Int. J. Vehicle Design, Vol. 0, Nos. 0/0, 0000 1

Artificial neural network modelling of driver handling behaviour in a driver–vehicle–environment system

Please supply e-mail addresses for co-authors, if possible.

Y. Lin

Concordia University, Concordia Institute for Information Systems Engineering, 1455 de Maisonneuve Blvd. West, Montreal, Quebec, H3G 1M8 Canada

P. Tang

The Pennsylvania State University, College of Engineering, University Park, PA 16802 USA

Please check fax and e-mail details are correct for W.J. Zhang, these were obtained from the internet.

W.J. Zhang*

The University of Saskatchewan, Advanced Engineering Design Laboratory, Saskatoon SK, S7N 5A9 Canada
Fax: (306) 966-5427 E-mail: Chris_Zhang@engr.usask.ca
*Corresponding author

Q. Yu

China Agricultural University, College of Vehicle Engineering, Beijing 100083, P.R. China

Abstract: Modelling driver handling behaviour in a driver–vehicle–environment (DVE) system is essentially useful for the design of vehicle systems and transport systems in the light of the safety and efficiency of human mobility. Driver handling behaviour is reflected in two aspects: the mental workload and the performance. Further, this behaviour is exposed through the interactions between driver–vehicle and driver–environment. There is generally a lack of the first principle with which to develop a model for human behaviour. In this study, several more sophisticated artificial neural network architectures for developing models for human drivers in a DVE system were used. The vehicle dynamics are modelled by a 3-d.o.f. model derived from the first principle. The experiment was performed and compared with a DVE simulation system in which the developed human driver behaviour model was included, together with the vehicle dynamics model. The comparative study showed that the simulation result is in good agreement with the experimental result, which further justifies the effectiveness of the developed driver behaviour model.

Keywords: driver behaviour modelling; driver–vehicle–environment system; neural networks.

Reference to this paper should be made as follows: Lin, Y., Tang, P., Zhang, W.J. and Yu, Q. (0000) 'Artificial neural network modelling of driver handling behaviour in a driver–vehicle–environment system', *Int. J. Vehicle Design*, Vol. 00, Nos. 0/0, pp.000–000.

Did Dr. Lin receive two PhD degrees or should one be an MSc?

Biographical notes: Dr Y. Lin is currently an Assistant Professor at Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada. She received her PhD in mechanical engineering, from the University of Saskatchewan, in 2003. She also received a PhD in vehicle engineering from China Agricultural University in 1997, and a BSc in mechanical and manufacturing engineering from Beijing Agricultural Engineering University in 1992. She was an Assistant Professor at the Department of Vehicle Engineering at China Agricultural University during 1997–1998.

Dr P. Tang is currently a Research Associate at the College of Engineering, Pennsylvania State University, USA. He received his PhD from the Pennsylvania State University in 2004, his MSc from the University of Manitaba, Winnipeg, Canada in 2000, and his BSc in mechanical and manufacturing engineering, from Beijing Agricultural Engineering University in 1992.

Dr W.J. Zhang is currently a Professor at the Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Canada. He received his PhD in mechanical engineering from the Technical University of Delft in 1994. He was an Assistant Professor at the Department of Manufacturing Engineering City University of Hong Kong during 1994–1998 and an Associate Professor and Industrial Research Chair sponsored by AECL during 1998–2003.

Q. Yu is a Professor Emeritus at the Department of Vehicle Engineering, China Agricultural University, Beijing, China.

1 Introduction

1.1 Motivation

Vehicles play an important role in the development of the world economy and in society. However, vehicles also cause problems, for example, transport congestion, environmental pollution, etc. Perhaps the most serious problem among them is accidents. There is a great need to develop accident reduction methods. Three elements, namely the driver, the vehicle and the environment are associated with accidents, and they cannot be viewed separately. The system which consists of these three elements is called the driver–vehicle–environment (DVE) system. In the DVE system, the driver plays four roles: supervising, controlling, actuating and sensing. Any defect or malfunction with any of these roles can lead to errors.

It has been shown that most accidents were attributed to driver error (Ervin and Guy, 1986; LeBlanc and Ervin, 1996). According to commonsense, human errors in manipulation can be attributed to two causes: inappropriate mental workloads and below-desired manipulation skills; these two causes could be coupled. It is, therefore, very important to understand and model the driver mental workload (Tang and Mann, 2003) and their handling capability or performance (Colburn, 1978; Wilde, 1988). The study reported in this paper was concerned with the development of such a model and investigation of the applicability of the model in a DVE simulation system (Lin and Yu, 1996). It is noted that such a simulation system can be used in vehicle design/maintenance, highway transport system design, and prediction and remedy of a driver's workplace injuries.

1.2 Literature review

Fenton applied classical control (1971) and a Linear Quadratic algorithm (1988) to design a controller. Ackermann and Sienel (1990) used Parameter Space Robust Control to design the automatic steering controller. These models did not take the driver's preview behaviour into consideration. They were compensatory models which relied on the anti-disturbance capability of the feedback controllers for tracking the trajectories. Lee (1989) developed a discrete time preview control algorithm for four-wheel steering passenger vehicles and found that the control accuracy was improved substantially by taking into account preview behaviour. MacAdam (1980) developed an optimal preview control algorithm and successfully applied this algorithm to lane tracking and lane changing tasks. However, this algorithm could only be applied to single input–output systems (MacAdam, 1981).

PATH (Program on Advanced Technology for the Highway) created four algorithms for vehicle lateral control (Shladover and Desoer, 1991): Finite Time Preview Control; Frequency Shaped Linear Quadratic (Peng, 1994; Peng and Tomizuka, 1991); Fuzzy-based Control (Hessburg and Tomizuka, 1993); and Feedback Linearisation and Surface Control (Pham et al., 1994). These models were based on an explicit vehicle dynamic model and limited to linear systems. In addition, a hybrid driver model was developed combining the discrete event control theory and classical control theory (Kiencke et al., 1999). In their model, use of the queue theory for the sensory process of drivers eliminated a concurrent perception of information which is often the case in the human perception process.

In general, the modelling of a complex system (e.g. human-machine system, social-technical system) has two basic starting points. One starting point is first principle or principles which fundamentally govern the behaviour of a system under interest. The other starting point is extensional behaviour of a system in a form of the stimuli-response pair. The product of any modelling, either based on the first starting point or the second starting point, is a representation clustering a set of variables and parameters. A proper combination of these two approaches should lead to a more efficient and effective model (Zhang, 2003). The models previously reviewed are clearly in the category of methods with the first starting point. Due to the lack of well understood first principles on human drivers, these models have not worked well. The following are, basically, the models with the second starting point.

The fuzzy logic and neural network theory were introduced to model human driver's behaviours. Shepanski and Macy (1987) proposed two training methods, the Master/Apprentice training and direct network-based training in the DVE system based on the multi-layer perception model and δ training rate algorithm. Kornhauser (1991) developed a back propagation model and an adaptive resonance theory model, respectively. The results showed that the back propagation model resulted in a slow convergence rate. The adaptive resonance theory model was not successful because such a model puts a stringent condition which, translated to driver behaviour level, requires that different drivers should take the same steering decision for the same scenario. Neusser (1993) studied the driver's handling properties through a record of a 'perfect' driver's 50,000 steering behaviour. They used a three-layer feed forward network which includes 21 neurons. There were five inputs to the network, namely vehicle longitudinal velocity, vehicle heading angle, road curvature, road width and lateral deviation. It was demonstrated that the control accuracy was higher than the conventional controllers.

