In today's freeway traffic there is very little control that can effectively influence the flow of traffic and manage congestion. Apart from ramp metering, the flow of traffic is mainly influenced by the actions of individual drivers. These actions are very erratic. They differ from one place to another and are responsible for traffic flow instabilities that lead to congestion and inefficient use of highways.

In this article we use an artificial neural network technique to model and control highway traffic in a single lane with no on- or off-ramps. The developed controllers generate the speed commands for each section of the lane that vehicles need to follow in order to achieve a desired traffic flow density distribution along the lane. In today's traffic, these speed commands could be communicated to drivers, who would then have to respond to them. This way raises human factors issues that need further investigation in order to assess possible benefits. In an automated highway environment, the speed commands could be communicated to the computer control system of the vehicle and followed directly without human errors and delays.

In this article, we simulate an automated highway environment and designed roadway neural network controllers to alleviate congestion. We use simulations to demonstrate that the use of feedback control on the macroscopic level could bring dramatic improvements on the characteristics of traffic flow. The same methodology could be applied to traffic in multiple lanes with on- and off-ramps.

Introduction

In most metropolitan areas, highway driving is the dominant mode of urban transportation. With the increasing demand of automobile and highway usage, highway traffic will continue to be an important social and economic problem. The current consensus of most traffic engineers and researchers is that highway capacity will be greatly increased if the facilities can be designed to be more intelligent and efficient by using advanced technologies. It has been argued that the use of smart sensors, actuators, computers, and software tools, etc. is more practical...
and economically advantageous than building new highways. One of the outcomes of these arguments and debates is the concept of automated highway systems (AHS), where automation techniques are applied to vehicles and roadway in order to increase the capacity and efficiency of existing facilities.

Congestion is one of the most serious problems in highway traffic. The instability inherent in traffic flow due to the behavior of human drivers is one of the main causes of congestion. With vehicle automation such as intelligent cruise control (ICC) [7] (an initial step towards AHS) that allows vehicles to follow each other automatically, the roadway may be able to control traffic density and eliminate congestion by commanding the average speed of vehicles along lanes. One way of designing such controllers is to use a mathematical model that describes the flow of traffic.

The average behavior of vehicles at a specific location and time instant in a lane can be expressed by the flow variables defined for traffic. The three most important traffic flow variables used to describe the state of traffic flow are flow rate \( q \), concentration or density \( k \), and speed \( v \). There is a fundamental relationship among these variables called the fundamental equation of traffic flow which is written as \( q = k \cdot v \), and is analogous to that of water flow studied in hydromechanics. An earlier attempt to describe the particular characteristics of traffic flow led to Greenshields' linear mathematical formula for speed-density relationship [5], followed by several nonlinear formulas proposed by Greenberg, Pipes, Munjal, Drew, and Drake et al. [4, 17, 12, 3, 8]. These relationships that describe the steady-state behavior of traffic flow are called traffic stream models. A volume-density relationship \( Q(k) \), which is the well-known fundamental diagram of traffic engineering, can be derived based upon these formulas and the fundamental equation. Based upon this understanding, Lighthill and Whitham [11] first proposed a model that explains kinematic waves phenomena in traffic flow. By assuming that the fundamental diagram is applicable, they formulated the continuity equation, a partial differential equation with density as the only state variable. Since this model is established based upon the analogy with incompressible fluid flow, it has certain limitations in describing traffic flow. Built upon the model of Lighthill and Whitham, several models have been proposed to modify or extend formulas in an attempt to overcome these limitations. Payne proposed a space increment in which the mean speed adjusts to downstream traffic density. Also, the assumption of stationary mean speed-density relationship that mean speed adjusts instantaneously to traffic density was removed by introducing a small time delay [16]. These changes led to a model with two state variables, Papageorgiou et al. [15] proposed an extension to this model to consider weaving phenomena. Recently, a modified model was proposed by Karaaslan, Varaiya, and Walrand [10] in order to eliminate some unrealistic phenomena observed in Payne's model. Generally speaking, mathematical modeling of traffic flow results in a nonlinear dynamic system. Because of the inconsistencies of driver behavior and variations in traffic flow patterns, one model validated in one area or one time may be inaccurate in another area and/or another time.

