

A Radial Basic Function with Multiple Input and Multiple Output Neural Network to Control a Non-Linear Plant of Unknown Dynamics

¹R. El-Kouatly and ²G.A. Salman

¹Department of Electronic and Communication Engineering,
Faculty of Mechanical and Electrical Engineering, Damascus University, Syria

²College of Computer Engineering and Information,
Syrian National Private University for Science and Technology, Syria

Abstract: In this study, the application of the Radial Basis Function (RBF) with Multiple Input and Multiple Output (MIMO) Neural networks to control two types of non linear model plants of unknown dynamics. For the first step a model of a control was developed using the variable liquid-level which can be use in a chemical plant, or power station, where the liquid-level is change within fixed real time. In this control system Radial Basis Function (RBF) neural networks was used to control the liquid-level of the plant. The second step introduced the changes of the liquid-level in real time, also Radial Basis Function with MIMO neural networks has been used to control the level liquid. The study shows that the proposed control system produces accurate results for the two types of models. However, we notice that the training, using back propagation, for the second model take a more considerable time than training the first model.

Key words: Radial Basis Function (RBF) neural networks, back propagation

INTRODUCTION

A direct adaptive control algorithm is developed for a class of nonlinear plants. No restriction has been imposed on the plant structure. The condition imposed is that the plant must satisfy the instantaneous input-output gain and should be positive. Many Artificial Neural networks (ANNs), for nonlinear control structure, have been employed (Kosko, 1997) and (Passino and Yurkovich, 1998). In line with the gain scheduling principle, however, the controller also has pseudo linear time-varying structure with the parameters being functions of the operating point (Loannou and Fidan, 2006).

It is well known that the response of a nonlinear plant generally can't be shaped into a desired pattern using a linear controller. Consequently, a nonlinear controller is required for such plants (Murray *et al.*, 2003; Xiaohong and Shen, 2005). One of the main difficulties in designing the nonlinear controller, however, is the lack of a general structure for it. Nonlinear controller design may be viewed as a nonlinear function approximation problem. In recent years, ANNs become an important element in approximating a nonlinear function (Wu, 1994), their ability to learn arbitrary nonlinear mappings can be effectively utilized to represent the controller nonlinearity.

The other advantages inherent in ANNs such as their robustness, parallel architecture and fault tolerant capability act as further incentives (Raun, 1997).

Many researchers (Diao and Passion, 2002; Nounou and Passion, 2004; Ge and Wang, 2002) have considered adaptive neural network to control nonlinear dynamical systems. All of these articles provide only local convergence analysis. In this study we developed the analysis of local convergence further toward to the global convergence, where for such purpose we use ANNs with many hidden layers and different number of neurons.

DIRECT ADAPTIVE CONTROL

The direct adaptive control means that the controller parameters are directly updated, without the need for the implementation of an identifier of the plant. Such method of control depends on recursive Least-Mean-Square (LMS) and gradient approximation to estimate the plant parameters (Murray *et al.*, 2003; Xiaohong and Shen, 2005). The basic idea of direct adaptive control can be shown in Fig. 1 (Murray *et al.*, 2003).

This problem can be solved using Kaczmarz's projection algorithm (Chen and Kalil, 1992). Consider the unknown parameter as an element $\hat{\theta}(t)$, in the consider range (R^n), such that:

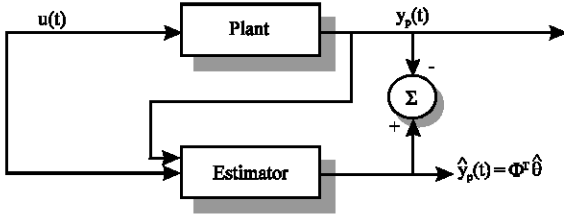


Fig. 1: Plant estimator using LMS approximation

$$\hat{y}_p(t) = \phi^T(t)\hat{\theta}(t) \quad (1)$$

where, $\phi^T(t)$ are known functions that may depend on other known variable and $\hat{\theta}(t)$ are unknown parameters.

From the interpretation it is clear that the n measurements, of the elements $\phi_1(t), \phi_2(t), \dots, \phi_n(t)$ must span the space R^n so that the parameter vector $\hat{\theta}$ is unique (Wu, 1994). Assume that an estimate $\hat{\theta}(t-1)$ is available and that a new measurement is obtained and since the measurement $\hat{y}_p(t)$ contains information only in the direction of the vector $\phi(t)$, in parameter space, we can update the estimate $\hat{\theta}$ as:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \beta\phi(t) \quad (2)$$

where, the parameter β is chosen so that $y_p(t) \equiv \hat{y}_p(t)$. Thus from Eq. 2 and 1, we obtain:

$$\begin{aligned} y_p(t) &= \phi^T(t)[\hat{\theta}(t-1) + \beta\phi(t)] \\ y_p(t) &= \phi^T(t)\hat{\theta}(t-1) + \beta\phi^T(t)\phi(t) \end{aligned}$$

Thus we have:

$$\beta = \frac{y_p(t) - \phi^T(t)\hat{\theta}(t-1)}{\phi^T(t)\phi(t)} \quad (3)$$

And the error has the form:

$$e(t) = y_p(t) - \phi^T(t)\hat{\theta}(t-1) \quad (4)$$

From Eq. 3, 4 and 2 we obtain:

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + \frac{y_p(t) - \phi^T(t)\hat{\theta}(t-1)}{\phi^T(t)\phi(t)}\phi(t) \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + \frac{\phi(t)e(t)}{\phi^T(t)\phi(t)} \end{aligned} \quad (5)$$

