Driver Car-following Behavior Simulation using Fuzzy Rule-based Neural Network

Linsen Chong, Montasir Abbas, Bryan Higgs, and Alejandra Medina

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

Montasir M. Abbas, Ph.D., P.E. (Corresponding Author)
Assistant Professor, Charles Via Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University
E-mail: abbas@vt.edu

Bryan Higgs
Graduate Research Assistant, Charles Via Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University
E-mail: bryan.higgs@vt.edu

Alejandra Medina
Senior Research Associate Virginia Tech Transportation Institute
E-mail: ale@vtti.vt.edu

ABSTRACT

This paper proposes a rule-based car-following model to simulate the driver decision process and model the associated vehicle longitudinal action in the car-following regime. In order 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 the associated traffic states. The driving rules are calibrated using vehicle trajectory data from the car-following situations of one driver. An artificial technique, reinforcement learning (RL), is applied in the fuzzy rule calibration. Vehicle longitudinal actions are estimated and used as the output of this model. The simulated vehicle actions are compared to the naturalistic data.

In the proposed methodology, the naturalistic traffic state and driving actions are extracted from the Naturalistic Truck Driving Study (NTDS) database provided by the Virginia Tech Transportation Institute (VTTI). Driving actions were recorded in instrumented vehicles that have been equipped with specialized sensor, processing, and recording equipment. Car-following situations are extracted by pre-defined criteria.

KEY WORDS: driving behavior, fuzzy logic, reinforcement learning, artificial neural network, car-following model, model calibration

INTRODUCTION

In the last fifty years, a large number of car-following models have been developed to capture driving behavior in traffic. When a vehicle is approaching a leading vehicle, it is interacting with the acceleration/deceleration action of a leading vehicle. Car-following models were constructed to represent the longitudinal behavior of drivers. Different car-following models have different bases. Most car-following models are based on a safety distance idea, where vehicles can always have a safety distance in order to avoid potential collisions [1].

Driver actions in car-following models are defined by pre-specified functions. Parameters of car-following models are calibrated by matching car-following model equations to field measured data. The calibration process typically has an objective function to minimize errors between the model estimation and field data. However, calibration methodology has a risk of causing biases. Optimal parameters may be logically meaningless even if they result in a minimum error.

The motivation of this paper is to model driver naturalistic actions during the car-following process through simulating driver decision policy which associates the observed traffic state to vehicle actions. The driving policy should be driver independent and consist of multiple rules that dominate different traffic states. Fuzzy logic is used to partition the traffic state space regime for these rules to look up. Reinforcement learning is used to determine the optimal action for rules. Fuzzy logic and reinforcement learning are coordinated to extract driving policies.

In our proposed methodology, the naturalistic traffic state and driving actions are extracted from Naturalistic Truck Driving Study (NTDS) database provided by the Virginia Tech Transportation Institute. Safety critical events from individual drivers are extracted based on the trigger of the events and car-following situations are also extracted as baseline references. Driving actions were recorded in instrumented vehicles that have been equipped with specialized sensor, processing, and recording equipment.

TRAFFIC STATES AND ACTIONS DURING CAR-FOLLOWING

It is assumed in this work that drivers behave according to the traffic. The traffic state is defined by a set of variables that can represent vehicle action kinematic conditions and its environment. Most car-following models suggest that vehicle speed, relative distance from leading vehicle, relative distance and acceleration of leading vehicle should be incorporated as traffic state variables. Additionally, as car-following behavior is a sequential episodic task; the actions taken earlier may affect the current actions currently. Because of the limitation of data measurement, it is assumed that the driver behaves only according to speed, relative distance, relative speed and acceleration.

One of the most widely used car-following models, the Wiedemann model[1, 2], uses the state space regime-based behavior framework. In the Wiedemann model, relative speed and relative distance space is divided into free driving, closing in, following and an emergency regime where the driver has different predefined acceleration equations in each regime. Drivers maintain their desired speeds in the free-driving regime where speeds are considered to be constant at the desired speed and no accelerations need to be taken. When a vehicle is approaching a predecessor, it is coming to the following regime where the driver reacts to its leader and makes a decision to accelerate or decelerate according to the distance, relative speed, and driving action of the leading vehicle. In the emergency regime when a driver anticipates an upcoming incident, the desired deceleration is taken to avoid upcoming conflicts.

