Design of A Neuro-Controller for Vehicle Lateral Velocity and Yaw Rate Based Genetic Algorithm With Model Reference Guided

Mr. Ahmed Sabah Abdul Ameer Al-Araji

Received on : 7/3/2005
Accepted on : 29/9/2005

Abstract

In this paper, the aim of control problem is to achieve required yaw rate and reduce lateral velocity in a short period of time to prevent vehicle from sliding out the curvature. The structure of the controller used consists of modified Elman recurrent neural networks that learned on-line by using genetic algorithm teachings. Using of both front and rear wheels steering simultaneously has automatically controlled the vehicle lateral motion when the vehicle rotates the curvature. Therefore, it is used a feedback neural controller that is learned on-line in order to control the transient state output of the system by minimizing the error between the actual output of the system and the model reference output. The evolutionary techniques based on this algorithm are employed for the model-reference adaptive control scheme for this system.

الخلاصة

الهدف الرئيسي من المسيطر في هذا البحث هو تحقيق معدل الدوران المطلوب و تقليل السرعة الجانبية للمركبة في اقل وقت ممكن لمنع المركبة من الانزلاق خارج المنعطف.
أن هيكلية المسيطر المستخدم تتكون من الشبكة العصبية المرجعة المحسنة أيلمن المتعلمة باستخدام الخوارزمية الجينية بطريقة On-Line .
وتتم عملية السيطرة على السرعة الجانبية للمركبة ومعدل الدوران المطلوب أثناء دوران المركبة لمنعطف من خلال توجيه العجلات الأمامية و الخلفية من المسيطر العصبي ذات التغذية الراجعة لكي يسيطر على الحالة العابرة لإخراج المنظومة من خلال تقليل الخطأ بين النموذج المرجع والإخراج الحقيقي للمنظومة.
وقد تم تطوير التقنية التي أساسها الخوارزمية الموضوعة للمسيطر المتكيف للنموذج المرجع لهذا النظام.

1-Introduction

The application of the intelligent control schemes has attracted the attention of researchers in the field of control engineering. Practical characteristics of neural networks, fuzzy and knowledge-based systems applicable in control include the representation of arbitrary, multi-dimensional problem, nonlinear mapping function, learning and adaptation through example and the ability to combine large amount of data to form decisions or pattern recognition [1]. So many researchers are fond of field that deals with automotive motion engineering. The practical example for automotive motion are transportation vehicles inside factories, ship ports, and in the future as pat of an integrated system of automated high way traffic. Major research activities in automatic vehicle control are devoted to longitudinal control, which keeps the interval between vehicles by controlling vehicle speed, and lateral control, which maintains the lateral position of the vehicle in the lane by controlling steering angles [2]. To improve the lateral velocity, and yaw rate time response [3], adaptive fuzzy technique

Control & System Dept. University of Technology

used when the parameters of vehicle velocity, mass, and tires state are variable and change the gains of the controller are change automatically.

2- Mathematical Model of the Vehicle

The dynamics of vehicle lateral motion depends on many parameters such as vehicle speed, vehicle mass and tires state on road. The independent control of lateral and yaw motion requires at least one additional control input, which is independent of the front steering angle. There are three possible solutions for these inputs:

1- A four-wheel steering system whose control inputs are front and rear steering independently, (Lee, 1989; Karnopp and Wuh, 1989).
2- Using braking forces with different distributively on wheels (Xia and Law, 1989; Abe ,1989).
3- Control of torque driving wheels either they are front or rear, (Matsumoto and Tomizuuka, 1992).

Vehicle lateral and yaw motion occurs at the vehicle horizontal plane (x-y plane) with coordinates fixed to vehicle body as shows in the figure (1) and related symbols shows in appendix (1).

The considerations of deriving lateral motion equations at the plane (x-y) [4] are:

1- Steering angles (δ_f, δ_r) are small values (0.01 to 0.1) degree.
2- Wheel slip angle (α_i) is small degree.
3- Yaw rate (r) is small rad/sec.

Front vehicle speed is constant U=C; that is to mean vehicle acceleration towards x-axis equal zero. There is no motion towards z-axis. The linear dynamical model of vehicle lateral motion [3] with interaction in multiple-input-multiple-output system are expressed as the state space equations as follows:

$$\dot{x} = Ax + Bu \qquad (1)$$

where $x = \begin{bmatrix} V \\ r \end{bmatrix}$

$$A = \begin{bmatrix} -\dfrac{C_f + C_r}{MU} & -\dfrac{C_f a - C_r b}{MU} - U \\ -\dfrac{C_f a - C_r b}{IU} & -\dfrac{C_f a^2 + C_r b^2}{IU} \end{bmatrix}$$

$$B = \begin{bmatrix} \dfrac{C_f}{M} & \dfrac{C_r}{M} \\ \dfrac{aC_f}{I} & -\dfrac{bC_r}{I} \end{bmatrix} \qquad u = \begin{bmatrix} \delta_f \\ \delta_r \end{bmatrix}$$

Front cornering stiffness $C_f = C_1 + C_2$ and rear cornering stiffness $C_r = C_3 + C_4$ are used for convenience [5].

δ_f, δ_r represent front steering angle and rear steering angle respectively and they are both system inputs.