A potential method for modeling and controlling traffic flow is one that employs learning so that dynamic changes in traffic are learned on-line and accommodated with the proper use of identification and control techniques. One such method is based on the use of Artificial Neural Networks (ANNs).

Due to their capability to classify patterns, ANNs have been proposed to passively control traffic flow on freeways; namely, to monitor the traffic flow and detect non-recurring congestion caused by incidents [20, 1]. Spatial and temporal traffic patterns of lane-blocking incidents are recognized by ANNs, and this information can be used to formulate effective countermeasures, which may involve provision of real-time traveler information, timely dispatch of emergency services and incident removal crews, and real-time control of traffic entering and on the freeway.

ANNs have also been proposed for identification and control of nonlinear dynamic systems by several researchers [13, 18]. The learning capability and universal approximation property of ANNs make them suitable for modeling and control of uncertain nonlinear and time-varying dynamic systems such as vehicular traffic.

In this article, we consider the modeling and control of traffic flow in a single highway lane using ANNs. The ANNs are first trained to accurately model the dynamics of the traffic flow. The ANN model is then used to control the traffic density and force it to follow a desired one. The objective of the controller is to eliminate congestion caused by inhomogeneous traffic density distributions along the lane. Computer simulations are used to demonstrate the effectiveness of the use of ANN techniques to model and control traffic flow.

### Macroscopic Traffic Flow Models

Analogous to real fluid, traffic involves flows, concentrations, and speeds. There is a natural tendency to attempt to describe traffic in terms of fluid behavior. The macroscopic view of traffic flow is based upon this hydrodynamic analogy. This point of view is more concerned with the overall average behavior of traffic than with the interactions between individual vehicles.

Treating traffic flow as a fluid analogue with density \( k(x,t) \) and volume \( q(x,t) \) at location \( x \) and time \( t \), we obtain, using the law of conservation of mass, the continuity equation

\[
\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = r - s
\]

(1)

where \( r \) and \( s \) represent exogenous on-ramp and off-ramp sources respectively.

We discretize the above equation as follows: We consider a highway lane which is subdivided into \( N \) sections with length \( L_i \) \((i = 1, ..., N)\) each having at most one on-ramp and one off-ramp as shown schematically in Fig. 1.

For a discrete time instant \( nT \), where \( T \) is the sample time interval, we introduce space-time-discretized traffic flow variables as follows.

- \( k(i) \): density in section \( i \) at time \( nT \).
- \( v(i) \): space mean speed in section \( i \) at time \( nT \).
- \( q(i) \): traffic flow leaving section \( i \) entering section \( i + 1 \) at time \( nT \).
- \( L_i \): length of freeway in section \( i \).

Based upon these variables, a space-time-discretized form of the continuity Equation (1) is given by
The model of Karaaslan, Varaiya, and Walrand [10], which was proposed as a modification of the model given by (5), is:

\[
v_1(n+1) = v_1(n) + \frac{T}{\Delta T} \left[ \frac{V_{f} [k_1(n)] - v_1(n)]}{k_1(n)} \right] + \frac{T}{\Delta T} \left[ k_1(n) - k_2(n) \right] + \frac{\Delta L_i}{k_1(n)} (k_1(n) - k_2(n)) + \frac{\Delta L_i}{k_1(n)} (k_1(n) + \kappa)
\]  

where \( \kappa \) and \( \kappa' \) are positive constants that prevent the convection and anticipation terms from becoming too large and

\[
\mu(n) = \begin{cases} 
\mu_1 & \text{if } k_{\text{max}}(n) > k_1(n) \\
\mu_2 & \text{otherwise}
\end{cases}
\]  

where \( \mu_1, \mu_2, \rho, \) and \( \sigma \) are constants to be determined and the equilibrium speed is modeled by

\[
V_e(k_1) = \frac{1}{m} \left( 1 - \frac{k_1}{k_{\text{max}}} \right)^m
\]  

where \( V_e, k_{\text{max}}, l, \) and \( m \) are constants to be identified for real traffic flow.

