Naturalistic Longitudinal and Lateral Risk-taking Driving Behavior Modeling during Safety-Critical Events

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

Driving behavior in traffic has been modeled quite successfully in simulation software using predefined car-following model 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 driving conditions. Also, detailed lateral maneuvering has not been simulated in most simulation software. The proposed methodology in this paper focuses on establishing a traffic state-action mapping rule to simulate real driver actions including risky behavior that a driver would take during safety critical events instead of the predefined actions given by car-following models. Traffic states are defined by the variables that are influential to driver action such as relative distance, speed or the acceleration of leading vehicle with the purpose of finding the most critical causalities of the events. State-action mapping rules are calibrated and validated using artificial neural networks.

KEY WORDS: safety-critical events, regression analysis, artificial neural network, car-following model, driving behavior

INTRODUCTION

Drivers react differently to different traffic situations. Driving behavior is a representation of a driver's driving objectives and patterns. For example, the objective of a driver while free-driving is to maintain the desired speed, while the objective in of a driver in emergency situations is to avoid incoming conflicts. Most car-following models are formulated by using different vehicle action equations when the driving objectives are different.

Driver actions in car-following models are defined by pre-specified rules. These rules are mostly interpreted by relating the traffic state situation that a driver observes to his/her applied response. Different car-following models consider different criteria in traffic states as causalities which stimulate the driver's reactions. However, in reality, these rules specified by car-following models may not necessarily capture naturalistic driving behavior due to the complexity and instability of the human decision making process. For example, in the emergency driving regime, longitudinal decelerations derived by most car-following models guarantee that the following vehicle will always keep a minimum safety distance from its leading vehicle, thus car-following models are not able to simulate crash and near crash events.

The motivation of this paper is to construct an agent-based model to simulate vehicle naturalistic actions in traffic, especially during safety-critical events. To achieve this goal, individual driver behavioral rules are extracted from the naturalistic data and used in agent training so that agent should be capable of replicating a driver like a clone. Because Artificial Neural Network (ANN) has shown some robustness and variability in driver behavior modeling [1, 2], we applied a special type of ANN Backpropagation (BP) to train agents. We also used the Wiedemann model as a comparison to our proposed methodology.

In our 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). 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.

DRIVER BEHAVIOR AND STATE SPACE REGIME IN CAR-FOLLOWING MODELS

In the last fifty years, a considerable amount of research has focused on modeling longitudinal vehicle actions in traffic, producing a large number of car-following models. One of the most widely used car-following model Wiedemann model[3, 4] uses the regime-based behavior idea. In Wiedemann model, relative speed and relative distance space is divided into free driving, closing in, following and emergency regime where driver has different predefined acceleration equations in each regime. Drivers maintain their desired speeds in free-driving regime where speeds are considered to be constant and no accelerations need to be taken. When a vehicle is approaching a predecessor, it is coming to the following regime where driver reacts to its leader and makes decision to accelerate or decelerate according to the distance, relative speed and driving action of the leading vehicle. In the emergency regime when driver anticipates an upcoming incident, evasive actions breaking should be taken to avoid upcoming events.

In the Wiedemann model, the emergency regime is named the *Danger* regime. The Wiedemann model assumes that drivers will always keep a safety distance at any time and can always stop before hitting the leading vehicle. When the distance to the leading vehicle is smaller than the risky distance (a prespecified threshold) ABX, the following vehicle uses its max deceleration, b_{min} to extend the headway [3, 5]. ABX and b_{min} are calibration parameters and are driver dependent. As long as a vehicle falls into the danger regime, it will always take the maximum deceleration.

SHORTCOMINGS OF CAR-FOLLOWING MODELS DURING SAFETY CRITICAL EVENTS

From the naturalistic driving database, we noticed that safety critical events defined by VTTI are not necessarily located in the emergency regime but can overlap with the car-following regime. In fact, the threshold between the car-following regime and the emergency regime can be ambiguous. When a driver is braking, the space headway to its leader is greater than the Wiedemann's threshold, and the deceleration is not as great as Wiedemann's. By using a smaller deceleration started at an earlier point before the upcoming conflict, the vehicle can reach the collision avoidance goal but without following Wiedemann's thresholds and equations.

Another finding is related to vehicle actions and traffic states. As mentioned before, driving behavior during safety critical events could be a complicated behavior. In this regime, the goal of the driver is to avoid crash, so a driver could hit a brake or control the steering wheel to execute a maneuver or both. From the naturalistic driving database, we notice that in most of the cases, vehicles were not only taking longitudinal action deceleration, but also taking lateral actions maneuvering simultaneously. Both actions worked together to avert potential conflicts. For example, a driver may take a maneuver and execute a lane change simultaneously while braking. Traffic states in this study can be represented by the condition of the following vehicle as its surrounding traffic environment including the information of its leader. Also, different from car-following regimes, actions in safety critical events are sequential tasks which mean actions at previous states can affect the action at current and following states. For example, when a driver decides to decelerate in the previous state, he/she is more likely to continue decelerating instead of executing a maneuver. Therefore, we think previous actions should be incorporated as the state variables. Because traffic stimuli and causalities sometimes can be very complicated and vary a lot, it is very hard to establish predefined longitudinal and lateral action models. Most existing car-following models tend to overlook the importance of lateral actions and complicated state action relations. So far, no lateral action models have been integrated in car-following models to simulate actions in safety critical events.