CAR-FOLLOWING STATE AND ACTION PARTITIONING

We presume that as a human decision process, car-following actions would be closely related to traffic state inputs. Driving rules here provide the mapping policy that a driver should follow. Actions are associated with traffic state inputs by following the driving rules. However, traffic state space has a lot of dimensions. Even the most primary GM family models require at least three state variables. Accordingly, a dimensionality problem could be a great pain to most simulators. One more dimension in the traffic state will cause an exponentially increasing number of state-action mapping rules. Additionally, state and action variables in our study are continuously distributed, which makes it a challenge to build a look-up table that relate states to actions.

In order to construct our look-up table, fuzzy logic is used to partition the traffic state variables. In our purposed methodology, fuzzy logic partition sets and fuzzy driving rules are embedded in a neural network structure. Continuous traffic state variables are clustered into several discrete fuzzy sets so that a limited number of fuzzy diving rules are needed and look-up tables can be constructed. By using a neural network structure, a combination of fuzzy sets can be used as a preservative of traffic state as the input of fuzzy rules. By referring to the look-up table, a driver simulator can take certain actions according to the fuzzy rules.

PROPOSED FUZZY LOGIC BASED NEURAL NETWORK

A neural Network acts as a driver simulator in this study. As Figure 1 shows, our proposed Neural Network structure has four layers. The first layer is the input layer. Each node of this layer represents a continuous state variable. The second layer is the fuzzy membership layer. States are fuzzified in this layer into linguistic terms as “Speed is High” and “Speed is Low”. Each node is a fuzzy set associated with a membership function as output. Note that one state input variable should have more than one fuzzy set to transfer continuous input variables to discrete sets without losing much information. Fuzzy membership functions can be triangular, trapezoidal and Gaussian [3] depending on the distribution of the original continuous variables. The third layer is the fuzzy rules layer. Each rule is connected with a number of antecedents (discrete fuzzy sets) from the second layer. A firing strength is the output of the fuzzy rule that indicates the strength of the rule. The fourth layer is the discrete action layer including a set of

discrete actions for similar to choose. Each fuzzy rule selects one discrete action. The output simulated action is the weighted average of the selected actions where fuzzy rule strengths are the associated weights. Several following paragraphs elaborate more on our proposed layer design method.

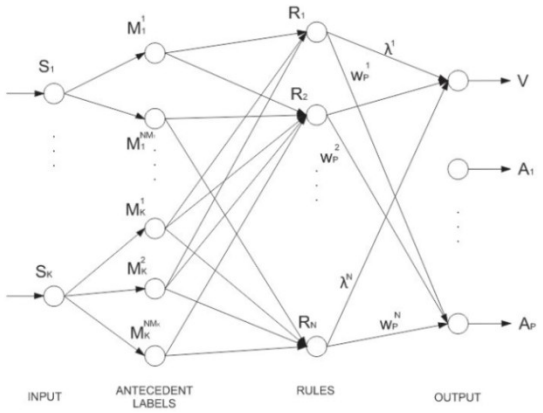

Figure 1.Neural Network Structure

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
w_q^j=weight between j^{th} fuzzy rule and action q
λ_j=weight between j^{th} fuzzy rule and critic
A_q =output of q^{th} discrete action
Where $i = 1, \dots n, a\ \ i\ \ = 1, .. NM_i, j = 1, \dots, m$ and $q = 1, .., P$

State Layer
Since most car-following models use space headway, speed and relative speed headway as traffic state, we keep these three variables as state variables. Also, as mentioned before, we believe that as a sequential task, the action of current state should be related to the action of previous state. We also incorporate previous acceleration action as a traffic state variable. Therefore, speed, space headway, relative speed, previous acceleration/deceleration are used as traffic state variables and defined in Equation (1). Because of the limitation of naturalistic data measurement, the acceleration of the leading vehicle is not collected by the current instrument. However, since acceleration is the derivative of speed, we consider using the relative speed as a surrogate measure of the leading vehicle acceleration.