The traffic flow models (2) to (8) contain several parameters whose values depend on the traffic flow characteristics that vary according to driving behavior and traffic pattern. One way of choosing these parameters is to use real traffic data to validate the models. A validated model, however, that is accurate for one place at a particular time of the day may not hold in another place at another time. The nonlinear and unpredictable characteristics of traffic flow dynamics make it difficult to have a universal traffic flow model that applies to all traffic situations at all times.

A promising approach to model traffic flow is to use a neural network that is continuously training itself to possible changing patterns of traffic and is therefore adaptive to changes and driving behaviors. Such a neural network model could then be used for control purposes as discussed in the following sections in order to improve traffic flow characteristics. The general form of the traffic flow models (2) to (6) is used to choose the structure of the ANN.

**Fundamentals of Artificial Neural Networks**

The development of artificial neural networks or connectionist models has been inspired by the nervous system of human brain. An artificial neural network comprises many basic processing elements (PES) connected in certain parallel structure. Each processing element or neuron is described by a nonlinear algebraic or differential equation. Associated with each interconnection there is an adjustable parameter or weight that changes according to a certain learning rule. According to their structures: neural networks can be broadly categorized as either feed-forward networks or recurrent networks. Feed-forward networks are also called static networks since the flow of input signal or information is directed to output and no returning paths are allowed. However, in recurrent networks, which are also called...
Fig. 2. A processing element.

Fig. 3. A 1.5-layer feed-forward network.

dynamic networks, either states or outputs can be fed back and a current signal is dependent upon the past one. This makes recurrent networks significantly different from feed-forward networks. Once the structure and a learning rule are given, a neural network is completely defined.

Here, we will focus on one of the most popular neural networks, called sigmoidal feed-forward networks. The output of the processing element is the nonlinear sigmoidal function of the sum of the inputs and a possible threshold. A sigmoidal function $u(s)$, which is nondecreasing, satisfies the conditions that $o(-\infty) = 0$, and $o(\infty) = 1$.

One choice of the smooth sigmoidal functions is expressed by

$$o(x) = \frac{1}{1 + e^{-cx}}$$

where $c$ is a constant determined by the shape of the function.

Thus, the mathematical expression of a processing element with input $x_i$, $i = 1, ..., n$ and output $y$, shown in Fig. 2, is

$$y = o\left(\sum_{i=1}^{n} w_i x_i + w_{n+1}\right)$$

where $w_i$ are adjustable weights.

The PEs are arranged in layers with signals flowing forward from input to output. It has been shown by Hornik [6] that multilayer feed-forward networks with as few as one hidden layer are capable of universal approximation in a very precise and satisfactory sense. Shown in Fig. 3 is a 1.5-layer feed-forward network with $p$ processing elements, the outputs $y_i$, $i = 1, ..., m$ of which each is just a weighted sum of the outputs of the hidden layer. The mathematical expression of this network is given by

$$y_j = \sum_{i=1}^{n} t_{ij} o\left(\sum_{w_i} w_{ij} x_i + w_{n+1,j}\right)$$

(11)

where $t_{ij}$ and $w_{ij}$ are adjustable weights.

