MODELING NATURALISTIC DRIVER BEHAVIOR IN TRAFFIC USING MACHINE LEARNING

Linsen Chong

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Civil Engineering

Montasir Mahgoub Abbas, Chair
Naren Ramakrishnan
Raghu Pasupathy

July 26th, 2011
Blacksburg, VA

Keywords: Driver behavior, car-following, safety critical events, machine learning

MODELING NATURALISTIC DRIVER BEHAVIOR IN TRAFFIC USING MACHINE LEARNING

Linsen Chong

ABSTRACT

This research is focused on driver behavior in traffic, especially during car-following situations and safety critical events. Driving behavior is considered as a human decision process in this research which provides opportunities for an artificial driver agent simulator to learn according to naturalistic driving data. This thesis presents two mechine learning methodologies that can be applied to simulate driver naturalistic driving behavior including risk-taking behavior during an incident and lateral evasive behavior which have not yet been captured in existing literature. Two special machine learning approaches Backpropagation (BP) neural network and Neuro-Fuzzy Actor Critic Reinforcement Learning (NFACRL) are proposed to model driver behavior during car-following situation and safety critical events separately. In addition to that, as part of the research, state-of-the-art car-following models are also analyzed and compared to BP neural network approach. Also, driver heterogeneity analyzed by NFACRL method is discussed. Finally, it presents the findings and limitations drawn from each of the specific issues, along with recommendations for further research.

DEDICATIONS

To my parents.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Montasir Abbas, for offering me the great opportunity to work in his group. I am grateful for his valuable guidance and support in my researches. I admire his never ending enthusiastic in transportation researches that always inspires me to conquer challenges in my study at Virginia Tech. Discussions with him are always full of intelligence and fun.

Additionally, I would like to thank Ms. Alejandra Medina from Virginia Tech Transportation Institute, for her valuable guidance and continuous support in our research project during my graduate study. I would also extend my gratitude to my committee members, Dr. Naren Ramakrishnan and Dr. Raghu Pasupathy for their variable suggestions and feedbacks for my work.

I would like to thank my friends Zain Adam and Bryan Higgs for their generous help throughout the research process. I would also like to thank my family and friends who have encouraged me in my academic pursuits.

ATTRIBUTIONS

Chapter 2 and Chapter 3 of this thesis are the outcomes from the collaborate work by members of Virginia Tech Signal & Operation Research and Education Systems Lab under the supervision of my advisor Dr. Montasir M. Abbas and Ms. Alejandra Medina, who is a senior researcher at Virginia Tech Transportation Institute. Dr. Abbas provided a lot of help in my understanding and formulation of the proposed neural network model. Ms. Alejandra Medina and my college Bryan Higgs helped me in naturalistic data filtering and validation process.

Chapter 2 of this thesis was presented at the 90th Annual Meeting of the Transportation Research Board, January 2011, Washington, D.C. and was accepted for publication in journal *Transportation Research Record*.

Chapter 3 of this these was submitted for presentation at the 91th Annual Meeting of the Transportation Research Board, January 2012, Washington, D.C. and publication in *Transportation Research Record*.

TABLE OF CONTENTS

DEDICATIONS iii

ACKNOWLEDGEMENTS iv

ATTRIBUTIONS v

TABLE OF CONTENTS vi

LIST OF FIGURES ix

LIST OF TABLES x

1 INTRODUCTION 1

1.1 SYNTHESIS ON PAST RESEARCH EFFORTS 1

1.1.1 Car-Following Models 1

1.1.2 Model Calibration with Trajectory Data 2

1.1.3 Modeling Safety Critical Events in Traffic 2

1.1.4 Drawbacks of Model Calibration Methods 3

1.1.5 Conclusion 3

1.2 RESEARCH OBJECTIVE AND TASKS 4

1.3 THESIS CONTRIBUTION 4

1.4 THESIS ORGANIZATION 4

2 DRIVER CAR-FOLLOWING BEHAVIOR SIMULATION USING BACKPROPAGATION NEURAL NETWORK 5

ABSTRACT 6

2.1 INTRODUCTION 7

2.1.1 Car-following Models: Brief Review 7

2.1.2 Driver Behavior Simulation Using Car-following Models 8

2.1.3 Car-following Models: Shortcomings 8

2.1.4 Agent-based Neural Network Driver Behavior Simulation 9

2.2 GHR CAR-FOLLOWING MODEL AND CALIBRATION 10

2.3 BP NEURAL AGENT DRIVER BEHAVIOR SIMULATION 11

2.3.1 Neural Agent Models 11

2.3.2 Backpropagation ANN Architecture 11

2.4 NATURALISTIC DRIVING DATA 13

2.4.1 Naturalistic Database Description 13

2.4.2 Car-following Situations Extraction and Selection 14

2.5 EXPERIMENT AND RESULTS 15

2.6 CONCLUSIONS AND FUTURE RESEARCH 19

3 AN AGENT-BASED REINFORCEMENT LEARNING MODEL FOR SIMULATING DRIVER HETEROGENEOUS BEHAVIOR DURING SAFETY CRITICAL EVENTS 20

ABSTRACT 21

3.1 INTRODUCTION 22

3.1.1 Background Information 22

3.1.2 Traffic States and Actions during Safety Critical Events 22

3.1.3 Agent-based Modeling on Driver Behavior 23

3.1.4 Reinforcement Learning: Brief Introduction 23

3.1.5 Reinforcement Learning Applied in Transportation 24

3.1.6 Drawbacks of Existing Reinforcement Learning Algorithms 24

3.1.7 The Proposed Reinforcement Learning Methodology 25

3.2 NFACRL METHOD 25
3.2.1 NFACRL Structure 25
3.2.2 Layer Design 27
3.2.3 NFACRL Output Actions 30
3.2.4 Weights Update: Reinforcement Learning Algorithm 30
3.3 NATURALISTIC DATA AND TRAINING PROCEDURE 32
3.3.1 Database Description 32
3.3.2 Safety Critical Events Extraction 33
3.3.3 State and Action Variables Selection 34
3.3.4 Training Process 34
3.4 AGENT TRAINING RESULTS 35
3.4.1 Preliminary Training Results 35
3.4.2 Cross Validation Results 38
3.4.3 The "Mega Agent" Idea 42
3.5 CONCLUSION AND FUTURE RESEARCH 46
4 SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS 48
4.1 SUMMARY 48
4.2 FINDINGS 49
4.3 LIMITATIONS 50
4.4 RECOMMANDATIONS FOR FUTURE RESEARCH 51
REFERENCES 52

LIST OF FIGURES

Figure 2-1 Neural Agent Structure 12
Figure 2-2 Speeds of Neural Agent and GHR Model (1Hz) 15
Figure 2-3 Speeds of Neural Agent and GHR (10Hz) 16
Figure 2-4 Neural Agent Model Training, Validation and Test 17
Figure 2-5 Neural Agent Acceleration Action in Validation 18
Figure 2-6 Validation Example for Neural Agent and GHR Model 18
Figure 3-1 NFACRL Structure 26
Figure 3-2 Acceleration of Agent A, Event A1 36
Figure 3-3 Yaw angle of Agent A, Event A1 36
Figure 3-4 Acceleration of Agent A, Event A2 37
Figure 3-5 Yaw angle of Agent A, Event A2 37
Figure 3-6 Acceleration of Agent B, Event B1 39
Figure 3-7 Yaw Angle of Agent B, Event B1 39
Figure 3-8 Acceleration of Agent B, Event A1 40
Figure 3-9 Yaw Angle of Agent B, Event A1 41
Figure 3-10 Acceleration of Agent A, Event B1 41
Figure 3-11 Acceleration of Agent A, Event B1 42
Figure 3-12 Acceleration of Mega Agent, Event A1 43
Figure 3-13 Yaw Angle of Mega Agent, Event A1 44
Figure 3-14 Acceleration of Mega Agent, Event B1 44
Figure 3-15 Yaw Angle of Mega Agent, Event B1 45

LIST OF TABLES

Table 3-1 R squared values for cross validation 42

Table 3-2 R squared values of the Mega Agent 45

1 INTRODUCTION

Vehicle actions have great impacts on traffic flow operations. Traffic operations may become deteriorated due to one or more drivers' behavior. For example, congestions may result from vehicle merging behavior at intersections and ramps; crashes may result from the risk-taking action made by an aggressive driver. When congestions and crashes happen, the whole traffic may be blocked and it would make the whole transportation system idle for a while during recovery.

Vehicle actions are closely related to driver behavior in traffic, especially when vehicles are interacting with each other. The task of driving could be different when surrounding traffic is different. In a near congested regime when a driver has little freedom to drive at free flow speed, the vehicle is restrained by the existence of the preceding vehicle and the driver will follow the leading vehicle most probably. When a driver observes a sudden break from the vehicle in front, the driving task should be avoiding the incoming conflict and the driver has the intention to hit the break or swerve to execute a maneuver.

In this research, we are interested in two types of driving behavior: car-following and evasive behavior. Car-following behavior happens when a driver is interacting with a leading vehicle as long as the relative distance is not so close and no emergent conflict can occur. Car-following behavior only considers executing longitudinal vehicle action acceleration. Evasive behavior happens mostly during safety critical events when two vehicles in the same lane are becoming too close and an evasive action from the following vehicle should be taken in order to avoid a rear-end collision. During a safety critical event, both longitudinal and lateral actions are taken into consideration.

As part of this research, we also focus on the heterogeneity of driver behavior. While driver behavior may be similar in free driving or car-following condition, it is expected to vary a lot during safety critical events since driver demography, vehicle mechanics, event causalities and traffic conditions could be totally different.

1.1 SYNTHESIS ON PAST RESEARCH EFFORTS

1.1.1 Car-Following Models

In the last fifty years, a considerable amount of research has focused on modeling longitudinal driving behavior, producing a large number of car-following models [1]. Car-following models describe the process by which drivers follow each other in the traffic stream and provide interaction between the leading vehicle and the following vehicle. The underlying factors that would eventually lead to the construction of these models was given by most literature include the relative velocity, relative spacing in which drivers of the following vehicle decide to deceleration or accelerate according to the perceived thresholds.

Most of the car-following models belong to safety distance models assuming that the following vehicle is able to stop before becoming too close to the leading vehicle. Most recent car-following models can guarantee that the minimum distance of two vehicles during the car-following task should always be greater than a safety distance to prevent safety critical events happen [2-7]. Other models, on the other hand, extend safety distance idea by incorporating human's recognition on different traffic state patterns. By assuming the driver will perform an action when a threshold (a function of speed difference and distance) is reached, the driver is no longer doing "following" behavior all the time, instead, cruise control and longitudinal sharp break behavior are included to model free driving and emergency situations respectively [3, 6, 7].

1.1.2 Model Calibration with Trajectory Data

Rakha et al [8] presented a macroscopic calibration methodology for Gipps model. Microscopic car-following model parameters were derived from the related parameters in macroscopic traffic stream models. Key macroscopic parameters (free-flow speed, speed at capacity, capacity and jam density) were calibrated using loop detector data. Menneni et al [9] represented another calibration methodology based on the integrated use of microscopic and macroscopic data. Next Generation Simulation (NGSIM) data collection effort was utilized to define ranges of calibration parameters for Wiedemann car-following model. Kesting and Treiber [5] used publicly available trajectory data to study car-following behavior on individual drivers. An instrumented car with a radar sensor in front provided relative speed and distance data required for calibration. Two models: Intelligent Driver Model (IDM) and Velocity Difference Model (VDM) were calibrated using a genetic algorithm to minimize deviations between simulated and observed trajectories. Ossen and Hoogendoorn [10] pointed out that the development of accurate and robust models reply on appropriate microscopic data. The car-following behavior between individual drivers was analyzed by making use of vehicle trajectory data extracted from high resolution digital images collected at a high frequency from a helicopter. A well-known GHR car-following model was calibrated to represent individual differences.

1.1.3 Modeling Safety Critical Events in Traffic

Hamdar et al [11] adjusted the specifications of microscopic traffic models that could capture congestion dynamics and model accident-prone behaviors on a highway section. Alternative specifications for car-following and lane-changing models were developed under different degrees of relaxation of the safety constraints and implemented in a microscopic simulation framework. The results suggested that these specifications offer an improved basis for situations that do not require an accident-free environment. Xin et al [12] proposed a model to capture unsafe driver behavior. In the proposed methodology, unsafe behavior is triggered when a driver is in a subconscious driving state and a modified Gipps car-following model was applied.

1.1.4 Drawbacks of Model Calibration Methods

Accordingly to the calibration efforts from previous studies, several drawbacks of model calibration methodology were pointed out. One significant problem results from the error of measurement during the calibration process. Ossen and Hoogendoorn [13] discussed about three findings of obtained calibration results using GHR car-following models: (1) measurement errors can yield a considerable bias; (2) parameters minimizing the objective function do not necessarily capture the car-following dynamics best and (3) measurement errors can substantially reduce the sensitivity of the objective function and reduce the reliability of the results. Similarly, Brockfeld et al [14] calibrated 10 car following models using the same data with the result showing that all models shared the same problem with particular set of data. Also, this finding pointed out that no model appears to be significantly better than any other model. Even though some of the models have more calibration parameters, they did not provide better results in general.

1.1.5 Conclusion

Car-following behavior extracted by car-following models is based on formulated equations that are used to map traffic environment drivers observe to longitudinal accelerations. However, since different car-following models were derived from different data resources, it is expected that no model is able to match all the data resources. In fact, each model is expected to work well with the traffic environment which is similar from its own resource. Therefore, when we have no prior knowledge about the data we want to use, it is difficult to choose the “best” model. In addition, as mentioned before, parameter measurement errors can deteriorate calibration efforts. Consequently, it may be risky to use model calibration method to represent individual car-following behavior.

In terms of research efforts on safety critical events, preliminary researches basically extend the capability of several existing car-following models so that they could be adapted to capture behavior during the emergency regime. However, systematic methodologies have not yet been fully developed maybe due to the lack of events data especially for individual drivers. Additionally, most existing models do not capture driver lateral behavior.

1.2 RESEARCH OBJECTIVE AND TASKS

The goal of this thesis is to encapsulate driver naturalistic behavior during car-following situations and safety critical events by using an artificial intelligence machine learning approach. The naturalistic data corresponding to particular drivers is used to train individual artificial agents so that any given agent should behave similar to the “guiding” driver.

The major tasks of this thesis are:

- To design an Artificial Neural Network (ANN) model to train agents which should be able to simulate individual driver longitudinal actions in car-following situations
- To design an Reinforcement Learning (RL) model to train agents which should be able to simulate driver longitudinal and lateral actions during safety critical events

1.3 THESIS CONTRIBUTION

This thesis uses two innovative approaches to capture naturalistic driver actions during two situations in traffic to avoid above mentioned calibration problems. Driver’s interactive behavior in traffic is considered to be a complicated human decision process which could be generalized as a traffic state-action mapping problem in this research. The thesis aims to enhance understanding on driver naturalistic behavior. Technically, this thesis is an attempt to test the capability of artificial intelligence machine learning approaches when applied to microscopic traffic flow theory.

1.4 THESIS ORGANIZATION

This thesis consists of five chapters. Chapter 1 presents an introduction, a synthesis of past research efforts, research objectives and contribution of this thesis. Chapter 2 presents a Backpropagation (BP) Artificial Neural Network (ANN) to model driver behavior during car-following situations. It explains BP ANN structure, agent training process and its comparison to the well-known GHR car-following model. Chapter 3 presents a fuzzy logic reinforcement learning model for simulating driver evasive behavior during safety critical events. Heterogeneities behavior among drivers during different events are also analyzed and discussed. Finally, Chapter 4 presents thesis conclusions including findings and limitations of our proposed methodologies and provides suggestions for future research.

2 DRIVER CAR-FOLLOWING BEHAVIOR SIMULATION USING BACKPROPAGATION NEURAL NETWORK

Linsen Chong
Graduate Research Assistant
Charles Via Department of Civil and Environmental Engineering
301-D, Patton Hall
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061
E-mail: linsenc@vt.edu

Montasir M. Abbas, Ph.D., P.E. (Corresponding Author)
Assistant Professor
Charles Via Department of Civil and Environmental Engineering
301-A, Patton Hall
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061
Phone: (540) 231-9002
E-mail: abbas@vt.edu

Alejandra Medina
Senior Research Associate
Virginia Tech Transportation Institute
3500 Transportation Research Plaza (0536)
Blacksburg, VA 24061
Phone: (540) 231-1508
E-mail: ale@vtti.vt.edu

ABSTRACT

Two microscopic driver behavior simulation methodologies, Gazis-Herman-Rothery (GHR) car-following model and a proposed neural network model, are compared in this paper. A backpropagation (BP) neural network is trained using car-following episodes from the data of one driver in the Naturalistic driving database to establish action rules for an agent driver simulator to follow under perceived traffic conditions during car-following situations. The GHR car-following model is also calibrated using the same set of data through the use of a Genetic Algorithm. The car-following episodes are carefully extracted and selected for model calibration/training. Performances of the two models are compared with the result showing that under 10Hz data resolution, agent simulator outperforms GHR model significantly and is able to capture the individual driver behavior with a 95% similarity in driving trajectory.

