depiction of Navlab 8 Sensor Coverage. The center vehicle is Navlab 8. Vehicles A, B, and C are sensed, vehicle D is in a blind spot.

the rear. Finally, a single line laser range finder is mounted behind the rear bumper. Sensor placement is shown in Figure 1, and range and resolution is given in table 1. Note that 360 degree sensor coverage is not available. Sensor coverage is shown in Figure 2. From it, you can see that in adjacent lanes, vehi-

Table 1: Sensor Description

Sensor	Range	Field of View	Resolution
Delco Radar	120m	12 degrees	1m/range, 2 degrees
Laser	120m	20 degrees	1 cm.
Blind Spot Sensor	3m	~70 degrees	Binary only

cles can be seen once they are about 34m ahead. These blindspots are due to the lack of side sensors on the front left and right of the van.

Besides sensors for obstacle detection, Navlab 8 also has a Differential Global Positioning System receiver, which has a resolution of +/- 3-5m. A yaw-rate gyro is mounted in the rear, along with a tilt sensor. These allow for better curve handling.

3.2. RALPH

The RALPH system [35] combines a lane tracker (RALPH), lane departure warning system (RALPH-WS), obstacle map generator (OPIE), and vehicle controller (PILOT). Besides these four main modules, there are numerous libraries developed to interface to different sensors and controller hardware. In development for the last 3 years, RALPH is capable of controlling a vehicle at highway speeds while tracking the lane as accurately as a typical human driver. The system is also capable of headway maintenance and lane changes under autonomous control.

RALPH can also be used as a data logger, under both autonomous and manual control. The default RALPH data record contains 51 fields, such as lateral displacement, upcoming road curvature, current vehicle curvature, yaw, yaw rate, velocity, obstacle information, road visibility, pitch, time to lane crossing, and vehicle latitude and longitude.

In addition to this, OPIE collates data from the various sensors, and provides an obstacle map in vehicle-centric coordinates. The position and relative velocity (in X and Y) of the nearest vehicle in Navlab's 6-neighbor (front, front left, front right, etc.) are provided.

4. Pre-Proposal Work

This section describes work that was done to support assertions made in Section 1.1, in which I claimed that individual drivers display different characteristics. The work has been done on two different sets of data. First, the data sets will be described, followed by experimental results on each set. The experiments performed include calculation of first order statistics, to look for blatant differences in driver style, along with neural modeling for steering prediction, similar to Nechyba's work.

4.1. Datasets

Two data sets are analyzed. One was collected using RALPH on a semi-truck, and the second was collected using RALPH on Navlab 8.

4.1.1. Carnegie Mellon Research Institute Data

The Carnegie Mellon Research Institute (CMRI) collected data on 8 truck drivers over a series of runs, mostly along the Pennsylvania Turnpike (I-76) between Pittsburgh and Philadelphia. The recording was done using RALPH, with a subset of the normal fields recorded. The recorded fields include lane position, road curvature, steering angle, turn signal state, velocity, and system uncertainty. See Figure 3 for examples of these signals. Post processing of the data was done to add fields for steering wheel velocity, lateral velocity, and time to lane crossing. This post-processing revealed problems during the data collection that resulted in the data of four drivers being eliminated for various reasons.

The normal RALPH reliability estimate is fairly instantaneous, and can briefly indicate low confidence when going under overpasses and when illumination changes. Even though the immediate confidence may be low, filtering in RALPH produces usable lane position estimates. Therefore, this uncertainty measure was filtered to look for average time between periods of low certainty within a given time frame. If the average time between uncertain measurements dropped below 8 seconds over the past minute, the entire region is marked uncertain. This produces a very clean usability signal.

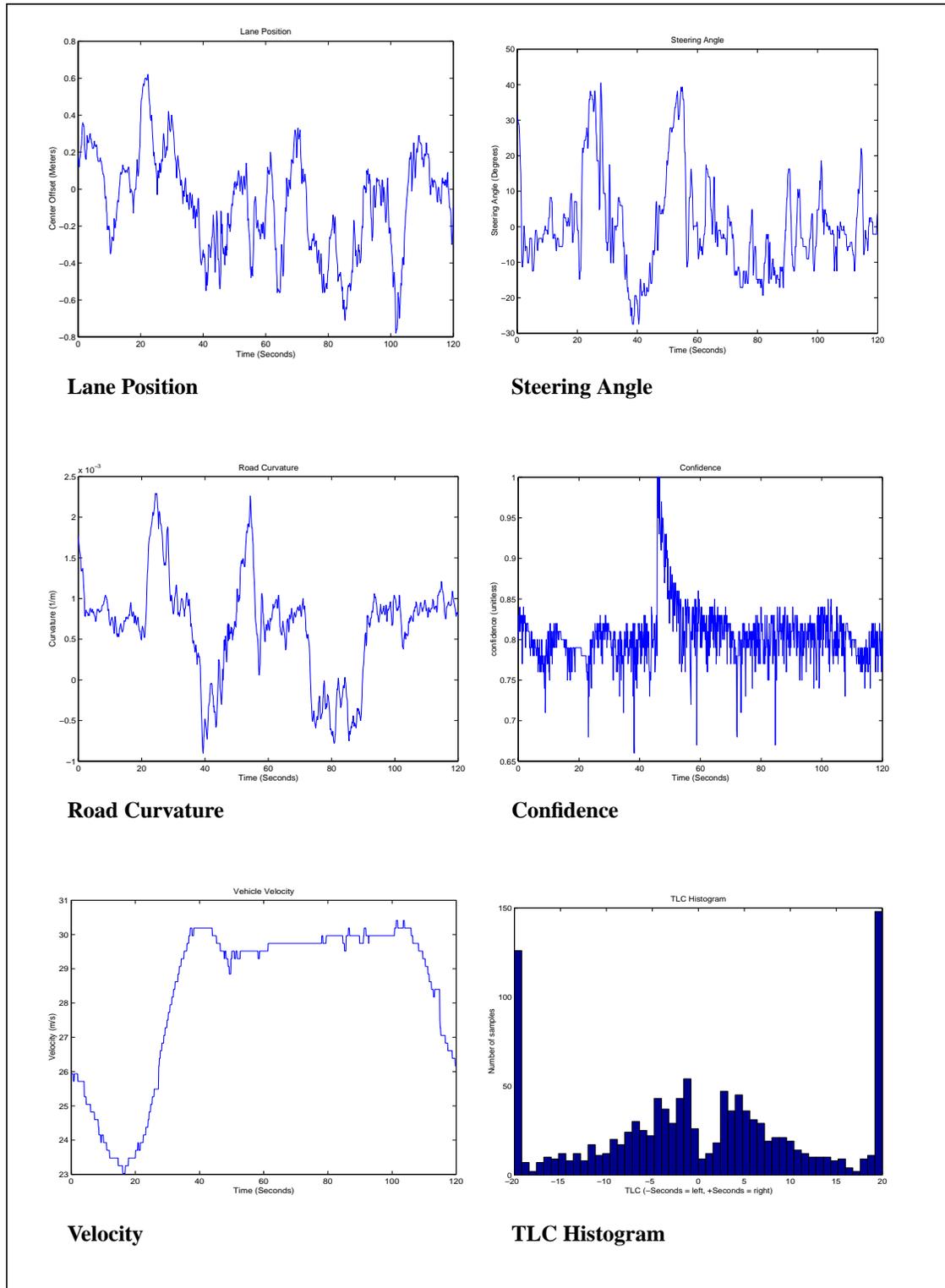


Figure 3: Plots of CMRI Truck Driver Data.

There is a caveat to this data. We recently found out that there is non-uniform bias in the road curvature estimates. Given the length of the runs, we would expect to see a mean curvature near 0. Most likely, this bias is due to minor shifts in camera yaw between runs. While the bias in curvature can be subtracted out, the possible shift in camera position may also cause a small error in the lane position estimate. However, Pomerleau [34] has determined that while a 1 degree shift in camera yaw can cause a straight road to appear as a 1200m radius curvature, the overall effect on lane displacement is negligible due to redundancy in the method used by RALPH to calculate lane position.

4.1.2. Initial CMU Data Study

Description

The initial user study, which is currently in progress, will use Navlab 8, a mini-van, to collect data from approximately 20 drivers, of both genders and over a range of ages (21-50). Potential subjects are required to have a valid US driver's license, and at least 4 years of driving experience in the US, with no major traffic violations, accidents, or DUIs. The subject is told only that we are interested in learning about driving behavior, for use in a possible warning system of some kind. Details are kept sketchy, to help avoid biasing the driver's behavior. The driver is also told that various information will be recorded, and that a video will be kept of the driver's eye view of the road (that is used by RALPH). Nothing that identifies the driver is recorded. The data that is collected is very similar to the CMRI data, except that it has less noise (because we're using a newer version of RALPH), and position and velocity data is recorded for surrounding vehicles, using OPIE as described in Section 3.1

The route is from Carnegie Mellon University to Grove City, which is 50 miles north of Pittsburgh. The route is primarily two lane (in each direction) highway driving, with short stretches of three lanes. This allows for nearly 1.5 hours of data on each subject. The driver is not told how to drive. The only instructions are to drive safely, and to try to remember to use the turn signal when changing lanes. I am present in the van during the test run, sitting in the passenger seat. The touch screen that displays the RALPH user interface, which is normally visible from the driver's side, is turned, and has an opaque hood over it, keeping it from view of the driver.

Experimental Effects

One concern is that the subject most likely has never driven a Silhouette, or even a mini-van. A mini-van is large enough that it is hard to get a good feel for the boundaries and available space, particularly on the right hand side. Due to this, most drivers initially tend to hug the left side of the road. However, this effect seems to subside within a half hour or so of driving. Therefore, all of the analysis was performed on data collected on the trip back, by which time the subject is more familiar with the space available to him, and hopefully, is displaying driving tendencies which are more natural to him, rather than induced by the unfamiliarity of the mini-van.