Let $X = [x_1, ..., x_n]^T$, $Y = [y_1, ..., y_m]^T$, and $\theta = [w_{11}, ..., w_{np}, t_{11}, ..., t_{pm}]^T$. Then the input-output relationship of the neural network can be expressed as

$$Y = N(X; \theta)$$

(12)

Let $Y_p = [y_{p1}, ..., y_{pm}]^T$, be the output vector of an unknown system with the same input $X = [x_1, ..., x_n]^T$, i.e.,

$$Y_p = F(X; \theta)$$

(13)

where $F(., \cdot)$ is a completely unknown function. The difference between $Y_p$ and $Y$ is the error given by

$$\Delta = Y - Y_p = N(X; \theta) - Y_p$$

(14)

If we now adjust the weights $\theta$ such that $\epsilon \rightarrow 0$, i.e., $Y \rightarrow Y_p$, then the input output characteristics of the unknown system (13) are matched by the neural network (12). The adjustment mechanism for $\theta$ may be developed using simple optimization techniques to minimize a certain cost function of $\epsilon$ [18, 19].

Modeling of Highway Traffic Using Neural Networks

Assuming that there is no on-ramp or off-ramp, the most general form of the density equation in the traffic flow models discussed in the foregoing section is

$$k_i(n+1) = g_i(k_i(n), v_{i-1}(n), k_{i-1}(n), v_i(n), k_{i+1}(n), v_{i+1}(n))$$

(15)

where $k_i(n)$ is the mean density at section $i$ at time $nT$ and $g_i$ is an unknown nonlinear function. Without loss of generality we express (15) in the form

$$k_i(n+1) = c k_i(n) + f_i(k_i(n), v_{i-1}(n), k_{i-1}(n), v_i(n), k_{i+1}(n), v_{i+1}(n))$$

(16)

where $c l$ is an arbitrary constant and

$$f_i(n) = g_i(n) - c k_i(n)$$

Our objective is to develop a neural network that identifies the nonlinear function $f_i$ that is used to generate the estimates $\hat{k}_i(n+1)$ that are as close to the measured $k_i(n+1)$ as possible for $i = 1, 2, ..., N$. We achieve this objective as follows: Let $N_{f_i}(\cdot; \theta_{f_i})$ be the neural network approximation of the nonlinear function $f_i$, where $\theta_{f_i}$ is a vector of the weights of the network. Then the following model generates the density estimates

$$\hat{k}_i(n+1) = c k_i(n) + N_{f_i}[k_i(n), v_{i-1}(n), k_{i-1}(n), v_i(n), k_{i+1}(n), v_{i+1}(n); \theta_{f_i}(n)]$$

(17)

that correspond to the weights $\theta_{f_i}(n)$ at time $nT$. 

October 1996
Control of Traffic Flow Using Neural Networks

Based on the neural network techniques described in the previous section, our objective is to design a controller that will generate speed commands to be followed by the vehicles at the various sections of the freeway lane so that a desired traffic flow rate can be achieved by maintaining a desired density profile. The speed commands are generated according to the desired density at each section.

We assume that there exists an inverse mapping $g_i^{-1}$ so that from (15) we obtain

$$v_i(n) = g_i^{-1}(k_{i-1}(n), v_{i-1}(n), k_i(n), k_i(n+1), v_{i+1}(n))$$

and

$$k_i(n+1) = g_i(k_{i-1}(n), v_{i-1}(n), k_i(n), g_i^{-1}(k_{i+1}(n), v_{i+1}(n)))$$

The speed commands to the vehicles at each section $i$ and each time $nT$ are generated as follows: First, a 1.5-layer feed-forward network is trained to approximate the inverse mapping $g_i^{-1}$ as shown in Fig. 5, where $h_i = [k_i(n), v_{i-1}(n), k_i(n+1), v_{i+1}(n)]^T$ is a measure of the deviation of the estimated $k_i$ from the actual measured $k_i$ at time $nT$.

The adjustment rule for the weights $\theta_f$ is chosen as [19]

$$\mu_i(n+1) = \theta_f(n) - \frac{\gamma_0}{\beta_0 + \|z_i(n)\|^2} z_i(n) e_i(n)$$

$$\theta_f(n+1) = \begin{cases} \mu_i(n) + M_0 \mu_i(n+1) & \text{if } |\mu_i(n+1)| \leq M_0 \\ M_0 \mu_i(n+1) & \text{if } |\mu_i(n+1)| > M_0 \end{cases}$$

(19)

where $0 < \gamma_0 < 2$, $\beta_0 > 0$, and $M_0 > 0$ are design parameters and

$$z_i(n) = \frac{\partial^2 N_f(h_i; \theta_f)}{\partial \theta_f}$$

We illustrate $N_f$ in the neural network configuration for section $i$ by Fig. 4.