KEY WORDS: driver behavior, car-following model, neural network

2.1 INTRODUCTION

Driver action simulation is an important part in modeling microscopic driver behavior. A driver action indicates driver behavior in terms of causalities and responses in traffic flow. Microscopic car-followings models provide us many powerful simulation tools to study individual driver behavior, interactions between leading and following vehicles, and cumulative macroscopic traffic phenomena. The performance of car-following models relies on the parameters of individual drivers that can represent unique driving behavior. Parameter calibration becomes a necessary process before car-following models can be applied to a simulation environment.

Driver actions in car-following models are determined by pre-defined driving rules. These rules are mostly interpreted by relating traffic state a driver observes to the response or action that the driver takes. Different car-following models consider different criteria as causalities that stimulate a driver's reactions. However, in reality, these rules specified by car-following models may not capture naturalistic driving behavior due to the complexity and instability of human decision making process.

In our approach, instead of using pre-specified driving rules from car-following models, we use a reactive structure Artificial Neural Network (ANN) to relate traffic states to driver actions. An ANN does not require a function to associate traffic state with actions, but rather requires sufficient real traffic and action data to capture the underlying relationship between states and actions. Therefore, ANN models estimate actions based on the real state-action mapping of naturalistic behavior.

We use the Naturalistic driving database from the Naturalistic Truck Driving Study (NTDS) [15] performed by the Virginia Tech Transportation Institute and Blance et al (in press, not available [16]) with the objective of finding the real causalities and responses of truck drivers in car-following situations. In this paper, we first describe the process of calibrating the well-known GHR car-following model for an individual driver. Then we construct a backpropagation (BP) neural network to train agents that represent the same driver.

2.1.1 Car-following Models: Brief Review

Car-following behavior, which describe the process by which drivers follow each other in the traffic stream have been studied for almost fifty years. A large amount of car-following models has been developed to represent longitudinal driver behavior, including safety distance models, collision avoidance models, psycho-physical action point models, and fuzzy logic-based models [1]. Most well-known car following models have been embedded in micro-simulation software, such as Pipes model in CORSIM [17], Gipps model in AIMSUN[2], Fritzsche model in Paramics[6], and Wiedemann model in VISSIM[7]. Car-following models assume that the following vehicle reacts according to the observed stimulus from its leader following predefined rules. Car-following models require specific defined functions to relate stimuli a following vehicle observes to the reaction it takes. For example, Gazis-Herman-Rothery (GHR) model uses speed difference and space headway as stimuli to determine driver acceleration action of the following vehicle. Wiedemann and model divides state space into several driving regimes by predefined thresholds where following vehicle reacts differently in those regimes [3, 7]. Gipps model uses vehicle dynamic as constrains and involves estimated leading vehicle deceleration to derive following vehicle acceleration [2].

2.1.2 Driver Behavior Simulation Using Car-following Models

Car-following model calibration is an important process to represent driver behavior and simulate vehicle trajectory. The calibration parameters are considered to be driver dependent and remain to be constant during all car following situations [7]. Therefore, the calibration process becomes an optimization problem with the objective of searching for the best car-following model parameters to minimize the deviation of estimated car-following trajectory from naturalistic vehicle trajectory. Accordingly, optimal parameter set is considered to the best representation of the real driver behavior.

2.1.3 Car-following Models: Shortcomings

Both macroscopic and microscopic data have been used in the calibration of car-following models in many literatures. Macroscopic traffic data provided by loop detectors has been used to compute macroscopic parameters such as free flow speed, jam density and speed at capacity. Car-following model parameters can be represented using steady state macroscopic traffic flow parameters [8]. However, loop detector data includes all the vehicle data that have passed by and results in calibration parameters representing an average behavior of all the drivers instead of individual drivers.

Microscopic calibration uses vehicle trajectory data (speed, distance, etc) to calibrate individual driver unique parameters. Microscopic data collection requires specialized sensor equipped instrumented vehicle or video observation above road [10], which means that collection efforts are expensive and time-demanding. However, technological advances in microscopic data collection methods in recent years have propelled an increase in using microscopic trajectory data in the calibration of car-following models.

Although car-following models can describe and explain driver behavior to some extent, logical and statistical errors reduce the reliability of these models. From real world experience, the driver takes actions based on the interactions of all observed stimuli in the driver environment. Unfortunately, most of the car-following models consider only some of the stimuli as causality of driver's response which may result in a biased conclusion. In addition, the parameters may not be constant for different stimuli. Studies have shown that reaction time in acceleration is longer than in deceleration [18] and the sensitivity parameter of speed difference in deceleration is 10% greater in deceleration than in acceleration [19].

Statistically, the chosen calibration methodology causes biases. As Ossen and Hoogedoorn pointed out [13], measurement errors can yield a considerable bias in the estimation results. Parameters minimizing the objective function do not necessarily capture the following dynamics best and measurement errors substantially reduce the sensitivity of the objective function and consequently reduce the reliability of the results. Brockfeld et al [14] tested validity of 10 car following models, showing that no model appears to be significantly better than any other model. All models shared the same problem with particular sets of data. The paper concluded that although some of the models have more calibration parameters, they did not provide better results in general.

2.1.4 Agent-based Neural Network Driver Behavior Simulation

To avoid these above mentioned problems, if an agent simulator can learn the driver behavior from the previous actions he/she took in his/her observed traffic environment, it may adopt driver behavioral rule and may act intelligently like its clone. The simulator will observe all the traffic state stimuli, make a judgment based on critical stimulus/stimuli and try to replicate driver reaction.

Agent-based modeling (ABM) is a relatively new paradigm for exploring complex system behavior [20].Within the transportation domain, ABM is particularly appropriate for modeling human decision making and action systems. Studies on ABM driver behavior include driver response to incidents, interaction between cars and trucks, driver behavior approaching a work zone, etc [21, 22]. Bonabeau [23] suggests ABM is best applied to simulations when: agents are heterogeneous, interactions of agents are complex, nonlinear or discontinuous, and agents exhibit complex behavior including learning and adaptation.

The ABM system in this paper uses an agent unit to simulate a driver. Car-following conditions are used as traffic state scenarios experienced by the agent. An Artificial Neural Network (ANN) is used to supervise the agent during training process. The agent receives supervised training based on the driving characteristics and state-action pairs extracted from the naturalistic trajectory data. After the training process, the neural agent should be able to replicate the actions when at the similar traffic states.

Neural networks have been applied in car-following model design in several previous studies [24, 25]. Jia et al [24] applied a neural car-following model using the test data collected by a technique known as “Five-Wheel System”. Speed of the following vehicle, relative speed, relative distance and desired speed were selected as four input variables and acceleration of the following vehicle was used as the output variable. Three types of drivers (risk drivers, ordinary drivers and conservative drivers) were classified by desired speed as the fourth input to represent driver heterogeneity. However, classification of different drivers based on desired speed alone causes the loss if other driver dependent characteristics are dominant. Furthermore, their study did not provide validation on the performance of the neural network.

Panwai and Dia [25] developed a neural car-following model and implemented in the AIMSUN simulation software. The data used in the modeling included speeds and distance headway of both leader and follower based on the premise that driver would select individual speeds and maintain a desired headway. Driving conditions were divided into five regimes by distance headway and speed difference. The developed model was interfaced using AIMSUN in validation and compared to the default model Gipps in AIMSUN with results showed that the neural network had an improvement of 20% compared to Gipps model embedded in AIMSUN. Although this novel model had a good approximation on field data, the five regimes seemed to be arbitrary defined and this neural model output is speed which is a state variable, not an action decision.

In this paper, we propose a widely used type of ANN: BP neural network and test its performance on degree of accuracy of its approximation on naturalistic driver action acceleration /deceleration. We train a neural agent with the purpose of extracting the driver car-following driving rules from a real driver.

2.2 GHR CAR-FOLLOWING MODEL AND CALIBRATION

GHR model is a general form of earlier car following models. In the perspective of GHR model, diver action acceleration is considered to be a function of speed, speed difference and spacing. GHR model's formulation is

$$a_n(t) = cv_n^m(t)\frac{\Delta v(t-T)}{\Delta x^l(t-T)}, \qquad (2\text{-}1)$$

where a_n is the acceleration of vehicle n at time t, Δx and Δv are the relative spacing and speeds between the leader vehicle $n-1$ and the following vehicle n, T is the driver perception reaction time. $c, m, l,$ and T are the four calibration parameters.

The performance of GHR model's simulation on driver's action depends on calibration parameters. As mentioned previously, a large number of calibration parameters do not necessarily have a better approximation on vehicle trajectory. Therefore, the four-parameter GHR model seems to be an acceptable car-following model in this study.

We use minimum sum squared error between estimated speed and naturalistic speed as the objective in optimization. The optimization tool adjusts the car-following model parameters to make the best match between GHR estimated speed v_{est} and the measured speed from field measurement v_f.

$$SSE = \sum_{t=0}^{N}(v_{est} - v_f)^2 \qquad (2\text{-}2)$$

We used genetic algorithms as the optimization tool during searching for the optimal in parameter space. In order to restrict the parameter space for the optimization to be reasonable, we applied the following constraints for minimum and maximum values from past empirical studies and calibration results [1]. m is restricted to the range [0, 1.5], l is restricted to [0, 2.5], T is restricted to [0, 2].

2.3 BP NEURAL AGENT DRIVER BEHAVIOR SIMULATION

2.3.1 Neural Agent Models

In our proposed neural agent model, the driver agent observes the traffic state (its environment) and reacts, which is very similar to the state perception and action mapping of a real-world driver. Our approach is based on a reactive structure using an artificial neural network (ANN). Unlike car-following models, ANN does not need a predefined function or an equation to associate traffic state with actions. Instead, ANN is able to extract state-action mapping rules from naturalistic driving states and actions. In fact, an ANN receives training from the state-action pairs a driver has already experienced from naturalistic data sets and formulates state-action mapping rules for agents to follow. With a set of input and output training data, an ANN is capable of providing mapping rules in the whole state space even when some state patterns are missing in data.

2.3.2 Backpropagation ANN Architecture

A backpropagation (BP) neural network is applied as the state-action mapping rule for action estimation in our approach. BP is a propagation of error which requires an agent to have basic knowledge of the desired output action from training data. A BP neural network calculates the error between desired output and actual BP output to propagate error back to neurons in the network. Network weights between layers are updated through training until the propagation of errors become relatively small and therefore weights value converges. BP neural network follows a gradient descent algorithm learning rule in training to gradually approximate driver actions using given input data and target output data.

A backpropagation neural network is composed of input layer, hidden layer(s) and the output layer. The kth hidden layer vector *s(k)* is computed from its upstream layer input vector *s(k-1).* A weighted sum of input and bias is calculated, and the results are transformed by a transfer function:

$$\begin{bmatrix} \boldsymbol{s_{1,k}} \\ \boldsymbol{s_{2,k}} \\ . \\ . \\ \boldsymbol{s_{m,k}} \\ . \end{bmatrix} = \left\{ \begin{matrix} \boldsymbol{h}\left[\sum_{i=1}^{n_{k-1}} \boldsymbol{w_{i,1}^{k} s_{i,k-1}} + \boldsymbol{b_1^k}\right] \\ . \\ . \\ . \\ \boldsymbol{h}\left[\sum_{i=1}^{n_{k-1}} \boldsymbol{w_{i,m}^{k} s_{i,k-1}} + \boldsymbol{b_m^k}\right] \\ . \end{matrix} \right\} \tag{2-3}$$

where n_k is the number of neurons in kth layer, $s_{m,k}$ is the value of the mth hidden neuron, $w_{i,m}^{k}$is the weight connecting the ith input neuron to the mth hidden neuron in the kth layer, b_m^k is the bias of the mth hidden neuron in the kth layer, and $h(\cdot)$ is the transfer function.

A nonlinear sigmoid transfer function is used here. This sigmoid transfer function takes the value from the summation results and turns them into values between 0 and 1:

$$\boldsymbol{h(z) = \frac{2}{1+e^{-2z}} - 1} \tag{2-4}$$

Similarly, the output layer vector *y(k)* is calculated as

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_l \\ \cdot \end{bmatrix} = \begin{Bmatrix} h\left[\sum_{i=1}^{n_k} w_{i,1}^o s_{i,k-1} + b_1^o\right] \\ \cdot \\ \cdot \\ \cdot \\ h\left[\sum_{i=1}^{n_k} w_{i,l}^o s_{i,k-1} + b_l^o\right] \\ \cdot \end{Bmatrix} \tag{2-5}$$

where $w_{i,l}^o$ is the weight connecting the *i*th hidden neuron and the *l*th output neuron, and b_l is bias for the *l*th hidden neuron. The neural network structure is shown in Figure 2-1.

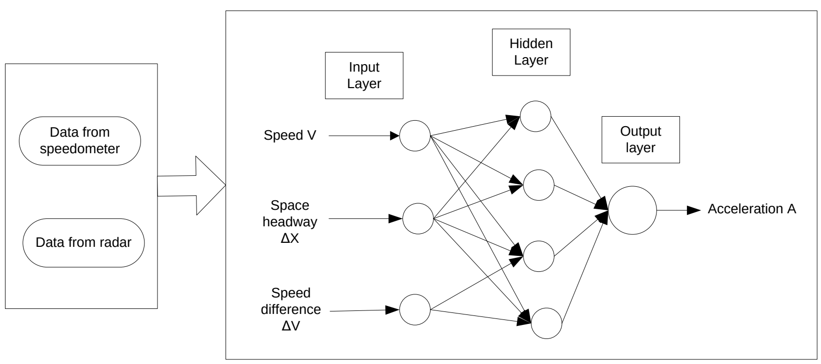

Figure 2-1 Neural Agent Structure

Backpropagation learning algorithm is divided into two phases: propagation and weight update. The propagation phase forward transfers training input through neural network to generate the propagation's output activations. Then, back propagation of output activations are transferred through the neural network using the target output to generate the gradient of all output and hidden neurons. In weight update phase, the derivative of mean squared error between neural network output and naturalistic target output and input activation are multiplied to get the gradient of weight. Weights are brought in the opposite direction of the gradient by subtracting a ratio from the weight learning rate. One iteration of BP learning can be written as

$$\boldsymbol{W_{k+1} = W_k - \alpha_k g_k} \tag{2-6}$$

where W_k is a vector of current weights and biases, g_k is the current gradient and α_k is the learning rate.

The agent receives the traffic environment information as the input layer of the neural network. Each input is weighted with an appropriate weight w. The sum of weighted inputs and bias become the input of the transfer function in hidden layer(s). Notice that we use a linear transfer function for the last layer so the neurons of the last layer are considered to be neural network output.

Before agent training, in order to compare the agent results with GHR car-following model, we use the same traffic state vehicle speed, relative distance and speed, as variables in the input layer and driver action acceleration as the single target output in the output layer. As the backpropagation training process requires a set of input and proper output (state-action pair) examples, we used trajectories from a number of naturalistic car-following situations as well as their corresponding accelerations during the training process.

2.4 NATURALISTIC DRIVING DATA

2.4.1 Naturalistic Database Description

For this research, to test our proposed method, we used data from Naturalistic Truck Driving Study (NTDS) collected by Virginia Tech Transportation Institute. As opposed to traditional epidemiological and experimental / empirical approaches, this *in situ* process used drivers who operate vehicles that have been equipped with specialized sensor, processing, and recording equipment. In effect, the vehicle becomes the data collection device. The drivers operate and interact with these vehicles during their normal driving routines while the data collection equipment is continuously recording numerous items of interest during the entire driving. Naturalistic data collection methods require a sophisticated network of sensor, processing, and recording systems. This system provides a diverse collection of both on-road driving and driver (participant, non-driving) data, including measures such as driver input and performance (e.g., lane position, headway, etc.), four camera video views, and driver activity data. This information may be supplemented by subjective data, such as questionnaire data.