MacAdam and Johnson (1996) demonstrated the use of elementary neural networks (a two-layer back propagation with adaptive learning rate and momentum) to represent the driver steering behaviour in double lane change manoeuvre and S-curve manoeuvre. Due to the limited data source available for neural networks, it was concluded that the adaptive nature of neural networks should be used for representing driver steering behaviour under a variety of operation scenarios. An and Harris (1996) adopted a Cerebellar Model Articulation Controller (CMAC) in developing an adaptive driver model for collision avoidance in the case of longitudinal lane following. The previous throttle angle, the vehicle's speed, range and range rate to the front vehicle were chosen as the inputs of the model, while the current throttle angle was chosen as the output of the driver model. The simulation results showed that the chosen input and output parameters were relevant parameters for modelling the driver's behaviour. Based on the evaluation techniques of one-step-ahead prediction of error performances, the error deviation of the CMAC model was found to be acceptable for the test scenario compared with that of the Conventional Linear Model (CLM) under both test track and a motorway environment. However, both models had similar characteristics of error correlation which were consistently larger than those for the test tracking case. It was analysed that this phenomenon might be caused by factors such as tiredness and agitation of drivers.

A two-layer Tansig and Linear neural network was applied to the directional control behaviour in conjunction with a seven-axle tractor-semi-trailer subject to varying directional manoeuvres (Yang et al., 1998). The position, velocity and acceleration of the vehicle were used as the input of the driver model and the front wheel steer angle was considered as the desired output of the trained network. The training set for their neural network was extracted from a proposed driver tractor-semi-trailer model. The network was trained based upon the Levenberg–Marquardt techniques and the mean squared error was examined. The results showed that the network architecture could be trained, while the effectiveness of the trained network was strongly influenced by different schemes. It was concluded that the lateral position errors formed the most important input variable for training the network, followed by preview error, and the lateral acceleration of the centre of the gravity of the tractor.

Other kinds of driver behaviour in traffic regulation and control were also studied using a back propagation network, such as the driver's reaction to some specific signals, and the violation of red lights in urban environments (Mussone et al., 1995).

Kamada et al. (1992) proposed a fuzzy logic lateral controller. The inputs were formed by the angle between the vehicle's centre line and the sequence of the marked lines on the road, the offset and its rate of change of a point on the centreline at a fixed distance in front of the vehicle from the sequence of the marker lines, which were provided by a video camera and image processor. Each of these fuzzy variables was defined using five linguistic variables with triangular membership functions. The steering angle was defined as the output obtained by applying the gravity centre defuzzification method. The whole system hardware was implemented, but the control accuracy was not well validated due to the limited experimental trials. Hessburg and Tomizuka (1993) developed a fuzzy logic controller for vehicle lateral guidance which consisted of three sub-controllers: preview, feedback and gain scheduling. The fuzzy preview controller perceived the curvature for each curved section of the reference track and then computed the required steering angle. The feedback controller generated the feedback steering angle based on the discrepancy between the current and the desired state variables. A trade off was made between these two control signals, and the final steering angle was given by the gain scheduling controller. It was reported that a considerable fluctuation existed in their system.

1.3 Objectives and scope

The literature review shows a general trend in the study of modelling of human behaviours, i.e. towards the greater application of computational artificial intelligence approaches, such as the neural network technique and the fuzzy logic theory. Such a trend is sound because the human driver's mental and physical behaviour is non-deterministic and highly non-linear, and hence there are seldom any first principles available governing such behaviour; see also the discussion above. The complexity in predicting human driver behaviours is further coupled with the complexity in predicting vehicle dynamics (e.g. the friction between the tyre and the road, etc.), as drivers behave under the dynamic situation of the vehicle and the environment. However, the current literature leaves some room for a further study. For example, MacAdam and Johnson's work (1996) has some limitations in that

- they did not consider the velocity and the acceleration of the yaw angle in their neural network model
- the neural network model they used is elementary
- the way they considered for the driver's preview was through three sensors for measuring the offsets (the current time t, $t + \tau_1$, and $t + \tau_2$, where τ_1, τ_2 are specific delays defined empirically).

The work reported by Yang et al. (1998) considered a tractor and a trailer instead of a car. Owing to this difference, the inputs to their neural network model might be different from a car. Besides, in their model only lateral motion parameters were considered. The study reported by Atsumi et al. (1993) considered different driving situations as the variation of vehicle velocity only. In general, the reported studies in the literature usually employed relatively simple network architectures to their particular problems. It is relatively difficult to generalise about the results in terms of the neural network models they used.

The study reported in this paper was a continuing project on the study of developing a fuzzy logic controller for human driver handling behaviours in a DVE system (Gu and Yu, 1993). The primary goal of this study was the development of a neural network-based model for human driver handling behaviour in a DVE system. Consistent with this goal, the following objectives are rendered for this study:

- Objective 1: to employ and compare more advanced neural network architectures/models in the modelling of driver behaviour.
- Objective 2: to develop a driver–vehicle–environment simulation system, in which the driver handling behaviour model is a part of the system.

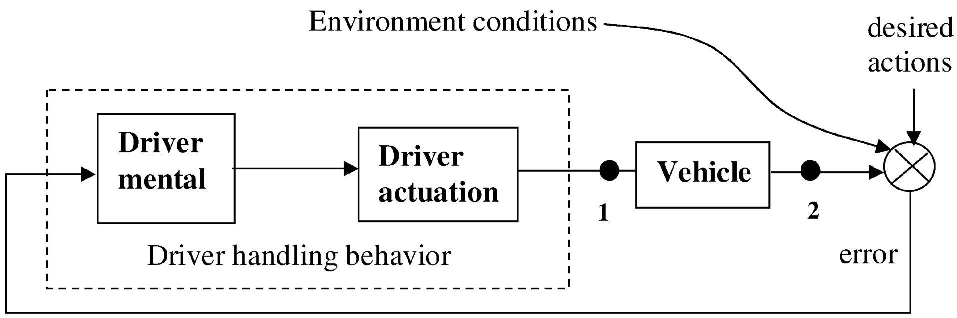
For Objective 1, different neural network architectures were employed for the same problem and compared to evaluate their suitability. For Objective 2, a test bed system was developed on which the effectiveness of the developed driver behaviour model was examined through a comparison of the simulation result (produced from this system) and the experimental result (conducted in this study). The next section presents a general architecture of a driver–vehicle–environment simulation system, where a context for a driver handling behaviour model can be explicitly made.