It can be shown [19] that if $N_f$ is linear with respect to $\theta_f$ and $f_i$ can be parameterized to be of the same form as $N_f$ with a corresponding unknown $\theta_f^*$ then (19) guarantees that $e_i(n) \to 0$ as $n \to \infty$.

Fig. 4. Neural network configuration for section $i$.

Fig. 5. Training scheme for $N_{f_i}$.

The error

$$e_i(n) = \hat{k}_i(n) - k_i(n) = N_f(h_i; \theta_f(n)) - f_i(n)$$

(18)

where $h_i = [k_i(n), v_{i-1}(n), k_i(n+1), v_{i+1}(n)]^T$ is a measure of the deviation of the estimated $k_i$ from the actual measured $k_i$ at time $nT$.

The adjustment algorithm is listed below:

$$\theta_f(n+1) = \begin{cases} \mu_i(n+1) + M_0 \mu_i(n+1) & \text{if } |\mu_i(n+1)| \leq M_0 \\ M_0 \mu_i(n+1) & \text{if } |\mu_i(n+1)| > M_0 \end{cases}$$

(20)

where $0 < \gamma_0 < 2$, $\beta_0 > 0$, and $M_0 > 0$ are design parameters.

We denote the output of the neural network $\hat{v}_i$ as given by the expression

$$\hat{v}_i(n) = N_{g_i}(k_{i-1}(n), v_{i-1}(n), k_i(n), k_i(n+1), v_{i+1}(n); \theta_{g_i})$$

Secondly, assume that the desired density at section $i$ is $k_d$. We set the input, $k_i(n+1)$, of $N_{g_i}$ to $k_d$ and denote the output as $\hat{v}_i$. Our controller is shown in Fig. 6, where $\hat{v}_i(n)$ is given by
It should be noted that even if $N_{k_0}$ is perfectly trained, $\hat{v}(n)$ is not equal to $v(n)$ since $k_{d_0}$ is not necessarily equal to the current $k_{d}(n+1)$. However, if $\hat{v}(n)$ converges to $v(n)$, the density $k_i$ is expected to approach $k_{d_i}$.

The above controller is an open-loop one and cancels the dynamics of the system using feed-forward action. Such controllers are known to have poor robustness properties with respect to modeling errors that are always present in real applications. This drawback of the neural network controller of Fig. 6 can be removed using the following approaches:

### Approach I

We consider the use of output feedback in Fig. 6 to obtain the closed-loop system shown in Fig. 7, where $F$ is a linear filter of the form

$$F(e_k(n)) = k_d + F_e(e_k(n))$$

where $F_e(\cdot)$ is an arbitrary linear filter and $e_k(n) = k_i(n) - k_d$. Since the neural network controller acts as the inverse model of the traffic flow dynamics, this closed-loop control falls into the class of internal model control (IMC) paradigm [9].

### Approach II

Adaptive control is another technique in dealing with uncertainties. Shown in Fig. 8 is a system configuration for direct adaptive control. It is built upon the IMC feedback configuration. $F(\cdot)$ is fixed; however, the weights of the neural network $N_{k_0}$ or the parameters of the controller in $\theta_{k_0}$, are adaptively adjusted based on Adjustment Law 2. Adjustment Law 2 has the same form as (19). The error for Adjustment Law 2 is defined as the difference between the desired density and the output of the forward neural network model, i.e., $k_i(n+1) - k_d$, and the sensitivity function is

$$\xi(n) = \frac{\partial^T N_{k_0; \theta_{k_0}, \theta_{k_0}}}{\partial \theta_{k_0}}$$

where $N_{k_0; \theta_{k_0}, \theta_{k_0}}$ represents the two 1.5-layer neural networks, $N_{k_0}$ and $N_{k_0}$, cascaded together. Adjustment Law 1 provides on-line adaptation and is the same as the off-line training law (19). However, Adjustment Law 2 assumes that the parameter vector $\theta_{k_0}$ has been accurately trained and is kept constant when $\theta_{k_0}$ is adaptively adjusted. With this assumption,
Table 1. Initial Conditions of Sectional Density and Speed in a Highway Lane

