A Transportable Neural-Network Approach to Autonomous Vehicle Following

Nasser Kehtarnavaz, *Senior Member, IEEE,* Norman Griswold, *Senior Member, IEEE,* Kelly Miller, and Paul Lescoe

***Abstract*— This paper presents the development and testing of a neural-network module for autonomous vehicle following. Autonomous vehicle following is defined as a vehicle changing its own steering and speed while following a lead vehicle. The strength of the developed controller is that no characterization of the vehicle dynamics is needed to achieve autonomous operation. As a result, it can be transported to any vehicle regardless of the nonlinear and often unobservable dynamics. Data for the range and heading angle of the lead vehicle were collected for various paths while a human driver performed the vehicle following control function. The data was collected for different driving maneuvers including straight paths, lane changing, and right/left turns. Two time-delay backpropagation neural networks were then trained based on the data collected under manual control—one network for speed control and the other for steering control. After training, live vehicle following runs were done under the neural-network control. The results obtained indicate that it is feasible to employ neural networks to perform autonomous vehicle following.**

***Index Terms*—Autonomous vehicle following, intelligent transportation systems, neural-network controller, unmanned vehicles.**

I. Introduction

The LAST decade has witnessed a considerable amount of research on autonomous vehicles. Out of this effort, only a few organizations have implemented automated vehicle control systems in real vehicles, for example, in the United States, Carnegie Mellon University and DARPA's Navlab [1], the University of Maryland and Martin Marietta's Alvin [2], General Motors' Lanlok [3], the PATH program at Berkeley [4], in Japan, Nissan's PVS project [5], and in Germany, the Prometheus VaMoRs project [6].

At Texas A&M University, we have also implemented an automated vehicle control system in a real vehicle. The research at Texas A&M has been done on vehicle following in contrast to most research, which has been performed on road following and collision avoidance. Autonomous vehicle following is defined as a vehicle controlling its own steering and speed while following a lead vehicle within a prescribed safe headway. In our previous papers [7]–[10], the architecture of this autonomous vehicle, named BART, has been explained in detail. BART, an acronym for Binocular Autonomous Research Team, is a Dodge Caravan (see Fig. 1) equipped with: 1) a binocular vision system to obtain the range and heading angle of the lead vehicle; 2) a PC to process data for generating appropriate control commands; and 3) a microprocessor servo system to implement steering, throttle, brakes, and transmission functions.

As shown in Fig. 2, the driving command generator (DCG) module translates a range and heading angle of the lead vehicle into an appropriate control command consisting of a steering and speed value for the autonomous vehicle. Originally, this task was accomplished by a conventional proportional and derivative/proportional and integral (PD/PI) controller, which required the experimental characterization of the nonlinear vehicle dynamics. For example, the relationship between the observed heading angle and the corresponding steering wheel angle was determined via experimentation for various speeds. Although the performance of the autonomous runs was satisfactory, it was desired to develop a control mechanism which was independent of individual and often unobservable nonlinear vehicle dynamics. The motivation was the transportability of such a controller to any vehicle regardless of its dynamics. This transportability attribute reduces the development time and, hence, the cost of a full-scale autonomous convoy following scenarios planned by the Army. A need exists in the military community for the deployment of autonomous vehicle following as part of the Army's unmanned ground vehicle (UGV) program. This need is concerned with the delivery of ammunitions of hazardous materials from one point to another such that the danger to the driver due to accidents is minimized. This scenario includes both on- and off-road terrains.

In this paper, we discuss how such a transportable controller for autonomous vehicle following is achieved through the use of neural networks. Although the neural-network approach has been used for other autonomous land navigation applications, in particular for road following [11]–[13], this work constitutes the first attempt to utilize it for vehicle following.

II. Transportable Neural-Network Approach

Neural networks are model-free paradigms that are capable of learning the nonlinear relationship between input/output vector pairs through examples. In our case, this approach is employed to learn the nonlinear relationships between the observed range and heading angle and the controllable steering wheel angle and speed. The incorporation of a neural network

Manuscript received June 22, 1995; revised October 21, 1996. This work was supported by Redzone Robotics, Inc. and the Army Tank Automotive Command (TACOM) under Contract 9106-PO-119.

N. Kehtarnavaz, N. Griswold, and K. Miller are with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128 USA (e-mail: kehtar@ee.tamu.edu).

P. Lescoe is with the U.S. Army Tank Automotive Command, Warren, MI 48397 USA.

Publisher Item Identifier S 0018-9545(98)00107-8.