2 The DVE simulation system

Figure 1 shows a general description of a DVE system. The driver handling behaviour consists of both the driver mental behaviour and the driver physical action behaviour. The desired action is the action a driver takes to manipulate the vehicle to follow the described motions. In the present study, three general types of vehicle motions were considered (see Figure 2):

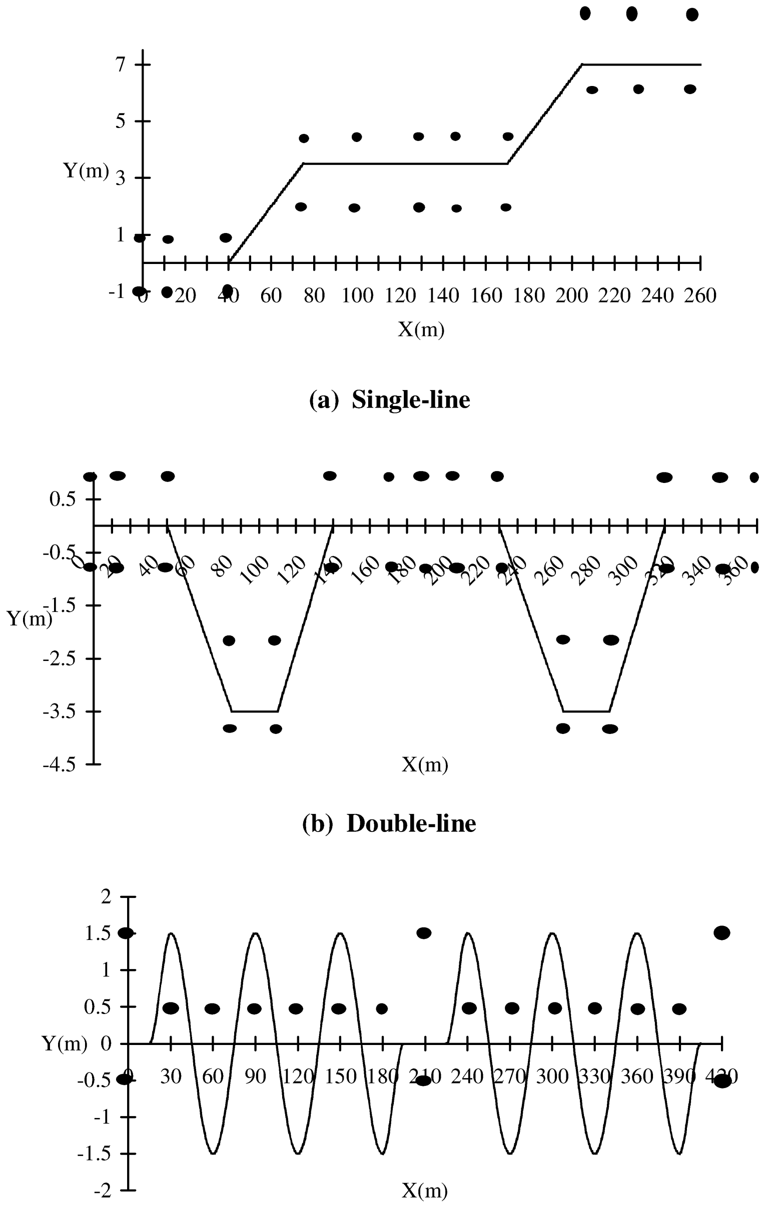
- the single-line motion which is the basic action for lane change
- the double-line motion which is the basic action for takeover
- the sine-line motion which is the basic action for S-curve turn.

Figure 1 DVE closed-loop system



Error: difference between the described actions and the actual actions;
1: the steering motion (e.g., angle);
2: the lateral and/or yaw motions.

Figure 2 Designed driving route



(a) Single-line

(b) Double-line

(c) Sine-line

(—— desired driving route; ● boundary of the route)

The environmental conditions are, generally, road conditions (e.g. traffic lights, cross sections, pedestrians, road conditions, etc.). The goal of a driver is to manipulate a vehicle to follow the desired motions under the environmental conditions per se. The general way the driver achieves this goal is to get the information about the vehicle dynamics and the error between the desired vehicle motion and the actual vehicle motion, to make his or her decisions, and to actuate the vehicle (see Figure 1). It is expected that drivers should have an adaptive control ability for the uncertainty created from the environment, the complexity from the vehicle dynamics and from the interaction between the driver and the vehicle and between the vehicle and the environment.

The computer simulation programme for the DVE system includes three modules: environment and task input module, vehicle dynamics module and driver controller module. At this phase of the study, the environment and task module includes a route library that contains single-line, double-line and sine-line routes and tasks (see Figure 2). The vehicle dynamic model includes a three degree-of-freedom (DOF) vehicle model, owing to a wide agreement that 3 d.o.f. is sufficient to represent vehicle dynamics when the lateral acceleration is smaller than 0.4 g (Smith and Starkey, 1995). In the present study, we aimed to develop a driver handling behaviour model. In this development, the different driving routes and dynamics of the vehicle were considered; however the interaction between the environment (e.g. traffic situation) and the driver was not considered.

The DVE system works as follows: The user is required to select a route and give an initial steering angle. Then, the 3-d.o.f. vehicle model calculates the values of the vehicle dynamic variables, namely, lateral acceleration, lateral velocity and yaw angle velocity. The fourth-order Lounge-Kutta method solves the 3-d.o.f. vehicle model. The lateral displacement is obtained based on the trapezoidal integral method. These variables are used as feedback signals to the driver handling model (see later discussion). The steering wheel angle can be obtained from the driver model, i.e. the neural network model. The updated steering wheel angle is fed to the 3-d.o.f. vehicle model, which triggers a new cycle of the process. The next section discusses neural network models for the DVE system.

3 Neural network model development

3.1 Inputs and outputs to the neural network

To develop a neural network model for the DVE system, the vehicle driving motion system is considered as a discrete time-variant system. Let the inputs and outputs to the neural network model be denoted by $INPUT(t_k)$ and $OUTPUT(t_k)$ at time t_k. The inputs are the information from the vehicle dynamics and the error. In particular, the inputs in the present study include:

- the yaw angular velocity $r(t_k)$
- the lateral velocity $\dot{Y}(t_k)$
- the lateral acceleration $\ddot{Y}(t_k)$
- the roll angle $\varphi(t_k)$
- the roll angle velocity $\dot{\varphi}(t_k)$
- the lateral displacement $Y(t_k)$
- the preview lateral offset $O(t_k)$.

The preview offset is widely recognised as effective information to drivers in making their decision about actions. In the present study, $O(t_k)$ was calculated by the following equation:

$$O(t_k) = f(t_k + T) - Y(t_k) - T\dot{Y}(t), \tag{1}$$

where T is the preview time, in the present study $T = 1.3\,\text{s}$.

Let $State(t_k)$ represent the inputs:

$$State(t_k) = \left\{r(t_k), \dot{Y}(t_k), \ddot{Y}(t_k), \varphi(t_k), \dot{\varphi}(t_k), Y(t_k), O(t_k)\right\}^T. \tag{2}$$

In order to consider some other factors that are possibly not parameterised in the model, e.g. the vehicle hysteresis (e.g. due to the friction between the tyre and the road), human driver time delay between the point where his or her decision is made and the point where his or her action is taken. Ten states before time t_k were collected, and the average state from them was evaluated, denoted by $\bar{Z}(t_k)$, i.e.

$$\bar{Z}(t_k) = \sum_{i=1}^{10} Z(t_{k-i}) \cdot \left(1 - \frac{i}{10}\right). \tag{3}$$

The input to the neural network model, is thus defined as

$$\begin{cases} INPUT(t_k) = \left\{State(t_k), \overline{State}(t_k)\right\}^T \\ State(t_k) = \left\{r(t_k), \dot{Y}(t_k), \ddot{Y}(t_k), \varphi(t_k), \dot{\varphi}(t_k), Y(t_k), O(t_k)\right\}^T \\ \overline{State}(t_k) = \left\{\overline{r}(t_k), \overline{\dot{Y}}(t_k), \overline{\ddot{Y}}(t_k), \overline{\varphi}(t_k), \overline{\dot{\varphi}}(t_k), \overline{Y}(t_k), \overline{O}(t_k)\right\}^T \end{cases}. \tag{4}$$

The output to the neural network model is defined as

$$OUTPUT(t_k) = \left\{\delta(t_k), \dot{\delta}(t_k)\right\}^T, \tag{5}$$

where $\delta(t_k)$ is the steering wheel angle and $\dot{\delta}(t_k)$ is the steering wheel angular velocity.