3- Recurrent Neural Networks /Genetic Algorithm Learning

Recurrent neural networks (RNN) have one or more feedback connections, where each artificial neuron is connected to the others [6]. The RNN structures are suitable to channel equalization and multi-user detection applications, since they are able to cope with channel transfer functions that exhibit deep spectral nulls, forming optimal decision boundaries and are less computationally demanding than MLP networks for these applications [7]. Among the available recurrent networks, modified E7lman networks as shown in figure (2) is one of the simplest types that can be trained using genetic algorithm and it used to minimize the oscillation or even instabilities to the training controller. The output of the jth context unit in the modified Elman network is given by:

$$h_j^o(k) = \alpha h_j^o(k-1) + h_j(k-1) \tag{2}$$

where $h_j^o(k)$ and $h_j(k)$ are respectively the output of the jth context unit and jth hidden unit and α is the feedback gain of the self-connections. The value of α adopted is the same for all self-connections and is not modified by the training algorithm. The value of α is between 0 and 1. A value of α nearer to 1 enables the context unit to aggregate more pattern outputs. The input and output units interact with the outside environment. While the hidden and context units do not. The input units are only buffer units " Scales". The output units are linear units, which sum the signals fed to them. The hidden units can have nonlinear activation functions such as sigmoidal activation functions. The context units are used only to memorize the previous activation's of the hidden units and can be considered to function as one-step time delays. From the figure (2) it can be seen that the following equations:

$$h(k) = F\{V1X(k), V2h^o(k)\} \tag{3}$$

$$O(k) = Wh(k) \tag{4}$$

where V1,V2 and W are weight matrices and F is a non-linear vector function. The multi-layered modified Elman neural networks shown in figure (2) that is composed of many interconnected processing units called neurons or nodes.

where:

$V1$: Weight matrix of the input units.

$V2$: Weight matrix of the context units.

W: Weight matrix.

L: Denotes linear node.

H: Denotes nonlinear node with sigmoidal function.

As can be seen the net consists of three layers: An input layer (buffer layer as scales), a single hidden layer and a linear output layer. The neurons in the input layer simply store the scaled input values. The hidden layer neurons perform two calculations. To explain these calculations, consider the general j'th neuron in the hidden layer shown in figure (3). The inputs Xi and h_i^o to this neuron consist of an ni – dimensional vector and (ni is the number of the input nodes). Each of the inputs has a weight V1 and V2 associated with it. The first calculation within the neuron consists of calculating the weighted sum net_j of the inputs as:

$$net_j = \sum_{i=1}^{ni} V1_{j,i} \times X_i + V2_{j,i} \times h_i^o \tag{5}$$

Next the output of the neuron h_j is calculated as the continuous sigmoid function of the net_j as:

$$h_j = \mathrm{H}(net_j) \tag{6}$$

$$\mathrm{H}(net_j) = \frac{2}{1+e^{-net_j}} - 1 \tag{7}$$

Once the outputs of the hidden layer are calculated, they are passed to the output layer. In the output layer, a single linear neuron is used to calculate the weighted sum (neto) of its inputs (the output of the hidden layer as in equation (8)).

$$\mathrm{neto}_k = \sum_{j=1}^{nh} W_{k,j} \times h_j \tag{8}$$

where nh is the number of the hidden neuro (nodes) and $W_{k,j}$ is the weight between the hidden neuron h_j and the output neuron. The single linear neuron, then, passes the sum (neto_k) through a linear function of slope 1 (another slope can be used to scale the output) as:

$O_k = L(neto_k)$ where L(x)=x (9)

Thus the outputs at the output layer are δ_f, δ_r which are denoted by O1, O2 respectively.

In this work, the GA with real coding rather than binary is used as follows: Each chromosome is considered as a list (or "vector") of the total weights of neural networks. The encoding is shown in figure (4) and the weights are read off the network in a fixed pre-defined order and placed in a vector. Each "gene" in the chromosome is a real number. To calculate the fitness of a given chromosome, the weights in the chromosome are assigned to the links in the corresponding modified Elman networks, the network is run on the training set, and an objective function is returned. An initial population of weight vectors was chosen to be 60 individuals, with each weight being between –1 and +1. The mutation operator adds a random value between –0.5 and +0.5 to the selected weight on the link. The crossover operator two mating vectors and exchanges the information by exchanging a subset their components. The result is a new pair of vectors, each of which carries components from both of the parent vectors. The mean square of error (MSE) for multi-input multi-output (MIMO) is used as an objective function to be minimized with the GA:

$$MSE = \sum_{k=1}^{Np} \frac{[(y_{m1}(k) - y_{p1}(k))^2 + (y_{m2}(k) - y_{p2}(k))^2]}{Np} \quad (10)$$

where:

Yp1(k) is the first output of the plant at sample k.

Ymr1(k) is the first output of the linear model reference at sample k.

Yp2(k) is the second output of the plant at sample k.

Ymr2(k) is the second output of the linear model reference at sample k.

Np is the number of the training patterns.

Since the GA maximizes its fitness function, it is necessary therefore to map the objective function (MSE) to a fitness function. It is used objective –to- fitness transformation is of the form [8, 9, and 10].

$$fitness = \frac{1}{objectivefunction + \mu} \quad (11)$$

Where μ is a constant chosen to avid division by zero.

4-Neural Networks Controller

The control of the multi-variable linear system is considered in this section. The approaches used to control the system undepend on the information available about the system **"does not use identification to the system"**. The feedback neural controller is very important because it is necessary to stabilize the tracking error dynamics of the system when the output of the system is drifted from the input reference. The feedback neural controller is used based on the minimization of the error between the model reference & the actual output system in order to achieve good tracking of the reference signal with minimum time and to use minimum effort. In direct model reference adaptive controller (MRAC) with parallel model reference used here for the feedback neural controller, the adjustable parameters of neural network controller are adapted by genetic algorithm technique [11]. The integrated control structure that consists of the model reference and neural controller type modified Elman recurrent neural networks thus brings together the advantages of the neural model with the robustness of feedback. The general structure of the neural controller can be given in the form of the block diagram shown in figure (5).

5-The Proposed Algorithm For The Neuro-Controller Type Modified Elman Recurrent Neural Networks

The following genetic procedure is introduced for training the modified Elman recurrent neural network controller for the (MIMO) plant to track the reference model trajectory:

Step 1: Initialize the genetic operators: the crossover probability Pc, the mutation probability Pm, the population size, and the maximum number of generations.

Step 2: Generate the initial population randomly.

Step 3: For each individual in the population, compute the objective function **MSE**, and then calculate the fitness function as in equation (25), where μ will be chosen as an input coefficient equal to 1.

Step 4: Put in descending orders all the chromosomes in the current population.