As part of the NTDS study (Blanco et all in press [16]), four companies and 100 drivers participated in this study. Each participant in this on-road study was observed for approximately 4 consecutive work weeks. One hundred participants were recruited from four different trucking fleets across seven terminals and one to three trucks at each trucking fleet were instrumented (nine trucks total). After a participant finished 4 consecutive weeks of data collection, another participant started driving the instrumented truck. Three forms of data were collected by the NTDS Data Acquisition System (DAS): video, dynamic performance, and audio. Approximately 14,500 driving-data hours covering 735,000 miles traveled were collected. Nine trucks were instrumented with the DAS.

In our test, the following vehicle is the instrumented vehicle. The following vehicle measurements include speed, longitudinal and lateral accelerations, yaw angle, heading and indications of turning signal. For leading vehicle measurement, range, range-rate and azimuth were collected by instrumented forward viewing radar from the following vehicle. Both leader and following vehicle data were recorded at 10Hz. Speed collected from speedometer, range and range-rate from radar were used as the same input in both GHR model calibration and neural agent training. Videos are used to double check with trajectory data.

2.4.2 Car-following Situations Extraction and Selection

Car-following situations were automatically extracted from the database to analyze car following driver behavior. The filtering process is an iterative process where initial values and conditions were used and after the events were flagged they were reviewed in the video data to adjust the values accordingly in order to obtain minimum noise. Visual inspection of the first subsets created revealed some none car-following events, so additional filtering was thus performed to remove these events from the database.

Specifically, car following periods were extracted automatically according to these conditions:
- Radar Target ID>0

This eliminates the points in time without a radar target detected
- Radar Range<=120 meters

This represents four seconds of headway at 112 km/h
- -1.9 meters<Range*Sin (Azimuth) <1.9 meters

This restricts the data to only one lane in front of the lead vehicle
- Speed>=20km/h

This speed was used in order to minimize the effect of traffic jams, but still leave the influence of congestion in the data
- Rho-inverse <=1/610 meters-1

This limits the curvature of the roadway such that vehicles were not misidentified as being in the same lane as the subject vehicle when roadway curvature is present.
- Length of car following period while range>= 30 seconds while range<=61 meters

The automatic extraction process was verified from a sample of events through video analysis. For the random sample of 400 periods, 392 were valid.

Notice that car-following situations selected for neural network training should cover a wide range of traffic state. Theoretically, neural agent should be able to choose proper actions even from the state the agent has not experienced but within the training range. The neural network will interpolate the action for the missing state. In most of car-following situations in the database, drivers travelled on a highway or interstate with a high speed. To avoid low speed missing in traffic state, we also use low speed situations in training and calibration. Choosing the “good” situations is not an easy task. Although 1133 car-following situations are ready to use in our research, using all of them may cause redundancy in training. In order to validate the neural agent model, we want to choose the situations showing a stable driving behavior. Driver actions in some of the situations show inconsistency, using them will generate more noise.

2.5 EXPERIMENT AND RESULTS

In our past research effort, we used one car following situation (64 seconds) in BP ANN training and GHR model calibration [26]. We trained the neural network using both 1Hz and 10Hz resolution data to compare agent to GHR model and tested the contribution of high resolution data to car-following calibration and training results since 10Hz data have 10 times traffic state data in training. Figure 2-2 and Figure 2-3 showed that neural agent model outperformed GHR in both resolutions. Using 1Hz naturalistic data in training and calibration, neural agent model has a 30.5% percent less sum squared error. Using 10Hz data resolution, neural agent model significantly improved degree of accuracy while GHR model did not improve much.

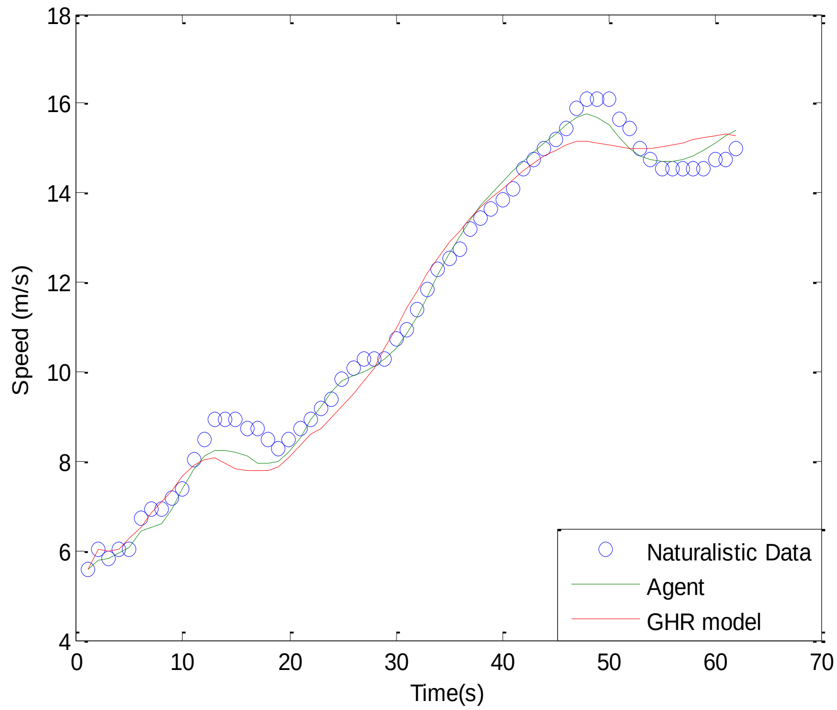

Figure 2-2 Speeds of Neural Agent and GHR Model (1Hz)

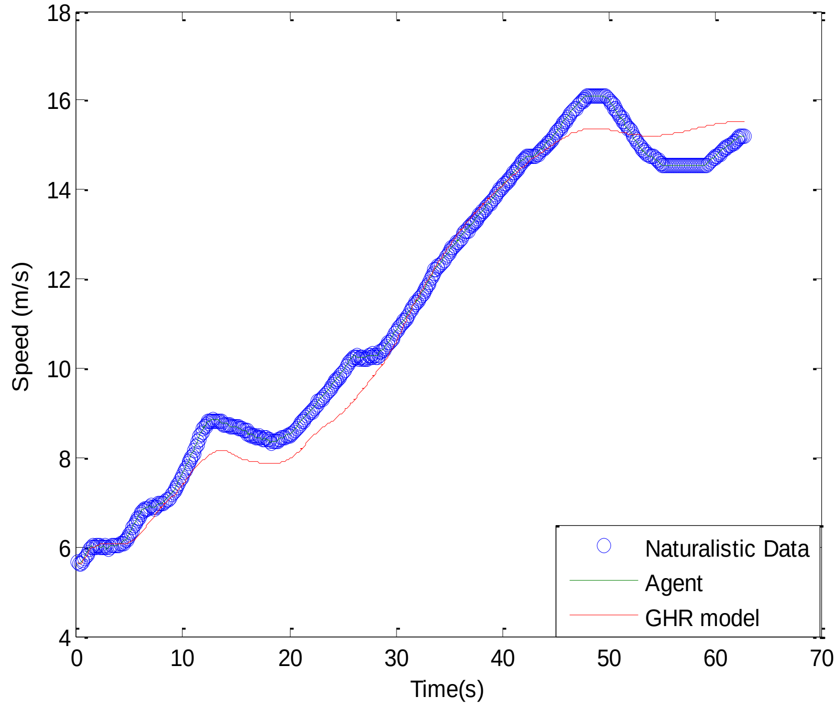

Figure 2-3 Speeds of Neural Agent and GHR (10Hz)

This research showed using 10Hz resolution in neural network training can capture driver actions quite well. See from Figure 2-3. Despite the fact that neural network may work perfect on one situation, using only one episode in training is not sufficient to capture a stable driver behavior. As mentioned before, to capture driver behavior, the training data should cover a large range for state-action mapping rule. However, when using multiple car-following episodes for training, more data may result in more variations in actions. Although using more data may reduce bias, calibration and training results may not work as well as using one episode.

In this paper, we select 10 car-following situations from one driver. The selected data includes 10732 traffic state data points for GHR model calibration and neural agent training from NTDS database. We use R^2 value as a performance measure of degree of accuracy between speeds from naturalistic data and estimated speeds from models in validation.

Naturalistic data points are randomly divided into three sets with 60% are used to train the network, 20% are used to validation how well the network generalized and the last 20% provide an independent test of network generalization to data that neural agent has never seen. Figure 2-4 shows the performance function (mean squared error) when neural agent is under training using all the available data points. Starting at a large value, the performance function decreases to a smaller value through the training process and indicates that neural agent is learning. Training stops when: the number of iterations, the performance function, the magnitude of gradient and training time greater than or less than the predefined threshold values. Also, validation stops when the validation error increased. Training continues as long as validation errors reduce. The training stops after 68 Epochs (iterations) when the network generalizes the best for validations. The result looks reasonable since test error and validation error have similar characteristics and no significant overfitting has occurred. Figure 2-5 shows the agent acceleration and naturalistic acceleration.

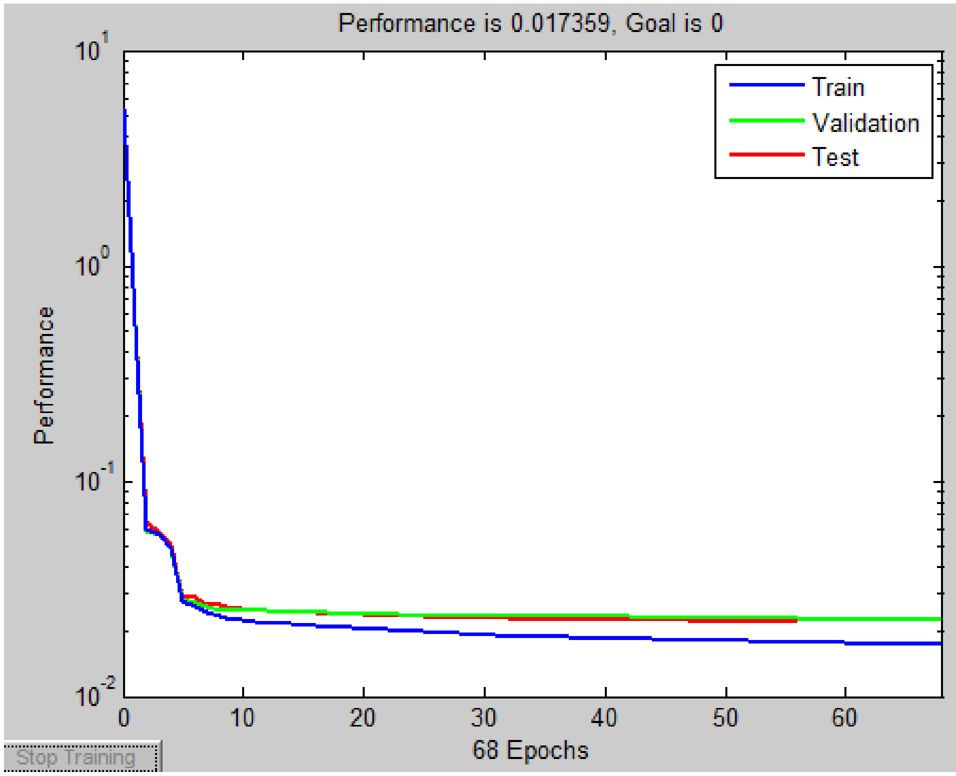

Figure 2-4 Neural Agent Model Training, Validation and Test

We pick up one car-following episode (48 seconds) as an example to illustrate validation performance of the both models in Figure 2-5 and Figure 2-6. In Figure 2-5, the neural agent has more mild accelerations than the naturalistic data. From Figure 2-6, the neural agent model simulates driver behavior quite well with a 95% R^2 value. However, the GHR model is not able to simulate driver behavior well in validation when using the optimal calibrated parameter sets (with a 57% R^2 value). Also, compared to the naturalistic data, the neural agent model looks more continuous and probably is a good approximation of naturalistic driver actions.

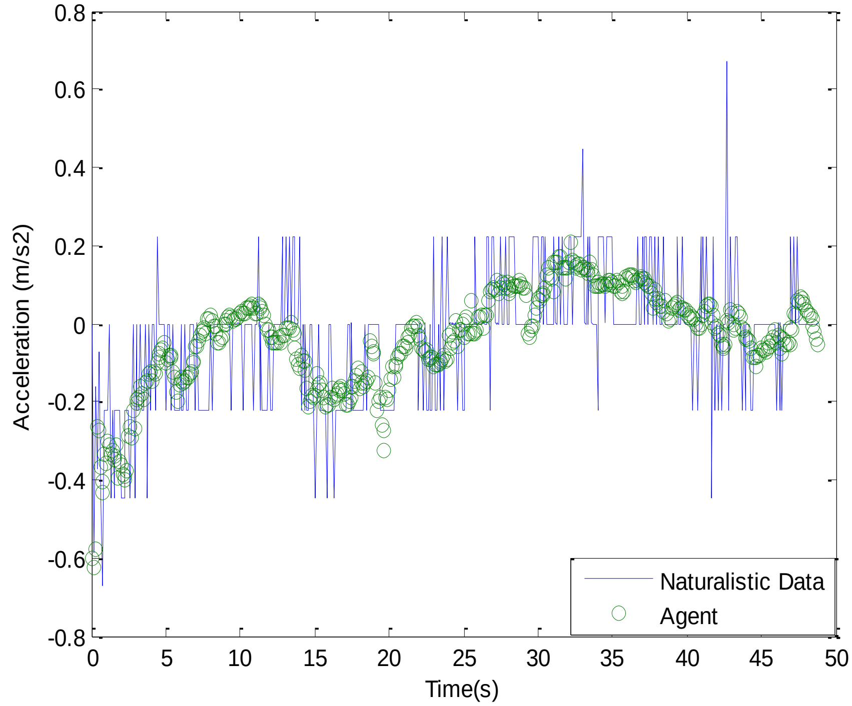

Figure 2-5 Neural Agent Acceleration Action in Validation

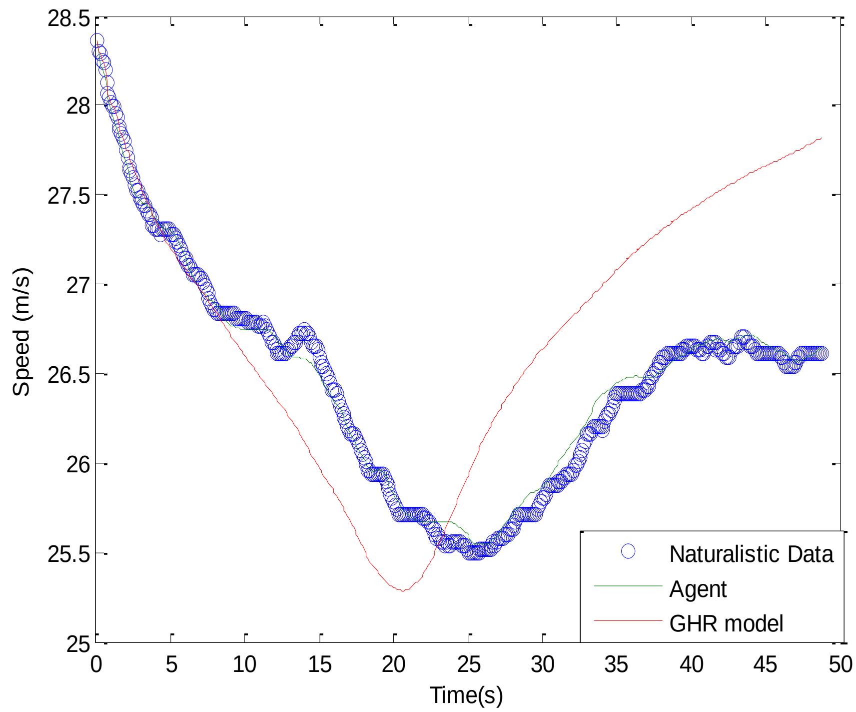

Figure 2-6 Validation Example for Neural Agent and GHR Model

2.6 CONCLUSIONS AND FUTURE RESEARCH

In this paper, as part of our preliminary research, we first propose the GHR car-following model calibration methodology using genetic algorithm. Then, we describe an agent-based backpropagation neural network modeling approach to simulate driver behavior. We used naturalistic microscopic vehicle trajectory data both in the neural agent training and the GHR model calibration. Validation is implemented on both models and shows that the neural agent is able to capture driver behavior quite well under proper training data selection. When compared to the GHR model, the neural agent model proves to have a better performance.

