A Solution to the Problem of Extrapolation in Car-Following Modeling Using an Online Fuzzy Neural Network

R.Kazemi[1] ,M.Abdollahzade[2,*]

1. Associated Professor, 2.Ph.D. Student Department of Mechanical Engineering, K.N. Toosi University of Technology, Tehran, Iran

Abstract

Car following process is time-varying in essence, due to the involvement of human actions. This paper develops an adaptive technique for car following modeling in a traffic flow. The proposed technique includes an online fuzzy neural network (OFNN) which is able to adapt its rule-consequent parameters to the time-varying processes. The proposed OFNN is first trained by an growing binary tree learning algorithm in offline mode, which produces favorable extrapolation performance, and then, is adapted to the stream of car following data, e.g. velocity and acceleration of the target vehicle, using an adaptive least squares estimation. The proposed approach is validated by means of real-world car following data sets. Simulation results confirm the satisfactory performance of the OFNN for adaptive car following modeling application.

***Keywords**: car-following models, extrapolation, online fuzzy neural network.*

1. Introduction

Modeling of real world processes are an important but challenging problem, faced by scientists and academia. This is mainly to the time-varying characteristics usually encountered in the real-world systems and processes. In cases when human actions are also involved, the problem of modeling becomes even worse, as humans often introduce complex behaviors into the problem.

Modeling of car following behaviors of the drivers in a traffic flow, is an important task in the field of microscopic traffic modeling. The car following models (CFM) play a key role in intelligent transportation systems (ITS) and are seen valuable to collision warning (CW) and collision avoidance (CA) systems [1]. The CFMs describe the processes in which the drivers follow each other in the traffic stream (Fig. 1), i.e. where a follower vehicle (FV) follows a leading vehicle (LV) [2]. Clearly, the human interactions in the car following process, challenges the car following modeling. Various approaches have been proposed in the literature to develop car-following models, including first-principle mathematical (or analytic) models as well as inputoutput models. While the analytic models, such as general motors (GM) and optimal velocity (OV) models, employ a set of algebraic or differential equations describe the car following behavior input-output models, e.g. neural networks and neuro-fuzzy Models use car following input-output measurement data to develop a model matching the measurements.

Among the first analytic car following models is the General Motors (GM)' or GHR model, developed by Chandler, Herman and Montroll at the General Motors research labs in Detroit [3]. The GM model is a stimulus-response model, where the relative velocity between the FV and LV serves as the stimulus.

Other important stimulus-response models include optimal velocity model (OV) and its derivatives [4]. In the OV model, it is assumed that the driver of the FV seeks a safe velocity determined by the distance from the LV.

The generalized optimal velocity (GOV) model was later developed by Sawada [5] based on the assumption that the FV's driver pays attention not only to its headway but also to the headway of the immediately preceding vehicle (LV).

During the last decade, new intelligent car-following models based on the computational intelligence (CI) techniques, Fuzzy logic [6], neural networks [7] and neuro-fuzzy models [8] have been developed. While fuzzy logic-based CFMs are based on description of the characteristics of human through fuzzy rules, the neural networksbased CFMs are usually constructed using the measurements of the driver's behavior.

As a pioneering research, Chakroborty and Kikuchi fuzzified the input variables of the GM model using a set of fuzzy rules and developed designed a Takagi-Sugeno (TS) fuzzy inference system (FIS) for car-following modeling [6]. Application of Mamdani FIS for car-following modeling has also been reported [9]. However, it's been discussed that owing to the complexities of the real traffic situations, a large number of fuzzy rules are required. Such difficulty limits the application of fuzzy logic for car-following modeling particularly for simulation of large networks [10].

Neural networks and neuro-fuzzy models are prediction oriented approaches which employ field measurement data to develop car-following models [11]. They usually use the past values of the relative speed, relative distance, and velocity of the FV to predict the acceleration or velocity of the FV [12], [10].

Despite the remarkable efforts, made by the researchers on developing effective car following models, but in almost all presented methods, the structure of the CFM is fixed through time and hence, changes and variations in the car following process, mainly due to human action, are not addressed. Therefore, it is concluded that new methods, which are able to take time-varying nature of the car following process into account, are critically needed.

