

CDNF-9604114--1
ANL/RA/CP--88386

Paper to be Presented at the Neural Network Applications in Highway and Vehicle Engineering Conference, April 8-10, 1996, Washington, D.C. Metropolitan Area.

An Artificial Neural Network Controller for
Intelligent Transportation Systems Applications *

RECEIVED

MAR 27 1996

OSTI

by

Javier E. Vitela **, Ulf R. Hanebutte, and Jaques Reifman

Reactor Analysis Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439 USA

** On Leave of Absence from
Instituto de Ciencias Nucleares
Universidad Nacional Autonoma de Mexico,
4510 Mexico, D.F.

The submitted manuscript has been authored by a contractor of the U. S. Government under contract No. W-31-109-ENG-38. Accordingly, the U. S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

*Work supported by the U.S. Department of Energy, Nuclear Energy Programs under Contract W-31-109-ENG-38

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED *at*

MASTER

An Artificial Neural Network Controller for Intelligent Transportation Systems Applications*

Javier E. Vitela[†], Ulf R. Hanebutte, and Jaques Reifman

Reactor Analysis Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439

Abstract

An Autonomous Intelligent Cruise Control (AICC) has been designed using a feedforward artificial neural network, as an example for utilizing artificial neural networks for nonlinear control problems arising in intelligent transportation systems applications. The AICC is based on a simple nonlinear model of the vehicle dynamics. A Neural Network Controller (NNC) code developed at Argonne National Laboratory to control discrete dynamical systems was used for this purpose. In order to test the NNC, an AICC-simulator containing graphical displays was developed for a system of two vehicles driving in a single lane. Two simulation cases are shown, one involving a lead vehicle with constant velocity and the other a lead vehicle with varying acceleration. More realistic vehicle dynamic models will be considered in future work.

1 Introduction

Emerging Intelligent Transportation Systems (ITS) and Automated Highway Systems (AHS) technologies have given rise to a myriad of nonlinear control problems that need to be addressed. Within this scope, much effort has been directed in recent years to the development of automatic longitudinal vehicle controls that focus on improving the safety and ef-

*Work supported by the U.S. Department of Energy, Nuclear Energy Programs, under Contract W-31-109-ENG-38.

[†]On Leave of Absence from Instituto de Ciencias Nucleares, Universidad Nacional Autónoma de México, 04510 México, D.F.

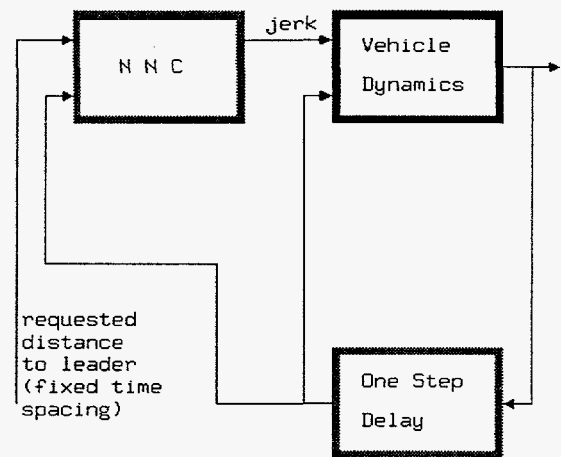


Figure 1: Neural Network Controller /Dynamical System Configuration.

iciency of the transportation systems. Most of this research has been carried out within the framework of nonlinear control methods and linear-quadratic optimal control theory.[1-4] In this work we use Artificial Neural Networks (ANNs) for this purpose because of the considerable attention they have received in recent years within the engineering community for control of nonlinear dynamical systems and for their ability to act as intelligent controllers. An “intelligent control” system is one which generates and executes timely control actions in order to satisfactorily guide a given system from an initial state to a terminal state without violating a prescribed set of constraints and objectives.[5] The strength of ANNs lies in their ability to approximate arbitrary nonlinear mappings,

thus providing nonlinear control designs that in general cannot be obtained by means of more traditional control techniques.

The purpose of this paper is to show how an AICC can be developed using Feedforward ANNs. An AICC is an assisting system capable of controlling the relative speed and distance between two adjacent vehicles in the same lane.[1] In order to differentiate between the two vehicles, the first one will be referred to as the leader and the second as the follower. The leader moves independently of the follower, the AICC is attached to the follower which can control its velocity and position with respect to the leader, i.e. there is no intervehicle communication. The resulting control system should be able to provide the appropriate rate of change of acceleration, i.e. the jerk, of the follower vehicle order to maintain a requested longitudinal distance between the two vehicles. The desired joint NNC-vehicle dynamics configuration is shown in Fig. 1. In general, this longitudinal distance is not a constant parameter but can be set either arbitrarily or according to a certain criterion, e.g. to keep a constant time headway.

