Dynamic Neural Controller with Somatic Adaptation

D.H. Rao and M.M. Gupta

Intelligent Systems Research Laboratory, College of Engineering
University of Saskatchewan, Saskatoon, Canada, S7N 0W0

Abstract- In the application of neural networks to control systems the nonlinear function, usually sigmoidal, is kept constant, and the gain of sigmoidal function is determined by trial and error technique. The heuristic selection of the gain, which determines the shape, of sigmoidal function and keeping it constant may limit the application of neural networks to complex systems involving nonlinear dynamics. In this paper we propose a neural structure which comprises of dynamic neural units with time-varying sigmoidal functions. The effect of sigmoidal gain on nonlinear dynamic systems is discussed. The learning and adaptive algorithm to determine the optimum sigmoidal gain, which results in selftuning of the neuron, is derived. The effectiveness of the proposed neural network is demonstrated through computer simulation studies.

1. Introduction

A nonlinear activation function, usually sigmoidal, is used in artificial neural networks to model the intercellular current conduction mechanism in biological neuron. It is currently understood that the biological neuron provides two distinct operations distributed over the *synapse*, the junction point between an axon and the dendrite, and the *soma*, the main body of the neuron. These two neuronal operations may be called (i) the *synaptic operation*, and (ii) the *somatic operation*. From the biological point of view, these two operations are physically separate, but, in the modeling of a biological neuron, these operations have been combined (for example, thresholding in the soma is transferred to the synaptic operation). Furthermore, in the biological neuron, there is a time-varying nonlinear relationship between the pulse rate at the synapse and the amplitude of the dendritic current [1]. This leads to a plausible inference that the main body of the neuron, the soma, may also be changing during neural activities, such as learning, adaptation, and vision perception.

However, the optimum gain of a nonlinear activation function in artificial neural networks is determined by trial and error technique. This heuristic selection of the gain of sigmoidal function may limit the application of neural networks to complex systems involving nonlinear dynamics. Because, an improper selection of sigmoidal gain may result in an undesirable or even an unstable response. Recently, Yamada and Yabuta have considered determining the optimum shape of the sigmoidal function and have applied to linear and simple nonlinear systems [2]. Independently, it was proposed in [3,4] that the parameter which controls the shape, namely the gain, of the nonlinear function can be considered as one of the adjustable parameters of the neural structure in addition to the synaptic weights. This component contributes to what is referred to as *somatic adaptation.*

In this paper, a three-stage neural network is developed using a *dynamic neural unit* (DNU), proposed by the authors in [5], as the basic computing element. A brief description of the DNU is given in the next section, followed by the learning and adaptive algorithm to modify the parameters of DNU in Section 3. A three-stage dynamic neural network is formed using DNU as the basic functional node in Section 4. The effectiveness of the proposed neural network structure is demonstrated through computer simulation results in Section 5, followed by conclusions in the last section.

2. The DNU: The Basic Neural Computing Element

The authors have proposed a different architecture to model the biological neuron, named the *dynamic neural unit* (DNU), whose structure is analogous to that of the reverberating circuit in a neuronal pool of the central nervous system [3,4]. The topology of the DNU embodies delay elements with feedforward and feedback synaptic weights, and a time-varying nonlinear activation function. The DNU performs two basic operations: (i) the *synaptic operation* which enables to determine the optimum synaptic weights for dynamic operations, and (ii) the *somatic operation* which determines an optimum gain of the nonlinear function for a given task.

The dynamic structure of the DNU consists of two delay elements and two feedforward and feedback paths weighted by the synaptic weights $\mathbf{a}_{ff}$ and $\mathbf{b}_{fb}$ respectively. This is a second-order dynamic structure which can be described by the following difference equation

$$v_1(k) = -b_1\, v_1(k-1) - b_2\, v_1 k-2) + a_0\, x(k) + a_1 x(k-1) + a_2\, x(k-2) \qquad (1)$$

where $x(k) \in R^n$ is the neural input vector, $v_1(k) \in R^1$ is the output of the dynamic structure, $u(k) \in R^1$ is the neural output, k is the discrete-time index, and $\mathbf{a}_{ff} = [a_0, a_1, a_2]$ and $\mathbf{b}_{fb} = [b_1, b_2]$ are the vectors of adaptable feedforward and feedback weights respectively. The vectors of signals and adaptable weights of the dynamic neuron are defined now as follows:

0-7803-0999-5/93/$03.00

$$\gamma(k,v_1,x) = [v_1(k-1)\ \ v_1(k-2)\ \ x(k)\ \ x(k-1)\ \ x(k-2)]^T, \quad (2)$$

and

$$\zeta_{(a_{ff}, b_{fb})} = \begin{bmatrix} -b_1 & -b_2 & a_0 & a_1 & a_2 \end{bmatrix}^T \qquad (3)$$

(where the superscript T denotes transpose).

Using (2) and (3), Eqn. (1) is rewritten as:

$$v_1(k) = \gamma^T(k,v_1,x)\, \zeta_{(a_{ff}, b_{fb})}. \qquad (4)$$

The nonlinear mapping operation on $v_1(k)$ yields a neural output u(k) given by

$$u(k) = \Psi[g_s\, v_1(k)] \qquad (5)$$

where $\Psi[.]$ is some nonlinear activation function, usually a sigmoidal function, and g_s is the parameter, called the somatic gain, which controls the slope of the nonlinear function. Any function $\Psi[.]$ is said to belong to the class of sigmoidal functions, if (a) $\Psi[v(k)]$ is a monotonically increasing function of v(k) in the interval $(-\infty, \infty)$, (b) $\Psi[v(k)]$ approaches or attains the asymptotic values 0 and 1 as v(k) approaches $-\infty$ and ∞ respectively, and (c) $\Psi[v(k)]$ has one and only one inflection point [2]. Mathematically, the sigmoidal function is: $\lim_{v(k)\to-\infty} \Psi[v(k)] = 0$, and $\lim_{v(k)\to\infty} \Psi[v(k)] = 1$. If the mathematical operations are to be extended to both the positive and negative neural outputs, thereby extending the neural activity to both the excitatory and inhibitory inputs, the activation sigmoidal function may be redefined as a hyperbolic tangent function given by

$$\Psi[v(k)] = \tanh\,[g_s\, v_1(k)] = \tanh\,[v(k)] \qquad (6)$$

where $v = g_s\, v_1(k)$. Fig. 1 shows $\Psi[.]$ and its derivative $\Psi'[.]$ which provides the axonal gain for different values of slope. As mentioned above, the adjustable parameters of the DNU are the feedforward and feedback weights, $\mathbf{a}_{ff}$, $\mathbf{b}_{fb}$ and the somatic gain g_s. The algorithm to modify these parameters is derived in the following section.

3. Learning and Adaptive Algorithm

The learning process involves the determination of feedforward and feedback weights, and somatic gain which minimize the error function in some optimal fashion. In an iterative learning scheme, the control sequence is modified in each learning iteration to make the neural output u(k) approach the desired state $u_d(k)$. The components of the parameter vector $\Omega_{(a_{ff}, b_{fb}, g_s)}$ and error e(k) vary with every learning trial k. As the number of learning trials increase, the information set reduces to only $\{\Omega^*_{(a_{ff}, b_{fb}, g_s)}(k), e^*(k)\}$ which indicates that the DNU parameters and the error have converged to the optimal values (not necessarily global). To achieve this a performance index, which has to be optimized with respect to the parameter vector, is defined as

$$J = \frac{1}{2} E\left\{ e^2(k;\Omega_{(a_{ff}, b_{fb}, g_s)}) \right\}. \qquad (7)$$

where E is the expectation operator. Each component of the vector $\Omega_{(a_{ff}, b_{fb}, g_s)}$ is adapted to minimize J based on the steepest-descent algorithm which may be written as

$$\Omega_{(a_{ff}, b_{fb}, g_s)}(k+1) = \Omega_{(a_{ff}, b_{fb}, g_s)}(k) + \delta\Omega_{(a_{ff}, b_{fb}, g_s)}(k) \qquad (8)$$