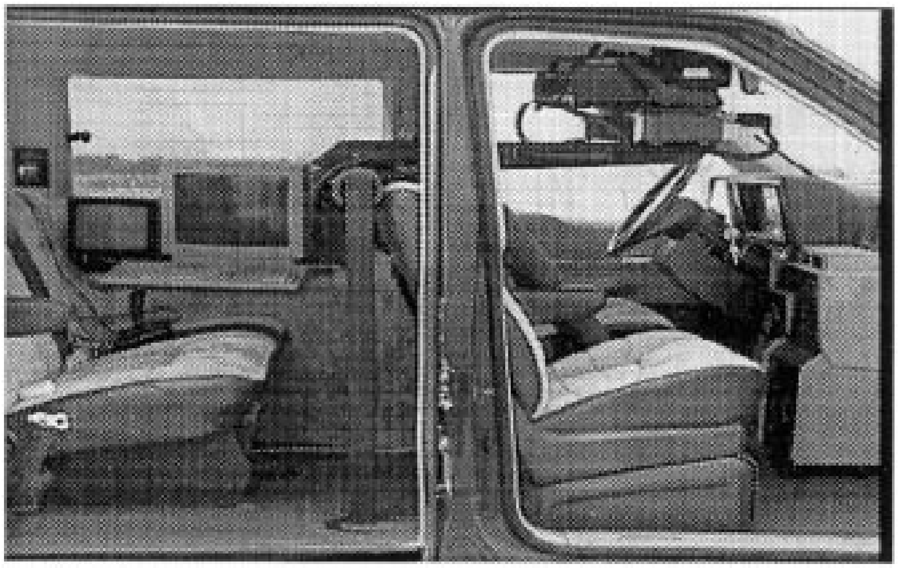


Fig. 1. Inside of BART autonomous vehicle.

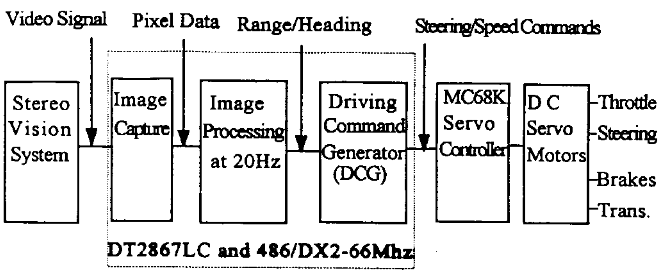


Fig. 2. Block diagram of BART system.

into the vehicle following system is done as follows. A human driver manually drives the vehicle while following a lead vehicle. Range and heading angle data from the vision system (input) and the corresponding commanded speed and steering wheel angle (output) are collected for various maneuvers. Based on the collected input/output vector pairs, two time-delay backpropagation neural networks, one for speed and the other for steering, are trained to learn the human driver's response. After a training period, the control is switched to the autonomous mode or the neural-network module to reproduce the human driver's driving. The neural-network module can be viewed as an autopilot mechanism for performing autonomous vehicle following.

It should be noted that the way human drivers turn the steering wheel and change the throttle is different than the way the servo controller does. If a driver was allowed to drive BART directly without going through the servo system, the data collected would not incorporate the servo dynamics. A joystick was therefore interfaced to the PC to allow the human driver to drive BART through the same servo controller as used by the DCG. The joystick approach was a relatively simple method for modeling the DCG function while going through the servo system. The joystick position was quantized and converted into a drive command that was then issued to the servo microprocessor to change the steering wheel angle and speed. The number and value of the quantized positions were determined through empirical testing to make the joystick manual driving easy for the driver.

III. Data Collection and Conditioning

As illustrated in Fig. 3, the input/output data is stored during the joystick manual driving. The input data consists of range (in feet) and heading angle (in degrees) of the lead vehicle obtained by the stereo vision system. The output data comprises the steering wheel angle (in degrees) and speed (in mph) provided by a so-called drive-by-wire translator. At this point, it should be pointed out that the control cycle in our system cannot be reduced below 0.5 s in order to have stable operation, a constraint imposed by the onboard microprocessor servo subsystem. Therefore, as a result, data samples were collected at a sampling time interval of 0.5 s. All data collection was performed on a runway facility at the Riverside Campus of Texas A&M University. The course of this runway facility is illustrated in Fig. 4.

Since it was neither necessary nor easy to handle both steering and speed control functions at the same time during manual control, two sets of data were collected—one for speed control and the other for steering control. Table I summarizes a list of an extensive data collection run for steering angle control. This data included various steering maneuvers at 15 mph and contained a total of 3005 samples (approximately 25 min). Similarly, Table II summarizes a list of an extensive data collection run for speed control. This data included different speeds along straight paths and contained a total of 1702 samples (approximately 14.5 min).

Once the data was collected, it was required to condition or normalize the data for the purpose of training the backpropagation neural networks. For the speed control network, the range error $\hat{\Delta}d$ was normalized to a value between -1 and $+1$ as follows:

$$\hat{\Delta}d = \frac{d - d_f}{s} \tag{1}$$

TABLE I
A LIST OF DATA COLLECTION TRIALS FOR STEERING CONTROL

Trial Number	Lead Vehicle Actions	Number of Samples
1	Lane changing between A and B	294
2	Lane changing between A and B	257
3	Lane changing between A and B	210
4	Lane changing between A and B	230
5	Lane changing between A and B, 3 right turns between B and F	513
6	3 left turns between F and B, lane changing between B and A	537
7	Lane changing between A and B, 3 right turns between B and F	456
8	3 left turns between F and B, lane changing between B and A	508

