Paper No. 01-3460

Implementing Adaptive Driving Systems for Intelligent Vehicles by Using Neuro-Fuzzy Networks

Y. T. Lin, F.-Y. Wang, P. B. Mirchandani, Long Wu, Z. X. Wang, Chris Yeo, and Michael Do

The application of supervised learning to train an intelligent vehicle with a neuro-fuzzy controller to mimic the driving behavior of a human driver is discussed. An initial fuzzy control system for vehicle driving was set up on the basis of general human driving experiences, and its control rules were modified to fit the driving behavior of an individual driver. This provides an effective mechanism to construct driving control systems with personality for automated intelligent vehicles.

Intelligent transportation systems (ITS) have been introduced to mitigate the delays and safety problems associated with the continuously increasing traffic congestion on national highways. During the last few years, significant resources and effort have been expended on feasibility studies and the development of concepts, architecture, and technologies for ITS.

It has been realized that a one-step deployment of ITS is both impractical and extremely expensive, and thus the emphasis has shifted to the development of technologies that can be incrementally introduced and tested. The technologies would be helpful now and fully beneficial when a complete ITS is deployed, where platoons of vehicles would travel from point to point at high speeds with little or no interactions with the drivers (unless they are getting on or off the system).

One important technology that is under incremental deployment is automated vehicle control. The idea is to put as much intelligence in the car as possible so that it is semiautonomous, requiring little instruction from or interaction with the driver of the vehicle.

Automated vehicle control systems have two basic functions: to keep the vehicle in the middle of its lane (lateral control) and to accelerate or decelerate to maintain a desired speed or a safe distance between vehicles (longitudinal control). Up until now, most research in automated vehicle control has applied model-based conventional control methods, such as a proportional-integral-differential (PID) feedback sliding model (*1–11*). These methods have several drawbacks. First, conventional control techniques require accurate vehicle dynamic parameters that are often very difficult, if not impossible, to obtain. Second, the data from onboard sensors are unavoidably noisy, which makes it difficult for a PID-type controller to generate accurate control commands. Finally, the longitudinal velocity of vehicles varies in a wide range, making the performance of the technique of gain scheduling, which is only suitable around a specific equilibrium point, unacceptable (*2, 12*).

Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85721.

In recent years, people in scientific and industrial fields have shown increasing interest in designing intelligent control systems that combine fuzzy logic and neural networks, and many implementations integrating fuzzy logic and neural networks have been proposed (*13–16*). Since 1990, the researchers have been attempting to implement fuzzy logic control systems (FLCS) with a neural network so that the knowledge structure of the FLCS is fully preserved in its network implementation (*17–20*). The unique knowledge structure embedded in this structured network enables it to carry out adaptive changes of fuzzy reasoning methods and membership functions for both input signal patterns and output control actions, and then recover these changes individually and completely later from its subnetworks. Gradient methods for optimization have been used to derive off-line training rules and online learning algorithms for the structured neuro-fuzzy network (NFN) (*21–25;* F. Wang, Network-Based Neuro-Fuzzy Control Technology: Hardware and Software Systems, U.S. patent pending, 1999).

VEHICLES WITH INTELLIGENT SYSTEMS FOR TRANSPORT AUTOMATION PROJECT AND AUTOMATED DRIVING BEHAVIOR WITH PERSONALITY

The Vehicles with Intelligent Systems for Transport Automation (VISTA) project was sponsored by Arizona in 1998 to perform research in intelligent vehicle and highway systems. The mission of the project is to develop an affordable intelligent vehicle that can be deployed within the next 5 to 10 years in the proposed intelligent express lanes on I-10 between Phoenix and Tucson.

The VISTA project focuses on the use of a hierarchical control platform that requires (*a*) less frequent and less spatially dense communication among the road, the traffic operations center, and the vehicles and (*b*) less computational effort for lateral and longitudinal control of the vehicle on the highway. To achieve these goals, three new techniques have been developed in the VISTA project:

1. The use of calibration-based vehicle control, rather than guidance-based vehicle control, reduces the cost of constructing and maintaining roadside sensors.
2. Trajectory planning and optimization based on long-range road information reduce the level of energy consumption and air pollution and increase the traffic throughput.
3. Distributed hierarchical agent-based control is used rather than traditional functional decomposition into sensing, planning,

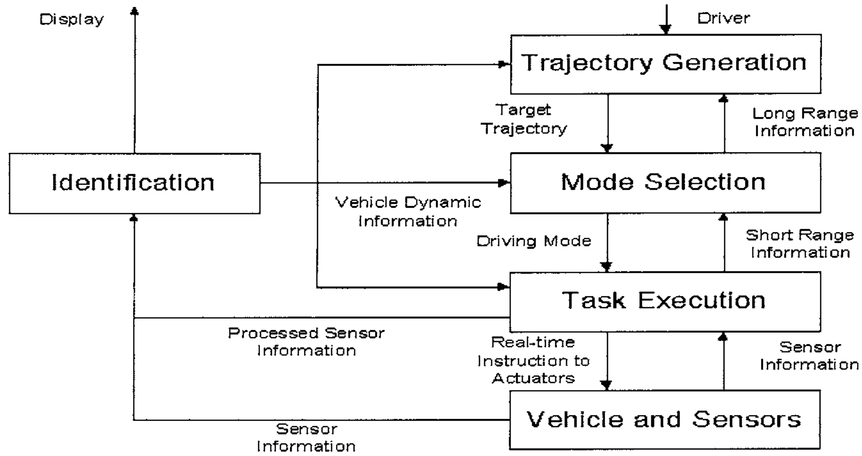


FIGURE 1 Hierarchical structure of VISTA control system.

and acting. Vehicle control systems are decomposed into hierarchically organized special-purpose task-achieving modules, called agent programs. Fuzzy logic–based driving agent programs (from human driving behaviors) are used for many modules. Figure 1 shows the control structure of the VISTA control system.