Based on the presented discussions, this paper focuses on developing an online fuzzy neural network (OFNN) to tackle the problem of car following process modeling. The proposed OFNN relies on an adaptive least squares estimation to account for changes in the car following process through the time. Moreover, the structure of the OFNN is determined via an input-space partitioning technique.

Besides, the CI-based techniques, such as fuzzy systems and neural networks, usually produce satisfactory interpolation results while their extrapolation performance is almost degraded, usually dues to normalization of the fuzzy validity functions or activation functions [13]. To overcome this shortcoming, which may manifest itself more severely in car following applications, this paper employs an input-space partitioning algorithm which avoids normalization and hence poor extrapolation performance.

This paper is organized as follows. Section 2 provides the details of the online fuzzy neural network model. The car-following strategy is explained in Section 3 and results on simulation on real measurement data are reported in Section 4. Findings of the paper are concluded in Section 5.

2. Online Fuzzy Neural Network

Fuzzy neural networks (FNN) have been widely employed in various modeling and identification applications [14]-[15]. They are inherently fuzzy inference systems (FIS) which are at least partly learned from measurement data. Hence, in case of the data stream, adaptive algorithms may be applied to estimate the parameters of the FNNs and hence, result in online FNNs.

The general structure of the OFNN with M fuzzy rules, can be expressed by,

$$\hat{y}=\sum_{i=1}^{M}\left(\theta_{i,0}(t)+\theta_{i,1}(t)x_1+\ldots+\theta_{i,p}(t)x_p\right)\Phi_i(\underline{x}) \tag{1}$$

Where, $\theta_{i,j}(t)$ are adaptive, time-dependent coefficients of the consequent part of the FIS, $\Phi_i(\cdot)$ indicate the premise part of the fuzzy rules, and x_1 to x_p are the inputs of the OFNN. **Error! Reference source not found.** shows the four-layer structure of the OFNN for three inputs and three fuzzy rules. The validity functions in the second layer are computed as described in Section 2.1.