<table>
<thead>
<tr>
<th>Section No. (i)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_i(0) \text{ vehicle/km})</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>52</td>
<td>52</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>(v_i(0) \text{ km/h})</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>29</td>
<td>29</td>
<td>81</td>
<td>81</td>
<td>81</td>
</tr>
</tbody>
</table>

\[ e_i(n) = \frac{\partial^T N_i}{\partial \theta_i} \left[ k_i; \theta_{i-1}, \theta_{i+1} \right] \frac{\partial^T \gamma(n)}{\partial \theta_{i-1}} \]

The overall equations of the neural network adaptive controller in Fig. 8 are listed below:

\[ e_i(n) = k_i(n) - k_{0i} \]

\[ F(e_i(n)) = k_{0i} + F_k(e_{0i}(\varepsilon)) \]

\[ e_i(n) = \hat{e}_i(n+1) - k_{0i} \]

\[ \hat{e}_i(n) = \frac{\partial^T N_i}{\partial \theta_i} \left[ k_i; \theta_{i-1}, \theta_{i+1} \right] \frac{\partial^T \gamma(n)}{\partial \theta_{i-1}} \]

\[ \mu_i(n+1) = \theta_{i-1}(n) - \frac{\gamma_0}{\beta_0 + \|\theta_i(n)\|^2} \hat{e}_i(n) \]

\[ \theta_{i-1}(n+1) = \left\{ \begin{array}{ll}
\mu_i(n+1) & \text{if } |\mu_i(n+1)| \leq M_\theta \\
\frac{\mu_i(n+1)}{|\mu_i(n+1)|} & \text{if } |\mu_i(n+1)| > M_\theta
\end{array} \right. \]

where \(0 < \gamma_0 < 2\), \(\beta > 0\), and \(M_\theta > 0\) are design parameters and \(F_k\) is a linear filter.

Since \(\hat{e}_i(n+1)\) is assumed to be a good estimate for \(k_{i}(n+1)\), as long as \(\theta_{i-1}\) are adjusted such that is driven to \(k_{0i}\), \(k_{i}(n+1)\) is expected to track \(k_{0i}\).

The above controllers are based on heuristics and their stability analysis is not established due to the complexity of the nonlinearities in the traffic flow model and neural networks. Their effectiveness can be evaluated by extensive simulations, as discussed in the next section.

Simulated Example

Congestion Due to Density Inhomogeneity

Consider a highway lane that is subdivided into ten sections \((N = 10)\) and at a certain time instant the density distribution is shown in Fig. 9, where sections 6, 7, and 8 have higher densities than the other sections. This inhomogeneity in traffic density is assumed to be caused by an accident at section 8.

The traffic flow instability can be predicted by the model of Karaaslan, Varaiya, and Walrand described by Equations (2), (3),
and (6)-(8) as shown in Figs. 10 and 11, where congestion caused by a disturbance due to the accident at section 8 remains and gets worse even though the initial disturbance was removed at $t = 0$. This congestion situation is used as an example for applying the proposed neural controllers. The parameter values in the model used for simulation are: $v_f = 93.1$ km/h, $k_{\text{low}} = 110$ veh/km, $l = 1.86$, $m = 4.05$, $\alpha = 0.95$, $\kappa = 4$ veh/km, $\mu_1 = 12$ km$^2$/h, $\mu_2 = 6$ km$^2$/h, $\rho = 120$ veh/km, $\sigma = 35$ veh/km, $\Delta t = 20.4$ seconds, $L_t = 500$ meters for $i = 1, 2, ..., 10$ and $T = 15$ seconds. The initial densities and mean speeds at all ten sections of the freeway are listed in Table 1.