FUZZY LOGIC–BASED KNOWLEDGE FOR LONGITUDINAL VEHICLE MOTION CONTROL

A human driver generally controls the longitudinal motion of a vehicle on the basis of an estimation of the distance between the controlled vehicle and the preceding vehicle and their relative speed. If the preceding vehicle is far ahead, the vehicle will be driven at a speed specified for the section of the highway. If the preceding vehicle is nearby, the driver will try to maintain a predetermined spacing between the vehicles. If the spacing between the vehicles decreases too rapidly, the driver will apply emergency braking. A human driver changes the running state of a vehicle by pushing down the throttle or brake pedal, which in turn produces forward or backward force to accelerate or decelerate the vehicle. If the vehicle needs a quick acceleration, the throttle pedal will be pushed down hard; if the vehicle needs a quick deceleration, the brake pedal will be pushed down hard. This driving knowledge can be described with fuzzy logic. The distance between the controlled vehicle and its preceding vehicle and the relative speed between the two vehicles (the rate of distance change) are taken as two input signals. The throttle/braking force is taken as output control. These variables are first transferred into seven linguistic terms: negative big, negative medium, negative small, zero, positive small, positive medium, and positive big (NB, NM, NS, ZE, PS, PM, PB). With these fuzzy terms, the human driver's experience can be expressed in a form such as the following:

If distance is NS and relative speed is PB, then apply a force of PM.

Corresponding to the seven terms of distance and seven terms of relative speed, 49 fuzzy rules can be constructed (see Table 1, where $\hat{d}$ and $\hat{v}$ are the fuzzified distance and relative speed between the controlled vehicle and the preceding vehicle, respectively).

With this rule base, a fuzzy controller can be built. Input signals are first linearly normalized into values in the range [–3, 3], then transferred into fuzzy terms. Figure 2 shows a mapping under the assumptions that the desired distance between the controlled vehicle and the preceding vehicle is 10 m, the distance can vary between 5.5 and 14.5 m, the desired relative speed between the two vehicles is 0 m/s, and the relative speed can vary between –4.5 and 4.5 m/s.

The output of the fuzzy logic controller is a crisp force F. When $F > 0$, propulsion force is applied by increasing the throttle angle. When $F < 0$, braking force is applied by pushing the brake pedal down. Because of the high nonlinearity of the vehicle dynamic characteristics, an accurate relationship between the throttle/brake positions and the forward/backward force cannot be specified. However, it is known that bigger throttle control values result in bigger accelerating forces and that bigger brake control values result in bigger decelerating forces. This common sense is applied to generate the throttle/brake control command on the basis of the crisp force F.

TABLE 1 Fuzzy Rule Base of the Vehicle Controller

$\hat{v}$ \ $\hat{d}$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NM	NM	NS	ZE	PS	PM
ZE	NS	NS	NS	ZE	PS	PM	PB
PS	NS	NS	ZE	PS	PM	PB	PB
PM	ZE	ZE	PS	PM	PB	PB	PB
PB	PS	PS	PM	PB	PB	PB	PB

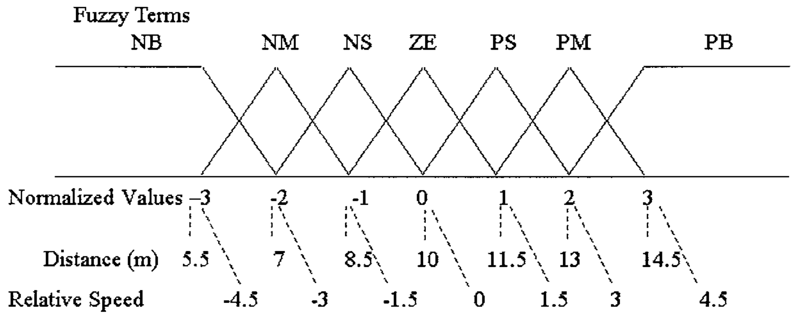


FIGURE 2 Mapping from measured input signal into linguistic terms.

SUPERVISED LEARNING OF LONGITUDINAL DRIVING BEHAVIORS

Neuro-Fuzzy Networks

Figure 3 shows the structure of an NFN implementation of a set of fuzzy logic decision rules. The network consists of three subnetworks of distinctive functionalities: pattern recognition (PR), fuzzy reasoning (FR), and control synthesis (CS). The PR subnetwork identifies the patterns of input variables according to the membership functions of linguistic terms. The FR subnetwork conducts fuzzy reasoning (conjunction) by calculating the firing strength of each decision rule. The CS subnetwork carries out the task of control synthesis by generating fuzzy control action and then defuzzifying it. Although the three neural networks are connected sequentially, the construction and training of these networks can be performed independently and simultaneously, and the procedure in the fuzzy logic–based decision making is fully preserved in its network implementation.