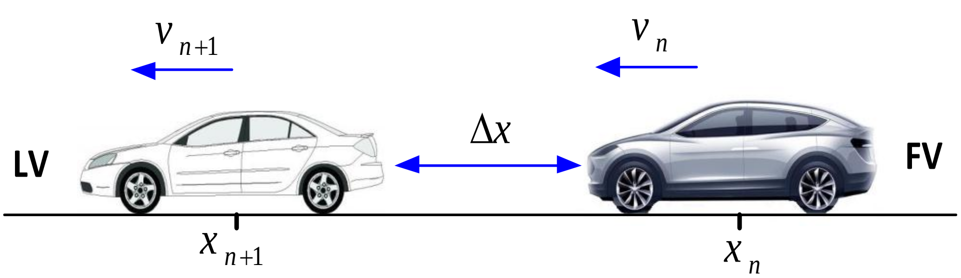

Fig1. Car-following behavior

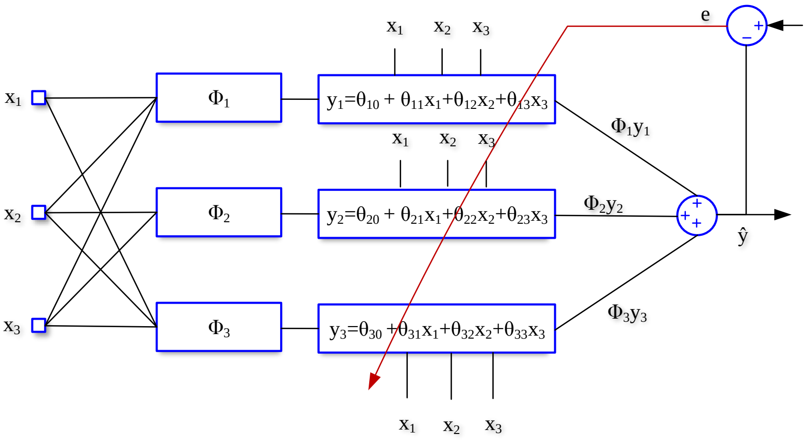

Fig2. Structure of the OFNN with three inputs and three fuzzy rules

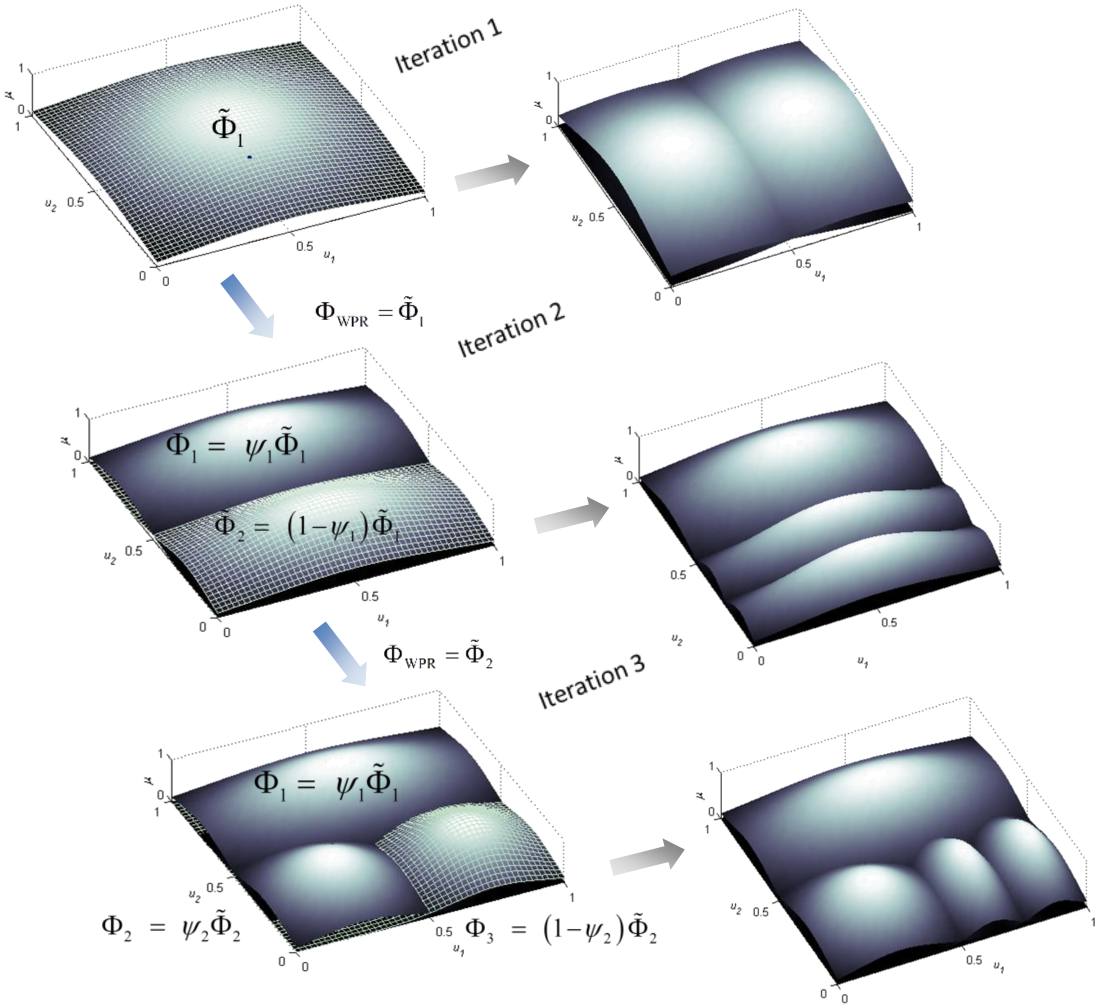

Fig3. Partitioning of input space by growing tee example in three iterations

Identification of the OFNN's Structure

The identification of the OFNN's structure is carried out through offline training. The main theme of the training algorithm is partitioning of the input space, carried out by a growing binary tree in this paper. The growing binary tree expands the structure of the OFNN in an iterative manner, until satisfactory performance is obtained.

At each iteration of the training algorithm, one fuzzy rule is added to the rule base of the OFNN. To be more specific, at each iteration, the worst-performing rule is replaced by two new rules, resulting in expansion of the fuzzy rule base by totally one rule. The partitioning of a two-dimensional input space by the growing binary tree is shown in **Error! Reference source not found.**.

