# Designing a Genetic Neural Controller of Differential Braking System for Vehicle Based on Model Reference

Mr. Ahmed Sabah Abdul Ameer Al-Araji

Received on: 5/ 12 /2006
Accepted on: 30 / 6 /2008

**Abstract**

In this paper, the structure of the controller is consists of a Modified Elman Neural Networks MENN model that is learned on-line by using genetic algorithm teachings in order to achieve required yaw rate and reduce lateral velocity in a short period of time to prevent vehicle from sliding out the curvature. By using differential braking system and front wheel steering angle has automatically controlled the vehicle lateral motion when the vehicle rotates the curvatures. The robust feedback neural controller is achieving the excellent transient state output of the system by minimizing the error between the model reference output and the model output of the system. Where the model of the system is also MENN that learned by two stages off-line and on-line, in order to guarantee that the model output accurately represents the actual output of the system by using dynamic Back Propagation Algorithm (BPA).

**الخلاصة**

لقد تم وضع في هذا البحث هيكلية مسيطر تتألف من الشبكة العصبية يلمـن المعدلـة MENN التـي تـتعلم ON-LINE باستخدام الخوارزمية الجينية لكي تحقق معدل الدوران المطلوب وتقليل السرعة الجانبية في اقصر وقت ممكن لمنـع انـزلاق المركبة خارج المنعطف باستخدام نظام الكبح ألفرقي وتوجيه العجلات الأمامية بصورة تلقائية نظامية للسيطرة على السـرعة الجانبية للمركبة ومعدل الدوران المطلوب أثناء دوران المركبة لمنعطف. أن المسيطر العصبي المحكم ذو التغذية الخلفيـة قـد حقق أفضل حالة عابرة لإخراج النظام من خلال تقليل الخطأ بين النموذج المرجع وإخراج الأنموذج العصبي للمنظومة.
أن النموذج العصبي للمنظومة هو أيضا" MENN المعلم بخطوتين OFF-LINE and ON-LINE لكي يضبط تماما" أخراج النظام الأصلي مع أخراج النموذج العصبي باستخدام خوارزمية الانتشار العكسي.

## 1-Introduction

The incorporation of vehicle stability enhancement systems into production vehicles is growing in popularity. The purpose of these systems is to actively control the vehicle under emergency situations where the car is at the physical limit of adhesion between the tires and the road. These emergency situations are those that the normal driver usually cannot handle, and often loses control of the vehicle [1]. A system, which automatically intervenes in such situations,

Control and System Eng. Dept. University of Technology

allows the driver to keep control of the vehicle and enhances the chance of avoiding an accident. Naito, et.al. [2] cite three points that an active safety system must address:

- A vehicle must provide good controllability by responding quickly and accurately (i.e. with the right amount of change) to the driver's operational inputs.
- A vehicle must provide good stability, with little change in behavior in relation to changes in driving conditions.
- There must be an effective control loop between the driver and the vehicle for conveying operational inputs and the vehicle response in order to ensure that the driver can easily recognize present operating conditions and also predict vehicle behavior.

There are various ways to address these control issues. H. & A. Sabah [3] investigate the use of independent front and rear wheel drive to control the vehicle to track a desired lateral velocity using fuzzy logic controller with variable gain structures. A. Sabah [4] used neural controller based genetic algorithm to control the vehicle to track a desired lateral velocity from front and rear steering angles. The most prevalent method is by combining the control of vehicle yaw rate and sideslip angle [5]. Summarizes the work on road vehicle motion control, with focus on different strategies to achieve synergetic effects for over-actuated systems [6]. A genetic neural fuzzy antilock brake system ABS controller is applied that consists of a non-derivative neural optimizer and fuzzy-logic components (FLC) as presented by Y.Z. [7]. It is used (ABS) senses when the wheel lockup is to occur, releases the brakes momentarily, and then reapplies the brakes when the wheel spins up again. ***The organization of this paper is as follows:*** Section two represents the 2 DOF vehicle mathematical model. Section three describes the use of feedforward neural networks to learn (Modified Elman Recurrent Neural Networks) as input-output model for system identification is examined with the corresponding neural nets and learning mechanism used for this purpose. Section four represents the core of the present paper, and it is suggested using feedback robust neural controller that will attain specific benefits towards a systematic engineering design procedure for neural control system. The proposed algorithm for the Robust Feedback Neural Controller type MENNs described by section five. Illustrative example, that clarify the features of the proposed strategy are given in sections two, three and four where an example is discussed in detail. Finally, section seven contains the conclusions of the entire work.

## 2- Two Degree of Freedom Vehicle Model

The non-linear 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 " four wheel steering system, braking forces, and torque driving wheel" [3,4].

In this paper, the focus is on the vehicle yaw rate and lateral velocity as the desired and the differential braking and front steering angle are the control action variables. The brake system is a challenging control problem because the vehicle-brake dynamics are highly non-linear with uncertain time-varying parameter. Intelligent controllers, such as neural or fuzzy, overcome these issues [8]. Neural controllers have the benefit of not requiring a mathematical model of the plant, while still being highly robust. Also, certain neural control designs to adapt themselves to improve its performance. Because of these features, neural controllers have been successfully implemented in the automotive field for controlling both wheel dynamics and vehicle dynamics [5].

The two DOF vehicle model as shown in figure (1) is widely used for lateral control design and has been shown to provide accurate response characteristics compared to more complex models for conditions up to 0.3 g lateral acceleration.

$$\begin{bmatrix} \dot{V} \\ \dot{r} \end{bmatrix} = A\begin{bmatrix} V \\ r \end{bmatrix} + B\begin{bmatrix} \delta_f \\ F_{BS} \end{bmatrix} \qquad (1)$$

$$\begin{bmatrix} y1 \\ y2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_f \\ F_{BS} \end{bmatrix}$$

where $y_1$=V and $y_2$= r

The linear dynamical model of vehicle lateral motion [9] with interaction in multi-input multi-output system are expressed as the state space equations where

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

$$B = \begin{bmatrix} \frac{C_f}{M} & 0 \\ \frac{aC_f}{I} & \frac{T}{2I} \end{bmatrix} \quad u = \begin{bmatrix} \delta_f \\ F_{BS} \end{bmatrix}$$

Equation (1) represents linear mathematical model of vehicle lateral motion with interaction in multiple-input-multiple-output system (see appendix 1) where V is lateral velocity, r is yaw rate and both are system variable states. $\delta_f, F_{Bs}$ represent front steering angle and brake steer force respectively and they are inputs to the system. Where the brake steer force can be described as the equation (2) from figure (1):

$$M_{BS} = \frac{T}{2}(F_{XR} - F_{XL})$$

(2)

$$F_{BS} = F_{XR} - F_{XL}$$

(3)

where:

$M_{BS}$ is brake steer moment.