Another problem is nervousness due to driving an expensive vehicle, being recorded, and having an experimenter present. Subjects have told me that they felt a bit tense, and were careful while driving. This can have the effect of reducing variability in driving behavior, which is exactly what I am looking for. I currently see three possible solutions. The first is to develop a version of RALPH which can be easily installed in a subject's vehicle. This would allow for unobtrusive monitoring, without having the experimenter present, in a vehicle the subject is comfortable with. An alternative to this is to loan out Navlab 8 to people who are making long (1+ day) trips, in the hopes that the drivers would get used to

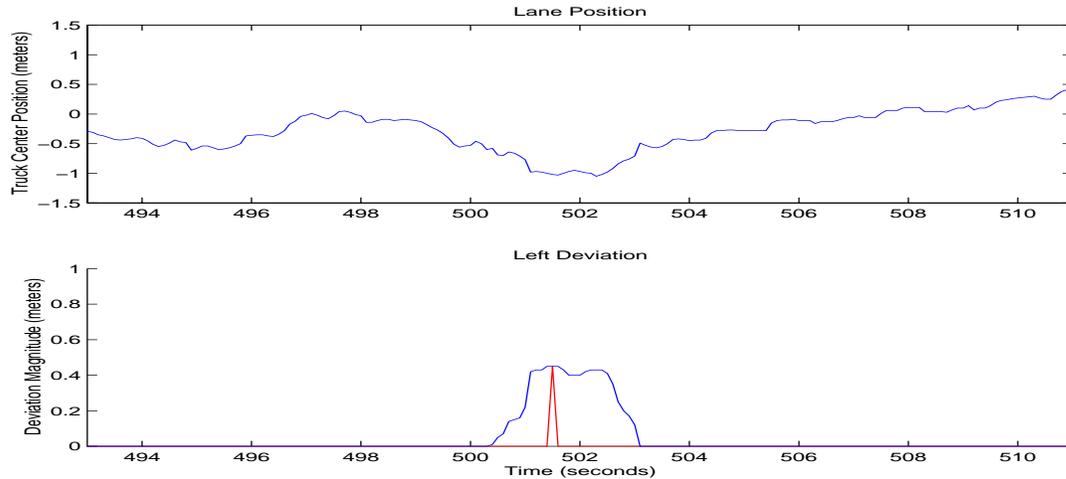


Figure 4: Lane excursion and corresponding alarm

the feel of the mini-van, and allow their normal driving style to express itself. At this point, the subject pool isn't large enough to have a good feel for how much these issues are affecting the data. The final possibility is the development of an accurate vision based car tracker. This would allow me to collect data on other drivers as I follow them, removing all experimenter effects.

4.2. Alarm Analysis

This section looks at the number of alarms generated by methods similar to the RALPH warning system at different alarm thresholds. The purpose of this experiment is to demonstrate that current alarm threshold schemes (such as lane position and TLC) are not always appropriate. This is because to reduce the number of triggers, a relatively low threshold must be selected.

There are two parameters which can be adjusted. The first is Time to Lane Crossing. An alarm can be triggered when the TLC value drops below a certain preset threshold, signalling an impending lane deviation. However, there is a problem with this metric. Namely, from my analysis of truck driver lane position, some truck drivers spend as much as 30% of their time with one wheel on or outside a lane boundary. By convention, this situation has a TLC of 0 seconds.

To get around this problem, a “virtual lane boundary” can be created, which effectively widens the lane. The width of this virtual lane boundary is the 2nd parameter which can be adjusted. Now, a TLC threshold can be used with the virtual lane. Note that a TLC threshold of 0 seconds is the same as triggering when a tire is just past the virtual lane boundary.

I computed the virtual lane boundaries required to maintain an alarm rate of between 1 and 5 alarms per hour, with a TLC of 0 seconds for the CMRI truck data. To do this, I generated alarms using no virtual lane, then picked the top n alarms, and extended the lane width to just past where the top $(n-1)^{th}$ alarm occurred. I excluded all data 10 seconds before and after a lane change (as marked by the beginning and end of the turn signal). I also differentiated between lane excursions and alarms. An excursion is the period of time from when an edge of the truck goes past a lane boundary to when it returns. One alarm is generated for each excursion. Figure 4 shows this. The top plot is the truck's lane position. An approximately 3 second deviation occurs beginning around 500.5 seconds.

Table 2: False Alarms Generated by RALPH-WS for Truck Drivers with no Virtual Lane Boundary

	Left Deviations	Right Deviations	Straight Roads	Left Curves	Right Curves	Total Alarms
Driver 1	7	24	14	1	16	31
Driver 2	78	30	44	37	27	108
Driver 3	165	61	60	117	49	226
Driver 4	162	42	36	127	41	204

Before computing alarm rates with virtual lane boundaries, I computed the number of alarms that occur with no virtual lane boundary, and a TLC of 0. This is what a current lane departure warning system such as RALPH would produce, if set to minimize the number of false positives. This is not entirely true, however, as RALPH does allow the user to exceed the lane boundary by a tunable amount. Table 2 gives the numbers of alarms triggered by motion to the left and the right. These alarms are further broken down by roadway geometry, i.e., straight roads, left curves, and right curves. A curve is defined as any segment of road with a radius $< 1500\text{m}$. The numbers are computed for one hour of four different truck drivers.

Driver 1 has the lowest false alarm rate, with 31/hour. Driver 4 would generate 240 alarms in 1 hour, if exceeding lane boundary were the only criteria for triggering an alarm. These numbers would be even higher if a non-0 TLC were used.

After computing virtual lane boundaries for 1-5 alarms/hour, the alarm rate does drop as expected. However, the required lane boundaries, which are shown in Table 3, are quite large. To maintain an alarm rate of 5/hour, driver 4 would need 0.75 meters on each side as a cushion. Driver 1 seems more reasonable, as he would need 0.35m on the left and 0.43m on the right for 1 alarm/hour

Determining the proper selection of TLC threshold and virtual lane boundary width to achieve a warning time is difficult. The naive solution is to extend the virtual lane boundaries to encompass all the deviations of the truck driver, and then set the TLC threshold to the desired warning time. However, as can be seen in Table 3, this would result in very large boundaries for some drivers. The average lane width is 3.6m. To allow a driver to encroach upon adjacent lanes by as much as 0.85m is unacceptable.

It would be desirable to minimize the size of the virtual lane boundary, as when the TLC threshold is set to 0 seconds, the driver is allowed to drive on the boundary, and I feel that it is appropriate to try and keep the driver within as tight a boundary as possible. However, doing so has an effect on the number of false alarms. This effect depends on the lateral velocity profile of the driver (drivers who exhibit large, rapid steering reversals would tend to have more alarms as they will have higher lateral velocities, and hence, lower TLCs), and the TLC threshold selected. I need to conduct simulation studies to gain a better understanding of the interaction between TLC threshold, virtual lane boundary width, driving style, and alarm rate.

Table 3: Virtual Lane Boundaries

Driver: Alarm	Left Boundary	Right Boundary	Driver: Alarm	Left Boundary	Right Boundary
Driver 1: 1	0.3500	0.4300	Driver 3: 1	0.7900	0.5100
Driver 1: 2	0.3500	0.2500	Driver 3: 2	0.7800	0.5100
Driver 1: 3	0.2800	0.2500	Driver 3: 3	0.7600	0.5100
Driver 1: 4	0.2500	0.2500	Driver 3: 4	0.7500	0.5100
Driver 1: 5	0.2500	0.1800	Driver 3: 5	0.7100	0.5100
Driver 2: 1	0.5700	0.5500	Driver 4: 1	0.8500	0.7500
Driver 2: 2	0.4100	0.5500	Driver 4: 2	0.8100	0.7500
Driver 2: 3	0.4100	0.4300	Driver 4: 3	0.7900	0.7500
Driver 2: 4	0.4100	0.3500	Driver 4: 4	0.7700	0.7500
Driver 2: 5	0.3900	0.3500	Driver 4: 5	0.7500	0.7500

Another factor is the nature of the excursions. I need to further investigate two things. The first the distribution of the length of the excursions for different drivers, and the second is the distribution of the magnitude of the excursions, i.e., how far beyond the lane do they get. If a driver spends a great deal of time with one side of his truck 5cm beyond a lane boundary, it is acceptable to set the virtual lane boundary to encompass that, as it is part of his driving style. However, if the majority of the excursions are short and high magnitude, then it may be reasonable to consider them true alarms, and not adapt to them.

Unfortunately, there is no contextual information with the truck driver data. I do not know which lane the driver was in during the deviations. It is possible, although unlikely given the large number of deviations, that the encroachments were all onto wide shoulders, and not onto adjacent lanes.

The above ideas for further investigation are useful to try and *improve* current TLC-based warning systems to handle highly variable drivers. However, I believe that the experiments on alarm rates and required virtual lane boundary widths shown in Tables 2 and 3 demonstrate that TLC based warning systems in their current form are inadequate, and that there are serious drawbacks to the whole method. I feel that learning normal driver response to situations could perform better than TLC, even with the addition of virtual lane boundaries, by learning what types of deviations are normal for the driver, and what types are not. This idea is tested using data collected from passenger car drivers in Section 4.4. The purpose of these tests, therefore, is not to put forth virtual lane boundaries as the solution to false alarms for truck drivers. Rather, it is to show how much time truck drivers spend outside of the lane boundaries, and how that makes current approaches to lane departure warning systems fail for these types of drivers.

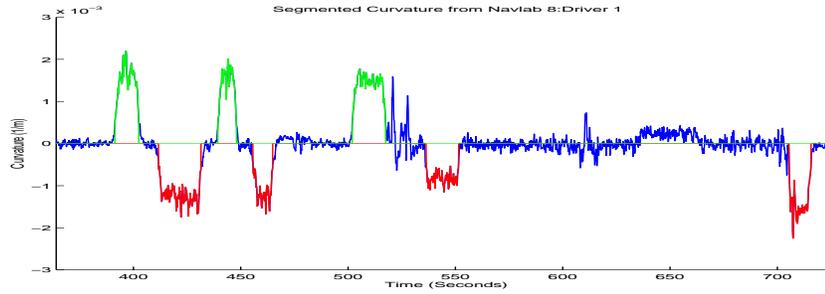


Figure 5: Plot of curvature segmentation of mini-van driver. Blue is straight, Green is right curve (positive curvature), and Red is left curve (negative curvature).

Table 4: Truck Driver Lane Keeping Statistics

	Straight Mean	Straight Stdev	Left Curve Mean	Left Curve Stdev	Right Curve Mean	Right Curve Stdev
Driver 1	0.1153	0.2894	-0.1154	0.3281	0.0175	0.1488
Driver 2	0.0067	0.3209	-0.3759	0.3079	0.1697	0.2643
Driver 3	0.1210	0.5864	-0.4201	0.2996	0.1127	0.3334
Driver 4	-0.1397	0.3607	-0.5089	0.2947	0.0940	0.3498

4.3. Driver Differences

This section contains results from experiments performed to examine the differences between individual drivers, and variation within a single driver. I look at basic statistical differences in lane position, to explore how distinct individual drivers are at such a coarse level.

To begin looking at whether or not differences in driver behavior are significant, I first computed statistics for 1 hour of 4 different truck drivers and 5 drivers from my own study. These statistics are computed separately for straight stretches, left curves, and right curves. I look at curves separately to get an idea of the amount of curve cutting that different drivers display. For the mini-van data, a curve is defined as any segment of data in which the immediate road curvature is less than 2000m. For the truck driver data, the threshold was set at 1500m, as this data is noisier and a tighter cutoff gave better qualitative results. Filtering is done to eliminate curves that are less than two seconds in duration, as it is unlikely that there is much curve cutting in short curves. Figure 5 shows a sample stretch of road curvature data (taken from mini-van data), segmented in straight, left curve, and right curve areas. While the segmentation is not perfect, simple thresholding does properly classify the majority of points.

Table 4 shows statistics for the truck drivers, and table 5 shows statistics for drivers from my study, taken on Navlab 8. From this data, it is pretty clear that there are gross differences in lane keeping performance. For instance, truck driver 4 drives to the left of center while on straight roads, while drivers 1 and 3 are off to the right. Truck driver 3 significantly cuts curves when going left, but does not really do so while turning right.