At each iteration of the growing binary tree, the validity function of two new rules ($\Phi_{new\,1}$ and $\Phi_{new\,2}$) are computed by multiplying the sigmoid function (ψ) and its counterpart ($1-\psi$) by the validity function of the worst-performing rule (Φ_{WPR}),

$$\Phi_{new\,1} = \psi\Phi_{WPR} \tag{1}$$

$$\Phi_{new\,2} = \left(1-\psi\right)\Phi_{WPR} \tag{2}$$

The sigmoid functions in (1) and (2) are expressed by

$$\psi\left(\underline{x}\right) = \frac{1}{1+\mathrm{e}^{-\left(d_0+d_1x_1+\ldots+d_px_p\right)}} \tag{3}$$

where, vector di = [di,0 di,1 … di,p]T includes the parameters of the sigmoid function which are determined heuristically.

It is seen from (1) and (2) that no normalization is applied on the generated validity functions, and hence, the poor extrapolation performance, due to the normalization, is avoided.

Having determined the validity function of the new rules, the rule-consequent parameters of the new rules during the offline training, $\theta_{i,j}\left(0\right)$, can be easily estimated by a weighted least squares technique. Assuming that the following N data samples are available,

$$\underline{X}_0 = \begin{bmatrix} \underline{x}_0^T(1) \\ \underline{x}_0^T(2) \\ \vdots \\ \underline{x}_0^T(N) \end{bmatrix}_{N\times p}, \quad \underline{Y}_0 = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}_{N\times 1} \tag{4}$$

where, $\underline{x}_0\left(i\right) = \left[x_1\left(i\right)x_2\left(i\right)\ldots x_p\left(i\right)\right]^T$.

Then by considering the following local error minimization problem,

$$\min_{\underline{\theta}_i}\left\{I_i = \sum_{j=1}^{N}\Phi_i\left(\underline{x}_0(j)\right)e^2(j)\right\}, i = 1,\ldots,M \tag{5}$$

where, $e\left(j\right) = y\left(j\right) - \hat{y}\left(j\right)$, the following WLS estimation is obtained for the rule-consequent parameters,

$$\underline{\theta}_i\left(0\right) = \left(\underline{R}_i^T\underline{D}_i\underline{R}_i\right)^{-1}\underline{R}_i^T\underline{D}_i\underline{Y}_0 \tag{6}$$

where,

$$\underline{R}_i = \left[1_{N\times 1}\ \underline{X}_0\right]_{N\times(p+1)} \tag{7}$$

and,

$$\underline{D}_i = \begin{bmatrix} \Phi_i\left(\underline{x}_0(1)\right) & 0 & \ldots & 0 \\ 0 & \Phi_i\left(\underline{x}_0(2)\right) & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \Phi_i\left(\underline{x}_0(N)\right) \end{bmatrix} \tag{8}$$

This procedure is carried on until a satisfactory performance for the offline model is achieved.

Online Estimation of Rule-Consequent Parameters

The rule-consequent parameters of the OFNN are estimated by an adaptive least squares algorithm in an online fashion. As mentioned earlier, it is assumed that an OFNN has already been trained and built offline a priori. Then, by arrival of the new data samples, the rule-consequent parameters of the OFNN are adapted to the new data by the adaptive least squares algorithm.

For the new data sample at time instant t, $\underline{x}\left(t\right)$, the rule-consequent parameters of the TFNN are adapted using the following adaptive least squares estimation,

$$\underline{\theta}_i\left(t\right) = \underline{\theta}_i\left(t-1\right) + \underline{\gamma}_i\left(t\right)e_i\left(t\right) \tag{10}$$

$$e_i\left(t\right) = y\left(t\right) - \underline{\tilde{x}}^T\left(t\right)\underline{\theta}_i\left(t-1\right) \tag{11}$$

$$\underline{\gamma}_i\left(t\right) = \frac{1}{\underline{\tilde{x}}^T\left(t\right)\underline{P}_i\left(t-1\right)\underline{\tilde{x}}\left(t\right) + \frac{\lambda}{\Phi_i\left(\underline{x}(t)\right)}}\underline{P}_i\left(t-1\right)\underline{\tilde{x}}\left(t\right) \tag{9}$$

$$\underline{P}_i\left(t\right) = \frac{1}{\lambda}\left(\underline{I} - \underline{\gamma}_i\left(t\right)\underline{\tilde{x}}^T\left(t\right)\right)\underline{P}_i\left(t-1\right) \tag{10}$$

where, $\underline{\tilde{x}}^T\left(t\right) = \left[1\ \underline{x}^T\left(t\right)\right]$.