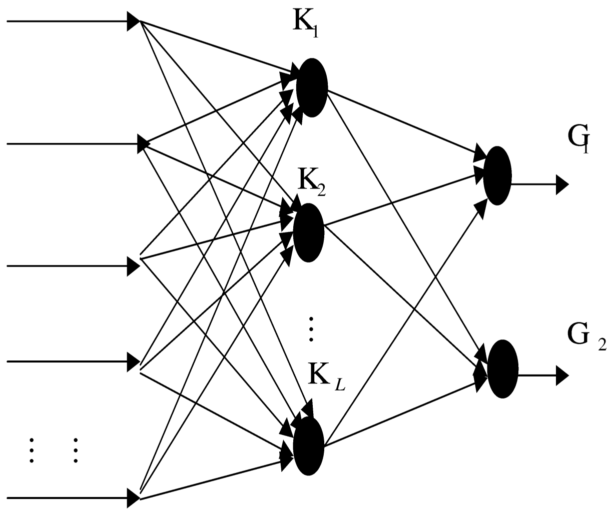
3.2 Neural network model configuration

Three configurations of the neural network model were employed for this application: the Counter Propagation Network (CPN) model, the Radial Basis Function Network (RBFN) model and the Back Propagation Network (BPN). In the following, a more detailed description of the CPN and RBFN models (respectively) will be given, while the BPN model refers to (Zurada, 1992).

3.2.1 The CPN model

The structure of the CPN model is shown in Figure 3. It is composed of one unsupervised training layer (Kohonen layer) and one supervised training layer (Grossberg layer) (Zurada, 1992). The Kohonen layer adopts a ‘self-organisation’ method to cluster the input data. This layer fulfils the functions of information compression and pattern recognition. Consequently, the whole network's training time is enormously reduced. The Grossberg layer performs the supervised training to ensure the accuracy of the network. The details of the training procedure for the CPN model are presented as follows.

Figure 3 CPN topology



3.2.1.1 Unsupervised training for the Kohonen layer

Define $\vec{X}$ as the input vector, i.e. $\vec{X} = \{x_1, x_2, \ldots, x_N\}^T$; N is the total number of neurons in the input layer. The sample input vector $\vec{X}_i$ is defined as $\vec{X}_i = \{x_1^i, x_2^i, \ldots, x_N^i\}^T$, $i = 1, 2, \ldots n$, where n is the total number of samples. Let $\vec{W}_j = \{w_{1j}, w_{2j}, \ldots w_{Nj}\}^T$ be the weight vector corresponding to neuron j in the Kohonen layer. The weights between the input layer and the Kohonen layer are determined based on the following steps:

Step 1: For a sample input vector, $\vec{X}_i$, calculate its distance to each neuron in the Kohonen layer, i.e.

$$ED_{ij} = \sum_{k=1}^{N} \left(x_k^i - w_{kj}\right)^2, \quad j = 1, 2, \ldots, L. \tag{6}$$

Step 2: Find the minimal value among ED_{ij} ($j = 1, 2 \ldots, L$), denoted as j^* and ED_{ij^*}.

Step 3: Adjust the weights between all the neurons in the input layer and neuron j^* in the Kohonen layer by following the method outlined below; details are referred to (Graf and Jackel, 1989).

Define h as the number of iterations, $h = 1, 2 \ldots,$. Define $\alpha(h)$ as a step, which will gradually reduce with the iteration. The iteration formula is given by (for the input vector i)

$$\vec{W}_{j^*,h+1} = \vec{W}_{j^*h} + \alpha_h \left[\vec{X}_i - \vec{W}_{j^*h}\right]. \tag{7}$$

The iteration stops when α_h is sufficiently small. Repeat the above steps for all the input samples, i.e. $i = 1, 2, \ldots n$.

3.2.1.2 *Supervised training for the Grossberg layer*

Following the training procedure for the Kohonen layer, a set of neurons in the Kohonen layer is obtained, which is denoted as $Q_{j^*} = \{j_i^* | i = 1, 2, \ldots n\}$. The following procedure applies to the training of the weight between Q_{j^*} and the output layer. Define $\vec{B}_i$ as the weight vector for the Grossberg layer between a neuron $j_i^* (i = 1, 2, \ldots n)$ and the output layer, i.e.

$$\vec{B}_i = \{b_{i1}, b_{i2}, \ldots b_{iG}\}, \tag{8}$$

where G is the total number of neurons in the output layer.

The goal of the training is to determine $\vec{B}_i$ $(i = 1, 2, \ldots n)$ by a set of pairs $\langle \vec{X}_i, \vec{Y}_i \rangle$, where $\vec{Y}_i$ is the output, i.e. $\vec{Y}_i = \{y_1^i, y_2^i, \ldots y_G^i\}^T$. The basic function for iteration to determine $\vec{B}_i$ is given by

$$\vec{B}_{i,h+1} = \vec{B}_{ih} + \beta\left[\vec{Y}_i - \vec{B}_{ih}\right], \tag{9}$$

where β is a constant $(0 \prec \beta \prec 1)$, given as 0.8 in the present study.

The stop criteria for iteration is

$$\left|\vec{B}_{h+1} - \vec{B}_h\right| \prec \varepsilon, \tag{10}$$

where ε is a small number.

3.2.2 *The RBFN model*

The RBFN model can be considered, initially, a two layer network model, as shown in Figure 4. The idea is to redefine the hidden layer as a non-linear function or transformation of the input layer. As such, the input layer is extended to be $x_1, x_2, \ldots, x_N, z_1, z_2, \ldots, z_M$. There are many definitions of the non-linear function. The RBFN model, in particular, defines the non-linear function as follows.

$$z_{ip} = \Phi_i\left(\left\|\vec{X}_p - c_i\right\|\right) = \exp\left\{-\sum_{j=1}^{N}\left[\frac{(x_{pj} - c_{ij})^2}{2\sigma_{ij}^2}\right]\right\}, \quad i = 1, 2, \ldots, M, \tag{11}$$

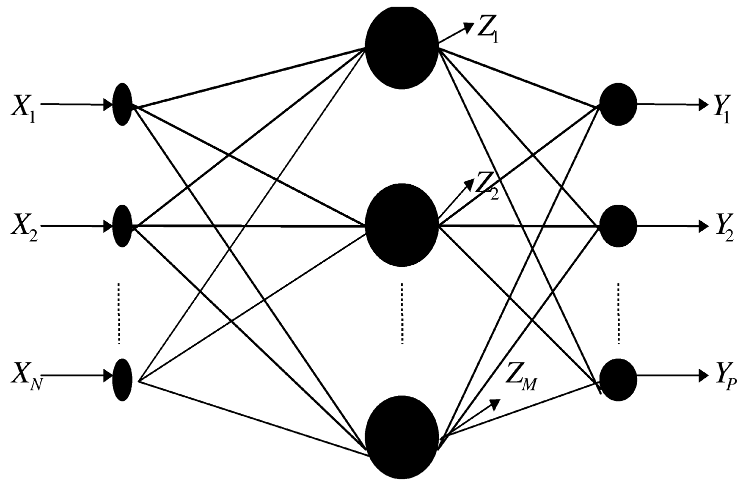
where z_{ip} is the output of the non-linear transformation of the neuron i in the hidden layer, corresponding to sample $\vec{X}_P$, $\vec{X}_P = \{x_{p1}, x_{p2}, \ldots x_{pN}\}^T$; $\vec{C}_i$ is a vector which is defined as $\vec{C}_i = \{c_{i1}, c_{i2}, \ldots, c_{iN}\}^T$; c_{ij} is the centre of the function Φ_i, and σ_{ij} is the width of the basis function. The output of the RBFN model is given by