$$S_1 = v$$
$$S_2 = \Delta x$$
$$S_3 = \Delta v$$
$$S_4 = a' \qquad (1)$$

where S_i is ith state variable, v is the vehicle speed, Δx is the space headway vehicle relative to its leading vehicle, Δv is the relative speed (speed of the leading vehicle minus the following vehicle), a' is the previous acceleration.

Fuzzy Set Layer

In this layer, for each state variable S_i, two linguistic terms are used in fuzzy sets definition as"S_i is high" and "S_i is low". A set of fuzzy membership functions needs to be defined in order to partition state variables. A triangular membership function is applied here. The upper and lower bounds of each state variable need to be pre-determined before using the triangular membership function. In this study, upper and lower bounds are extracted from car-following situations that an individual driver has experienced. For each state, the lower bound is the minimum value and the upper bound is the maximum value so no information is lost in the membership function transfer.

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

$$\mu_{Low}(S_i) = \begin{cases} 1 & S_i \le S_{lb,i} \\ \frac{S_{ub,i}-S_i}{S_{ub,i}-S_{lb,i}} & S_{lb,i} < S_i < S_{ub,i}\ i = 1,2..6 \\ 0 & S_i \ge S_{ub,i} \end{cases} \quad (2)$$

$$\mu_{High}\ S_i = \begin{cases} 0 & S_i \le S_{lb,i} \\ \frac{S_i-S_{lb,i}}{S_{ub,i}-S_{lb,i}} & S_{lb,i} < S_i < S_{ub,i}\ i = 1,2..6 \\ 1 & S_i \ge S_{ub,i} \end{cases} \quad (3)$$

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.

Fuzzy Rules Layer

Fuzzy rules in the third layer provide a state action mapping policy to determine which action to select for a certain state. Each fuzzy rule is associated with 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",
THEN Action "Deceleration $a = -0.5m/s^2$"

In our designed neural network, each fuzzy rule is associated with four fuzzy sets originating from four continuous state variables. Since each state variable has two fuzzy sets "Low" and "High", the number of fuzzy sets combination should be $2^4 = 16$. 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 thej^{th} fuzzy rule is

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

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

Action Layer

The action, acceleration, is also a continuous variable. Due to the constraints of the neural network structure, it is impossible to enumerate all the possible continuous actions. Instead, a discrete action set is defined here to include a number of representative action values for fuzzy rules to select. In this research, five discrete acceleration values are used in the acceleration set *A*. These discrete actions are considered to be the action candidates in our state-action mapping policy look-up table.

$$A = \{a_{d1}, a_{d2}, a_{d3}, a_{d4}, a_{d5}\} \tag{5}$$

We use lower quartile (25th percentile, cuts of lowest 25% of data), median (50th percentile, cuts of 50% of data), upper quartile (75th percentile, cuts off highest 25% of data) and the maximum.

Neuron Weights

Weights are located between the fuzzy rule layer and the action layer. Weight w_k^j connects the j^{th} fuzzy rule with the k^{th} action output. The action weight values w_k^j show competition between actions and are used for action selection methodology for the fuzzy rules. Since there are five discrete action candidates connected with each fuzzy rule, the fuzzy rule should choose the one which has the maximum weight. Weights are calibrated using a reinforcement learning algorithm. Another type of weights λ^j connect the j^{th} fuzzy rule with the next state. Weights λ^j are only serve as auxiliary parameters for w_k^j updates.

Action Output

For each fuzzy rule, one discrete acceleration in set *A* is selected. Since 16 rules are in the network, the output action, acceleration a, is the weighted average of all the selected discrete actions by all the rules. Firing strength is used as weight for the fuzzy rules so the output actions and generated as

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

where a is the continuous acceleration output.