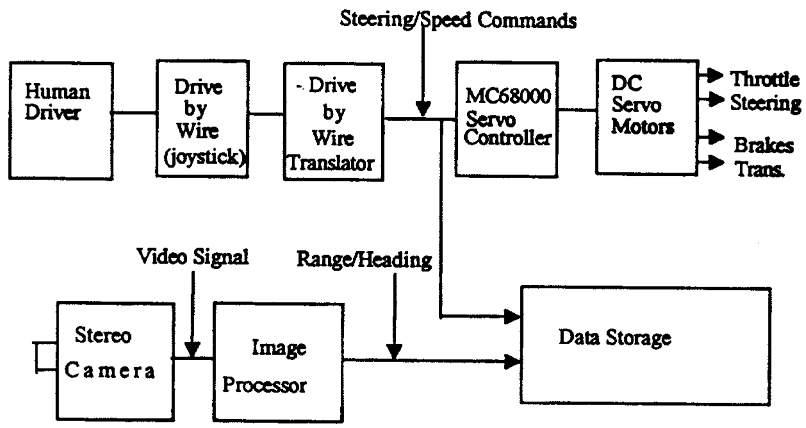


Fig. 3. Block diagram of manual control set up during data collection.

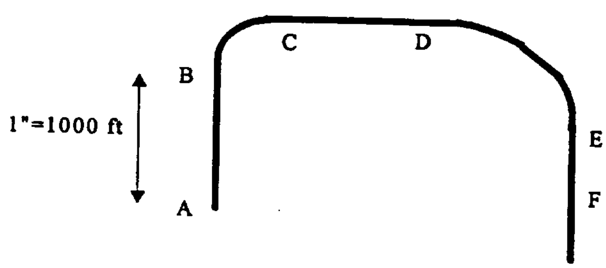


Fig. 4. Runway course used for data collection.

where d denotes the range, d_f the follow distance, s the difference between the maximum valid range $d_{\max}$ and d_f for $d > d_f$, and the difference between d_f and the minimum valid range $d_{\min}$ for $d < d_f$. The speed was also normalized by

$$\hat{v} = \frac{v^-}{v_{\max}} \tag{2}$$

where v^- indicates the last commanded speed and $v_{\max}$ the maximum valid speed.

For the steering angle network, the heading angle θ was normalized to a value between -1 and $+1$ as follows:

$$\hat{\theta} = \frac{\theta}{\theta_{\max}} \tag{3}$$

where $\theta_{\max}$ denotes the maximum valid heading angle reflecting the field-of-view of the cameras. The range d was also normalized by

$$\hat{d} = \frac{d}{d_{\max}}. \tag{4}$$

For most of the runs, the follow distance d_f was set to 68 ft, the maximum valid range $d_{\max}$ to 120 ft, the minimum valid range $d_{\min}$ to 30 ft, the maximum speed $v_{\max}$ to 20 mph, and the maximum valid heading angle $\theta_{\max}$ to 19.5° corresponding to one half the cameras' field-of-view of 39°.

IV. NETWORK ARCHITECTURE AND TRAINING

It was necessary to train the neural networks based on the current and past samples in order to be able to detect the lead vehicle's motion trends in time and the human driver's response to them. Hence, a time-delay neural-network (TDNN) architecture was selected to perform the DCG function. TDNN is a backpropagation neural network that uses a time-delayed sequence as its input vector [14]. This allows it to deal with input data that are presented over time such as range and heading angle samples. Here, two separate TDNN's were employed—one TDNN was trained to perform speed control and

TABLE II
A LIST OF DATA COLLECTION TRIALS FOR SPEED CONTROL

Trial Number	Lead Vehicle Actions	Number of Samples
1	0 mph, increase to 10 mph, stop, increase to 10 mph, stop	303
2	0 mph, increase to 15 mph, stop, increase to 15 mph, stop	235
3	0 mph, increase to 18 mph, stop, increase to 18 mph, stop	258
4	0 mph, increase to 10 mph, stop, increase to 10 mph, stop	325
5	0 mph, increase to 15 mph, stop, increase to 15 mph, stop	286
6	0 mph, increase to 18 mph, stop, increase to 18 mph, stop	295

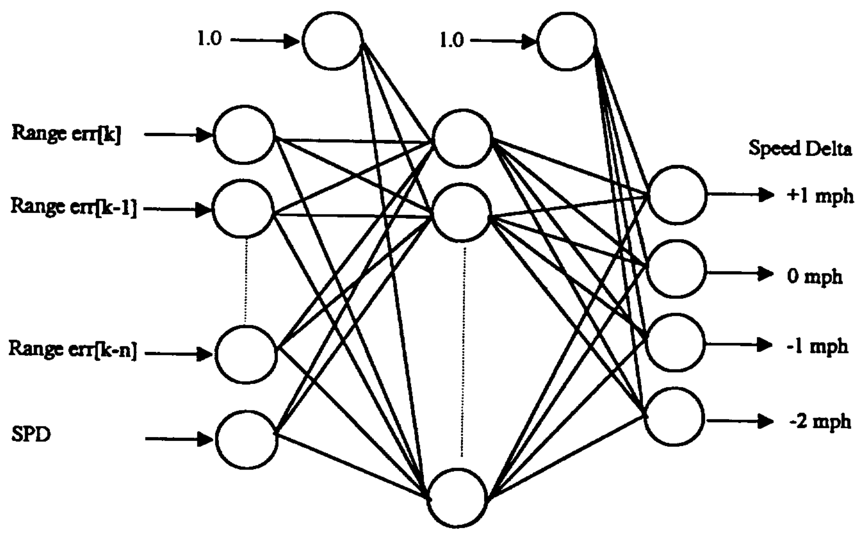


Fig. 5. Architecture of the speed control network.