PROPOSED APPROACH: AGENT-BASED MODELING

Agent-based modeling (ABM) is a relatively new paradigm for exploring the behavior of complex systems [6]. Within the transportation domain, ABM is particularly good at modeling systems in which human decision making and action is a critical component. Existing ABM studies in driver behavior simulation include driver response to incidents, interaction between cars and trucks, driver behavior approaching a work zone, etc [7-9]. Bonabeau [10] 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 another than existing car-following models.

Driver individual behavior rules should be used to teach agent to learn. Behavior rules associate actions with traffic state provide a driver-specific driving policy for its agent to follow. So when an agent experiences a certain traffic state, the policy will map the traffic state to associated actions. By using naturalistic driving data in training, agents will learn to adopt driving rules in the training procedure and should be capable of replicating driver 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.

ARTIFICIAL NEURAL NETWORK

Artificial Neural Network (ANN) is a widely used learning algorithm in agent training, especially in supervising learning. In supervised learning, a set of examples pairs are given and the objective of learning algorithm is to find the mapping functions between given inputs and outputs example. Sample outputs evaluate the goodness of fit of mapping rules and provide feedback for learning algorithm to adjust mapping rules. In this paper, since naturalistic driving database can be utilized as the sample traffic states and naturalistic actions, ANN supervised learning can be used to extract driving rules and train agents.

Neural networks have been applied in car-following model design. Jia et al [11] designed a car-following model using vehicle speed, relative speed, relative distance and driver type to in ANN training. Panwai and Dia [1] tested several types of ANN to in car-following behavior simulation. Although ANN seems to have a good approximation on car-following behavior according to preliminary results, the traffic state variables, the traffic regimes and driver types were somehow arbitrary selected and may cause biases. Also, existing approaches are dealing with car-following behavior which is relatively easy to simulate rather than the complicated behavior in safety critical events.

In this paper, we select a general-purpose network paradigm backpropagation (BP) neural network as the ANN type for training. BP has a relatively simple structure with fast computational speed but shows some robustness when calibrating dynamic behavior such as the sequential car driving task. Moreover, according to its nature, BP can adapt to driver behavior change and have a prediction on driver action when action data are not sufficient; it could be interesting to test the goodness of vehicle longitudinal and lateral action estimation in our problem.

BP ANN DISCRIPTION

In our proposed agent-based neural network model, the driver agent observes the traffic state (its environment) and reacts, which is very similar to the decision process of human driver. BP ANN does not need a pre-specified function to associate states with actions but requires the sample state-action pairs as references to construct mapping rules. BP ANN provides a non-linear approximation method based on example outputs from the training data sample. With a set of limited input and output training data, BP ANN is capable to provide state-action mapping rules for the whole state space even when some state patterns are not provided in training data.

As the name tells, BP is a propagation of error. BP neural network calculates the error between desired output and actual sample output and propagates error back to each neuron in the network. Network weights between layers are updated through training until the propagation of errors become relatively small and weights value converges. BP neural network follows a gradient descent algorithm learning rule to gradually approximate driver actions using the input data (traffic state) and target data (actual actions) from the training episodes.

Backpropagation neural network is composed of an input layer, hidden layer(s) and an 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:

$$\begin{bmatrix} s_{1,k} \\ s_{2,k} \\ . \\ . \\ s_{m,k} \\ . \end{bmatrix} = \begin{Bmatrix} h\left[\sum_{i=1}^{n_{k-1}} w_{i,1}^{k} s_{i,k-1} + b_1^k\right] \\ . \\ . \\ . \\ h\left[\sum_{i=1}^{n_{k-1}} w_{i,m}^{k} s_{i,k-1} + b_m^k\right] \\ . \end{Bmatrix} \quad (1)$$

where n_k is the number of neurons in *k*th layer $s_{m,k}$ is the value of the *m*th hidden neuron, $w_{i,m}^{k}$is the weight connecting the *i*th input neuron and the *m*th hidden neuron in the *k*th layer, b_m^k is the bias for the *m*th hidden neuron in the *k*th 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:

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

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

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_l \\ . \end{bmatrix} = \begin{Bmatrix} h\left[\sum_{i=1}^{n_k} w_{i,1}^{o} s_{i,k-1} + b_1^{o}\right] \\ . \\ . \\ . \\ h\left[\sum_{i=1}^{n_k} w_{i,l}^{o} s_{i,k-1} + b_l^{o}\right] \\ . \end{Bmatrix} \tag{3}$$

where $w_{i,l}^{o}$ is the weight connecting the ith hidden neuron to the lth output neuron, and b_l is the bias for the lth hidden neuron. A linear transfer function is used to transform from the last hidden layer to the output layer. The BP neural network structure is shown in Figure 1.

