Artificial Neural Network Models For Car Following: Experimental Analysis And Calibration Issues

Article *in* Journal of Intelligent Transportation Systems · May 2013

DOI: 10.1080/15472450.2013.801717

2 authors:

Chiara Colombaroni
Sapienza University of Rome

Gaetano Fusco
Sapienza University of Rome

Artificial Neural Network Models For Car Following: Experimental Analysis And Calibration Issues

CHIARA COLOMBARONI[1] **and GAETANO FUSCO**[1]
[1] Research Centre for Transport and Logistics, Sapienza University, Italy

The paper deals with the application of Artificial Neural Networks to model the car following driver's behavior. The study is based on experimental data collected by several GPS equipped vehicles that follow each other on urban roads. A 'swarm' stochastic evolutionary algorithm has been applied in training phase to improve convergence of the usual error-back propagation algorithm. Validation tests highlight that ANNs provide a quite good approximation of driving patterns and can be suitably implemented in micro-simulation models. More advanced applications to ITS may concern Advanced Driver Assistance Systems and are addressed to conform braking actions to drivers' expectations.

Keywords Car following models, GPS tracking experiments, microsimulation

Car following models aim at catching the basic fundaments of traffic flow and model the relationship between consecutive drivers in a traffic stream. The first car following models have been developed during the experiments carried out by General Motors in the early '60s on specifically dedicated tracks. They are based on the basic relationship *stimulus*-response, which assumes the acceleration of the following vehicle as proportional of the relative speed to the leader car (Chandler *et al*., 1957) or as a function of vehicle speed and spacing (Gazis *et al*., 1961). Simpler linear models of follower's speed were developed in the same years (Kometani and Sasaki, 1959) and enhanced later on (Gipps, 1981). Some authors argued that the stimulus-response model cannot be based on a unique sensitivity function for any value of spacing and relative speed and introduced different functions for perception thresholds that discriminate the models holding in the different driving conditions: following, approaching, emergency breaking, free flowing (Leutzbach and Widemann, 1986).

In the last years, availability of high precision kinematic GPS has made it possible accurate vehicle tracking and has stimulated so a renewed interest in observing car following behavior through new experiments and developing new models (among others we mention here: Ossen and Hoogendorn, 2004; Punzo and Simonelli, 2007; Viti *et al*., 2008; Wu *et al*., 2003; Brockfeld *et al*., 2005).

Artificial intelligence (AI) seems to be a very suitable paradigm to model the psycho-physical mechanism on the basis of driving behavior. More specifically, among different AI paradigms, Artificial Neural Networks (ANN) have been conceived in the early '40s with the aim of reproducing the brain mechanism (McCulloch and Pitts, 1943) and have got a great development in later years thanks to efficient algorithms for training them on large sets of observed data (Ruelhardt *et al*., 1986). ANNs are robust estimators of non-linear, stochastic and noisy phenomena, such as driving behavior. Thus, one can envision training a specific ANN for a single driver and applying it for simulating his or her behavior within microsimulation traffic models. A more advanced application can be addressed to ADAS systems in order to conform braking actions to drivers' preferences and avoid undesired actions (Bifulco *et al*., 2010).

Very few experiments have been carried out so far to apply ANNs to car following behavior. Main features of the most remarkable of them are summarized in the following.

Panwai and Dia (2007) applied different ANN architectures (feed-forward, fuzzy ARTMAP and Radial basis function) to classify observed driving conditions. Results showed that neural network models outperformed the Gipps and the so-called psychophysical car following models. Moreover, feed-forward ANN provided slightly better results than fuzzy ARTMAP and much better than radial basis function. Zhang *et al*. (2008) developed a feed-forward ANN that supplies follower's acceleration by taking as inputs time headway and time-to-collision with respect to the leader. Neither Panwai and Dia nor Zhang *et al*. dealt with driver's reaction time, which plays though a significant role in classical car-following behavioral models.

Simonelli *et al*. (2009) proposed a different approach and introduced a learning machine, based on an ANN, capable to learn the attitudes of the actual driver and then to conform adaptive cruise-control system actions to driver's preferences.

Authors tested two different ANNs: a feed-forward ANN, where a time offset is introduced between inputs and outputs that takes into account the time delay of follower's response, and an Elman network, which introduces a feedback connection from hidden to input layer with the aim of recognizing dynamic patterns expected in car following behavior. Both ANNs outperformed traditional Gipps model.

