

Application of Elementary Neural Networks and Preview Sensors for Representing Driver Steering Control Behaviour

CHARLES C. MACADAM* and GREGORY E. JOHNSON**

SUMMARY

This paper demonstrates the use of elementary neural networks for modelling and representing driver steering behaviour in path regulation control tasks. Areas of application include uses by vehicle simulation experts who need to model and represent specific instances of driver **steering control** behaviour, potential on-board vehicle technologies aimed at representing and tracking driver steering control behaviour over time, and use by human factors specialists interested in representing or classifying specific families of driver steering behaviour. Example applications are shown for data obtained from a **driver/vehicle** numerical simulation, a basic driving simulator, and an experimental on-road test vehicle equipped with a camera and sensor processing system.

1. INTRODUCTION

This analysis illustrates the use of basic neural network concepts, combined with appropriate sensor information, to represent driver steering control behaviour in path regulation activities. Identification of driver steering behaviour is the goal, based upon "observation" or processing of assumed sensor inputs that provide information about vehicle motion and orientation with respect to the roadway. Use of time delayed sensor inputs is a key feature of the technique proposed here. The motivations for using neural network approaches can range widely, but normally stem from some need to represent adaptive control system behaviour. For vehicle control applications this frequently means *human* control behaviour. In the area of **driver/vehicle** simulation, neural networks enable the modelling of specific **driver/vehicle** control interactions. Another area of interest is in emerging IVHS (Intelligent Vehicle Highway Systems)

• Associate Research Scientist, The University of Michigan Transportation Research Institute, Ann Arbor, Michigan, 48109, USA.

**Engineer in Research, The University of Michigan Transportation Research Institute, Ann Arbor, Michigan, 48109, USA.

Correspondence should be addressed to: Dr. Charles **MacAdam**, The University of Michigan, UMTRI, 2901 Baxter Road, Ann Arbor, MI 48109, USA. Telephone: (313) 936-1062, Fax: (313) 936-1068, E-Mail: cmacadam@umich.edu.

on-board control technologies presently being studied and possibly introduced into certain vehicle markets. IVHS concepts such as lane departure "co-driver" assists [1,2,3,4], aimed at mitigating run-off-road accidents and/or regulation of lateral lane position, as well as longitudinal control systems for "intelligent cruise" [5], headway control [6], and collision avoidance systems [7], are examples of such target applications. Each application is presumed to have a need for representing driver control behaviour as part of its basic intelligence scheme. In the case of lane departure co-driver assists, future path predictions relative to the highway lane depend, in good part, upon the current driver steering input. In the case of intelligent cruise control systems, some representation or understanding of individual driver longitudinal acceleration preferences would be presumed, based upon observations by the system of actual driver longitudinal control behaviour.

The desirability of designing control technologies that mimic human-like behaviour is frequently a motivating reason for applying neural network concepts [3,8,9,10,11,12,13]. The design and creation of control technologies compatible with human expectations, in a traditional human-machine system, are important for the acceptance and use of such new technological assists by average drivers [14]. There is also the practical, common sense observation by most drivers that we are all different and display individual driving styles. Such differences need to be addressed by any control technology expected to assist and interact cooperatively with us during the driving experience. The issue of adaptive control then becomes a key design element that must be a part of any such system intended to provide the necessary robustness when dealing with the myriad of possible driver behavioural patterns and expectations. The well known ability of neural networks to "learn," or to recognize patterns of behaviour [3,8,9,10,11,12,13,15,16,17,18,19], is clearly seen as a desirable quality greatly benefiting the designer of such human-machine cooperative systems. It is this adaptive property of neural networks that is being utilized in this paper to illustrate how driver steering control behaviour can be modelled and identified over time. For some applications, such as on-board systems focused on vehicle control, this may imply some form of periodic updating of the driver representation.

Previous research particularly related to the topic of this paper appears in references [3,4,9,10,11,20,21,22,23,24]. Cheng et al [20] used a backpropagation neural network as a controller for an automated guided vehicle system. Camera images were used as inputs to the network to then generate a command reference yaw rate signal as a control input for the vehicle. Fujioka and Takubo [21] used a neural network approach to model driver steering behaviour using driving simulator data to train the network. Kehtarnavaz and Sohn [22] illustrated how neural networks can be used to control the steering of an autonomous vehicle when equipped with passive or active sensors providing range and heading angle information. The networks were trained using real data obtained from vehicle-tracking test runs. Kornhauser et al and Lubin et al

[23,24] used a graphical driving simulator environment to gather human response data for training candidate neural networks. The goal was to investigate various designs of artificial neural networks for processing the resulting images and then generating acceptable steering commands for the vehicle. Excellent results were reported for both straight and curved road driving scenarios under a variety of conditions. Kraiss and Kuettelwesch [3,4] demonstrated how neural networks could be trained to guide a vehicle through an obstacle course by emulating a human teacher. A two lane car driving task was also used to illustrate the training procedure. Kraiss and Kuettelwesch also proposed and demonstrated the use of neural networks as an information source for driver assistant systems. Recently Neusser et al [9] used measurements of human drivers to train a neural network controller to autonomously steer and guide a vehicle along public highways using optical sensor inputs. Last, but not least, the important work of Pomerleau, Thorpe, and others at Carnegie-Mellon [10,11] using the ALVTNN system for autonomous steering control of vehicles, is likewise noted. Recent progress in training efficiency has been reported for ALVINN by using human steering response data. Once trained for a particular type of road, autonomous control of the vehicle by ALVINN has been demonstrated at highways speeds.

As indicated above, identification of driver steering control behaviour is the goal of this paper, based upon neural network processing of assumed sensor inputs that provide information about vehicle motion and orientation relative to the roadway. A distinguishing feature of this work is the use of time-delayed sensor information and the use of driver/vehicle response training data obtained from several different sources. The lateral path control problem serves as the basis for the discussion that follows.

2. DESCRIPTION OF ASSUMED SENSOR MEASUREMENTS

Potential on-board systems focused on lateral control issues are assumed to include some form of sensor package that can provide continuous information about the motion of the vehicle on which it is mounted relative to the roadway. This would likely include lateral displacement information, as well as heading or yaw information relative to the roadway orientation. The paper assumes a sensor that is capable of looking forward and to the side of the vehicle to provide such relative motion information as, for example, in [25]. It is also assumed that relative displacement information at several distinct points over the look-ahead or preview interval of the sensor are its basic outputs.

The specific sensor pattern described below forms the basis for all sensed motion available to, and flowing into, the sensor processing system considered here. That is, all relative lateral displacements available from the preview sensor (looking forward and to the side of the vehicle), are assumed to be available for

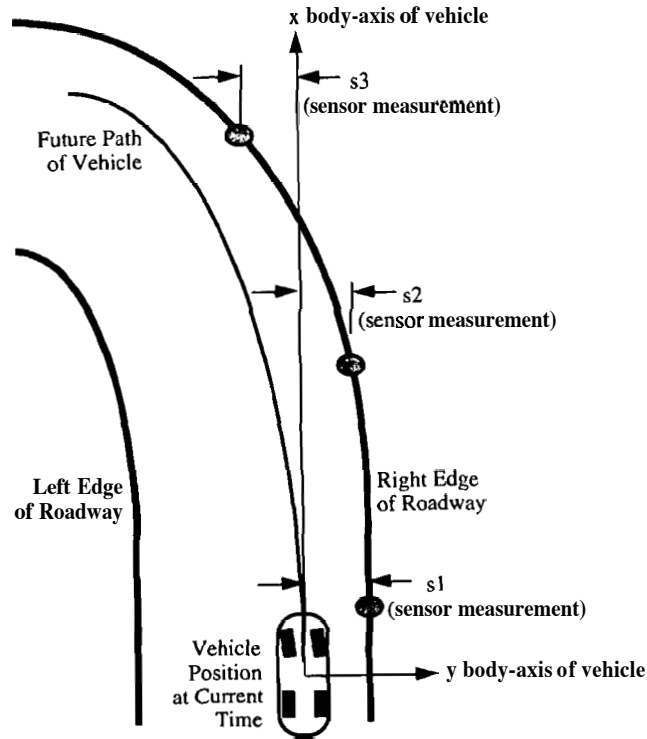


Fig. 1. Overhead view of vehicle with assumed preview sensor measurements.

several distinct locations ahead of the vehicle in time (or position). The sensor measurements are also assumed to be independent of any specific hardware implementation (laser, CCD camera, infrared, etc.) In the examples used in this paper, three total samplings of relative vehicle positions are assumed available; one is very near the front end of the vehicle, and the remaining two samples are located at two points ahead of the vehicle (typically separated at 0.2 to 0.5 second intervals, or corresponding travel distances based upon the speed of the vehicle). See Figure 1.

As seen in Figure 1, the relative lateral displacements measured by the sensor are labeled as s_i ($i = 1, 2, 3$) and represent the simple lateral offsets between the forward projection of the vehicle longitudinal body axis and some common road edge marker (e.g., the typical white fog line used for marking lane edges, or some equivalent and generally detectable feature of the road edge). Consequently, at any point in time, several simultaneous measurements over the immediate preview interval are output by the sensor and represent lane edge positions of the roadway ahead, relative to the current forward projection of the longitudinal body-axis of the vehicle. Note that these displacement measurements from the sensor are not future path errors (as traditionally described),

unless the vehicle is currently traveling in a perfectly straight, zero-sideslip condition with no steering input. (Future path error information can be obtained, if needed, by projecting the future path of the vehicle relative to the longitudinal body axis of the vehicle, and then subtracting the described sensor measurements.)

