# Nonlinear system identification with a feedforward neural network and an optimal bounded ellipsoid algorithm

JOSÉ DE JESÚS RUBIO AVILA, ANDRÉS FERREYRA RAMÍREZ AND CARLOS AVILÉS-CRUZ
Departamento de Electrónica, Area de Instrumentación
Universidad Autónoma Metropolitana, Unidad Azcapotzalco.
Av. San. Pablo 180 Col. Reynosa Tamaulipas. Azcapotzalco, 02200 México D. F.
MÉXICO
jrubio@correo.azc.uam.mx, fra@correo.azc.uam.mx, caviles@correo.azc.uam.mx

*Abstract:* -Compared to normal learning algorithms, for example backpropagation, the optimal bounded ellipsoid (OBE) algorithm has some better properties, such as faster convergence, since it has a similar structure as the Kalman filter algorithm. Optimal bounded ellipsoid algorithm has some better properties than the Kalman filter training, one is that the noise is not required to be Guassian. In this paper optimal bounded ellipsoid algorithm is applied train the weights of a feedforward neural network for nonlinear system identification. Both hidden layers and output layers can be updated. In order to improve robustness of the optimal of the optimal bounded ellipsoid algorithm, dead-zone is applied to this algorithm. From a dynamic systems point of view, such training can be useful for all neural network applications requiring real-time updating of the weights. Two examples where provided which illustrate the effectiveness of the suggested algorithm based on simulations.

*Key-Words:* Neural Networks, Optimal Bounded Ellipsoid (OBE), Modeling, Identification.

## 1 Introduction

Recent results show that neural network technique seems to be very effective to identify a broad category of complex nonlinear systems when complete model information cannot be obtained. Neural networks can be classified as feedforward and recurrent ones [1]. Feedforward networks, for example multilayer perceptrons, are implemented for the approximation of nonlinear functions in the right hand side of dynamic plants. Even though backpropagation has been widely used as a practical training method for neural networks, there are some limitations such as slow convergence, local minima and sensitive to measurement noise.

Gradient-like learning laws are relatively slow. In order to solve this problem, many methods in the identification and filter fields have been proposed to estimate the weights of neural networks. For example extended Kalman filter is applied to train neural networks in [2],[3] and [4], they can give least-square solutions. Most of them use static neural networks. In [5] the output layer must be linear and the hidden layer weights are chosen at randomly. A faster convergence with the extended Kalman filter is reached with decoupling structure [6], however the computational complexity in each interaction is increased, it require of large amount of memory. Decoupled Kalman filter with diagonal matrix P in [7] has a similar algorithm with the gradient algorithm. A main drawback of the Kalman filter training is that theory analysis requires the uncertainty of neural modeling satisfies Gaussian process.

In 1979 L.G.Khachiyan indicated how an ellipsoid method for linear programming can be implemented in polynomial time [8]. This result has caused great excitement and stimulated a flood of technical papers. Ellipsoidal technique is an advantageous and helpful tool in state estimation of dynamic systems with bounded disturbances [9]. There are many potential applications to problems outside of the domain of linear programming. [10] obtained confidence ellipsoids which are valid for a finite number of data points. [11] presented an ellipsoidal propagation such that the new ellipsoid satisfies an affine relation with another ellipsoid. In [12], the ellipsoid algorithm is used as an optimization technique that takes into account the constraints on cluster coefficients. [13] described in detail several methods that can be used to derive an appropriate uncertainty ellipsoid for the array response. In [14], the problem concerning asymptotic behavior of ellipsoidal estimates is considered for linear discrete time systems. There are few application of ellipsoid on neural networks. In [15] unsupervised and supervised learning laws in the form of ellipsoids are used to find and tune the fuzzy function rules. In [16] ellipsoid type of activation function is proposed for feedforward neural networks.

Optimal bounding ellipsoid (OBE) algorithms offer an attractive alternative to traditional least-squares methods for identification and filtering problems involving affine-in-parameters signal and system models. The benefits include low computational efficiency, superior tracking ability, and selective updating that permits processor multi-tasking. In [17] multi weight optimization for OBE algorithms is introduced. In [18], a simple adaptive algorithm is proposed that estimates the magnitude of noise .To the best of our knowledge, neural networks training with the ellipsoid or the optimal bounded ellipsoid algorithm has not yet been established in the literature.