$F_{XR}$ and $F_{XL}$ are front and rear longitudinal tire forces.

## 3- Identification of Dynamical System Using Neural Network Modeling

This section focuses on linear system identification using multi-layered feedforward Modified Elman Recurrent model neural network, The neural network is trained using Dynamic Back-Propagation Algorithm. A feedforward neural network can be seen as a system transforming a set of input patterns into a set of output patterns, and such a network can be trained to provide a desired response to a given input. The network achieves such a behavior by adapting its weights during the learning phase on the basis of some learning rules.

### 3-1 Structure of Modified Elman Neural Networks Model

The Elman neural network is a globally feed-forward locally recurrent networks model. It owns a set of context nodes to store the internal states. Thus, it has certain dynamical characteristcs over static neural networks, e.g., multilayer perceptrons and radial-basis function networks. However, its training and convergence speed are usually very slow and not suitable for time critical applications, such as on-line system identification and adaptive control. Hence, an improved Elman neural network, the modified Elman Neural Network (MENN), was recently propsed and applied successfully to dynamical system identification [10]. The structure of MENN is given in figure (2) is one of the simplest types that can be trained using dynamic BPA and it used to minimize the oscillation or even instabilities to the training controller. The output of the context unit in the modified Elman network is given by:

$$h_c^o(k) = \alpha h_c^o(k-1) + \beta h_c(k-1) \quad (4)$$

where $h_c^o(k)$ and $h_c(k)$ are respectively the output of the context unit and hidden unit and $\alpha$ is the feedback gain of the self-connections and $\beta$ is the connection weight from the hidden units (c'th) to the context units (c'th) at the context layer. The value of $\alpha$ and $\beta$ are selected randomly between (0 and 1). From figure (2) it can be seen that:

$$h(k) = F\{V1U(k), V2h^o(k)\}$$

(5)

$$O(k) = Wh(k)$$

(6)

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 hidden layers.

$V2$: Weight matrix of the context layers.

$W$: Weight matrix of the output layer.

$L$: Denotes linear node.

$H$: Denotes nonlinear node with sigmoidal function.

To explain these calculations, consider the general j'th neuron in the hidden layer shown in figure (3). The inputs 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 [11]:

$$net_j = \sum_{i=1}^{nh} V1_{j,i} \times U_i + \sum_{c=1}^{C} V2_{j,c} \times h_c^o \quad (7)$$

C and nh number of the context nodes and hidden nodes. For the standard design recurrent neural networks, the number of the context nodes is equal to hidden nodes nh=C, then c=j. Next the output of the neuron $h_j$ is calculated as the continuous sigmoid function of the $net_j$ as:

$$h_j = \mathrm{H}(net_j) \quad (8)$$

$$\mathrm{H}(net_j) = \frac{2}{1+e^{-net_j}} - 1 \quad (9)$$

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 (10)).

$$neto_k = \sum_{j=1}^{nh} W_{kj} \times h_j \quad (10)$$

Where $W_{kj}$ 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:

$$O_k = L(neto_k) \quad (11)$$

Thus the outputs at the output layer are yaw rate and lateral velocity which are denoted by O1, O2 respectively.

The learning (training) algorithm is usually based on the minimization (with respect to the network weights) of the following objective (cost) function MSE mean square of error as given in equation (12) by using dynamical back propagation algorithm with series-parallel configuration learning.

$$E = \frac{1}{P}\sum_{i=1}^{P}[(r^i(k) - y_{m1}^i(k))^2 + (v^i(k) - y_{m2}^i(k))^2] \quad (12)$$

where P is number of identification patterns (from 1 to 1000), $r^i$ is the yaw rate of the vehicle of each step and $y_{m1}^i$ is the model output of the plant of each step and $v^i$ is the lateral velocity of the vehicle of each step and $y_{m2}^i$ is the model output of the plant of each step.

## 4-Feedback Robust Neural Networks Controller

The control of linear dynamical system is considered in this section. The approach used to control the system depends on the information available about the system and the control objectives. The information of the unknown linear system can be known by the input-output data only and the system is considered as Modified Elman Neural Networks Model. The first step in the procedure of the control structure is the identification of the plant from the input-output data. 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 based on the minimization of the error between the model reference & the model 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 genetic algorithm 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 50 individuals, with each weight being between –1 and +1. The mutation operator adds a random value between –0.75 and +0.75 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:

$$MSE=\frac{1}{np}\sum_{k=1}^{np}[(y_{mr1}(k)-y_{n1}(k))^2+(y_{mr2}(k)-y_{n2}(k))^2] \quad (13)$$

where:

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

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

np is the number of the training desired patterns (1 to 50).

Since the GA maximizes its fitness function, it is necessary therefore to map the objective function (MSE) to a fitness function. The following objective fitness transformation [4, 11] is used.

$$fitness=\frac{1}{MSE+\mu} \quad (14)$$

Where $\mu$ is a constant chosen to avoid division by zero.

The final integrated control structure that consists of the reference model and neural network controller with the robustness of feedback can be shown in figure (5).

## 5-The Proposed Algorithm For The Robust Feedback Neural Controller Type MENNs

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 (14), where $\mu$ will be chosen as an input coefficient equal to 1.

Step 4: Put in descending orders all the chromosomes in the current population, according to this fitness level.

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.

## 6- Case Study

In this section, the vehicle parameters as given by appendix (2) is taken to clarify the features of the neural controller explained in sections three, four and applied the algorithm in section five. Scaling function has to be added at the neural network terminals to convert the scaled values to actual values "where the differential braking range is $\pm 5000N$ and front steering angle is $\pm 0.1$rad" and vice versa in order to overcome a numerical problem that is involved within real values. Therefore the signals entering to 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 given in appendix (3).