The present paper comes into this research stream and aims at investigating some expected capabilities of ANNs.

The first question is the capability of a single ANN to reproduce the driver behavior in different traffic conditions.

The second question is the capability of usual training algorithms for ANNs to deal with highly noisy data. This issue has a significant importance from a practical point of view, since car following data are collected at very short intervals (usually 0.1s) and require filtering techniques to reduce the high measurement noise.

The third question concerns the delay of driver's reaction to leader's stimulus. In literature, this delay is usually assumed to be constant and is estimated through repetitive calibrations of car following model for different values of the reaction time. However, the reaction time may change depending on the specific situation (*e.g*., in following or in approaching the leading vehicle). Furthermore, it is possible that the follower's reaction is not affected only by the stimulus perceived in one given instant before; indeed, it is reasonable that the stimulus-reaction mechanism be distributed on an even short time interval. In fact, the driver is conscious of his (or her) previous actions and it is expected that he (or she) can adapt his (or her) current behavior, not only with respect to the current distance from the desired state, but also with reference to the expected distance, which depends on the actions already performed. This assumption is based on the hypothesis that the driver has a strategy in mind and tries to apply it in a smooth way.

Finally, a forth question concerns the implementation of the ANN model into a micro-simulation model. Representation of drivers' behavior would be more realistic; moreover, a new experimental method for model calibration could be applied. In on-the-field experiment, a representative sample of drivers might be observed in real driving tasks by equipping their vehicles with devices for measuring spacing and speed (*e.g.*, radar). Alternatively, drivers might be involved in laboratory experiments on virtual reality driving simulator, where an even wider set of driving conditions might be simulated. A comparison with the actual driver behavior observed in analogous real conditions will be required in this case to reduce the bias between virtual and real conditions.

In this paper we describe an experiment of tracking GPS equipped vehicles along different urban street routes and we introduce and test different specifications of ANNs to address the aforementioned questions. The research is still ongoing. Nevertheless, it has provided useful indications and several suggestions for further research.

EXPERIMENTAL TESTS

Several experiments were conducted by tracking from 2 to 4 vehicles equipped with one GPS device on different urban streets in Rome, Italy. Position of every vehicle was collected at 10Hz frequency. Data were post-processed to eliminate systematic errors by using the GPS reference station at Sapienza University of Rome. Speed and accelerations were computed by numerical derivation and filtered by applying moving median method.

Accuracy of GPS devices was assessed preliminarily by comparing the measured positions of two devices, mounted on a steel bar at 1 m distance (Fig.1a). Average errors resulted of 2 mm in stillness and ranged from 0.2 to 17 cm in motion. Different experiments, involving different drivers, were conducted on both local and main streets, in order to reproduce different traffic conditions, including stopping and starting at signals, as well as approaching and moving on roundabouts. Characteristics of different tests are reported in Table 1. Route of experiment (L, S) is shown in Fig.1b.

It is worth mentioning that only useful data sets for calibration are reported here. Indeed, tests were even much longer, but many data were useless because either poor GPS signals or uninteresting driving conditions, like stopped vehicles at signals or free-flow speed. However, the length of useful experiments is similar or larger of that used by other authors.

Table 1: Details of experiments: ID of drivers involved, maximum and minimum values of relative speed (Δv), spacing (Δx) and follower acceleration (a_f)

Drivers	(L, S)	(L, S’)	(A, M)
Characteristics of tests	*Length* = 1234,7m *Time* = 154 s	*Length* = 1234,7m *Time* = 154 s	*Length* = 6446,8m *Time* = 477 s
Maximum values of variables	Δv = 3.15 m/s Δx = 31.36 m af = 1.48 m/s^2	Δv = 3.06 m/s Δx = 35.5 m af = 1.057 m/s^2	Δv =10.02 m/s Δx = 93.36 m af = 1.7 m/s^2
Minimum values of variables	Δv = -3.07 m/s Δx = 4.51m af=-1.36 m/s^2	Δv = -3.07 m/s Δx = 4.67 m af= - 1.29 m/s^2	Δv= - 4.75 m/s Δx =5.7 m af= - 2.7m/s^2

Note: ID (*L*, *S’*) refers to simulated follower trajectory.