where $\Omega_{(a_{ff}, b_{fb}, g_s)}(k+1)$ is the new parameter vector, $\Omega_{(a_{ff}, b_{fb}, g_s)}(k)$ is the present parameter vector, and $\delta\Omega_{(a_{ff}, b_{fb}, g_s)}(k)$ is an adaptive adjustment in the parameter vector. In the steepest-descent method, the adjustment of the parameter vector is made proportional to the negative of the gradient of the performance index J, that is,

$$\delta\Omega_{(a_{ff}, b_{fb}, g_s)}(k) \propto (-\nabla J) \text{ where } \nabla J = \frac{\partial J}{\partial \Omega_{(a_{ff}, b_{fb}, g_s)}}$$

therefore,

$$\delta\Omega_{(a_{ff}, b_{fb}, g_s)}(k) = -\,\text{dia}[\mu]\, \frac{\partial J}{\partial \Omega_{(a_{ff}, b_{fb}, g_s)}} = -\,\text{dia}[\mu]\, \nabla J \qquad (9)$$

where dia[μ] is the matrix of individual adaptive gains. In the above equation, the dia[μ] and the constraint on the parameter vector are respectively

$$\text{dia}[\mu] = \begin{bmatrix} \mu_{a_i} & 0 & 0 \\ 0 & \mu_{b_j} & 0 \\ 0 & 0 & \mu_{g_s} \end{bmatrix}, \text{ and}$$

$$\| \Omega_{(a_{ff}, b_{fb}, g_s)}(k) \| = \left| \sum_{i=1}^{n} \sqrt{\Omega^2_{i\,(a_{ff}, b_{fb}, g_s)}(k)} \right| = 1 \tag{10}$$

where μ_{a_i}, i = 0,1,2, and μ_{b_j}, j =1,2, are the individual gains of the adaptable parameters of DNU which determine the stability and the speed of convergence to the optimal values, and $\|.\|$ represents the norm of $\Omega_{(a_{ff}, b_{fb}, g_s)}(k)$. Let us represent the synaptic weight vector as $\phi_{(a_{ff}, b_{fb})}$. The gradient of performance index with respect to $\phi_{(a_{ff}, b_{fb})}$ is then given by:

$$\frac{\partial J}{\partial \phi_{(a_{ff}, b_{fb})}} = \frac{1}{2} E\left[\frac{\partial [u_d(k) - u(k)]^2}{\partial \phi_{(a_{ff}, b_{fb})}}\right]$$

$$= E\left\{-e(k)\left[\frac{\partial \Psi(v)}{\partial v} \frac{\partial v}{\partial \phi_{(a_{ff}, b_{fb})}}\right]\right\} \text{ (by chain rule)}$$

$$= E\left\{-e(k)\left[g_s \operatorname{sech}^2[v_1(k)]\, s\phi_{a_{ff_i}}(k)\right]\right\} \tag{11}$$

where $s\phi_{a_{ff}, b_{fb}}(k) = \dfrac{\partial v(k)}{\partial \phi_{(a_{ff}, b_{fb})}} = g_s \dfrac{\partial v_1(k)}{\partial \phi_{(a_{ff}, b_{fb})}}$

represents a vector of parameter-state (or sensitivity) signals. The parameter-state signals for feedforward and feedback weights are given by the relations respectively:

$$s\phi_{a_{ff_i}}(k) = g_s\, [x\,(k - i)], \quad i = 0, 1, 2, \text{ and}$$

$$s\phi_{b_{fb_j}}(k) = -\, g_s\, [v_1\,(k - j)], \quad j = 1,2. \tag{12}$$

The proof of Eqn. (12) is given in [4]. Similarly, the gradient of performance index with respect to somatic gain (slope) g_s, $\frac{\partial J}{\partial g_s}$, is given by

$$\frac{\partial J}{\partial g_s} = \frac{1}{2} E\left[\frac{\partial [u_d(k) - u(k)]^2}{\partial g_s}\right]$$

$$= E\left\{ -e(k)\left[g_s \operatorname{sech}^2[v_1(k)]\, v_1(k)\right] \right\}. \tag{13}$$