Table 5: Navlab 8 Driver Lane Keeping Statistics

	Straight Mean	Straight Stdev	Left Curve Mean	Left Curve Stdev	Right Curve Mean	Right Curve Stdev
Driver 1	-0.1045	0.2742	-0.2342	0.2553	0.0904	0.2466
Driver 2	0.0354	0.3598	0.0028	0.2316	0.0595	0.5096
Driver 3	-0.1034	0.3214	-0.1479	0.1891	0.1612	0.2247
Driver 4	-0.3307	0.3590	-0.4987	0.3123	-0.0399	0.3570
Driver 5	-0.1247	0.2979	-0.1739	0.2256	-0.0141	0.2990

Table 6: Maximum Differences in Lane Position Mean Over the Course of a Day

	Dr1, Day 1	Dr1, Day 2	Dr2, Day 1	Dr2, Day 2	Dr3, Day 3	Dr3, Day 2	Dr4, Day 1	Dr4, Day 2
Mean Spread	0.15m	0.22m	0.17m	0.24m	0.20m	0.23m	0.26m	0.13m

Navlab driver 4, who is the only one of the Navlab drivers who has experience driving a van, tended to stay off to the left, and was slightly left of center even while making right turns. Driver 2 was in the center of the lane, although his standard deviation during right curves was quite high. Numerous conclusions such as these can be drawn about the drivers in the above tables.

The same driver also displays differences in mean lane position as a function of time. The truck drivers, who were driving 8-10 hours a day, showed drifts of up to 25cm over the course of a day, when the lane position average was computed in (approximately) hour-long segments. While this may not be a large amount, the trucks are wide enough that a perfectly centered driver only has about 0.5 meters of free space on each side. Table 6 shows the maximum spread of mean lane position for 2 days of 4 different drivers. Each day is segmented into hour long chunks, and lane changes, curves, and unreliable data are filtered out.

The above data shows that there are differences in long term, gross behavior of drivers, both between and within drivers, which points to a need for adaptation. The real, more difficult question is whether these differences exist when looking at short term (on the order of seconds) driver behavior. For instance, when do drivers reverse their steering direction to maintain safe lane tracking? Although I need to do experiments to bear this out, I believe that there are differences at smaller time scales.

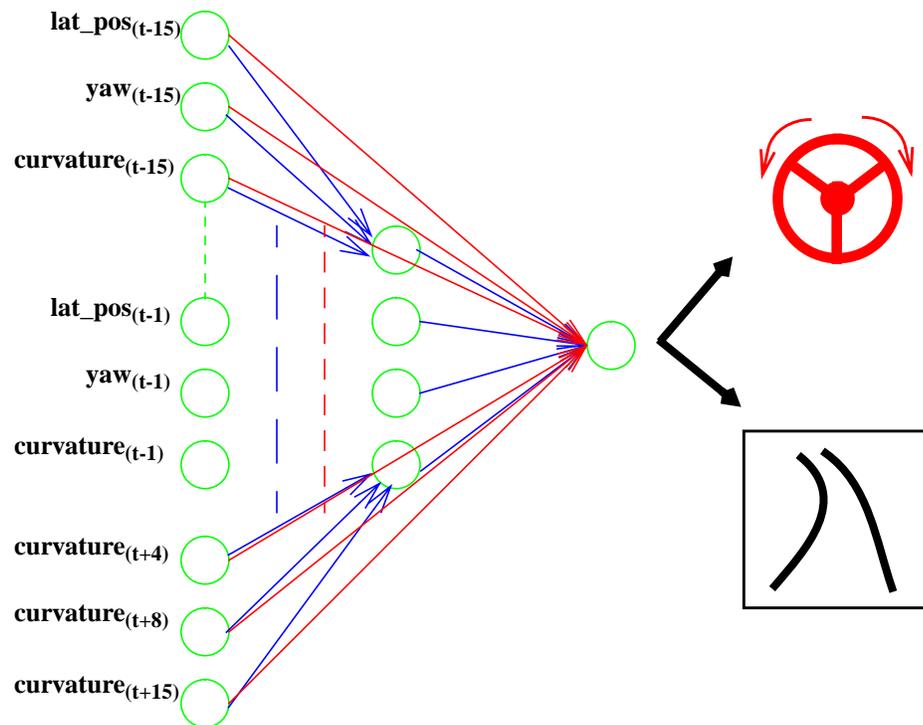


Figure 6: Neural Net Architecture. The inputs are a history of lateral position, yaw, and curvature, plus upcoming curvature. The output is a single value representing steering wheel angle or vehicle curvature.

4.4. Neural Network Based Warning System

One possible method of implementing a driver adaptive warning system is to use a neural net to predict the driver's actions based on previous driving examples. Feraric [15] and Nechyba [29] have taken this approach. My own experiments take a time series of vehicle state as input, and attempt to predict steering wheel position or vehicle curvature. The purpose of running these experiments is not to determine if a neural net can learn to drive. Rather, it is to evaluate the performance of a net trained on driver *a* against data from driver *b*. In other words, can it learn an interesting driver characteristic that is specific to the training driver, and is consistent over different examples of the same driver?

4.4.1. Neural Net Architecture

The architecture I have chosen for my experiments is a multi-layer perceptron using the Error Back-propagation (BP) [38] training algorithm. BP is simple, yet powerful enough to capture non-linear effects, and trains fairly quickly. It is therefore worth trying before testing more complicated learning schemes. The network, which is shown in Figure 6, takes a 15 sample (1 second) history of lateral position, yaw, and road curvature, along with road curvature from 0.25, 0.5, and 1 second ahead. Therefore, there are 48 input units. I (empirically) chose 4 hidden units, and 1 output unit to represent

Table 7: Mini-Van Driver Dataset Statistics

Driver	Train Set Start Time	Train Set End Time	Train Set Mean LanePos	CV Set Start time	CV Set End Time	CV Set Mean LanePos	Test Set Start Time	Test Set End Time	Test Set Mean LanePos
Driver 2	00:15:47	00:17:50	-0.0536	00:18:04	00:20:29	0.1130	00:25:02	00:27:52	0.0776
Driver 3	00:13:45	00:16:44	-0.0608	00:16:44	00:20:44	-0.0617	00:20:44	00:26:15	-0.0286
Driver 4	00:10:33	00:13:15	-0.2796	00:17:31	00:19:39	-0.1918	00:19:39	00:23:59	-0.1950
Driver 5	00:05:39	00:07:48	-0.0876	00:07:48	00:14:29	-0.0976	00:14:29	00:18:34	-0.1002
Driver 6	00:14:42	00:17:06	-0.2490	00:17:45	00:22:06	-0.2958	00:30:59	00:35:12	-0.2835

Note: Mean Lane Position is in meters, and time is in hh:mm:ss format. Times are relative to trial start time.

steering direction or vehicle curvature (the use of vehicle curvature is explained below). Shortcut connections are used from the input layer to the output layer to capture the linear relationship between the input and output. Training was done using 2-2.5 minutes of data, with the training stopped when cross-validation set error was minimized.

4.4.2. Datasets

The datasets, which consist of training, cross-validation (CV), and test sets were taken from 5 mini-van drivers. I chose to use the mini-van drivers over the truck drivers for these initial experiments as the truck driver data has more noise, and I have video of the mini-van driver's views. As described in Section 4.1, each mini-van run was a round trip from Pittsburgh to Grove City, PA. I do not use data from the Pittsburgh to Grove City segment, as drivers are becoming used to driving a mini-van during this time. For each driver, I tried to find training, CV, and test sets with similar gross statistics, although this was not always possible. The mean lane position, along with start and end times (relative to the beginning of the run from Grove City to Pittsburgh) of each data set are shown in Table 7. Note that training sets are generally 2-3 minutes in length, while test sets are 3-6 minutes in length. I tried to equalize the lengths of the training sets. However, the presence of lane changes in the data made that difficult, as I did not want to include them in the datasets. I also tried to include similar curves in each data set, although that too was difficult, due to the nature of the segmentation. Note that while the mean lane positions of drivers 3, 5, and 6 do not vary more than 0.05m, the mean lane positions of drivers 2 and 4 vary by up to 0.15m. This, as we will see in the following section, has an impact upon the neural network training and generalization performance.

4.4.3. Results

Test Set Results

The first test trained the BP net on 2.0 minutes from driver 2, and tested against segments from drivers 2-6. The test segments were 4-5 minutes long. Driver 1 was not included in these trials because a software problem caused his data to be sampled at half the required rate. Figure 7 shows the results of evaluations of the net on driver 2 and driver 4.

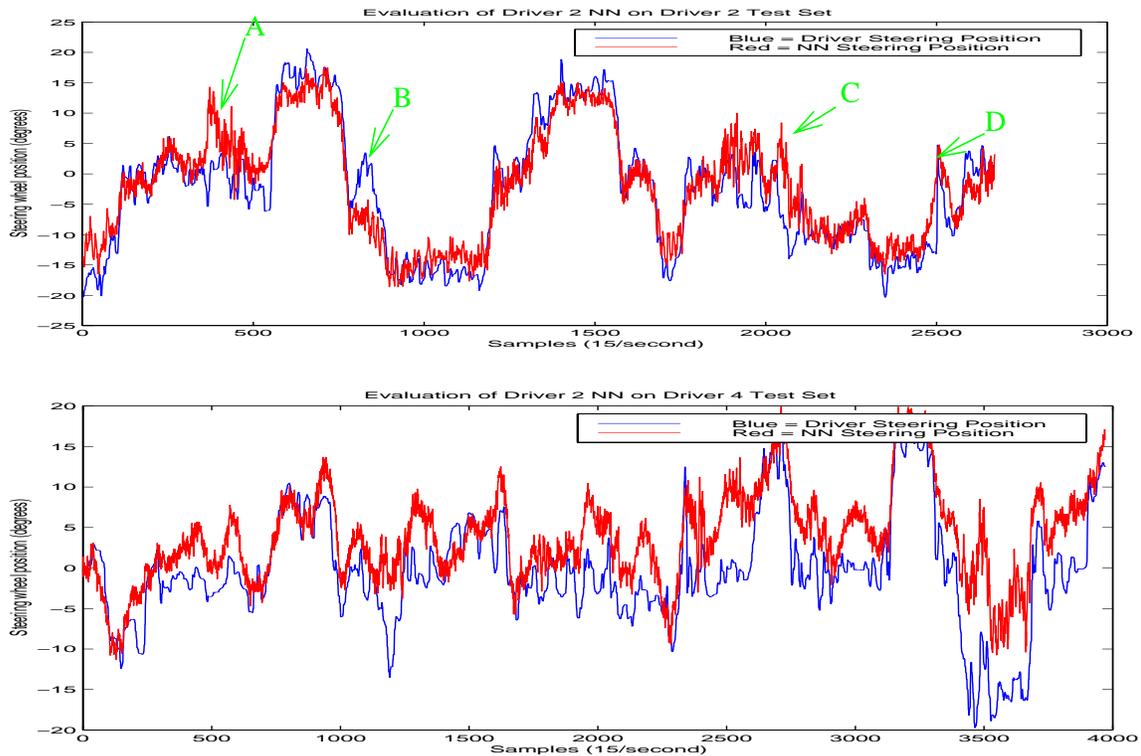


Figure 7: These plots show the result of evaluating an NN trained on driver 2 on drivers 2 and 4. The top plot is $NN_{(2,2)}$, and the bottom plot is $NN_{(2,4)}$.