Fig.1a: Route of experiment (L, S).

Fig. 1b: Pair of GPS antennas used for accuracy measures.

FORMULATION OF NEURAL NETWORK MODELS

Since the first formulation of perceptron by Rosenblatt (1968), numerous different formulations of ANNs have been provided in the last years. In this paper we use the classic feed-forward multilayer ANN, which in first experiments by Panwai and Dia demonstrated to outperform other more complex formulations. Moreover, FFW ANN formulation allows us expressing delay between input and output explicitly, while Elman ANN embeds dynamic patterns of car following into the network structure.

Mathematical formulation of a 2-layer feed-forward ANN is:

$$z_j = f\left(\sum_{h=1} b_{j,h} \cdot f\left(\sum_{i=1} a_{h,i} \cdot x_i\right)\right) \quad (1)$$

where: z_j is the output, x_i is the input vector, f is the transfer function, a_{hi} and b_{jh} the matrices of weights of the input-hidden and the hidden-output layers, respectively.

The mathematical formulation can be illustrated through the node-link representation (Fig.2).

Different specifications of feed-forward ANNs and the input variables were considered: relative speed, spacing, follower's speed, estimated leader's acceleration or combinations of them. The output is follower's acceleration, which is indeed the variable that driver controls directly, by acting on either braking or accelerator pedal.

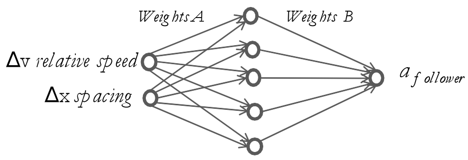

Fig. 2: Feed-forward neural network with 2 input neurons, 5 neurons in the hidden layer and 1 output neurons.

In all of them, inputs (*stimuli*) and output (*response*) were lagged to take into account the reaction time. This reflects the classic stimulus-reaction model that assumes the follower's response is delayed with respect to the stimulus from the leader by the follower's reaction time, which is assumed to be constant. In such a way, the mechanism stimulus-reaction is time-independent and the corresponding neural network is memoryless.

As mentioned in the introductive section, we aim at investigating the realism of a more complex reaction mechanism, which assumes that the stimulus-reaction mechanism is distributed on a time interval. Thus, we introduce a FFW ANN with memory, which introduces am explicit time-dependent relationship between output and the input layer:

(2)

In this case, the structure of ANN consists of 6 neurons of input, corresponding to the input values in the 3 last time intervals, *i.e.*: 1.5s, 1.0s e 0.5s before. Such a different structure with memory is depicted in Figure 3.

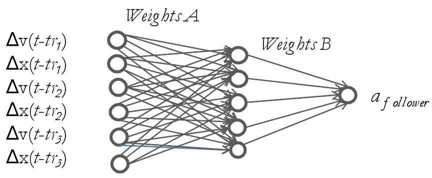

Fig. 3: Feed-forward neural network with memory having 6 input neurons, 5 neurons in the hidden layer and 1 output neuron.

LEARNING PROCESS: ERROR BACK PROPAGATION

Matrices of weight are the coefficients of the neural network and have to be calibrated in order to reproduce some given sets of input-output observations with minimum errors. Because of the complex non-linear structure of the neural network, calibration requires an iterative process, whose result depends on the initial values of the weights and the order of data processing. In analogy with human processes, such a calibration process is called ‘training’ or ‘learning’.

The classic learning algorithm for feed-forward ANNs is the error-back propagation (EBP) algorithm, applied also by abovementioned applications of ANN to car following. EBP seeks to minimize the sum of square errors between observed and computed outputs by applying a two step gradient approximation.

In our study, different numbers of neurons of the hidden layer (NHL, varying from 2 to 9) and of the input layer (NIL, varying from 1 to 3, corresponding to the different above mentioned variables), different values of learning coefficients D (varying from 0.1 to 1) as well as different data sets (drivers pairs L,S and A,M) were examined during the learning process. Moreover, specific neural networks for different driving modes, whose definitions were taken from Panwai and Dia, were tested. A sigmoid transfer function was used in all experiments.

Table 2 reports main characteristics of different tests and summarizes the best results obtained for each experiment and each neural network specification, in terms of mean square error (MSE) and mean square normalized error (MSNE).