The training pattern (Np) was taken as 50 as the desired trajectory. The modified Elman recurrent neural networks are used to minimize the performance error between the reference model and the model output. The equation of the reference model for the two outputs is taken from [12] for more stability and without any oscillation in the response:

$$\begin{bmatrix} ymr1(k+1) \\ ymr2(k+1) \end{bmatrix} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} ymr1(k) \\ ymr2(k) \end{bmatrix} + \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} y1_{des}(k+1) \\ y2_{des}(k+1) \end{bmatrix}$$

(15)

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

$$\begin{bmatrix} emr1(k+1) \\ emr2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} ymr1(k+1) \\ ymr2(k+1) \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{m1}(k+1) \\ y_{m2}(k+1) \end{bmatrix}$$

(16)

where $emr1(k)$ and $emr2(k)$ is the reference model error outputs.

The performance of the proposed controller is evaluated using the closed-loop step lateral velocity and yaw rate responses for linear system. 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:

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

(17)

where $R$ is curvature radius.

*Case-1:*

For off-line identification with series-parallel configuration a model described by MENN as shown in figure (2) where six nodes in the single hidden layer and six nodes in context layer.

BPA with learning rate $\eta = 0.1$ and the input-output patterns (1000) as a learning set, then after 3500 epochs for the Pseudo-Random Binary Signal (PRBS) inputs as shown in figure (6-a). The mean square of error (MSE) is equal to $1.35 \times 10^{-6}$. After training the weights of the hidden layer, context layer and output layer can be shown in appendix (4).

Figure (6-b and c) compares the time response of the model with the actual plant output for the random input as learning set. Step change variable inputs as shown in figure (7-a) as testing set and figure (7-b and c) compares the time response of the model with the actual plant output respectively.

An on-line updating of the weights of the neural network will be carried out to ensure the output of the model will be equal to that of the plant, so that calculation of the feedback control action will be fairly accurate.

The neural controller model described by MENN as shown in figure (4) where three nodes in the single hidden layer and three nodes in context layer.

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 figures (8-a & b), and the weights of the hidden layer, context layer and output layer can be shown in appendix (5).

Figure (8-a) is the lateral velocity response and its fast response with very small overshoot $\pm 2.5\times10^{-7}$. Steady-state error is equal to zero and the transient time is approximately equal to 0.1 sec when the vehicle velocity is change as (15, 25, 35) m/sec with fixed curvature radius equal to 100m.

Figure (8-b) is the yaw rate response and its fast response with no overshoot. Steady-state error is equal to zero and the transient time is approximately equal to 0.05 sec when the vehicle velocity is change as (15, 25, 35) m/sec with fixed curvature radius equal to 100m.

The robustness of feedback neural control action will be kept the maximum amplitude of the lateral velocity in the transient response is equal to ($\pm 2.5\times10^{-7}$ 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 brake steer force "differential braking" and front steering angle as shown in figures (8-c & d).

The error between the two desired outputs and the two actual outputs of the system is very small as shown in figure (8-e& f).

*Case 2:*

When applying random input as a disturbance $\delta_{dis}$ that has maximum magnitude ($2\times10^{-3}$ rad) as shown in figure (9-a). The response of the yaw rate of the vehicle is not drifted from the desired and it has very small overshoot as shown in figure (9-b) and also the lateral velocity of the vehicle is very small oscillation magnitude $\pm4\times10^{-3}$ as shown in figure (9-c). The differential braking and the front steering angle can be shown in figures (9-d & e). And the error between the desired yaw rate and actual output for this case can be shown in figure (9-f & j).

Figure (10) is described the best objective function ***MSE*** for the MIMO system.

## 7-Conclusions

The results of lateral motion simulation show that the controlling on lateral velocity and yaw rate by robust neural controller proposed have been achieved at different velocity of the vehicle by successfully simulated to multi-input multi-output linear system. That is to be completed by controlling front wheel steering and differential braking system. The structure of the controller is MENNs as an adaptive controller with genetic algorithm learned. Interviewing the results of simulations, the side velocity is fluctuating around the desired during the transient response. That means occur lateral motion slightly. The vehicle moves from its own path to lateral path. Thus later position should be taken as variable state that must be controlled to improve vehicle rotation stability. For reason above, it is urgent need for developing vehicle mathematical model of lateral motion in order to appear a state of each tire independently. It is also to be taken into consideration the other variables input in vehicle lateral motion control system such as increasing or decreasing the front steering angle as a disturbance effects to make lateral motion more stability by robust feedback neural networks.

## Appendix (1)

### *Nomenclature*

a= distance from the center of mass to front axle

b= distance from the center of mass to rear axle

T=vehicle track

$C_i$= 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

$\delta_f$ = front steering angle

$F_{BS}$ =brake steer force

$\delta_{dis}$= front steering angle disturbance

$M_{BS}$ is brake steer moment.

$F_{XR}$ and $F_{XL}$ are front and rear longitudinal tire forces.

## Appendix (2)

### *Vehicle nominal parameters*

M=1000Kg

a=1m

b=1.5m

T=1.5m

I=1500Kg m$^2$

$C_f$ =55000 N/rad

$C_r$= 45000 N/rad

U= 15, 25 & 35 m/sec

R=100m

## Appendix (3)

### *The Real-Coded Genetic Algorithm*

Population size ($N_{POP}$) is equal to 50.

Crossover Probability (Pc) is equal to 0.8.

Mutation Probability (Pm) is equal to 0.05.

Maximum number of generations is 2000.