2 Simple Nonlinear Vehicle Dynamics Model

For simplicity, the dynamical equations of the follower will be written in a reference coordinate system which moves with the leader. In addition, the following set of assumptions is used to obtain a simplified discrete time model for the follower in this reference system:

- The relative position, velocity and acceleration of the follower with respect to the leader are accessible and can be directly measured.
- The acceleration of the follower is assumed constant during fixed intervals of time, or time steps. After each time step, the value of the acceleration is updated based upon the value of the jerk at the immediately preceding time step.
- The relative acceleration of the follower is assumed independent of the velocity of the leader and it is constrained to remain between $+2m/sec^2$ and $-5m/sec^2$.
- The control variable, i.e. the jerk, is independent of the speed of the leader and can

take arbitrary values within the interval between $-5m/sec^3$ and $+5m/sec^3$.

With the above assumptions the dynamic equations of the follower are given by,

$$\hat{x}_{k+1} = \hat{x}_k + \hat{v}_k \Delta t + \frac{1}{2} \hat{a}_k \Delta t^2, \quad (1)$$

$$\hat{v}_{k+1} = \hat{v}_k + \hat{a}_k \Delta t, \quad (2)$$

$$\hat{a}_{k+1} = \begin{cases} \hat{a}_{max} & : \hat{a}_k + u_k \Delta t > \hat{a}_{max} \\ \hat{a}_{min} & : \hat{a}_k + u_k \Delta t < \hat{a}_{min} \\ \hat{a}_k + u_k \Delta t & : \text{otherwise} \end{cases}, \quad (3)$$

where \hat{x}_k , \hat{v}_k and \hat{a}_k are the relative position, velocity and acceleration of the follower relative to the leader. Together these constitute the state of the system denoted by \mathbf{z}_k , and u_k is the control variable. Here, the subindex k represents the time step, with $k = 1, \dots, k_f$. The control variable u is constrained to take values from the following set

$$u_{min} \leq u_k \leq u_{max}. \quad (4)$$

In an actual vehicle, both constraints on the jerk and on the acceleration, Eqs. (3) and (4), will also depend on the actual velocity of the vehicles. Our purpose in choosing this simplified model was to study the feasibility of using ANN for AICC; more realistic models including these additional constraints will be considered in future work.

The purpose of a properly designed controller is to provide the appropriate control action u_k , given the current state of the system, in order to obtain a new state of the system \mathbf{z}_{k+1} such that the successive application of the control law drives the dynamical system toward a desired final state without violating the constraints imposed on the system and on the control variables.

The transformation equations from the relative position (\hat{x}), velocity (\hat{v}) and acceleration (\hat{a}) of the follower to the absolute position (x), velocity (v), and acceleration (a) are given by

$$x_{F,k+1} = x_{L,k} + v_{L,k} \Delta t + \frac{1}{2} a_{L,k} \Delta t^2 + \hat{x}_{F,k+1}, \quad (5)$$

$$v_{F,k+1} = v_{L,k} + a_{L,k} \Delta t + \hat{v}_{F,k+1}, \quad (6)$$

$$a_{F,k+1} = a_{L,k} + \hat{a}_{F,k+1}, \quad (7)$$

where the subscripts F and L denote the follower and lead vehicle, respectively.

3 Neural Network Controller

In this paper a controller is designed using a feedforward neural network with sigmoid activation units. The objective of the Neural Network Controller (NNC) is to provide the sequence of control actions that drives the follower vehicle starting from an arbitrarily selected initial state \mathbf{z}_1 within a pre-specified region in phase space \mathbf{z} , toward a target state \mathbf{z}_t specified by $\hat{x}_{k_f} = \hat{x}_t$, $\hat{v}_{k_f} = 0$, and $\hat{a}_{k_f} = 0$. In other words, the final state is that in which the follower vehicle is located at a desired distance from the leader and remains there, i.e. with zero relative velocity and acceleration. The joint NNC-dynamical system configuration is illustrated in Fig. 1 where the output of the NNC is the jerk and the dynamical system is described by the set of Eqs. (1)-(3).