Alternatively, because robust calibration has been reported for Gipps and Intelligent Driver Model (IDM) models, further research may be focus on testing optimal parameters for Gipps, IDM model and compare to BP neural network. Also, we will test other types of ANN in driver action estimation.

The neural agent has been proved to be a successful way of modeling driver car-following behavior. As neural agent is able to learn from different driver reactions under different traffic environment, it is recommended to study the heterogeneity of actions when a driver faces various stimuli that current car-following models cannot handle. For example, it would be interesting to study the execution and duration of lateral lane changing behavior. In the car-following episodes we use, we found out, especially in a congested traffic environment, drivers typically take longitudinal (acceleration/deceleration) and lateral actions (swerving) simultaneously. As more data is available with car-following situations for different drivers, including safety critical events where actions are more complicated, it is feasible to use lateral trajectory information as another ANN training input variable and simulate lateral actions based on motion of the steering wheel degrees. Moreover, in this research scope, the car-following situations we use are from the same driver, further research can be done to test the robustness of neural agent and heterogeneity in different drivers.

3 AN AGENT-BASED REINFORCEMENT LEARNING MODEL FOR SIMULATING DRIVER HETEROGENEOUS BEHAVIOR DURING SAFETY CRITICAL EVENTS

Linsen Chong
Graduate Research Assistant
Charles Via Department of Civil and Environmental Engineering
301-D, Patton Hall
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061
E-mail: linsenc@vt.edu

Montasir M. Abbas, Ph.D., P.E. (Corresponding Author)
Assistant Professor
Charles Via Department of Civil and Environmental Engineering
301-A, Patton Hall
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061
Phone: (540) 231-9002
E-mail: abbas@vt.edu

Bryan Higgs
Graduate Research Assistant
Charles Via Department of Civil and Environmental Engineering
301-D, Patton Hall
Virginia Polytechnic Institute and State University
Blacksburg, VA, 24061
Phone: (540) 921-7185
E-mail: bryan.higgs@vt.edu

Alejandra Medina
Senior Research Associate
Virginia Tech Transportation Institute
3500 Transportation Research Plaza (0536)
Blacksburg, VA 24061
Phone: (540) 231-1508
E-mail: ale@vtti.vt.edu

ABSTRACT

Driving behavior in traffic has been modeled quite successfully in simulation software using predefined car-following models rules. However, most car-following models are not capable of representing naturalistic driving behavior during safety-critical events, since they were designed to adhere to safe. Also, vehicle detailed lateral maneuvering have not been simulated in most simulation software. The proposed methodology in this paper focuses on establishing a traffic state-action mapping rule to simulate naturalistic driver actions including risky behavior that a driver would take during safety critical events instead of the predefined actions by car-following models. To analyze individual driver characteristics and extract driving behavior rules, a fuzzy rule based neural network is constructed with the objective of presenting driver action rules under associated traffic states. A special machine learning approach Neuro-Fuzzy Actor Critic Reinforcement Learning (NFACRL) is proposed as a methodology to train an agent driver simulator. Vehicle longitudinal and lateral actions are estimated and used as output of this model. The simulated vehicle actions are compared with naturalistic data.

KEY WORDS: driving behavior, fuzzy logic, reinforcement learning, artificial neural network

3.1 INTRODUCTION

3.1.1 Background Information

Drivers behave differently in different regimes. As a widely used microscopic simulation model Wiedemann model [3, 7] suggests, driver behavior is different in free driving, losing in, following and emergency regimes.

Drivers maintain their desired speeds in free-driving regime where speeds are considered to be near constant and no acceleration and swerving need to be taken. When a vehicle is approaching a vehicle in front, it is coming into the following regime where drivers decide to accelerate or decelerate according to the relative distance, relative speed and driving action of the leading vehicle. In the emergency regime when driver can anticipate an upcoming conflict, evasive actions breaking and maneuvering could be taken simultaneously to avoid upcoming events.

The motivation of this paper is to model vehicle naturalistic actions during safety critical events through simulating driver decision policy to associate observed traffic state to vehicle actions. Driving policy should be driver independent and consists of multiple rules dominate for different traffic states. Fuzzy logic is used to partition traffic state space regime for these rules to look up. Reinforcement learning is used to associate the optimal action with fuzzy rules. Fuzzy logic and reinforcement learning are coordinated to extract driving policy and embed the policy to an agent simulator.

Safety critical events in this study are located in the emergency regime. Most car-following models such as the Wiedemann model assume that drivers will always keep a safety distance so that vehicles should always be able to stop before hitting its leading vehicle if an emerging conflict could happen. However, in real world cases, actions from drivers are not always safe and are not simply longitudinal deceleration. For example, a driver may take a maneuver and execute a lane change. Also, as traffic stimuli and causalities are sometimes vary case by case; it is very hard to establish predefined longitudinal and lateral action models. So far, no lateral action models have been used to simulate actions for safety critical events.

3.1.2 Traffic States and Actions during Safety Critical Events

Drivers behave according to traffic state. Traffic state is defined by a set of variables that can represent vehicle kinematic conditions and its relative location to its leading vehicle. As most of car-following model suggests, vehicle speed, distance from the leading vehicle and relative speeds are considered as traffic state variables. Additionally, we assume driving during safety critical events is a set of sequential actions, so that actions happened earlier may contribute to the action decision being taken currently. For example, when a driver decides to decelerate in the previous state, he/she is more likely to continue decelerating in the current state instead of executing a maneuver. Therefore, actions of the previous state are considered to be state variables of the current state.

3.1.3 Agent-based Modeling on Driver Behavior

Agent-based modeling (ABM) is a relatively new paradigm for exploring the behavior of complex systems [20]. Within the transportation domain, ABM is particularly good at modeling systems in which human decision making and action is a critical component. Bonabeau [23] suggests ABM is best applied to simulations when the interactions of agents are complex, nonlinear or discontinuous, the agents are heterogeneous where each individual is different, and the agents exhibit complicated behavior including learning and adaptation. In this paper, vehicle actions in safety critical events are complicated and driver independent. Therefore, ABM could provide new insights to understand driver behavior.

Driver individual behavior rules should be used to teach an agent simulator to learn. Behavior rules associate actions with traffic state and provide a driver depend driving policy for its agent to follow. So when agent experiences a certain traffic state, the policy will map the traffic state to the associated actions. By using naturalistic driving data in training, agents will learn to adopt driving rules during the training procedure and should be capable of replicating driver behavior and vehicle actions when training is completed. Accordingly, if safety critical events data is used in training, agents will perform driver specific naturalistic action which can probably result in a crash or near crash.

3.1.4 Reinforcement Learning: Brief Introduction

Reinforcement learning is a relative new methodology to develop artificial agents. Reinforcement learning is a sub-area of machine learning family to determine the actions an intelligent agent supposes to take in an environment to maximize some notion of long-term goals [27] .The objective of reinforcement learning algorithms is to find a policy that maps traffic states to their optimal actions. In our research scope, when an agent is following the optimal policy, it should act close enough to the naturalistic vehicle actions. One of the remarkable research papers by Jouffe illustrates the basic mechanism of reinforcement learning [28]. Reinforcement learning reinforces agent actions when they perform approximately close to target actions and penalize actions which are far away from them. The only information available for learning is the system feedback, which describes in terms of reward and punishment on the task the agent has to realize. At each time step, the agent receives a reinforcement signal according to the last action it has performed in the previous state. The problem involves optimizing not only the direct reinforcement, but also the total amount of reinforcements the agent can receive in the future. Finally, reinforcement learning should be able to extract driving rules from naturalistic dataset and establish a similar driver specific state action mapping rules.

3.1.5 Reinforcement Learning Applied in Transportation

Reinforcement learning has been applied a lot in network route choice analysis and real time traffic signal control. Avineri proposed a feedback reinforcement mechanism in modeling route-choice decision-making under uncertainty [29]. Bogers et al used reinforcement learning to simulate traveler route choice behavior with the assumption that travelers learn about route travel time performance from their experiences [30]. In traffic signal optimization research, Abdulhai proposed a Q-learning algorithm in an isolated intersection and then a corridor with coordinated intersections to find the optimal timing plans in a dynamic traffic environment [31, 32]. Bingham used neural network fuzzy logic to partition traffic state of intersection for controller agent to learn [33]. Abbas and his research group developed a number of reinforcement learning control logics for signalized intersections including dilemma zone optimal green time extension and real time optimal cycle length and split determination [34-36]. Zhang et al used a Neuro-Fuzzy Actor-Critic control method for a real time isolated intersection control and arterial control in Texas [37].

3.1.6 Drawbacks of Existing Reinforcement Learning Algorithms

Although reinforcement learning in these research efforts has achieved great accomplishments, still many limitations occur. Dimension problem is a great pain of most agent controllers/simulators. If the number of traffic state variables is large, computation time would become a burden in training. Because the reinforcement learning mechanism has to associate state with action and update mapping rules, more dimensions in traffic state will cause exponentially increasing number of mapping rules.

Moreover, the existing reinforcement learning efforts are mainly dealing with discrete states and actions mapping problem so that mapping rules can always follow a look-up table to relate a discrete traffic state to a discrete action. However, traffic state variables (vehicle speed for example) especially in our study have continuous distributions, which make it infeasible to build a look-up table to relate states to actions.

Another main drawback of the established reinforcement learning algorithm in transportation is the lack of ability to generate continuous actions. Instead, most of the studies use discrete action sets consisting of a limited number of actions for the agent to select. Abdulhai's Q learning mechanism has a binary action option phase extension or end [31]. Zhang improved this action selection methodology but the controller agent action is still restricted by choosing only one of the provided six actions [37]. This problem makes established reinforcement learning algorithms limited in continuous action simulation.

3.1.7 The Proposed Reinforcement Learning Methodology

In this research, we proposed a revised reinforcement learning methodology to solve the traffic state dimensional problem and continuous action generation problem. In fact, this revised Neuro-Fuzzy Actor Critic Reinforcement Learning approach can alleviate most of the computation burden that happens to the previously developed approaches. Using naturalistic traffic states and driving actions during crash and near crash events, this approach can reproduce actual driver behavior during safety critical events.

For the paper layout, first the proposed Neuro-Fuzzy Actor Critic Reinforcement Learning (NFACRL) methodology is presented including its framework, input and output components. Secondly, the naturalistic driving database and safety critical events extraction process are described. Subsequently, the agent training results are presented with a performance evaluation. Finally, cross validation between different agents are performed, driver heterogeneity is illustrated and the idea of "mega" agent is presented.

3.2 NFACRL METHOD

3.2.1 NFACRL Structure

NFACRL is a type of actor-critic reinforcement learning method. In this study, NFACRL uses current traffic state as inputs and generates vehicle action as outputs. NFACRL then loads following states data and decides the following actions. As its name indicates, NFACRL has a neural network structure and uses fuzzy logic to cluster traffic state variables and transfer information between layers. Continuous traffic states are converted into a limited number of discrete fuzzy sets and decrease dimensionality. Fuzzy rules are considered as state-action mapping rules to associate fuzzy sets and actions. Reinforcement learning is the training process to obtain fuzzy rules.

As Figure 3-1 shows, our proposed NFACRL structure has four layers. The first layer is the input layer. Each node of it represents a state variable. The second layer is the fuzzy membership layer. Continuous states variables are fuzzified into linguistic terms as "Speed is High" and "Speed is Low" in this layer. Each node is a fuzzy set with a membership function as output. Notice that one state input variable should have more than one fuzzy sets so that continuous input variables could be transferred to discrete sets without losing much information. Fuzzy membership functions can be triangular, trapezoidal and Gaussian [38] depend on the distributions of state variables. The third layer is the fuzzy rules layer. Each rule is connected with a number of antecedents (discrete fuzzy sets) from fuzzy set layer. A firing strength is the output of a fuzzy rule which indicates its strength. The fourth layer is the actor critic layer including Actor and Critic nodes. Critic nodes are associated with the value of next state after the agent choosing an action. Actor nodes are from one discrete action sets. Each fuzzy rule selects one discrete action. NFACRL output action is the weighted average of all the actions selected by fuzzy rules where fuzzy rule strengths are the weights. Several following paragraphs will elaborate more on the structure design.

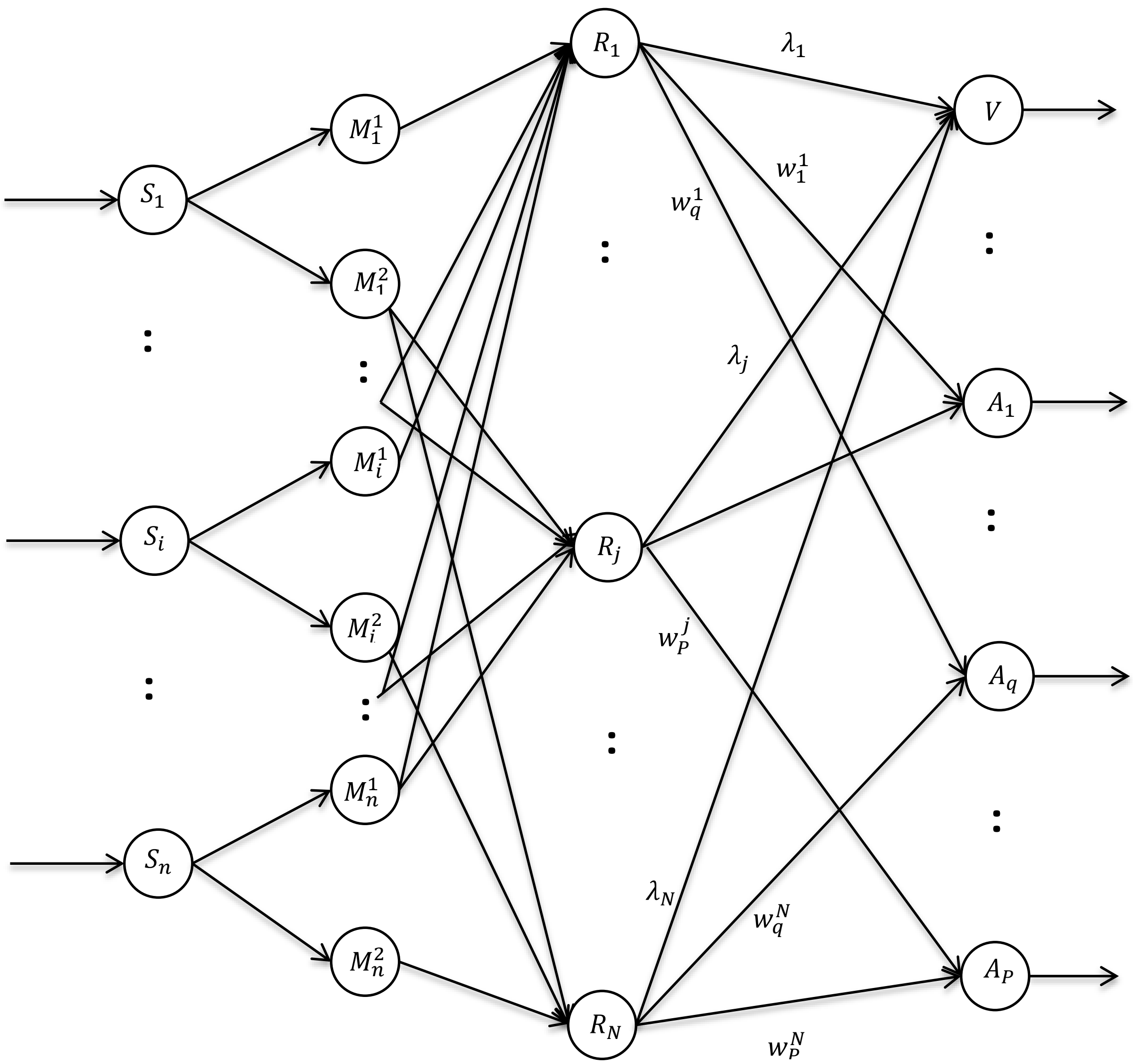

Figure 3-1 NFACRL Structure

S_i=the i^{th} input variable (state)
n =number of input variables
NM_i=number of fuzzy sets or membership functions for the S_i
$M_i^{a(i)}$=$a(i)^{th}$ fuzzy set or membership function for the i^{th} input variable
R_j=the j^{th} fuzzy rule
N=number of fuzzy rules
λ_j=weight between j^{th} fuzzy rule and critic
w_q^j=weight between j^{th} fuzzy rule and action q
V =critic output
A_q =output of q^{th} discrete action
Where $i = 1, \dots n, a(i) = 1, .. NM_i, j = 1, \dots, m$ and $q = 1, .., P$

3.2.2 Layer Design

3.2.2.1 State Layer

Most safety distance based car-following models assume speed, relative speed and distance as important stimuli when leading and following vehicles are interacting with each other. In addition, during safety critical events, since lateral lane offset (lateral position relative to the center of the lane), acceleration and vehicle yaw angle (angle of vehicle longitudinal relative axis to lane markings) of the previous state are also important for the driver action decision process, thus these three variables are also incorporated into state variables. Traffic state variables and defined as $S_6 = o$ (**3-1**) in NFACRL structure.

$$
\begin{aligned}
S_1 &= v \\
S_2 &= \Delta x \\
S_3 &= \Delta v \\
S_4 &= a' \\
S_5 &= y' \\
S_6 &= o
\end{aligned}
\tag{3-1}
$$

where S_i is the ith state variable, v is the vehicle speed, Δx is the space headway (relative distance from the leading vehicle), Δv is the relative speed (speed of the leading vehicle minus the following vehicle), a' is the previous acceleration, y' is the previous yaw angle and o is the vehicle lateral position offset.