Results obtained have been compared with those provided by traditional stimulus-reaction car following models, calibrated on the same data set, whose results are more widely described in Cipriani *et al.* (2010). In all tests ANNs outperformed traditional models, as exemplified in Fig.4, which compares observed follower acceleration and corresponding values provided by Chandler Model (R^2=0.59) and ANN (whose R^2=0.74).

Table 2: Details of experiments and synthesis of results

Drivers	(L, S)		(L, S')		(A, M)		
Characteristics of tests	*Length* = 1234,7m *Time* 154 s		*Length* = 1234,7m *Time* 154 s		*Length* = 6446,8m *Time* 477 s		
Normalization constants	Δv=6.2 m/s Δx=31.36 m af =1.4 m/s2		Δv=6.8 m/s Δx=35.5 m af =1.3 m/s2		Δv=10.02 m/s Δx=93.36 m af =2.7m/s2		
Neural Network memory-less	D=0.1 NHL=5 NIL=2	MSNE=0.098 MSE=0.19	D=1 NHL=5 NIL=2	MSNE=0.004 MSE=0.006	D=0.1 NHL=5 NIL=2		MSNE=0.06 MSE=0.51
Neural Network with memory	D=0.1 NHL=5 NIL=6	MSNE=0.088 MSE=0.15	D=1 NHL=5 NIL=6	MSNE=0.005 MSE=0.01			
Neural network for different driving modes					D=1 NHL=5 NIL=2	MSNE *Free Flow*=0.12 *Danger*=0.54 *Approaching*= 0.41 *Car f ollowingI*=0.53 *Car f ollowingII*=0.46	MSE *Free Flow*=0.97 *Danger*=0.77 *Approaching*=0.82 *Car f ollowingI*=0.69 *Car f ollowingII*=1.7

Gaetano Fusco 20/11/11 18:33
Commento: ANN with memory has NIL=6
Check MSNE e MSE of ANN with different driving modes: Car following I 0.53 vs 0.011?

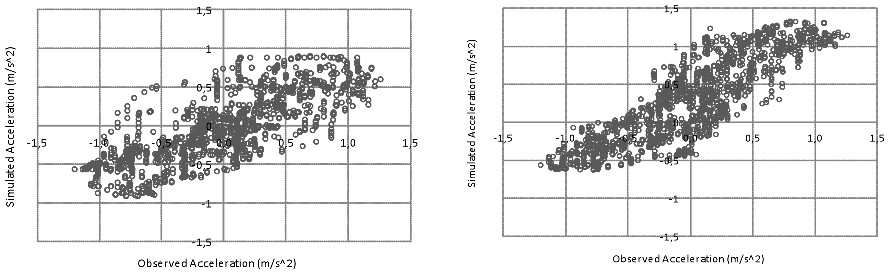

Fig. 4 – Comparison of observed follower acceleration and values provided by Chandler Model (R^2=0.59, on the left) and Neural Network (R^2=0.74, on the right)

It is worth noting that the results of the training process, although better than estimates provided by traditional car-following models, are significantly dependent on the values of learning coefficients. In many cases we experienced divergence of the objective function, as shown in Fig.5a. However, even different ANN models provide a quite satisfactory approximation of follower's acceleration also in presence of high noise of data. An example is shown in Fig. 5b, which is referred to two ANNs with 5 hidden neurons and spacing and relative speed as inputs.

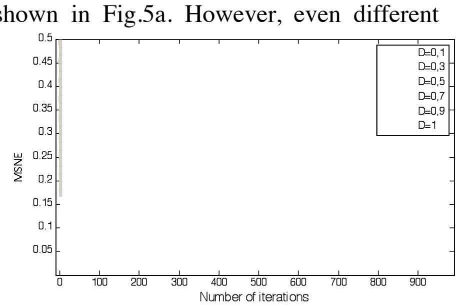

Fig.5a– Convergence of learning process of a FFWANN by EBP algorithm for different values of learning coefficients D.

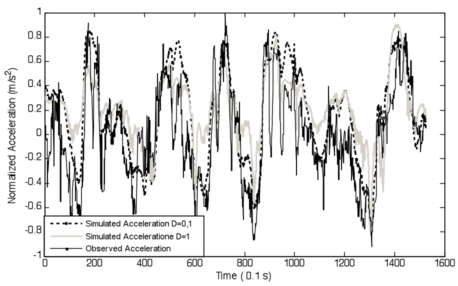

Fig. 5b – Comparison of observed follower's acceleration and the corresponding outputs provided by two FFW ANNs trained using learning coefficients D=0.1, D=1.