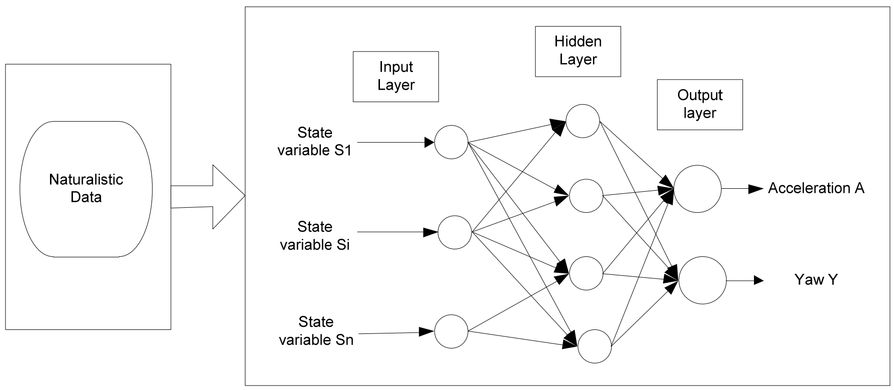

FIGURE 1. BP ANN Structure

Backpropagation learning algorithm is divided into two phases: propagation and weight update. 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 neural network using the target output to generate the gradients of all output and hidden neurons. In the weight update phase, the output delta 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 can be written as:

$$W_{k+1} = W_k - \alpha_k g_k \tag{4}$$

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). The neurons of the last layer are the output of the transform function.

In agent training, as mentioned before, we use speed, space headway, relative speed, previous acceleration/deceleration, previous yaw angle and lane offset as traffic state variables for the BP

ANN input layer (See Equation (5)). We use acceleration and yaw angle as representatives of vehicle longitudinal and lateral actions for the output layer (see Equation (6)).

$$
\begin{aligned}
S_1 &= v \\
S_2 &= \Delta x \\
S_3 &= \Delta v \\
S_4 &= a' \\
S_5 &= y' \\
S_6 &= o
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
A_1 &= a \\
A_2 &= y
\end{aligned}
\tag{6}
$$

where S_i is ith state input variable, A_i is the ith output action 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, y' is the previous yaw angle (the angle between vehicle longitudinal axis and lane marking), $o$is the vehicle lateral position offset (vehicle location relative to the center of the lane), a is the acceleration associated with the current state and y is the current yaw angle.

NATURALISTIC DRIVING DATA

To test our proposed method, we used data from Naturalistic Truck Driving Study (NTDS) collected by VTTI. 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), 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.

EXPERIMENT

In our test, the following vehicle is the instrumented vehicle. The measured vehicle trajectory data include speedometer output, longitudinal and lateral accelerations, yaw angle, heading and indications of turning signal, brake and accelerator. For the leading vehicle, range, range-rate and azimuth were collected by instrumented forward viewing radar from the following vehicle. Speed collected from speedometer, range and range-rate from radar, yaw angle and lane offset extracted from video recording were used as traffic state variables. Acceleration from the accelerometer was used as longitudinal traffic action and yaw angle was used as a lateral action. Although yaw angle is not strictly a lateral action variable, however, we found out that the gyro (vehicle angular speed) data was too noisy, so we considered yaw as an equivalent lateral action. The safety critical events were identified and analyzed in a previous work by VTTI[12]. The method used to identify the safety critical events were triggers or thresholds on individual variables that were collected. 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

In our preliminary efforts, one driver from the 8 Truck Study was selected since the 8 Truck Study is the latest study and provided the most accurate data. We use all the safety critical events available for sufficient training purpose to avoid biases. Although conditions and casualties of different events can be totally different, BP ANN theoretically should still capture rules of individual driver quite well because different events are located at different state space with little overlaps.

Before training, errors from data collection measurement should be excluded. We arbitrarily set up additional constraints to make sure that state data are plausible.

- 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
- Offset<=72 inch and Offset>=-72 inch

BP ANN TRAINING RESULTS

The training dataset are randomly divided into three sets with 60% used to train the network, 20% to validate how well the network generalized and the rest 20% provide an independent test of network generalization for data that the neural agent has never seen. Figure 2 shows the performance function of the neural agent training data using all the available data points. Starting at a large value, the performance function decreases to a smaller value through the training process. It shows that the neural agent is learning. Training stops when the number of iterations, the performance function, the magnitude of gradient and the training time are greater or less than the predefined threshold values. Also, validation stops when the validation error increases. Training continues as long as the validation errors continue to reduce.