Step 5: Select individuals using hybrid selection method (Roulette Wheel plus deterministic selection). The real coded genetic operators of mutation and crossover (single point) is applied.

Step 6: Stop if a maximum number of generations of genetic algorithms are achieved, otherwise increment the generations by one and go to Step 3.

5- Case Study

In this section, the vehicle parameters as appendix (2) is taken to clarify the features of the neural controller explained in section three and applied the algorithm in section four. And convert the state space equation (2) that is described the system to the linear discrete-time invariant systems [12] in order to easy solves the algorithm in section four.

$$x(k+1) = Gx(k) + Hu(k) \tag{12}$$

where x(k) is the state vector, u(k) is the input vector, and T is the sampling period The matrices G and H depend on the sampling period T "once this period is fixed, G and H are constant matrices".

$$G(T) = e^{AT} \tag{13}$$

$$H(T) = \left(\int_0^T e^{AT} dt\right) \times B \tag{14}$$

After substitution the parameters of the vehicle in appendix (2), in the equation (2) and then convert to the linear discrete-time equation when the sampling period T is equal to 0.1sec by using equations (14 & 15). The system has three cases because the vehicle velocity U is changed to (10, 20 & 30) m/sec.

In order to overcome a numerical problem that is involved within real values. Scaling function has to be added at the neural network terminals to convert the scaled values to actual values and vice versa. Therefore the signals entering to or emitted from the network have been normalized to lie within (-1 & +1).

In this simulation, the proposed control scheme is applied to the vehicle model and the real-coded genetic algorithm is set to the following parameters:

Population size (N_{POP}) is equal to 60.

Crossover Probability (Pc) is equal to 0.8.

Mutation Probability (Pm) is equal to 0.05.

Maximum number of generations is 1500.

The training pattern (Np) used was taken as 250 as the desired trajectory. The modified Elman recurrent neural networks are used to minimize the performance error between the model reference and the actual output. The equation of the model reference for the two outputs is taken from [13] for more stability and without any oscillation in the response:

$$y_{mr}(k+1) = 0.1y_{mr}(k) + 0.9y_{des}(k+1) \quad (15)$$

Convergence is achieved when the performance error falls below a pre-specified value.

$$e_{mr}(k+1) = y_{mr}(k+1) - y_p(k+1) \quad (16)$$

where $e_{mr}(k)$ is the model reference error output.

The performance of the proposed controller is evaluated using the closed-loop step lateral velocity and yaw rate responses for linear discrete-time model as equations (14 & 15). The desired lateral velocity must be zero to over come the vehicle may rotate around itself at high vehicle velocity. And desired yaw rate must be verified:

$$(r_d = \frac{U}{R}) \quad (17)$$

where R is curvature radius.

After training it can be observed that the actual output of the system is following the desired trajectory (model reference) can be shown as the figure (6 & 7). Where the figure (6) is the yaw rate response and its fast response with no overshoot and steady-state error is zero and the transient time is approximately equal to 0.1 sec when the vehicle velocity is change as (10, 20, & 30) m/sec with fixed curvature radius equal to 100m.

Figure (7) is the lateral velocity response that has fast response with very small magnitude oscillation range of ($\pm 1.35 \times 10^{-7}$) approximately equal to zero as the desired lateral velocity to over come the vehicle may rotate around itself when the vehicle velocity is change (10, 20, & 30) m/sec with fixed curvature radius equal 100m. The robustness of feedback neural control action will be kept the maximum magnitude of the lateral velocity in the transient response is equal to ($\pm 1.35 \times 10^{-7}$) m/sec when the velocity of the vehicle is changed and achievement the desired lateral velocity and yaw rate.

The yaw rate control and the lateral velocity can be achieved by two feedback control action "the front steering angle and rear steering angle" as shown in figures (8 & 9). The error between the two desired outputs and the two actual outputs of the system is very small as shown in figure (10 & 11)). Figure (12) is described the best objective function ***MSE*** for the ***MIMO*** system.

6-Conclusions

The structure of the modified Elman recurrent neural network as an adaptive controller with geneitic algorithm learned as the proposed structure of controller and successfully simulated to multi-input multi-output linear system as the example. Using feedback neural controller to control the front and rear wheels steering. So that, the output of the system lateral velocity and yaw rate follows the output of the predefined desired inputs "model reference" and genetic algorithm is used to learn the controller with minimum time and more stability of the controller.

The proposed control structure has shown the ability to minimize the error between the desired output model reference and the actual output of the system as well as the control action, excellent set point tracking, as it was clear when applied to the example.

Appendix (1)

Nomenclature

a= distance from the center of mass to

front axle
b= distance from the center of mass to rear axle
C_i = tire cornering stiffness
g= acceleration of gravity
I= vehicle moment of inertia
M= vehicle mass
r= yaw rate
r_d = desired yaw rate
R= curvature radius
U= vehicle velocity
V= lateral velocity
Y_i = road-tire interaction force
α_i =wheel slip angle
δ_i =steering angle
η_i =wheel traveling direction

Appendix (2)

Vehicle nominal parameters

M=1000Kg
a=1m
b=1.5m
I=1500Kg m^2
C_f =55000 N/rad
C_r = 45000 N/rad
U= 10, 20 & 30 m/sec
R=100m

Reference

[1]-Passino K. M. and Yurkovich S. "Fuzzy Control" wesety longman Inc.1998.

[2]-N. Matsumoto,"Vehicle Lateral Velocity & Yaw Rate Control Input", J of Dynamic system, Measurement & Control, Vol. 114, September 1992, pp 401-407.

[3]- H. S. Al-Araji & A. S. Al-Araji " Vehicle lateral velocity and yaw rate control using robust integral controller" Engineering and Technology journal by University of Technology Vol. 23, No. 3, pp. 95-103, 2004.

[4]-Wong J., "Theory of Ground Vehicles", New York, N.Y. Wiely. 1993.

[5]-J.C. Whiteheed, "Rear Wheel Steering Dynamics Compared to Front Steering", Jof Dynamic System, Measurement & Control, Vol. 112 March 1990, pp 88-93.