Feedforward neural networks are composed of nonlinear computational elements or nodes that are arranged in L layers with $L \geq 3$. The output is obtained by processing the values of the input variables from the first layer, input, to the L -th layer or output, through the $L - 2$ remaining hidden layers. The number of nodes in each layer will be denoted here by $J(l)$ with $l = 1, \dots, L$. Each one of the $J(1)$ nodes in the first layer are mapped through the identity function, i.e. each of these nodes receives as input one of the $J(1)$ elements of the input vector and passes them unaltered to serve as the input values to the nodes of the second layer. The nodes in the hidden and output layers are constituted by nonlinear mappings from a multidimensional input received from the nodes of the immediately preceding layer to a one dimensional output, e.g., through a sigmoid function.[6] Thus, the activation of the j -th node in the l -th layer, $\mathcal{O}_j^{(l)}$, is in general given by

$$\mathcal{O}_j^{(l)} = f_j(\text{input}_j^{(l)}) \quad . \quad (8)$$

Here $\text{input}_j^{(l)}$, the effective input to node j , is the weighted sum of the outputs $\mathcal{O}_i^{(l-1)}$ of the immediately preceding layer, i.e.,

$$\text{input}_j^{(l)} = \sum_{i=1}^{J_{l-1}} \omega_{ji}^{(l)} \mathcal{O}_i^{(l-1)} \quad . \quad (9)$$

where $\omega_{ji}^{(l)}$ denotes the weight connecting the output of node i in layer $l - 1$ to the node j in layer l and f_j is the nonlinear activation function of this node.

Given a set of weights in the NNC, we denote by E_p the mean square error between the desired final

state of the dynamical system \mathbf{z}_{pt} and the actual final state \mathbf{z}_{pk_f} , obtained during the the p -th trajectory generated by the joint NNC-dynamical system configuration. The total error, E , is given by the the sum of these individual errors after the presentation of a set of P trajectories, i.e.,

$$E = \sum_{p=1}^P E_p = \frac{1}{2} \sum_{p=1}^P |\mathbf{z}_{pk_f} - \mathbf{z}_{pt}|^2 \quad . \quad (10)$$

Thus, for a fixed topology of the Feedforward NN representing the controller, the design of the NNC entails determining the set of weights \mathbf{w} that minimize the total error E defined above. For this purpose a formal methodology known as Back-propagation Through Time (BPTT),[7,8] was used to calculate the gradient of the error E in weight-space, ∇E , and to reduce this error by updating the weights iteratively using steepest descent, or some other optimization technique such as conjugate gradients. The systematic procedure of reducing the training error E is known as training of the neural network, regardless of the mathematical technique chosen to adjust the weights.

An NNC code has been developed at Argonne National Laboratory to train a Feedforward ANN with the purpose of providing control laws of the form $\mathbf{u} = \mathbf{u}(\mathbf{z}, \mathbf{w})$ for time discrete dynamical systems of the form $\mathbf{z}_{k+1} = \mathbf{F}(\mathbf{z}_k, \mathbf{u}_k)$, where \mathbf{z}_k and \mathbf{u}_k represent the state and the control variables respectively, at time step k ($k = 1, 2, \dots$). Our code is capable of handling in a straightforward fashion the cases where the plant dynamics, \mathbf{F} , are given either by a set of analytical equations obtained by first principles, or techniques such as ANN modeling. In order to speed up the training process of the NNC, this code makes use of the method of conjugate gradients.[9] This technique is a quadratically convergent gradient method that locates an unconstrained local minimum of a multidimensional function. The method is guaranteed to locate the minimum of a quadratic function in a finite number of steps if round-off errors could be avoided. For non-quadratic functions, which is the case of the training error $E(\mathbf{w})$ in Eq. (10), this procedure is iterative and the algorithm will lead to the bottom of whatever valley it starts in.

The following steps summarize the iterative algorithm used by the code to train a neural network as a dynamic system controller :

- Step 1: An admissible region of the extended phase-space (which in our case includes the de-

sired relative position of the follower with respect to the leader) is divided in P number of cells.

- Step 2: Randomly select the initial values of the set of weights that specify the NNC.
- Step 3: At each iteration a set of trajectories are generated by selecting the initial states randomly from each one of the P cells and allowing the NNC-dynamical system to evolve in time until a pre-specified maximum number of time steps is reached or until some other stopping criterion is satisfied.
- Step 4: Convergence is achieved when for the entire set of P trajectories the convergence criterion $|z_i - z_{it}| \leq \epsilon$, for all i , is satisfied; where ϵ is a pre-specified error range and the subindex i corresponds to the components of the state-space vector. Otherwise proceed to the next step.
- Step 5: Using BPTT, calculate the gradient of the error, ∇E , produced by all the different trajectories and update the weights using the method of conjugate gradients.
- Step 6: Repeat steps three through five until the training of the NNC is successfully completed, i.e. when the entire set of P trajectories, each of which starts from a different cell, reaches the target z_t specified above within the error range ϵ .