From the above equations, the algorithm to update the DNU parameters can be written as

$$a_{ff_i}(k+1) = a_{ff_i}(k) + \mu_{a_i} E\left\{e(k)\, g_s \operatorname{sech}^2[v_1(k)]\, s\phi_{a_{ff_i}}(k)\right\}$$

$$, \; i = 0,1,2, \tag{14a}$$

$$b_{fb_j}(k+1) = b_{fb_j}(k) + \mu_{b_j} E\left\{e(k)\, g_s \operatorname{sech}^2[v_1(k)]\, s\phi_{b_{fb_j}}(k)\right\}$$

$$, \; j = 1,2, \text{ and} \tag{14b}$$

$$g_s(k+1) = g_s(k)\left[1 + \mu_{g_s} E\left\{e(k) \operatorname{ech}^2[v_1(k)]\, v_1(k)\right\}\right]. \tag{15}$$

Equations (14a) and (14b) refer to synaptic operation, while (15) to somatic operation of the DNU. The detailed algorithm derivation and its implementation may be found in [4].

4. Multi-Layered Dynamic Neural Network

Multi-layered networks may be formed by simply cascading a group of single layers, the output of one layer provides the input to the subsequent layer. In this section, a multi-layer dynamic neural network is developed by considering the DNU as the basic computing node. Let the output of a DNU be written as

$$\omega(x) = \Psi\left[g_s\left(w(k, a_{ff}, b_{fb})\right)x(k)\right] \tag{16}$$

where Ψ [.] is a sigmoidal function with varying slope g_s, and the terms in square brackets form an argument of Ψ [.]. A multi-stage dynamic neural network can be formed by cascading a group of single stage (layer) DNUs while the output of one stage provides the input to the subsequent stages. In this paper, the controller is configured to have a three-stage neural network consisting of an input-stage, an intermediate-stage and an output-stage which produces a control output, u(k), as

$$u(k) = u_{31}(k) + u_{32}(k) + u_{33}(k) \tag{17}$$

where $u_{31}(k)$, $u_{32}(k)$ and $u_{33}(k)$) are the outputs of the dynamic neural units $w^{(31)}(.)$, $w^{(32)}(.)$ and $w^{(33)}(.)$, respectively, and are given by

$$u_{31}(k) = \Psi\left[\left\{g_s^{(31)} w^{(31)}(k, a_{ff}\, b_{fb})\, [u_{21}(k) + u_{22}(k) + u_{23}(k)]\right\}\right] \tag{18a}$$

$$u_{32}(k)=\Psi\left[\left\{g_s^{(32)} w^{(32)}(k, a_{ff}\, b_{fb})\, [u_{22}(k)+ u_{21}(k) + u_{23}(k)]\right\}\right] \quad (18b)$$

$$u_{33}(k)=\Psi\left[\left\{ g_s^{(33)} w^{(33)}(k, a_{ff}\, b_{fb})\, [u_{23}(k)+ {}_{22}(k) + u_{21}(k)]\right\}\right]. \quad (18c)$$

Combining these equations and substituting into and simplifying Eqn. (18) yields

$$u(k) = \Psi\left[\left\{ g_s^{(31)} w^{(31)}(k, a_{ff}, b_{fb}) + g_s^{(32)} w^{(32)}(k, a_{ff}, b_{fb}) + g_s^{(33)} w^{(33)}(k, a_{ff}, b_{fb})\right\}\right] \bullet$$
$$\Psi\left[\left\{ g_s^{(21)} w^{(21)}(k, a_{ff}, b_{fb}) + g_s^{(22)} w^{(22)}(k, a_{ff}, b_{fb}) + g_s^{(23)} w^{(23)}(k, a_{ff}, b_{fb})\right\}\right] \bullet$$
$$\Psi\left[\left\{ g_s^{(11)} w^{(11)}(k, a_{ff}, bb_{fb}) + g_s^{(12)} w^{(12)}(k, a_{ff}, b_{fb})\; g_s^{(13)} w^{(13)}(k, a_{ff}, b_{fb})\right\}\right] \bullet x_s(k) \quad (19)$$

where $x_s(k)$ is the weighted input vector and is given by

$$x_s(k) = \left\{\begin{bmatrix} (x_1) \\ (x_2) \\ (x_3) \end{bmatrix} \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix}\right\} = x^T . s. \quad (20)$$