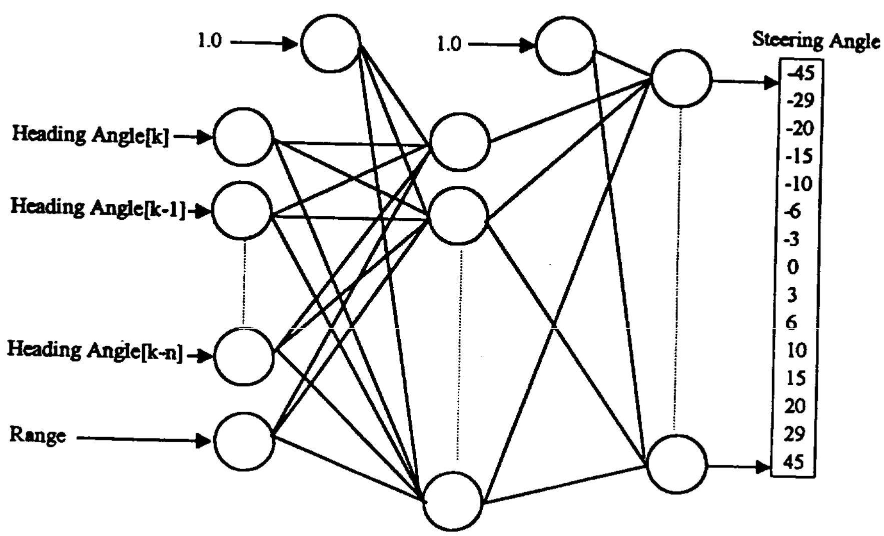
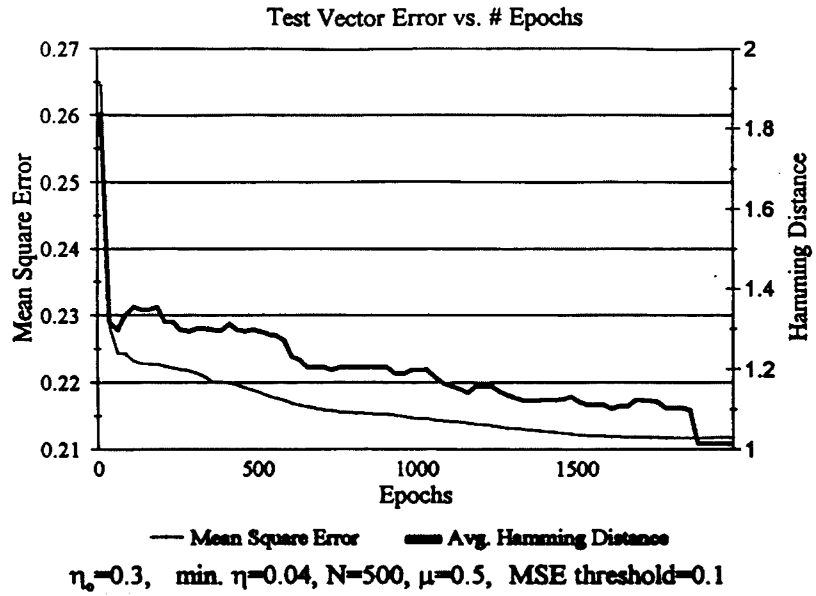


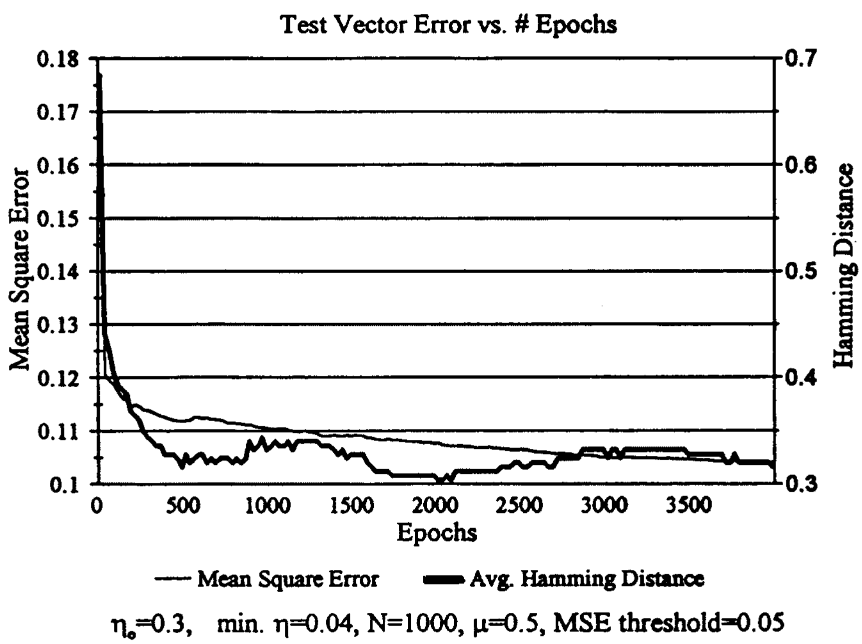
Fig. 6. Architecture of the steering control network.

the other steering angle control. By using two separate neural networks, it was possible to achieve an optimum network design for each control function. In other words, considering that the number of past inputs had to be determined based on responses of typical human drivers, this number could be obtained through experimentation if the control functions were learned separately.