To understand the reasoning behind using several previewed sensor inputs, it is helpful to note what kinds of additional information are available (in an approximate manner) when simple summing and differencing operations are performed on these basic sensor outputs. For example, the *difference* between the intermediate sensor measurement and the immediate near-field measurement, or $(s_2 - s_1)$, divided by the longitudinal distance between them, is an approximate measure of the angle between the current vehicle heading and the nearby road orientation, referred to subsequently as "relative yaw angle." Likewise, the difference between two nearby relative yaw angles (i.e., second differences), yields some approximate information about upcoming roadway curvature. In a similar manner, weighted *sums* of the same sensor outputs can provide lateral displacement information at points between the locations of the sensor measurements. For example, the sum of the first two sensor measurements, divided by 2, provides an estimate of the relative lateral displacement midway between these two respective sensor locations. Consequently, "intelligent" weighting of these basic sensor outputs provide relevant information about the vehicle orientation, its location relative to the immediate roadway, and upcoming changes in roadway direction.

To extract *time derivative information* from these same basic sensor measurements, time-delayed versions of these signals can be utilized. For example, the difference between one relative yaw angle measurement and the same yaw angle measurement *delayed in time*, provides a signal estimate proportional to the vehicle yaw rate. Similarly, differences between these derived vehicle yaw rates in time can provide estimates of vehicle yaw acceleration. Thus, by providing an augmented set of network inputs that include the three immediately sensed lateral displacement measurements $\{s_1(t), s_2(t), s_3(t)\}$ and two corresponding sets of their time delayed responses $\{s_1(t-\tau_1), s_2(t-\tau_1), s_3(t-\tau_1)\}$, $\{s_1(t-\tau_2), s_2(t-\tau_2), s_3(t-\tau_2)\}$, approximate time derivative information becomes available to the network as additional inputs. The values τ_1 and τ_2 represent the time lags of the delayed inputs.

To understand why such sensor inputs may be desirable, it has been shown in numerous prior studies of closed-loop driver modelling research [26,27,28,29,30] that driver steering control depends to varying degree upon a variety of vehicle motion cues. Key feedback cues used by drivers for path regulation typically identified in such studies include previewed errors associated with vehicle lateral position and yaw angle, as well as driver sensations of yaw rate, sideslip, and lateral acceleration. Accordingly, any proposed neural network aimed at representing the steering control behaviour of human drivers would logically require sensed information consistent with these same types of

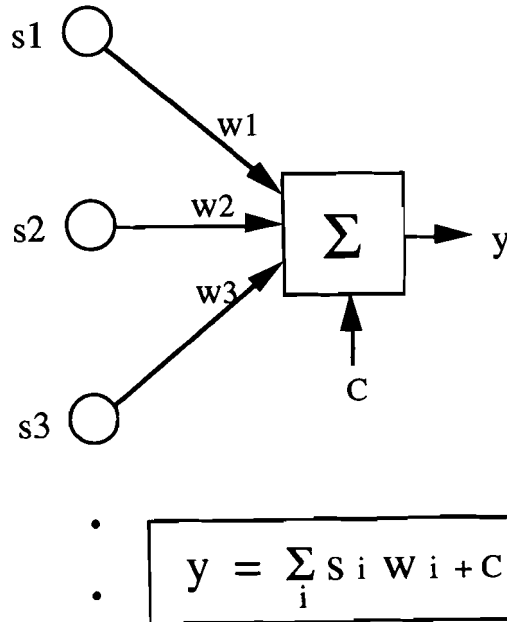


Fig. 2. Single input layer, single output neuron.

basic vehicle responses, namely, estimates of lateral position, yaw angle, and their corresponding rates of change.

3. NEURAL NETWORK ARCHITECTURES CONSIDERED

One of the simplest architectures for neural networks [15,16,17] is the so-called single-layer structure comprised of several inputs and a single output ("neuron") shown in Figure 2.

This type of structure provides a simple mathematical linear relationship between the inputs, s_i , and the output, y , as:

$$y = C + w_1 s_1 + w_2 s_2 + w_3 s_3 + \dots \quad (1)$$

where C is a constant bias or offset, w_i are adjustable weights, and the "neuron" is a simple summing junction. By proper selection of the weights, w_i , a suitable relationship (or linear regression) might be identified for use in certain problems. In the case of modelling closed-loop driver-steering response, many models, particularly those intended for use within the linear driving regime, lend themselves well to forms similar to that given by equation (1).

By associating y with driver steering response and the s_i with various sensor

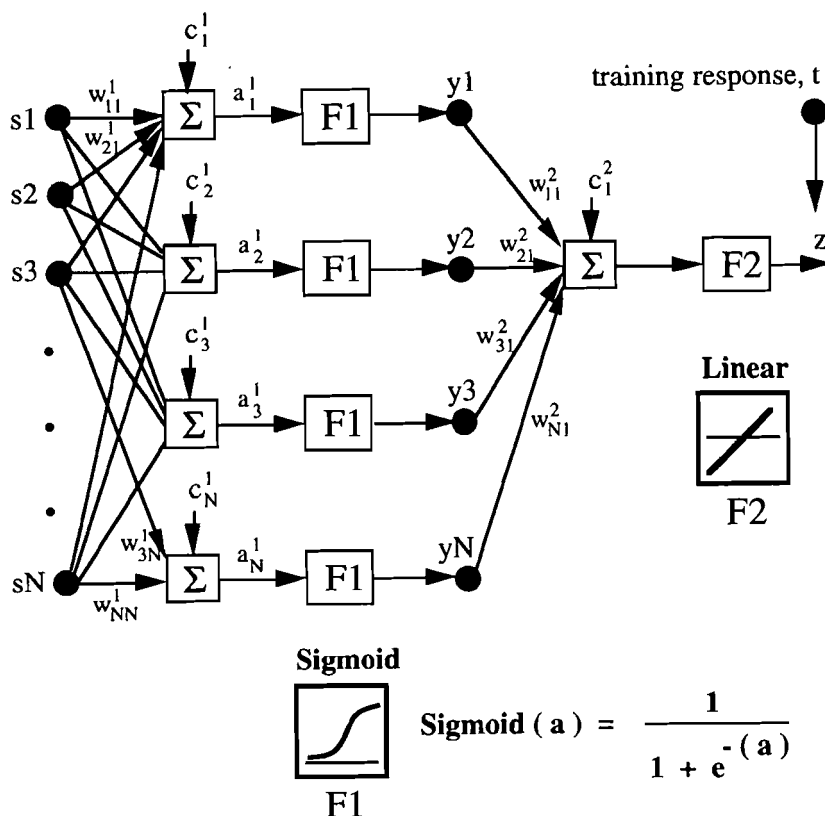


Fig. 3. Two-layer, neural network architecture with nonlinear elements

inputs described above (including the time-delayed sensor inputs), a very simple model or relationship can be proposed as a starting point for investigating the potential use of neural networks which accept lateral displacement sensor measurements as their basic input.

As mild non-linearities are introduced from such sources as steering system hysteresis (due to friction or gear box lash) and tire cornering force sensitivities (to speed or load), the need arises to extend the neural network architecture beyond that of a simple linear relationship in order to properly admit such influences. In this case, a two-layer architecture, as seen in Figure 3, might be proposed.

In Figure 3, N sensor inputs are presented to the first layer of the network and are processed through weighting values, w^1_{kj} , and a nonlinear transfer function, F1, identified as a sigmoid function, commonly used in neural networks [15,16,17]. The weighting value superscript, i, refers to the network layer; the kj subscripts denote the respective input (sensor) and output layer indices.

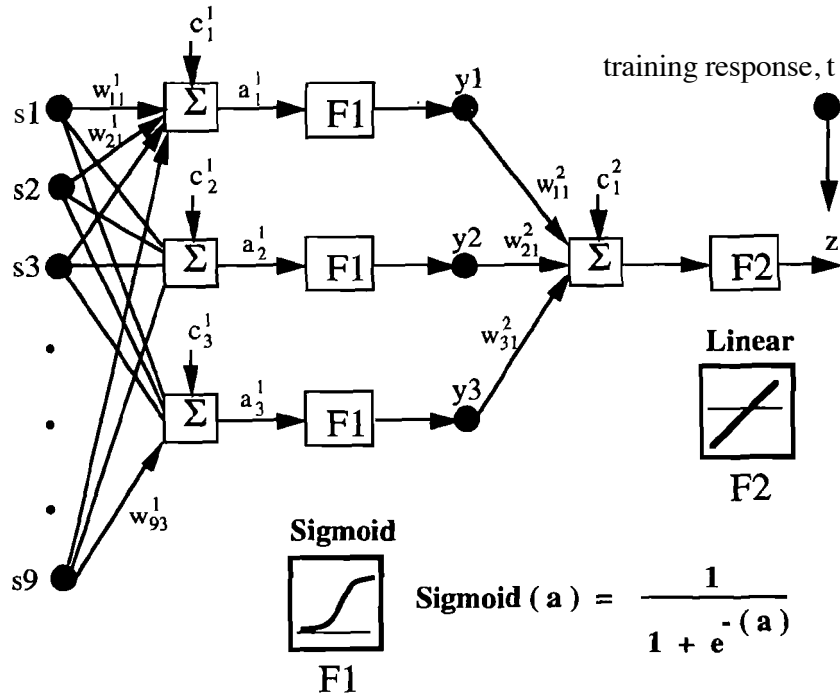
The sigmoid function defined here is output-limited to the range from 0 to 1. (This and similar nonlinear "activation functions" used in neural networks generally need to be mathematically differentiable to facilitate so-called back-propagation algorithms [15, 16, 17] used in the "training," or weight adjustment process, of the network.) The outputs of the first layer, $y_1 \dots y_N$, act as an input vector to the second layer, which in this case is comprised of a single neuron element having a linear transfer function, F_2 , and a single output, z . The N nonlinear first-layer outputs lie in the range of 0 to 1 and are then combined linearly to produce the final network output, z . If z is associated with driver steering response and the s_i ($i = 1, N$) inputs associated with the lateral path sensor inputs, the simple linear network architecture of Figure 2 is now extended to include additional nonlinear features.