NEURON WEIGHTS CALIBRATION AND UPDATE: REINFORCEMENT LEARNING

Reinforcement Learning: Brief Introduction

The objective of reinforcement learning algorithms is to find a policy that maps states to their optimal actions [4]. Driver simulator actions are reinforced when they perform approximately closer to goals and punished when are far away from the objectives. The only information available for learning is the system feedback, which describes, in terms of reward and punishment, the task the agent has to realize. 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 extracts driver behavioral rules from the naturalistic dataset and establishes driver specific state action mapping rules.

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

Weight Update

Action weights are updated through a reward function of the selected action a and the following state value $V_{s_{t+1}}$. Assume S_{t+1} is the following state after state S_t when actions a is taken at time t. Value V_{t+1}should be

$$V_{s_{t+1}} = \sum_{j=1}^{N} FS_{t+1,R_j}\lambda_j \tag{7}$$

Temporal difference (TD) error [4] is calculated as

$$\delta_t = r_{t+1} + \gamma V_{s_{t+1}} - V_{s_t} \tag{8}$$

Where r_{t+1} is the reward function of action a taken at state S_t, γ is the discounting factor.

Temporal difference error updates all the weights λ.

$$\lambda_{t+1} = \lambda_t + \beta\delta_t FS_{s_t} \tag{9}$$

For the j^{th} fuzzy rule, temporal difference error only updates the weights connected to the selected discrete actions.

$$w_{k,t+1}^{j} = w_{k,t}^{j} + \beta\delta_t FS_{s_t} \tag{10}$$

where β is the learning rate

After the simulator executes actions $a_t$at state S_t, the Neural Network receives a punishment or reward from the reward function $r_{t+1}$and estimates state S_{t+1}. The Neural Network updates λ and acceleration weights w_k^j according to equations (9) and (10).

Reward Function

The reward function provides guidance for a driver simulator to follow. The reward function encourages the simulator to take actions which are close to the true driver actions and penalizes actions which are far away from the true driver actions. The simulator will know the outcome of the actions only after they have been chosen. When an action outcome is close to the driver's action, the reward function is set to positive to increase the probability for the agent to choose similar actions in the future. On the contrary, when the performance is bad, the reward function should be negative.

The relative error is calculated as

$$e = \left| \frac{a - a_n}{a_n} \right| \quad (11)$$

where e is the absolute relative error of acceleration, a_n is the naturalistic actions from the Naturalistic database.

A non-negative training parameter e_{th} is defined as the acceptance threshold. When e is less than e_{th}, RL encourages simulator to take action a by using reward function:

$$r = \alpha(e_{th} - e) \quad (12)$$

where r is the reward function of acceleration and α is the scaling factor.

NATURALISTIC DRIVING DATA

For this research, to test our proposed method, we used data from the 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 [5], 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 measured vehicle trajectory data includes speedometer output, longitudinal and lateral accelerations, yaw angle, heading, indications of turning signal, brake and accelerator. For the leading vehicle information, range, range-rate and azimuth were collected by instrumented forward viewing radar from the following vehicle. Both the leader and following vehicle data were recorded at 10Hz.

Identification and extraction of car following situations

Car-following situations were automatically extracted from the enormous volume of driving data in the database in order to analyze the car following driver behavior. The filtering process is an iterative process where initial values and conditions are used and after the events are flagged they are 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 non 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 70 mph

- -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 are not misidentified as being in the same lane as the subject vehicle when roadway curvature is present.

- Length of car following period >= 30 seconds

The automatic extraction process was verified from a sample of events through video analysis. For the random sample of 50 periods, all 50 were valid car following periods.

EXPERIMENT

In our preliminary efforts, one car-following situation from one driver from the NTDS study was selected with the assumption that the car-following behavior of one driver does not change in different car-following situations.

Before training, errors from data collection should be excluded. We arbitrarily set up additional constraints to make sure that state data are plausible as any data outside these criteria would have questionable validity.