$$y_q = \sum_{k=1}^{M} w_{qk} \cdot z_k, \tag{12}$$

where y_q is the output of the q node; w_{qk} is the weight of the output layer. The training of the RBFN model includes the determination of (a) c_{ij}, (b) σ_{ij}, and (c) w_{ij}. A so-called two-stage training algorithm (Orr, 1996) was adopted in the present study. In this algorithm, the determination of the c_{ij} and σ_{ij} is decoupled with the determination of w_{ij}. In particular, the determination of c_{ij} and σ_{ij} is based on the sample input vector $\vec{X}_k$, $k = 1, 2, \ldots P$, where P is the total number of the sample or training data. The determination of the weight w_{ij} is through the supervised training (i.e. training using target information $\vec{Y}_P$).

Ref. Orr, 1998 – changed to 1996 to match ref. list. Is this OK?

Figure 4 RBFN topology



3.3 Training data, data production and implementation

Five male drivers participated in the production of the training set. Their driving experiences were widely spread from one year to 20 years. Five different cars were used. Five drivers used five cars, alternately. The drivers were requested to drive one hour every day within one week. They drove with different required velocities: 30–90 km/hr. The road condition was dry and even. Nearly 10,000 data points were produced. Among them about 500 points were used for cross testing of the neural network models. A LC-761 Velocity Meter and a CDY-1A Vehicle Dynamic Instrument were used to measure vehicle dynamic parameters (e.g. longitudinal and lateral velocity, yaw angle velocity, longitudinal and lateral acceleration). All the measured signals were collected by an LSFX-IV Vehicle Road Testing Data Acquisition System with the sampling frequency of 50 Hz. The data were processed with the LabView data processing system.

The CPN model was implemented using Turbo C, while both RBFN and BPN models were implemented in the Matlab environment (Lin, 1997). The system configuration of the computer used for training all the three models was as follows: operating systems: Windows 95; windows version: 4.0; processor: Pentium; total physical memory: 32396 kb; available physical memory: 4540 kb; USER memory available: 86%; GDI memory available: 90%; swap file size: 14848 kb; swap file usage: 31%; swap file setting: dynamic; available space on drive C: 107264 kb; available space on drive D: 489624 kb. Main design parameters of the RBFN model were as follows:

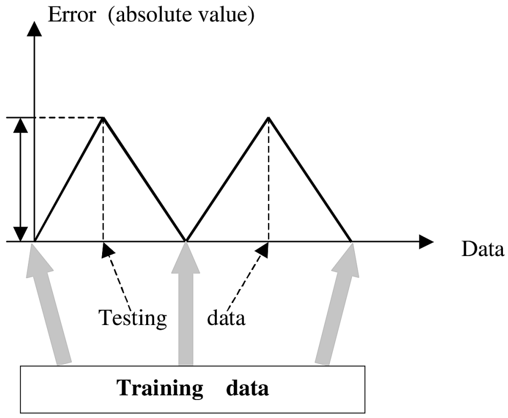
- $DF = 10$, frequency of progress displays (in neurons)
- $ME = 1000$, maximum number of neurons
- $EG = 0.02$, sum-squared error goal
- $SC = 0.01$, spread constant for radial basis functions.

4 Results and discussion

4.1 Comparison between BPN, CPN and RBFN

For the purpose of comparison, three measures for neural network models are considered: training time, error-tolerance, and accuracy. The training time is the time needed to train a neural network. The error-tolerance was measured by deliberately giving noises to selected training data and then observing the change in the output of the model. The accuracy was measured by calculating the error for the testing data points (see Figure 5). It is noted that in Figure 5 the testing result shows a so-called ‘zigzag’ pattern, which means that the trained neural network may produce a kind of ‘constant’ error to some input to the neural network model.

Figure 5 Validation of the extension capability of neural network models



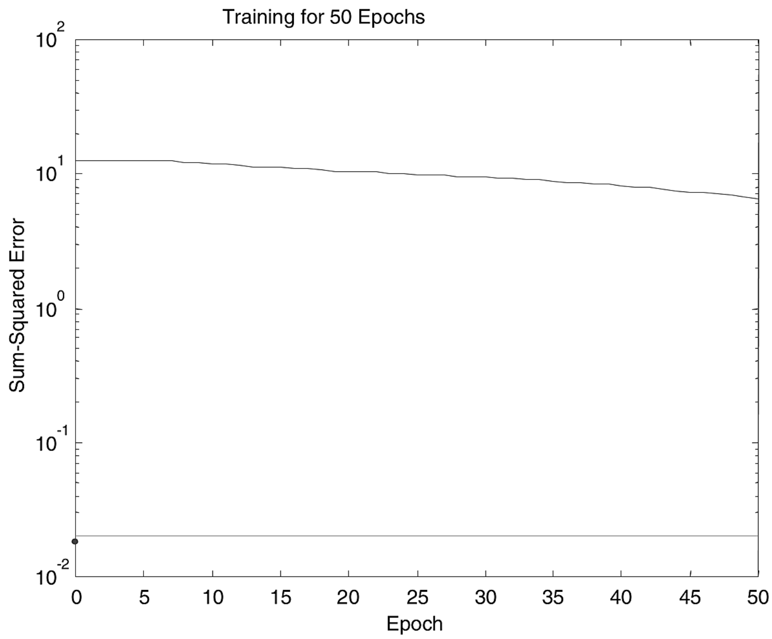
In the following, the three different neural network configurations are compared in terms of these three measures: training time, accuracy, and error-tolerance capability.

4.1.1 Training time

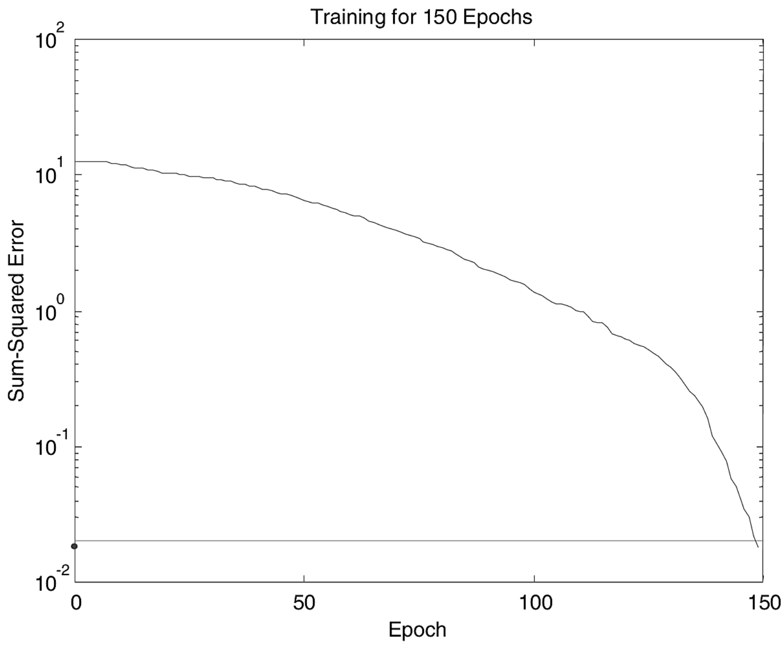
With the 9500 training data, the complete training took 8.5 minutes for the CPN model, 8 minutes for the RBFN model, and 40 minutes for the BPN model. It is noted that the implementation conditions between the RBFN/BPN and the CPN were slightly different, i.e. one in the Matlab environment and one in the C-language environment; the latter is faster than the former.