## endix (4)

### *The weights of the identifier model MENN*

$$V1_{JJ} = \begin{matrix} 0.933 & -3.22 & 2.37 & 0.355 \\ 1.722 & 0.461 & 1.43 & 0.212 \\ -2.11 & 0.638 & -1.201 & 0.393 \\ -0.343 & 0.711 & 0.657 & -0.91 \\ 1.022 & -0.439 & 0.78 & -0.317 \end{matrix}$$

$$
= \begin{matrix} -0.17 & 0.345 & 0,788 & 1.09 & 3.22 & 0.29 \\ -022 & 0.57 & -0.01 & 0.51 & -0.102 & -1.67 \\ -0.389 & 0.121 & 0.023 & 0.571 & 2.77 & 0.748 \\ 0.45 & 0.023 & 0.34 & -0.25 & 0.45 & 0.754 \\ 0.731 & 0.331 & 1.05 & -0.54 & 0.7 & 0.387 \\ 0.671 & -0.27 & 0.37 & -0.366 & -0.357 & -0.33 \end{matrix}
$$

$$
W_{k,j} = \begin{matrix} 0.719 & -0.521 & 0.389 & 0.223 & 0.711 & -0.103 \\ -0.02 & -1.89 & 0.374 & 0.451 & 0.331 & -0.023 \end{matrix}
$$

## Appendix (5)

*The weights of the controller model MENN*

$$
V1_{j,i} = \begin{matrix} 0.210 & -0.345 & -0.531 & 0.71 \\ -0.329 & -0.703 & -0.539 & 0.662 \\ 0.567 & 0.724 & 0.579 & 0.119 \end{matrix}
$$

$$
V2_{j,c} = \begin{matrix} -0.439 & 0.665 & 0.102 \\ -0.19 & -0.331 & -0.349 \\ 0.639 & 0.711 & 0.306 \end{matrix}
$$

$$
W_{k,j} = \begin{matrix} 0.711 & -0.541 & 0.379 \\ 0.662 & 0.103 & -0.709 \end{matrix}
$$

## Reference


[1]- A.T.van Zanten, R. Erhaedt, & G. Pfaff. VDC, The vehicle dynamics control system of Bosch. SAE International Congress and Exposition, 1995.

[2]- A.T.van Zanten, R. Erhaedt, K. Landesfeind & G. Pfaff. VDC, Systems development and perspective. SAE World Congress, 1998.

[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]- A. S. Al-Araji "Design of a neuro-controller for vehicle lateral velocity and yaw rate based genetic algorithm with model reference guided" Iraqi journal of computers, communication, control and systems engineering by University of Technology Vol. 6, No.1, pp. 30-42, 2006.

[5]-K. R. Buckholtz " Use of fuzzy logic in wheel slip assignment-part I: yaw rate control" SAE 2002 world congress. March 4-7,2002.

[6]- J. Andreasson, C. Knobel, & T. Bunte "On road vehicle motion control striving towards synergy" The 8th international symposium on advanced vehicle control, August 20-24, 2006.

[7]- Y. Lee and H. Zak" Designing a genetic neural fuzzy antilock-brake-system cointroller" IEEE transactions on evolutionary computation, Vol.6, No.2, April 2002.

[8]-S. K. Mazumdar & C.C. Lin. Investigation of the use of neural networks for antiskid brake system design" Proceedings of the IEEE international conference on neural networks, 1995.

[9]-T. Pilutti, G. Ulsoy, & D. Hrovat "Vehicle steering intervention through differential braking" Journal of dynamic systems, measurement, and control. Vol. 120, September 1998.

[10]-X.M. Gao, X.Z. Goa, J.M.A. Tanskanen, and S.J. Ovaska" Power control for mobile DS/CDMA systems using a modified MLMAN neural network controller "IEEE vehicular technology conference, pp 750-754. May 1997.

[11]- A. S. Al-Araji "Genetic algorithm and Elman neural network used for tuning the parameters of the PID neural controller based model reference" Iraqi journal of computers, communication, control and systems engineering by University of Technology Vol. 5, No.1, pp. 111-126, 2005.

[12]- Ahmed S. Al-Araji" A Neural controller with a pre-assigned performance index" M.Sc. Thesis, University of technology, November 2000.