PR Neural Networks

For each signal reading s_i, a neural network SN_i is constructed to match its values with the linguistic terms in the set of signal patterns A_i. In other words, the functionality of SN_i is to calculate membership functions $\mu_{s_i^k}(x)$ for $k = 1, \ldots, p_i$, $i = 1, \ldots, m$. Figure 3 shows a three-layer SN_i. Initially, this network is trained with the specified membership functions for terms in A_i. At this stage, network SN_i is not required to learn the membership functions at high accuracy since the specified membership functions are usually subjective. Note that if two sensor readings have identical sets of linguistic terms, they can use the same network at the beginning. Through network learning, the membership functions of linguistic terms can be changed adaptively later for better performance.

FR Neural Networks

For each decision rule r in the knowledge base, a neural network RN_r, $r = 1, \ldots, R$ is used to calculate the firing strength of the rule. Thus, RN_r is actually a network implementation of conjunction operator. Figure 3 shows a three-layer RN_r. By changing its weights, it can be made to implement various logic operations approximately. Therefore the initial training of RN_r can be carried out by using any of these norms, or even their combinations, and the network can be easily modified for new fuzzy reasoning by the use of learning algorithms. Clearly, as long as every rule has the same number of linguistic terms in its precondition, the same FR network can be chosen for all the control rules at the initial stage. In this paper, the algebraic product is used for initial training.

CS Neural Networks

Control synthesis is the process of determining the final crisp control according to the firing strengths of rules and membership functions

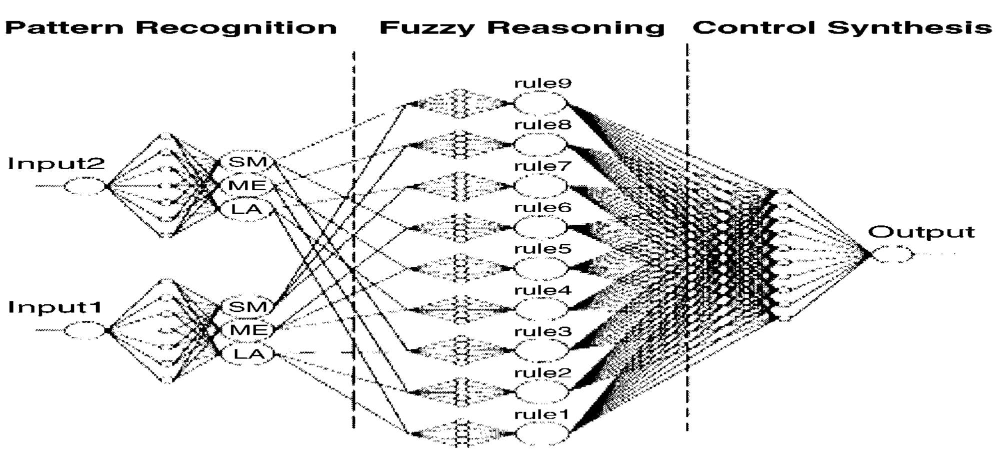


FIGURE 3 Subnetworks and integration of NFNs.

of linguistic terms defined for control actions. This involves deducing consequences for individual rules, generating resultant fuzzy control, and then converting it into a crisp value. Figure 3 illustrates a two-layer neural network CN_j for the synthesis of control component u_j, $j = 1, \ldots, n$. The first layer is introduced for calculating fuzzy controls. In this layer, a neuron is created for each of the elements in the universe of discourse U_j. Given the firing strengths of control rules, neuron k produces the value of the membership function of the resultant control at a specific u_{jk}, $1 \le k \le n_j$. Note that the algebraic product operator has been used for conjunction in rule consequence deduction and the bounded sum for disjunction in the resultant fuzzy controls. Since the logic operations have been fixed, there is no need for initial network training. The initial weights of network CN_j at this layer can be calculated from membership functions as

$$w_{jkr} = \mu_{U_j^{Cjr}}(u_{jk}) \qquad k = 1, \ldots, n_j \tag{1}$$

while the bounded sum is implemented by a linear activation function $f(x) = x$, if $0 \le x \le 1$; $f(x) = 0$, if $x < 0$; and $f(x) = 1$, if $x > 1$. The second layer carries out the task of defuzzification. Initial values of weights in this layer can be determined according to the center of area. Thus, weights of the second layer are given by

$$\gamma_{jk} = u_{jk}/\sigma_j \qquad \sigma_j = \sum_{k=1}^{n_j} \mu_{jk} \qquad u_j = \sum_{k=1}^{n_j} \gamma_{jk}\mu_{jk} \qquad j = 1, \ldots, n \tag{2}$$

where μ_{jk} is the value of neuron k at the first layer. These initial weights of networks CN_j, $j = 1, \ldots, n$, can be changed later by learning algorithms to improve control performance. However, learning will only change the membership functions of control actions and the defuzzification algorithm, not the logic operations involved in control synthesis.

Integration of Neural Networks: NFNs

Once SN_i, RN_r, and CN_j have been created, the final step toward a structured NFN is to connect those networks appropriately according to the original set of fuzzy decision rules. Figure 3 also presents the integrated NFNs.

From Fuzzy Control Rules to the NFN

For vehicle longitudinal control, the NFN has two PR subnetworks, 49 FR subnetworks, and one CS subnetwork. They are built and trained separately.