In order to test whether abovementioned difficulties of convergence of the error-back propagation algorithm are due to the noise that affects the data, we tested the effectiveness of EBP algorithm in an environment with very low noise, made by applying the well-known Chandler's car following model to obtain acceleration of the follower given the leader's trajectory.

Leader's data are collected on the field, while the follower's data have been computed by the Chandler model. Even for a learning coefficient D=1, which exhibited the worst convergence, the algorithm converges regularly toward very low value of the error, as highlighted in Fig.6a. So, it's reasonable thinking that the bad convergence of Error Back Propagation is due to the high noise of data. Fig. 6b compares follower's acceleration diagrams obtained by Chandler model and the corresponding neural network. Approximation in deceleration phases is almost perfect, while in acceleration phases it's anyway very good, only slightly underestimating the acceleration peaks. Values of MSNE (0.004) and MSE (0.006) are referred as test (LS') in Tab.2.

MULTI-REGIME FORMULATION

Comparison of observed and computed trajectories has highlighted that some ANN models fit experimental data well in some phases of motion, but are flawed in others. This is the case of the example already shown in Fig.5b. Accelerations are quite well reproduced, while strong decelerations are underestimated. This occurrence suggests that the complexity of driving tasks can be hardly reproduced by a simple feed-forward ANN with one 5-neuron hidden layer and a more complex multi-regime formulation should be introduced. On this regard, we tested 5 different ANN models for different driving conditions: free-flow, approaching, regular following, cautious following, braking. Each condition was individuated by given thresholds in the spacing-relative speed plane. While poor training results were expected for the free-flow condition, where follower's accelerations are weakly related to the leader's trajectories, one could expect good results from the other models.

However, none of the ANNs trained on these specific data sets selected from homogeneous traffic conditions outperformed the single ANN model trained on the whole data set. Even if the set is not large enough to give a definitive answer to this question, one can argue that training on homogeneous conditions did not allow the neural network exploring the whole value interval of the variables and so weakened the learning process.

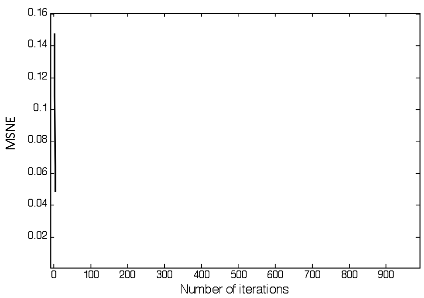

Fig. 6a – Convergence of learning process of a FFW ANN by EBP algorithm using Chandler model data.

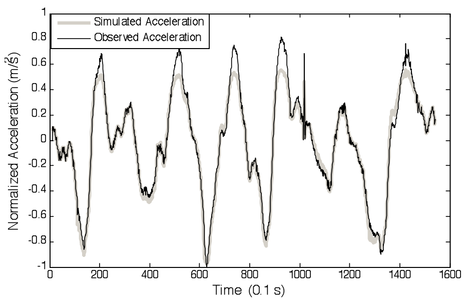

Fig. 6b – Comparison of follower's acceleration computed by Chandler model and corresponding output of FFW ANN trained on the same data.

On the other hand, ANNs have a rather complex mathematical structures that are reasonably expected to be able to reproduce not only driver's classification task among driving conditions but to reproduce the typical driver's task in a traffic stream with no overtaking. So, it is not only reasonable, but it would be even a requirement that ANN provide a good representation of driver behaviour in different traffic conditions.

TIME-DEPENDENT FORMULATION

A further question concerns the time-dependency of follower's reaction with respect to leader's stimuli, which has been introduced in the aforementioned time-dependent ANN. Results of training process (reported in Table 2, experiment L,S) highlight that ANN with memory improves the performance of traditional memoryless FFW ANN as 10% and 21%, with reference to MSNE and MSE, respectively. Of course, the opposite occurs when ANN with memory is applied to train Chandler model with constant reaction time of 1s (experiment L,S').

A NEW TRAINING ALGORITHM

Analysis of learning process has highlighted that performances of EBP algorithm are heavily affected by noise of data.