3.2.2.2 Fuzzy Sets Layer

Each state variable is linked to two mutual exclusive fuzzy sets in the second layer in this study. For state variable S_i, these two fuzzy sets are defined by linguistic terms: "S_i is high" and "S_i is low". Each fuzzy set is associated with a membership function. A triangular fuzzy membership function is used to uniform a state variable and transfer its information to the two fuzzy sets. Triangular membership function requires upper and lower bound of each state variable which could be extracted from safety critical events that a driver had experienced. For each state, lower bound and upper bound are the maximum and minimum state variable value. During the training process, upper and lower bounds are considered to be constant.

Membership functions for fuzzy sets "Low" and "High" are defined as:

$$\mu_{Low}(S_i) = \begin{cases} 1 & S_i \leq S_{lb,i} \\ \frac{S_{ub,i}-S_i}{S_{ub,i}-S_{lb,i}} & S_{lb,i} < S_i < S_{ub,i} \; i = 1,2..6 \\ 0 & S_i \geq S_{ub,i} \end{cases} \qquad (3\text{-}2)$$

$$\mu_{High}(S_i) = \begin{cases} 0 & S_i \leq S_{lb,i} \\ \frac{S_i-S_{lb,i}}{S_{ub,i}-S_{lb,i}} & S_{lb,i} < S_i < S_{ub,i} \; i = 1,2..6 \\ 1 & S_i \geq S_{ub,i} \end{cases} \qquad (3\text{-}3)$$

where $\mu_{Low}(S_i)$ is the membership function for fuzzy set "S_i is Low", $\mu_{High}(S_i)$ is the membership function for fuzzy set "S_i is High", $S_{lb,i}$ is the lower bound of state variable S_i and $S_{ub,i}$ is the upper bound of state variable S_i

3.2.2.3 Fuzzy Rules Layer

Neurons in the third layer represent fuzzy rules. Fuzzy rules provide a state action mapping policy to determine the "optimal" actions in the fourth layer for NFACRL agent to select. Each fuzzy rule is applied to one combination of fuzzy sets from the second layer. For example, a fuzzy rule can be represented as:

WHEN "S_1 is low" and "S_2 is low" and "S_3 is high" and "S_4 is high" and "S_5 is high""S_6 is high", THEN Action "Deceleration $a = -0.05g$"

In our designed neural network, each fuzzy rule is associated with six fuzzy sets originated from six continuous state variables. Since each state variable has two fuzzy sets "Low" and "High", the number of fuzzy sets combination should be $2^6 = 64$. Equivalently, there are 16 fuzzy rules in this study.

The product of the membership functions connected with each rule is used to compute its firing strength.

Firing strength for the j^{th} fuzzy rule is

$$FS_{Rj} = \prod_{i=1}^{6} \mu_{a(i)}^{j}(S_i) \tag{3-4}$$

where $a(i)$ is the linguistic term of fuzzy set (either "Low" or "High") for the i^{th} input state variable and j represents the j^{th} fuzzy rule.

3.2.2.4 Actor and Critic Layer

Critic nodes represent the value of next state under the current policy. Actor nodes represent actions to be selected by the fuzzy rules. Two types of actions acceleration and yaw angle are used as equivalent vehicle longitudinal and lateral actions. Notice that acceleration and yaw angle are continuous variables. However, due to the constraints of neural network structure, it is impossible to enumerate all the possible continuous actions. Instead, two discrete actions sets are defined here which include a number of representative action values for fuzzy rules to select. Discrete actions here are neurons of the fourth layers in the neural network. In this study, five discrete longitudinal acceleration values and five lateral yaw angle values are considered as action candidates. Two sets are used to store actions, one for acceleration A and one for yaw angle Y. Ten action values are considered as constant parameters during training.

$A = \{a_{d1}, a_{d2}, a_{d3}, a_{d4}, a_{d5}\}$
$Y = \{y_{d1}, y_{d2}, y_{d3}, y_{d4}, y_{d5}\}$

3.2.2.5 Neuron Weights

Weights are located between fuzzy rule layer and action layer. Critic weight λ^j links the j^{th} fuzzy rule to critic output. Action w_k^j links the j^{th} fuzzy rule to the k^{th} action output. Weights w_k^j show competitions between discrete actions in the same action set and are used as references for action selection methodology for fuzzy rule j. Weights λ^j are used to compute the value of traffic state V_S which is the sum product of firing strength FS_{R_j} and weights λ^j.

$$V_S = \sum_{j=1}^{64} FS_{R_j} \lambda_j \tag{3-5}$$

where FS_{R_j} is the firing strength for the j^{th}fuzzy rule R_j. V_s is the value of state s.

3.2.3 NFACRL Output Actions

3.2.3.1 Greedy Selection Algorithm

Fuzzy rule R_j selects discrete action a_{dk} from set A when weight w_{ak}^{j} is the largest among accelerations $a_{d1}$to a_{d5}. Similarly, R_j selects discrete action y_{dk} from set Y when weight w_{yk}^{j} is the largest in set Y.

3.2.3.2 Continuous Action Generation

For each fuzzy rule, one discrete acceleration and one discrete yaw angle are selected. Therefore, the output actions acceleration a and y are the weighted average of all the selected discrete actions by all the rules. Firing strengths of fuzzy rules are used as weights so NFACRL agent actions are generated as

$$a = \sum_{j=1}^{64} FS_{R_j} * a_{dk}^{j} \tag{3-6}$$

$$y = \sum_{j=1}^{64} FS_{R_j} * y_{dk}^{j} \tag{3-7}$$

where a is continuous longitudinal acceleration and y is lateral yaw angle.

Instead of generating discrete actions, this NFACRL method is able to generate continuous action variables.

3.2.4 Weights Update: Reinforcement Learning Algorithm

Reinforcement learning algorithm updates the weights between the third layer and the fourth layer.

3.2.4.1 Weights Update

Weights $\lambda^{j}$and w_{k}^{j} are updated according to temporal difference (TD) error. TD error consists of value of the current state, value of the following state and a predefined reward function. Reward function is an estimation of how good the selected actions are at state S_t.

Temporal difference (TD) error [27] is calculated as

$$\delta_{a,t} = r_{a,t+1} + \gamma V_{s_{t+1}} - V_{s_t} \tag{3-8}$$

$$\delta_{y,t} = r_{y,t+1} + \gamma V_{s_{t+1}} - V_{s_t} \tag{3-9}$$

where $r_{a,t+1}$is the reward function when action a is taken at state S_t, $r_{q,t+1}$is the reward function when action y is taken, γ is the discounting factor. V_{s_t} is the value of current state and $V_{s_{t+1}}$ is the value of the following state.
V_{t+1} is calculated as

$$V_{s_{t+1}} = \sum_{j=1}^{N} FS_{R_j,t+1}\lambda_{j,t} \quad (3\text{-}10)$$

Notice that as the NFACRL agent generates two actions, two responding reward functions are computed and two temporal difference errors are calculated.

λ and w are updated by TD errors, where

$$\lambda_{a,t+1}^{j} = \lambda_{a,t}^{j} + \beta\delta_{a,t}FS_{R_j,t} \quad (3\text{-}11)$$

$$\lambda_{y,t+1}^{j} = \lambda_{y,t}^{j} + \beta\delta_{y,t}FS_{R_j,t} \quad (3\text{-}12)$$

$$w_{ak,t+1}^{j} = w_{ak,t}^{j} + \ \beta\delta_{t,a}FS_{R_j,t} \quad (3\text{-}13)$$

$$w_{yk,t+1}^{j} = w_{yk,t}^{j} + \ \beta\delta_{y,a}FS_{R_j,t} \quad (3\text{-}14)$$

where β is the learning rate

TD errors only update weights connected to the selected discrete actions a and y. From Equation $w_{ak,t+1}^{j} = w_{ak,t}^{j} + \ \beta\delta_{t,a}FS_{R_j,t}$ (**3-13**) and $w_{yk,t+1}^{j} = w_{yk,t}^{j} + \ \beta\delta_{y,a}FS_{R_j,t}$ (**3-14**), when discrete action a_{dk}/y_{dk} is selected by rule R_j, only weight $w_{ak,t}^{j}/w_{yk,t}^{j}$ is updated.

3.2.4.2 Reward Function

Reward function provides guidance for an agent to follow. Reward function encourages agent to take actions which are close to driver actions and penalizes actions which are far away. Agent will know the outcome of the actions only when they have been chosen. When the performance of an action outcome is good, reward function should be positive to provide a greater probability for the agent to choose in the future. Vice versa, when performance is bad, reward function should be negative and reduces the probability to choose. In this paper, actions from naturalistic driving database are used as references.

First, relative error is calculated as

$$e_a = |\frac{a-a_n}{a_n}| \qquad (3\text{-}15)$$

for acceleration and

$$e_y = |\frac{y-y_n}{y_n}| \qquad (3\text{-}16)$$

for yaw angle

where e_a is the absolute relative error of acceleration, e_y is the absolute relative error of yaw angle, a_n, y_n are the naturalistic actions from database.

Then, a non-negative parameter e_{th} is defined as an acceptance threshold. When e_a or e_y is less than e_{th}, reward function is positive and the value of related weights will increase and increases the probability of fuzzy rules selecting a and y in the future.

$$r_a = \alpha(e_{th} - e_a) \qquad (3\text{-}17)$$

$$r_y = \alpha(e_{th} - e_y) \qquad (3\text{-}18)$$

where r_a is the reward function of acceleration, r_y is the reward function of yaw angle and α is the scaling factor.

Initially, when an agent has no idea about which discrete action to take, it may choose an extremely bad action and e_a, e_y become too large. When that condition happens, reward function becomes negatively high and causes interruption to weights update. Therefore, reward functions need to be adjusted to make sure they have a lower bound.

$$r_a = -\alpha \text{ when } e_a \geq 1 \qquad (3\text{-}19)$$

and

$$r_y = -\alpha \text{ when } e_y \geq 1 \qquad (3\text{-}20)$$

3.3 NATURALISTIC DATA AND TRAINING PROCEDURE

3.3.1 Database Description

We use safety critical events from the 8 Truck Database of Naturalistic Truck Driving Study (NTDS) collected by Virginia Tech Transportation Institute (VTTI) for agent training. NTDS uses drivers who operate vehicles that have been equipped with specialized sensor, processing, and recording equipment. In effect, the vehicle becomes the data collection device. The drivers operate and interact with these vehicles while the data collection equipment is continuously recording numerous items of interest during the entire driving periods. This system provides a diverse collection of both on-road driving and driver (participant, non-driving) data, including measures such as driver input and performance (e.g., lane position, headway, etc.), four camera video views, and driver activity data.

In 8 Truck Study (Blanco et all, in press [16]), each participant in the on-road study was observed for approximately 4 consecutive work weeks. One hundred participants were recruited from four different trucking fleets across seven terminals and one to three trucks at each trucking fleet were instrumented (nine trucks total). After a participant finished 4 consecutive weeks of data collection, another participant started driving the instrumented truck. Three forms of data were collected by the Naturalistic Truck Driving Study (NTDS) Data Acquisition System (DAS): video, dynamic performance, and audio. Approximately 14,500 driving-data hours covering 735,000 miles traveled were collected. Nine trucks were instrumented with the DAS.

In our test, the following vehicle is the instrumented vehicle. The measured subject vehicle data include speed, yaw angle, lane offset and accelerations. Range and range-rate were collected by instrumented forward viewing radar from the following vehicle. Speed was collected from the speedometer. Yaw angle and lane offset were extracted from video recording. Acceleration from the accelerometer was used as longitudinal traffic action and yaw angle was used as lateral action.

3.3.2 Safety Critical Events Extraction

Crash and near crash are used as safety critical events in this study. The characteristics of crash and near crash events were identified and analyzed in a previous work by VTTI [15].

Crash: Any contact with an object, either moving or fixed, at any speed. Crash includes other vehicles, roadside barriers, and objects on or off of the roadway, pedestrians, cyclists, or animals.

Tire strike is also considered to be a crash: Any contact with an object, either moving or fixed, at any speed in which kinetic energy is measurably transferred or dissipated where the contact occurs on the truck’s tire only. No damage occurs during these events (e.g., a truck is making a right turn at an intersection and runs over the sidewalk/curb with a tire).

Near-Crash: Any circumstance that requires a rapid, evasive maneuver (e.g., hard braking, steering) by the subject vehicle or any other vehicle, pedestrian, cyclist, or animal, in order to avoid a crash.

For an event to be flagged, only one of the triggers has to be met. Those triggers are as follows:

- Longitudinal Acceleration greater than or equal to -0.2g
- Forward Time-to-Collision of less than or equal to 2 seconds
- Swerve greater than or equal to 2 rad/sec^2
- Lane Tracker Status equals abort (lane deviation)
- Critical Incident Button
- Analyst Identified

3.3.3 State and Action Variables Selection

Before training, training parameters: upper/lower bounds of fuzzy sets and discrete action set values are determined from naturalistic events. In this experiment, we use minimum and maximum state variables as lower and upper bounds.

As discrete action sets should represent driver naturalistic actions, one basic idea is to extract representative actions from the naturalistic driving data. Statistically, the five representative action values of accelerations/ yaws can be the minimum, lower quartile (25th percentile, cuts off lowest 25% of data), median (50th percentile, cuts off 50% of data), upper quartile (75th percentile, cuts off highest 25% of data) and the maximum. Notice that agent output action is the weighted average of discrete actions. When using minimum and maximum discrete actions, the weighted average will make agent action impossible to reach naturalistic maximum and minimum. Therefore, 1.2 times maximum value is used instead of maximum when maximum is positive and 0.9 times maximum is used when maximum is negative. 0.9 times minimum is used when minimum is positive and 1.2 times minimum is used when minimum is negative. Through this change, the range of agent action is extended.

3.3.4 Training Process

In our designed training process, at one time step of one event, fuzzy rules scan theirs associated weights, select the optimal actions and update the weights. NFACRL keeps updating the weights from the beginning of the events until the end. In fact, weights are trained and updated 10 times the length of events (10Hz resolution data) per event iteration. Theoretically, when the differences of critic/actor weights between two consecutive iterations become very small, the training process is considered to be finished. However, the proposed NFACRL is a heuristic methodology so no global optimal agent behavior is guaranteed. Therefore, it is possible that the convergence of the weights may be premature and NFACRL results in a local optimal solution. One way to avoid this premature convergence is to give the agent sufficient training iterations. In this event example, we tried 400 iterations in agent training. As a safety critical event has about 1500 timing steps (150 seconds) in average, driver fuzzy rules have been trained 1500*400=600000 times and the agent should result in a near optimal approximation on naturalistic driver behavior.

During the learning process, a memory discount factor γ, a learning factor β and a reward function scaling factor α affect the learning speed of NFACRL. γ controls the memory fade speed where the value of recently occurring states are weighted more. β shows how fast the agent adjusts to the new information. α controls the magnitude of the reward function and weights change and e_{th} controls the sign of the reward function. In this test, $\beta = 0.6$, $\gamma = 0.9$, $\alpha = 10$ and $e_{th} = 0.2$.

Based on our preliminary efforts, two drivers were selected. We use all their safety critical events available which should be sufficient for training purposes and avoiding bias effects. Although conditions and casualties of different events can be totally different, NFACRL theoretically can still capture rules of individual drivers quite well because different events are located at different state spaces dominated by different fuzzy rules.