Each of the *SN*s has one input process element (PE) and seven output PEs, corresponding to seven linguistic terms of input signal. Several pairs of distance value/membership value of linguistic terms are used for training the network. The numbers of hidden layers and hidden PEs are determined during the training process. Once this network is trained, it is cloned into two, one for converting the distance and the other for converting the relative speed into membership value of linguistic terms.

Each of the 49 *RN*s has two input PEs and one output PE. The two input PEs correspond to the membership values of a linguistic term of distance and a linguistic term of relative speed, respectively. The output PE corresponds to the firing strength of a fuzzy rule. Several pairs of membership value of distance, relative speed linguistic terms, and resulting firing strength of a fuzzy rule are used for training the network and determining the number of hidden layers and hidden PEs. Once the network is trained, it is cloned into 49, each corresponding to a fuzzy rule.

The *CN* has 49 input PEs, each corresponding to the firing strength of a fuzzy rule; 7 hidden PEs, each corresponding to the membership value of a linguistic term of the fuzzy output force; and 1 output PE, which gives the crisp control force value. In any time step, most fuzzy rules have a firing strength of 0 except for those that are really triggered. By integrating these subnetworks, an NFN for vehicle longitudinal control can be obtained.

Supervised Learning Using Error Backpropagation

The backpropagation learning algorithm developed for standard multilayer feed-forward neural networks can be easily generalized to an NFN. As for multilayer neural networks, this will enable the NFN to tune its weights to match the recorded longitudinal driving behavior of individual drivers. To find the supervised learning algorithm for the NFN, the error function is defined as

$$e = \frac{1}{2}\sum_{j=1}^{n} \left\| u_j^d - u_j \right\|^2 \tag{3}$$

where u_j^d is the recorded throttle and braking motor control value. If only initial values of weights γ_{jk} are to be calculated, the rules for updating weights of the NFN are exactly the same as those for the standard multilayer neural networks, except that weights between any two neurons with no direct connection are treated as zero. This will change the defuzzification algorithm through training. Otherwise, the rule for updating weight ω_{jkr} in the first layer of CN_j has to be modified as follows:

$$\omega_{jkr}(t+1) = \omega_{jkr}(t) + \eta\alpha_r(u_j^d - u_j)\left(\gamma_{jk} - \sum_{l=1}^{n_j} u_{jl}\mu_{jl}/\sigma_j^2\right) f'(\mu_{jk}) \tag{4}$$

where

$f'(x)$ = 1 when $0 < x < 1$ and $f'(x) = 0$ otherwise;
η = learning rate, with $0 < \eta < 1$; and
t = number of iteration steps in training.

This result can easily be obtained from gradient calculation. In this case, the error term δ_r backpropagated to the output neuron of RN_r is found to be

$$\delta_r = \sum_{j=1}^{n} (u_j^d - u_j)\frac{\partial u_j}{\partial \alpha_r}$$

$$\frac{\partial u_j}{\partial \alpha_r} = \sum_{k=1}^{n_j} \varpi_{jkr}\left(\gamma_{jk} - \sum_{l=1}^{n_j} u_{jl}\mu_{jl}/\sigma_j^2\right) f'(\mu_{jk}) \tag{5}$$

After supervised learning has been completed, membership functions and fuzzy conjunction operators can be recovered by breaking the NFN into subnetworks of pattern recognition, fuzzy reasoning, and control synthesis. Specifically, from SN_i the refined membership functions of signal patterns for s_i are obtained, and from RN_r the modified conjunction operator for rule r is obtained. Note that, after learning, different control rules would have different conjunction operators. To obtain the updated membership functions of control actions for u_j, only one input neuron, say α_r, of CN_j is set to 1, and all others to zero. In this way, output values of neurons in the first layer of CN_j present the new membership function for control term u_j^{cjr}. Like conjunction operators, a fuzzy control action used by two or more control rules could have different membership functions in different rules after training.

EXPERIMENTAL RESULTS

Experiments were conducted on a test vehicle (Figure 4) with the neuro-fuzzy controller described in the preceding section. The aim of these experiments is to examine the controller's performance and the result of supervised learning.

The test vehicle is a 1989 Chevrolet Celebrity station wagon. Three experiments were conducted. The set rule for the experiments is to keep the distance between the controlled vehicle and the preceding vehicle at 10 m. A human driver manipulates the preceding vehicle.

Experiment 1

The vehicle is controlled by the controller that has not been trained after integration. Figure 5 shows the recorded throttle/brake control values and the distance between the two vehicles and their relative speed. The distance between the two vehicles fluctuated widely, though it basically remained around the desired value, with the shortest distance being 8.0 m and the longest being 13.2 m. The performance of this controller is similar to the basic fuzzy controller: both behave like a new driver who understands driving principles but lacks hands-on experience and therefore cannot manipulate a vehicle gracefully.

Experiment 2

The vehicle is controlled by a human driver who tried to keep the distance between the two vehicles at 10 m. Figure 6 shows that the average distance (10.45 m) between the two vehicles was a little above the desired value. The reason may be that a human driver cannot measure the distance as accurately as it can be measured by radar. However, the distance fluctuation is much smaller than in the first test. Figure 6*a* demonstrates that the human driver maintains the distance by making minor frequent adjustments of the throttle/brake control value.