The dynamic neural system described by the above equations maps an n-dimensional input vector $x(k) \in R^n$ into a p-dimensional neural output vector $u(k) \in R^p$. In Eqn. (20), x and s are the vectors of input signals and the scaling factors respectively. In Eqn. (19), the first term represents the output stage, the middle term the intermediate stage and the last term the input stage. For applications to the control of unknown nonlinear dynamic systems, we have used a three-stage dynamic neural network with each stage comprising of two DNUs. The formulation of nonlinear control problem and the computer simulation results are discussed in the next section.

5. Computer Simulation Results

A nonlinear plant considered in this paper is of the form: $y(k+1) = f[y(k), y(k-1),.., y(k-n+1); u(k), u(k-1),..., u(k-m+1)]$, where $[u(k), y(k)]$ represents the input - output pair of a SISO plant at time k, and $m \le n$. The block schematic of the control scheme is shown in Fig. 2. The problem to be addressed consists of finding a control signal u(k) that will force the output y(k) to track asymptotically the desired output $y_d(k)$; that is, $\lim \left[y_d(k) - y(k)\right] = 0$ as $k \to \infty$. To achieve this objective, the following assumptions about the nonlinear plant are made:

Assumption 1: For any state $q \in R^n$: $0 < k_1 \le |f[.]|$.

Assumption 2: For any $k \in [0, \infty]$ the desired output $y_d(k)$ and its n-derivatives $y_d^{(1)}(k), y_d^{(2)}(k),..., y_d^{(n)}(k)$, are uniformly bounded; that is, $\left|y_d^{(i)}(k)\right| \le m_i$, $i = 0,1,2,.,n$.

Assumption 3: There exist coefficients a_{ff} and b_{fb} such that $\hat{f}[.]$ and $\hat{g}[.]$ approximate the nonlinear functions f[.] and g[.] respectively with an accuracy ε on Ω, a compact subset of R^n; that is, $\max \left|f[.] - \hat{f}[.]\right| \le \varepsilon$, and
$\max \left|g[.] - \hat{g}[.]\right| \le \varepsilon, \ \forall\, q \in$ on Ω.

The reference (desired) signal is applied to the input $x_1(k)$ and the plant output is fed back to the input $x_2(k)$. The components of the scaling vector, s_1, s_2, were set to [1, -1] respectively.

Example 1: As mentioned above, an unknown nonlinear plant is assumed to be governed by the difference equation $y(k) = f[y(k-1), y(k-2); u(k), u(k-1),u(k-2)]$ with an unknown function

$$f[.] = \frac{\left[\sin\left\{\pi\left(y^2(k-2) + 0.5\right)\right\}\right] + 0.3\sin(2\pi u(k))}{1 + u^2(k-1) + u^2(k-2)},$$

which was changed at k = 250 to

$$f[.] = \sin\left\{\pi\left(y^2(k-1)+y^2(k-2)\right)\right\} + \sqrt{\left|\left\{u^2(k) + u^2(k-1) + u^2(k-2)\right\}\right|}.$$

The plant parameters were $\beta_{ff} = [1.2, 1, 0.8]$, and $\alpha_{fb} = [1.3, 0.9, 0.7]$. The input to the system was $x(k) = \sin(2\pi k / 250)$ in the interval [-1, 1]. The simulation results obtained for this case are shown in Fig. 3. In Fig. 3a are shown the error and output responses. Figs 3b and 3c show the adaptation in somatic gain (slope) g_s of the activation function with respect to learning trials, k and performance index J respectively. It can be observed from simulation results that the neural network was able to drive the plant toward the desired response even in the presence of changing nonlinear characteristics. The optimum somatic gain was found to be 0.56.