Training of the network occurs when time history inputs are presented to the network and the resulting output, z , is compared to a desired response (observed or measured), referred to as the "target," t , seen in Figure 3. Errors observed at the output layer are propagated backward in the network to provide information necessary for then adjusting the various network weights prior to the next input/output cycle or time sample [15,16,17]. The goal of most training algorithms is to minimize the mean squared error at the output layer. Typically, after hundreds or thousands of such samplings and corresponding weight adjustments, the network will converge towards a set of network weight values that produces a suitable match between the desired target response and the output response calculated by the network. In this sense, network training algorithms can be viewed as a time-based parameter optimization procedure. Certain specialized numerical software packages exist to perform such calculations; reference [31] was used to implement the neural network calculations appearing subsequently in this paper.

4. DATA SOURCES

To demonstrate the feasibility of using neural networks for representing driver steering control behaviour and to study associated sensitivity issues, sample data were obtained from three separate sources: (1) numerical simulation of a driver-vehicle system with a simulated on-board sensor providing lateral displacement "measurements" like those seen in Figure 1; (2) use of a desktop driving simulator [32], also with a simulated sensor to record the same previewed path deviations; and (3) the use of an experimental full-scale vehicle, described in the next section, designed to collect data analogous to that seen in Figure 1 using an on-board video camera and processing system. In each of these cases, the path deviation data (sensor time histories) used as input to the neural network, and the corresponding driver steering responses (target time histories), were obtained by conducting routine driving maneuvers (lane-changes, curve traversals, and straight-line driving scenarios). In the case of the numerical

NEURAL NETWORKS AND PREVIEW SENSORS



s_1 = sensor 1 measurement depicted in Figure 1
 s_2 = sensor 2 measurement depicted in Figure 1
 s_3 = sensor 3 measurement depicted in Figure 1

 $s_4 = s_1(t-\tau_1)$ = time-delayed $s_1(t)$ signal
 $s_5 = s_2(t-\tau_1)$ = time-delayed $s_2(t)$ signal
 $s_6 = s_3(t-\tau_1)$ = time-delayed $s_3(t)$ signal

 $s_7 = s_1(t-\tau_2)$ = time-delayed $s_1(t)$ signal
 $s_8 = s_2(t-\tau_2)$ = time-delayed $s_2(t)$ signal
 $s_9 = s_3(t-\tau_2)$ = time-delayed $s_3(t)$ signal

 $22 > 21$

Fig. 4. Neural network architecture and sensor inputs used in this study.

simulations, closed-loop lane change and S-curve path-following maneuvers were conducted with a standard driver/vehicle model [29,30,33]. The simulation was also augmented to record the previewed lane deviations (sensor measurements) described earlier in Figure 1 as additional output variables. These recorded sensor "measurements" from the various simulated driving

maneuvers, along with their corresponding driver steering responses, were used as the first example source of time history inputs for training the proposed neural network. In the case of the driving simulator example, straight-line driving was used to collect similar types of **driver/vehicle** responses as a second independent source of training inputs for the neural network analyses. Lastly, on-road data collected by an UMTRI data acquisition system (see next section) and a human factors test vehicle [34], were used to provide yet a third source of time history data for training and analyzing the neural network. The on-road data provide a practical evaluation of the proposed concept by introducing realistic data containing potential nonlinearities and ordinary sensor alignment irregularities into the numerical processing procedure used in the neural network training.

The following Figure 4 defines the sensor input measurements used from each of the above data sources. It also shows the *specific* network architecture used in each of the calculation results that follow.

In general, the same three previewed path sensor inputs were **measured/** recorded in each of the above scenarios. From each of these three sensor inputs, six additional time history inputs were generated as time-delayed versions of the three directly measured signals. Three of these signals corresponded to a delay of τ_1 seconds; the remaining three additional sensor signals corresponded to a delay of τ_2 seconds ($\tau_2 > \tau_1$). The delay parameters τ_1 and τ_2 were typically confined to the range of 0.1 to 0.6 seconds. Note also that only three nonlinear outputs are present at the first layer of this architecture, and that these hidden layer outputs are then further combined through a single linear neuron element defining the output, z (desired driver steering response). The target training response, t , is compared with z during the network training process. Initial trials using 2 and 4 neurons in the first layer provided similar results. Additional numbers of neurons resulted in somewhat longer training times, but more accurate **matchings**. With fewer neurons, the network trained faster, but produced less accurate **matchings**. Three **first-layer** neurons were selected to provide a balance between training time and accuracy for this particular application. Typical training times for most of the example calculations appearing in this paper were a few minutes using commercial neural network software running on common workstations.

5. SENSOR AND COMPUTER HARDWARE – EXAMPLE OF AN EXPERIMENTAL ON-BOARD VEHICLE SYSTEM

An experimental UMTRI lane-edge sensor and computer system used to collect and process actual on-road image data and convert it to previewed lateral offset measurements is described in this section. One of the example neural network calculations using sample data collected by this system appears subsequently

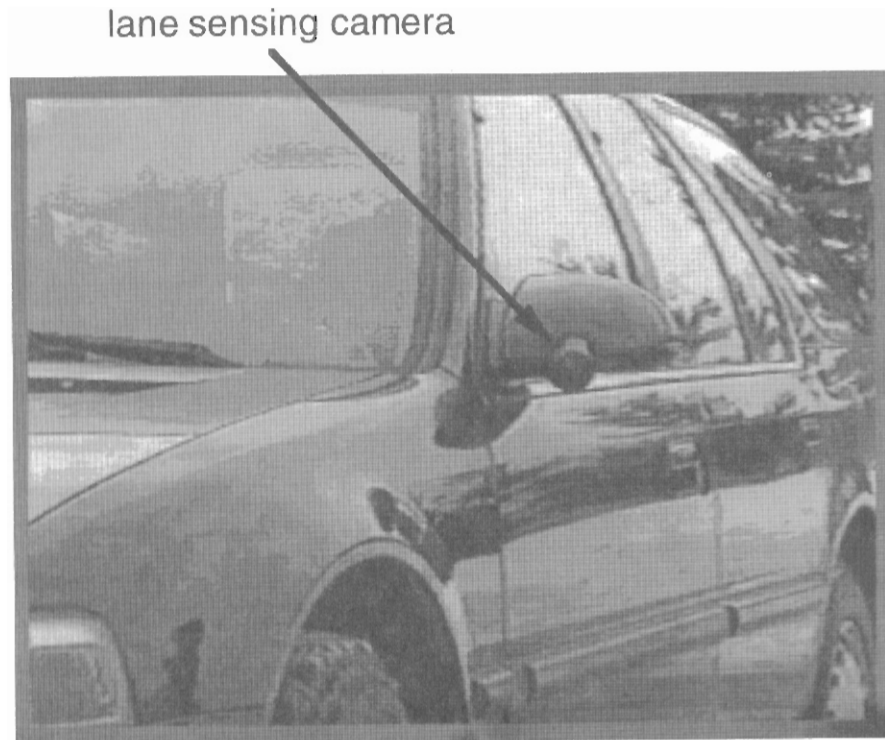


Fig. 5. Video camera mounted in driver-side mirror pod of test vehicle.

in Section 6. The described system detects white lines along road edges and computes relative lateral offsets of the vehicle at three forward locations.

Hardware

The system input is provided by a relatively small monochrome (1 inch diameter) CCD camera mounted in the driver-side mirror housing of the test vehicle. See Figure 5.

The camera is equipped with automatic gain control and a 4.8 mm auto-focus lens to provide high contrast under diverse lighting conditions. The camera produces composite video output at standard video rates (15 millisecond fields). The system's processor is a 33 MHz Intel 80486 computer, capable of producing lateral position measurements at 20 Hz. The computer uses a simple 8-bit gray scale frame grabber (512 x 480 pixels) to digitize images. The frame grabber only provides digitization and performs no other processing function.

Camera Scene and Image Segmentation

The camera is oriented to obtain a wide range of road data. An example scene is

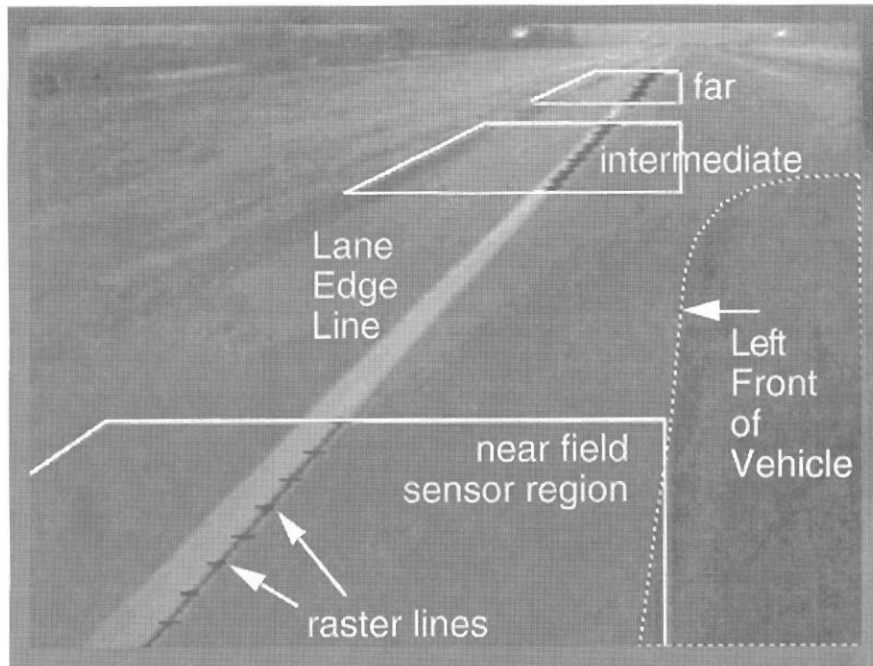


Fig. 6. Video camera scene showing the three sensor processing regions.

seen in Figure 6. The analysis regions extend longitudinally from a position just adjacent to the left front wheel out to a distance seventy feet ahead of the vehicle. The lateral range covers a width of approximately five feet at the nearest point, to full road widths at larger distances.