To improve learning performances, we tested an alternative approach and we applied a swarm optimization algorithm to train the ANN. This algorithm is inspired to the behavior of social natural organisms, like ants and bees, and applies a collective strategy to reach the minimum of an objective function. Each individual of the swarm moves according to a strategy that combines different rules: both random steps and movements toward social and individual previous bests.

The objective function is defined as the average square error of the computed outputs with respect to the observed values of acceleration. The algorithm is illustrated in the following, by using the matrix notation, where the symbol ($\times$) represents the matrix product.

Given the data set of observed input $\mathbf{I}$ and output $\mathbf{U}$:

Initialize a set of p individuals with random weights $\{\mathbf{A}_1, \mathbf{B}_1, \mathbf{A}_2, \mathbf{B}_2, \ldots, \mathbf{A}_p, \mathbf{B}_p\}$
for each iteration k
  for each individual i
    $\mathbf{z}_{i,k} = f(\mathbf{B}_i \times f(\mathbf{A}_i \times \mathbf{I}_k)$
    $q_i = \frac{1}{N}\sum_{k=1}^{N}\left(\mathbf{U}_k - \mathbf{z}_{i,k}\right)^2$
  end
  for each individual i
    find $\{\mathbf{A}°, \mathbf{B}°\}$=argmin$\{q_i\}$
    $\mathbf{A}_i = (1-\beta)\mathbf{A}_i + \beta\mathbf{A}° + \alpha\boldsymbol{\varepsilon}$
    $\mathbf{B}_i = (1-\beta)\mathbf{B}_i + \beta\mathbf{B}° + \alpha\boldsymbol{\varepsilon}$
  end
end

Swarm algorithm always exhibited a regular convergence and found out better solutions than the traditional EBP algorithm (as shown in Fig. 7). This is probably because Swarm algorithm, based on a continuous stochastic optimization, ensures a smoother progression, while EBP, based on gradient computation, may have an oscillatory behavior in case of a highly non-convex objective function. Swarm algorithm resulted also more robust with respect to the number of individuals and the values of the coefficients α and β.

Figures below show the trend of the algorithm for different values of random movement (α), approaching speed to the global optimum solution (β) and initial population (p).

The algorithm converges as quickly as the values of factors grow; unlikely the Error Back propagation, in all tests Swarm algorithm converges to the same value 0.054 of the error.

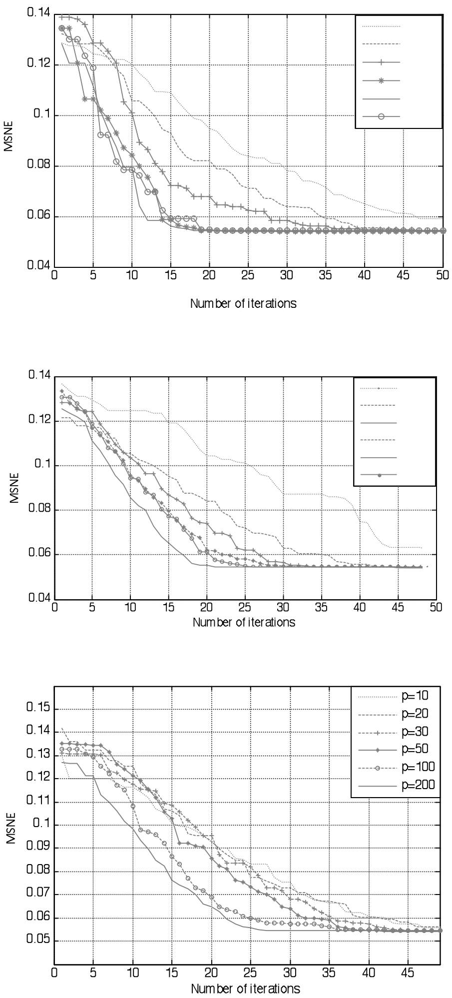

Fig. 7 – Convergence of learning process of a FFW ANN by a swarm algorithm for different values of random movement (α), approaching speed to global optimum solution (β) and initial population (p).