Furthermore, the RBFN and BPN models were trained with the same set of training data as in the Matlab environment. The results of the RBFN and BPN models are shown in Figures 6 and 7, respectively. In Figure 7, the information of learning rate is shown to get more idea of the training speed of the BPN model (the initial learning rate of BPN was set as LR = 0.01). The comparison of Figures 6 and 7 shows that the network resource (i.e. to network elements and connections) is greatly reduced with the RBFN model (trained RBFN only has 150 epochs; whereas BPN has 2067 epochs). Also, not directly given in Figures 6 and 7, the training time of the RBFN model is significantly less (reduction at 30% of the training time needed with the BPN model). For example, in one training session, the RBFM model took about two minutes, while BPN took about seven minutes.

Figure 6 RBFN training process

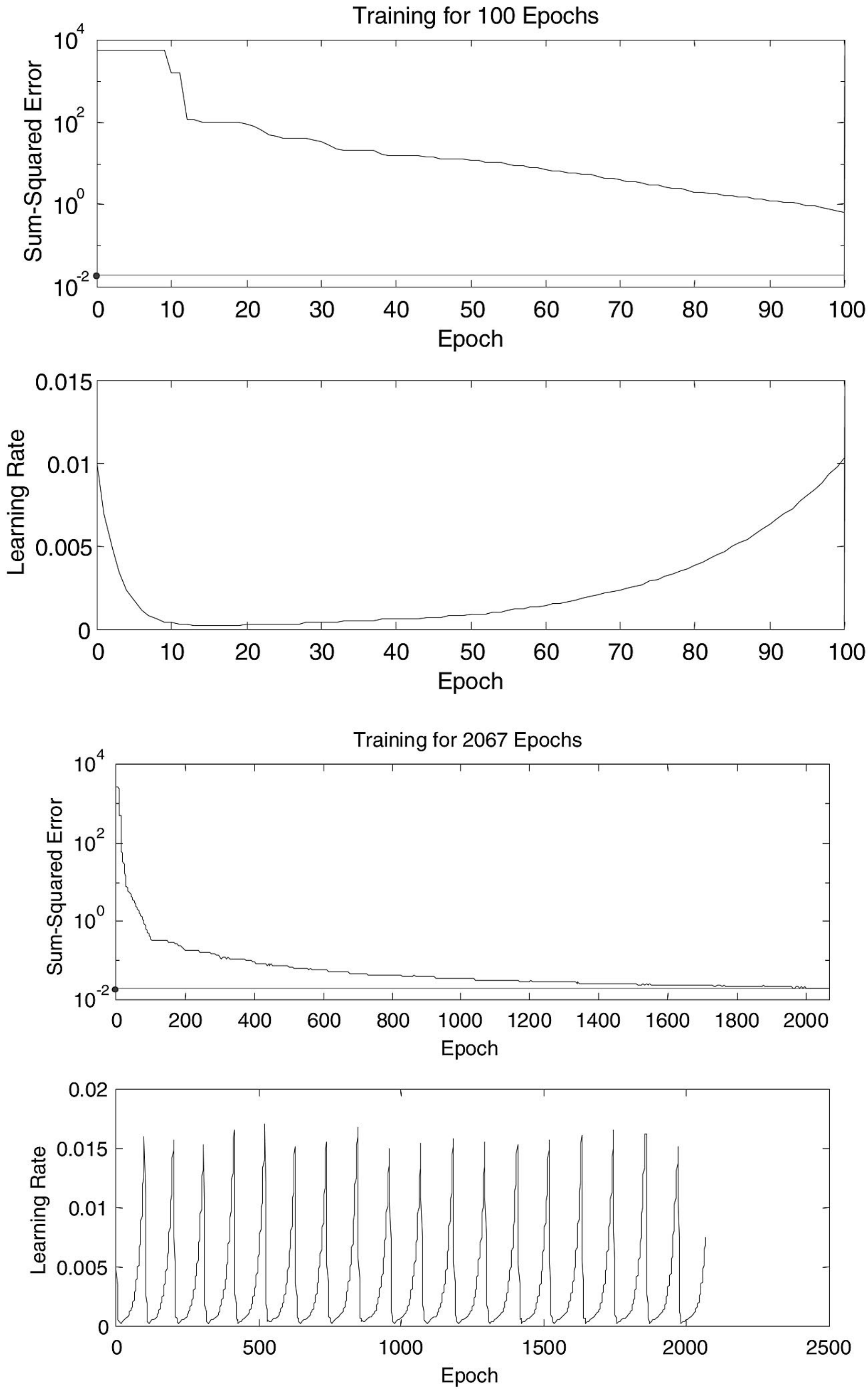


(a)



(b)

Figure 7 BPN training process



4.1.2 Error-tolerance capability

It was verified that the CPN model achieved the best error-tolerance result amongst the three models. This capability with the CPN model is brought about by the advantages of the weight-updating scheme of the Kohonen layer where the weight-updating scheme was extended to the weights of the vicinity of a node, instead of a single node. As such, 'similar' data tend to be clustered into the same class. The RBFN model achieved the best accuracy among the three models. In the following, some more detailed results of the RBFN and BPN models were presented.

Furthermore, the error-tolerance was examined through the following idea. The RBFN model was put into the DVE simulation system (see Figure 1). In the DVE system, the vehicle part was a model not a real system. Notice that when the driver behaviour model was developed, a real vehicle was used to produce the training data and thus further the neural network model. That said, there is a discrepancy between the training data and the data in the simulation system (produced by the vehicle dynamic model). Such a discrepancy is viewed as a kind of noise of the training data.

4.1.3 Accuracy

A cross validation test of the trained RBFN was performed. The result is shown in Figure 8. It can be seen that the maximal error (absolute) is about 0.16 degree. Note that usually the range of magnitude of the steering angle is about 45 degrees. The relative maximal error would be 0.4%, which is very small.

In summary, the comparison of BPN, CPN and RBFN is shown in Table 1 (H: Highest; M: Medium; L: Lowest):

Table 1 Comparison between BPN, CPN and RBFN

	BPN	*CPN*	*RBFN*
Training time	H	M	L
Accuracy	M	L	H
Error-tolerance	L	H	M

4.2 Comparison between simulation results and experimental results

The road experiment was carried out, in which the same device and instruments were used. The experiment was performed in an open field for the purpose of eliminating any disturbance, e.g. traffic, etc. (see Figure 9). Two drivers were hired for the experiment. They were required to drive with different speeds to follow three routes. The drive with these speeds should not make the lateral acceleration of the car greater than 0.4 g. The payment was concrete and the tyre of the vehicle was full of air. These experimental conditions ensured the accuracy of the vehicle dynamics model used in the DVE simulation system.