From now on the notation $NN_{(n,m)}$ will denote the results of the evaluation of a test set from driver m tested using an NN trained on driver n . The top plot shows $NN_{(2,2)}$, and as expected, the qualitative fit is good. Driver 2, as shown in Table 5, keeps to the center of the lane. Overall, the NN trained on driver 2 attempts to steer to keep the vehicle in the center of the lane as well.

There are relatively large discrepancies at points A-D, marked on the plot. Points B and C were caused by perceptual glitches. RALPH briefly (for about 1.5-2 seconds) mistook a straight road for a curving road. Therefore, the NN predicted a steering deflection to follow the curve. Point A appears to be a driver-induced oscillation -- first to the right, then to the left. The movement to the left was greater than that to the right, and the net attempts to compensate by steering to the right, to try and bring the vehicle back to the center of the lane. Point D is another deviation to the left. The data indicates that there is no other vehicle on the right. Therefore, the deviation was either driver or crosswind induced. At both A and D, the TLC values briefly dipped to 2 seconds.

The 2nd plot, showing $NN_{(2,4)}$, however, does not match the expected output very well. This is understandable, as driver 4's mean lane position is almost a foot to the left, and the net has been trained on a driver who drives down the center of the lane. The net's steering output is more rightward than the driver, because it keeps trying to pull the vehicle to the center of the lane.

Table 8 shows the RMS steering error (in degrees) of $NN_{(x,y)}$, $2 \leq (x,y) \leq 6$. I.e., row n indicates the results of evaluating drivers 2-6 on a net trained on driver n . The boldfaced entry for row n indicates which driver the net trained on driver n performed best on. Ideally, the diagonal entries would be boldfaced, as that would indicate the NN is properly classifying drivers. However, only two out of the 6 drivers (Drivers 3 and 6) are properly classified. Drivers 2 and 5 are mis-classified. However, the dif-

Table 8: NN Classification Results (RMS Degrees) of Mini-Van Drivers

	Driver 2	Driver 3	Driver 4	Driver 5	Driver 6
Driver 2	4.2740	3.7019	6.3180	4.0165	6.3794
Driver 3	4.3852	2.6641	3.4262	2.7311	2.9751
Driver 4	3.5159	2.4826	3.2087	2.8144	2.9088
Driver 5	5.9210	3.6108	4.5027	3.5946	3.1631
Driver 6	3.5780	2.9457	3.0660	3.2385	2.9349

ferences between the correct classification and the network classification for these drivers are less than 0.6 degrees. Driver 4 is an anomaly. The network trained on driver 4 performs very poorly when evaluated on driver 4's test set. However, there is a 0.0850m difference in mean lane position between the training set and test set for this driver, which could be a possible reason.

What is interesting to note is that all drivers were classified into one of two categories: Driver 3 or Driver 6. From Table 7, you can see that Drivers 3 and 6 both had consistent mean lane positions in their respective training, CV, and test sets. The test sets of the drivers who were mis-classified mostly had mean lane positions which were similar to the training set mean lane positions of the categories into which they were classified. Driver 4, however, was classified as driver 3, when by mean lane position, it seems closer to driver 6. I am not sure why this is. I will have to look more closely at this driver, along with the net trained on this driver to determine what is happening.

I did the above experiments to determine whether or not a BP trained NN could be trained as a "canonical" driver model - one which learns a consistent, yet interesting feature of a driver. The results indicate that the model I have chosen does not completely accomplish this. While this is a weak result in driver classification, it does not invalidate the use of this approach as a driver specific warning system. One explanation for this is that driver differences were muted in this data, due to the subject's lack of familiarity with the test vehicle, and their nervousness. Furthermore, the test sets were up to 10-15 minutes apart from the training sets, and the road geometry changes a lot on the route we drove. The distance between training and test sets was necessary, as the sets were segmented by lane changes, and some drivers performed many lane changes, making it difficult to find stretches longer than 1 or 2 minutes. I expect an on-line, continuous learning system would have better results, as it would be able to update its model as conditions changed. Another possible reason for the weak performance is that I only included raw lane position, yaw, and curvature as inputs to the net. Adding pre-processed inputs, such as providing information on steering behavior (which I believe can differ between drivers) could improve results. Furthermore, no training set management was done to prevent network bias from occurring.

Lane Change Results

While the above results are interesting, the true purpose of this experiment is to gauge the performance of my neural net approach as a lane departure warning system. To do so, I evaluate a chosen neural net on a set of 'true' alarms, to see how it would react. In this case, a true alarm is a lane change. The goal of these tests is to see at what point during a lane change does the predicted steering output begin to differ from the driver's steering. The earlier this happens, the better.

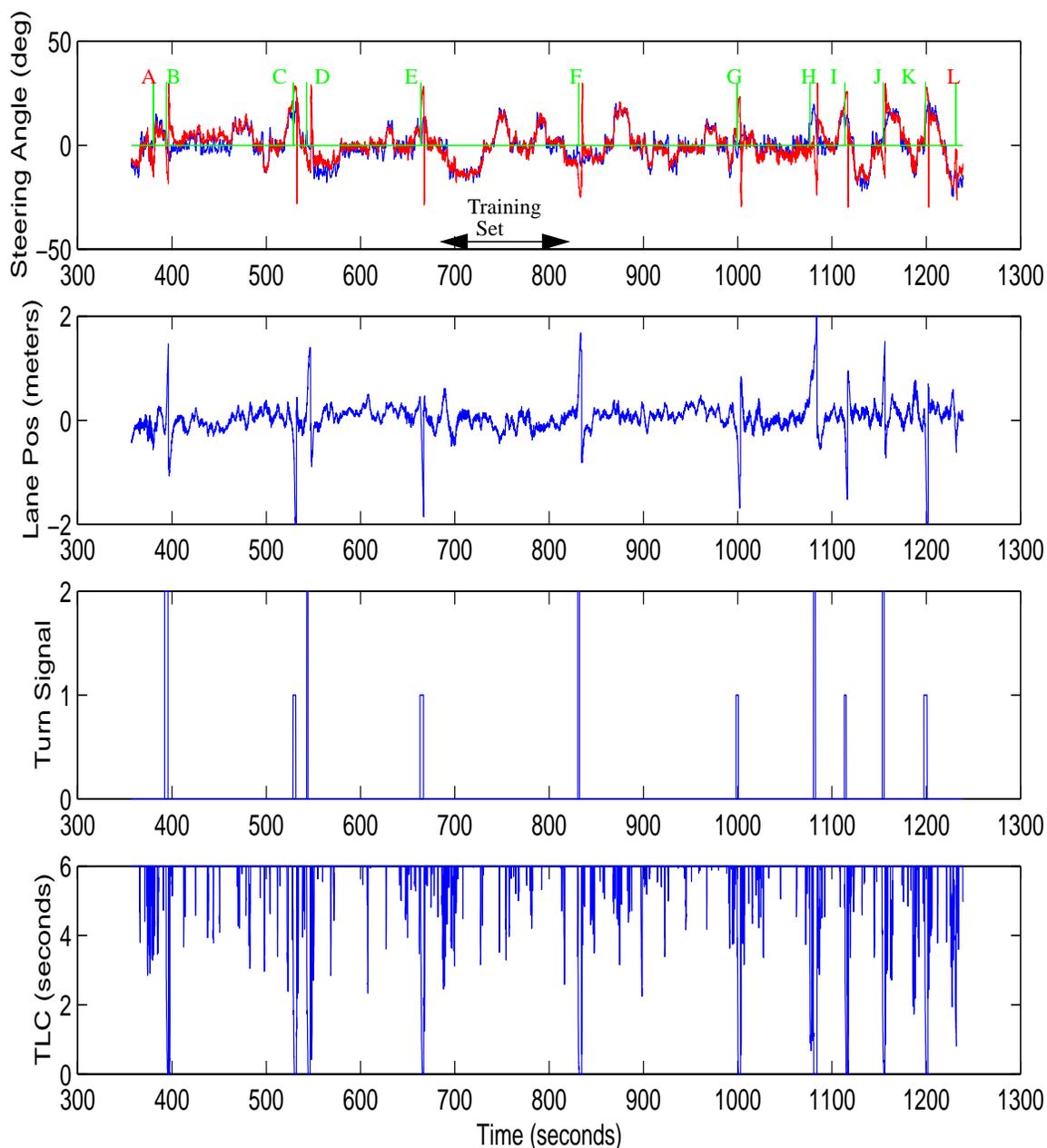


Figure 8: Results of lane change detection. The green spikes in the top plot indicate where a lane change is detected.

Figure 8 shows the results of evaluating nearly 15 minutes of data from driver 2 on a network trained on driver 2. I picked driver 2 for this experiment because even though his short term behavior was similar to other drivers (in terms of drift and variance of lane position), the long term behavior (on the order of 15+ minutes) was more stable than the other drivers. This is an advantage for off-line training, because the net only needs to be trained once. For other drivers, the mean lane position varies more. I believe that an on-line system which continually retrained the model would work well for these drivers, whereas the current off-line system would not, as I do not train multiple models for a single driver.

Table 9: Comparison Between NN and TLC Alarm Response

Lane Change	TLC Value (seconds)	Warning Time Gain (seconds)	Lane Change	TLC Value	Warning Time Gain
B	2.34	0.54	G	5.81	1.06
C	1.80	0.45	H	4.38	0.39
D	6.00	1.09	I	5.41	1.01
E	3.19	0.60	J	0.58	-0.13
F	0.48	-0.22	K	1.16	0.06

The top plot is an overlay (in red) of the NN steering prediction over the driver's steering output. The green spikes indicate where the difference between the predicted and actual output is greater than 12.8 degrees. This threshold was empirically selected to achieve a false alarm rate similar to that of a TLC system, which is used as a baseline, and is explained below.

This test set contains 10 true lane changes, but there were 12 alarms. In Figure 8, the true lane changes (marked B through K) are labelled in green, while the false alarms (marked A and L) are labelled in red. After each trigger, the alarm is disabled for 10 seconds, to prevent multiple triggers for the same lane change. Table 9 shows results for the 10 valid lane changes. These results come from comparing the performance of a TLC based warning system with the NN based warning system. The column marked 'TLC Value' shows the TLC values at the time the NN prediction generates an alarm for each of the true lane changes. In other words, it is what the TLC threshold would have to be to generate an alarm at the same time as the NN. To get a measure of how much earlier the warning is given using the NN, I also generate alarms with a TLC system using a TLC threshold of 1.0 second, although these points are not shown in Figure 8 to reduce clutter. Other studies have used TLC thresholds of 0.7s and 1.3s [43]. I chose 1.0 to increase the warning time, and to take into account the experimental effect of the driver perhaps driving more carefully. The column of Table 9 marked 'Warning Time Gain' shows the *increase* in warning time provided by the NN system over the TLC system. Therefore, a positive number in this column indicates that the NN approach performed better than TLC by detecting the lane change earlier. A negative number means the opposite -- the TLC system detected the lane change before the NN system. I computed the difference in alarm times by hand selecting corresponding NN and TLC alarms and measuring the time between them.