It has to be pointed out that the above algorithm cannot be considered to be a strict conjugate gradient method because at each iteration of the method an entire new set of initial states is chosen. Nevertheless, this technique allows us to sample uniformly the region of interest in the phase space. Hence, the problem of memorizing or over-training, which commonly appears in pattern recognition using ANN, does not occur here.

4 Training Results

The NNC considered in this work is a three-layered Feedforward ANN with 4-12-1 units per layer respectively, with the units being mapped through sigmoid functions. The four inputs in the first layer correspond to the current relative position, velocity, and acceleration of the follower together with the desired relative position with respect to the lead vehicle. The

activation level of the output node gives the appropriate control action, i.e. the jerk, to be applied during the next time step. The number of hidden nodes was empirically determined.

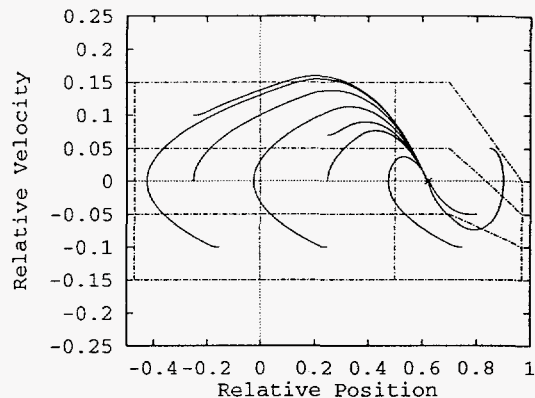


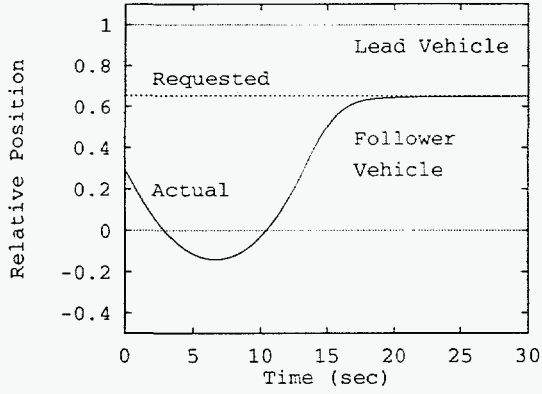
Figure 2: Typical trajectories in the normalized $\hat{x} - \hat{v}$ plane.

For consistency purposes we define the extended phase space to be composed of four elements, the relative position, velocity and acceleration, and the desired position of the follower with respect to the lead vehicle. During the training process, the initial states in the extended phase space were all chosen with zero acceleration. The rest of the values were chosen randomly by dividing the reduced space of initial states in 27 cells. Thus, the number of trajectories generated at each iteration of the algorithm was $P = 27$.

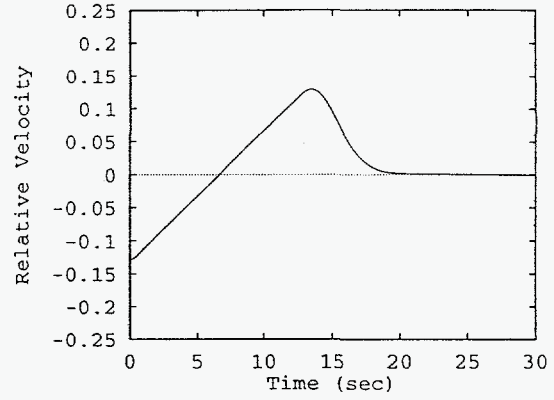
The NNC was successfully trained after the presentation of 51,867 trajectories using an error convergence parameter $\epsilon = 0.01$. Figure 2 shows typical trajectories in the relative position - relative velocity plane, for initial states with zero relative acceleration. All trajectories terminate at the arbitrarily selected distance of 37.5 meters behind the lead car. For simplicity, in this figure, and throughout the rest of the paper, the coordinates in the relative reference system are normalized such that 1 unit of length equals 100 meters. In addition, the position of the lead vehicle is always assumed to be at $\hat{x} = 1$, $\hat{v} = 0$, and $\hat{a} = 0$ in the normalized phase-space.