The input vector to the speed control network consisted of seven components or nodes: the normalized current speed and the current and five previous range errors. The number of input nodes was a function of the time dependency of the drivers' responses to range which was found by experimentation [15]. The output vector to the speed control network was a binary vector having four components or nodes corresponding to the



(a)

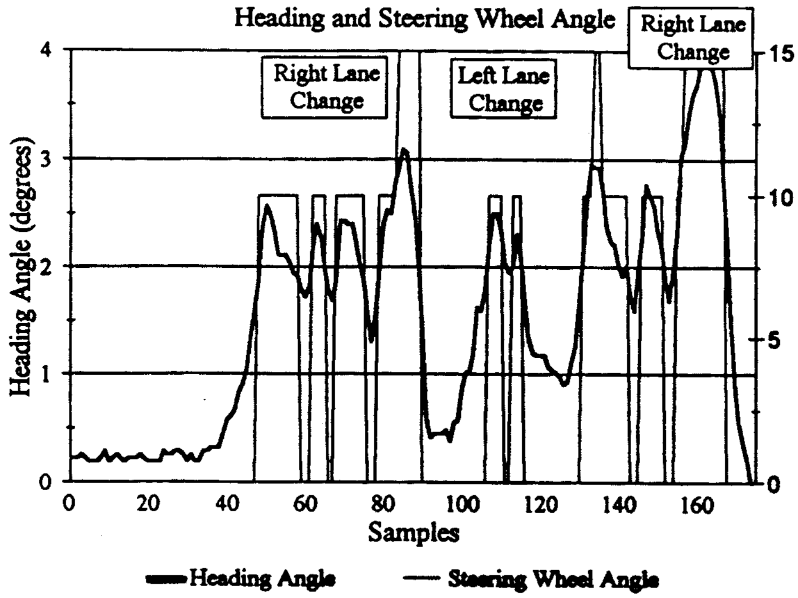


(b)

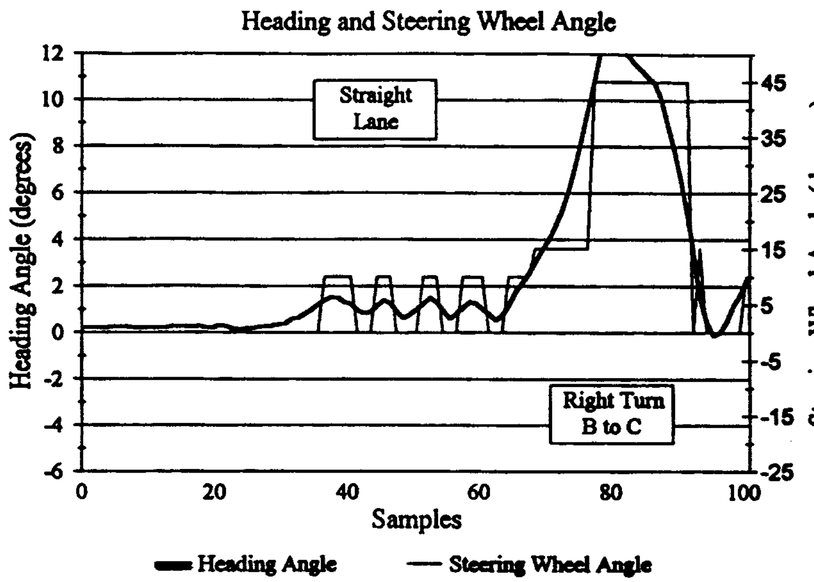
Fig. 7. Typical learning curves of the (a) steering network and (b) speed network.

speed changes -2, -1, 0, and 1 mph, i.e., joystick quantization levels. The number of hidden layer nodes was set to 21 after carrying out an analysis of the network behavior. This analysis included the recall performance of the network having a different number of hidden layer nodes and using test data different than the training data collected during a different run. The network with 21 hidden layer nodes gave the least amount of error. In addition to mean-square error, Hamming distance was used to provide a more accurate performance measure. This is because only the value of the output node with the largest activation is transmitted to the microprocessor, and this output is represented by the binary bit 1 and all other nodes by the binary bit 0. Therefore, as shown in Fig. 5, the topology of the speed network was considered to be a three-layer structure having 7–21–4 nodes.