Also from Fig. 2 we can conclude that

$$u(t) = \phi^T(t)\theta(t) \quad (6)$$

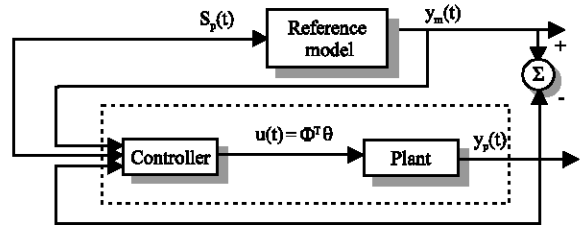


Fig. 2: Direct adaptive control using LMS approximation

$$y_p(t) = f(u(t)) \quad (7)$$

and thus,

$$y_p(t) = f(\phi^T(t)\theta(t)) \quad (8)$$

$$\theta(t) = \theta(t-1) + \beta\phi(t) \quad (9)$$

$$\begin{aligned} y_m(t) &\equiv y_p(t) \\ y_m(t) &= f(u(t)) \end{aligned} \quad (10)$$

$$y_m(t) = f(\phi^T(t)[\theta(t-1) + \beta\phi(t)]) \quad (11)$$

$$y_m(t) = f(\phi^T(t)\theta(t-1)) + \beta f(\phi^T(t)\phi(t)) \quad (12)$$

Since the term $\phi^T(t)\phi(t)$ is not depend on $f(\cdot)$ then we have:

$$y_m(t) = f(\phi^T(t)\theta(t-1)) + \beta\phi^T(t)\phi(t) \quad (13)$$

and thus

$$\beta = \frac{y_m(t) - f(\phi^T(t)\theta(t-1))}{\phi^T(t)\phi(t)} \quad (14)$$

$$e(t) = y_m(t) - f(\phi^T(t)\theta(t-1)) \quad (15)$$

Using Eq. 14, 15 and 9 we have

$$\begin{aligned} \theta(t) &= \theta(t-1) + \frac{y_m(t) - f(\phi^T(t)\theta(t-1))}{\phi^T(t)\phi(t)}\phi(t) \\ \theta(t) &= \theta(t-1) + \frac{\phi(t)e(t)}{\phi^T(t)\phi(t)} \end{aligned} \quad (16)$$

To avoid potential problems that may occur when $\phi(t) = 0$, the projection algorithm can be modified in practice, as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\mu * \phi(t)}{\varepsilon + \phi^T(t)\phi(t)}e(t) \quad \varepsilon > 0 \quad (17)$$

ε : is a small number depend on n and number of hidden layers in the ANNs, where, $\theta \equiv \hat{\theta}$ and μ is a learning rate depend on the type of ANNs, ε and g_{max} given in Eq. 19 such that

$$0 < \mu < \frac{2}{g_{max} + \varepsilon}.$$

Figure 2 shows the direct adaptive control using LMS approximation (Raun, 1997).

PROBLEM STATEMENT

Consider the general model of the Single-Input-Single-Output (SISO) nonlinear plant given by:

$$y_p(t) = f(y_p(t-1), \dots, y_p(t-n), u(t-d), \dots, u(t-n)) \quad (18)$$

where, $y_p(t)$ is the measured plant output, $u(t)$ is the measured plant input (control output), n is the known plant order, d is the known plant delay and $f(\cdot)$ is an unknown nonlinear function.

Further, the function $f(\cdot)$ is assumed to be continuous and bounded such that:

$$0 < g_{min} - \varepsilon \leq \frac{\partial f(\cdot)}{\partial u(t-d)} \leq g_{max} + \varepsilon < \infty \quad (19)$$

From Eq. 19 we conclude that the function $f(\cdot)$ is monotonically increasing in $u(t-d)$, which is analogous to the assumption that the instantaneous input gain in a linear plant (frequently denoted by b_0) be positive. The problem, now, is to design a direct and stable adaptive control strategy that drives the plant output to a reference command.

Here, we provide some results to facilitate the development of proposed self-tuning algorithm. Assume that $y_p(t)$ being generated from arbitrary functions $\phi(t)$ using the following equations:

$$u(t) = \bar{\phi}(t)\theta^0 \quad (20)$$

$$y_p(t) = \bar{f}_i(u^0(t-d)) \quad (21)$$

where, the functions $\bar{f}_i(\cdot)$ satisfy

$$0 < g_{min} - \varepsilon \leq \frac{\partial \bar{f}_i(\cdot)}{\partial u(t-d)} \leq g_{max} + \varepsilon < \infty \quad (22)$$

Equation 21 is essentially a simplified representation of Eq. 18. For each time instant t , the function $\bar{f}_i(\cdot)$ can be related by the following relationship:

$$\bar{f}_i(x) \equiv f(y_p(t-1), \dots, y_p(t-n), x, u(t-d-1), \dots, u(t-n)) \quad (23)$$

For given $y_p(t-1), \dots, y_p(t-n), u(t-d-1), \dots, u(t-n)$. Using Eq. 23, we can verify that Eq. 19 and 22 are equivalent. Therefore, the following development, which is based on Eq. 21 as well, applies to the plant, represented by Eq. 18.

Now, consider the estimation of the parameter θ^0 from the observations $\phi(t)$ and $y_p(t)$ employing the following algorithm, modified (given that t, d, n_1, n_2 are integers and t is expressed as $t = n_1 + n_2d$, with $0 \leq n_1 < d$, we define t [model d] as the integer value n_1), in which its original (Ge and Wang, 2002),

$$\theta(t) = \begin{cases} \theta(t-1) + \frac{\mu \phi(t-d)e(t)}{\varepsilon + \phi^T(t-d)\phi(t-d)} & \text{if } t[\text{mod } d] = 0 \\ \theta(t-1) & \text{otherwise} \end{cases} \quad (24)$$

$$u(t) = \phi^T(t)\theta(t) \quad (25)$$

$$e(t) = y_p(t) - \bar{f}_i(u(t-d)) \quad (26)$$

$$0 < \mu < \frac{k}{g_{max} + \varepsilon}, \quad \varepsilon > 0 \quad \text{and} \quad 1 \leq k \leq 2 \quad (27)$$

where, $\theta(t)$ denotes the estimate of θ^0 after the t^{th} observation.