The camera orientation allows near and far field data to be computed using only one camera and frame grabber. At close distances the lane line may be blocked by the vehicle's front quarter, or may disappear under the tire and be obscured. At larger distances the entire road is usually visible. Since the camera is mounted off center from the vehicle's pitch and roll axes and depends upon a spatially fixed calibration procedure, large vehicle motions can impart errors into the measured data. These errors can be corrected using ancillary transducers which describe the vehicle's attitude and correct the calibration accordingly. This was not done for the relatively small motion data appearing in this paper.

Video Analysis

To collect data, the image is divided into three trapezoidal analysis regions as seen in Figure 6: a near field, a middle or intermediate field, and a far field. Each field is treated as a separate measurement covering a pre-determined longitudinal and lateral space. The shape and location of each space can be independently adjusted. The analysis proceeds in two steps: 1) the detection of a lane

edge line and its corresponding lateral displacement, and 2) the association of displacements to a lane line using the procedure in [35]. Those estimates which are correctly associated in the processing scheme represent the final sensor outputs.

Line Detection

Each trapezoidal sensor region contains a set of evenly spaced rows (image raster lines) that are analyzed. Lane edge lines are evident in rows which contain squared-off peaks relative to the mean intensity of the row. Peaks also have physical characteristics such as expected width and intensity. The threshold, TH, for peaks is determined by a multiple of the row standard deviation, s , offset by a fraction of the difference between the mean and the maximum peak value:

$$TH = K1 * s\{I(\text{row})\} + K2 * (I_{\max}(\text{row}) - E\{I(\text{row})\}) \quad (2)$$

where,

$I(\text{row})$	is a vector of intensity values for each row
$I_{\max}(\text{row})$	is the maximum peak value for the row
$E\{I(\text{row})\}$	is the mean value for the row

The parameters K1 and K2 are weights dependent upon (1) the number of lane lines being searched (the algorithm can track N lines) and (2) an estimate of the ratio of roadway pixel intensities to lane marker pixels within the row [35]. This ratio is relatively constant due to the camera's auto-iris and electronic gain control. Although the algorithm can search for N lane markings, short rows (in the far field) with large values of N (> 4) can produce frequent false targets since the statistics of such markings approach those of road clutter.

Pixel locations of suspect lane edges are transformed to lateral displacement values using a second order function. The coefficients of this function comprise the spatially-fixed calibration previously mentioned. Within each sensor region, all displacements returned must be associated to a single lane marking before being accepted as valid measurements.

Association

The association procedure generates a histogram for each sensor region, using bins representing one tenth of a foot of lateral displacement. The histogram's maximum value indicates the most consistent measurement; all measurements in this bin are retained. Other measurements within a fixed region centered about this bin are then included.

The width of the region of inclusion is dependent upon the physical properties of the sensor patch, rather than the histogram statistics. Each sensor region

has an allowable angular deviation of the lane markings and an acceptable error or noise margin. The far sensor region is readily influenced by road curvature (in the horizontal plane) and has few pixels per row, thus providing measurements with greater deviations and less accuracy than the corresponding near field sensor measurements. The near field sensor, however, displays nearly zero angular deviation (assuming markings are parallel to the vehicle) and a large number of pixels in each row help minimize noise levels.

Simple linear regression is used to fit the set of validated points from each sensor region to a line. Measurements whose residuals fall outside a preset value are discarded and the regression is repeated. If the correlation term of the second fit is acceptable, all remaining points are considered valid lateral displacement measurements. If at any time during the association the number of validated points falls below three, the sensor is considered to have no data for that sample.

6. DATA PROCESSING RESULTS

To illustrate the neural network data processing procedure, three example calculations are presented below. The data used in each example for training the network and deriving the steering control representations were obtained from the three separate data sources described above: 1) use of a vehicle dynamics numerical simulation to generate the driver/vehicle time history responses (and corresponding preview sensor data), 2) use of a desktop driving simulator with a human subject during a path regulation driving task and the recording of those system responses, and 3) use of the full-scale test vehicle and on-board video processing system described above in Section 5. Each of these data sources are processed in an analogous manner for these example calculations using the neural network architecture identified in Figure 4. Inputs to the neural network are the previewed sensor time histories resulting from a particular driving maneuver. The target response used in the supervised training was the driver steering time history corresponding to the same driving maneuver.

Numerical Simulation Example

In this first example, a fully nonlinear, closed-loop, driver/vehicle simulation [33] was used to generate the appropriate time history outputs used in the neural network processing. The vehicle model was augmented to provide three preview sensor "measurements" identical to those described in Figure 1. The near field measurement provided a sensor output at a point 0.1 seconds ahead of the vehicle mass center; the intermediate sensor measurement was 0.4 seconds ahead; the far field measurement was 0.7 seconds ahead. These three time history signals and their *time-delayed* counterparts described in Section 4, were used as a set of nine sensor inputs to the network architecture

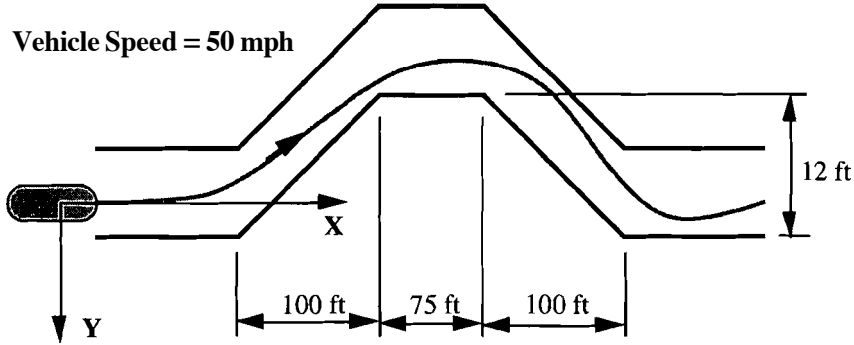


Fig. 7. Closed-loop, double lane change maneuver used in simulation example.

seen in Figure 4. The corresponding driver steering wheel time history, also generated by the model, was used as the target time history during the supervised network training calculations described subsequently. Two simulated driving maneuvers are used in this first numerical simulation example: 1) a closed-loop, double lane change maneuver conducted at 50 mph (see Figure 7) lying within the linear operating regime of the vehicle, and 2) a high lateral acceleration maneuver through an S-shaped curve (see Figure 8) fully exercising the nonlinear characteristics of the driver/vehicle simulation.

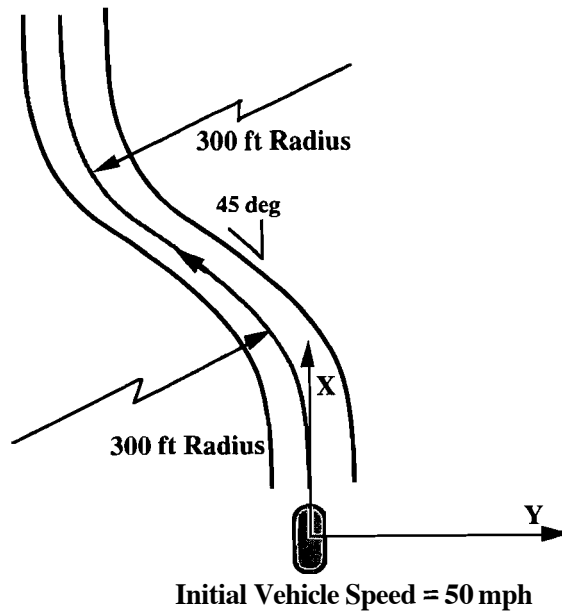


Fig. 8. S-curve geometry / maneuver used in simulation example.

The simulated vehicle weighs 2900 lb with a 55/45 fore-aft weight distribution. Conventional nonlinear tire and suspension characteristics were represented within the computer model. The normalized tire/surface friction adhesion limit was assumed to be 0.8 for the indicated computer runs. Outputs