Summary statistics are shown in Table 10. While the NN had more false alarms (2 vs. 1 in a 15 minute period), there was an average gain in warning time of 0.48s, which is significant. To achieve a similar warning time using TLC would require a threshold of 3.11 seconds, as this is the average of the 'TLC Value' column. I therefore set the TLC threshold at 3.11 seconds, and re-ran the experiment. This resulted in the TLC system generating 10 false alarms, which is significantly higher than the 2 produced by the NN.

It is interesting that to achieve (an average) gain of 0.48s in warning time, the TLC threshold had to be changed from 1.0s to 3.11s. By definition, it should have had to be changed to 1.48s. I believe this is because of the cases (D, G, H, and I) where the NN predicts the lane change significantly in advance of TLC. These lane changes are due to slow drift, which doesn't cause great changes in TLC (because of low lateral velocity), but does cause a prediction error in steering output.

Table 10: TLC vs. NN Alarm Response Summary

TLC Value Mean	3.11s	NN False Alarms	2
Warning Time Gain Mean	0.48s	TLC False Alarms	1

The results in Table 9 include the “outliers,” which are boldfaced or colored red. The boldfaced entries signify where the NN gave a much earlier warning than TLC, and the red entries (lane changes F and J) indicate where the NN triggers a warning after TLC. I picked lane changes D and J to study more closely, as they are at the extremes of the observations. Figure 9 shows a blown up version of Figure 7, centered around the 2 lane changes, with the TLC triggers added and marked in magenta.

D_{NN} and D_{TLC} are the trigger points of the NN and TLC on lane change D. What is interesting about this lane change is that it is a slow drift, compared to the others. From the video recording, this lane change maneuver is occurring on a relatively straight road. The driver just slowly starts to drift to the right. It is slow enough at points that the TLC doesn’t drop very fast, until just before the full lane change occurs. I feel that this example is very close to what could happen in a real ROR situation due to unintended steering input. The early trigger in this case, relative to TLC, is encouraging.

The triggers for lane change J are marked as J_{TLC} and J_{NN}. In this case, J_{TLC} occurs before J_{NN}. According to the video, the driver is in the left lane and decides to make a right lane change. However, the right lane change begins while the vehicle is in a right curve. The driver steers right, to follow the curve, and the net predicts this. However, he has to oversteer to actually change lanes. The net notices this, and tries to pull him back to the left. The driver does steer along the curve as he should, but only more so. This additional steering is detected by the NN, but not as quickly as in some of the other cases.

While the NN generally does better than TLC, it does have one more alarm over the 15 minute test. The two alarms are marked as points A and L in Figure 8. Alarm A is a false alarm which is not triggered by TLC. It was caused by a perceptual error, in which changes in vehicle pitch caused the curvature during a straight segment to actually appear leftward. It is therefore classified as a nuisance alarm, according to my definitions in Section 1.2. The second false alarm occurred for both the NN and TLC trials, and is a safe false alarm. This alarm was actually caused by a slow rightward drift during a left curve. The driver then corrected. The correction, however, was too great, and he overshot a bit and went to the left. A contributing factor to this alarm is perceptual in nature - the lateral position estimate is incorrect due to changes in lighting.

The perceptual error which caused the first alarm was not something which would affect TLC, as it cares only about instantaneous lane position and lateral velocity, and not current or future curvature. This is a problem that I will have to address, even though it is perceptual in nature. Another problem which I will have to deal with is the lack of consistency in training. I had to evaluate a few networks to find one which worked well enough for these results. I think that despite the use of a cross validation set, some overtraining was occurring. This is because the net that ended up doing best was not the net which had the lowest cross validation error. This is perhaps not that surprising, given that the training, cross-validation, and testing sets are separated in time, and driver behavior can change in the time frames I am experimenting on.

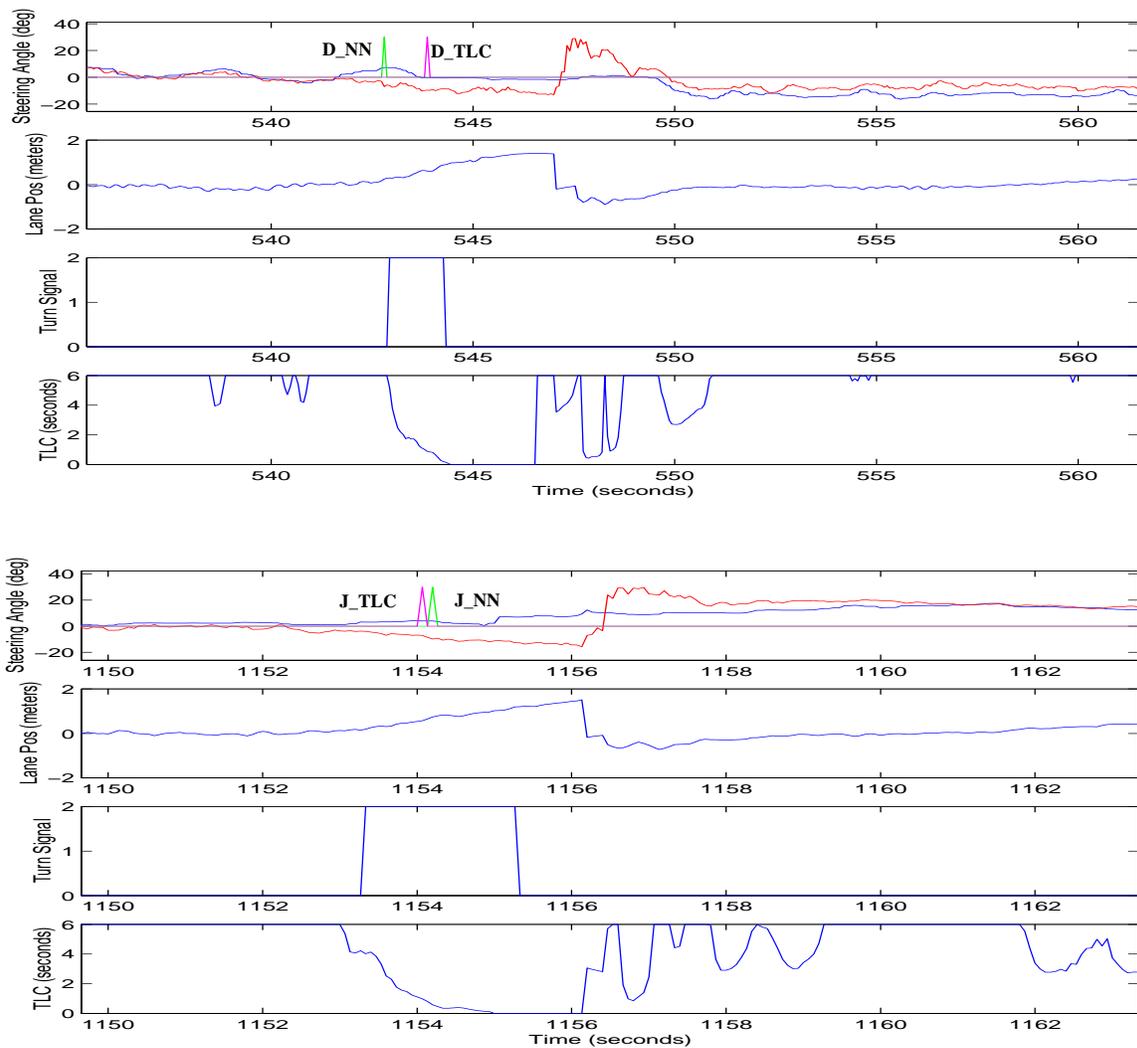


Figure 9: Close up of Lane Changes D and J

Overall, I feel these results are encouraging. I have shown that using a fairly simple learning scheme to predict steering wheel position realizes a fairly large gain in warning time, with a small increase in false alarms over TLC, due to increased sensitivity to perceptual errors. In fact, I could argue that false alarm performance also improves, as thresholding TLC high enough to achieve the same warning time as the NN increases false alarms dramatically.

Normalized Inputs

The most obvious differences between the drivers is mean lane position. From the above results, it looks as if the NN is only learning mean lane position and variance. To test this, I normalized the inputs, so that the lane position, yaw, and curvature each had a mean of 0 and a standard deviation of 1. This removes the lane position bias that existed in the previous experiments. Rather than using hand-wheel position as output, I used vehicle curvature, which is computed using vehicle yaw rate and velocity. The reason for doing this was to try and account for steering differences in curves due to dif-

Table 11: NN Classification Results (RMS Curvature) of Mini-Van Drivers

	Driver 2	Driver 3	Driver 4	Driver 5	Driver 6
Driver 2	0.3344	0.3153	0.3746	0.2964	0.3329
Driver 3	0.3578	0.2573	0.2735	0.3087	0.2840
Driver 4	0.3111	0.2180	0.2483	0.2430	0.1997
Driver 5	0.3667	0.2297	0.2726	0.2508	0.1691
Driver 6	0.3392	0.2974	0.3039	0.3147	0.2528

Note: All entries should be multiplied by 1.0e-3

ferences in superelevation of the road. Superelevation is the banking of the road used in curves to help maintain stability. However, due to lack of standards, two curves with the same curvature may have different amounts of superelevation. These two otherwise identical curves would therefore require the driver to maintain different steering wheel positions for any given velocity. Vehicle curvature (in a global coordinate frame) is not affected by this.

The classification results are shown in Table 11. These results are similar to those in Table 8, with the exception of Driver 2, who is now mis-classified as driver 5 instead of driver 3. These results for the test sets show that the differences diminish, but do not disappear when mean lane position is no longer a factor. This indicates that the network is learning mean lane position, along with something else, which I have not yet identified. I will have to further investigate what the network is learning, using sensitivity analysis or weight analysis.

4.4.4. Discussion

The above sections have demonstrated the following: 1) driver differences exist, and they can be captured using a connectionist approach. 2) These differences seem to be more complicated than mean lane position, although it is yet unclear what they are. 3) Using a neural net trained on a driver as a lane departure warning system outperforms TLC both through an increase in warning time, and decrease in the number of false alarms for equivalent warning time. Item 3 is the most important, as far as having an impact on my work. The experiments in driver classification were done only to investigate whether or not a canonical driver model, immune to changes in driver behavior over time, could be learned. The driver classification results show two things: 1) It may be advantageous to classify drivers into categories, rather than learn individual models, and 2) vehicle state may not be sufficient to learn an individual driver model -- adding some function of driver control output may improve classification results. Regarding future research, item 1 is discussed further in Section 7.4. Item 2 involves driver classification, and I will investigate this further in terms of improving the driver model to enhance warning system performance -- not in terms of improving classification performance.