[6]-Rodrigo C. & Raimundo S." An approximate minimum BER approach to multi-user detection using recurrent neural networks" IEEE. 2002.

[7]-S. Haykin, "Neural Networks:A Comprehensive Foundation" 2nd Edition, Prentice-Hall, 1999.

[8]-Al-Said I.A.M." Genetic Algorithm Based Intelligent Control".Ph.D.,Thesis, University of Technology, Jan. 2000.

[9]-Jasim M." Evolutionary Techniques for Intelligent PID Controllers" M.Sc. Thesis, University of Technology, March 2001.

[10]-Pham D.T. and Xing L." Neural Networks for Identification, Prediction and Control", Springer-Verlag, London Ltd, 3rd Printing 1997.

[11]- K. S. Narendra and S. Mukhopadhyay, "Adaptive control using neural networks and approximate models," IEEE Trans. Neural Networks, vol. 8, no. 3, pp. 475-485, 1997.

[12]- John Dorsey "Continuous and Discrete Modeling, Identification, Design & Implementation" Georgia Institute of Technology Atlanta, Georgia, 1st edition 2002.

[13]-E. P. Nahas, M. A. Henson and D. E. Seborg, "Nonlinear internal model control strategy for neural network models," Computers Chem. Eng., vol. 16, no. 12, pp. 1039- 1057, 1992.