3.4 AGENT TRAINING RESULTS

3.4.1 Preliminary Training Results

Figure 3-2 and Figure 3-3 show the longitudinal agent action acceleration and lateral action yaw angle during one event of Driver Agent A. The blue scatter plots represent the naturalistic driving actions and the green curves show the agent actions. NFACRL captures driver naturalistic behavior quite well in this event with an R squared degree of accuracy 0.981 and 0.967 for acceleration and yaw angle respectively. We also test NFACRL performance on the other event of the same driver, see Figure 3-4 and Figure 3-5.

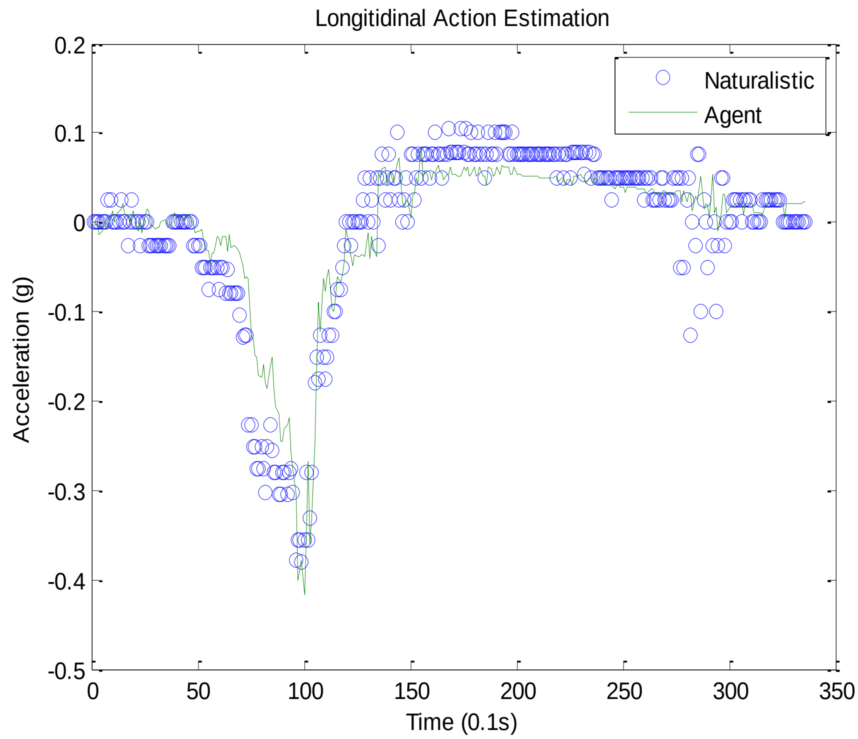

Figure 3-2 Acceleration of Agent A, Event A1

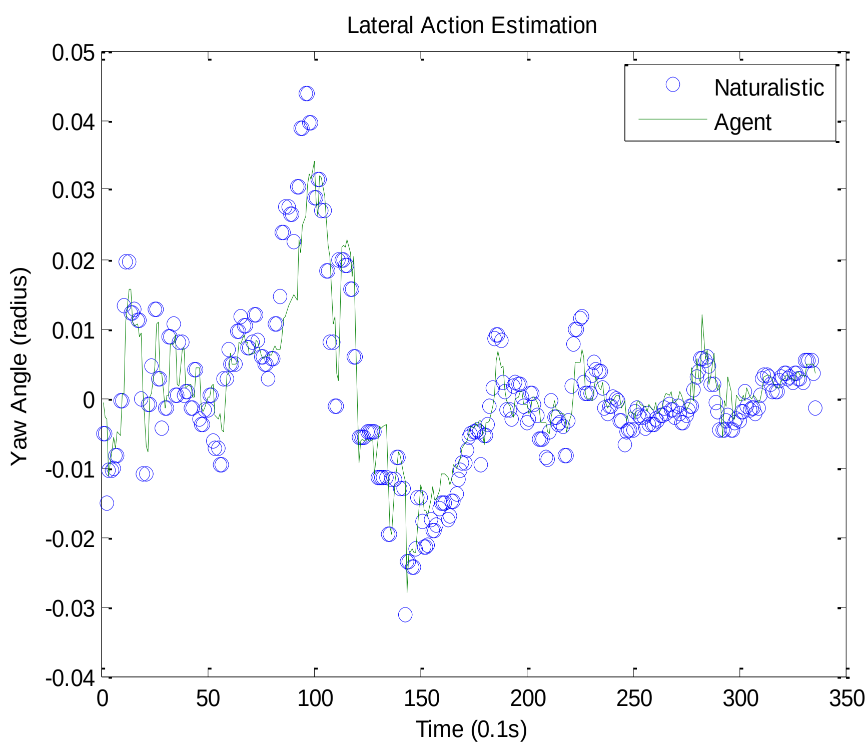

Figure 3-3 Yaw angle of Agent A, Event A1

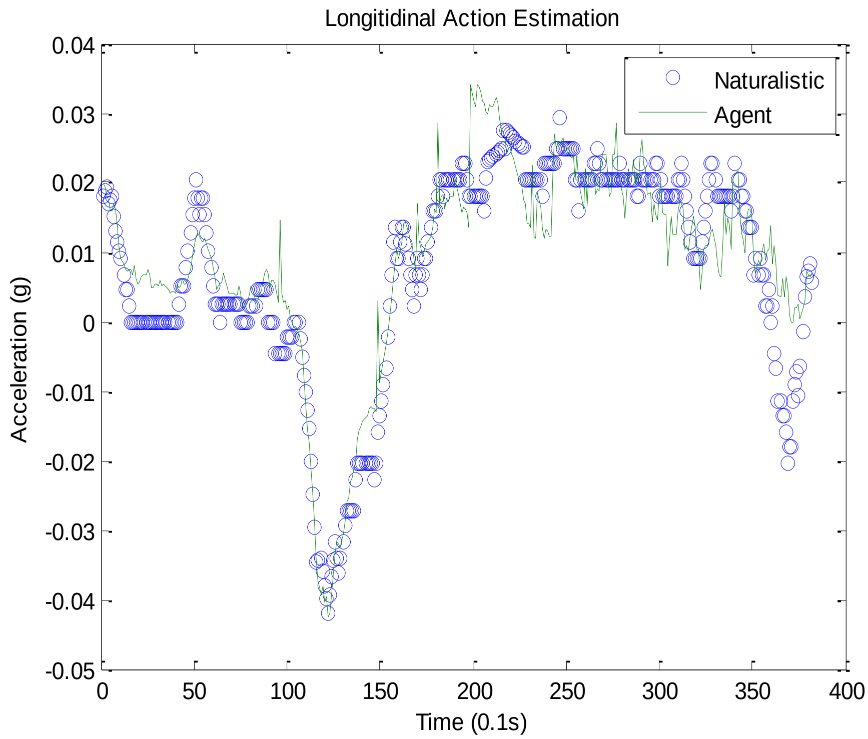

Figure 3-4 Acceleration of Agent A, Event A2

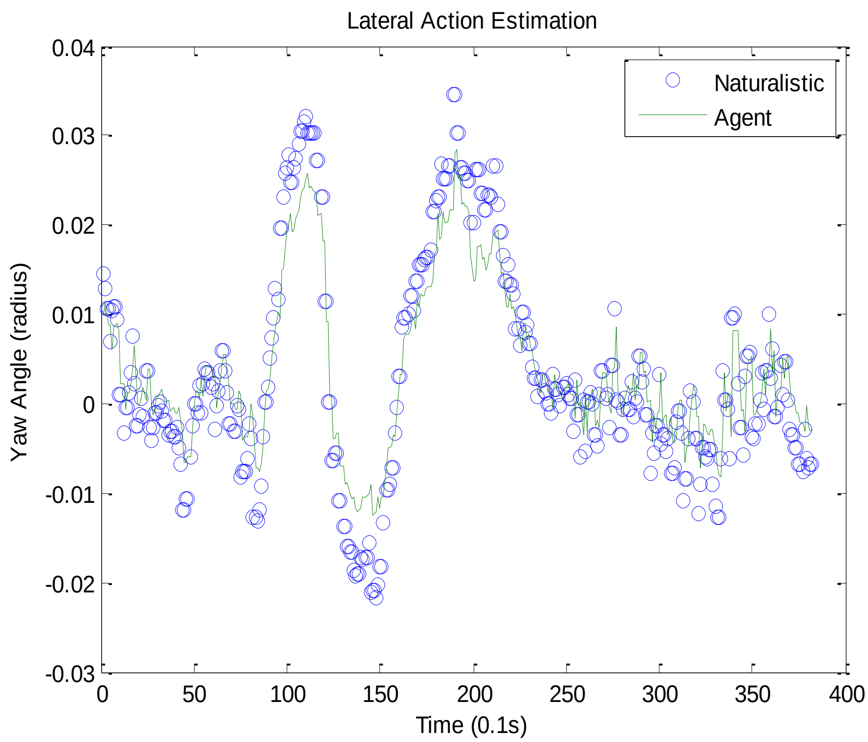

Figure 3-5 Yaw angle of Agent A, Event A2

The Agent A captures Driver A's naturalistic behavior during both events A2 and A1 quite well. It seems that even if Event A1 and A2 may vary, Driver A seems to behave similarly. It may prove the concept that NFACRL methodology will not result in an "average" driving behavior even when using more events as in training.

From the above four figures, we can see that during some part of the events, such as $t = 28s$ to $30s$ in Figure 3-2, the action of Agent A diverges a little bit from naturalistic data. Two reasons may explain the differences: data collection errors in traffic state variables and unstable driver behavior within the event. Occasionally, the leading vehicle does not fall into the range of the radar detection zone so the agent assumes there is no vehicle in front. Consequently, the wrong traffic state leads to the wrong action. Also, human behavior is hard to keep constant. Within one event, a driver may react differently in similar traffic states.

3.4.2 Cross Validation Results

The idea of cross validation is to show the different behavior of different drivers when they experience an "imaginary" same event. To reach this purpose, driving behavior of one driver (for example driver A) needs to be obtained by using his/her events in training and subsequently states from another event (for example Event B) experienced by another driver (driver B) should be used to show the driver A's actions during Event B.

In our test, we trained Agent A and simulate Agent A's actions using the event states from Driver B. Similarly, we simulate Agent B's actions using Driver A's events. To compare, first, Figure 3-6 and Figure 3-7 show an event example (Event B1) of Agent B trained by using all its safety critical events.

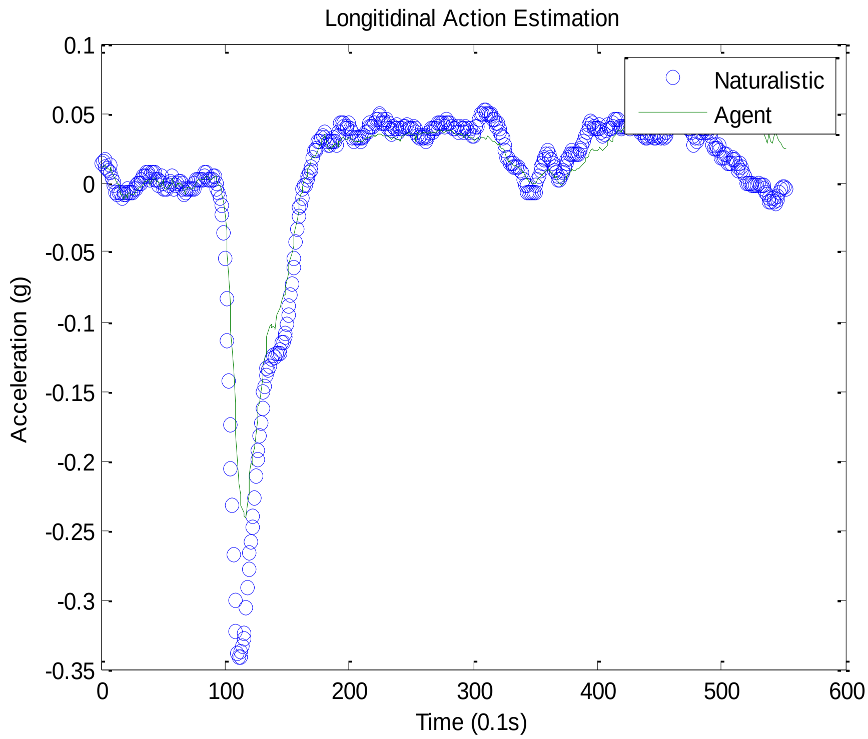

Figure 3-6 Acceleration of Agent B, Event B1

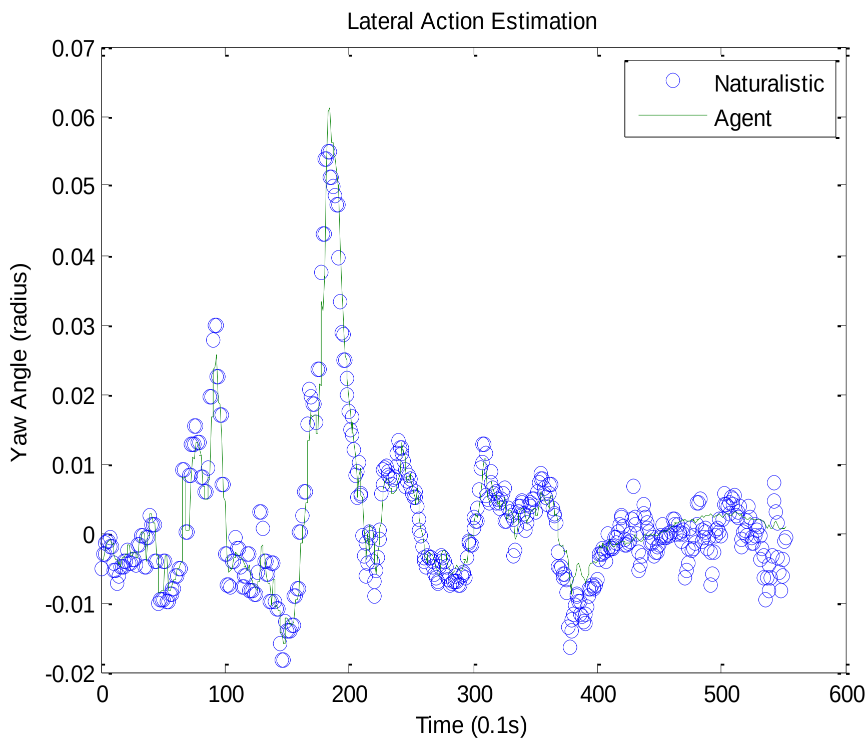

Figure 3-7 Yaw Angle of Agent B, Event B1

Figure 3-8 and Figure 3-9 show the longitudinal and lateral actions of Agent B in one event from Driver A. Figure 3-10 and Figure 3-11 show the longitudinal and lateral actions of Agent A in one event from Driver B. It is very clear that Driver A and Driver B perform heterogeneous behavior.