In this paper the optimal bounded ellipsoid algorithm is modified with dead-zone technique such that it can be used for training the weights of a feedforward neural network for nonlinear system identification. Both hidden layers and output layers can be updated. From a dynamic systems point of view, such training can be useful for all neural network applications requiring real-time updating of the weights.

# 2 Feedforward Neural Networks Modeling with OBE

Consider following unknown discrete-time nonlinear system:

$$y(k) = f[x(k)] \quad (1)$$

Where $x(k) = [y(k-1),...,y(k-n),u(k-1,...,u(k-m)] =$ $x(k) = [x_1(k)...x_N(k)] \in \Re^N$, (N=n+m) is the input vector, $|u(k)|^2 \le \bar{u}$, $y(k)$ is the output of the plant, $f$ is general nonlinear smooth function $f \in C^\infty$. We use the following feedforward neural network to identify the nonlinear plant (1)

$$\hat{y}(k) = V_k \sigma[W_k x(k)] \quad (2)$$

where $\hat{y}(k) \in \Re$ represents the output of the neural network. The weight in output layer is $V_k \in R^{1xM}$ , the weight in hidden layer is $W_k \in R^{MxN}$ , σ is M-dimension vector function $\sigma = [\sigma_1 ... \sigma_M]^T$.

$$\sigma[W_k x(k)] = [\sigma_1(\sum_{j=1}^{N} w_{1,j} x_j), \sigma_2(\sum_{j=1}^{N} w_{2,j} x_j), ...\sigma_M(\sum_{j=1}^{N} w_{M,j} x_j)]^T \quad (3)$$

where $\sigma_i$ is a sigmoid function. Thye model given in (2) is a series-parallel model given in [21]. According to the Stone-Weierstrass theorem [19], the unknown nonlinear system (1) can be written in the following form

$$y(k) = V_k \sigma[W_k x(k)] - \eta(k) \quad (4)$$

where $\eta(k)$ is unmodeled dynamic. By [19] we know that the term $\eta(k)$ can be made arbitrarily small by simply selecting appropriate the number of neurons in the hidden layer, in this paper, it is M. In the case of two independent variables, a smooth function $f$ has the following Taylor series expansion near the point $[x_1^0, x_2^0]$,

$$f = \sum_{k=0}^{l-1} \frac{1}{k!} \left[ (x_1 - x_1^0) \frac{\partial}{\partial x_1} + (x_2 - x_2^0) \frac{\partial}{\partial x_2} \right]_0^k f + \varepsilon \quad (5)$$

where ε is the remainder of the Taylor formula. If we let $x_1$ and $x_2$ correspond $W_k x(k)$ and $V_k, x_1^0, x_2^0$ correspond $W_1 x(k)$ and $V_1$, then we have

$$V_k \sigma[W_k x(k)] = V_1 \sigma[W_1 x(k)] + \sigma^T(\cdot) V_k^T + V_k \sigma'(\cdot) W_k x(k) + \varepsilon \quad (6)$$

where $V_1$ and $W_1$ are set of known initial constant weights. We define the modelling error as

$$\zeta(k) = \varepsilon + V_1 \sigma[W_1 x(k)] - \eta(k) \quad (7)$$

substituting (6) and (7) into (4) we have

$$y(k) = \sigma^T(\cdot) V_k^T + V_k \sigma'(\cdot) W_k x(k)] + \zeta_k \quad (8)$$

We pretend to rewrite (8) linear in parameters, but it is not straight, then we rewrite (8) in the following form:

$$y(k) = B^T{}_{i,k} \theta_{i,k} + \zeta_k \quad (9)$$

where $i = 1 \cdots N$ , $\theta_k = [\theta_1(k), \cdots \theta_N()]^T$, $\theta_i(k) = [V_k, W_{i,k}^T]^T \in R^{2Mx1}$ and $B_k^T = [B_{1,k}, \cdots B_{N,k}], B_{i,k}^T = [\sigma, x_1 V_k \sigma'] \in R^{1x2M}$

The output of the recurrent neural network (1) is defined as:

$$\hat{y}(k) = \sigma^T(\cdot) V_k + V_k \sigma'(\cdot) W_k x(k) \quad (10)$$

or linear in parameters:

$$\hat{y}(k) = B_{i,k}^T \theta_{i,k} \quad (11)$$

Denote the training error as :

$$e(k) = y(k) - \hat{y}(k) \quad (12)$$

Now we use $N$ optimal bounding ellipsoid algorithms (OBE) to train the feedforward neural network (10) such that the identification error $e(k)$ is bounded.