Experiment 3

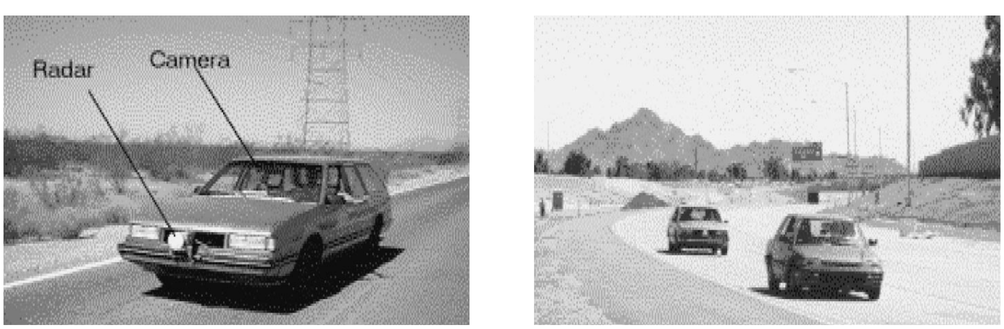
The neuro-fuzzy controller is trained with a supervised learning method on the basis of the data recorded in Experiment 2. Figure 7 shows that the distance between the two vehicles is kept at around 10 m, with the shortest distance being 8.3 m and the longest distance being 11.8 m. The fluctuation is much smaller than was achieved in Experiment 1 (before the controller is trained). At the same time, the relative speed is basically between –5 and 5 m/s. Figure 7*a* shows that the neuro-fuzzy controller also manipulates the vehicle with a minor degree of adjustment to the throttle/brake control value. This indicates that a neuro-fuzzy controller can acquire at least part of a human driver's driving behavior through supervised learning.

The intrinsic reason for performance improvement of the neuro-fuzzy controller can be found by decomposing the NFN into three subnetworks and recovering the refined membership functions and modified fuzzy conjunction operators.

Figure 8*a* compares the original and refined membership functions of distance. It can be seen that after supervised learning, the boundaries between the different linguistic terms have become blurred. The range of each membership function becomes wider than before and thus overlaps with neighboring fuzzy terms. The fuzzy terms NM and NS coincided for the most part, as did PM and PS. This indicates that a human cannot easily distinguish NM from NS, or PM from PS. The same phenomenon can be found in Figure 8*b*, where the original and refined membership functions of relative speed are compared. Figure 8*b* shows that a human driver is more sensitive to the decrease of relative speed than to the increase of relative speed, which can be verified from the fact that the revised membership functions are shifted slightly to the right.

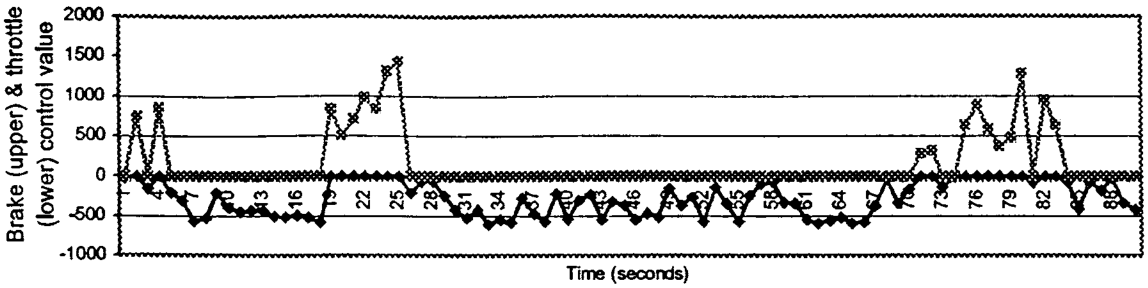
Figure 9 illustrates the modified fuzzy conjunction operator for Rules 1 and 24. In the original neuro-fuzzy controller, every rule applies the algebraic product operator. However, it can be seen that after training, Rule 1 still applies the algebraic product operator, while Rule 24 changes to the intersection operator. Therefore, the firing strength of Rule 24 is increased. This shows that after training, the weight of Rule 24 (and other rules that produce mild output) becomes bigger than before.

Figure 10 shows the recovered membership function of the CS neural network. It can be seen that after training, the range of each fuzzy term (except ZE) is shifted toward the middle. This indicates that for a given control output term, the controller produces a crisp

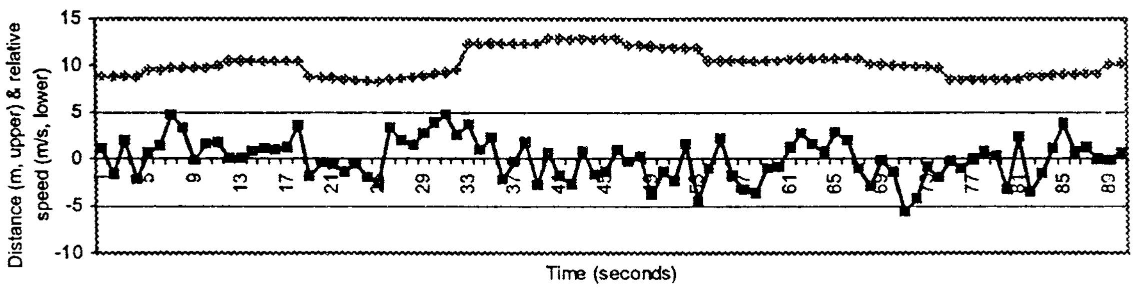


(a) (b)

FIGURE 4 Performance of the initial neuro-fuzzy controller: (*a*) VISTA vehicle; (*b*) demonstration of autonomous control of VISTA car at Highway 51 on April 27, 1999.