RADIAL BASIS FUNCTION NETWORKS (RBF)

A RBF neural network can be represented by a two-hidden-layer feed forward network structured, the first layer of the network consists of RBF neurons and the second layer is linear neurons. The inputs to the neural network are directly connected to the RBF neurons are then connected to the linear neurons in the second layer through adaptable weights. For the sake of definiteness, let us assume that there are j inputs to the RBF neural network, i RBF neurons in the first layer and q linear neurons in the second layer.

The transfer function of the i^{th} RBF neuron in the first layer has the form:

$$z_i(x) = r(\|x - \rho_i\|^2 / \sigma_i^2) \quad (28)$$

where, $z_i(x)$ is the output of the i^{th} RBF neurons and $r(\cdot)$ is a radial basis function. The most common used RBF is the Gaussian function given below:

$$z_i(x) = e^{-(\|x - \rho_i\|^2 / \sigma_i^2)} \quad (29)$$

In which x is the input vector of dimension j and ρ_i is the center vector of dimension j of the i^{th} RBF neurons and σ_i the positive scalar of the width of the i^{th} RBF neuron. Both ρ_i and σ_i can be different for different RBF neurons and this offering flexibility in locally shaping for the RBF neural network structure. The transfer function for the i^{th} linear neuron is given by the following matrix notation:

$$M = W \times z(x) \tag{30}$$

where, $M = [M_1, M_2, \dots, M_i]^T$ lumps the outputs of all of the linear neurons together to form a vector; $z(x) = [z_1(x), z_2(x), \dots, z_i(x)]^T$ and W is an $i \times q$ matrix with the iq^{th} entry w_{iq} interpreted as the adaptable weight from the i^{th} RBF neuron in first layer to the q^{th} linear neuron in the second layer.

Equation 30 apparently decomposes all of the parameters involved in a RBF neural network into two categories: the matrix W containing all of the adaptable weights and the vector $z(\cdot)$, containing all such structural parameters as the number of neurons i , the centers ρ_i and the widths σ_i .

CONTROLLER STRUCTURE

In the proposed adaptive control strategy, the controller structure is fixed a priori, but its parameters are directly updated based on the input-output measurement employing a stable adaptive scheme. It is well known that satisfactory control of a nonlinear plant generally requires a nonlinear controller. Accordingly, a nonlinear controller structure has been chosen. In line with the gain scheduling principle, however, the controllers also have a pseudolinear time-varying structure with the corresponding parameters being functions of the operating point (Naraendra and Parthasarathy, 1990).

We assume that the plant operating point is measurable or may be inferred. Examples of such systems include hydraulic drigs, Ac motors, power plant control systems and turbofan engines (Wang *et al.*, 1995). The controller requires the measurement of the input-output as well as the variables representing the operating point. The controller parameters are assumed to be nonlinear functions of the operating point. The nonlinear functions are approximated by a set of neural networks. The parameters of these networks are directly updated online based on the input-output measurements. In general the controller structure has the form (Chen and Kalil, 1992),

$$\begin{aligned} u(t) = & -r_1(\theta_t) * u(t-1) - \dots - r_{p-1}(\theta_t) * u(t-p+1) \\ & + S_0(\theta_t) * y_p(t) + \dots + S_{\sigma-1}(\theta_t) * y_p(t-\sigma+1) \\ & + K_0(\theta_t) * y_m(t) + \dots + K_{\gamma-1}(\theta_t) * y_m(t-\gamma+1) \end{aligned} \tag{31}$$

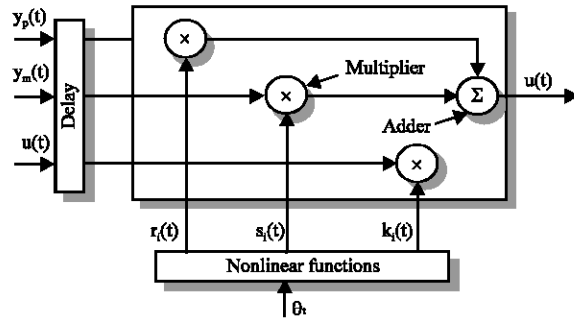


Fig. 3: Controller structure

where, the order of the controller is given by $\max(\rho, \sigma, \gamma)$ (Fig. 3). Typical choices for ρ and σ are given as $\rho = \sigma \geq n$, where n is the dimension of the space R^n . This guideline stems from linear control theory that allows conciliation of all open-loop poles and placement of all closed-loop poles at the desired locations in the Z -plane. The choice of γ , however, depends on the reference signal $y_m(t)$ and corresponds to its moving average modeling.

The quantity $y_m(t)$ is the desired output, which can be generated from an appropriate reference model driven by the set point $S_p(t)$. The parameters $r_i(\theta_t)$, $S_i(\theta_t)$ and $K_i(\theta_t)$ can be regarded as the time-varying parameters of the pseudolinear controller and the variable θ_t contains all of the measurements that effect the plant operating point. Although the controller parameters $[r_i(\theta_t)$ and $S_i(\theta_t)]$ shape up the transient response, the introduction of the parameters $K_i(\theta_t)$ ensures set point tracking. All of these controller parameters are assumed to be nonlinear function of the operating points.