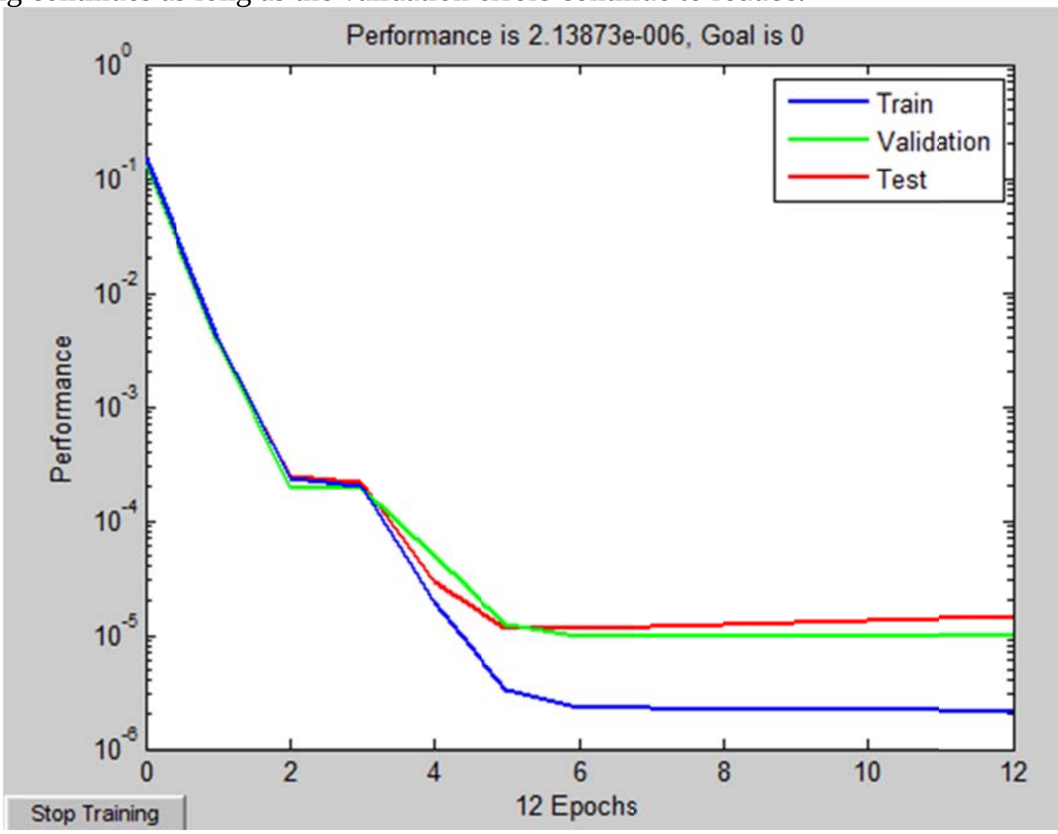

FIGURE 2. BP ANN Model Training, Validation and Test

Figure 2 shows that the training stops after 12 iterations when the network generalized the best solution for validations. But for the limitation of training data, the safety critical events selected are near crash events and it is not easy to determine the required event numbers to prevent pre-mature convergence. In this paper, we used all the event data available in training with the assumption that all the data were used in training which are around 900 data points with 10Hz resolution. From the perspective of BP ANN, the more data BP uses per iteration, the less number of iterations needed to converge. In our experiment, the training process does not require much iteration to converge. From Figure 4, since the test error and validation error have similar characteristics and no significant overfitting occurred, the training could complete but not for premature convergence.

We picked up one near crash event (34 seconds) as an example to evaluate and validate the performance of the BP ANN. Figure 3 and Figure 4 represent the simulated longitudinal and lateral actions results (blue scatter plots) compared to naturalistic actions (red line). Figure 3 shows that the vehicle experienced a sharp deceleration from to around and then returned to normal driving situation. In terms of car-following models, this vehicle would probably experience one than one driving regime and the accelerations are less likely to have a smooth and continuous profile. However, BP is able to capture both vehicle longitudinal and lateral actions quite well.