Table 12: Braking Onset Time and Velocity Change

Driver 1	Braking Time (s)	Velocity (m.p.h.)	Driver 2	Braking Time (s)	Velocity (m.p.h.)	Driver 3	Braking Time (s)	Velocity (m.p.h.)
Run 1	-0.5	-4/50	Run 1	0.0	-7/48	Run 1	2.5	-10/50
Run 2	-1.5	-3/52.5	Run 2	0.75	7/48	Run 2	3.25s	-5.5/50

4.5. Curve Negotiation Differences

Until now, my work has been on lateral control differences between drivers and how they impact lane departure warning performance. To demonstrate that there are differences in curve negotiation behavior, I computed the braking onset time of 6 runs. The 6 runs consist of 3 drivers, each making two runs over the same stretch of rural road. This data was collected in early 1997 on rural roads, and is not part of the initial CMU data study. I am using it because it has sharper curves than the data which I am collecting. Figure 10 shows a plot for each run. The top of each plot is curvature, and the bottom is vehicle velocity. The beginning of the curve is marked with an 'x', and the beginning of braking is marked with an 'o'. The curve onset was initially selected by hand, by looking for the point where road curvature begins to shift. These estimates were then refined using on-board differential GPS to line up the points within +/- 3m along the direction of travel, which is within the resolution of the GPS. Because this data does not contain the brake pedal position, I used a decrease in velocity to mark the braking onset. Table 12 shows braking onset time and change in velocity for each run. The braking time column is the time at which deceleration occurs, relative to the curve start position. A negative time indicates that deceleration did not begin until the driver was within the curve. The velocity is the change in velocity (in m.p.h.) due to the deceleration, along with final velocity after the deceleration. Note that in most cases, the deceleration continues while in the curve, and even after exiting it.

The data illustrates 3 different behaviors. All drivers negotiated the curve at a velocity of about 48-50mph. However, driver 1 tended to brake within the curve. Driver 2 decelerates in run 1, but actually accelerates in run 2. Driver 3 was relatively consistent in his deceleration behavior, and tended to decelerate before the other two drivers. This begins to show that there are differences in curve negotiation that can be exploited. However, many additional experiments need to be performed to see how consistent these differences are, how they change over time, and how they vary with curvature.

5. Research Issues

A question that has to be asked is, if a lane departure warning system should be adaptive, what form should this adaptation take? As the previous work shows, the current primary form of adaptation is based on a time to lane crossing (TLC) metric, but this doesn't seem to be sufficiently powerful. Numerous issues arise in building a usable lane departure warning system. They include learning problems such as the scale of adaptability and recognition of valid lane changes. Practical issues such as usability also present a challenge. The following sections address some of the issues.

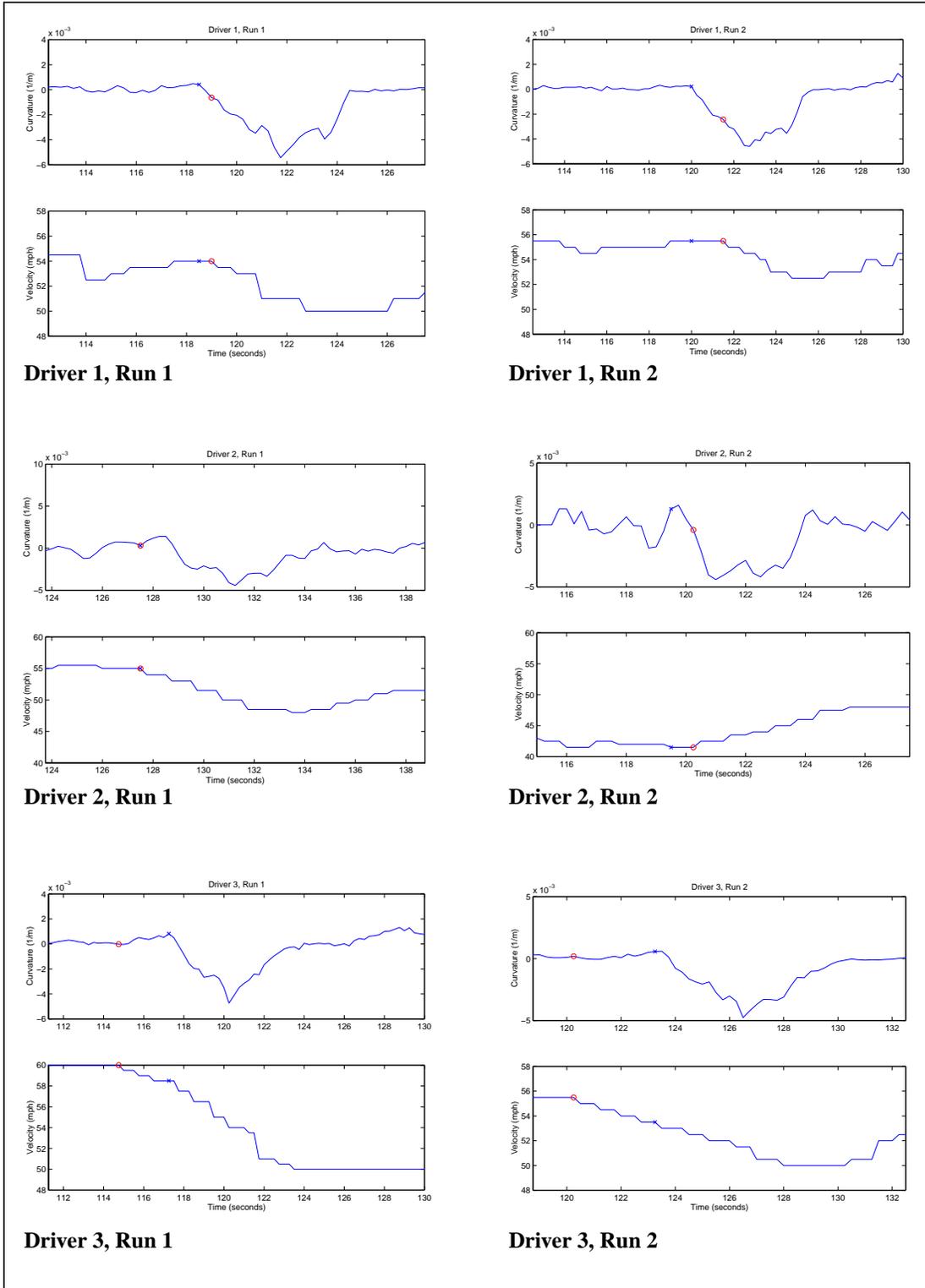


Figure 10: Plots of braking onset. Each row is a driver. The top of each plot is curvature, the bottom is velocity. The beginning of the curve is marked with an 'x', and the braking onset is marked with an 'o'.

5.1. Model Input

Given that I am interested in modeling driver behavior at a very low level, I have to be able to determine the appropriate inputs for the model. Most likely, mean lane position over the last n seconds would not work well, as it is a long term input. Steering wheel reversal point, which is related to TLC, may be one appropriate input. Some questions to answer would be, in what portions of the state space (where state space includes vehicle pose and control input) does the driver normally affect changes in the control input? How are these changes related to surrounding vehicles and roadway geometry? Simply determining a TLC threshold is not appropriate, however, as that assumes that people drive as hypothesized by Godthelp [17], and that has not been proven. More likely, there are other metrics which need to be included, such as steering wheel reversal rate. Knipling and Wierwille [22] define a set of metrics for use in drowsy driver detection. Some of those may be appropriate for this work.

5.2. Learning Method

The choice of learning method is obviously very important, and will affect the input choice. In my current work, I have taken a simple neural network architecture, given it raw inputs, and achieved a reasonable result. Taking this work further will require much more thought on the relative strengths and weaknesses of various ML algorithms. The method chosen will have to be trainable on-line, and be able to forget what it has learned at a rate which corresponds to the changes in driver behavior. Knowing when to stop learning is also important, and must be addressed when selecting the algorithm. What makes this an interesting learning problem is that the algorithm needs to be able to follow a moving target on-line, and know when it has come “close enough” to the target (in this case, driver behavior) to stop learning. It also needs to be able to detect that the target has moved, and re-learn. This re-learning, however, is domain dependent - re-learning should not occur if the driver is beginning to drive unsafely due to drowsiness or incapacitation. The method also needs to be able to generate a confidence measure based on how well it believes it has learned the current driver. I currently see two possible approaches. The first (which the NN approach demonstrates) is to learn a specific model for each driver. The second approach is to learn classes of drivers, and then cluster new drivers into these classes.

The first approach involves recording data on the driver during a training process, and then developing a model based on it. I see this as learning a policy, in which state predicts action. I will discuss extensions to my current work along these lines, along with the use of locally weighted learning [1] in Section 7.4.

While the first approach is conceptually simple, the second approach has advantages that may make it worthwhile. For instance, if a set of driver classes could be determined, it may be faster to classify the driver than to learn a new model for it. This could reduce the initial training time. Particularly, the idea has elegance if were to turn out that the driver classes had easily interpretable physical meaning (i.e., different levels of aggressiveness, tighter lateral control, etc.). A method for exploring these ideas will be discussed in Section 7.4.

The above paragraphs talk about developing a predictive model of the driver, and then comparing the prediction against the driver’s actual behavior. Another methodology, which I have not explored, is to classify driver actions and the resulting vehicle state into ROR situation or not-ROR situation categories. In other words, treat the problem as one of classification. This is interesting from a Machine Learning standpoint, as the classification has to occur without examples of one of the classes - i.e., ROR situations.

5.3. Learning History

As I showed in Section 4.3, not only are there differences in lane keeping performance between drivers, but within drivers as well. A driver will drive the first hour of an eight hour trip differently than the seventh, although both performances may be adequate for safety. The differences in performance related to time, however, have to be accounted for, to achieve the goal of minimizing false alarms. Given that driver behavior changes over time, I need to investigate how quickly it changes, and what forms (in terms of the yet to be decided model inputs) the changes take. While I want to be able to adapt to long term changes resulting from road conditions or minor driver fatigue, adapting to micro-sleep or drunkenness is definitely a bad idea. This makes the issue of selecting or modifying the proper learning algorithm vital to this thesis.

Another problem is that near instantaneous changes in driver behavior may occur in certain situations, particularly when roadway conditions change. For instance, a driver who has moved from a small two lane highway to a wide three lane highway with large shoulders may now drive less tightly, because of the additional space. This is a possible source of additional false alarms, at least until the system can adapt to the new driving style.

5.4. Surrounding Vehicles

A driver's behavior partially depends on the traffic he is driving in. The extent of this effect needs to be investigated. A simple example of this effect is the tendency of drivers to drift away from trucks or other large vehicles that are passing them. An alarm should not sound in this situation unless the drift is greater than expected for the driver. Another situation in which the environment should modulate the warning system is in collision avoidance maneuvers. A driver may swerve onto the shoulder to avoid hitting an obstacle, such as an animal or tire. Sukthankar [40] dealt with surrounding vehicles at the tactical level by using potential fields. This could be a promising approach for my work as well. The strength of the field would depend on the location of the vehicle and individual behavior, which is to be learned.