The input vector to the steering control network consisted of four components or nodes: the normalized current range and the normalized current and two previous heading angles. Again, the number of input nodes was a function of the time dependency of several drivers' response to a heading angle which was found by experimentation [15]. The output vector to the steering control network was a binary vector having 15 components or nodes corresponding to the steering wheel angles 45°, -29°, -20°, -15°, 10°, -6°, -3°, 0°, 3°, 6°, 10°, 15°, 20°, 29°, and 45°, i.e., joystick quantization levels. The number of hidden layer nodes was set to 12 after carrying



(a)



(b)

Fig. 8. Typical steering responses during the autonomous neural-network driving.

out an analysis of the network behavior. Therefore, as shown in Fig. 6, the topology of the steering network was considered to be a three-layer structure having 4–12–15 nodes.

Backpropagation training was used to set up the connection weights of the TDNN's. Refer to [14] for details of backpropagation training. Fig. 7(a) and (b) illustrates typical learning errors versus the number of training epochs for speed and steering networks, respectively. An epoch denotes one cycle of training data. The network training parameters include: (1) learning rate η given by $\eta = \eta_0(0.5)^{\frac{\text{epoch}}{\text{N}}}$, where η_0 denotes the initial learning rate and N a decay rate; (2) minimum learning rate $\eta_{\min}$, i.e., if $\eta < \eta_{\min}$, then $\eta = \eta_{\min}$; (3) momentum factor μ; and (4) stopping criteria consisting of a maximum number of iterations and mean-square-error threshold.

It should be mentioned that the training time was a function of the number of samples in the training data set, stopping criteria, and hardware used. The training procedure in our case was done on the onboard PC 486 machine for the following two cases: 1) built-in weights—this case corresponded to extensive training based on 3900 iteration epochs and 25 min of data for steering and 14.5 min of data for speed. The training time for this case took a few hours and 2) on-the-fly weights—this case corresponded to a few minutes (about 10 min) of training for shorter paths. Although in both cases, the autonomous vehicle managed to follow the lead vehicle successfully, the first case provided a smoother ride. This is because, in the first case, the network was given enough time to learn the optimum weight values. Furthermore, in the first case, the network's response was based on a large number of