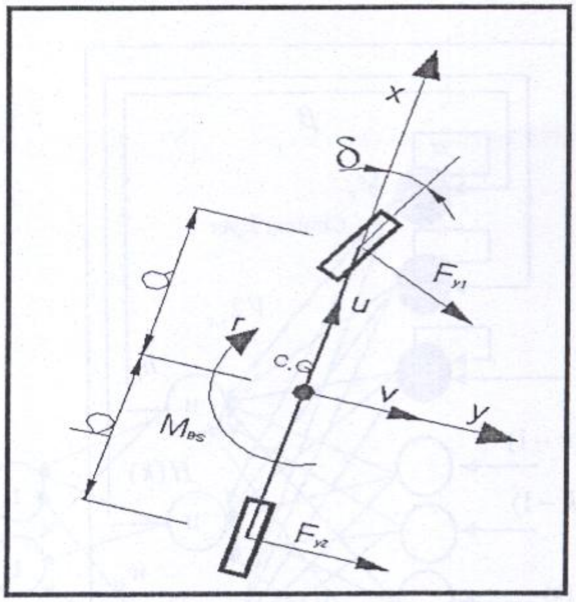



**Fig (1): Two DOF Vehicle Model**

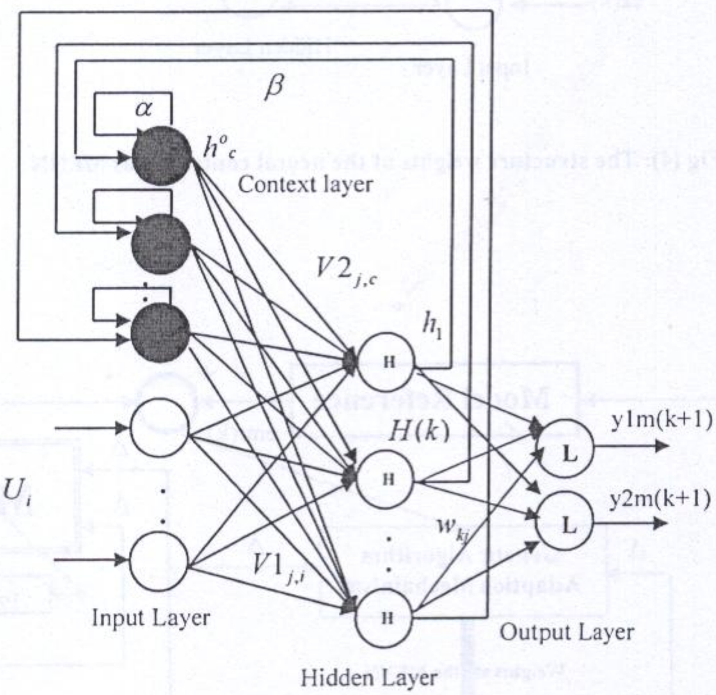



**Fig (2): The Modified Elman Recurrent Neural Networks**

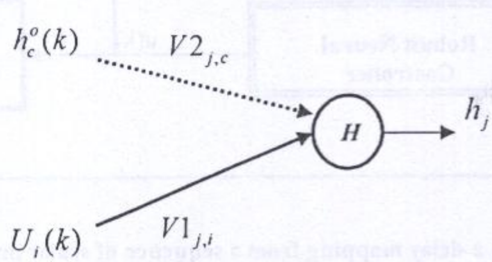



**Fig (3): Neuron j in the hidden layer.**

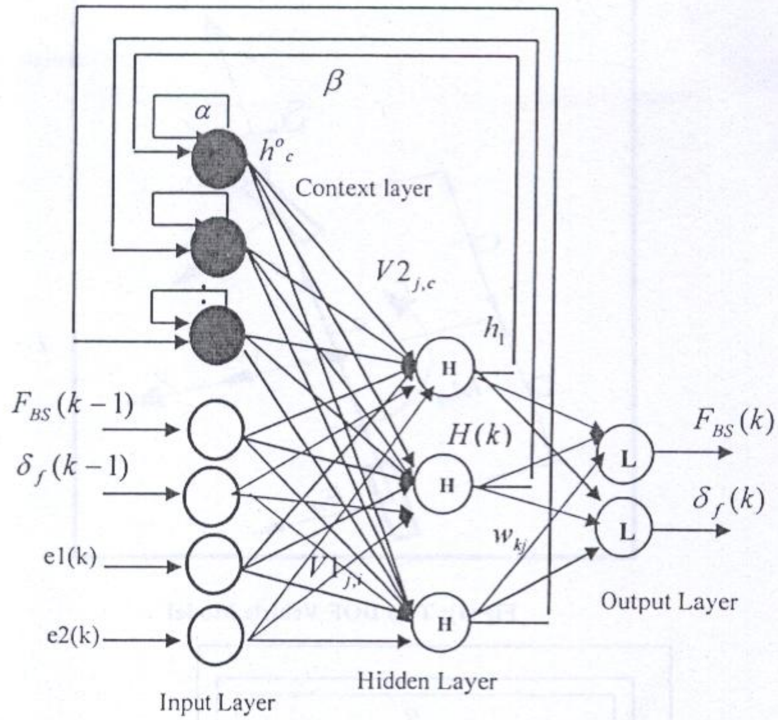
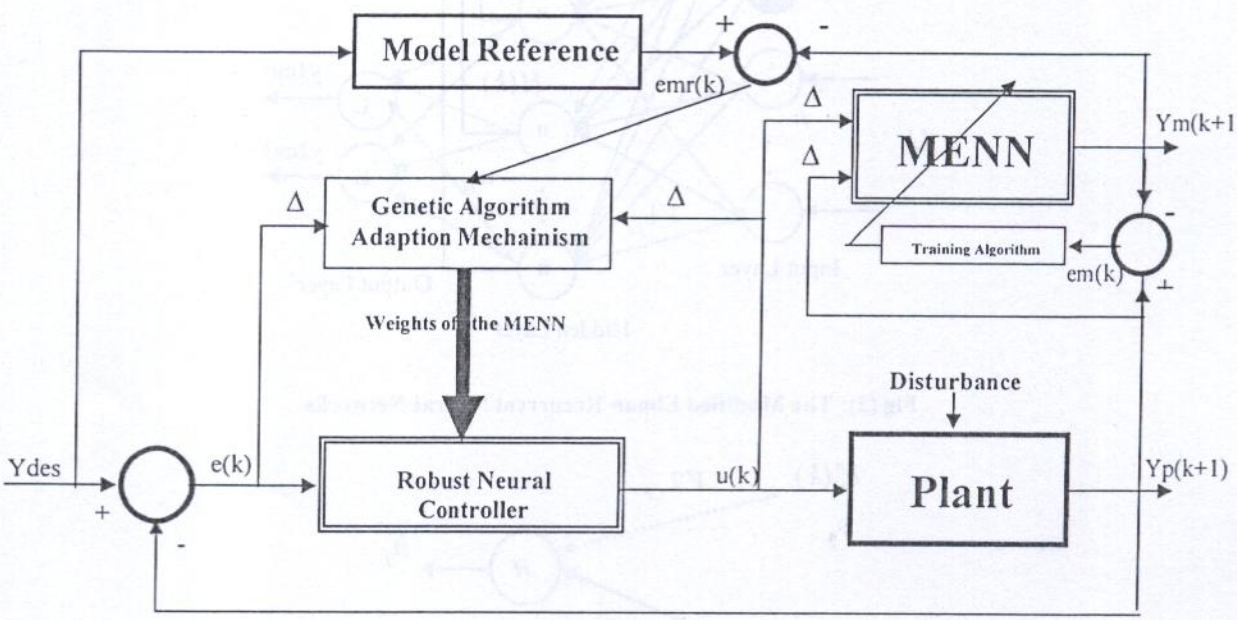



**Fig (4): The structure weights of the neural controller as MENN**




**$\Delta$ is defined as a delay mapping from a sequence of scalar inputs and outputs**

**Fig (5): The General Proposed of the Robust Neural feedback Controller**

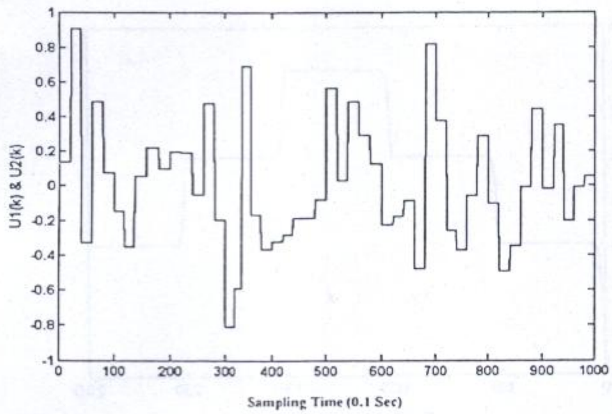



Fig (6-a): The PRBS inputs signal used to excite the plant

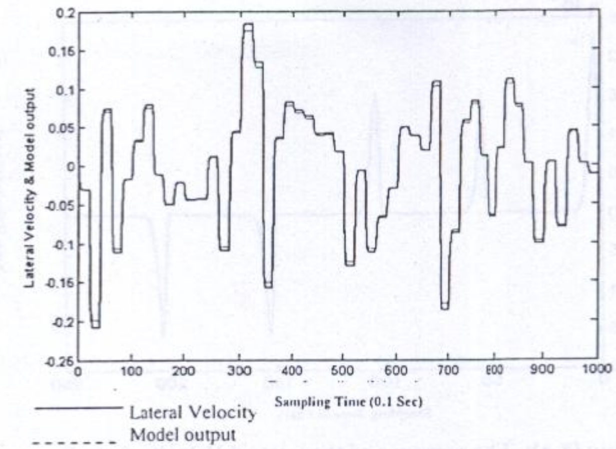



Fig (6-b): The response of the Lateral Velocity & Model output for the learning patterns