According to **Error! Reference source not found.**-(10), the rule consequent parameters of the OFNN are adapted to new data samples.

Car-Following Strategy

The OFNN developed in this paper is used to predict the velocity and acceleration of the FV's driver during a car-following situation. To achieve this goal, previous values of acceleration and velocity together with the distance and relative velocity between FV and LV are employed to build the CFM. The models for predicting the acceleration and velocity of the FV can be expressed as,

$$a(t+1) = f_a\left(\underline{a}, \underline{v}_r, \underline{x}_r\right) \tag{11}$$

$$v(t+1) = f_v\left(\underline{v}, \underline{v}_r, \underline{x}_r\right) \tag{12}$$

where, fa and fv are nonlinear functions which predict the future values of acceleration and velocity at time t + 1, respectively and the vectors a, v, vr, and xr include the previous values of acceleration, velocity, relative velocity and relative distance, respectively. The OFNN identifies functions fa and fv using the measurement data. **Error! Reference source not found.** shows the structure of car-following modeling using the proposed approach.

4.Simulation Results and Discussion

To evaluate the performance of the proposed approach, the results of simulation on real data sets are presented here. For this purpose, the US Federal Highway Administration's I-80 data are employed **Error! Reference source not found.**]. The data were collected in 0.1-sec intervals. Any measured sample in this data set has 18 features of each drive-vehicle-unit (DVU), such as longitudinal and lateral position, velocity, acceleration, time, number of road, vehicle class, front vehicle, etc. Moreover, a FNN without an online training is used for the purpose of comparison.

In order to assess the proposed OFNN performance numerically, the root man square error (RMSE), defined below, is used.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(y_t - \hat{y}_t\right)^2} \tag{13}$$

where, $y(t)$ and $\hat{y}_t$ are the actual and identified outputs at sample t, respectively, and N is the number of identified samples.

The data collected for two different vehicles are used to validate the proposed OFNN-based approach.

4.1Car-Following for Vehicle A

The velocity and acceleration profiles for vehicle A are shown in **Error! Reference source not found.**. Each quantity includes 450 data samples, where the first 350 samples are used to build the initial model in offline mode and the next 100 samples serve as a test data set to evaluate model performance in online mode.

The performance of the model for this case is shown in **Error! Reference source not found.** and **Error! Reference source not found.**. Moreover, these Figs. demonstrate that performance of the FNN is inferior to the OFNN's, particularly in case of acceleration. Besides, favorable extrapolation of the OFNN is obvious from **Error! Reference source not found.** and **Error! Reference source not found.**.

To perform more comprehensive comparison, the results obtained by both methods are also reported in Tables 1. The much better performance of the proposed approach in prediction of acceleration and velocity is seen from these tables.