5.5. Curve Warning Adaptation

I do not believe that there are very many variables involved in longitudinal curve handling. One possible form of adaptation would be to learn when the driver normally applies braking, if he is going too fast for an upcoming curve, as this is an area where driver differences have been demonstrated. Curve handling is also very vehicle specific; trucks handle curves *very* differently than passenger cars, and safety thresholds will have to take this into account.

5.6. Predictability vs. Efficiency

When people use a system of any kind, they like to be able to build an internal model of how it works. They also like for their system to be efficient. In the context of a ROR warning system, efficiency can be thought of as being proportional to warning time, and inversely proportional to false alarm rate. It is important for drivers to know that when they do something, the response of the system is predictable and accurate. The current version of the RALPH warning system, for instance, makes an allowance for "curve-cutting," or the tendency to drift towards the inside of the lane while driving on a curved road.

The system allows the driver to drift a bit more over a lane boundary on a curve than it does on a straight road. Passengers have reported [34] surprise at the lack of an alarm when the vehicle drifts a bit into the shoulder during a curve, because in other situations (i.e., straight roads), a similar lane position has resulted in an alarm.

Alternatively, the more predictable a system is, the more likely it is to be functionally limited. A newer version of RALPH-WS has been developed which sounds an alarm strictly based on lane position. While it is very easy for the system to display the current boundaries, and for the user to know where he is in relation to them, the lack of adaptability can cause false alarms for certain drivers. The boundaries are user settable, which allows the driver to account for his own tendency to drive close to a lane marker. However, there is no easy mechanism to adapt to changing driver behavior.

The problem comes down to finding the right balance between predictability and efficiency. Neither extreme is optimal in the sense of user acceptance. I feel that for a lane departure warning system, the model created has to have an easily displayable indication of how the driver is performing in regards to that model. It should be intuitive, and should not change very quickly. This can help give the user a feeling that the system is operating reliably and predictably.

5.7. Sensitivity to Perceptual Errors

My current work uses more information than a TLC model does, by incorporating yaw and curvature. This additional information comes at the price of additional noise. The nuisance alarm which I show in Section 4.4 illustrates this. The alarm occurred because of errors in the curvature estimate, which affected the neural net much more than the TLC model. It is not appropriate to just classify these as perception errors and blame RALPH. One possibility is to disable the alarm system when the confidence drops too low. This is what is currently done in RALPH-WS. This is not an ideal solution, however. Currently, I do not have enough data to determine how much more sensitive my work will be to perception errors than RALPH-WS is. This will have to be addressed in the future work.

5.8. System Evaluation

For this work, system evaluation is a research issue. It is large enough that it is presented in the next section.

6. System Evaluation

This section will talk about how I will quantitatively and qualitatively evaluate my proposed system against the RALPH lane departure warning system. Positive results in both categories are important. Simply building a system which gives earlier warnings or reduces false alarms is not good enough if the system does not behave in a manner that the user feels is consistent. Similarly, a simple, predictable system which does not accurately trigger warnings and has many false alarms is not useful. The following two sections discuss how I plan to quantitatively and qualitatively analyze my work.

6.1. Quantitative Analysis

The most effective quantitative analysis for a system like this is to measure how many lives are saved. Unfortunately, that is not an easy statistic to determine. In similar situations, other researchers such as Tijerina [43] and Burgett [6] have taken a step back, and looked at the situation immediately before an event that they wish to measure. It is likely there are more lane departures than fatalities, as not every departure results in a death or accident. Burgett, in particular, has started with the simple assertion that given 100 accidents occur without a warning system, a warning system which is 50% effective will result in prevention of 50 accidents. From there, he presents a probabilistic framework to go from vehicle state to likelihood of crash. One possibility is to use vehicle state data from the Battelle Monte Carlo simulations described in Section 1.2. From this data, it may be possible to identify vehicle and road states which lead to RORs. A reduction in the occurrence of these states during a warning system deployment would indicate a quantitative improvement over current systems.

6.2. Qualitative Analysis

Qualitative analysis of the warning system will require regular drivers to use it. Ultimately, the success of this thesis will be defined by how useful the end result is to the average driver. This means that a user study will have to be performed. This study will have to overcome some of the difficulties in current data collection effort, such as unfamiliarity with the test vehicle, and nervousness due to experimenter presence.

There are a few approaches to solving this problem. The first is to loan out Navlab 8 to drivers who are going on extended trips, and allow them to try out both my new warning system, and the RALPH warning system. This is similar to what was done by Sayer et al. [39] to evaluate their ACC system. Alternatively, it may be possible to develop a version of the RALPH lane tracker and warning system that is easy to install in subject's vehicles. Work is already in progress on this. This has the advantage of allowing the drivers to drive vehicles that they are already familiar with. Another idea is to use the test track at the Transportation Research Center facility in Columbus, Ohio. This would allow for controlled experiments to intentionally distract the driver, such as in the Iowa simulator study [43], which could not be conducted on a public highway.

7. Proposed Work

I address many issues in Section 5. I do not believe that I will be able to solve each of them completely. In this section, I will detail what I will do to achieve the goals of my thesis, in roughly chronological order.

7.1. Data Collection

I will continue the data collection I have begun. I plan to continue until data on a total of 20 subjects has been collected. Up to now, I have been using members of the Robotics Institute as subjects. While this has the advantage of easy recruiting, it has the disadvantage that many members of RI are somewhat familiar with the work that I am doing. Due to insurance reasons, I can only use CMU members as subjects. Therefore, I plan to recruit outside of RI, through the use of posters and b-board messages.

The current route, which is from Pittsburgh to Grove City, has many curves. However, these curves are not that sharp. Generally, they are no sharper than 500m. While this is more than enough to capture curve cutting behavior, it is not optimal for studying longitudinal differences in curve negotiation. Therefore, I also plan to collect data on rural routes, which can have many sharp curves.

7.2. Navtech Map Evaluation

The use of GPS for curve warning systems requires an accurate map. I have access to digital maps produced by Navtech. These maps include high resolution latitude and longitude data on Pittsburgh, along with lower resolution data on surrounding areas. I will have to evaluate this map, to determine whether or not it is accurate enough for use. This will involve driving over different city and rural while collecting road curvature and GPS data. I will then register this data against the Navtech map, and compute road curvature accuracy.

If it turns out that the Navtech maps are not accurate enough, Pomerleau has developed a GPS mapping system using onboard differential GPS. Routes are mapped by repeatedly driving over them. This system works at a resolution of +/- 3-5m, and is therefore more than accurate enough for a curve warning system. This will allow me to continue my work, even though large scale applicability will be limited until more accurate maps are publicly available.

7.3. Curve Behavior Analysis and Model Development

Until now, I have done little analysis of the differences in curve speed keeping. While I have begun to show that there are differences in braking onset (See Section 4.5), I need to further quantify this. To this end, I will analyze the braking onset and average velocity through curves of varying sharpness. The goal of this is not to build tables of how different people react to curves. Rather, it is to gain insight to determine how best to model these differences. One possibility is to use a neural net which outputs a probability of the driver decelerating at any particular instant.

7.4. Lateral Control Model Selection

A large portion of the work I need to do will be in the area of model development. There are two issues here. The first is to determine what needs to be learned. The second is to determine how to learn it. Regarding the first issue, I am currently predicting driver output given vehicle state history. In other words, I learn how the driver normally reacts, and then flag any situation in which he does not act as predicted. However, this can result in false alarms in situations where the driver doesn't react as predicted, yet aren't really dangerous. It also opens up the issues of learning safe vs. unsafe behavior, which is very domain dependent. An alternative approach is to learn the expected driver trajectory over the next 1-2 seconds, and evaluate the final vehicle state to see if it is dangerous. This alleviates the problem of learning safe vs. unsafe behaviors, as learning unsafe behavior would cause the system to predict that the driver's unsafe control outputs would place the vehicle in a dangerous state. However, a disadvantage to this is that it may be difficult to learn to predict situations which are dangerous, as training data for this will be sparse.

Another aspect to the first issue is whether to learn individual models, or classifications of drivers. To this end, I will look at how to cluster similar drivers together, to allow for quicker initial training. This will involve developing distance, or similarity metrics between drivers. A very simple one is mean lane position. The issue may be complicated by the fact that two drivers who may be similar in one area

(mean lane position), may be quite different in other areas, such as reaction to surrounding vehicles. Properly weighing these differences is important. It will also be important to know when a driver cannot be categorized in the available clusters, requiring a new category be created. The benefit of this approach is two-fold: the first is that it forces a better understanding of the real differences between drivers, compared to individually building models in isolation. The second advantage is the decrease in initial training time, if a new driver could be quickly binned in the available categories.

To examine the second issue, which is to determine the proper learning framework, I will begin by extending the work that I have done using neural nets by testing it against other drivers, and for longer periods of time. This will include deeper analysis of the cause and nature of false alarms (both nuisance and safe). I do not plan to explicitly deal with unsafe false alarms, as I believe that the form of the warning (i.e., loud alarm vs. gentle steering wheel motion) is the prime factor in causing overreactions to alarms, and this is outside the scope of my thesis. As I mentioned in Section 5.1., the choice of inputs is very important. I am currently using lane position, yaw, and road curvature. Additional information could help force the net to learn other types of slightly longer term behavior, such as steering wheel reversal rate, nominal yaw rates and lateral accelerations. I will have to explore the data I have further to determine what other differences exist, and how to exploit them. I will also investigate other neural architectures. In particular, using architectures which explicitly encode temporal relationships such as BackProp Through Time [20] may improve performance and learning speed.

I will also look at how memory based learning (MBL) could be applied to the problem. One advantage of MBL, or locally weighted regression (LWR) [1], is that training is essentially cost-free (although this is at the expense of query time). Using MBL also simplifies the training set management that would be required for a neural net approach. Confidence measures can be derived for specific situations, allowing incremental learning to proceed more quickly. For instance, if the system has learned how the driver responds in left curves, the warning system could be active during left curves, even though it has not yet experienced other parts of the state space. From a usability standpoint however, activating and deactivating the system in different situations may be confusing for the driver.

However, one feature of MBL which is an advantage in many domains, is actually a disadvantage for my application -- it is that MBL does not forget old experiences as new ones accrue. In a domain where behavior is non-stationary, this is a serious issue. Applying forgetting to MBL in a principled manner is still an open area of research. A contribution in this area would be applicable to domains beyond driving where the proper response to a situation changes over time.

7.5. Evaluation On CMRI Truck Driver Data

I would like for my system to be useful to truck drivers as well as passenger car drivers, because truck drivers are particularly prone to ROR situations due to the long hours they spend driving. This presents a challenge, as truck drivers are much more variable in their behavior than passenger car drivers. I will have to expand upon my work using the virtual lane boundaries to account for the tendency of truck drivers to drive outside their lane. Finding the minimal amount to increase the lane, while still maintaining warning accuracy will be an interesting problem. Final evaluation on truck drivers will most likely not be possible. However, CMRI has recently acquired a four degree-of-freedom truck simulator which may be useful for both additional data collection, and system evaluation.