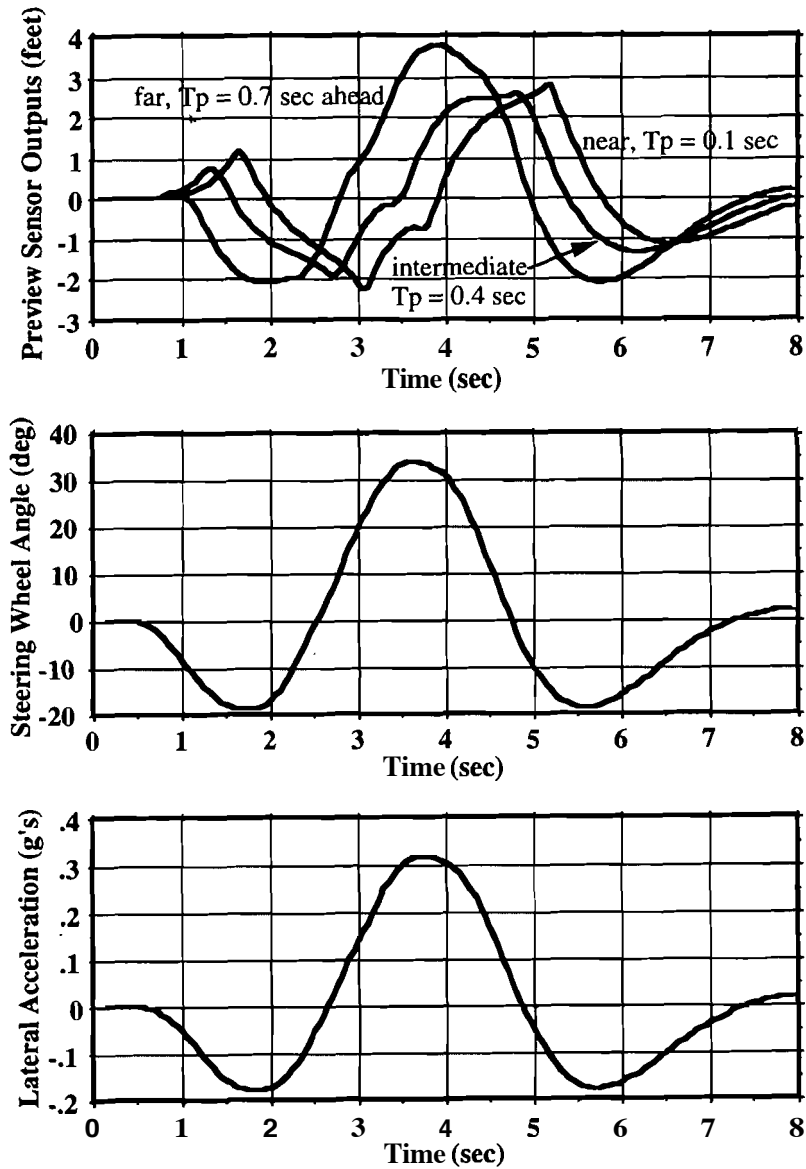
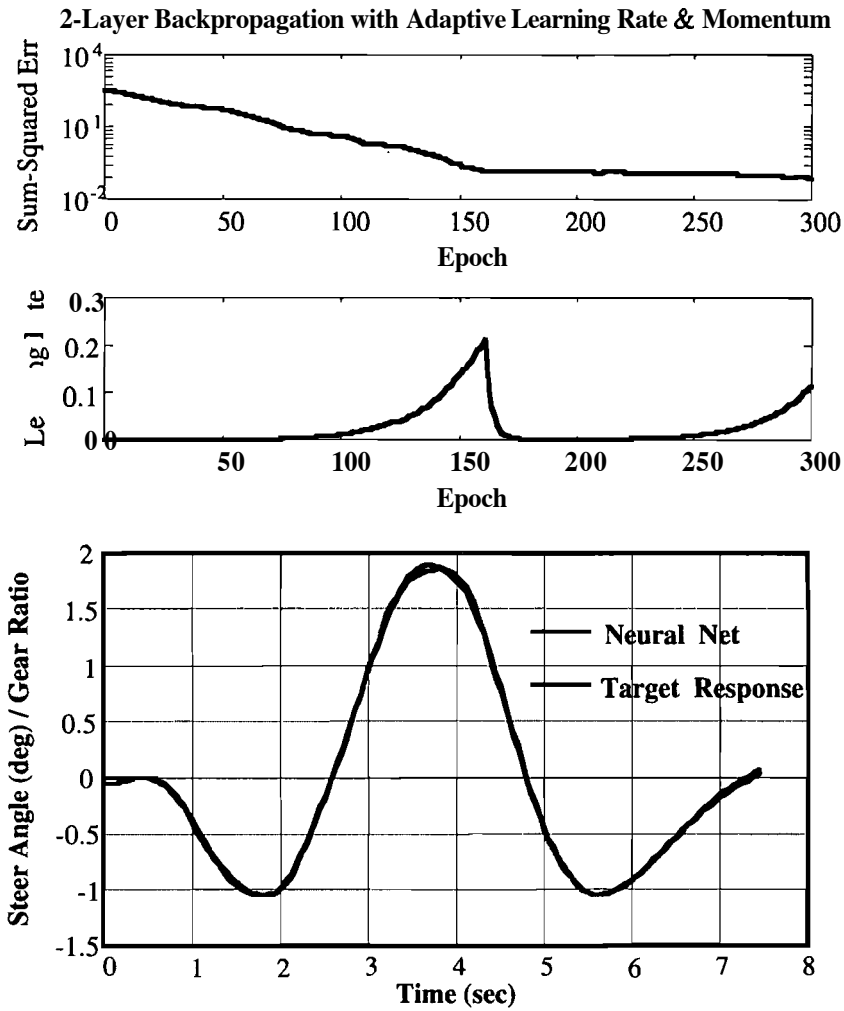


Fig. 9. Driver/vehicle simulation responses from the double lane change maneuver.



First Layer Network Weights —

W11 -> W91: -0.614 0.378 -0.018 -0.811 -0.874 -0.464 1.050 -0.924 -0.107
 W12 -> W92: -0.710 0.674 0.229 -0.105 -0.592 0.045 0.319 -0.043 0.760
 W13 -> W93: 0.287 -0.031 -0.833 0.570 -0.730 0.073 0.214 -0.202 0.326
 C1 -> C3: 0.415 -0.429 0.924

Second Layer Network Weights —

W11 -> W31: -0.899 1.554 -0.775 C1: 0.431

Fig. 10. Neural network training results for the double lane change maneuver

from each of these maneuvers is presented below, along with the resulting neural network processing of these numerical simulation results.

Double Lane Change Maneuver –

Simulation results from the double lane change maneuver described in Figure 7 are seen below in Figure 9. The first graph in Figure 9 shows the time history outputs from the simulated preview sensor as the driver/vehicle system maneuvers through the double lane change course.

The three separate lines correspond to the three preview points ahead (in time) of the vehicle mass center ($T_p = 0.1, 0.4, 0.7$ seconds ahead). The second and third graphs of Figure 9 show the closed-loop driver steering response and the vehicle lateral acceleration (mass center) recorded for the maneuver. The basic driver/vehicle response seen here is fairly representative of this general category of handling maneuvers.

These recordings were then used as inputs to the neural network architecture of Figure 4. The three preview sensor outputs seen in Figure 9 and their time-delayed replicates, each delayed by 0.3 and 0.6 seconds respectively, provided the nine sensor inputs seen in Figure 4. The driver steering response seen in Figure 9 was used as the target training time history at the output port of the neural network. These ten time histories were then presented to the network repeatedly (each total time history presentation is referred to as an "epoch" here) until a suitable matching between the training time history and the network output was obtained. These calculations were performed with commercial software [31] designed for such neural network analyses.

The results of the neural network training procedure are seen in Figure 10. The top graph of Figure 10 shows the sum-squared error at the output port (error between the desired target time history and the neural network output) during the training procedure.

The second graph shows how the learning rate used in the neural network algorithms varied during the training procedure. (The basic training algorithm was "back-propagation with an adaptive learning rate and momentum" [31]) The bottom graph shows the match achieved between the target response (driver steering response normalized by the steering system gear ratio) and the neural network output after 300 training epochs. Numerical values of the corresponding network weights (defined previously in Figure 4) are listed at the bottom of Figure 10. As seen in the graph, a rather good match between the neural network output and the target steering response is obtained. Examination of the sum-squared error graph of Figure 10 also suggests that nearly the same degree of match could be obtained with approximately half as much training.

With this result, the simulated driver steering behaviour in this particular maneuver could then be accurately represented and reproduced with the neural network structure presented in Figure 4, using the network weight values listed in Figure 10.