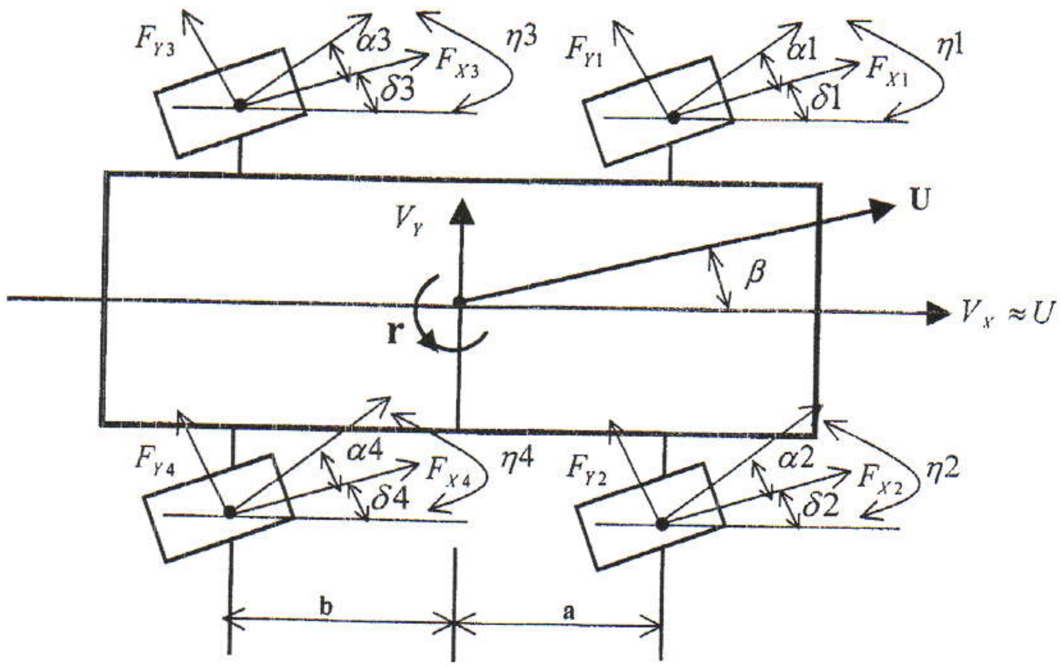


Fig (1): Schematic of the mathematical vehicle model

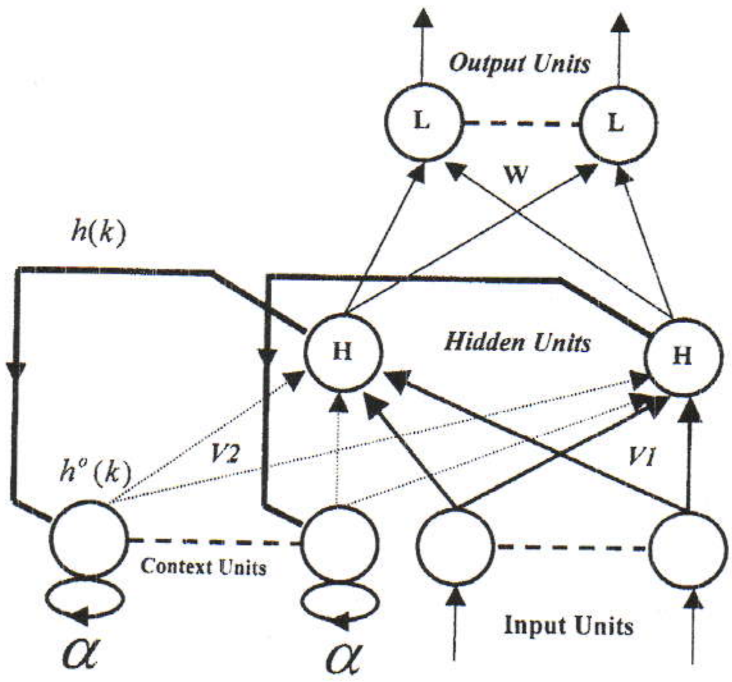


Fig (2): The Modified Elman Recurrent Neural Networks

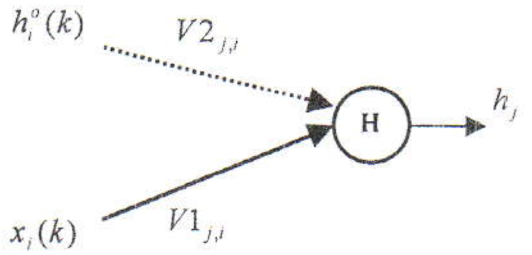
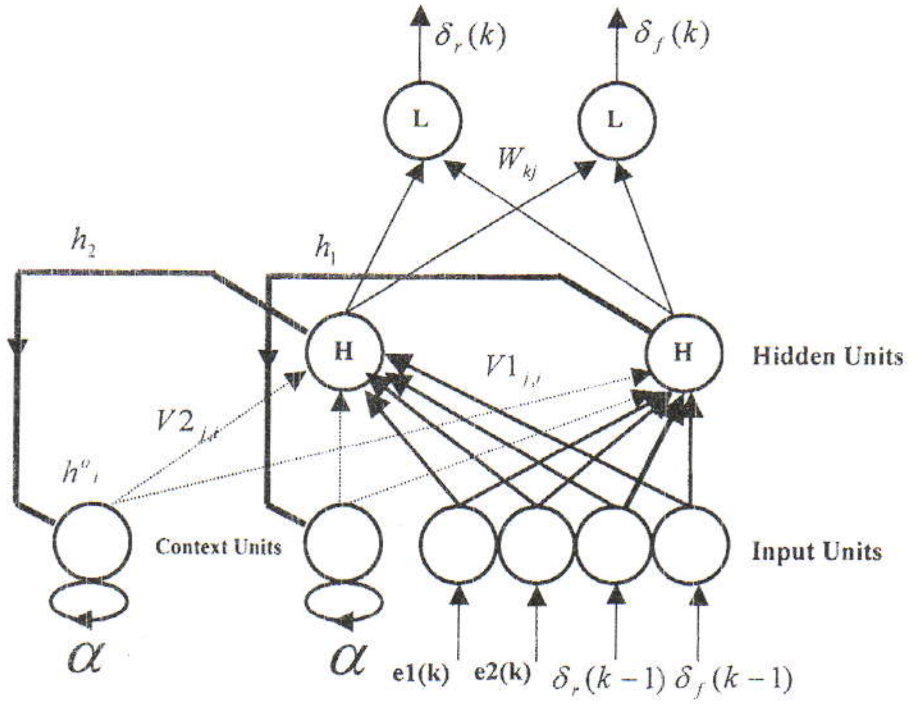


Fig (3): Neuron j in the hidden layer



$V1_{1,1}$	$V1_{1,2}$	$V1_{1,3}$	$V1_{1,4}$	$V1_{2,1}$	$V1_{2,2}$	$V1_{2,3}$	$V1_{2,4}$	$V2_{1,1}$	$V2_{1,2}$	$V2_{2,1}$	$V2_{2,2}$	$W_{1,1}$	$W_{1,2}$	$W_{2,1}$	$W_{2,2}$

Fig (4): The weights of modified Elman recurrent neural networks Encoding real-value to the chromosomes