5 Simulation Results

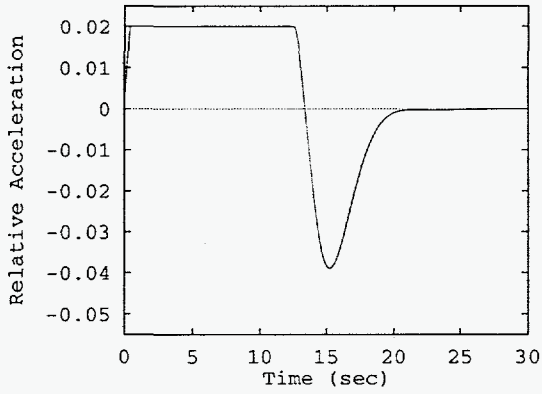
In order to illustrate the behavior of the NNC obtained with the above procedure, we present two cases: one involving a lead vehicle with constant ve-



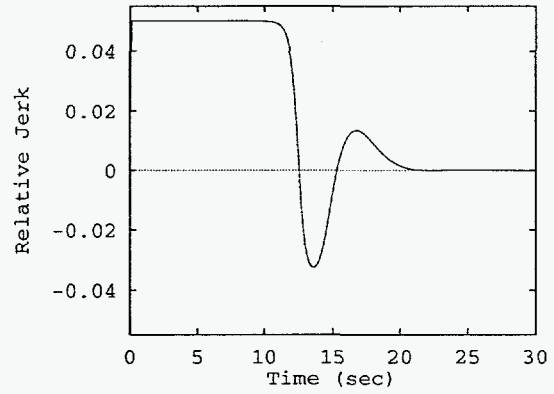
(a) Position.



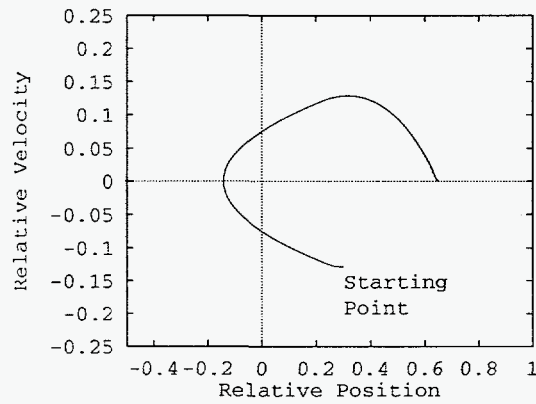
(b) Velocity.



(c) Acceleration.



(d) Jerk.



(e) Phase-Space Diagram.

Figure 3: Time behavior of the normalized relative position, velocity, acceleration and NNC control action (jerk), together with the corresponding trajectory in the $\hat{x} - \hat{y}$ plane for the case of a lead vehicle with constant velocity.

locity, and the other involving a lead vehicle with varying acceleration. In both cases the same ANN is used to provide the control law that drives the follower vehicles to the desired location with respect to the leader.

5.1 Case 1: Lead Vehicle With Constant Velocity

In this example, the following arbitrary initial conditions of the two vehicles were specified: the velocity of the leader is set to $v_L = 27.8m/s$ ($100km/hr$) and remains constant during the entire simulation. The follower has an initial absolute velocity of $15.0m/s$ ($54km/hr$) and is located 70 meters behind the lead vehicle. Both vehicles have zero initial acceleration. It is requested that the NNC drives the follower vehicle from its initial conditions to a constant time headway of 1.25 seconds behind the leader (or 37.5 meters), and maintains this relative position afterwards.

Figure 3 shows the resulting behavior of the follower vehicle in the previously defined normalized relative reference system. Figure 3(d) shows the control action (jerk) as a function of time, as provided by the NNC. Figures 3(a-c) show the resulting behavior of the relative position, velocity and acceleration of the follower with respect to the leader. As observed in Fig. 3(a), the initial distance between the two vehicles increases because in the coordinate system, which is moving with the leader, the follower vehicle has initially a negative velocity. This velocity first increases to appropriate values, and then vanishes asymptotically in order to reach the requested location (i.e. in this case 37.5 meters behind the lead vehicle). It is observed from these figures that the desired position, indicated by a fine-dotted line in Fig. 3(a), is reached after 20 seconds into the simulation.