NEURAL NETWORKS AND PREVIEW SENSORS

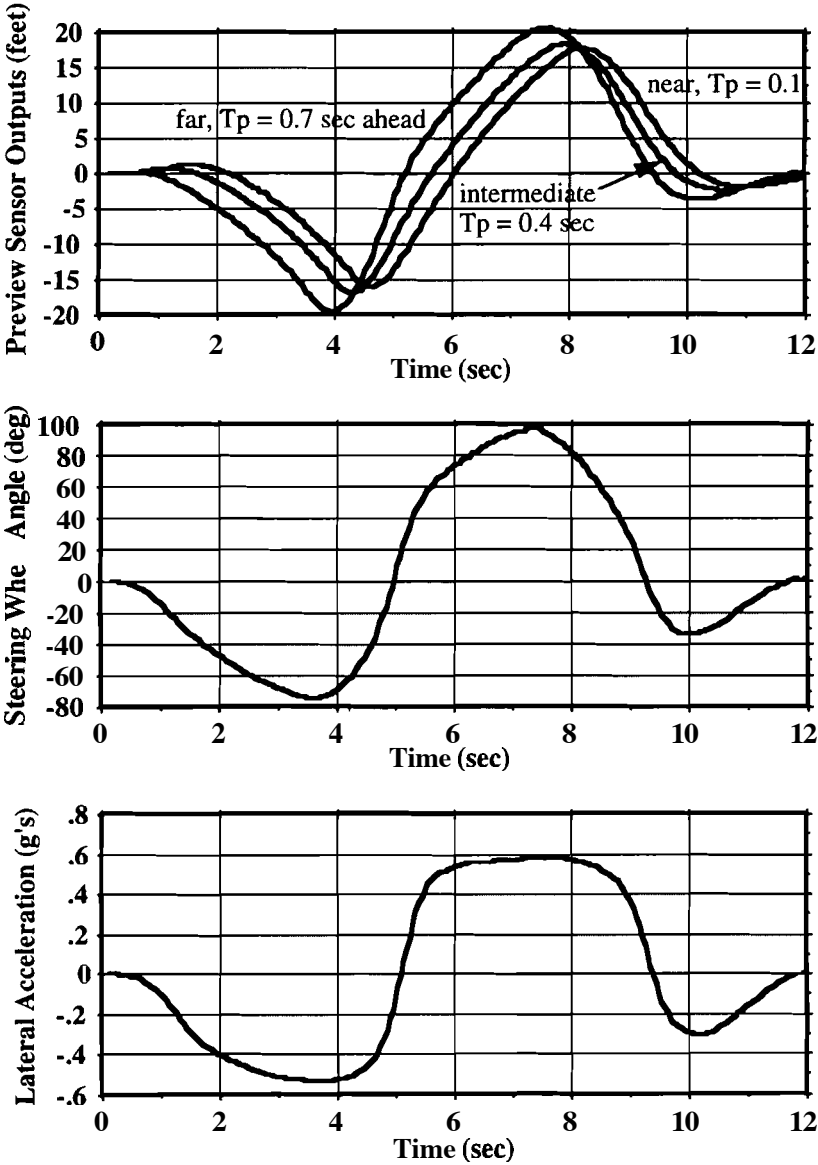
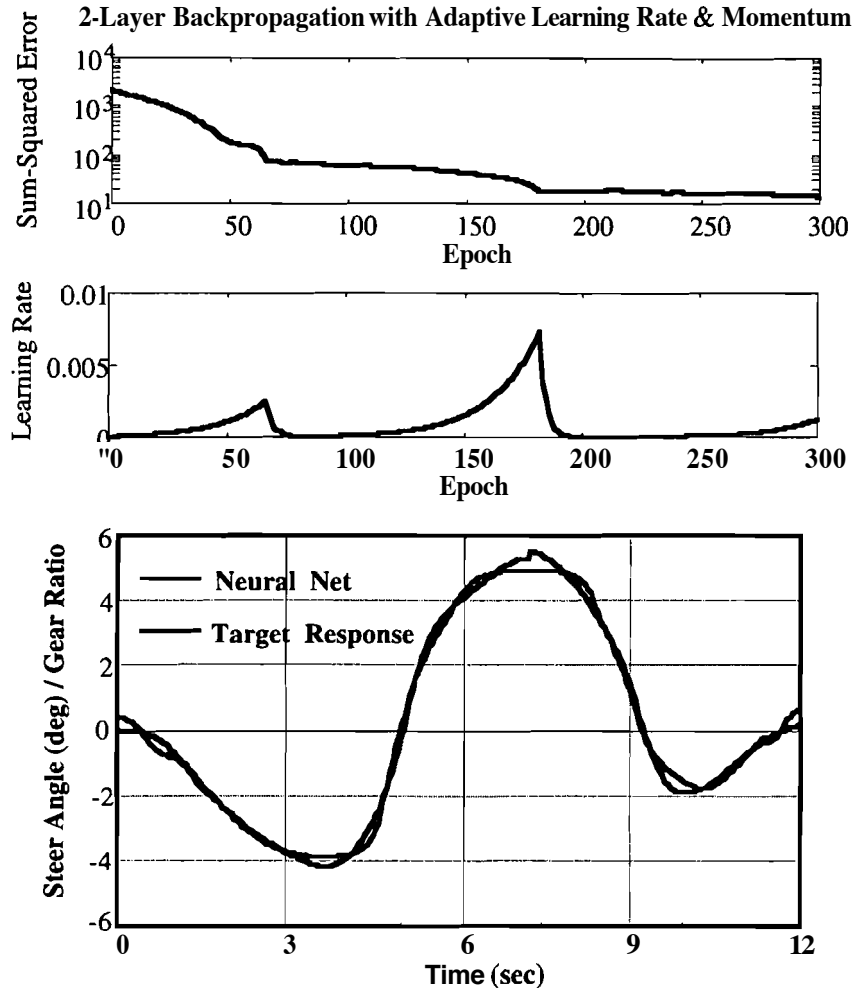


Fig. 11. Driver/vehicle simulation responses from the s-curve maneuver.



First Layer Network Weights —

W11 -> W91: -0.439 0.648 0.310 -0.540 -0.716 -0.219 1.159 -0.613 0.108

W12 -> W92: -0.545 0.819 0.206 -0.060 -0.637 -0.157 0.366 -0.221 0.429

W13 -> W93: 0.222 -0.205 -0.825 0.436 -0.708 0.154 0.213 -0.033 0.231

C1 -> C3: -0.438 -0.655 0.298

Second Layer Network Weights —

W11 -> W31: -3.099 3.514 -2.626 C1: 1.521

Fig. 12. Neural network training results for the s-curve maneuver.

S-Curve Maneuver –

To examine what might happen if the maneuvering regime for the same simulated **driver/vehicle** system is extended to higher lateral accelerations and the recorded time histories reprocessed, the S-Curve maneuver defined in Figure 8 was employed. The road course seen in Figure 8 was now driven by the same simulated **driver/vehicle** system at an average speed of 50 mph. The time history responses, analogous to those seen in Figure 9 for the double lane change maneuver, are shown here in Figure 11.

For this maneuver, the nonlinear **driver/vehicle** system is fully exercised as it negotiates the S-Curve path. Tire adhesion limits are being reached as evidenced by the saturating levels of the lateral acceleration time history (approximately 0.6 g's seen in the bottom graph). The vehicle is sliding hard through both turns and is unable to maintain accurate lateral lane position as the **tire/road** friction limits are reached. The **driver/vehicle** system remains directionally stable but is incurring large path errors as it proceeds through the turns of the S-Curve maneuver.

The resulting time history sensor outputs and the corresponding driver steering response taken from Figure 11 were then used in the same neural network training procedure as before. These results are seen in Figure 12.

Again, a reasonably good match is obtained between the desired target steering response and the network output. Although the degree of fit is poorer than that seen in the previous double lane change maneuver example (by comparing the sum-squared mathematical measures), the fit is still well within a usable tolerance range for many practical applications requiring approximations or estimates of driver steering behaviour under such conditions. These types of simulation results also suggest a potential for on-board vehicle technologies to perform such calculations "on-the-fly" to mathematically represent driver steering behaviour in real time. Such technologies could update changes in driver characterizations, and in general, track the adaptive and changing control behaviour exhibited by drivers over time as in [1,2,4].

Driving Simulator Example

The next example employs a desktop driving simulator [32] equipped with a steering torque motor and steering system model to generate similar **driver/vehicle** responses for a straight-line driving scenario. The vehicle dynamics model of the simulator contains three degrees of freedom (lateral, yaw, and roll motions) in addition to the steering system. The example data seen below in Figure 13 correspond to a driving scenario in which a test subject is initially offset laterally from a designated lane position (approximately 7 feet), and the subject then attempts to null the lateral offset through corrective steering.

The top graph of Figure 13 shows the three simulated preview sensor outputs analogous to those described previously in Figures 1, 9, and 11. Following a delay of approximately 2 seconds, the subject steers the simulator vehicle to the right to null out the initial path offset. The corresponding steering response

recorded for this maneuver is seen in the second graph of Figure 13. The lateral acceleration time history is shown in the bottom graph.

The sensor time histories and recorded driver steering response were then input to the same neural network architecture as before and processed in an