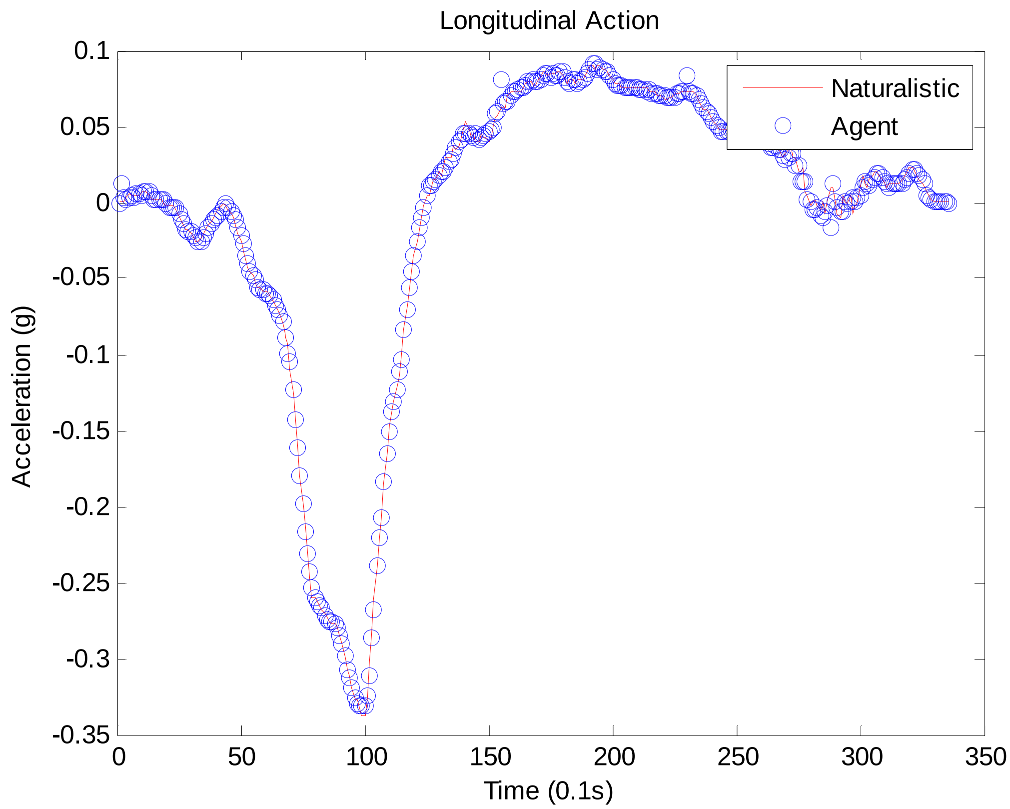

FIGURE 3. BP ANN Longitudinal Acceleration Validation Performance

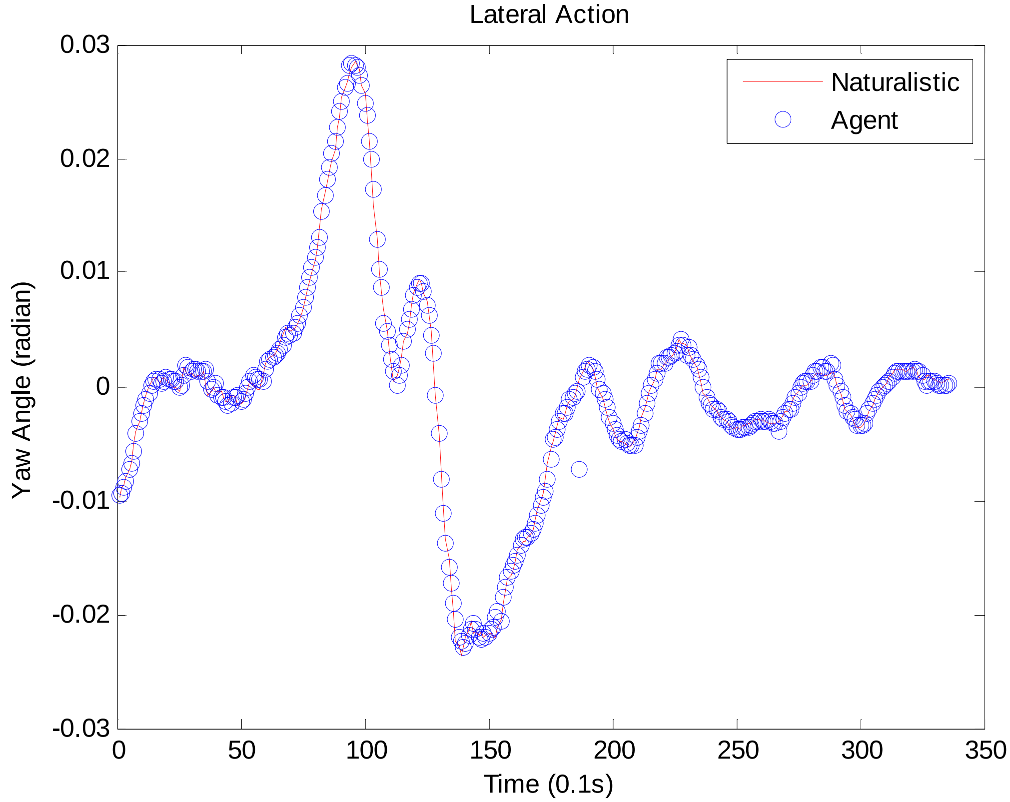

FIGURE 4. BP ANN Lateral Yaw Angle Validation Performance

SIMULATED TRAFFIC STATE TRAJECTORY BY BP ANN

From the output actions of BP ANN and the current state, the longitudinal speed and lateral offset trajectory can be simulated following basic kinematic equations.

$$v_t = v_{t-1} + a_{t-1} * \Delta t \tag{7}$$
$$o_t = o_{t-1} + v_{t-1} * \Delta t * \sin(y_{t-1}) \tag{8}$$

where v_t is the vehicle speed at time t, o_t is the vehicle lateral location offset at time t, a_{t-1} is the simulated acceleration at time $t-1$, y_{t-1} is the simulated yaw angle at time $t-1$, and Δt is the length of time step (0.1s)

Figure 5 and Figure 6 represent the reconstructed speed and offset trajectory according to outputs from BP ANN. Simulated speed and offset are very close to naturalistic data which means that the BP ANN neural agent performs quite well under the support of sufficient naturalistic data.