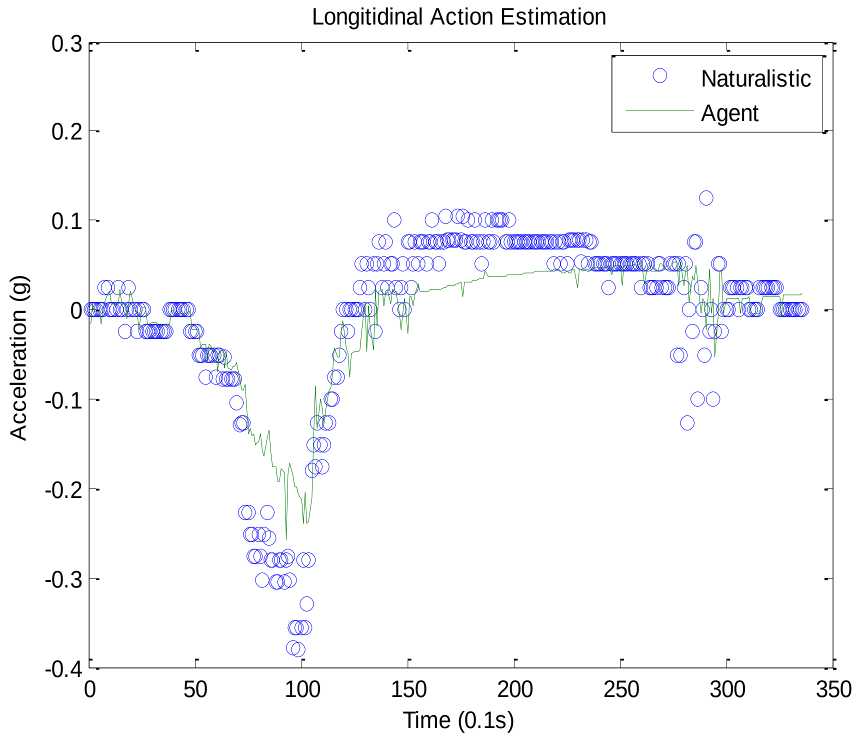

Figure 3-8 Acceleration of Agent B, Event A1

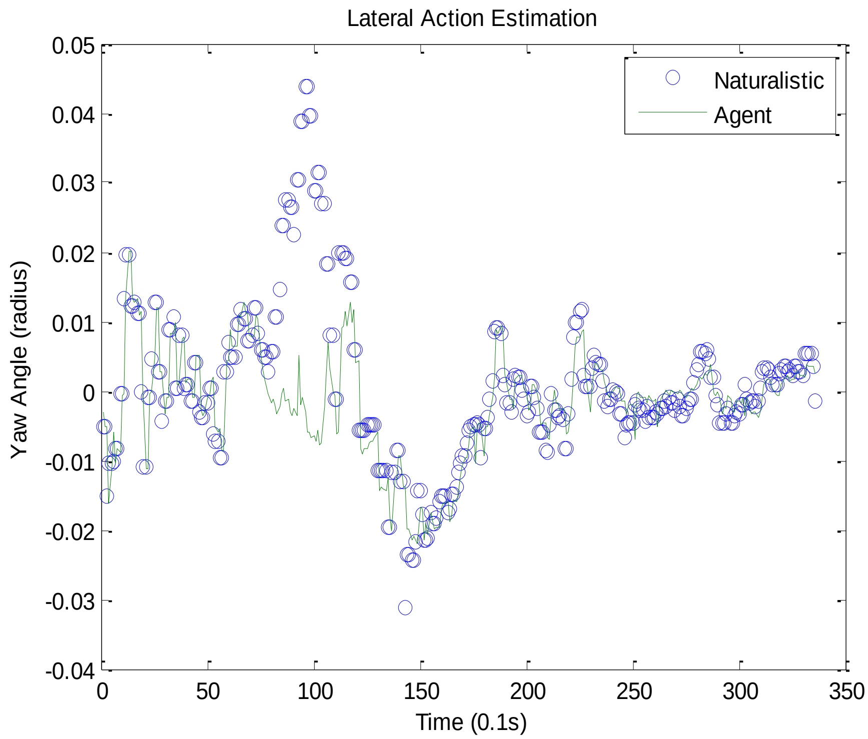

Figure 3-9 Yaw Angle of Agent B, Event A1

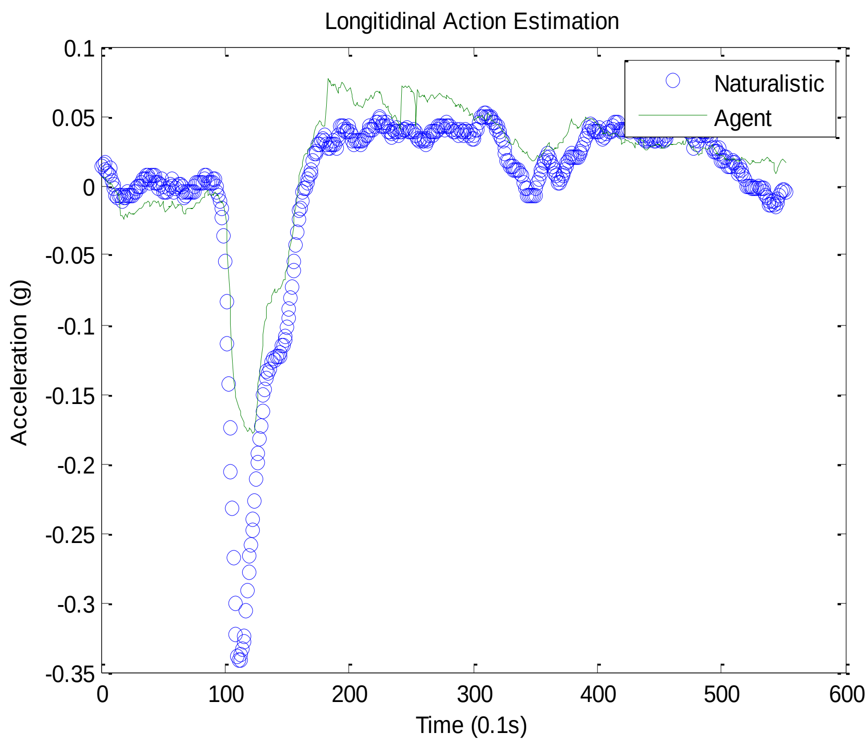

Figure 3-10 Acceleration of Agent A, Event B1

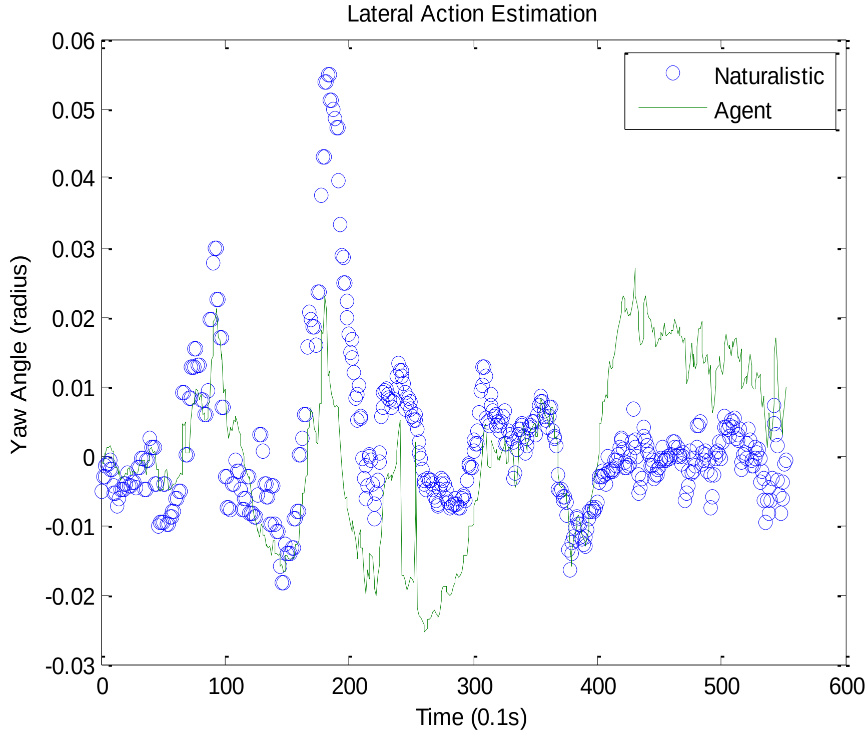

Figure 3-11 Acceleration of Agent A, Event B1

Table 3-1 uses R squared values as a statistical representation of the degree of accuracy of Agent performance. The upper left part and lower right part of Table 3-1 represent degree of approximation when using events of same driver in training and validation. It is much greater than the cross validation part (upper right and lower left) which also shows clear heterogeneities between two drivers.

Table 3-1 R squared values for cross validation

Event	Agent A		Agent B	
	Acceleration	Yaw	Acceleration	Yaw
Driver A	0.98	0.97	0.81	0.83
Driver B	0.82	0.60	0.97	0.92

3.4.3 The "Mega Agent" Idea

We want to design an imaginary agent which can capture the behaviors of both Driver A and Driver B but not an average behavior. Our revised NFACRL methodology may handle this challenge. According to the nature of traffic state variables, the state space in this problem has six dimensions. Thus differences in one dimension (lower speed and high speed for example) can make two states separated from each other in the state space. From the naturalistic data, as we found out, state variables from different events vary a lot, so it is less likely that two events have many overlaps in state space. Notice that NFACRL different fuzzy rules dominate different regimes of the state space divided by fuzzy sets of state variables. Consequently, the training process of the imaginary "Mega" Agent, is actually adjusting the fuzzy rules which are taking charge of relevant state space regimes where safety critical events are located.

We use all the events of Driver A and Driver B to train the Mega Agent. Training parameters fuzzy sets thresholds and discrete action sets are adjusted before training. Performances of the Mega Agent (use Event A1 and B1) are presented in Figure 3-12, Figure 3-13, Figure 3-14 and Figure 3-15.

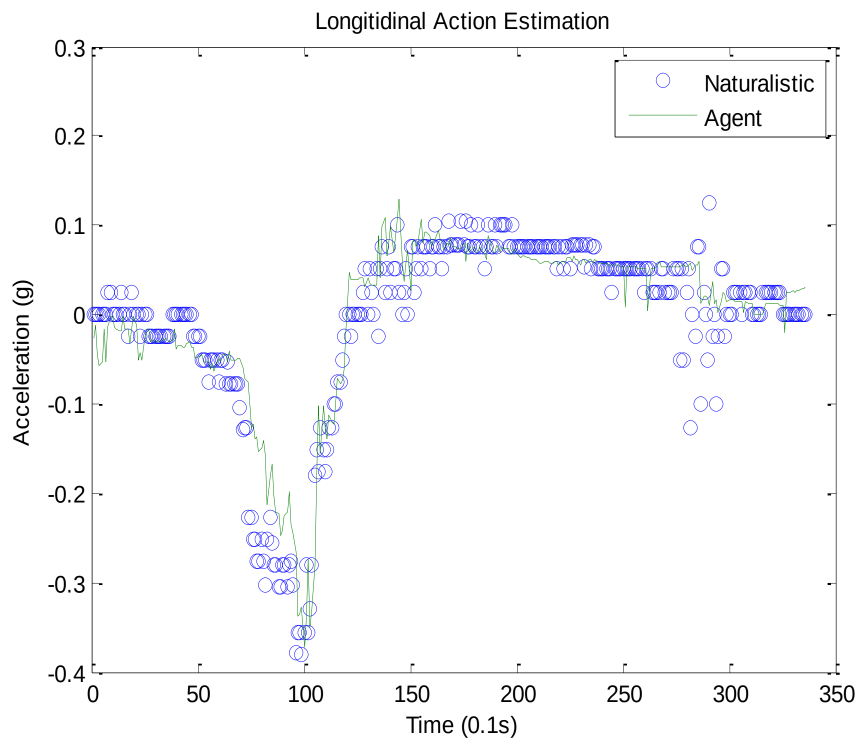

Figure 3-12 Acceleration of Mega Agent, Event A1

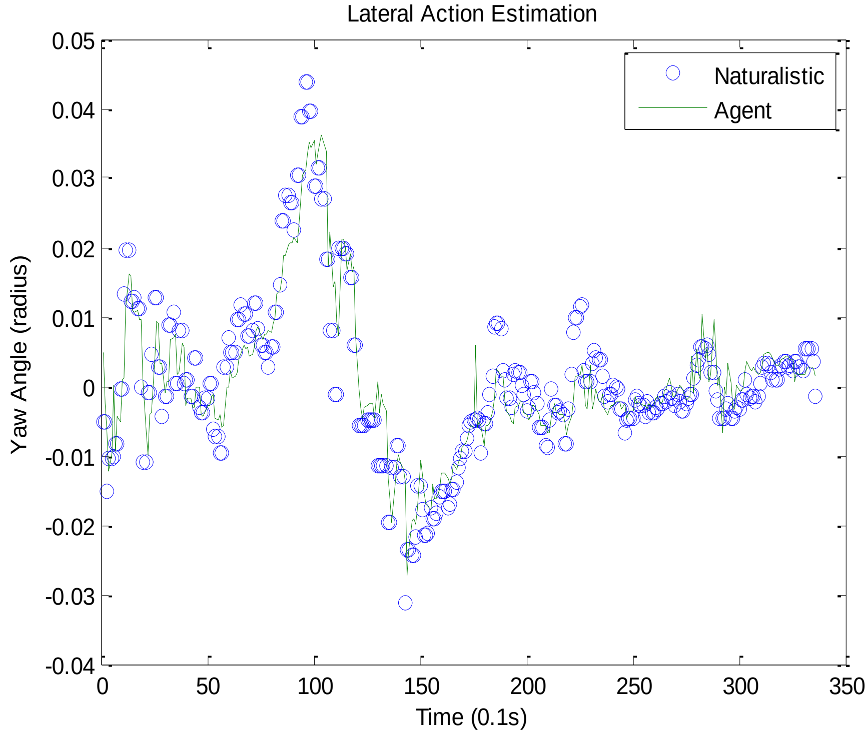

Figure 3-13 Yaw Angle of Mega Agent, Event A1

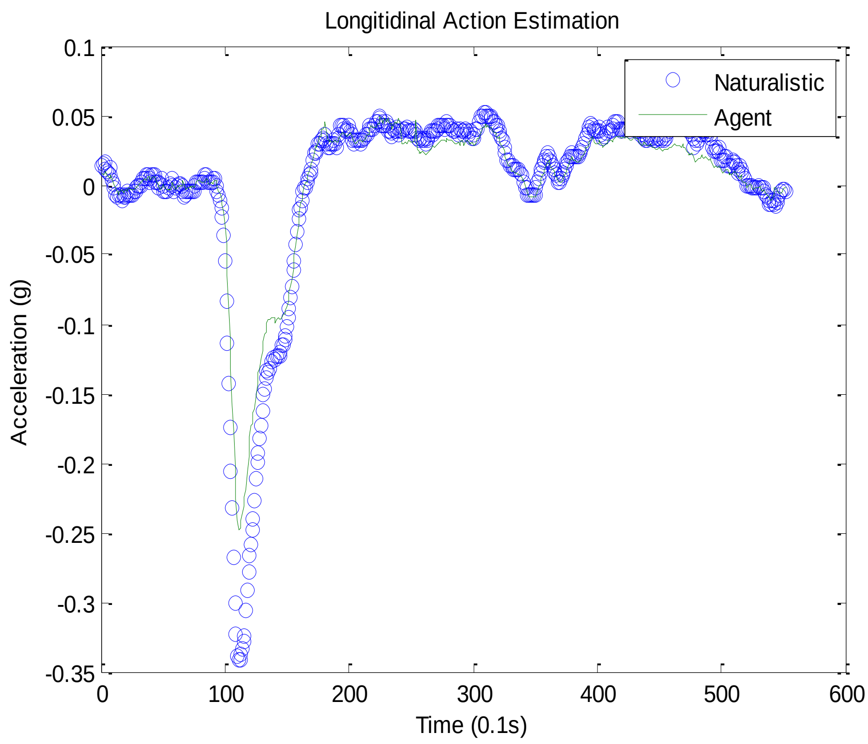

Figure 3-14 Acceleration of Mega Agent, Event B1

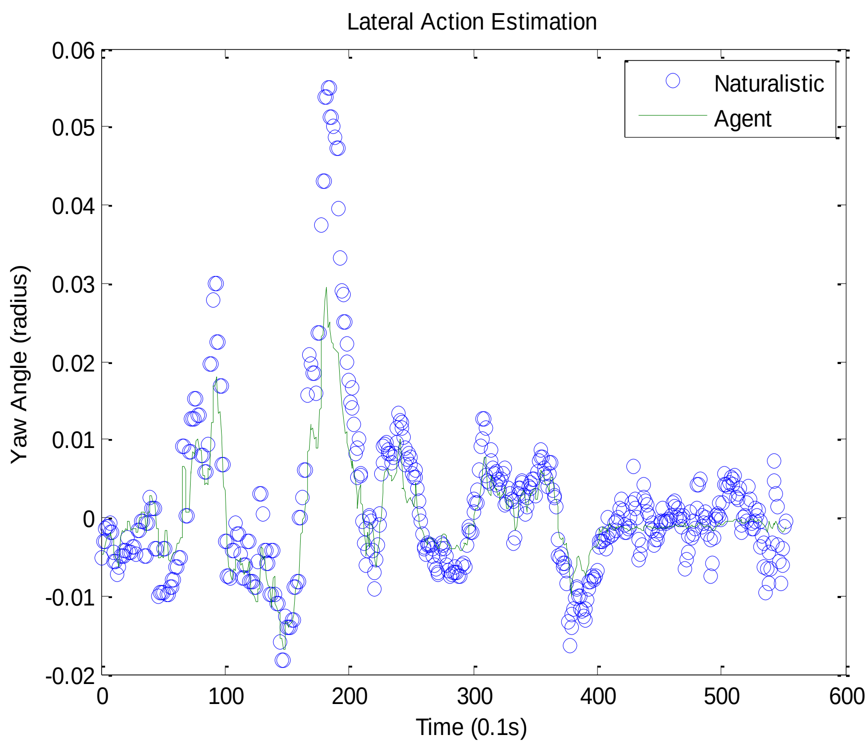

Figure 3-15 Yaw Angle of Mega Agent, Event B1

Compared to Figure 3-8, Figure 3-9, Figure 3-10 and Figure 3-11, the Mega Agent is capable to differentiate between Agent A and Agent B.