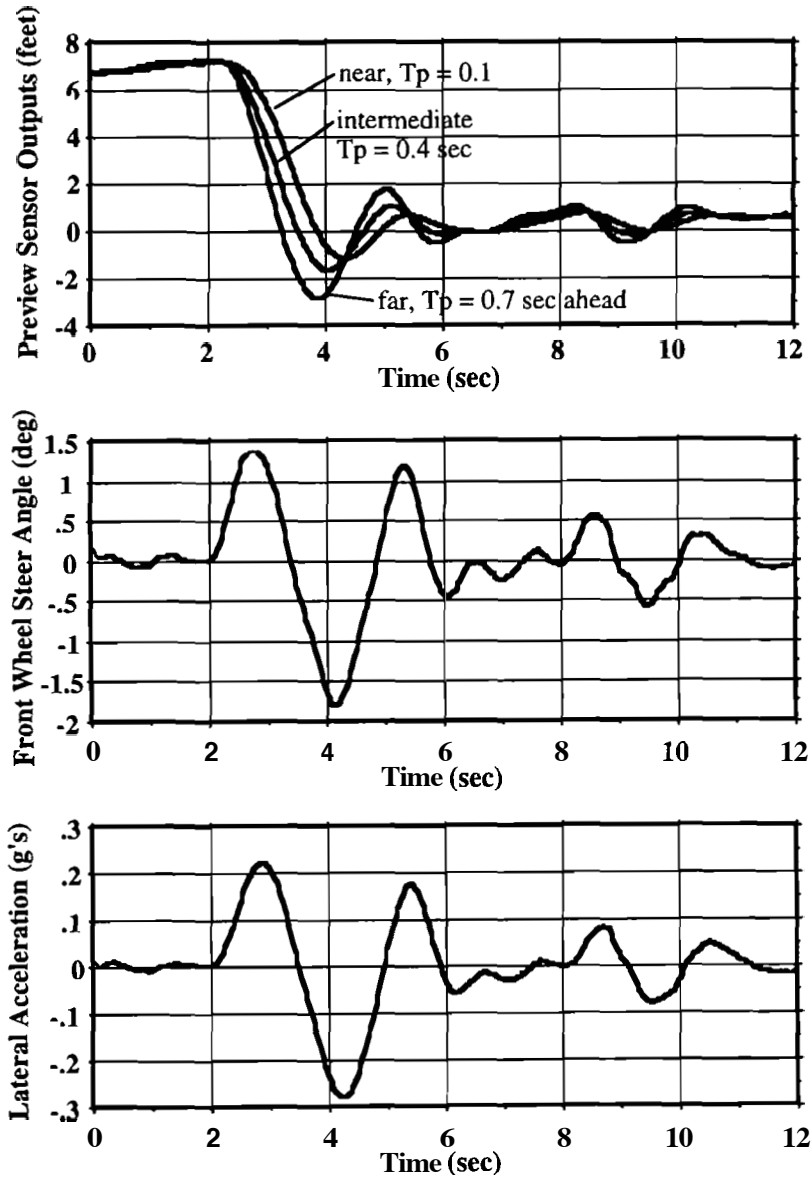
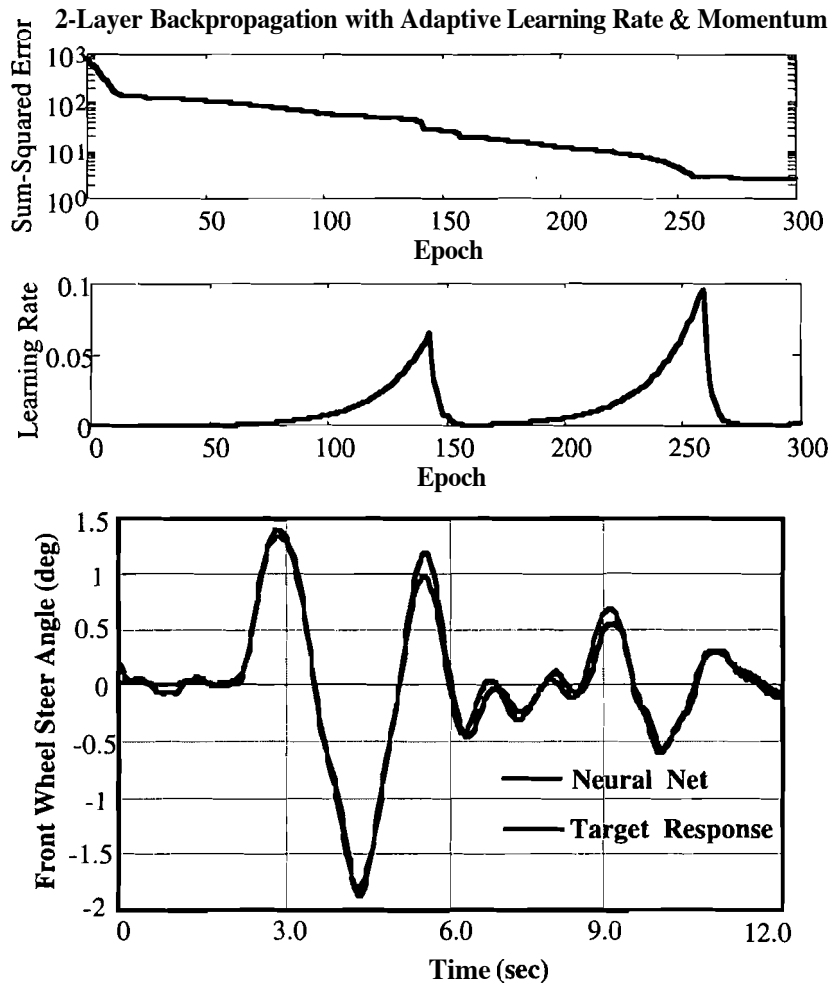


Fig. 13. Driver/vehicle responses from the driving simulator example.



First Layer Network Weights —

W11 -> W91: -1.396 0.068 0.293 0.964 0.018 -0.017 0.294 -1.133 0.334
 W12 -> W92: -1.520 0.683 0.904 0.232 -0.122 0.421 0.625 0.256 0.783
 W13 -> W93: -0.474 -0.466 -0.565 1.130 -0.378 0.365 0.225 -0.020 0.648

C1 -> C3: -0.832 -0.748 -0.352

Second Layer Network Weights —

W11 -> W31: -3.070 -0.176 -1.455 C1: 1.608

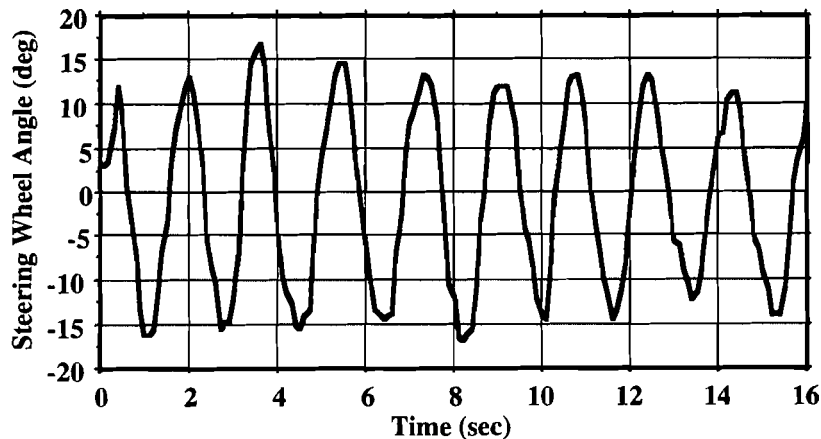
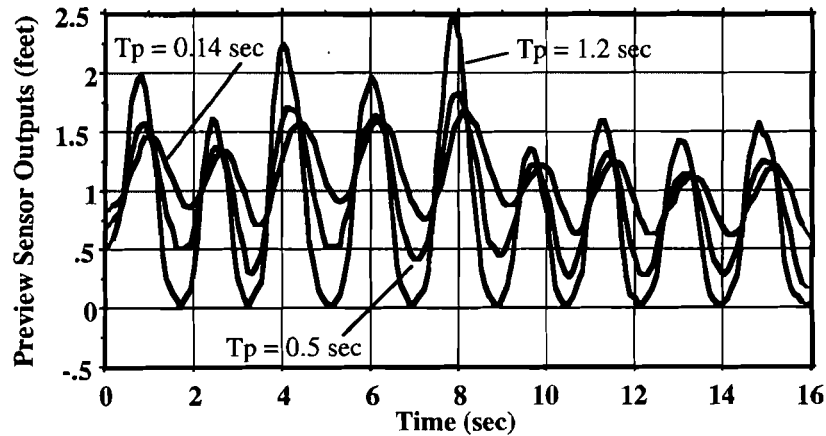
Fig. 14. Neural network training results for the driving simulator example.

identical manner. The results of these neural network training calculations are seen in Figure 14.

Note that the achieved matching between the driving simulator subject response and the neural network representation is quite good, including small steering peculiarities normally present in experimental human response data of this kind. It is evident that the neural network exhibits an ability to not only represent the more idealized driver steering behaviour present in a numerical simulation environment (as in the previous example), but is also capable of accurately representing the smaller undulations and other human oddities present in more realistic driver steering waveforms seen in this example.

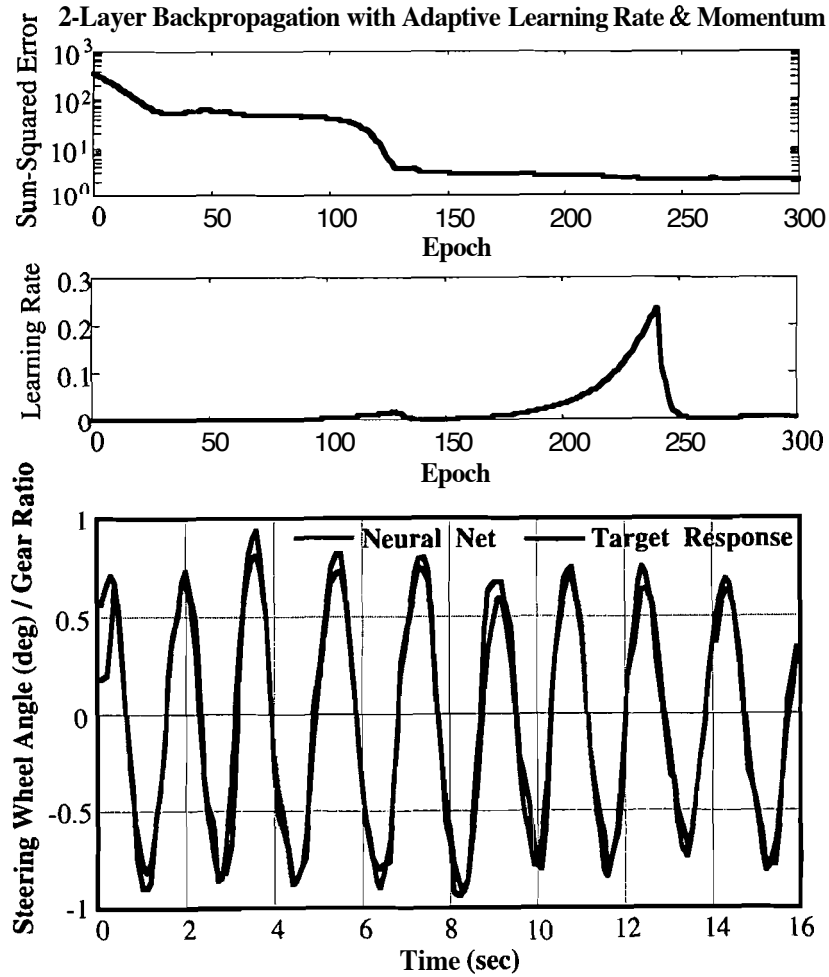
On-Road Test Vehicle Example

This last example uses data gathered by the on-road test vehicle and



Driver/vehicle responses from the on-road test vehicle example.