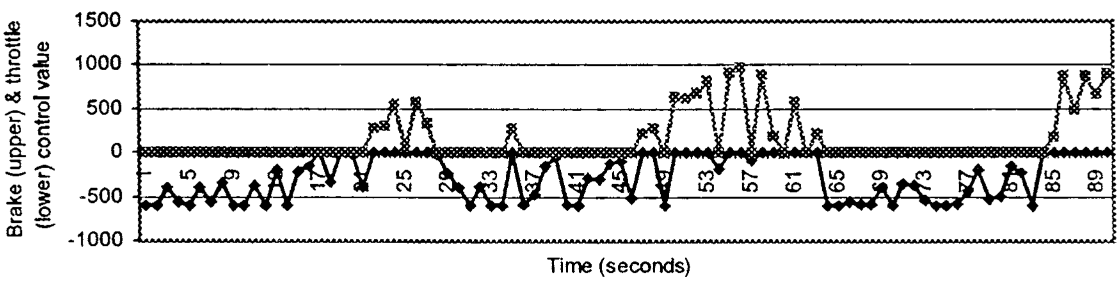


(a)

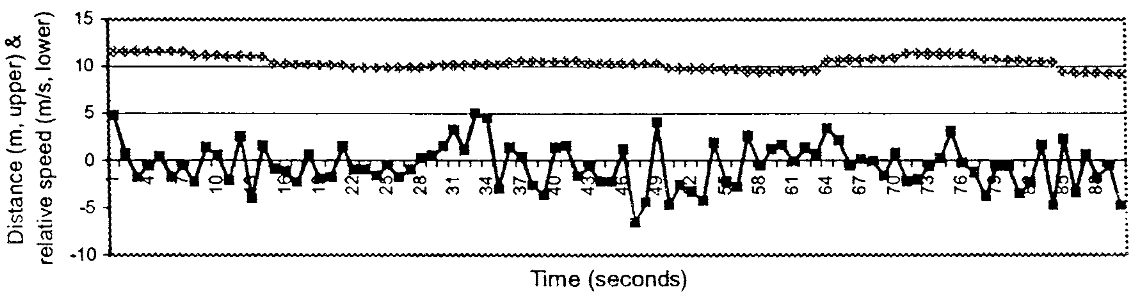


(b)

FIGURE 5 Performance of the initial neuro-fuzzy controller: (*a*) brake and throttle control value; (*b*) distance and relative speed.



(a)



(b)

FIGURE 6 Performance of human driver in vehicle longitudinal control: (*a*) brake and throttle control value; (*b*) distance and relative speed.

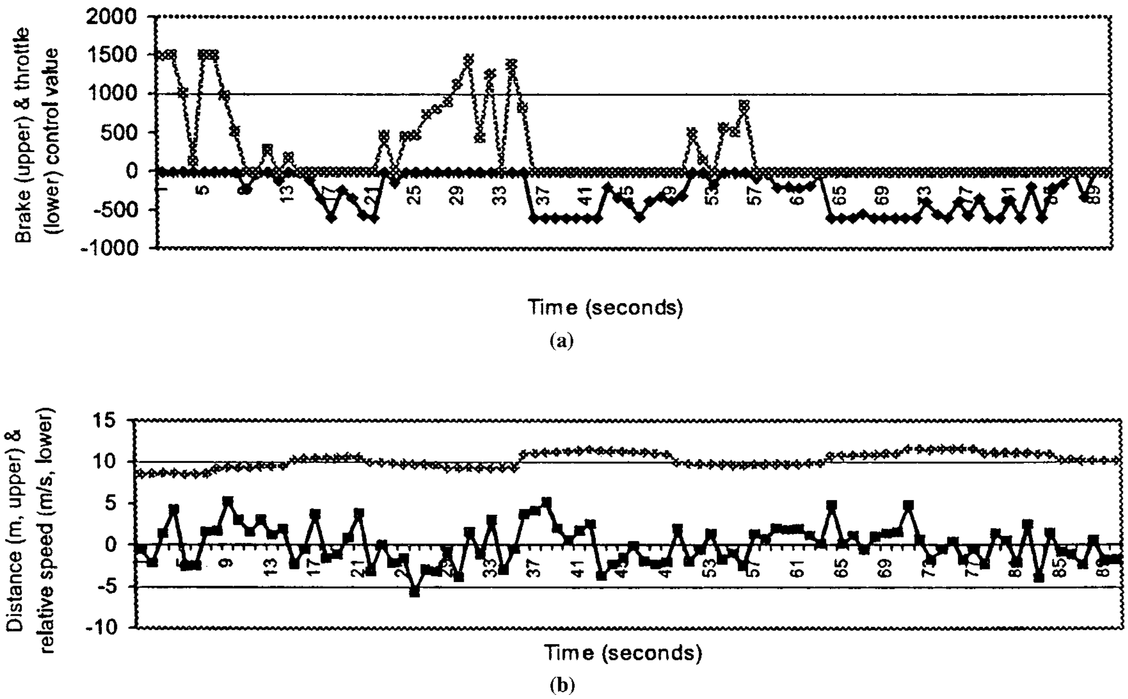


FIGURE 7 Performance of the neuro-fuzzy controller after supervised learning: (*a*) brake and throttle control value; (*b*) distance and relative speed.

output value smaller than that produced before training. This partially explains why and how the trained controller improved its performance.