- Speed>=0 km/h
- Range>=0 feet and Range<=400 feet (when there is no leading vehicle in front, we assume Range=400 feet)
- Range Rate>=-10 feet/second

Also, the training parameters' upper/lower bounds of fuzzy sets and the discrete action set values are determined from the database. In this experiment, we use the minimum and maximum value from the selected naturalistic training state variables as lower and upper bounds. Discrete action set variables are selected based on quartiles from naturalistic data as mentioned before. Notice that the filtering thresholds defined above are just used to filter out the errors in measurement. The thresholds for fuzzy sets are directly loaded from the naturalistic car-following situations.

In our designed training process, at one time step of one event, the fuzzy rules scan their associated weights, select the optimal actions and update the weights. The neural network keeps updating the weights from the beginning of the events until the end. In fact, weights are trained and updated by 10 times the length of events (10Hz resolution data) and one iteration is finished. Theoretically, when the differences of critic/actor weights between two consecutive iterations become very small, the training process is considered to be finished. However, reinforcement learning is a heuristic methodology so no global optimal agent behavior is guaranteed. In fact, the convergence of the weights may be premature and result in a local optimal. To avoid this premature convergence, we tried 400 iterations in training. In the selected car-following situation, the simulator has around 1000 timing steps. Accordingly, fuzzy rules have been trained 1000*400=400000 times, which was deemed to result in a near optimal approximation of the driving behavior.

When six traffic state variables are loaded into the neural network, the values of all the fuzzy state membership functions are assigned and firing strengths of all the 64 rules are determined. Since a fuzzy rule selects one discrete action, all the membership functions and all 64 rules are visited during one iteration.

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 gets the new information. α controls the magnitude of the reward function and weights and e_{th} controls the sign of the reward function. In this test,$\beta = 0.6$, $\gamma = 0.9$,$\alpha = 10$ and $e_{th} = 0.2$.

PRELIMINARY TRAINING RESULTS

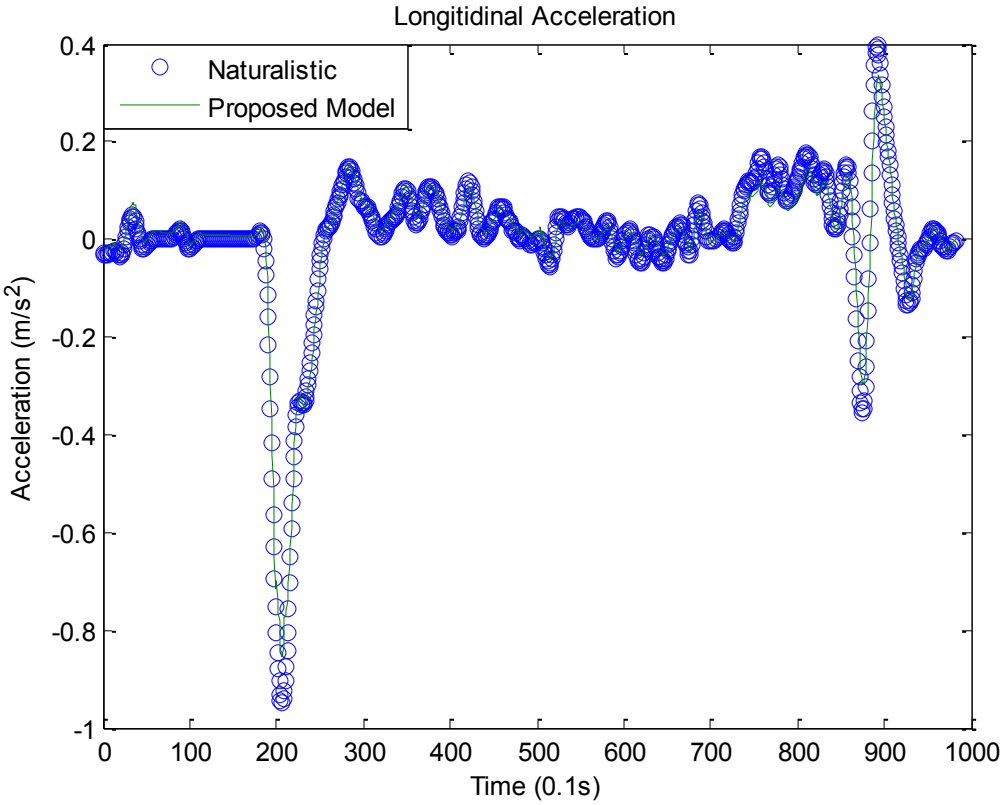

Figure 2. Acceleration of Neural Network versus naturalistic data the selected car-following period