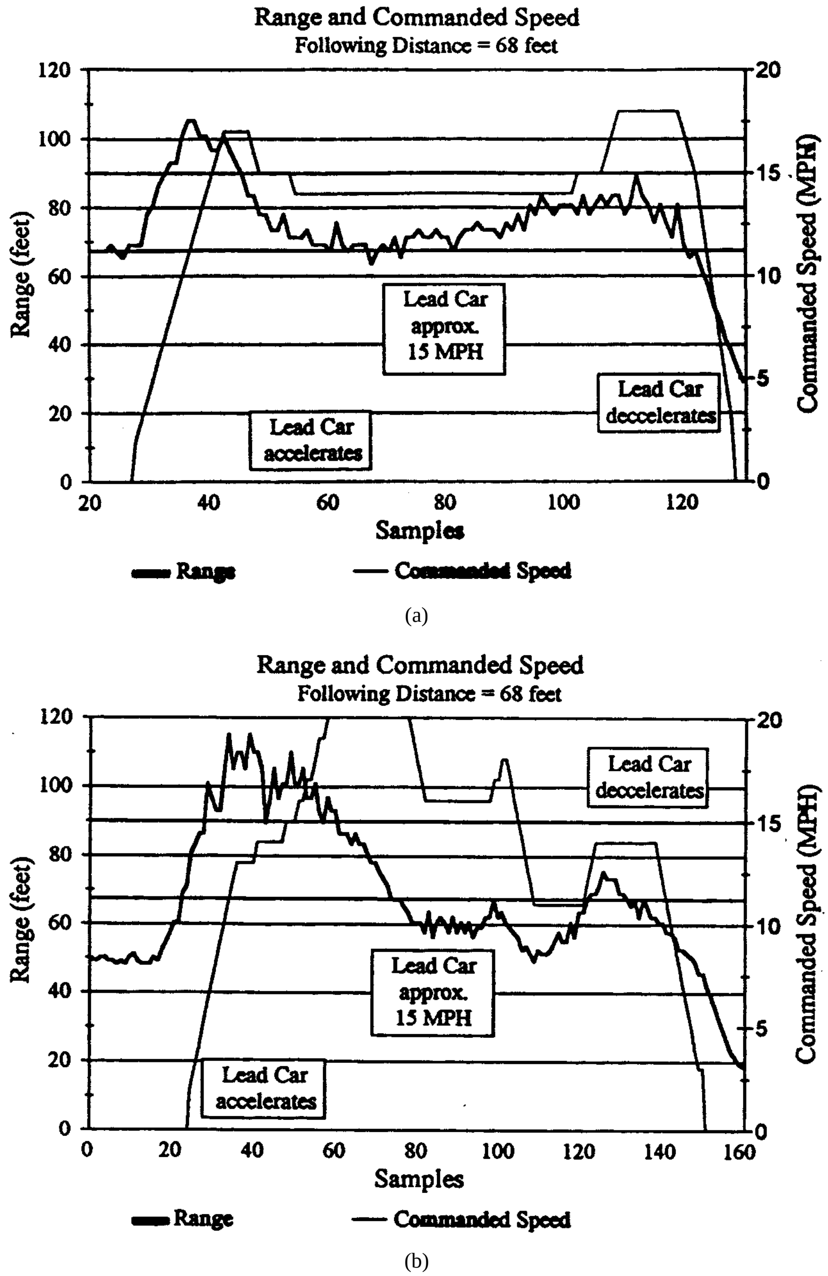


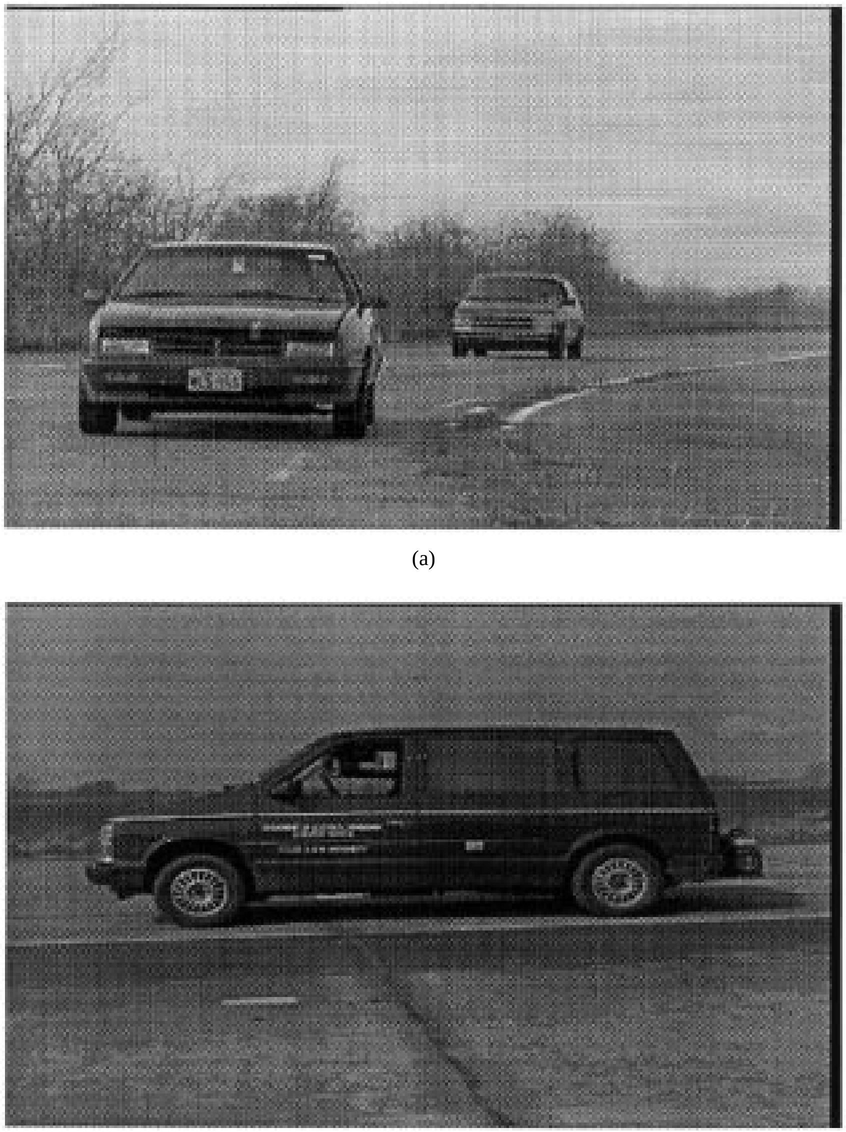
Fig. 9. Typical speed responses during the autonomous neural-network driving.

samples or variations, whereas in the second case, it was based on only a limited number of samples or variations.

It should be noted that the nonlinear mapping learned by the neural network is applicable only to the situations and the environment in which training data is gathered. Situations that are completely different than those observed during data collection would not be exactly identifiable by the neural network, and the response of the network would be an error optimizing generalization based on the previously learned situations.

V. Full-Scale Live Runs and Conclusions

Full-scale live runs were conducted to test the developed neural-network controller. Figs. 8 and 9 illustrate the steering wheel angle and speed commands generated by the neural-network controller in a typical test run. Fig. 8(a) corresponds to a lane-changing maneuver between points A and B and Fig. 8(b) to a right-turn maneuver from B to C of the runway course shown in Fig. 4. As can be observed, under the neural-network control, BART waited until the lead vehicle had started lane changing and then changed its lane and straightened out much like the human driver did during training. Fig. 9(a) and (b) corresponds to two speed change maneuvers consisting of acceleration, constant speed, and deceleration between points A and B of the runway course. As can be seen, after a speed command was transmitted, the servo controller applied the throttle or brake to achieve the commanded speed after a delay. The typical responses shown in these figures represent the human driver's response to the vehicle following function. Several live test run experiments are provided on a



(a)

(b)

Fig. 10. Snapshot images of the vehicle following test runs.

videotape (a copy of the videotape can be obtained by writing to the authors). Two snapshot images of these runs are shown in Fig. 10.

In conclusion, this experimental work has shown the development and testing of a neural-network approach to control the speed and steering of a vehicle for the purpose of performing autonomous vehicle following. The strength of this approach is that it does not require the characterization of nonlinear vehicle dynamics. As a result, the control mechanism can be transported to any vehicle regardless of its dynamics. The live test results obtained and presented on a videotape indicate that it is feasible to employ a neural network to achieve autonomous vehicle following.