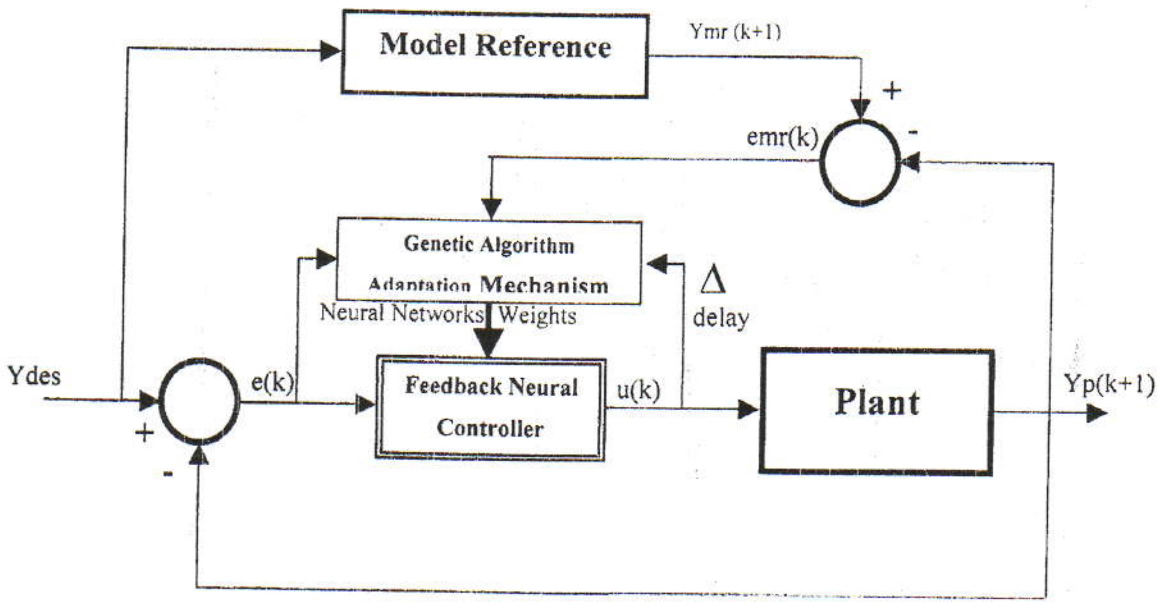


Fig (5): The General Proposed of Neural Networks Controller

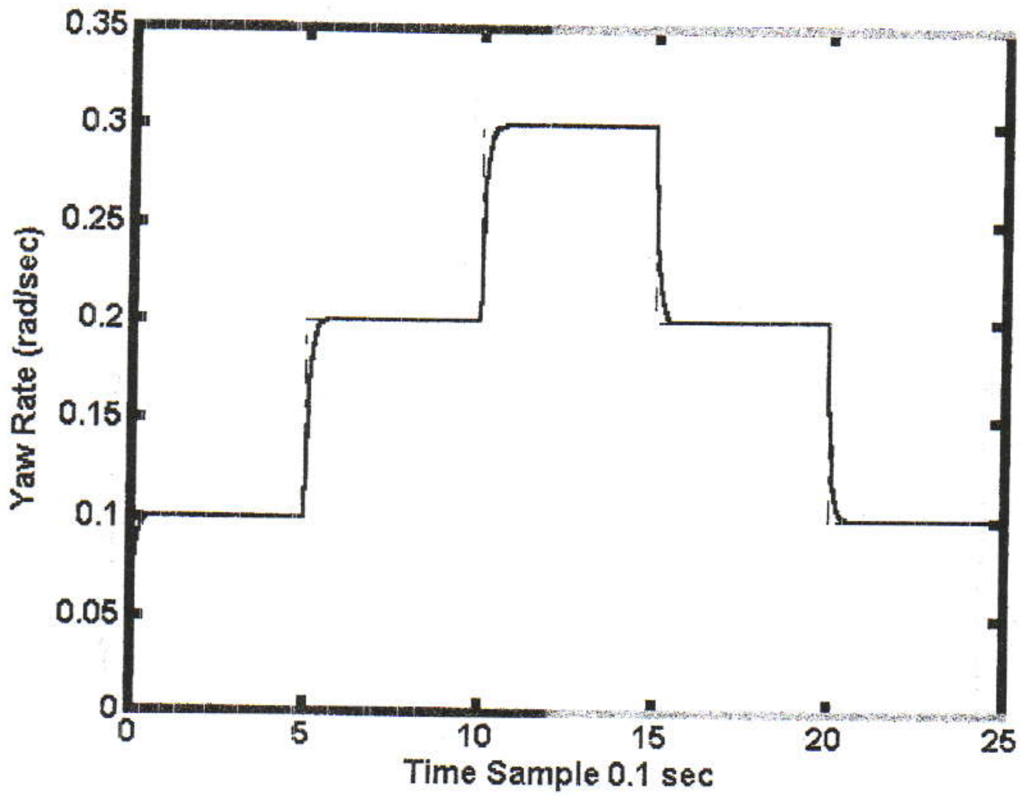


Fig (6): Yaw rate response (rad/sec)

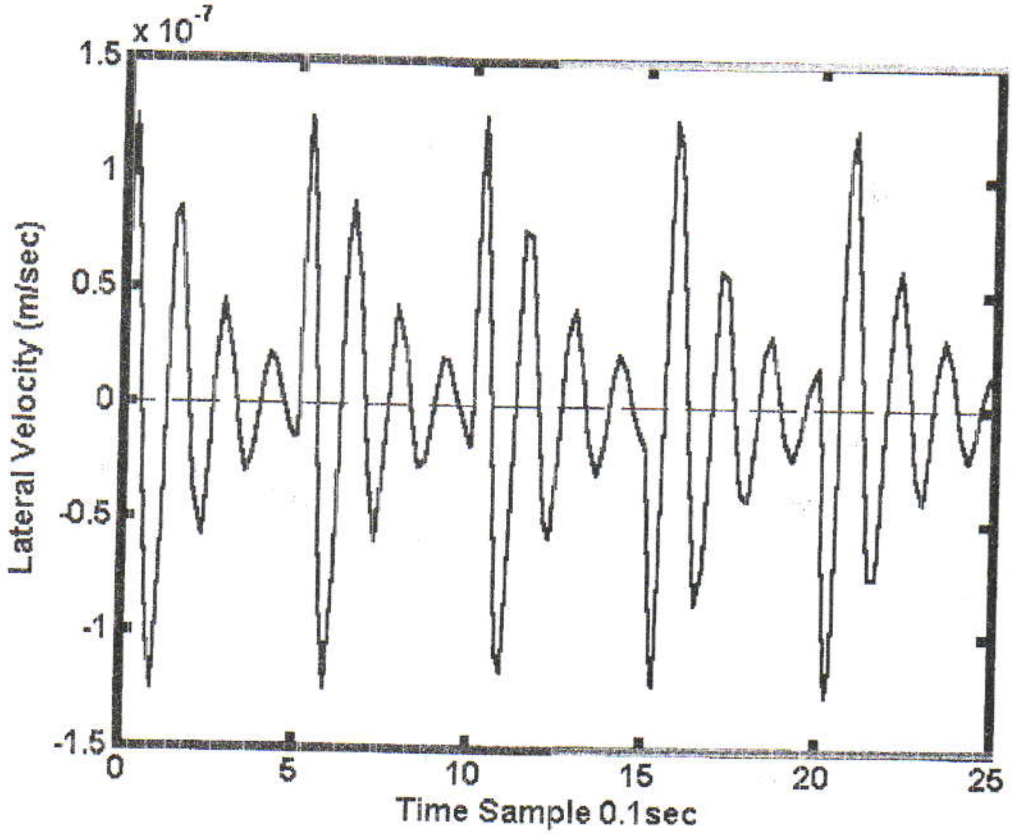


Fig (7): Lateral Velocity response (m/sec)