Training of Neural Network Model

Consider a highway lane that is subdivided into ten sections. We assume that all the sections in the freeway are the same and their behavior can be described by the model of Karaaslan, Varaiya, and Walrand using Equations (2), (3), and (6)-(8). In Fig. 4, we use a 1.5-layer feed-forward network of ten PEs ($\pi = 10$) as in Fig. 3 for $N_i$. Two hundred sets of training data for the ten-section highway lane (2,000 pairs of density-velocity input data and 2,000 pairs of output data) are generated according to the model of Karaaslan, Varaiya, and Walrand by randomly selecting the initial densities and velocities that distribute uniformly in their possible ranges ($0 < \nu(n) < 130, 0 < k(n) < 110$). The constant in the sigmoidal function (9) is chosen as $c = 0.5$. The parameters in the adjustment law (19) are chosen as: $\gamma_0 = 0.1, \beta_0 = 1.0, M_\theta = 10$. We have also arbitrarily chosen $c = 0.5$ in the density dynamic equation (16). The performance of the learning of a neural network can be evaluated by the mean square error, which is defined by

$$\text{MSE}(n) = \frac{1}{n} \sum_{i=1}^{n} e(i)^2$$

(21)

where $e$ is the error signal. As shown in Fig. 4, $e$ is defined as $\hat{k}_i(n+1) - k_i(n+1)$. During the course of training, the mean square error is decreasing and approaches a small number. Shown in Fig. 12 is the converging mean square error for the network $N_i$.

Similarly, we use another 1.5-layer feed-forward network of ten PEs for $N_i$ in Fig. 5. The same training data used to train $N_i$ are used to train $N_i$. The constant in the sigmoidal function (9) is also chosen to be $c = 0.5$. The parameters in the adjustment law (20) are chosen as: $\gamma_0 = 0.1, \beta_0 = 1.0, M_\theta = 10$. The error signal used to calculate the mean square error in (21) is defined as $\hat{\nu}_i(n) - \nu_i(n)$. Shown in Fig. 13 is the converging mean square error for the network $N_i$. After $10^6$ iterations of training, the network $N_{i-1}$ is able to generate the estimates of the mean speed with accuracy. Shown in Fig. 14 are a few samples of the output of the trained network $N_{i-1}$. The solid line represents the mean speed generated by the model of Karaaslan, Varaiya, and Walrand, and the dashed line is the output from the network $N_{i-1}$. 

October 1996
Open-Loop Control

Take the well-trained network $N_{d^{-1}}$ as the controller in Fig. 6. We assume that the desired density for all the sections is $k_d$ and substitute $k_d$ for $k_d$ as an input to $N_{d^{-1}}$. The initial conditions are set to be the same as in the uncontrolled case, which has been shown to cause congestion. By assigning $k_d = 35 \text{ veh/km}$, the controller generates the speed commands as shown in Fig. 15 for vehicles at each section to follow. In the meantime, the traffic flow instability is being removed and the congestion being alleviated. Fig. 16 shows that the traffic flow is able to track the desired density accurately in less than 15 minutes.

Internal Model Control

Assume that the dynamics of the traffic flow described by the model of Karaaslan, Varaiya, and Walrand in (3) has changed such that the weighting factor $\alpha$ is 30% smaller than the nominal value of 0.95. In the face of this uncertainty, the open-loop controller is not able to effectively control the traffic flow with the same initial conditions as in the uncontrolled case. However, the robust internal model controller in Fig. 7 with

$$F(e_d(n)) = k_d(n) + e_d(n)(1 - e^{-1})$$

where $e_d(n) = k_d(n) - k_d$, can tolerate the uncertainty. The speed commands generated by the feedback controller are shown in Fig. 17. If the commands are followed by the vehicles at each section, the traffic flow will track the desired density pretty accurately as shown in Fig. 18.