THE STRUCTURE OF DIRECT ADAPTIVE CONTROL USING RADIAL BASIS FUNCTIONS (RBFs)

This paper study also consider the Mingle-Input-Multi-Output (MIMO) structure neural model. The structure of MIMO for RBFs is shown in Fig. 4. In this study we assume that RBF network is a processing structure consisting of an input layer, two hidden layer and an output layer. The hidden layer of a RBF network consists of an array of nodes (neuron) and each node contains a parameter vector called a center.

The node calculates the Euclidean distance between the center and the network input vector and passes the result through a radially symmetric nonlinear function. The output layer is essentially a set of linear combiners. The overall input-output response of an N_j -input, N_q -output RBF network is a mapping:

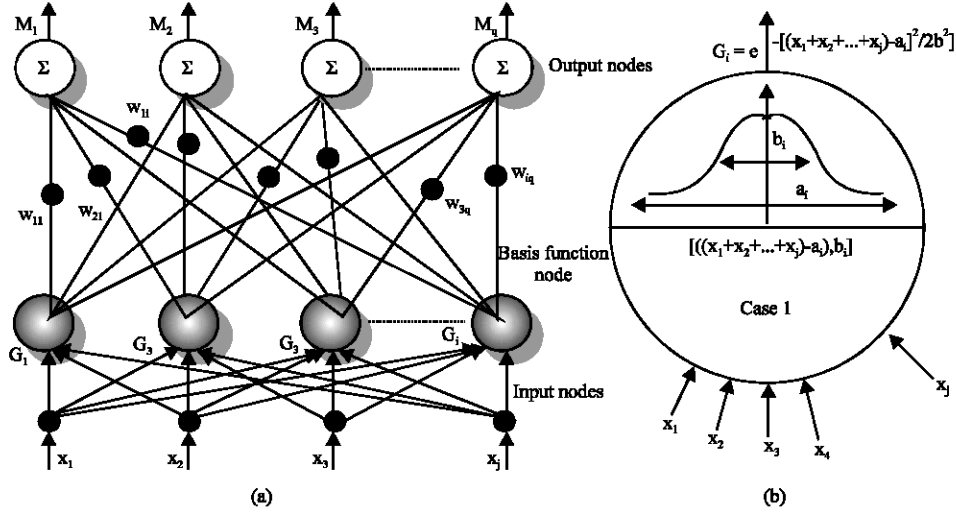


Fig. 4: Radial basis function network (a) model (neuron) and (b) basis function

$$f_{RBF} : \mathbb{R}^{N_j} \rightarrow \mathbb{R}^{N_q} \text{ where } f_{RBF_q} = \sum_{i=1}^{N_i} w_{i,q} * G_i \quad (32)$$

In the first case we used the following model:

$$x = (x_1 \ x_2 \ x_3 \ \dots \ x_j) \quad 1 \leq j \leq N_j$$

$$f_{RBF_q} = \sum_{i=1}^{N_i} w_{i,q} * G_i [((x_1 + x_2 + \dots + x_j) - a_i), b_i] \quad (33a)$$

$$f_{RBF_q} = \sum_{i=1}^{N_i} w_{i,q} * e^{-((x_1 + x_2 + \dots + x_j) - a_i)^2 / 2b_i^2} \quad (33b)$$

In the second case we used the following model:

$$x = (x_1 \ x_2 \ x_3 \ \dots \ x_j) \quad 1 \leq j \leq N_j$$

$$f_{RBF_q} = \sum_{i=1}^{N_i} w_{i,q} * G_i [\|x - a_i\|^2, b_i] \quad (34a)$$

$$f_{RBF_q} = \sum_{i=1}^{N_i} w_{i,q} * e^{-((x_1 - a_i)^2 / 2b_i^2) + \dots + e^{-((x_j - a_i)^2 / 2b_i^2)} \quad (34b)$$

Then:

$$r_q \approx \sum_{i=1}^{N_i} w_{i,q}^r G_i(\theta_i) \quad 1 \leq q \leq N_q \quad (35)$$

$$S_q \approx \sum_{i=1}^{N_i} w_{i,q}^s G_i(\theta_i) \quad 1 \leq q \leq N_q \quad (36)$$

$$K_q \approx \sum_{i=1}^{N_i} w_{i,q}^k G_i(\theta_i) \quad 1 \leq q \leq N_q \quad (37)$$

where, $\theta_i = x_1, x_2, \dots, x_j \quad 1 \leq j \leq N_j$ operating points, $w_{i,q}$ are the weights of the linear combiners, $\|\cdot\|$ denotes the Euclidean norm, a_i are the RBF centers, $G(x_1, x_2, x_3, \dots, a_i, b_i)$ is a function from $\mathbb{R}^+ \rightarrow \mathbb{R}$ and N_j is the number of input nodes and N_i is the number of hidden nodes (Sanner and Stoline, 1992).

The neural weight update for MIMO structure in the direct adaptive control model is given by Sanner and Stoline (1992):

$$u(t) = \phi^T(t) * \theta \quad (38)$$

where, θ parameter can be updated, $\theta(t)$ measurement of known functions. For the computational purpose, using Eq. 35, 36, 37 and 31, we have:

$$\begin{aligned} u(t) = & -G_i(\theta_i) * u(t-1) * w_{i1}^r - G_i(\theta_i) * u(t-1) * w_{i21}^r - \dots \\ & \dots - G_i(\theta_i) * u(t-1) * w_{iN_i 1}^r - \dots - G_i(\theta_i) * u(t-\rho+1) * w_{i\rho-1}^r \\ & - G_i(\theta_i) * u(t-\rho+1) * w_{i2\rho-1}^r - \dots - G_i(\theta_i) * u(t-\rho+1) * w_{iN_i \rho-1}^r \dots \\ & + G_i(\theta_i) * y_p(t) * w_{i0}^s + G_i(\theta_i) * y_p(t) * w_{i20}^s + \dots \\ & \dots + G_i(\theta_i) * y_p(t) * w_{iN_i 0}^s + \dots \\ & \dots + G_i(\theta_i) * y_p(t-\sigma+1) * w_{i\sigma-1}^s + G_i(\theta_i) * y_p(t-\sigma+1) * w_{i2\sigma-1}^s \\ & + \dots + G_i(\theta_i) * y_p(t-\sigma+1) * w_{iN_i \sigma-1}^s + G_i(\theta_i) * y_m(t) * w_{i0}^k + \\ & G_i(\theta_i) * y_m(t) * w_{i20}^k + \dots + G_i(\theta_i) * y_m(t) * w_{iN_i 0}^k + \dots \\ & \dots + G_i(\theta_i) * y_m(t-\gamma+1) * w_{i\gamma-1}^k + G_i(\theta_i) * y_m(t-\gamma+1) * w_{i2\gamma-1}^k \\ & + \dots + G_i(\theta_i) * y_m(t-\gamma+1) * w_{iN_i \gamma-1}^k \end{aligned} \quad (39)$$

Thus if we use Eq. 25 we get,

$$\phi(t) = \begin{bmatrix} -G_1(\theta_1) * u(t-1) \\ -G_1(\theta_1) * u(t-2) \\ \vdots \\ -G_1(\theta_1) * u(t-\rho+1) \\ \hline -G_1(\theta_1) * y_p(t) \\ -G_1(\theta_1) * y_p(t-1) \\ \vdots \\ -G_1(\theta_1) * y_p(t-\sigma+1) \\ \hline -G_1(\theta_1) * y_m(t) \\ -G_1(\theta_1) * y_m(t-1) \\ \vdots \\ -G_1(\theta_1) * y_m(t-\gamma+1) \end{bmatrix} \quad \theta = \begin{bmatrix} w_{11}^r \\ w_{21}^r \\ \vdots \\ w_{N_1,1}^r \\ w_{1,\rho-1}^r \\ w_{2,\rho-1}^r \\ \vdots \\ w_{N_1,\rho-1}^r \\ \hline w_{10}^s \\ w_{20}^s \\ \vdots \\ w_{N_1,0}^s \\ w_{1,\sigma-1}^s \\ w_{2,\sigma-1}^s \\ \vdots \\ w_{N_1,\sigma-1}^s \\ \hline w_{10}^k \\ w_{20}^k \\ \vdots \\ w_{N_1,0}^k \\ w_{1,\gamma-1}^k \\ w_{2,\gamma-1}^k \\ \vdots \\ w_{N_1,\gamma-1}^k \end{bmatrix}$$

(40)

The weights for generating the parameters (r_{p-1})

The weights for generating the parameters (s_{σ-1})

The weights for generating the parameters (k_{γ-1})

In order to compute the control input, we require the knowledge about the parameter vector θ . We modify the propose algorithm for the adaptive controlling of the plant given (Ahmed, 2000) such that $u(t) = \phi^T(t) * \theta(t)$.

$$\theta(t) = \begin{cases} \theta(t-1) + \frac{\mu * \phi(t-d) * e(t)}{\varepsilon + \phi^T(t-d) * \phi(t-d)} & \text{if } t[\text{modd}] = 0 \text{ and } \varepsilon > 0 \\ \theta(t-1) & \text{Otherwise} \end{cases}$$

(41)

$$e(t) = y_m(t) - y_p(t)$$

(42)

$$0 < \mu < \frac{2}{g_{max} + \varepsilon}$$

(43)

where, $y_p(t)$ and $y_m(t)$, are the actual and the desired values of the plant output respectively and $y_m(t)$, can be generated as the output of an appropriate reference model.

SIMULATION RESULTS

Here, we develop a scheme, using RBF neural network, for a direct adaptive control which suited to be applied for controlling liquid-level system.

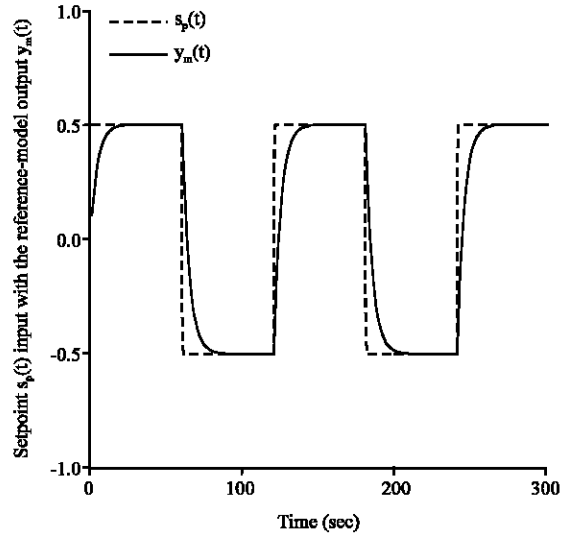


Fig. 5: The reference-model output of the example one

Model 1: In this case a liquid-level system can be described by the following delay difference equation:

$$y_p(t) = 0.9722 * y_p(t-1) + 0.3578 * u(t-1) - 0.1295 * u(t-2) - 0.3103 * y_p(t-1) * u(t-1) - 0.04228 * y_p^2(t-2) + 0.1663 * y_p(t-2) * u(t-2) - 0.03259 * y_p^2(t-1) * y_p(t-2) - 0.3513 * y_p^2(t-1) * u(t-2) + 0.3084 * y_p(t-1) * y_p(t-2) * u(t-2) + 0.1087 * y_p(t-2) * u(t-1) * u(t-2)$$

(44)

Where, $y_p(t)$ is the output and $u(t)$ is the input of the system, this model is obtained through identification of a laboratory-scale liquid-level system (Sales and Billings, 1990).