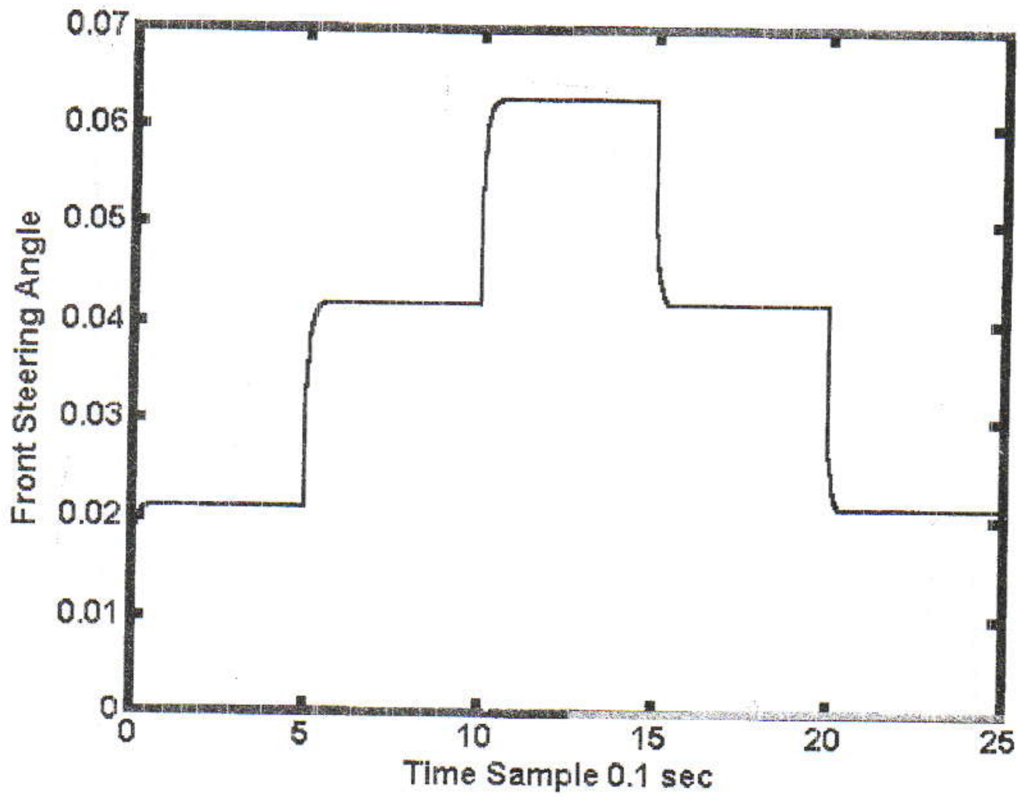


Fig (8): The Front Steering Angle "Control Action"

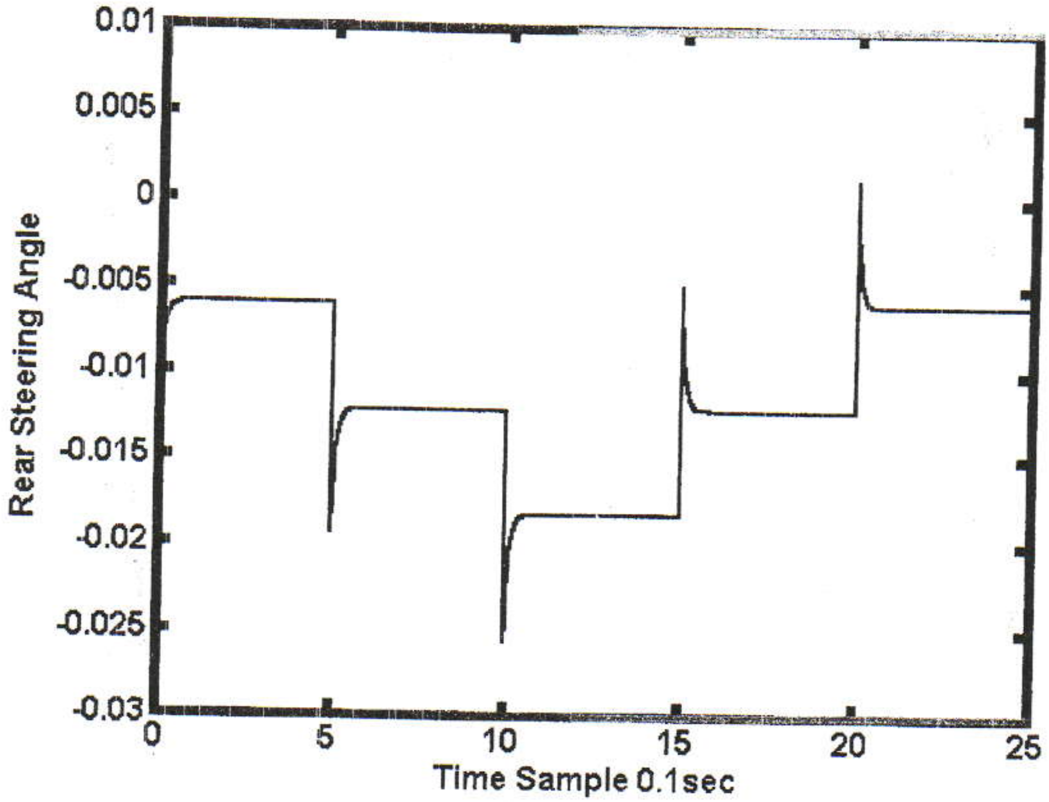


Fig (9): The Rear Steering Angle "Control Action"