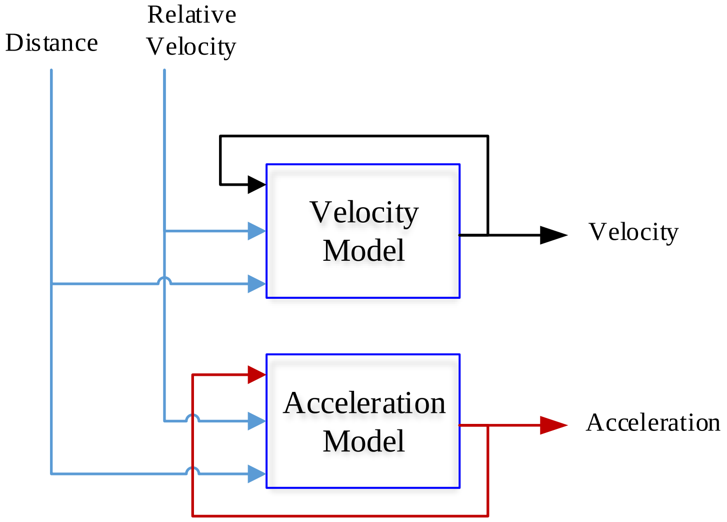

Fig4. velocity and acceleration modeling approach

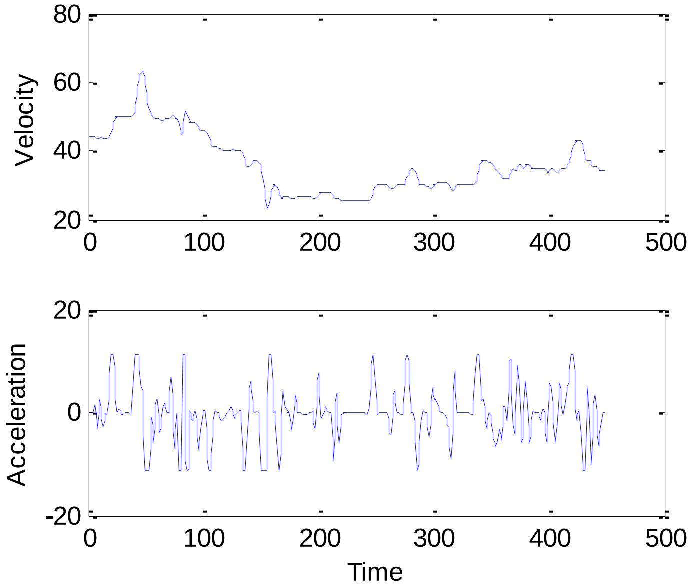

Fig5. Velocity and acceleration for case A

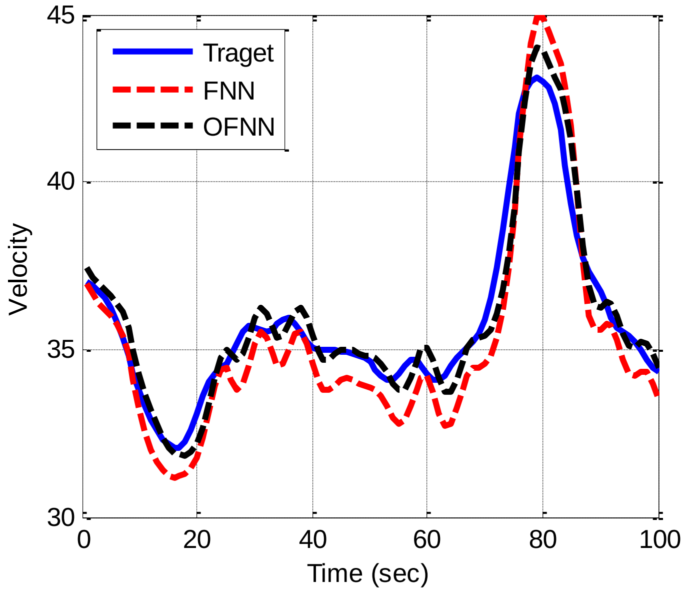

Fig6. Velocity prediction for vehicle A

Table 1. RMSE comparison for case A

	Velocity	Acceleration
FNN	1.12	2.92
OFNN	0.62	0.56

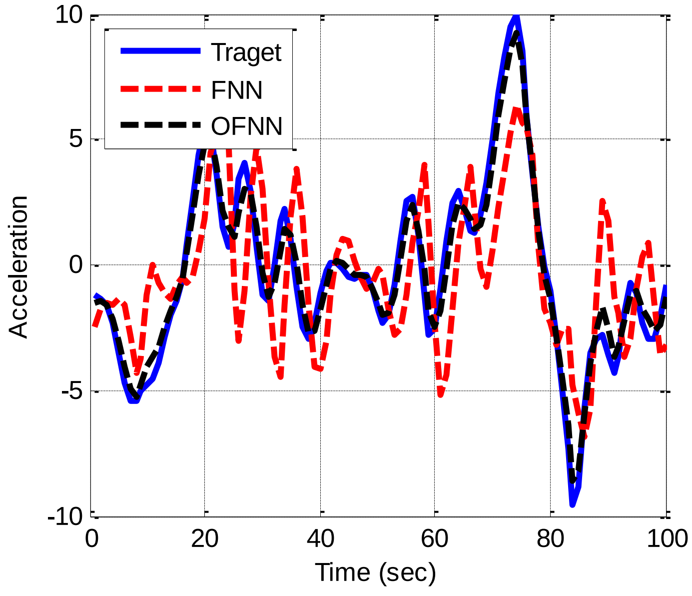

Fig7. Acceleration prediction for vehicle A

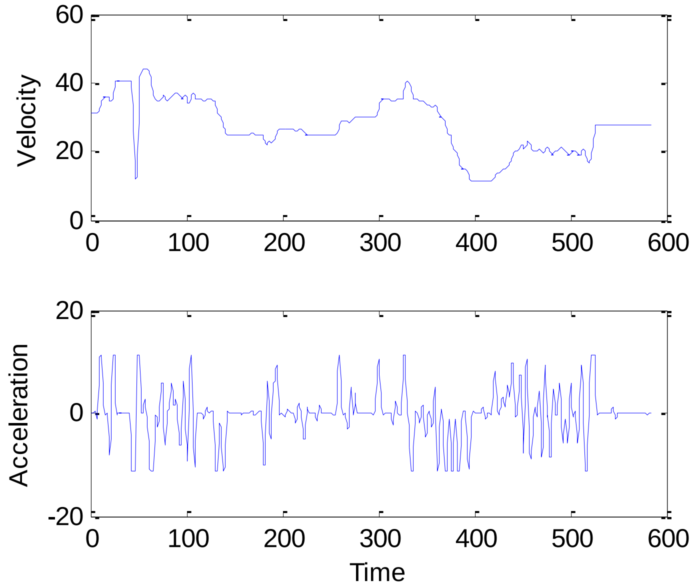

Fig8. Velocity and acceleration profile for case B

Table 2. RMSE comparison for case B

	Velocity	Acceleration
FNN	0.80	2.46
OFNN	0.24	0.82

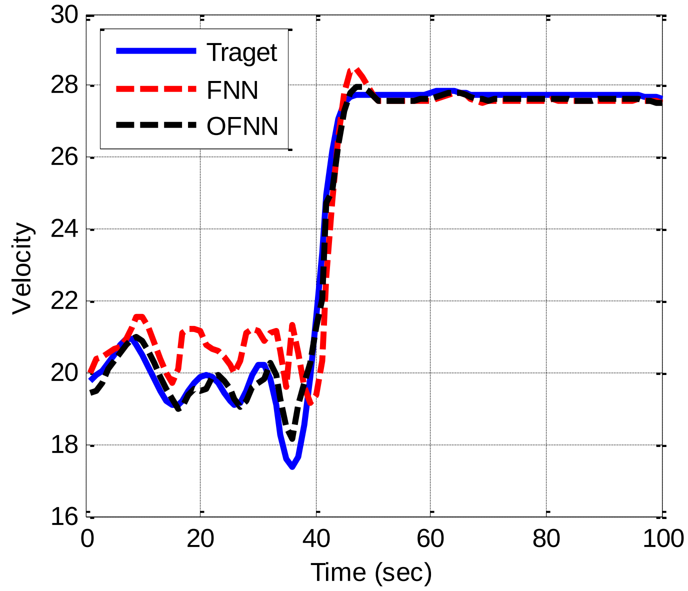