In reality, a conservative driver might never happen to experience safety critical events but an aggressive driver may have. In that case, the conservative driver has no idea about the actions to do to get out of the emergency regime if he/she gets into it in a sudden fashion. Through this Mega Agent training, the conservative driver will "learn" the crash avoidance actions from the aggressive driver and help him/herself to evade upcoming crashes.

Table 3-2 R squared values of the Mega Agent

Event	Agent A		Agent B		Mega Agent	
	Acceleration	Yaw	Acceleration	Yaw	Acceleration	Yaw
Driver A	0.98	0.97	0.81	0.83	0.98	0.95
Driver B	0.82	0.6	0.97	0.92	0.97	0.91

Table 3-2 shows the R squared values of the Mega agent. R squared values of the Mega Agent area are quite high although a little bit lower than those when training and validation use the same events. In fact, Mega Agent is capable of mimicking the behaviors of Driver A and B at the same time without losing much driver specificity.

3.5 CONCLUSION AND FUTURE RESEARCH

This paper developed a framework to simulate individual driver actions during safety critical events. An agent-based artificial intelligence machine learning technique Neuro-Fuzzy Actor Critic Reinforcement Learning (NFACRL) methodology was used to model driving behavior. Naturalistic Driving Database was used to train and validate driver agents.

The advantages of NFACRL lie mainly in its capability to simulate heterogeneous vehicle actions in complicated traffic state environment. It is worth mentioning that this research is an attempt to apply reinforcement learning technique in solving high dimensional state problems and continuous action simulation problem simultaneously in transportation research especially in traffic flow theory. From the perspective of microscopic traffic behavior modeling, the proposed methodology is able to simulate lateral action, which brings new insights in modeling driver maneuvering behavior during safety critical events. The proposed methodology also looks into events from different drivers and proves behavior heterogeneities.

This paper focuses on safety critical events. The next step of this research is the extension of the capability of NFACRL to simulate other traffic behavior, such as lane-changing behavior, driver merging behavior at the upstream and downstream of ramps and evacuation behavior. It would be interesting to model individual driver behavior or decision making process under these traffic conditions.

For the technical part of this paper, the training parameters speed, memory and scaling factor are fixed for all the events, but it would be interesting to see the performance of NFACRL under different combinations of these factor sets. As most of artificial intelligence methods, NFACRL is a heuristic approach so there is no guarantee that training results would be optimal. Therefore, training parameters play an important role to guide NFACRL to find a near optimal. During our test, we found out that performance of agents is very sensitive to the driver dependent training parameters discrete action sets and state bounds. Theoretically, it would be better if these parameters are pre optimized before training. However, since the only way to test agent performance is through NFACRL training, parameter optimization and training forms a cycle and computation time would be exponentially increasing. In our approach, we set the parameters based on statistical quartiles and yield good results. Future research will be focused on the optimization methodology for training parameters and improve agent performance.

4 SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

4.1 SUMMARY

This thesis is an innovative attempt to apply an artificial intelligence technique machine learning in the scope of microscopic traffic flow modeling. BP ANN and RL are proposed as two potential but beneficial machine learning approaches which could enhance simulation accuracy and help understanding driver behavior in traffic. Based on our preliminary efforts, BP ANN and RL have shown some promising results.

The major advantage of machine learning technique is to directly learn driving characteristics from naturalistic data without using a prior defined model. In fact, two methods used in this thesis do not require safety model constraints (safety distance for example) so that they are capable to capture any actions from drivers and replicate naturalistic safety critical events. Therefore, simulating more complicated behavior such as merging behavior, gap acceptance behavior and lane-changing behavior may become feasible in the future.

The motivation of this thesis to learn driver behavior from "target" actions, which are naturalistic acceleration and yaw angle. In fact, BP and NFACRL should be able to learn from any vehicle actions as long as they are available in the naturalistic driving database. For example, steering wheel degree can be modeled using BP. However, it is difficult to design a steering wheel degree model according to vehicle dynamics and human factor. Another motivation of NFACRL technique is to fill in the gap of current research on the lateral behavior when no current research efforts have reached possibly because of lack of events data.

It may worthwhile to compare the proposed BP neural network and NFACRL model to the state-of-the-art car-following models. Most of state-of-the-art car-following models are designed according to human factors and vehicle dynamics. The process of driver behavior simulation in most car-following models is the process of data fitting. Some parameters in car-following models have certain logical meanings and are restrained. In fact, most car-following models have limited parameters and the parameters do not have much freedom to change. As a result, when using car-following models to match naturalistic data, the degree of approximation may not perform well because of limited parameters and parameter constraints. Conversely, BP neural network and NFACRL model have a lot of parameters which directly come from naturalistic driving data without constraints. As a result, BP and NFACRL can perform better than state-of-the-art car-following models.

4.2 FINDINGS

Chapter 2 of this thesis presents the use of a Backpropagation (BP) neural network to simulate vehicle car-following actions determined by driving behavior. The simulated action of the developed neural network shows a close match on naturalistic data when neural network is under proper training. As expected, the performances of BP neural network rely on the quantity, accuracy and resolution of naturalistic data.

Chapter 3 of this thesis focuses on the use of Neuro-Fuzzy Actor Critic Reinforcement Learning approach to simulate complicated vehicle actions during safety critical events in traffic. The purpose of using NFACRL is its capability of dealing with continuous traffic state and continuous action mapping problems with a good degree of accuracy. This leads to our attempt to testify heterogeneities in different drivers under the "imaginary" same events.

Several figures in chapter 3 show some inaccuracies from NFACRL method. Naturalistic data itself, train process and train mechanism itself are supposedly three major reasons. First, from the experiments, we notice that NFACRL seems to work well when naturalistic actions are normal and continuous. According to the NFACRL logic, under the assumption that current actions are correlated to previous actions, current actions are supposed to be similar actions as previous ones. Therefore, a sharp break could not followed by an acceleration immediately. However, when naturalistic data is not continuous such as the last part of Figure 3-12, NFACRL will make biases. Second, the training process itself may also contribute to the inaccuracy. Take Figure 3-14 as another example, it is obvious that most of the naturalistic accelerations are around zero expect a sequence of decelerations from 10s to 15s. NFACRL treats acceleration and deceleration equally in training since both of them are shared by the 64 fuzzy rules. During the training process, NFACRL experiences more acceleration actions and would result in an "unbalanced training" where agent is not paying special attention to simulate negative deceleration. Third, the inaccuracies from training mechanism itself may result from the reward function. Accordingly to reinforcement learning algorithm, reward function is considered as a short term objective function. In this research, reward function uses an error acceptance term so NFACRL is allowed to make errors within a small amount and these errors may result in inaccuracies.

4.3 LIMITATIONS

Limitations of this thesis include two parts: the quantity and quality of naturalistic events data and the nature of machine learning technique itself.

Although NTDS provides huge amount of data in total, when it comes to individual driver data especially event data, limited number of events data can be ready to use. Notice that near crash and crash events in this study are actually rear-end conflicts, which is relatively rare from database. So far, for an individual agent training purpose, only two or three events can be found. This may result in a biased driver behavior.

Another limitation of event data comes from the data quality collected from radar. Range and range rate collection from radar only works when the leading vehicle is located within the detection zone. Radar works well in most of the car-following situations but can be problematic during safety critical events. Because vehicle swerved to execute a maneuver sometimes, the location of leading vehicle might become out of the radar detection zone and the radar would lose the target. In terms of radar range collection, if the radar loses the target, it returns with a range “zero” which is far from reality. Moreover, the merge movements of the leading vehicle in front cannot be represented in range data also. In reality, the following vehicle will choose to decelerate and “help” its predecessor to merge smoothly most of the time. However, radar cannot detect the merging vehicle until it is directly in front. So what happen in such case is that radar shows no vehicle in front but the following vehicle is decelerating rather than free driving. Radar does not tell the right story.

In terms of machine learning, both approaches used in this thesis are heuristic so no global optimal behavior is guaranteed. BP ANN seems to have been analyzed a lot and proved some robustness based on current knowledge while the understanding on reinforcement learning is still limited in literature. Limited literature discussed about the calibration on reinforcement learning parameters which could be applied in this thesis. In our special case, ten discrete action parameters reveal sensitivities on simulation results. In the proposed approach, parameters are chosen according to data distribution but may not result in the best fit on target actions. This limitation leads to the development of an optimization problem where discrete actions can be used as decision variables and sum squared error between NFACRL outputs and naturalistic puts could be the objective function. Training convergence is another limitation worth mentioning. Theoretically, learning stops when the differences of neuron weights become very small. However, it is difficult to tell whether it is a premature convergence or not. In this special case, using a large number of training iteration should avoid premature convergence but not in an efficient way.

4.4 RECOMMANDATIONS FOR FUTURE RESEARCH

Following the line of thought from chapter 4.3, an auxiliary optimization tool is suggested to enhance accuracy and reliability of our proposed reinforcement learning approach. Since genetic algorithm (GA) has proved to be an efficient away in learning parameter calibration, it is suggested to incorporate GA into the training process. Though iterations, GA should able to find the optimal parameters eventually.

The research scope of this thesis is simulating the behavior of the following vehicle. Any information of the leading vehicle is solely depends on radar. Machine learning is not used to extracting leading vehicle behavior. The future research is suggested to model leading vehicle actions simultaneously along with the following vehicle. It would be interesting to use multi-agent simulation methods such as game theory to capture the interactive behavior for both leaders and followers.

This paper uses two machine learning methodologies to simulate two types of driving behavior. It would be interesting to see their performances under the same situation. Analyzing driver heterogeneous behavior during car-following situations using more individual driver might be a good point to investigate. Since base on our preliminary research, only one individual driver was analyzed, therefore, using more driver data in both BP neural network training and NFACRL training may show the heterogeneities between different drivers. Additionally, the characteristic of the two proposed models such as parameter sensitivity and model robustness could be compared.

REFERENCES

1. Brackstone, M. and M. McDonald, *Car-following: a historical review.* Transportation Research Part F: Traffic Psychology and Behaviour, 1999. **2**(4): p. 181-196.
2. Gipps, P.G., *A Behavoural Car-Following Model for Computer Simulation.* Transportation Research Board, 1981. **15B**(2-C): p. 7.
3. Wiedemann, R., *Simulation des Strassenverkehrsflusses.* 1974.
4. Gazis, D.C., R. Herman, and R.W. Rothery, *Nonlinear Follow-The-Leader Models of Traffic Flow*. Operations Research, 1961. **9**(4): p. 545-567.
5. Kesting, A. and M. Treiber, *Calibrating Car-Following Models using Trajectory Data: Methodological Study*. 2008: p. 17.
6. Fritzsche, H.-T., *A model for traffic simulation simulation* Transportation Engineering Contrinution 1994. **5**.
7. Wiedemann, R. and U. Reiter, *Microscopic Traffic Simulation, the simulation.* 1992.
8. Rakha, H., C.C. Pecker, and H.B.B.C. Cybis, *Calibration Procedure for Gipps Car-Following Model.* 2007: p. 13.
9. Menneni, S., C.P.D. Sun, P.E., and P. Vortisch, *An Integrated Microscopic and Macroscopic Calibration for Psycho-Physical Car Following Models* TRB 2009 Annual Meeting CD-ROM 2008: p. 17.
10. Ossen, S. and S.P. Hoogendoorn, *Car-following Behavior Analysis from Microscopic Trajectory Data* 2004.
11. Hamdar, S.H. and H.S. Mahmassani, *From existing accident-free car-following models to colliding vehicles exploration and assessment.* Transportation Research Record, 2008(2088): p. 45-56.
12. Xin, W., et al., *The less-than-perfect driver: A model of collision-inclusive car-following behavior.* Transportation Research Record, 2008(Compendex): p. 126-137.
13. Ossen, S. and S.P. Hoogendoorn, *Validity of trajectory-based calibration approach of car-following models in presence of measurement errors.* Transportation Research Record, 2008(2088): p. 117-125.
14. Brockfeld, E., R.D. Kuhne, and P. Wagner, *Calibration and Validation of Microscopic Traffic Flow Models.* p. 9.
15. Olson, R., et al., *DRIVER DISTRACTION IN COMMERCIAL VEHICLE OPERATIONS*. 2009, Center for Truck and Bus Safety, Virginia Tech Transportation Institute: Blacksburg VA. p. 285.
16. Blanco, M., et al., *Investigating critical incidents, driver restart perios, sleep quantity, and crash countermeasures in commercial vehicle opertations using naturalistic data collection.*
17. Pipes, L.A., *Operational analysis of traffic dynamics.* Journal of Applied Physics, 1953. **24**(3): p. 274-281.
18. Forbes, T.W., *Human Factor consideration in traffic flow theory.* Highway Research Board Record, 1963. **15. HRB**: p. 60.
19. Gazis, D.C., R. Herman, and R.B. Potts, *Car-following theory of steady-state traffic flow.* Operations Research, 1959. **7**(4): p. 499-505.
20. Bernhardt, K.L.S., *Agent-Based Modeling in Transportation.* Transportation research Circular: Artificial Intelligence in Transportation, 2007.
21. Peeta, S., W. Zhou, and P. Zhang. *Modeling and mitigation of car-truck interactions on freeways*. 2004: National Research Council.

22. Hidas, P., *Modelling lane changing and merging in microscopic traffic simulation.* Transportation Research Part C: Emerging Technologies, 2002. **10**(Compendex): p. 351-371.
23. Bonabeau, E., *Agent-based modeling: Methods and techniques for simulating human systems.* 2002. **99**.
24. Jia, H., Z. Juan, and A. Ni. *Develop a car-following model using data collected by "five-wheel system"*. in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*. 2003.
25. Panwai, S. and H. Dia, *Neural Agent Car-Following Models.* IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 2007. **8**(1): p. 11.
26. Abbas, M., L. Chong, and A. Medina, *Neural-Agent Driver Behavior Modeling Using Naturalistic Data.* Proc., 2010 13th International IEEE Conference on Intelligent Transportation Systems, ITSC '10, September 19, 2010 - September 22, 2010, Institute of Electrical and Electronics Engineers Inc.
27. Sutton, R.S. and A.G. Barto, *Reinforcement Learning:An Introduction*. 1988, London, England: The MIT Press Cambridge, Massachusetts.
28. Jouffe, L., *Fuzzy inference system learning by reinforcement methods.* IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 1998. **28**(Compendex): p. 338-355.
29. Avineri, E. and J.N. Prashker, *Sensitivity to Travel Time Variability: Travelers' Learning Perspective.* Transportation Research, 2005. **Part C**: p. 27.
30. Bogers, E.A.I., M. Bierlaire, and S.P. Hoogendoorn, *Modeling Learning in Route Choice.* Transportation Research Record, 2007. **2014**: p. 8.
31. Abdulhai, B., R. Pringle, and G.J. Karakoulas, *Reinforcement learning for True Adaptive Traffic Signal Control.* Journal of Transportation Engineering, 2003. **May/June**: p. 8.
32. Jacob, C. and B. Abdulhai, *Automated adaptive traffic corridor control using reinforcement learning: Approach and case studies.* Transportation Research Record, 2006(1959): p. 1-8.
33. Bingham, E., *Reinforcement learning in neurofuzzy traffic signal control.* European Journal of Operational Research, 2001. **131**(2): p. 232-241.
34. Adam, Z.M., M.M. Abbas, and P. Li, *Evaluating Green-Extension Policies with Reinforcement Learning and Markovian Traffic State Estimation.* Transportation Research Record, 2009(No.2128): p. 9.
35. Abbas, M.M., et al., *Illustrative field configuration and evaluation of traffic-responsive control.* Journal of Transportation Engineering, 2009. **135**(Compendex): p. 591-599.
36. Chong, L. and M. Abbas. *Neuro-fuzzy actor critic reinforcement learning for determination of optimal timing plans*. in *13th International IEEE Conference on Intelligent Transportation Systems, ITSC 2010, September 19, 2010 - September 22, 2010*. 2010. Funchal, Portugal: Institute of Electrical and Electronics Engineers Inc.
37. Zhang, Y., Y. Xie, and Z. Ye, *Development and Evaluation of a Multi-Agent Based Neuro-Fuzzy Arterial Traffic Signal Control System*. 2007.
38. Jiang, J.S.R., C.T. Sun, and E. Mizutani, *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. 1997, Upper Saddle River, New Jersey: Prentice Hall.