VALIDATION AND APPLICATION IN SIMULATION

Different ANN specifications were validated in two different ways. In the first, as usual, a part of data was used in the learning phase and the remaining for validation. The second kind of test consisted in applying the ANN to simulation, taking the trajectory of first vehicle as given. Unlike the former, where all the inputs of the ANN are given, at each step of the latter, the ANN computes acceleration, speed and position of the following vehicle. Thus, relative speed and spacing depends on the previous outputs provided by the ANN.

Fig.8a and Fig.8b depict observed and estimated acceleration obtained in validation by the ANN with 2 hidden neurons, trained by swarm and EBP algorithm, respectively, on a data set of a 55s that were not used for training. In both cases, results are qualitative acceptable. EBP trained ANN reproduces quite well the acceleration phase, but underestimates significantly one deceleration phase. However, swarm trained ANN provides in any case a quite good approximation of observed data, without large errors. Finally, the so trained ANNs have been validated in a preliminary simulation test.

Inputs of the application were: the observed leader's acceleration; the initial spacing between leader and follower; the initial speed of the follower. ANN had been applied to compute follower acceleration depending on spacing and relative speed, calculated 1s before. Spacing and speed of follower were calculated applying kinematic equations. To avoid too short spacing a minimum value of 4.4 m has been imposed.

Fig.9a plots the observed space trajectories of a leader-follower pair and the corresponding follower's trajectory simulated by the ANN. It is worth noting the good correspondence between observed and simulated follower trajectories. The impact of follower reaction on vehicle trajectories can be appreciated in Fig. 9b, which represents a platoon of vehicles. It is possible to identify the kinematic waves progression corresponding to changes of speed of the leader that propagate in the traffic stream.

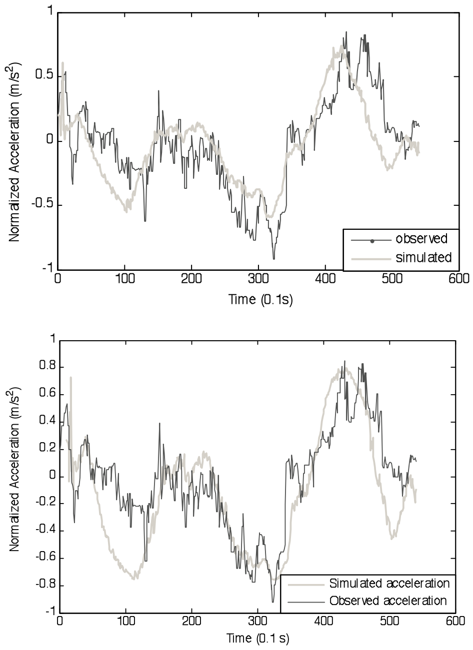

Fig. 9 – Results of validation of FFW ANN trained by Swarm (above) and EBP (bottom) algorithm.

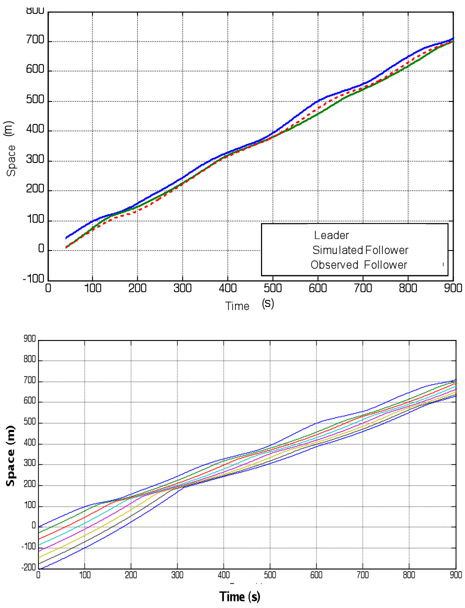

Fig. 11 – Leader and follower observed and follower simulated space trajectories (above) and simulated trajectories of a platoon of vehicles (bottom).

CONCLUSIONS AND FURTHER DEVELOPMENTS

Different formulations and training algorithms for ANN of car following models have been presented. Preliminary tests highlight that the best trade-off between efficiency and complexity is provided by a single feed-forward ANN having 5 neurons in the hidden layer and taking relative velocity and spacing as inputs, trained by a swarm algorithm.

Simulation of a short platoon provides a first exemplification of results achievable by implementing the ANN model into a microsimulation model. In the present case, all vehicles have the same characteristics. However, one can envision training different neural networks to reproduce the driving behavior of different people and then simulate traffic dynamic in a more realistic way.