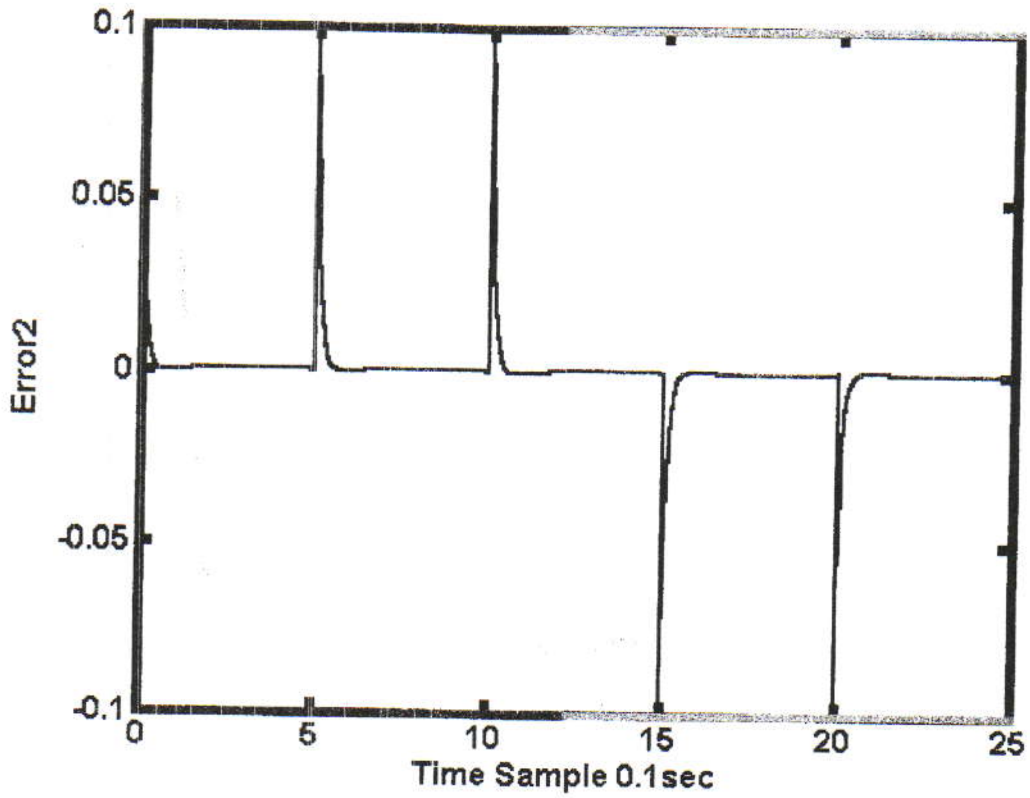


Fig (10): The Error between Set point & the Yaw Rate

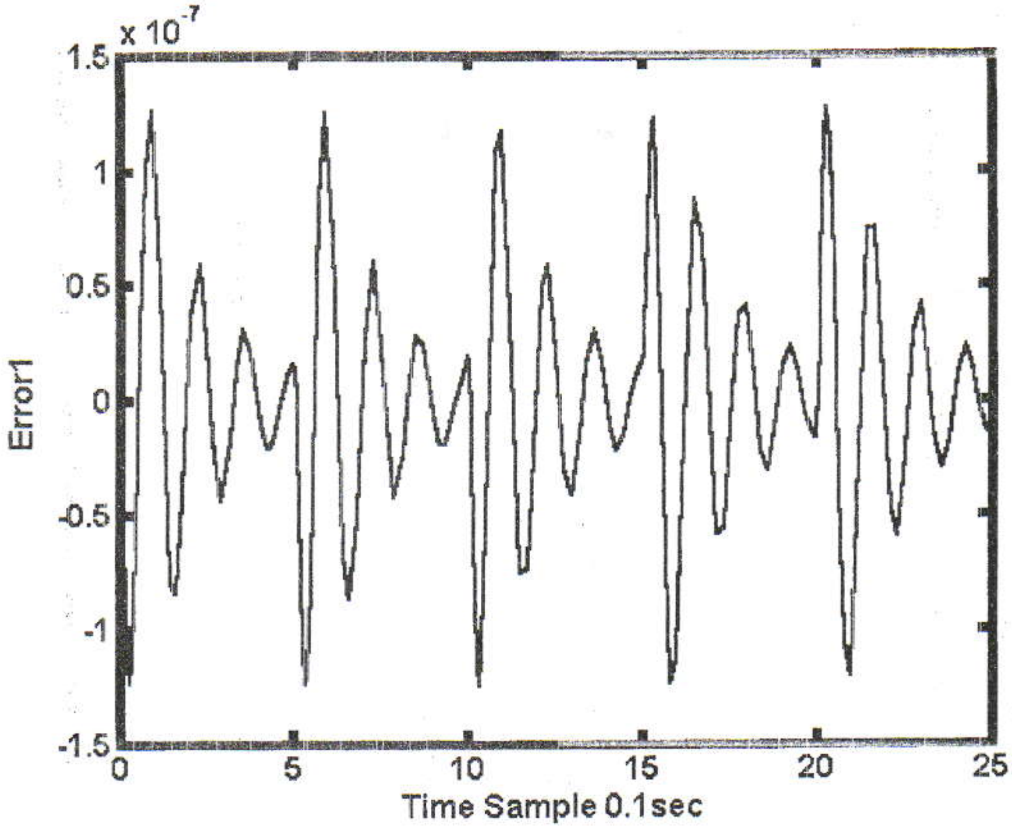


Fig (11): The Error between Set point & the Lateral Velocity

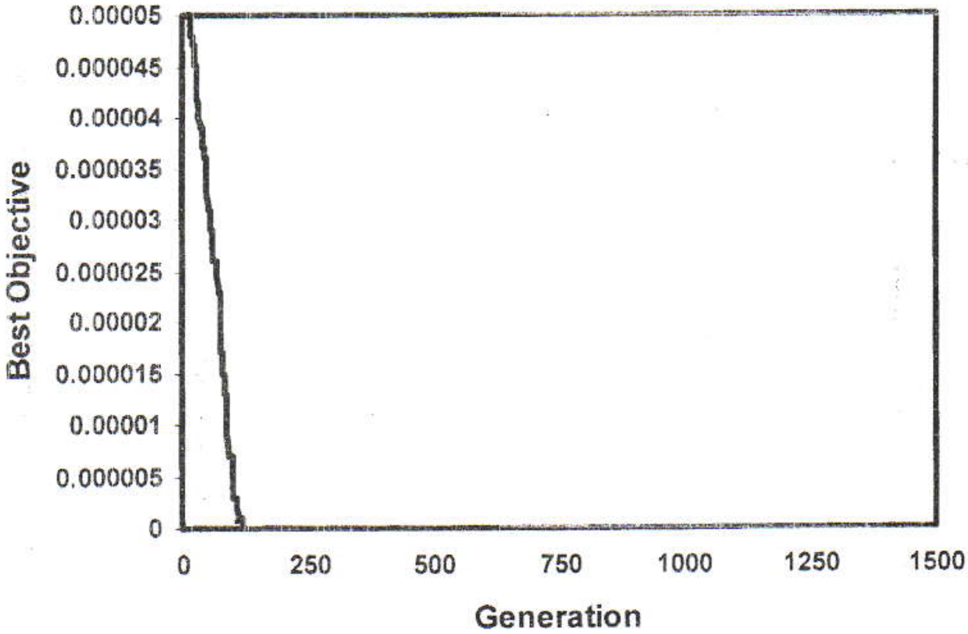


Fig (12): The best mean square error for the system MIMO