The reference-model: The $y_m(t)$ has been taken as the output of a first-order reference-model with transfer function:

$$G_m(z) = \frac{0.2 * z}{(z - 0.8)}$$

And driven by the setpoint $S_p(t)$ of amplitude ± 0.5 shown in Fig. 5.

We first assume that the time delay $d = 1$, so that the parameters have been updated every time. We consider operating point θ_i has been taken as $\theta_i = [y_p(t), e(t)]$ and $g_{max} = 2$, so that the learning rate $0 < \mu < 1$, $\epsilon > 0$, with $\mu = 0.95$, $\epsilon = 0.001$. Five Gaussian basis functions with centroids a_i , [ranging uniformly between -0.6 and +0.6] have been chosen as: [$a_1 = -0.48, a_2 = -0.24, a_3 = 0.0, a_4 = 0.24, a_5 = 0.48$]. The performance was measured by computing the error using (Wang *et al.*, 1995):

Table 1: The change of the performance measure of error E (least mean square), with the number of iteration N, depending on the values of ϵ for case 1, using MIMO adaptive control with $d = 1$ and $\mu = 0.95, \epsilon = 10, b = 0.3$

$\epsilon = 25$		$\epsilon = 10$		$\epsilon = 1$		$\epsilon = 0.001$	
LME E	No. of iteration N	LME E	No. of iteration N	LME E	No. of iteration N	LME E	No. of iteration N
1	0.003673	1	0.00627	1	0.050653	1	0.085084
41	0.000398	49	0.000397	207	0.000399	464	0.000399
1333	0.0003	612	0.0003	659	0.0003	1313	0.0003
1338	0.000299	744	0.000299	679	0.000299	1343	0.000299
1428	0.000277	2925	0.00029	1001	0.00029	1734	0.00029
1837	0.00031	4500	0.000288	4814	0.000281	3751	0.00028
6000	0.000281	6000	0.000289	6000	0.000281	6000	0.000279

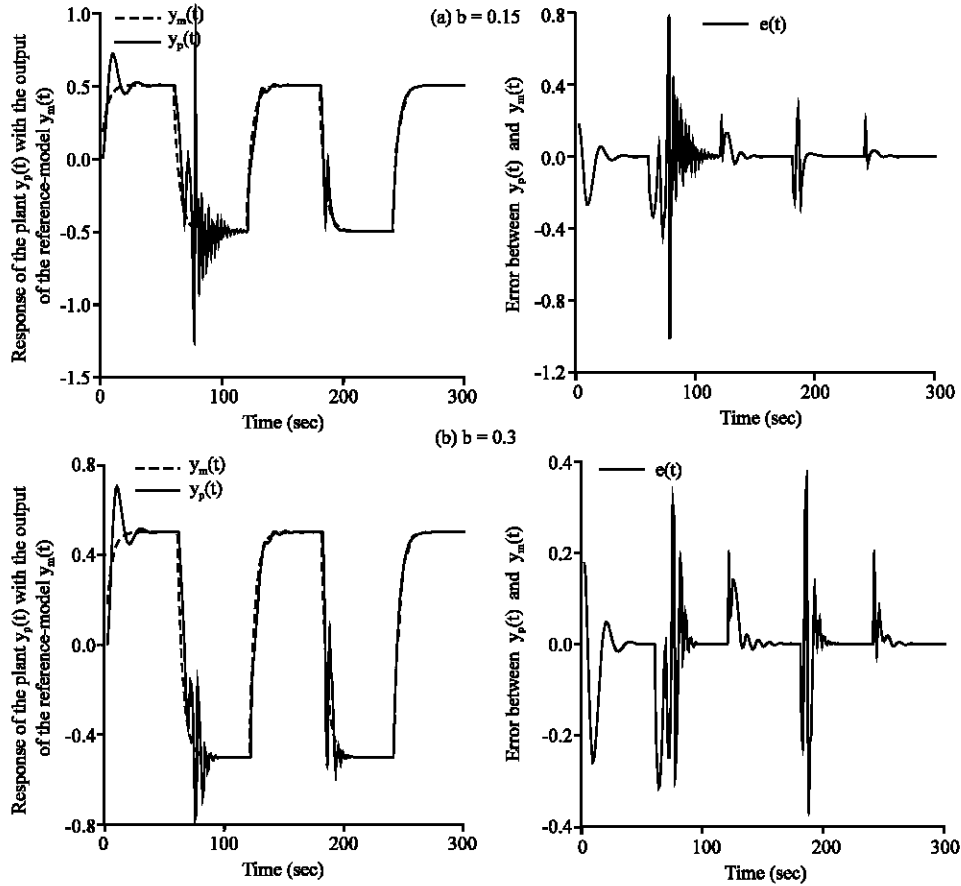


Fig. 6: The response of the plant $y_p(t)$, with the output of the reference-model $y_m(t)$ and error between them $e(t)$ (a) For $N = 1, E = 0.007119$, (b) For $N = 1, E = 0.03673$. For example one, (case 1) using direct adaptive control with [MIMO, RBFs, $d = 1$]

$$E = \frac{1}{N_k} \sum_{k=0}^{N_k} (y_m(k) - y_p(k))^2 \quad 45)$$

Using spread of RBFs equal $b = 0.15$ with no intersection between RBFs, for the first iteration $N = 1, E = 0.007119$, the results is shown in Fig. 6a. Then a spread of RBFs, equal $b = 0.3$ with intersection between RBFs, with the first iteration $N = 1, E = 0.03673$, the results is shown in Fig. 6b.

Table 1 shows the performance measure of error E with iteration N for different values of ϵ . The

selected value of $\epsilon = 10$ is the best value of performance measure of error.