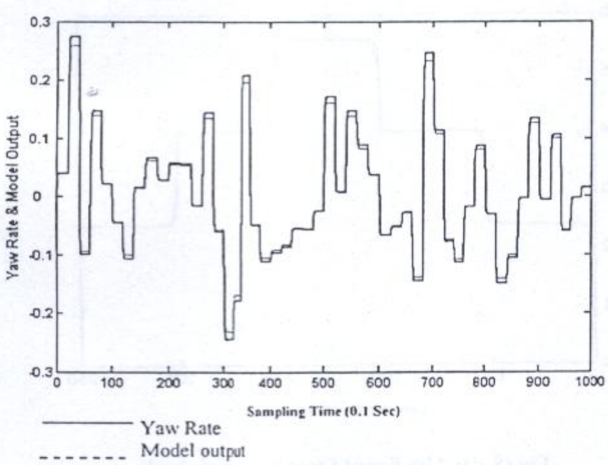



Fig (6-c): The response of the Yaw Rate & Model output for the learning patterns

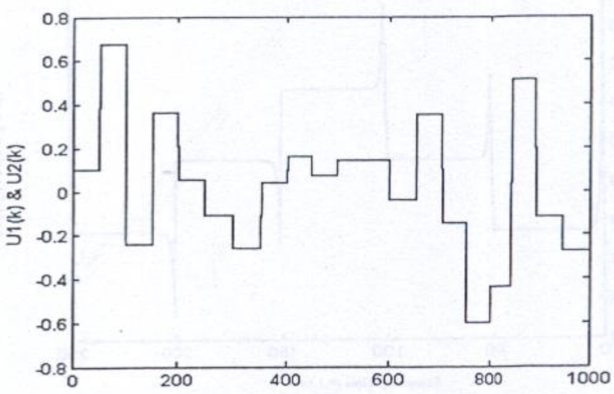



Fig (7-a): The step change inputs signal used to test Model

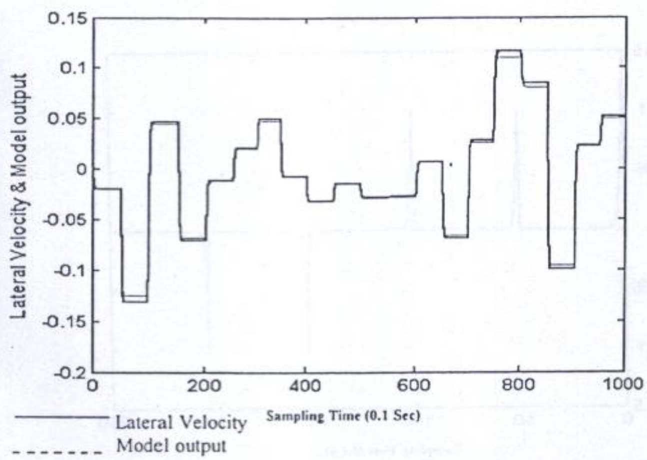



Fig (7-b): The response of the Lateral Velocity & Model output for the testing patterns

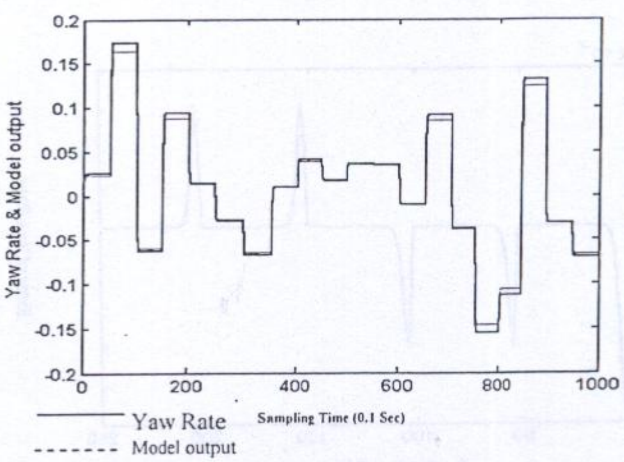



Fig (7-c): The response of the Yaw Rate & Model output for the testing patterns

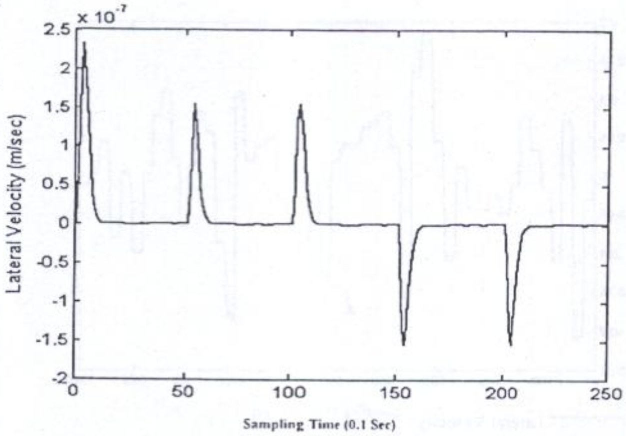



Fig (8-a): The response of the Lateral Velocity (m/sec)

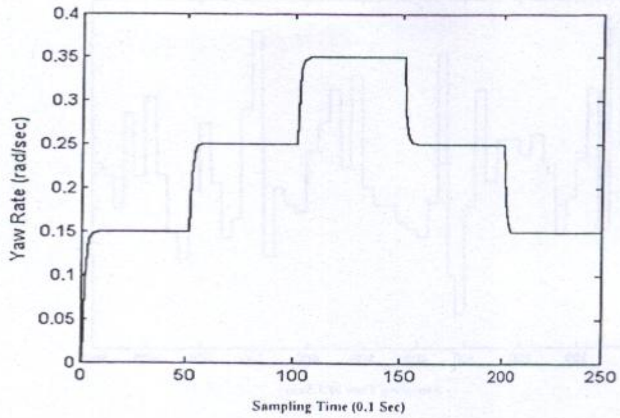



Fig (8-b): The response of the Yaw Rate (rad/sec)