ACKNOWLEDGMENT

The authors wish to thank the Texas Transportation Institute for the use of their runway facility and for their help in producing the videotape. They would also like to thank H. Lee for writing a menu-driven software interface for this project.

REFERENCES

[1] C. Thorpe *et al.*, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 362–373, 1988.

[2] M. Turk *et al.*, "VITS—A vision system for autonomous land vehicle navigation," *IEEE Trans. PAMI*, vol. 10, pp. 342–361, 1988.

[3] S. Kenue, "Lanelok: Detection of lane boundaries and vehicle tracking using image processing techniques," in *SPIE Proc. Mobile Robots*, vol. 1195, 1989, pp. 221–245.

[4] S. Shladover *et al.*, "Automated vehicle control developments in the PATH program," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 114–130, 1991.

[5] M. Taniguchi *et al.*, "The development of autonomously controlled vehicle (PVS)," in *Proc. Vehicle Navigation and Information Syst. Conf.*, 1991, pp. 1137–1141.

[6] E. Dickmanns and V. Graefe, "Applications of dynamic monocular machine vision," *Machine Vision Applicat.*, pp. 241–261, 1988.

[7] N. Kehtarnavaz, N. Griswold, and J. Lee, "Visual control of an autonomous vehicle (BART)—The vehicle-following problem," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 654–662, Aug. 1991.

[8] N. Griswold and N. Kehtarnavaz, “Experiments in real-time visual control,” in *SPIE Proc. Mobile Robots*, vol. 1388, Aug. 1991, pp. 342–349.
[9] N. Kehtarnavaz, J. Lee, and N. Griswold, “Vision-based convoy following by recursive filtering,” in *Proc. American Control Conf.*, 1990, pp. 268–273.
[10] N. Kehtarnavaz, N. Griswold, and J. Kim, “Real-time visual control for an intelligent vehicle—The convoy problem,” in *SPIE Proc. Real-Time Image Processing*, vol. 1295, 1990, pp. 236–246.
[11] A. Pomerleau, “ALVINN: An autonomous land vehicle in a neural network,” in *Advances in Neural Information Processing*, vol. 1. San Francisco, CA: Morgan-Kaufman, 1989.
[12] A. Pomerleau, *Neural Network Perception for Mobile Robot Guidance.* Holland: Klumer, 1993.
[13] I. Rivals, D. Canas, L. Personnaz, and G. Dreyfus, “Modeling and control of mobile robots and intelligent vehicles by neural networks,” in *Proc. IEEE Intelligent Vehicle Symp.*, Paris, France, Oct. 1994, pp. 137–142.
[14] J. Zurada, *Artificial Neural Systems.* St. Paul, MN: West, 1992.
[15] K. Miller, “A portable neural network approach to vehicle tracking,” M.S. thesis, Texas A&M Univ., College Station, TX, May 1994.

Nasser Kehtarnavaz (SM’92) received the B.S. degree in electronic and communication engineering from the University of Birmingham, Birmingham, U.K., in 1982 and the M.S. and Ph.D. degrees in electrical and computer engineering from Rice University, Houston, TX, in 1984 and 1987, respectively.

He is an Associate Professor of Electrical Engineering and Computer Science at Texas A&M University, College Station, and an Adjunct Associate Professor of Radiology at Baylor College of Medicine, Houston, TX. He has had industrial experience in various capacities at the Houston Health Science Center, AT&T Bell Labs, Physical Research, and the U.S. Army TACOM. His research areas include image analysis, neural/fuzzy systems, and autonomous vehicle applications. He is currently serving as an Associate Editor of the *Journal of Electronic Imaging.*

Norman Griswold (SM’90) received the B.S. degree from Clarkson College, Potsdam, NY, in 1960, the M.S. degree from the University of Cincinnati, OH, in 1971, and the D.Eng. degree in electrical engineering from the University of Kansas, Lawrence, in 1976.

He is currently a Professor and Interim Department Head at the Department of Electrical Engineering, Texas A&M University, College Station. Prior to joining Texas A&M University, he was a Staff Engineer at the AF Avionics Laboratory, Wright Aeronautical Labs, Dayton, OH. His research areas include modeling of the binocular fusion system for stereo viewing, transmission of stereo information, and investigative work in autonomous land vehicles.

Kelly Miller received the B.S. and M.S. degrees in electrical engineering from Texas A&M University, College Station, in 1987 and 1994, respectively.

His industrial experience includes being an Avionics Engineer for General Dynamics, Ft. Worth, TX, ASIC Development Engineer for Compaq Computer Corporation, Houston, TX, and Design Engineer for Cirrus Logic Corporation, Broomfield, CO.

Paul Lescoe received the B.S. degree in electrical engineering from Wayne State University, MI, in 1975 and the M.S. degree in electrical engineering from University of Detroit, MI, in 1982.

He is the Chief of the Robotics and Driver’s Automation Office at the U.S. Army Tank Automotive and Armaments Command, Warren, MI.

Mr. Lescoe is a Member of ITS America, the Association for Unmanned Vehicle Systems International Trustee, and Director of Intelligent Transportation Systems of Michigan.