For number of iteration equal to $N = 1001$, the response $y_p(t)$ with the output of the reference $y_m(t)$ and the error $e(t)$ is shown in Fig. 7. It is appear that the performance is better the performance measure of error E.

Model 2: In this case a liquid-level system, change with time, can be described by the following difference equation (Ahmed, 2000):

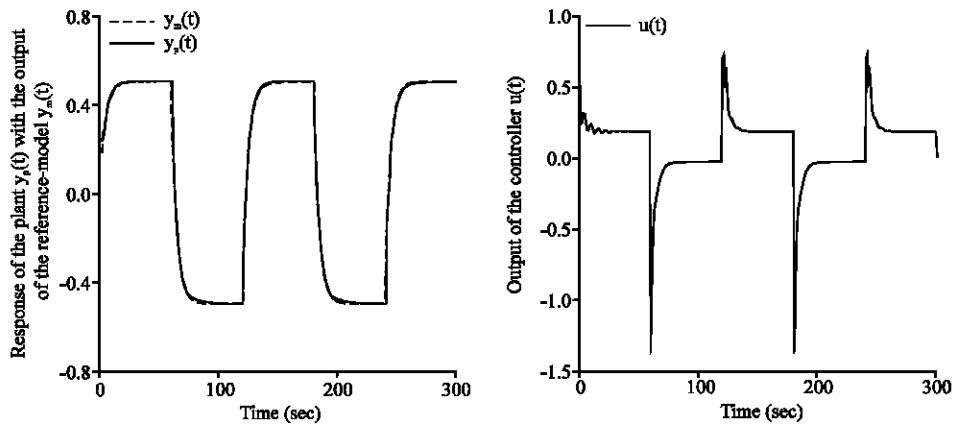


Fig. 7: The response of the plant $y_p(t)$, with the output of the reference-model $y_m(t)$, error between them $e(t)$ and output of the controller $u(t)$, using direct adaptive control with $d = 1$ and $\mu = 0.95$, $\varepsilon = 10$, $b = 0.3$. For Model one, (case 1) using direct adaptive control with [MIMO, RBFs, $d = 1$]

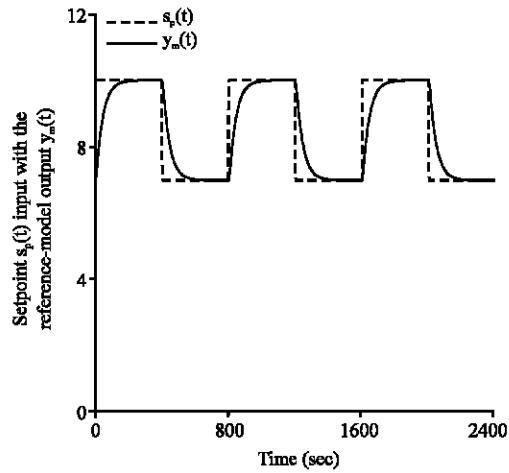


Fig. 8: The reference-model output of the model 2

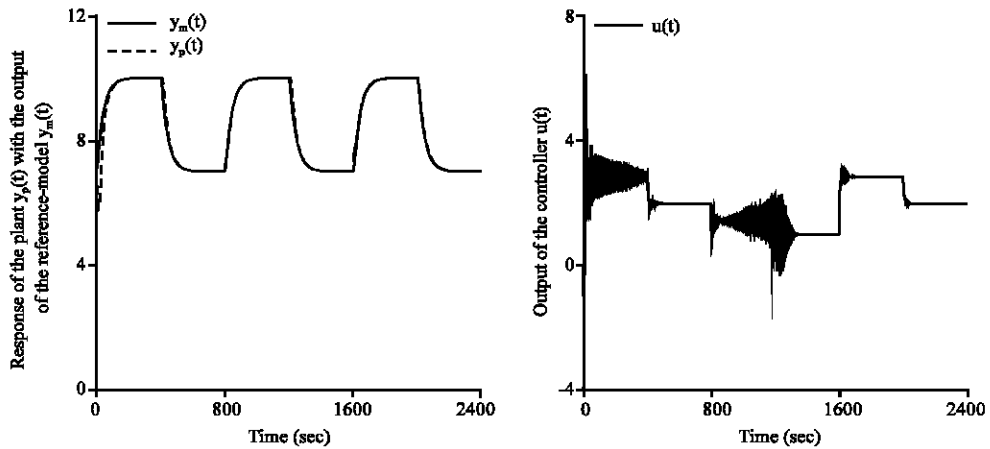


Fig. 9: The response of the plant $y_p(t)$, with the output of the reference-model $y_m(t)$, error between them $e(t)$ and output of the controller $u(t)$ for model 2. (case 1) using direct adaptive control with [MIMO, RBFs]