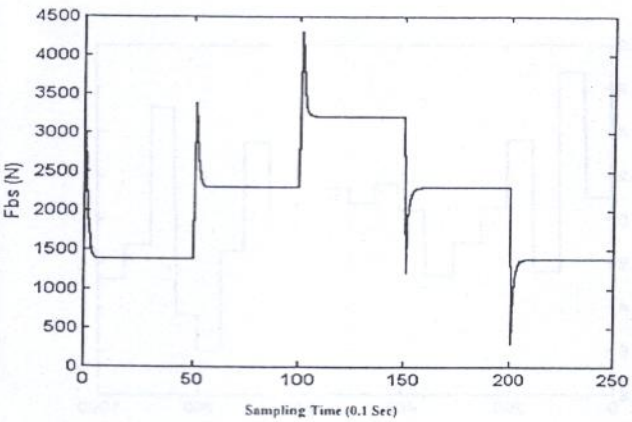



Fig (8-c): The Differential Braking (N)

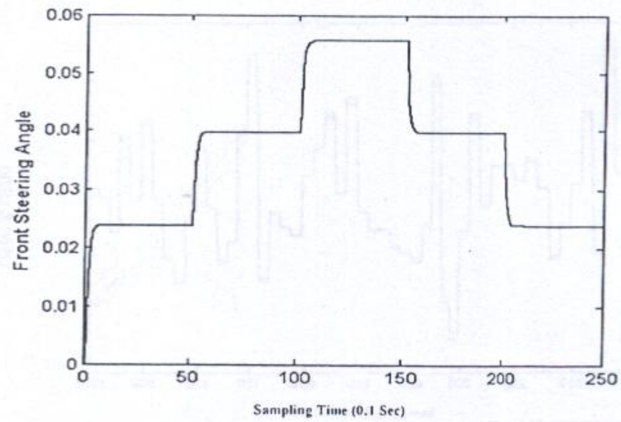



Fig (8-d): The Front Steering Angle (rad)

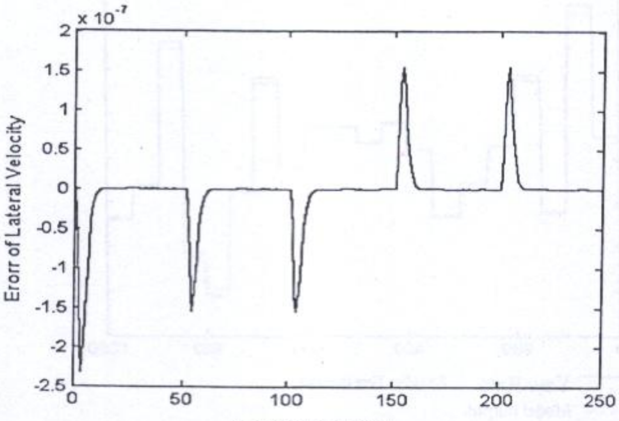



Fig (8-e): The Error between the Lateral Velocity & the Model Reference

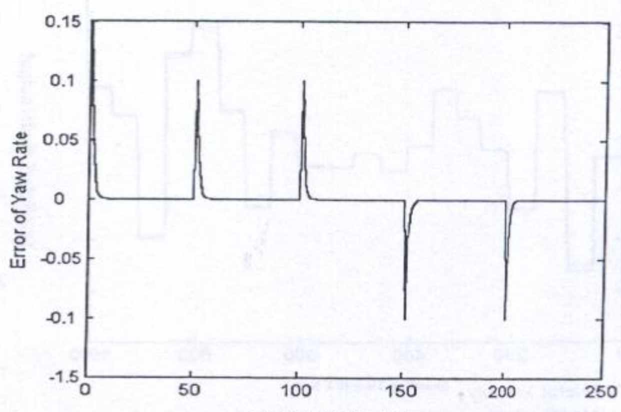



Fig (8-f): The Error between the Yaw Rate & the Model Reference

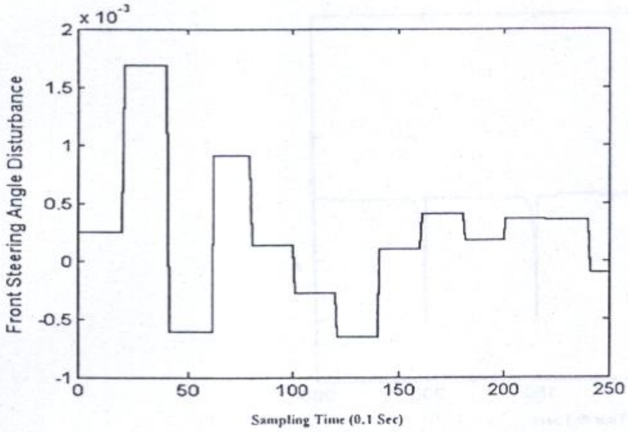



Fig (9-a): The step change disturbance input to the plant

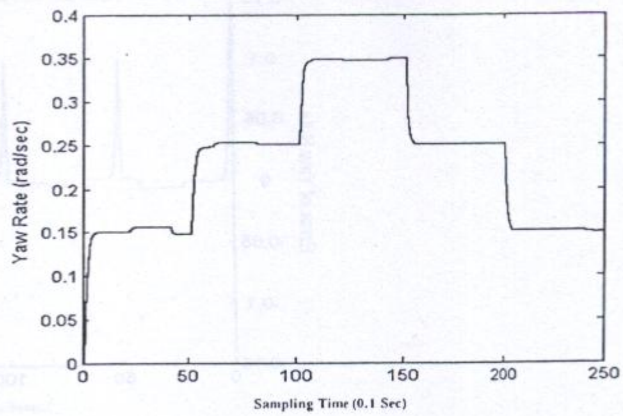



Fig (9-b): The response of the Yaw Rate (rad/sec)