7.6. Accounting for Surrounding Vehicles

Rather than adding inputs to the driver model to represent surrounding vehicles, I plan to model them using a potential field approach. I feel that this is the correct method because the presence of other vehicles induces predictable behavior in drivers. If a driver is in the left lane, and there is a large vehicle to his right, he will most likely allow himself to drift to the left somewhat (although the initial passing of a large vehicle may actually suck the driver towards it). The adaptation required here is to learn how much of a drift is normal. While it is possible to learn the reactions of the driver to surrounding vehicles using the same method with which I will learn his control behavior, doing so increases the dimensionality of the state space, which can have a detrimental effect on training time and accuracy. Explicitly modeling the effects of surrounding vehicles on the driver allows me to get around this.

7.7. Final User Study

The final thing I need to do is evaluate the complete system. This will involve outfitting subjects vehicles with a version of RALPH-WS and my system, and record system performance while the subject drives on a long trip. Depending on the number of subjects available, I have two options. The first (preferable) is to place each subject in one of three groups: A control group with the warning system active, but no alarm, and two test groups -- one using the RALPH warning system, the other using mine. A fairly large number of subjects would be required to make this feasible (approximately 30). Given that I will most likely have fewer subjects, I could have RALPH-WS active for a certain number of days, and my system active for some days, and then ask the user to evaluate both systems. Preparing my system for the user study will require thought about how to interact with the user using the RALPH-WS user interface, along with the predictability of the system. The model I will use will be more complicated in some respects than current systems. However, most drivers would want some indication of how they are performing relative to the model, and this has to be easy for the average user to understand. Furthermore, system sensitivity should be tunable by the user. When I develop the model, I will have to keep these concerns in mind. Even though the main contributions of this work will be technical, the final evaluation will be subjective. Balancing both considerations is a difficult problem.

8. Expected Contributions

The primary contribution of this thesis is an advancement in the state of the art of lane departure and curve negotiation warning systems. Secondary advancements in the domains of driver modeling and continually adaptive learning will be required. I also see this work as providing a contribution to the area of driver assistance systems. The technology developed here will be applicable towards making lane keeping assistance systems which attempt to keep the driver in a groove that he is used to, as opposed to the center of the lane, as current systems do [45]. The adaptive curve warning component could also be used to enhance the driver's comfort while using ACC systems by braking in curves where the driver normally would. A further contribution is an evaluation of the system by untrained users. In the following list, I see the 1st item as the main contribution, and the rest as subsidiary contributions.

1. A working driver adaptive lane departure warning system which performs qualitatively and quantitatively better than RALPH-WS, on the road, using the perceptual abilities of the RALPH lane tracker.

2. A working driver adaptive GPS based curve negotiation warning system, tied into 1.
3. A contribution to continually adaptive learning.
4. Evaluation of the full system on a Navlab vehicle, with naive drivers..
5. Analysis of effects of vehicle environment on lane keeping behavior.
6. A greater understanding of low level driver behavior, which can extend to other control tasks where the operator behavior to a given situation changes over time.

9. Schedule

The following is a schedule for completion of this thesis. My expected defense date is August, 1999

Table 13: Schedule

	Spring '98			Summer '98			Fall '98			Winter '98			Spring '99			Summer '99		
Complete Data Collection	█	█	█															
Map Evaluation				█	█													
Lateral Model Development							█	█	█	█	█							
Environmental Effects										█	█	█	█	█				
Curve Negotiation Model													█	█	█			
Final User Evaluation and writing.																█	█	█

10. References

- [1] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 1997.
- [2] Parag Batavia. An annotated bibliography of human driver models. Available from <http://hoagie.ius.cs.cmu.edu:8000/parag/publications.html>.
- [3] John Baxter and John Y. Harrison. A nonlinear model describing driver behavior on straight roads. *Human Factors*, 21(1):87–97, 1979.
- [4] George A. Bekey, Gerald O. Burnham, and Jinbom Seo. Control theoretic models of human drivers in car following. *Human Factors*, 19(4):399–413, 1977.
- [5] Arthur T. Bergan, Robert J. Bushman, and Brian Taylor. Intelligent truck rollover advisory systems. In *Proceedings of the SPIE Intelligent Systems and Automated Manufacturing Conference*, Pittsburgh, Oct 1997.

- [6] August Burgett. Dot's approach to its safety evaluation. In *Proceedings of the Safety Evaluation of Intelligent Transportation Systems Workshop*, pages 61–77, Reston, VA, May 1995.
- [7] G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.
- [8] G.A. Carpenter, S. Grossberg, and J.H. Reynolds. Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4:565–688, 1991.
- [9] G.A. Carpenter, S. Grossberg, and D.B. Rosen. Fuzzyart: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.
- [10] Robert E. Chandler, Robert Herman, and Elliot W. Montroll. Traffic dynamics: Studies in car following. *Operations Research*, 6:165–184, 1958.
- [11] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman and Hall, 4th edition, 1989.
- [12] E.R.F.W. Crossman and H. Szostak. Man-machine models for car steering. In *Fourth Annual NASA-Univ. Conference on Manual Control*, pages 171–195, March 1968.
- [13] Robert D. Ervin et al. The crewman's associate for path control: An automated driving function. Technical Report UMTRI No. 95-35, University of Michigan, 1995.
- [14] P. Fancher, S. Bogard, M. Hagan, R. Ervin, and Z. Bareket. Data processing procedures for identifying driver/vehicle properties associated with the control of headway. In *Proceedings, IEEE Intelligent Transportation Systems Conference*, Boston, MA., Nov. 1997. IEEE.
- [15] J.P. Feraric and R. Onken. Daisy - a driver assisting system which adapts to the driver. In *IFAC Man-Machine Systems Symposium*, pages 523–528, 1995.
- [16] Kazutoshi Fukuda. A method for prevention of encroachment on opposite lane at a sharp curve. In *Proceedings of the 2nd World Congress on Intelligent Transportation Systems*, pages 1157–1162, Yokohama, Japan, Nov 1995.
- [17] Hans Godthelp, Paul Milgram, and Gerard J. Blaauw. The development of a time-related measure to describe driver strategy. *Human Factors*, 26(3):257–268, 1984.
- [18] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Pub. Co., Reading, MA., 1989.
- [19] J.A. Hadden, J.H. Everson, D.B. Pape, V.K. Narendran, and D.A. Pomerleau. Modeling and analysis of driver/vehicle dynamics with "run-off-road" crash avoidance systems. In *Proceedings of the 30th ISATA Conference*, 1997.
- [20] Mohamad H. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, Cambridge, MA, 1995.
- [21] Petros A. Ioannou. Autonomous intelligent cruise control. *IEEE Transactions on Vehicular Technology*, 42(4):657–672, November 1993.

- [22] Ronald R. Knippling and Walter W. Wierwille. Vehicle-based drowsy driver detection: Current status and future prospects. In *Proceedings of IVHS America*, pages 245–256, April 1994.
- [23] Chiu-Feng Lin. *Lane Sensing and Path Prediction for Preventing Vehicle Road-Departure Accidents*. Ph.D. thesis, University of Michigan, 1995.
- [24] Andrew Liu and Alex Pentland. Towards real-time recognition of driver intentions. In *Proceedings, IEEE Intelligent Transportation Systems Conference*, Boston, MA, November 1997. IEEE.
- [25] Chia-Shang Liu and Huei Peng. Road friction coefficient estimation for vehicle path prediction.
- [26] Duane T. McRuer, R. Wade Allen, David H. Weir, and Richard H. Klein. New results in driver steering control models. *Human Factors*, 19(4):381–397, 1977.
- [27] K. Naab and G. Reichart. Driver assistance systems for lateral and longitudinal vehicle guidance - heading control and adaptive cruise control. In *Proceedings of the International Symposium on Advanced Vehicle Control*, pages 449–454, 1994.
- [28] Michael C. Nechyba and Yangsheng Xu. On discontinuous human control strategies. In *Submitted, Proceedings IEEE Int. Conference on Robotics and Automation*. IEEE, 1998.
- [29] Michael C. Nechyba and Yangsheng Xu. Abstracting human control strategy into computational models: Learning architecture and input selection. In *Proceedings of the International Conference on Robotics and Automation*, 1997.
- [30] Michael C. Nechyba and Yangsheng Xu. Stochastic similarity for validating human control strategy models. In *Proceedings of the International Conference on Robotics and Automation*, 1997.
- [31] R. Onken and J.P. Feraric. Adaptation to the driver as part of a driver monitoring and warning system. In *The Second International Conference on Fatigue and Transportation: Engineering, Enforcement, and Education Solutions.*, February 1996.
- [32] Tom Pilutti and Galip Ulsoy. On-line identification of driver state for lane-keeping tasks. In *Proceedings of the American Control Conference*, pages 678–681, June 1995.
- [33] L.A. Pipes. An operational analysis of traffic dynamics. *Journal of Applied Physics*, 24:271–281, 1953.
- [34] Dean A. Pomerleau. Personal communication.
- [35] Dean A. Pomerleau. Ralph: Rapidly adaptive lateral position handler. In *Proceedings, IEEE Symposium on Intelligent Vehicles*, Detroit, MI., Sept. 1995. IEEE.
- [36] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1), 1986.
- [37] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. In *IEEE ASSP Magazine*, pages 4–16, Jan. 1986.

- [38] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362, Cambridge, MA, 1986. Bradford Books/MIT Press.
- [39] J.R. Sayer, M.L. Mefford, P.S. Fancher, R.E. Ervin, and S.E. Bogard. An experimental design for studying how driver characteristics influence headway control. In *Proceedings, IEEE Intelligent Transportation Systems Conference*, Boston, MA., Nov. 1997. IEEE.
- [40] Rahul Sukthankar. Situational awareness for tactical driving. Technical Report CMU-TR-RI-97-08, Carnegie Mellon University, January 1997.
- [41] Hiroshi Takahashi and Kouichi Kuroda. A study on automated shifting and shift timing using and driver's mental model. In *Proceedings, Intelligent Vehicles*, 1996.
- [42] Kazuya Tamura and Hiroshi Sekine. A study of self-reliant cornering speed control system. In *Proceedings of the 2nd World Congress on Intelligent Transportation Systems*, pages 1163–1168, Yokohama, Japan, Nov. 1995.
- [43] Louis Tijerina, James L. Jackson, Dean A. Pomerleau, Richard A. Romano, and Andrew D. Petersen. Driving simulator tests of lane departure collision avoidance systems. In *Proceedings of ITS America Sixth Annual Meeting*. ITS America, April 1996.
- [44] National Highway Transportation and Safety Administration. Fatality analysis reporting system. On the web at <http://www-fars.nhtsa.dot.gov/fars>.
- [45] R. Tribe, K. Prynne, and I. Westwood. Intelligent driver support. In *Proceedings of the 2nd World Congress on Intelligent Transportation Systems*, pages 1187–1192, Yokohama, Japan, Nov 1995.
- [46] Walter W. Wierwille, Gilbert A. Gagne, and James R. Knight. An experimental study of human operator models and closed-loop analysis methods for high-speed automobile driving. *IEEE Transactions on Human Factors in Electronics*, 8(3):187–201, 1967.
- [47] Liang Zhao. Personal communication.