Fig9. Velocity prediction for case B

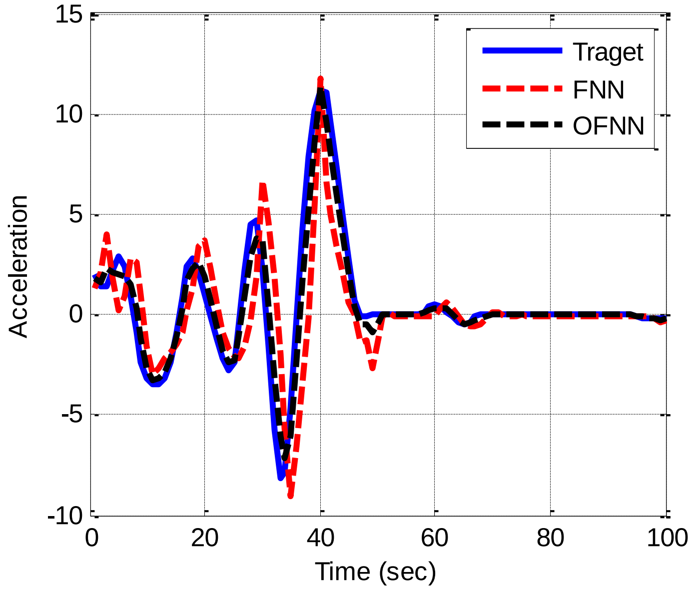

Fig10. Acceleration prediction for case B

4.2Car Following for Vehicle B

The velocity and acceleration profiles for vehicle B are shown in **Error! Reference source not found.**. Obviously the profile of the velocity is quite different compared to vehicle A in previous case. The target and predicted value of velocity and acceleration are shown in **Error! Reference source not found.** and **Error! Reference source not found.**. In this case, the OFNN has outperformed its offline version, i.e. FNN. The numerical comparisons, presented in Table 2 confirm this conclusion.