Figure 2 shows the longitudinal neural network simulator action of acceleration during our selected car-following situation. The blue scatter plots represent the naturalistic driving actions and the green curves show the agent actions. The simulator captures driver naturalistic behavior quite well even when the driver had two sharp deceleration periods and the acceleration was not smooth.

Figure 2 is an example of training results for an individual driver by using a limited number of car-following situations based on our preliminary effort. Since the training parameters are directly determined by data from training sets, they should be different if using different situations in training. We suggest that when selecting car-following situations, the selected situations can cover the whole state space so the effect of the biased parameters could reach the minimum. In Figure 2, we just show that state-action mapping rules actually work by using the training dataset. In terms of validation, traffic state variables of the validation situations should not be out of the range of the training state space to avoid the possibility of the neural network running into singularities.

CONCLUSIONS AND FUTURE RESEARCH

In this paper, we proposed a fuzzy rule based neural network approach to model the driver's decision process during car-following situations. Fuzzy logic is used to partition traffic state variables and a reinforcement learning technique is used in the fuzzy rule policy calibration and update process. We used naturalistic, individual driver car-following data in driving rule training. Simulated acceleration is compared with naturalistic acceleration in our selected car-following period. Our preliminary results show that neural network is able to capture driver behavior quite well.

This paper is an attempt to use artificial fuzzy rules to simulate driver car-following behavior. The main motivation is to extend the capability of fuzzy rules so that they should be able to deal with continuous state-action mapping problems. By using a limited number of fuzzy rules and limited number of discrete action parameters, the dimensions of a state-action mapping table is significantly decreased which could be a release of computational burden in continuous state-action mapping problems.

It is worth mentioning some limitations of our proposed methodology:
(1) In this research scope, the comparison of our proposed fuzzy-logic model and the established car-following models has yet to be discussed. From Figure 2, our proposed methodology shows a good fit on naturalistic data but with the expense of millions of iterations and more calibration parameters. It is worthwhile to analyze the performances of car-following models using the same sets of data.

(2)

The next step in this research could be to test the capability of the proposed fuzzy rule based car-following model using more car-following situations from the same driver. In this paper, we are assuming that the car-following behavior of the same driver is not changing, which could be biased. Comparison to more car-following models might also be a good way to validate our proposed methodology as well as to test the robustness and viability of the technique. Also, heterogeneity of different drivers is something that is worth looking at. Our proposed model may bring some new insights to represent driver heterogeneous behavior in traffic.

ACKKNOWLEDGMENT
This material is based upon work supported by the Federal Highway Administration under Agreement No. DTFH61-09-H-00007. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the Author(s) and do not necessarily reflect the view of the Federal Highway Administration.

The authors would like to express thanks to Dr. C. Y. David Yang, the FHWA Agreement Officer's Technical Representative, for his continued support and guidance during this project. They would also like to thank individuals at Virginia Tech and the Virginia Tech Transportation Institute who contributed to the study in various ways: Greg Fitch, Shane McLaughlin, Brian Daily and Rebecca Olson.

REFERENCES
1. Wiedemann, R., *Simulation des Strassenverkehrsflusses.* 1974.
2. Wiedemann, R. and U. Reiter, *Microscopic Traffic Simulation, the simulation.* 1992.

3. 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.
4. Sutton, R.S. and A.G. Barto, *Reinforcement Learning:An Introduction*. 1988, London, England: The MIT Press Cambridge, Massachusetts.
5. 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.