Example 2: Since controllers designed based on neural network architectures exhibit learning and adaptive capabilities, the control law is independent of the plant configuration. In this example is shown this adaptive capability of the neural network by changing plant models during the control process. The different nonlinear plant models are well described in [6]. The changes in plant dynamics were made as follows:

Model III, for $0 \le k < 250$:

$$f[.] = \frac{\left[2+\cos\left\{7\pi\left(y^2(k-1)+y^2(k-2)\right)\right\}\right] + e^{-y(k)}}{\left[1+y^2(k-1)+y^2(k-2)\right]},$$

$$g[.] = \frac{\sqrt{\left|\left\{u^2(k)+u^2(k-1)+u^2(k-2)\right\}\right|}}{\left[1+u^3 k\right]}),$$

Model I, for $250 \le k < 750$:

$$f[.] = = u^3(k) + 0.3\sin(2\pi u(k-1)) + 0.1\sin(5\pi u(k-2)),$$

Model IV, for $750 \le k < 1050$:

$$f[.] = \frac{\left[2+\cos\left\{7\pi\left(y^2(k-1)+y^2(k-2)\right)\right\}\right] + e^{-u(k)}}{\left[1+u^2(k-1)+u^2(k-2)\right]}, \text{ and}$$

Model II, for $1050 \le k \le 1500$:

$$f[.] = \frac{0.1\sin\pi\sqrt{|y^2(k)|}}{\left[1+y^2(k-1)+y^2(k-2)\right]}.$$

The system input is same as in the above example. The output and somatic gain responses are shown in Fig. 4a and 4b respectively. As observed from this figure that the neural network was able to adapt very fast to the changing models of the nonlinear plant. Figure 4c shows the system response with a neural network having fixed sigmoidal function of gain 0.5. This example demonstrates the effect of nonlinear function on system response.

6. Conclusions

In this paper we have described a dynamic neural network which determines the optimum gain of the sigmoidal function for a given task. This selftuning facilitates expansion of the application range of neural networks, for an improper selection of sigmoidal gain may result in undesirable or even an unstable response. The effectiveness of the proposed neural network structure has been demonstrated through computer simulation results.

References

[1] W.J. Freeman, "Dynamics of image formation by nerve cell assemblies", in E. Basar, H. Flohr, H. Haken and A.J. Mandell, Eds., Synergitcs of the brain, Berlin, *Springer-Verlag*, 1983.

[2] T. Yamada and T. Yabuta, "Neural network controller using autotuning method for nonlinear functions", *IEEE Trans. on Neural Networks*, Vol. 3, No. 4, pp. 595-601, July 1992.

[3] M.M. Gupta and D.H. Rao, "Synaptic and somatic adaptations in dynamic neural networks", *Second Int. Conf. on Fuzzy Logic and Neural Networks*, Fukuoka, Japan, pp. 173-177, July 17-22, 1992.

[4] M.M. Gupta and D.H. Rao, "Dynamic neural units with applications to the control of unknown nonlinear systems", *The Journal of Intelligent and Fuzzy Systems*, [In Press].

[5] M.M. Gupta and D.H. Rao, " Dynamic neural units in the control of linear and nonlinear systems", *The International Joint Conference on Neural Networks (IJCNN)*, Baltimore, pp. 100-105, June 9-12, 1992.

[6] K.S. Narendra and K. Parthasarthy, "Identification and control of dynamical systems using neural networks", *IEEE Trans. on Neural Networks*, pp. 4-27, Vol. 1, No. 1, March 1990.

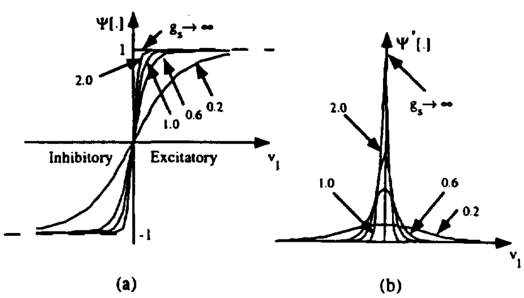

Fig.1: (a) Sigmoid function $\Psi[.] = \tanh[g_s v_1]$, and

(b) The derivative sigmoid function ,Ψ [.], which tends to become a sign function as the slope → ∞ ; that is, the slope Ψ [.] tends to become very narrow with an increasing value of g_s.