Figure 8 Absolute error of the RBFN model

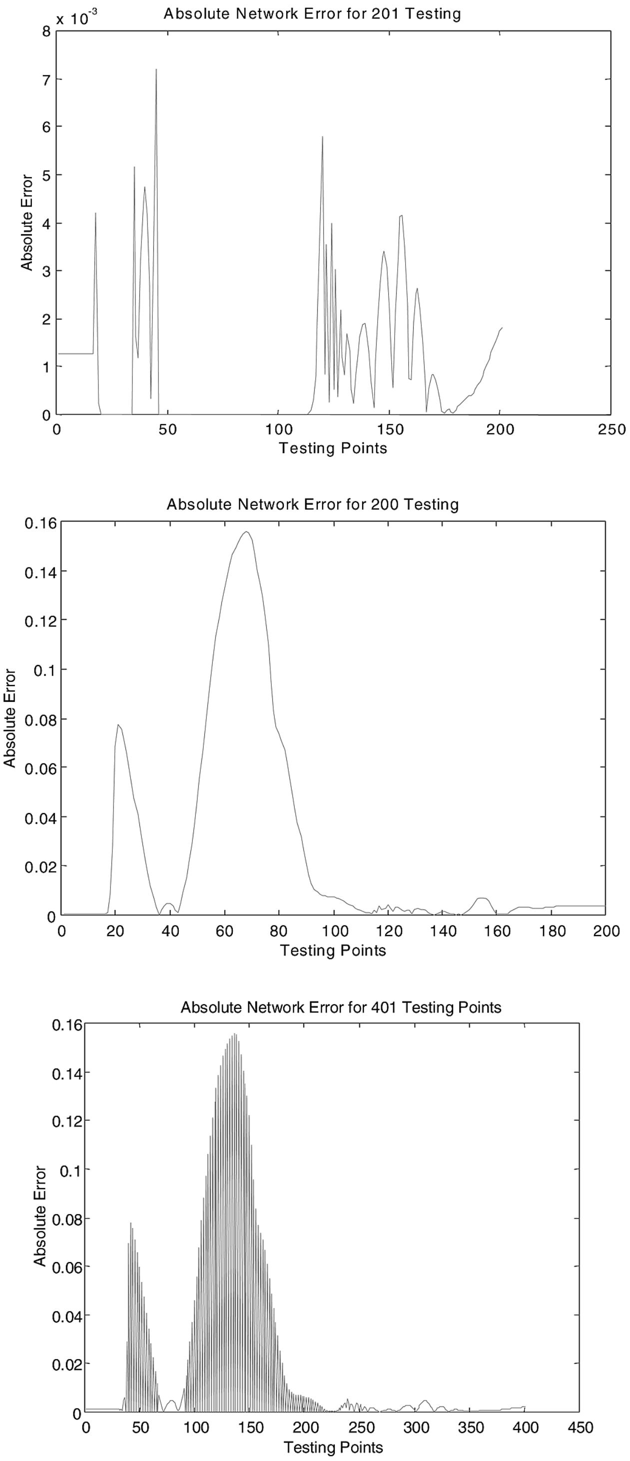
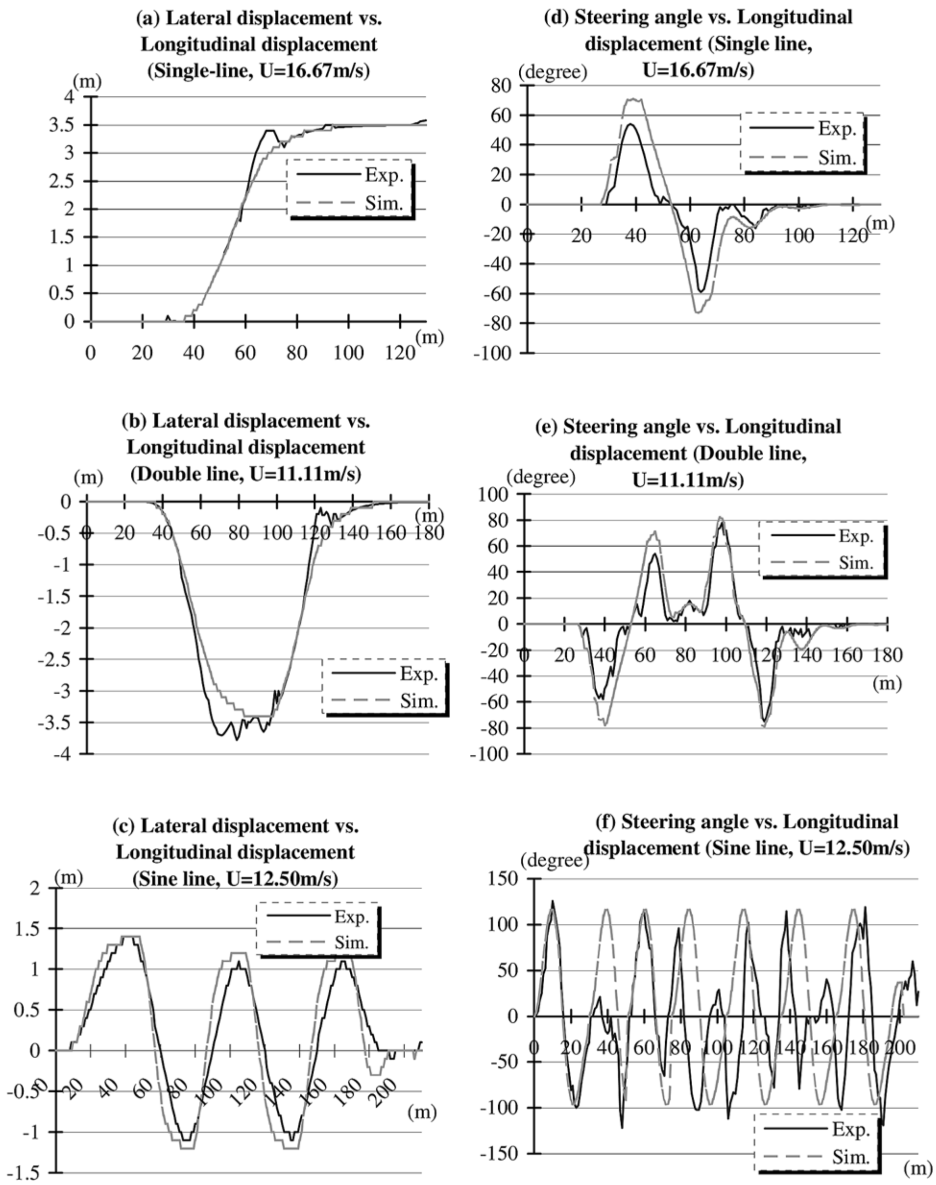


Figure 9 Experiment set-up

Figure 10 presents the result of the comparison. In Figure 10(a–c), the comparison on the driving route was shown. A good agreement can be found between the simulation result and the experimental result. In Figure 10(d–f), the steering angle versus the longitudinal progression is shown. Some time delay between the simulation result and experimental result can be observed from Figure 10(e). A closer observation shows that the simulation result is later than the actual result. This discrepancy may be due to relatively larger lateral acceleration which is close to the validity limit of the used vehicle dynamic model (i.e. 0.4 g lateral acceleration). Overall, the experimental result is in good agreement with the simulation result (the driver behaviour model plus the vehicle dynamics model), which implies the developed RBFN model has a good fault-tolerance capability.

Figure 10 Simulation results vs. experiment results



5 Conclusions

The study reported in this paper focused on two objectives, as stated in the introduction. Both theoretical and experimental studies were performed. As a result, with respect to Objective 1, a suitable neural network model – the Radial Basis Function Network model – shows promise in modelling driver behaviour in the

Driver–Vehicle–Environment system. The training time with the Radial Basis Function Network model can be further reduced by giving constant centre and/or the constant width of the basis function, or partial constants and partial variants, being tailored to more specific driving situations. With respect to Objective 2, the Driver–Vehicle–Environment simulation system was developed and the simulation results were in good agreement with the experimental results.