NEURAL NETWORKS AND PREVIEW SENSORS



First Layer Network Weights —

W11 -> W91: 0.065 0.889 0.385 -1.003 -1.202 -0.133 0.629 -0.932 0.320
 W12 -> W92: -0.755 1.012 0.786 0.023 -0.226 0.442 0.584 0.268 0.836
 W13 -> W93: -0.781 -0.910 -1.140 0.549 -0.481 0.314 0.802 0.326 0.912

C1 -> C3: -0.406 -0.855 0.584

Second Layer Network Weights —

W11 -> W31: -1.099 -0.234 1.363 C1: -0.234

Fig. 16. Neural network training results for the on-road test vehicle example.

data collection system described in Section 5. Aside from the on-board instrumentation system, consisting of the lane-sensing camera and computer processing equipment, the vehicle is a conventional passenger vehicle (station wagon) weighing approximately 3500 lb in its tested state. The image data was collected and processed in real time to produce the three preview lane sensor measurements described previously in Section 5. Processing and storing of these data were performed at a rate of 10 Hz. The lane sensor and driver steering response information was stored to disk as time histories for subsequent processing by the neural network analyses.

An example test run collected by the system is seen in Figure 15 for an average vehicle speed of 32 mph. The test driver is regulating the vehicle along a straight road section using a mild oscillatory steering input to produce the observed response.

As before, three preview sensor measurements are shown in the top graph. The three sensor measurements seen here were obtained directly from the on-board processing system by averaging the individual raster line measurements within each of the three sensor regions (described previously in Figure 6). The center of the far field measurement lies approximately 60 feet ahead of the vehicle mass center (1.2 seconds ahead); the intermediate measurement is at an average distance of 25 feet ahead (0.5 seconds); the near field measurement is located approximately 7 feet ahead (0.14 seconds). The measured driver steering response is seen in the bottom graph.

These recorded measurements were then applied as inputs to the same neural network structure as used previously and processed in an identical manner. Again, the three sensor measurements were each delayed in time by 0.3 and 0.6 seconds respectively to provide a total of nine sensor inputs to the network architecture of Figure 4. The neural network training calculations and the obtained results are shown in Figure 16.

As seen, a good match is achieved between the measured target driver steering response and that calculated by the neural network after 300 epochs. The corresponding network weights identified for these data are shown at the bottom of Figure 16. Thus, use of the neural network structure proposed in Figure 4, using the weight assignments listed in Figure 16, could likewise be used to model or represent the driver steering response seen in this example.

Lastly, it should be noted that use of time-delayed sensor inputs was especially helpful in improving the network training times and accuracies for both the on-road test data example seen here and for the previous driving simulator example.

7. CONCLUSIONS

This paper has demonstrated the basic use of elementary neural networks for modelling and representing driver steering behaviour in path regulation control tasks. Areas of application include uses by vehicle simulation experts who need

to model and represent specific instances of driver steering control behaviour, potential on-board vehicle technologies aimed at representing and tracking driver steering control behaviour over time, and use by human factors specialists interested in representing or classifying specific families of driver steering behaviour. Extension of this and related work to other technical realms is clearly possible, as is the application of this specific work to other vehicle control areas such as longitudinal headway control and collision avoidance systems.

A key element in this paper was the use of time delayed sensor information in order to permit the neural network to derive necessary time derivative information in its formulations of driver steering control. This technical detail, though not especially obvious, is seen as an important reason for much of the indicated success presented here. Use of time-delayed sensor information was observed to improve training time and accuracy by the neural network representations, particularly when utilizing human steering response measurements obtained from the driving simulator and test vehicle sources.

Finally, it should be noted that the results seen here apply to the very specific operating conditions of the example cases. However, the variety of driving scenarios and data sources considered in the above examples would suggest that the adaptive nature of neural networks can be used effectively for representing driver steering control behaviour under a wide range of operating conditions. This could also apply to temporal variations in driver control behaviour induced as a result of changing driving conditions or driver properties. Further research into issues of practical implementation of neural network concepts and adaptive control is anticipated in future work.

REFERENCES

1. **Aono**, S.: Next step in automotive electronic control. Proceedings, 39th **IEEE** Vehicular Technology Conference: Gateway to New Concepts in Vehicular Technology, 1989.
2. **Kopf**, M. and **Onken**, R.: Monitoring and Warning System for Driver Support on German Autobahn. Proceedings, ISATA International Symposium, Florence, Italy, INRETS, 1991.
3. **Kraiss**, K.F. and **Kuettelwesch**, H.: Teaching neural networks to guide a vehicle through an obstacle course by emulating a human teacher. Proceedings, International Joint Conference on Neural Networks - IJCNN 90, 1990.
4. **Kraiss**, K.F. and **Kuettelwesch**, H.: Identification and application of neural operator models in a car driving situation. Proceedings, 5th **IFAC/IFIP/IFORS/IEA** Symposium on Analysis, Design and Evaluation of Man-Machine Systems, 1992.
5. **Muller**, R. and **Nocker**, G.: Intelligent Cruise Control with Fuzzy Logic. Proceedings, Intelligent Vehicles '92 Symposium, Detroit, IEEE Industrial Electronics Society, 1992.
6. **McMahon**, D. et al: Longitudinal Vehicle Controllers for IVHS: Theory and Practice. Proceedings, American Control Conference, Chicago, 1992.
7. **Neusser**, S., **Hoefflinger**, B. and **Nijhuis**, J.: A Case Study in Car Control by Neural Networks. Proceedings, **ISATA** International Symposium, Florence, Italy, INRETS, 1991.
8. **Fix**, E.: Neural network based human performance modeling. Proceedings, IEEE 1990 National Aerospace and Electronics Conference - NAECON, 1990.
9. **Neusser**, S. et al.: Neurocontrol for Lateral Vehicle Guidance. *IEEE Micro*, Vol. 13, No. 1, 1993, pp. 57-66.

10. Pomerleau, D.: Efficient Training of Artificial Neural Networks for Autonomous Navigation. *Neural Computation*, Vol. 3, No. 1, 1991.
11. Pomerleau, D.: Progress in Neural Network-Based Vision for Autonomous Robot Driving. Proceedings, Intelligent Vehicles '92 Symposium, Detroit, IEEE Industrial Electronics Society, 1992.
12. Schiller, I. and Tench, K.A.: A neural-network-based autonomous underwater vehicle guidance system. Proceedings, Sixth International Symposium on Unmanned Untethered Submersible Technology, 1989.
13. Shepanski, J.F. and Macy, S.A.: Manual Training Techniques of Autonomous Systems Based on Artificial Neural Networks. Proceedings, IEEE First International Conference on Neural Networks, 1987.
14. Hancock, P. and Parasuraman, R.: Human Factors and Safety in the Design of Intelligent Vehicle-Highway Systems (IVHS). *J. of Safety Research*, Vol. 23, No. 4, (1992). pp. 181-198.
15. Freeman, J. and Skapura, D.: Neural Networks: algorithms, applications, and programming techniques. Addison-Wesley, 1991.
16. Hecht-Nielsen, R.: Neurocomputing. Addison-Wesley, 1989.
17. Wasserman, P.: Neural Computing: Theory and Practice. Van Nostrand Reinhold, 1989.
18. Brown, M. and Harris, C.J.: Intelligent control for autonomous vehicles using real-time adaptive associative memory neural networks. Proceedings, 2nd International Conference on Artificial Neural Networks, 1991.
19. Luebbers, P.G. and Pandya, A.S.: Video-image-based neural network guidance system with adaptive view-angles for autonomous vehicles. Proceedings, Applications of Artificial Neural Networks II, 1991.
20. Cheng, R.M.H., Xiao, J.W. and LeQuoc, S.: Neuromorphic controller for AGV steering. Proceedings, IEEE International Conference on Robotics and Automation, 1992.
21. Fujioka, T. and Takubo, N.: Driver Model Obtained by Neural Network System. *JSAE Review*, Vol. 12, No. 2, 1991. pp. 82-85.
22. Kehtarnavaz, N. and Sohn, W.: Steering control of autonomous vehicles by neural networks. Proceedings, American Control Conference, 1991.
23. Kornhauser, A.L.: Neural network approaches for lateral control of autonomous highway vehicles. Proceedings, Vehicle Navigation & Information Systems, 1991.
24. Lubin, J.M. et al: Analysis of a neural network lateral controller for an autonomous road vehicle. Proceedings, Future Transportation Technology Conference and Exposition, 1992.
25. Yasui, Y. and Margolis, D.: Lateral Control of Automobiles Using a Look-Ahead Sensor. Proceedings, International Symposium on Advanced Vehicle Control, Tokyo, Japan, SAE of Japan, 1992.
26. McRuer, D.T. et al: New Approaches to Human-Pilot/Vehicle Dynamic Analysis. Tech. Report AFFDL-TR-67-150, WPAFB, 1968.
27. McRuer, D.T., Allen, R.W. and Klein, R.H.: New Results in Driver Steering Control Models. *Human Factors*, Vol. 19, No. 4, 1977. pp. 381-397.
28. Weir, D., DiMarco, R.J. and McRuer, D.T.: Evaluation and Correlation of Driver/Vehicle Data, Vol II. Final Technical Report, NHTSA, DOT-HS-803-246, 1977.
29. MacAdam, C.C.: Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. June, 1981. pp. 393-399.
30. MacAdam, C.C.: Development of Driver/Vehicle Steering Interaction Models for Dynamic Analysis. Final Technical Report, UMTRI-88-53. TACOM Contract DAAE07-85-C-R069, 1988.
31. MathWorks: Neural Network Toolbox User's Guide / MATLAB. 1992.
32. MacAdam, C., Green, P. and Reed, M.: An Overview of Current UMTRI Driving Simulators. *UMTRI Research Review*, Vol. 24, No. 1, 1993.
33. MacAdam, C.C. et al: A Computerized Model for Simulating the Braking and Steering Dynamics of Trucks, Tractor-Semitrailers, Doubles, and Triples Combinations. User's Manual - Phase 4, UM-HSRI-80-58, 1980.
34. Sweet, R. and Green, P.: UMTRI's Instrumented Car. *UMTRI Research Review*, Vol. 23, No. 4, 1993.
35. Blake, A. and Yuille, A.: Active Vision. MIT press, Cambridge, 1992.