5.2 Case 2: Lead Vehicle With Varying Acceleration

In this example, the initial conditions of the two vehicles are the following: the absolute velocities of the follower and lead vehicle are $25.0m/s$ ($90km/hr$) and $27.8m/s$ ($100km/hr$), respectively; the accelerations of the follower and leader are initially zero and $+2.0m/s^2$, respectively, and the initial distance between them is assumed to be 70 meters. During the simulation the acceleration of the leader alternates between $+2.0m/s^2$ and $-2.0m/s^2$ every 10 seconds.

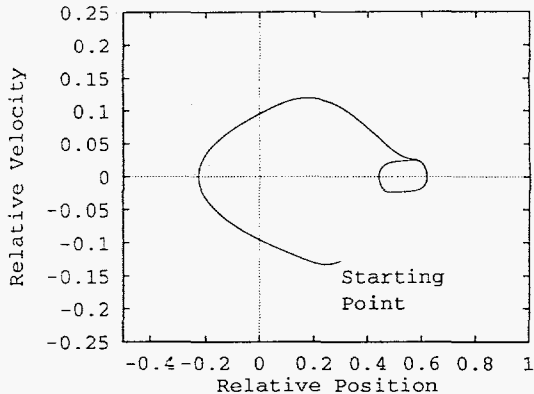


Figure 4: Trajectory in $\hat{x} - \hat{y}$ space of the follower in the case of a lead vehicle with varying acceleration.

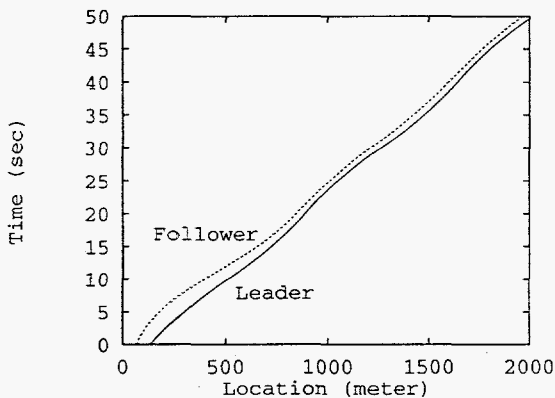


Figure 5: Location vs. time diagram for the case of a lead vehicle with varying acceleration.

Similarly to the previous example, it is requested that the NNC controls the follower in order to maintain a constant time headway of 1.25 seconds behind the leader.

In Fig. 4 we show the behavior of the follower in the normalized relative $\hat{x} - \hat{y}$ plane where as in the previous example, the leader is located at $(1, 0)$. Due to the initial relative velocity of the follower and the positive acceleration of the leader, the distance between the two vehicles increases initially until the follower has acquired a positive relative velocity and then reaches the proximity of the requested distance. Since in this example the velocity of the leader has a periodic linear oscillatory behavior, the requested distance between both vehicles (due to the imposed constant time headway requirement) shows the same