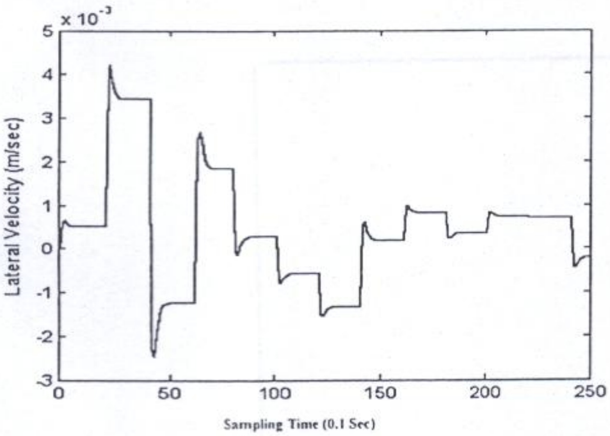



Fig (9-c): The response of the Lateral Velocity (m/sec)

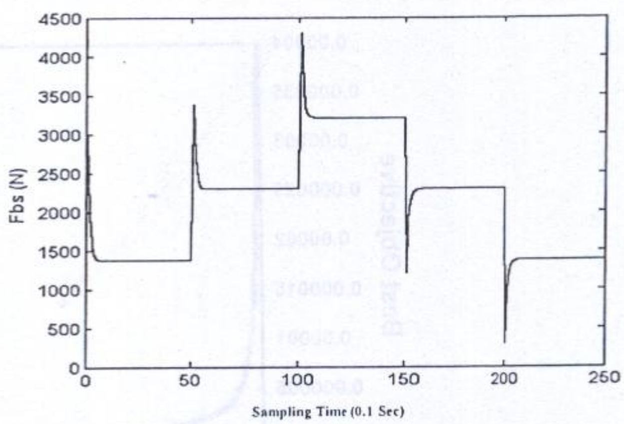



Fig (9-d): The Differential Braking (N)

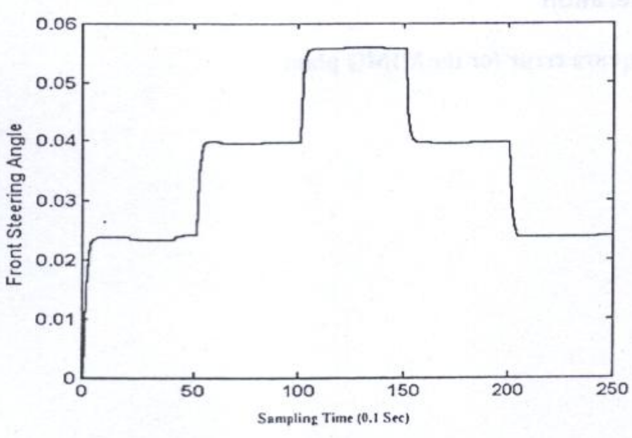



Fig (9-e): The Front Steering Angle (rad)

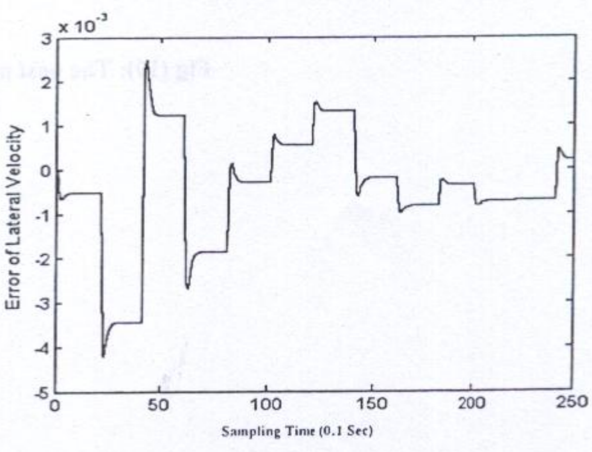



Fig (9-f): The Error between the Lateral Velocity & the Model Reference

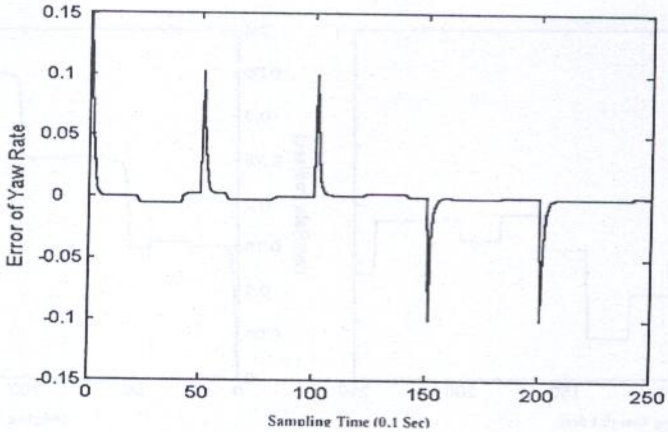



**Fig (9-g): The Error between the Yaw Rate & the Model Reference**

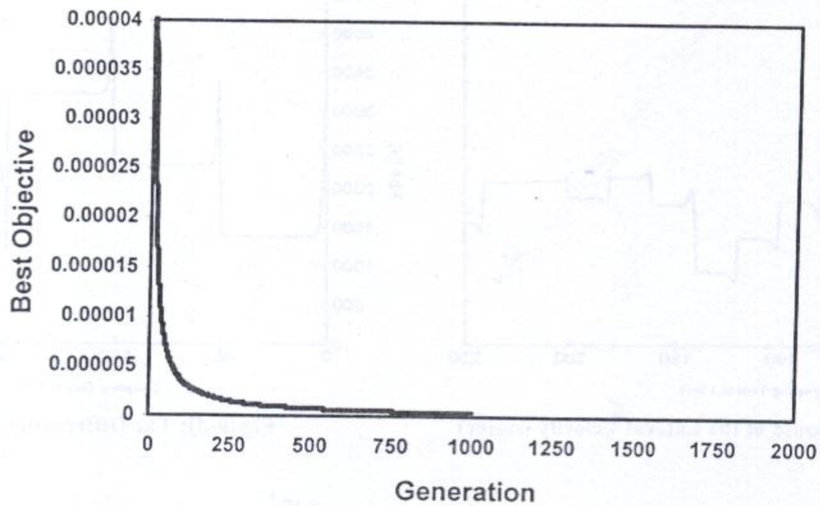



**Fig (10): The best mean square error for the MIMO plant**