Conclusion

This paper proposed an adaptive approach to model car following process in traffic flows. The proposed online fuzzy neural network employed an adaptive least squares estimation to address the time-varying behavior of the car following processes, which is mainly due to the involvement of the human actions. Besides, a growing binary tree was also utilized to identify the initial structure of the OFNN through offline training. The growing binary tree, does not apply normalization during the construction of the validity functions, and hence avoids poor extrapolation performance. The results of simulations using US Federal Highway Administration's NGSIM data confirmed the accurate performance of the proposed approach.

References

[1]. Ghaffari A, Khodayari A, Hosseinkhani N. and Saeed Salehinia. The effect of a lane change on a car-following manoeuvre: anticipation and relaxation behaviour. Proc ImechE, Part D: Journal of Automobile Engineering, 0954407014547930, in Press.

[2]. Eben Li S and Peng H. Strategies to minimize the fuel consumption of passenger cars during car-following scenarios, Proc ImechE, Part D: Journal of Automobile Engineering 2012; 226: 419-429.

[3]. Chandler RE, Herman R and Montroll EW. Traffic dynamics: studies in car following. Operations Research 1958; 6: pp. 165-184.

[4]. Bando M, Hasebe K, Nakayama A, et al. Dynamics model of traffic congestion and numerical simulation. Physic Review E 1995; 51: 1035–1042.

[5]. Sawada S. Nonlinear analysis of a differential-difference equation with next-nearest-neighbour interaction for traffic flow. Journal of Physics A: Mathematical and General 2001; 34: 11253 – 11259.

[6]. Chakroborty P and Kikuchi S. Evaluation of the General Motors based car-following models and a proposed fuzzy inference model. Transportation Research Part C 1999; 209-235.

[7]. Shankar KVR, Mathew TV and Dey A. Neural Network Based Car following Model. In: International Conference on Advances on Materials,Mechanics and Management (AMMM2010), 2010, pp. 314-320.

[8]. Li S. Simulation of car-following decision using fuzzy neural networks system. IEEE Proceeding of Intelligent Transportation Systems 2003; 1: 140-145.

[9]. Zheng P and McDonald M. Application of Fuzzy Systems in the Car-Following Behavior Analysis. In: Wang L and Jin Y(eds) Fuzzy Systems and Knowledge Discovery. Springer Berlin Heidelberg, 2005, pp. 782-791.

[10]. Khodayari A, Ghaffari A, Kazemi R and Braunstingl R. A Modified Car-Following Model Based on a Neural Network Model of the Human Driver Effects. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 2012; 42: 1440,1449.

[11]. Ghaffari A, Khodayari A and Faraji M. Optimal fuzzy control system design for car-following behaviour based on the driver–vehicle unit online delays in a real traffic flow. Proc ImechE, Part D: Journal of Automobile Engineering 2014; 228: 1440-1451.

[12]. Zheng J, Suzuki K and Fujita M. Car-following behavior with instantaneous driver–vehicle reaction delay: A neural-network-based methodology. Transportation Research Part C: Emerging Technologies 2013; 36: 339-351.

[13]. Abdollahzade M, Mahjoob M, Zaringhalam R and Miranian A. Locally linear neuro-fuzzy (LLNF) electricity price forecasting in deregulated power markets. Int. J. Innovative computing, Inform. Control 2010; 6: 1-15.

[14]. Abdelazim T and Malik OP. Identification of nonlinear systems by Takagi–Sugeno fuzzy logic grey box modeling for real-time control. Control Engineering Practice 2005; 13: 1489-1498.

[15]. Kukolj D and Levi E. Identification of complex systems based on neural and Takagi-Sugeno fuzzy model. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 2004; 34: 272-282.

[16]. US Federal Highway Administration's Next Generation Simulation. [online]. Avialable: http://ngsim.fhwa.dot.gov.