6 Future work

The discrepancy between the simulation result and the experimental result under the sine route may be alleviated by increasing the preview time. This is where more work needs to be done. Another possible cause for this delay may be attributed to the effect of driver mental workload, as it was observed that the mental workload of driver when driving the sine route is higher than for the single line or double line routes (Lin, 1997). Future work should consider how to incorporate mental workload in the driver behaviour model. In Lin (1997), the driver's mental workload was also measured, particularly the heart rate and blood pressure. It was found that these two measures were sensitive to the situation under interest. One possible way to incorporate the mental workload is to define these two measures as part of the input to the RBFN model.

Acknowledgement

This work has been supported by the Natural Science and Engineering Research Council of Canada (NSERC) and Chinese Natural Science Foundation (CNSF). All the authors thank these sources of support.

References

Ackermann, J. and Sienel, W. (1990) 'Robust control for automatic steering', *Proceedings of the American Control Conference*, pp.795–800.

An, P.E. and Harris, C.J. (1996) 'An intelligent driver warning system for vehicle collision avoidance', *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 26, No. 2, pp.254–261.

Atsumi, B., Sugiura, S. and Kimura, K. (1993) 'Evaluation of mental work load in vehicle driving by analysis of heart rate variability', *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting*, pp.574–579.

Colburn, C.J. (1978) 'Perceived risk as a determinant of driver behavior', *Accident Analysis & Prevention*, Vol. 10, pp.131–141.

Ervin, B. and Guy, Y. (1986) 'The influence of weights and dimensions on the stability and control of heavy-duty trucks in Canada', *University of Michigan Transportation Research Institute (UMTRI) Report*, No. 86–35.

Fenton, R.E. (1971) 'On the steering of automated vehicles: theory and experiment', *IEEE Transactions on Automated Control*, Vol. AC-21, No. 3, pp.306–315.

Fenton, R.E. (1988) 'On the optimal design of an automotive lateral controller', *IEEE Transactions on Vehicular Technology*, Vol. 37, No. 2, pp.108–113.

Graf, H.P. and Jackel, L.D. (1989) 'An electronic neural network circuits', *IEEE Circuits and Devices Magazine* (July), pp.44–48.

Gu, X. and Yu, Q. (1993) 'Driver-vehicle-environment closed-loop simulation of handling and stability using fuzzy control theory', *Proceedings of 13th IAVSD Symposium*, Chengdu, China, pp.172–181.

Hessburg, T. and Tomizuka, M. (1993) 'Fuzzy logic control for lateral vehicle guidance', *Proceedings of the IEEE Conference on Control Applications*, Vol. 2, pp.581–586.

Kamada, H., Gotoh, T. and Yoshida, M. (1992) 'Visual control system using image processing and fuzzy inferences', *Systems and Computers in Japan*, Vol. 23, No. 3, pp.13–25.

Kiencke, U., Majjad, R. and Kramer, S. (1999) 'Modeling and performance analysis of a hybrid driver model', *Control Engineering Practice*, Vol. 7, No. 8, pp.985–991.

Kornhauser, A.L. (1991) 'Neural network approaches for lateral control of autonomous highway vehicles', *VNIS '91 Vehicle Navigation and Information Systems Conference*, Vol. 2, pp.1143–1145.

LeBlanc, D. and Ervin, R.J. (1996) 'CPAC: an implementation of a road-departure warning system', *IEEE Contr. Syst. Mag.*, Vol. 16, pp.61–71.

Lee, A.Y. (1989) 'Algorithm for four-wheel-steering passenger vehicles', *Advanced Automotive Technologies*, pp.83–98.

Lin, Y. (1997) *Study on Handling Stability of Driver–Vehicle–Environment Closed-loop System Based on Neural Networks and Fuzzy Theory*, PhD thesis, China Agricultural University.

Lin, Y. and Yu, Q. (1996) 'A study on driver model using neural network systems', *The Proceedings of the 12th ISTVS International Conference*, Beijing, China, Oct. 7–11, pp.341–348.

MacAdam, C.C. (1980) 'An optimal preview control for linear systems', *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, Vol. 102, No. 3, pp.188–190.

MacAdam, 1981 – please supply no. of journal and page range.

MacAdam, C.C. (1981) 'Application of an optimal preview control for simulation of closed-loop automobile driving', *IEEE Transactions on System, Man and Cybernetics*, SMC-11, Vol. 6.

MacAdam, C.C. and Johnson, G.E. (1996) 'Application of elementary neural networks and preview sensors for representing driver steering control behavior', *Vehicle System Dynamics*, Vol. 25, pp.3–30.

Mussone, L., Reitani, G. and Rinelli, S. (1995) 'An ANN model to study driver behavior in urban environment', *10th International Conference on the Applications of Artificial Intelligence in Engineering*, pp.419–426.

Neusser, S. (1993) 'Neural control for lateral vehicle guidance', *IEEE Micro*, pp.57–66.

Orr, M.J.L. (1996) 'Introduction to redial basis function networks', *Technical Report* (April), Centre for Cognitive Science, University of Edinburgh.

Peng, H. (1994) 'A reusability study of vehicle lateral control system', *Vehicle System Dynamics*, Vol. 23.

Peng, H. and Tomizuka, M. (1991) 'Preview control for vehicle lateral guidance in highway automation', *Proceedings of the 1991 American Control Conference*, pp.2621–2627 (also *ASME Journal of Dynamic Systems, Measurements and Control*, 1993, Vol. 115, pp.679–686).

Pham, H.A., Hedrick, J.K. and Tomizuka, M. (1994) 'Combined lateral and longitudinal control of vehicles for automated highway systems', *Proceedings of the 1994 American Control Conference*, Baltimore, MD.

Shepanski, J.F. and Macy, S.A. (1987) 'Manual training techniques of autonomous systems based on artificial neural networks', *IEEE 1st International Conference on Neural Networks*, pp.21–24.

Shladover, S.E. and Desoer, C.A. (1991) 'Automatic vehicle control developments in the PATH program', *IEEE Transactions on Vehicular Technology*, Vol. 40, pp.114–130.

Smith, D.E. and Starkey, J.M. (1995) 'Effects of model complexity on the performance of automated vehicle steering controllers: model development, validation and comparison', *Vehicle System Dynamics*, Vol. 24, pp.163–181.

Tang, P. and Mann, D.D. (2003) 'Ergonomic requirements of a visual information display unit used for guidance of agricultural vehicles', *Journal of Agricultural Safety and Health*, Vol. 9, No. 1, pp.47–60.

Wilde, G.J.S. (1988) 'Risk homeostasis theory and traffic accidents – propositions, deductions and discussion of dissension in recent reactions', *Ergonomics*, Vol. 31, No. 4, pp.441–468.

Yang, X., Stiharu, I. and Rakheja, S. (1998) 'Study of driver factors affecting the directional response of an articulated vehicle using neural networks', *Transactions of CSME*, Vol. 22, No. 3, pp.291–306.

Zhang, W.J. (2003) 'A remark on modeling of complex systems', *Technical Report* (*AEDL-WJZ-2003-02*), Advanced Engineering Design Laboratory, University of Saskatchewan.

Zurada, J.M. (1992) *Introduction to Artificial Neural Systems*, St. Paul, MN: West Publishing Company.