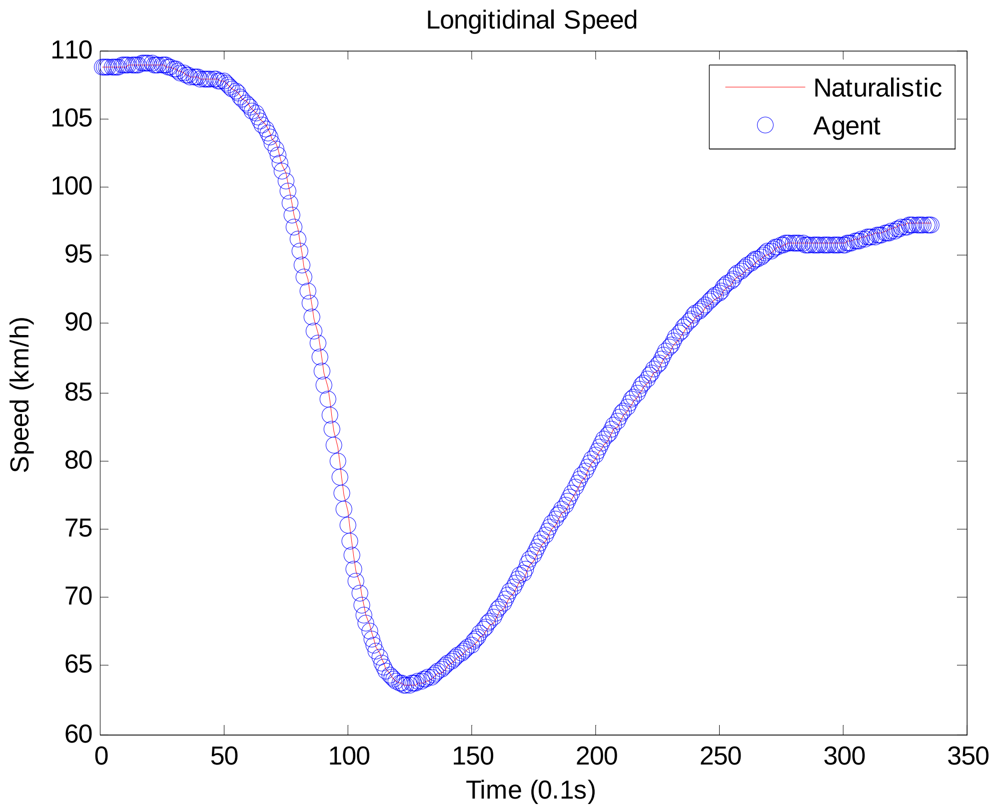

FIGURE 5. BP ANN Speed Trajectory

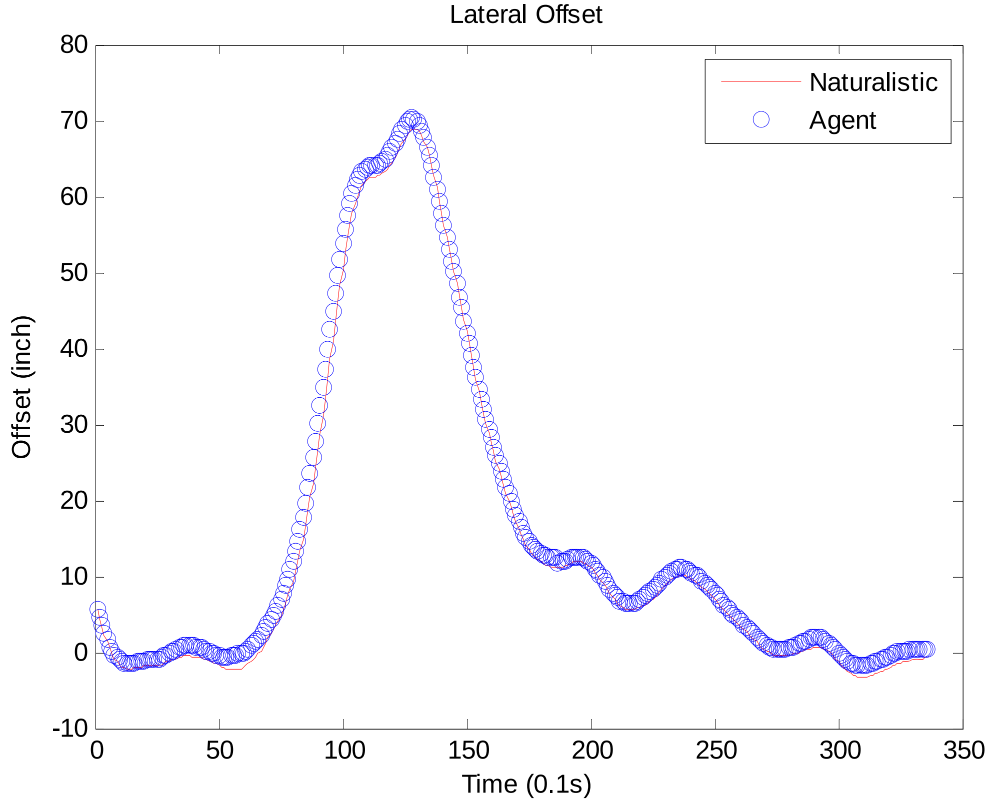

FIGURE 6. BP ANN Lateral Offset Trajectory

WIEDEMANN MODEL CALIBRATION FOR SAFETY CRITICAL EVENTS

As a comparison to our proposed methodology, the existing Wiedemann model is also calibrated in this study. To simplify the problem, for this event example, we only consider two regimes: emergency regime and car-following regime. As defined by Wiedemann, when space headway is less than a predefined threshold ABX, the vehicle is in the emergency regime. Or else, the vehicle is in the car-following regime. In the emergency regime, the acceleration of the following vehicle in order to avoid a collision is:

$$b_n = \frac{1}{2} * \frac{(\Delta v)^2}{AX-(\Delta x - L_{n-1})} + b_{n-1} + b_{min} * \frac{ABX-((\Delta x - L_{n-1}))}{BX} \tag{9}$$

$$b_{min} = -BMINadd - BMINmult * RND + BMINmulit * v_n \tag{10}$$

where b_n is the deceleration of the following vehicle, b_{n-1} is the deceleration of the leading vehicle, Δv is the relative speed, Δx is the relative distance, L_{n-1} is the length of vehicle, $ABX, AX, BX, BMINadd$ and $BMINmult$ are calibration parameters and RND is a random number between 0 and 1.

For car following regime, the acceleration or deceleration rate b_n is defined as

$$b_n = BNULLmult * (RND4 + NRND) \tag{11}$$

where $BNULLmult$ is a calibration parameter, $RND4$ is a normally distributed driver parameter and $NRND$ is a normally distributed random number.

Since we have more car-following situations for the same driver in the naturalistic driving database, these calibration parameters can be determined by using a heuristic optimization methodology (Genetic Algorithm). For more technical detail of Wiedemann model calibration, please refer to our previous paper [13]. After calibration parameters are determined, Wiedemann deceleration equation (9), (10) and equation (11) can be applied to get the vehicle longitudinal acceleration for the same event example.