The calibration procedure that involves virtual driving simulator is rather general and can be applied to different contexts without requiring complex experiments of tracked vehicles. Moreover, it can be useful to simulate some technical measures that act on the driver behavior (for example, driver attention observations) and that cannot be estimated elsewhere.

The research is still ongoing and moves in different directions. One concerns a more complex framework that applies neuro-fuzzy rules to process inputs for a multi-regime ANN formulation that applies different ANNs to different driving tasks.

Another issue worth of further research is to assess a formulation that supplies follower's speed in output. Even if it may lead to unfeasible values of acceleration, and then it is less suitable for applications to Advanced Driving Assistance Systems, it is simpler to implement and is expected providing more accurate estimates of speed and positions. Finally, another line of research consists of looking for a more compact formulation, obtained for example from a suitable combination of input variables. Interesting results were obtained by expressing relative speed and spacing in polar coordinates, where acceleration results uncorrelated with the anomaly and then can be expressed as a function of the only radius.

REFERENCES

Bifulco, G.N., Pariota, L., Simonelli, F., and Di Pace, R. (2010). Development and testing of a fully Adaptive Cruise Control system. In Fusco G. (ed.) Proc. of Int. Conf. *Models and Technologies for ITS*. Roma, 22-24 June 2009. Aracne Publ., Italy.

Brockfeld, E., Kuhne, R.D., and Wagner, P. (2005). Calibration and validation of microscopic traffic flow models. *Transp. Res. Rec*., **1934**, pp.179-187.

Chandler, R.E., Herman, R., and Montroll, E.W. (1957). Traffic dynamics: studies in car following. *Op. Res*, **6**(2), pp. 165-184.

Cipriani, E., Colombaroni, C., Fusco, G., and Mannini, L. (2010). Use of GPS Data for the Analysis of Vehicular Interaction Models. *Int. Conf. Does Traffic Data Support Traffic Models?* Annecy, France, 6-9 July 2010.

Gazis, D.C., Herman, R.L, Rothery, R. W. (1961). Non linear follow the leader models of traffic flow. *Op. Res*., **9**(4), pp. 545-567.

Gipps, P.G. (1981). A Behavioral Car Following Model for Computer Simulation. *Transpn. Res. B*, **15**, pp.105-111.

Kometani, E. and Sasaki, T. (1959). Dynamic behaviour of traffic with a non-linear spacing–speed relationship. in Herman R. (ed.) Proc. of Symp. *Theory of Traffic Flow*, New York, pp. 105–119.

Leutzbach, W. and Wiedemann, R. (1986). Development and applications of traffic simulation models at the Karlsruhe Institut fur Verkehrwessen, *Traffic Eng. and Control*, **27**(5), pp. 270–278.

McCulloch, W.S. and Pitts, W.S. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. of Math. Bioph*. **5**, pp.115-330.

Ossen, S. and Hoogendoorn, S.P. (2004). Car following behavior analysis from microscopic trajectory data. *Transp Res. Rec*., **1934**, pp. 13-21.

Panwai, S. and Dia, H. (2007). Neural Agent Car-Following Models. *IEEE Trans. on Intell. Transp. Syst*., **8**(1).

Punzo, V. and Simonelli, F. (2007). Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Transp. Res. Rec*., **1934**, pp. 53-63.

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representation by back-propagating error. *Nature*, **323**.

Simonelli, F., Punzo, V., Bifulco G. N., and De Martinis, V. (2009). Human-like Adaptive Cruise Control System through a learning machine approach, in *Applications of Soft Computing*, **52**.

Viti, F., Hoogendoorn, S. P., Alkim, T. P., and Bootsma, G. (2008), Driving behavior interaction with ACC: results from a Field Operational Test in the Netherlands, in *2008 IEEE Intelligent Vehicle Symposium*, pp. 745-750.

Wu, J., Brackstone, M., and McDonald, M. (2003). The validation of a microscopic simulation model: a methodological case study. *Transpn. Res. C*, **11**(6), pp. 463-479.

Zhang, L., Wang, J., Li, K., Yamamura, T., Kuge, N. and Nakagawa, T. (2007). Driver Car-Following Behavior modeling using Neural Network Based on Real Traffic Experimental Data. *15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting*. New York.