The revision documented here is only for a specific driver. Different drivers with different driving habits or behaviors will cause different revisions. In this way, the automated driving system could acquire a personality from an individual human driver.

CONCLUSION AND FUTURE DIRECTIONS

The research presented here shows that a neuro-fuzzy controller for an automated vehicle can be trained with data derived during the driving process of a human driver. The advantage of this method is that it can improve the performance of existing neuro-fuzzy controllers and keep the original knowledge structure, which makes it much

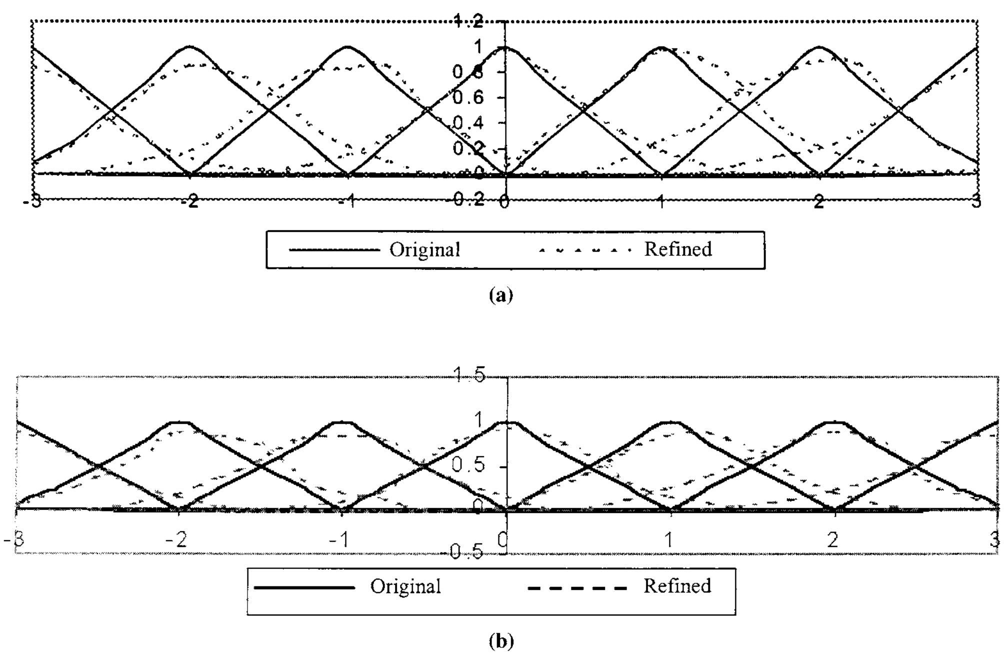


FIGURE 8 Recovered membership function of PR subnetwork: (*a*) membership function of distance; (*b*) membership function of relative speed.

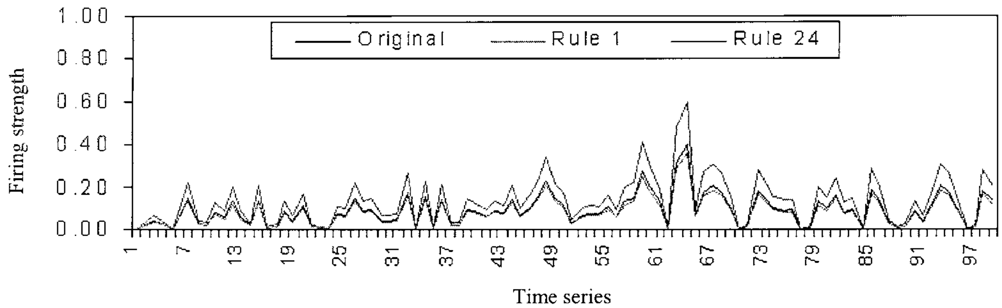


FIGURE 9 Recovered fuzzy conjunction operators of FR subnetwork.

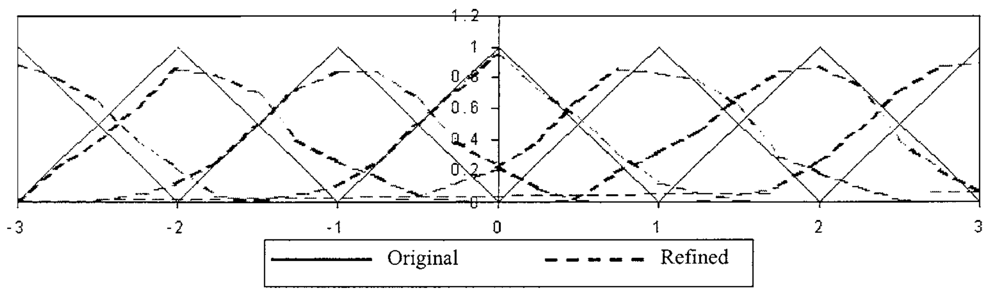


FIGURE 10 Recovered membership function of CS subnetwork.

easier for people to analyze how and why the performance of the controller changes after training.