BP ANN AGENT AND WIEDEMANN MODEL COMPARISON

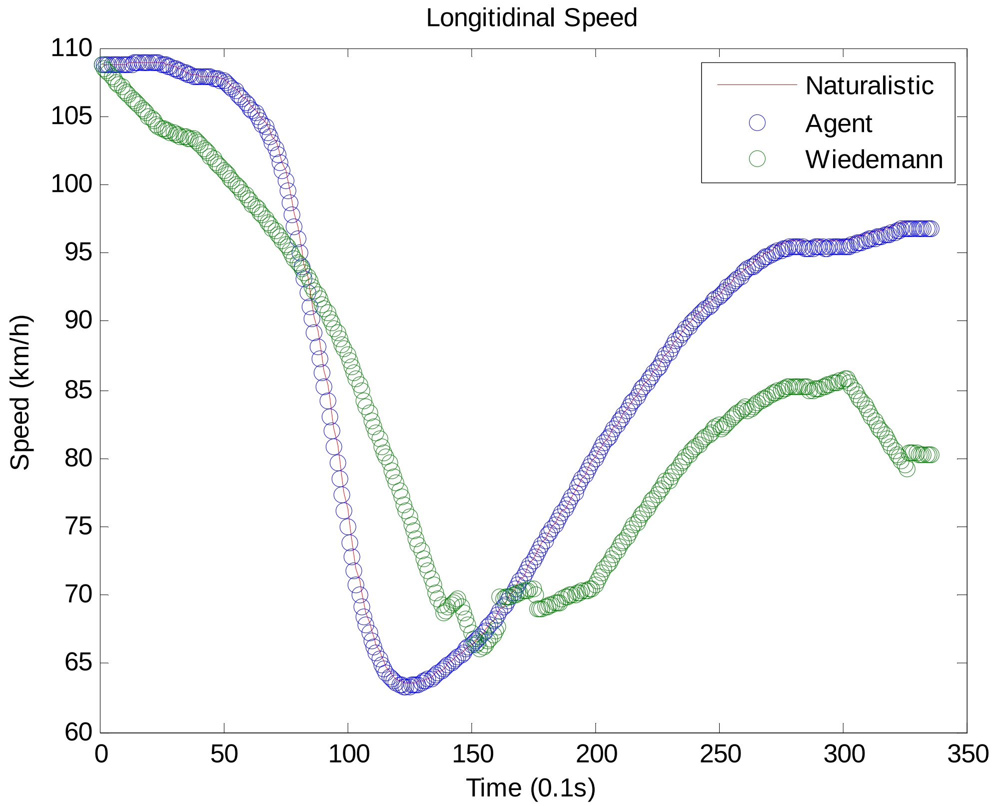

FIGURE 7. BP ANN Agent and Wiedemann Model

Figure 7 shows the performance of the Wiedemann model and a BP ANN agent. Apparently, in this event, the Wiedemann model uses a big deceleration in the beginning to avoid a potential conflict. This result shows consistency with the concept of safety distance in the Wiedemann model definition. Accordingly, the BP ANN agent driver is more willing to take risk while the Wiedemann driver is always using the maximum deceleration during events. However, in our previous research, the Wiedemann model performs quite well when using a large number of car-following situations in calibration and validation [13]. The performance in this example may result from the calibration methodology itself. Compared to large amount of car-following situation data, the number of safety critical events are very few. From the perspective of calibration methodology, insufficient event data may not have any influence on the calibration parameters. In fact, parameters are most probably determined by the driver behavior in car-following regime.

CONCLUSIONS AND FUTURE RESEARCH

In this paper, we first proposed an agent-based backpropagation neural network modeling approach to simulate driver longitudinal and lateral behavior during safety critical events. We used naturalistic microscopic vehicle near crash event in agent training. Simulated speed and offset trajectory are used in validation. Also, the Wiedemann model is calibrated as a comparison to agent performance. Our preliminary results show that the BP ANN agent is able to capture the driver behavior quite well no matter which regime the vehicle is in. Instead, the Wiedemann car-following model always assumed that the driver would take the maximum deceleration in safety critical events. The Wiedemann model may work well in the car-following regime, but lack the capability to simulate driver dependent risk taking behavior in the emergency regime, probably because of insufficient driving data.

This paper represents the potential opportunities for practical applications which might provide realistic simulations in traffic operations. The results could lead to several practical applications in simulation software.

1. Since Wiedemann model is capable of simulatiing car-following behavior quite well and ANN is proved to be a good approximation in simulating driving behavior during safety critical events in the emergency regime, cooperating Wiedemann and ANN could provide a better simulation on realistic driving behavior simulation. Therefore, the switching mechanism from Wiedemann to ANN and ANN to Wiedemann could be important. From our recent study, statistical discriminant analysis can provide switching thresholds to distinguish traffic states between Wiedemann and ANN. By “setting up” predefined thresholds in simulation software, safety critical events simulate by ANN could be incorporated into existing car-following models.