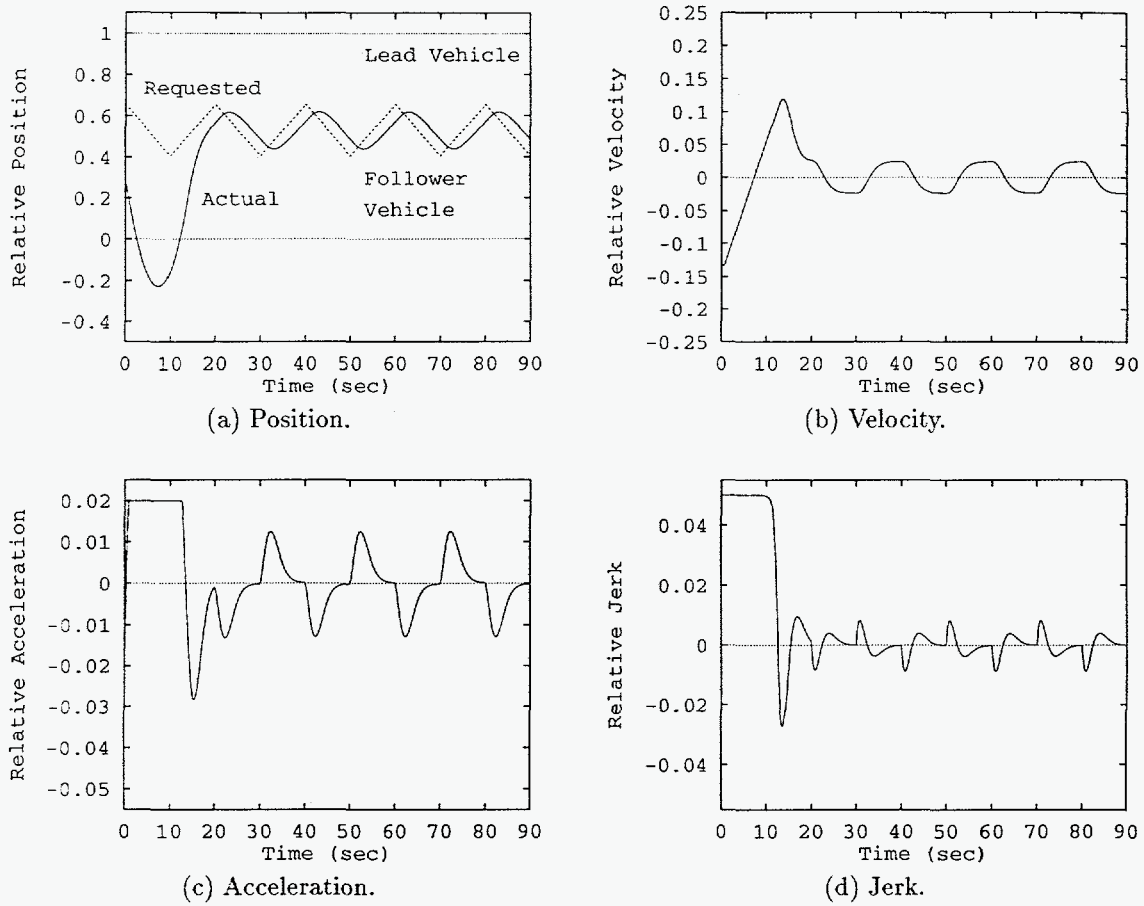


Figure 6: Time behavior of the normalized relative position, velocity, acceleration and NNC control action (jerk) in the case of a lead vehicle with varying acceleration.

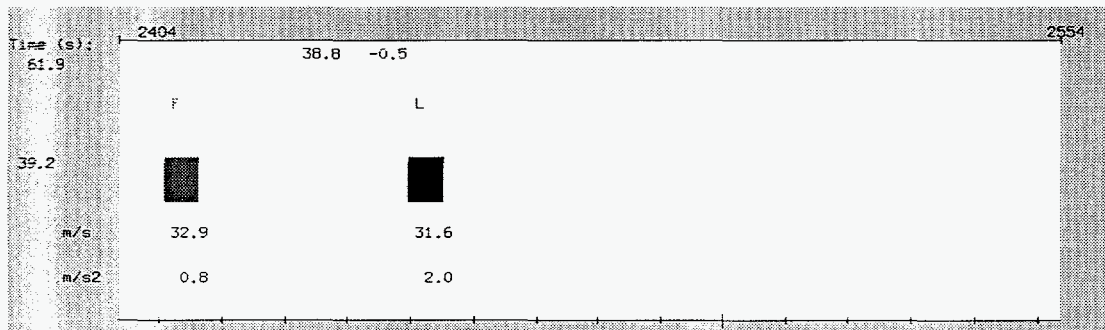


Figure 7: Typical screen display of sliding road segment (fixed at the location of follower vehicle).

type of behavior which is manifested by the closed cycle trajectory in the $\hat{x} - \hat{y}$ plane (Fig. 4).

The location versus time diagram for the leader and follower is given in Fig. 5, which shows the "wavy" path of the leader due to its cyclically changing velocity, and the path of the follower as a result of the AICC. The normalized relative position, velocity and acceleration of the follower with respect to the leader are shown in Fig. 6(a-c), and the control action (jerk) provided by the NNC as a function of time is given in Fig. 6(d). As discussed previously, because the velocity of the lead vehicle oscillates during the course of the simulation, the requested distance between both vehicles varies in time, as illustrated by the zig-zag dotted line in Fig. 6(a).

6 Graphical Display of AICC Simulator

An AICC-simulator with color graphic displays has been developed to facilitate the testing and evaluating of the AICC system. The simulator provides animated graphics showing the motion of two vehicles on a road segment in three different ways. Two windows in the simulator provide the location of the vehicles on the road. One of these windows shows the relative location of the vehicles from the point of view of the follower (sliding road view) and the other shows the actual position of the vehicles on the road. A third window contains a 3-D view of the road as seen by the follower. The screens displaying the sliding road view and the 3-D road view are shown as gray-scale copy in Fig. 7 and Fig. 8, respectively.