REFERENCES

1. McMahon, D., V. K. Narendran, D. Swaroop, J. Hedric, K. S. Chang, and P. E. Devlin. Longitudinal Vehicle Controllers for IVHS: Theory and Experiment. *Proc., American Control Conference,* Chicago, Ill., June 1992, pp. 1753–1757.
2. Hsu, A., S. Sachs, F. Eskafi, and P. Varaiya. The Design of Platoon Maneuvers for IVHS. *Proc., American Control Conference,* Boston, Mass., June 1991, pp. 2545–2550.
3. Hedrick, J. K., D. McMahon, V. Narendran, and D. Swaroop. Longitudinal Vehicle Controller Design for IVHS System. *Proc., American Control Conference,* Boston, Mass., June 1991, pp. 3107–3112.
4. Swaroop, D., and J. K. Hedric. Direct Adaptive Longitudinal Control of Vehicle Platoons. *Proc., 33rd Conference on Decision and Control,* Lake Buena Vista, Fla., Dec. 1994, pp. 684–689.
5. Connolly, T. R., and J. K. Hedrick. Longitudinal Transition Maneuvers in an Automated Highway System. *Proceedings of the ASME Dynamic Systems and Control Division,* Vol. 58, 1996, pp. 25–32.
6. Omae, M., and T. Fujioka. Longitudinal Control of Platoon Treated as One Dynamic System. *Proceedings of the ASME Dynamic Systems and Control Division,* Vol. 64, 1998, pp. 691–703.
7. Li, P., L. Alvarez, R. Horowitz, P. Chen, and J. Carbaugh. A Safe Velocity Tracking Controller for AHS Platoon Leaders. *IEEE Proc., 35th Conference on Decision and Control,* Kobe, Japan, Dec. 1996, pp. 2283–2288.
8. Sheikholeslam, S., and C. A. Desoer. Longitudinal Control of a Platoon of Vehicles with No Communication of Lead Vehicle Information. *Proc., American Control Conference,* Boston, Mass., June 1991, pp. 3102–3107.
9. Chien, C. C., and P. A. Ioannou. Automatic Vehicle-Following. *Proc., American Control Conference,* Chicago, Ill., June 1992, pp. 1748–1752.
10. Chien, C. C., Y. Zhang, and C. Y. Cheng. Autonomous Intelligent Cruise Control Using Both Front and Back Information for Tight Vehicle Following Maneuvers. *Proc., American Control Conference,* Seattle, Wash., June 1995, pp. 3091–3095.
11. Raza, H., and P. A. Ioannou. Vehicle Following Control Design for Automated Highway Systems. *IEEE Control Systems Magazine,* Vol. 16, No. 6, 1996, pp. 43–60.
12. Shladover, S. E. Vehicle-Follower Control for Dynamic Entrainment of Automated Guideway Transit Vehicles. *Journal of Dynamic Systems, Measurement, and Control,* Vol. 101, No. 6, Dec. 1979, pp. 314–320.
13. Pao, Y. *Adaptive Pattern Recognition and Neural Networks.* Addison-Wesley, Reading, Mass., 1992.
14. Lee, C. C. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, I & II. *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 20, No. 2, 1990, pp. 404–435.
15. Berenji, H. R. Neural Networks and Fuzzy Logic in Intelligent Control. *Proc., IEEE International Symposium on Intelligent Control,* Philadelphia, Pa., 1990, pp. 916–919.
16. Antsaklis, P. Neural Networks in Control Systems. *IEEE Control Systems Magazine,* Vol. 12, No. 2, 1992, pp. 8–10.
17. Werbos, P. Neurocontrol and Fuzzy Logic: Connection and Design. *International Journal of Approximate Reasoning,* No. 6, 1992, pp. 185–219.
18. Wang, F. Building Knowledge Structures into Neural Networks Using Fuzzy Logic. In *Robotics and Manufacturing: Recent Trends* (M. Jamshidi et al., eds.), ASME Press, New York, 1992.
19. Wang, F. Design of Adaptive Fuzzy Control Systems Using Neural Networks. *Proc., 1st Chinese World Congress on Intelligent Control and Intelligent Automation,* Beijing, China, 1993, pp. 516–522.
20. Wang, F., and D. Chen. Learning Laws for Neural Network Implementation of Fuzzy Control Systems. *Proc., IEEE International Conference on Systems, Man, and Cybernetics,* San Antonio, Tex., 1994.
21. Wang, F., H. Kim, and M. Zhou. Dynamic Back Propagation for Neuro-Fuzzy Networks. *Proc., International Conference on Electronics and Information Technology,* Beijing, China, 1994.
22. Wang, F., and H. Kim. Implementing Adaptive Fuzzy Logic Controllers with Neural Networks: A Design Paradigm. *Journal of Intelligent and Fuzzy Systems,* Vol. 3, No. 3, 1995, pp. 165–180.
23. Wang, F., and D. Chen. Rule Generation and Modification for Intelligent Controls Using Fuzzy Logic and Neural Networks. *Journal of Intelligent Automation and Soft Computing,* 1997.
24. Wang, F., Z. Huang, D. Chen, and P. Lever. Refinement and Generation of Decision Rules Through Training and Augmentation of Neural Networks. *International Journal of Intelligent Control and Systems,* Vol. 2, No. 3, 1998, pp. 329–360.
25. Wang, F. *Implementation of Neuro-Fuzzy Networks Using Radial-Basis Function Networks.* RAL Technical Report 10-1. University of Arizona, Tucson, 1997.

Publication of this paper sponsored by Committee on Intelligent Transportation Systems.