2. Risk-taking driving behavior captured by ANN has the capability to generate a crash or near crash event which can eventually cause the whole traffic break down or result in congestion. By setting up an idle time in simulation software when safety critical events happen (two vehicles stop on the road during this time), the whole traffic flow is affected by idle time and the traffic operation may become much closer to reality. It may worthwhile to analyze the effect of risk-taking behavior of one driver on the traffic operation of the whole system.

The neural agent has been proved to be a successful way of modeling driver behavior in lateral action estimation. In our future research, we intend to study the execution and duration of lateral lane changing behavior because of vehicles taking longitudinal and lateral actions simultaneously. In this paper, we mainly focused on driving behavior during safety critical events. Analysis of lane-changing data may bring insights in more understanding of the driving decision process.

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. Panwai, S. and H. Dia, *Neural Agent Car-Following Models.* IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, 2007. **8**(1): p. 11.
2. Chong, L., M. Abbas, and A. Medina, *Driver Behavior Simulation Using Agent-based Backpropagation Neural Network.* TRB 2010 Annual Meeting CD-ROM, 2011.
3. Wiedemann, R., *Simulation des Strassenverkehrsflusses.* 1974.
4. Wiedemann, R. and U. Reiter, *Microscopic Traffic Simulation, the simulation.* 1992.
5. Olstam, J.J. and A. Tapani, *Comparison of Car-following models*. 2004, VTI Meddelande 990A, Swedish National Road Transport Research Institute.
6. Committee, A.I.a.A.C.A., *Artifical Intelligence in Transportation*, in *Artifical Intellegence*, T.R. Board, Editor. 2007: Washington, DC. p. 88.
7. Peeta, S., W. Zhou, and P. Zhang. *Modeling and mitigation of car-truck interactions on freeways*. 2004: National Research Council.
8. Hidas, P., *Modelling lane changing and merging in microscopic traffic simulation.* Transportation Research Part C: Emerging Technologies, 2002. **10**(Compendex): p. 351-371.
9. Louisell, C., J. Collura, M. Knodler, and K. Heaslip, *A Simple Algorithm for Quantifying the Impact of Driver Behavior on Traffic Flow and Safety During Forced Merges in Work Zones.* 85th Annual Meeting of the Transportation Research Board, Washington, D.C., 2006., 2006.
10. Bonabeau, E., *Agent-based modeling: Methods and techniques for simulating human systems.* 2002. **99**.
11. 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.
12. 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.
13. Abbas, M., et al., *Comparison of Car-Following Models when Calibrated to Individual Drivers using Naturalistic Data.* TRB 2011 Annual Meeting CD-ROM, 2011.