Figure 7 gives the screen at 61.9 seconds into the simulation of case 2. The follower (identified by "F") is at a stationary location at the far left, while the sliding road segment (150m length) shows the road between 2,404m and 2,554m. The absolute velocity and acceleration are shown at the bottom of the corresponding vehicles. The requested distance is displayed on the left side, while the actual distance and the deviation from the requested distance is given above the vehicles.

The information regarding the relative position, velocity, acceleration and jerk of the follower vehicle as a function of time is also displayed graphically as the simulation progresses.

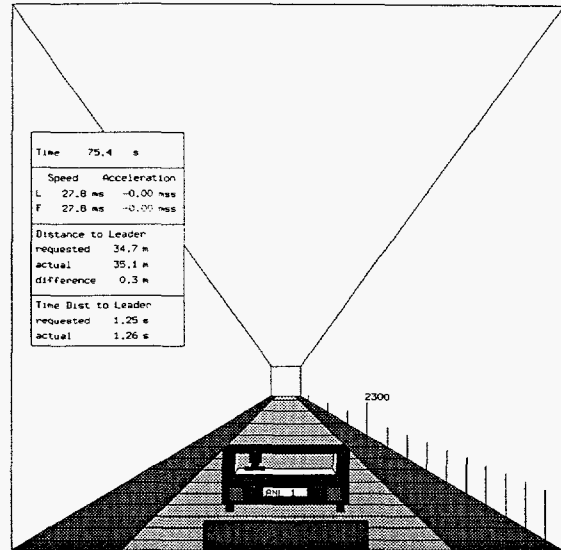


Figure 8: Typical screen display of 3-D road view.

7 Conclusions

Using a simple nonlinear dynamic model, a NNC was successfully designed to perform the tasks of an AICC. For this purpose a Feedforward ANN with sigmoid activation functions was used. The resulting controller was tested in an AICC-simulator with animated graphical displays. While this paper considered the use of the AICC in a system of only two vehicles, work is currently being done to test the behavior of the NNC in multiple vehicle platooning and stop-and-go traffic. These more complex applications should not pose difficulties for the NNC controller. In order to study the behavior of larger systems containing many AICC equipped vehicles the NNC is being incorporated into the Argonne ITS simulator.[10] More realistic vehicle dynamics models will also be considered in future work.

References

- [1] Morello, E., Benz, Th., and Ludmann, J. (1994), *AICC Assessment*. Proc. First World Congress on ATT/IVHS, Nov 30 - Dec 3, 1994, Paris, France.
- [2] Godbole, D.N., and Lygeros, J., *Longitudinal Control of a Lead Car of a Platoon*. IEEE Trans-

actions on Vehicle Technology, Vol.43, No. 4, Nov 1994, pp 1125-1135.

- [3] Fancher, P.S., Bareket, Z., Johnson, G., and Sayer, J., *Evaluation of Human Factors and Safety Performance in the Longitudinal Control of Headway*. Proc. Second World Congress on Intelligent Transport Systems, Yokohama, Japan. Nov 1995.
- [4] Urban, T.J., Pue, A.J., and Chiu, H.Y., *A State-Constrained Approach to AVHS Control of Vehicle Following*. Proc. Second World Congress on Intelligent Transport Systems, Yokohama, Japan. Nov 1995.
- [5] Vemuri, V.R. (Ed.), *Artificial Neural Networks: Concepts and Control Applications*. IEEE Computer Society Press, Los Alamitos, CA, 1992.
- [6] Rumelhart, D.E., Hinton, G.E., and Williams, R.J., *Learning Internal Representations by Error Propagation*. Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol. I, p.319, D.E. Rumelhart and J.C. McClelland, Eds. MIT Press (1986).
- [7] Werbos, P.J., *Backpropagation Through Time: What It Does and How to Do It*. Proc. of the IEEE, Vol.78, No.10, 1550 (1990).
- [8] Piché, S.W., *Steepest Descent Algorithms for Neural Networks Controllers and Filters*. IEEE Trans. on Neural Networks, Vol.5 No.2, 198 (1994).
- [9] Reifman, J. and Vitela, J.E., *Accelerating Learning of Neural Networks with Conjugate Gradients for Nuclear Power Plant Applications*. Nuclear Technology, Vol.106, No. 2, 225 (1994).
- [10] Ewing, T.F., Canfield, T., Hanebutte, U.R., Levine, D., and Tentner, A., *Large-Scale Intelligent Transportation Systems Simulation*. Proc. High Performance Computing '95, April 9 - 13, 1995, Phoenix, AZ.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.