Neural Network Adaptive Control

Assume that the parameter vector $\theta_{e^{-1}}$ for the network $N_{e^{-1}}$ in Fig. 5 is inaccurately trained or the traffic flow parameters have changed. For the sake of simplicity, in simulation we randomly select $\theta_{e^{-1}}$ such that each weight in $\theta_{e^{-1}}$ is either 15% smaller or 15% larger than the nominal value which has been correctly trained for the inverse model $N_{e^{-1}}$. With this uncertainty, the open-loop controller is no longer effective in control.
The ANNs structures and simulated using a nonlinear traffic flow to adjust its parameters in Fig. 19. Adaptively controlled speed commands: $k_d = 35$.

The speed commands generated by the adaptive controller are shown in Fig. 19 and the traffic flow is shown to track the desired density with accuracy in Fig. 20.

Conclusions

In this article we consider the use of Artificial Neural Networks (ANNs) for modeling and controlling traffic flow on the macroscopic level. The ANNs consist of layers and their weights are adjusted using robust adaptive laws developed in the area of adaptive control.

The proposed ANNs structures can be trained to model traffic flow quite accurately. Several controllers are designed based on the ANNs structures and simulated using a nonlinear traffic flow model that describes traffic flow in a single highway lane. The controllers accept measurements of the average density and speed at each section of the lane and each sampling time and generate the appropriate speed commands to be followed by the vehicles. The control objective is to force the actual traffic density to converge to a desired one.

Extensive simulations have been performed to demonstrate the effectiveness of ANNs controllers in managing congestion and improving traffic flow.

The methodology proposed in this article can be easily extended for a full-scale traffic speed and flow control problem. Traffic on a multiple-lane freeway involves lane changes and on/off ramp metering. The effect of lane changes can be regarded as noise to the traffic density, and the problem reduces to the training of neural networks with noisy data. Traffic density also depends on the values of the flow rates of the on/off ramp traffic.

In this article we assume no on/off ramps for the portion of the lane under consideration. Our methodology can be extended to lanes with on/off ramps. In this case a strategy needs to be developed to control the volume of the traffic at the ramps in addition to the control of the traffic density in the lane. This problem is currently under investigation.

References


Fu-Sheng Ho received the B.Sc. degree in mechanical engineering from National Taiwan University in 1983 and the M.S. and Ph.D. degrees, both in electrical engineering, from Virginia Polytechnic Institute and State University, Blacksburg, VA, in 1988 and 1993, respectively. In 1993, he joined the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA. He is currently a Research Assistant Professor in the same department. His research interests include adaptive control, neural networks, industrial automation, and automated highway systems.

Petros Ioannou received the B.Sc. degree with first-class honors from University College, London, England, in 1978 and the M.S. and Ph.D. degrees from the University of Illinois, Urbana, IL, in 1980 and 1982, respectively. From 1975 to 1978, he held a Commonwealth Scholarship from the Association of Commonwealth Universities, London, England. He was awarded several prizes, including the Goldsmith Prize and the A.P. Head Prize from University College, London.

From 1979 to 1982, he was a research assistant at the Coordinated Science Laboratory at the University of Illinois. In 1982, he joined the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA. He is currently a professor in the same department and the Director of the Center of Advanced Transportation Technologies. He teaches and conducts research in the areas of adaptive control, neural networks, and intelligent vehicle and highway systems. In 1984, he was a recipient of the Outstanding Transactions Paper Award for his paper "An Asymptotic Error Analysis of Identifiers and Adaptive Observers in the Presence of Parasitics," which appeared in the IEEE Transactions on Automatic Control in August 1982. He is also the recipient of a 1985 Presidential Young Investigator Award for his research in adaptive control. He was an Associate Editor for the IEEE Transactions on Automatic Control from 1987 to 1990 and he is currently on the editorial board for the International Journal of Control. Dr. Ioannou is a member of SAE, IVHS America, and of the AVCS Committee of IVHS America.