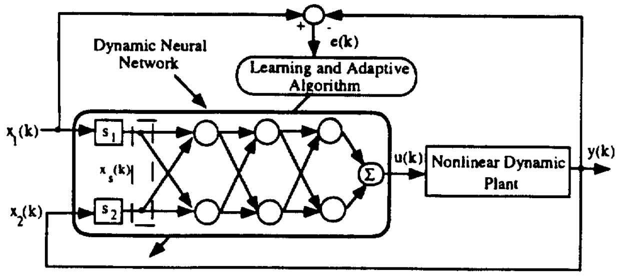

Fig. 2: The control scheme using dynamic neural network for the control of unknown nonlinear dynamic plants. Each circle represents a DNU.

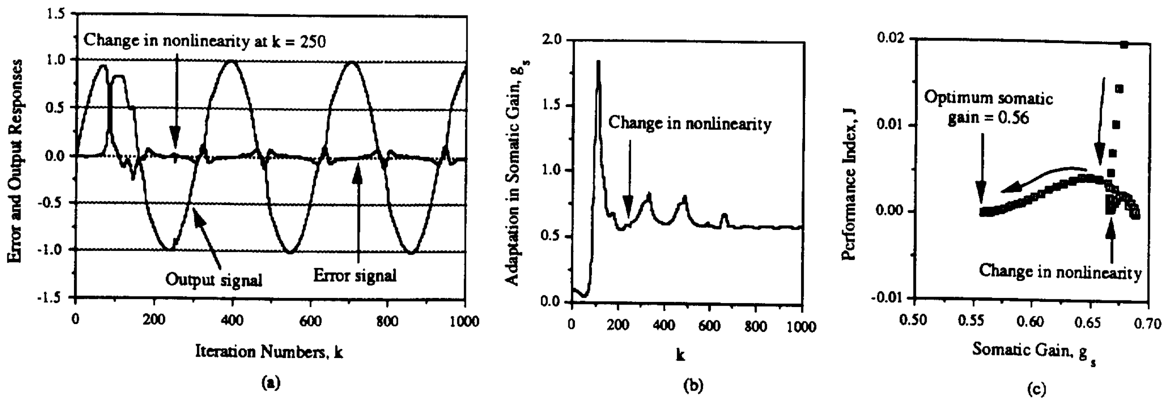

Fig. 3: (a) The error and output responses, (b) the adaptation in somatic gain, g_s, and (c) performance index variation with respect to somatic gain.

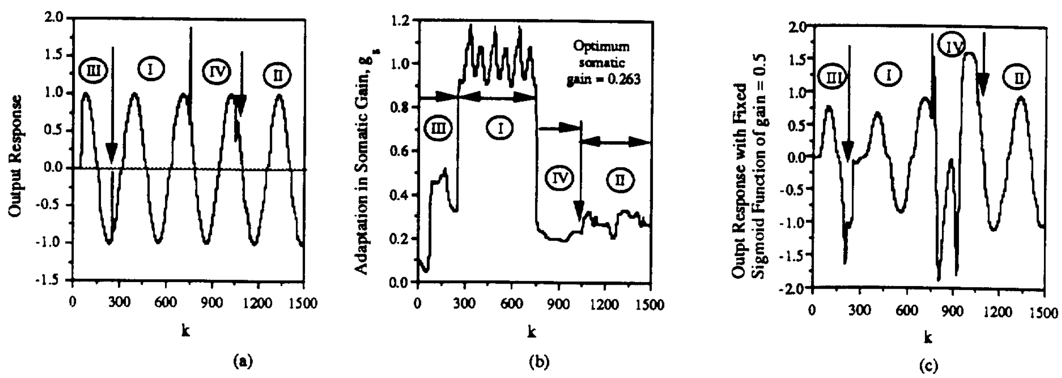

Fig. 4: (a) The error response. Circles with numbers I, II, III and IV denote the different nonlinear models
(b) the adaptation in somatic gain, g_s, for variations in nonlinear plant models, and
(c) The output response using a neural network having fixed sigmoidal function with gain 0.5.