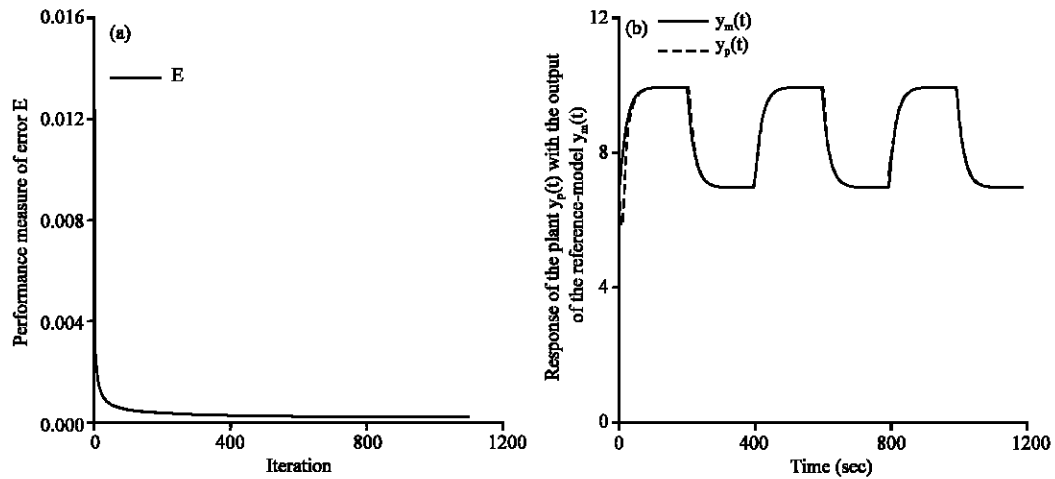


Fig. 10: (a) The performance measure of error E. (b) and the response of the plant $y_p(t)$, with the output of the reference-model of the model 2

$$\begin{aligned}
 G(s) &= \frac{3.5 * e^{-4*s}}{(200s^2 + 45s + 1)} && \text{for } 0 \text{ to } 800(\text{sec}), \\
 G(s) &= \frac{7 * e^{-4*s}}{(200s^2 + 45s + 1)} && \text{for } 800 \text{ to } 1600(\text{sec}), \\
 G(s) &= \frac{3.5}{(40s + 1)} && \text{for } 1600 \text{ to } 2400(\text{sec}),
 \end{aligned} \tag{46}$$

The reference-model: The $y_m(t)$, has been taken as the output of a second-order reference-model with transfer function:

$$G_m(s) = \frac{1}{(200s^2 + 45s + 1)}$$

And driven by the setpoint $S_p(t)$ of amplitude with maximum value equal 10, to minimum value equal 7 as shown in Fig. 8.

We consider operating point θ , has been taken as $\theta_t = [y_p(t), e(t)]$ and $g_{max} = 2$, so that the learning rate $0 < \mu < 1$, $\epsilon > 0$, with $\mu = 0.95$, $\epsilon = 10$. Five Gaussian basis functions with centroids a_i , [ranging uniformly between +7 and +10] have been chosen as: $[a_1 = 7.3, a_2 = 7.9, a_3 = 8.5, a_4 = 9.1, a_5 = 9.7]$. The performance was measured by computing the error using Eq. 45.

Using spread of RBFs equal $b = 1.5$ with intersection between RBFs, for $N = 1$ and then repeated for $N = 1100$, $E = 0.00025$, the results is shown in Fig. 9.

Figure 10a shows that the performance measure of error E when the parameters of the plant are changed, we can see no changes in the response, of the plant as shown in Fig. 10b.

CONCLUSIONS

From our numerical results in model 1 and 2, we conclude that back propagation neural network algorithm can be used effectively to compute and estimate the adaptive control for nonlinear plants. However, as the neural hidden layers, or neurons, increases we may get more accurate estimation of the nonlinear plant control. The learning rate μ depend on the upper bound of the continuous function $f(\cdot)$, $g_{max} + \epsilon$ and on the plant delay d .

REFERENCES

- Ahmed, M.S., 2000. Neural networks-based direct adaptive control for a class of nonlinear plants. IEEE Trans. Autom. Control, 45 (1): 119-125.
- Chen, F.C. and H.K. Kalil, 1992. Adaptive control of nonlinear systems using neural networks. Int. J. Control, 55 (5-7): 1299-1317.
- Diao, Y. and K.M. Passion, 2002. Adaptive fuzzy-neural control for interpolated nonlinear systems. IEEE Trans. Fuzzy Syst., 10 (5): 583-595.
- Ge, S.S. and C. Wang, 2002. Direct adaptive neural networks control for a class of nonlinear systems. IEEE Trans. Neural Networks, 13 (1): 214-221.
- Kosko, B., 1997. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall India Private Limited USA, Inc.
- Loannou, P. and B. Fidan, 2006. Adaptive Control Tutorial -Advances in Design and Control. Society for Industrial and Applied Mathematics, Philadelphia, PA 19104-2688, USA.

- Murray, R.M., K.J. Astrom, S.P. Boyd, R.W. Brockett and G. Stein, 2003. Future directions in control in an information-rich world. *IEEE Control Syst. Mag.*, 32 (2): 20-23.
- Naraendra, K.S. and K. Parthasarathy, 1990. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, 1 (1): 2-27.
- Nounou, H.N. and K.M. Passino, 2004. Stable auto-tuning of adaptive fuzzy-neural controllers for nonlinear discrete-time systems. *IEEE Trans. Fuzzy Syst.*, 12 (1): 70-83.
- Passino, K.M. and S. Yurkovich, 1998. *Fuzzy Control*. Addison Wesley Longman, Inc.
- Raun, D., 1997. *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms*. Kluwer Academic Publishers.
- Sales, K.R. and S.A. Billings, 1990. Self-tuning control of nonlinear ARMAX models. *Int. J. Control*, 51 (24): 753-469.
- Sanner, R.M. and J.J.E. Stoline, 1992. Gaussian networks for direct adaptive control. *IEEE Trans. Neural Networks*, 3 (6): 837-863.
- Xiaohong, J. and T. Shen, 2005. Adaptive feedback control of nonlinear time-delay systems: The LaSalle-Razumikhin-based approach. *IEEE Trans. Autom. Control*, 50 (11): 1909-1913.
- Wang, H., G.P. Liu and M. Brown, 1995. *Advanced Adaptive Control*. Elsevier Science, New York.
- Wu, J.K., 1994. *Neural Networks and Simulation Methods*. Electrical Engineering and Electronics A series of Reference Books and Textbooks, Marcel Dekker, Inc.