**Definition 1** *A real 2M-dimensional ellipsoid, centered on* $x^*$, *can be described as:*

$$E(x^*, P) = \left\{ x \in R^{2M} \middle| (x - x^*)^T P (x - x^*) \le 1 \right\}$$

where $P \in R^{2Mx2M}$ is a positive-semidefinite symmetric matrix.

The orientations of the ellipsoid are presented by the eigenvectors of P, they are $\left[u_1,...,u_{2M}\right]^T$. The axes are given by the eigenvalues of P, $\left[\lambda_1,...,\lambda_{2M}\right]^T$. Figure 1 shows the two dimensions case.

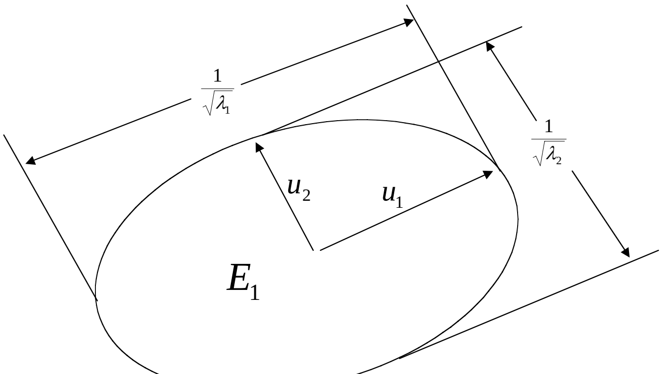


Figure 1: Ellipsoid set

**Definition 2** *The ellipsoid intersection of two ellipsoids* $E_a(x_1,P_1)$ *and* $E_b(x_2,P_2)$ *is another ellipsoid defined as* $E_c$,

$$E_c = \left\{x \in R \left| \lambda(x-x_1)^T P_1(x-x_1) + (1-\lambda)(x-x_2)^T P_2(x-x_2) < 1\right.\right\} := E_a \cap_e E_b$$

*where* $0 \le \lambda \le 1$, $P_1$ *and* $P_2$ *are positive-semidefinite symmetric matrices*

The normal intersection of the two ellipsoid sets $E_a \cap E_b$ is not an ellipsoid set in general. The ellipsoid set $E_c$ contains the normal intersection of ellipsoid sets, $E_a \cap E_b \subset E_c$. There exists a minimal ellipsoid set corresponding to $\lambda*$, see [11], [20] and [22]. In this paper, we will not try to find $\lambda*$, we will design a algorithm such that the new ellipsoid intersection will always be smaller. Figure 2 shows this idea.

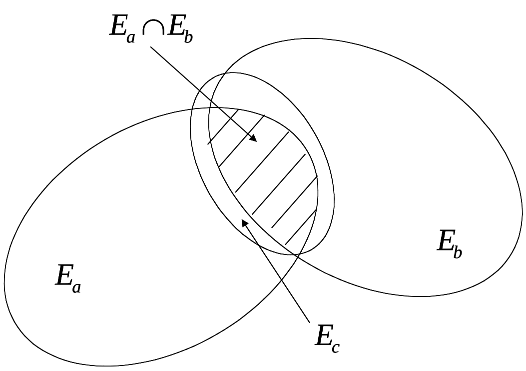


Figure 2: Ellipsoid set that contains the intersection of two ellipsoid sets.

Now we use the ellipsoid definition on neural identification, we define parameter error ellipsoid $E_k$ as

$$E_k = \left\{\theta_i(k) \left| \tilde{\theta}^T{}_i(k) P_k^{-1} \tilde{\theta}_i(k) \le 1\right.\right\} \tag{13}$$

where $\tilde{\theta}_i(k) = \theta_i * - \theta_i(k)$, $\theta_i *$ is the unknown optimal weight that minimize the modeling error $\zeta_k$ in (9), $P_k = P^T{}_k > 0$.

In this paper, we use the following two assumptions.

**A1** It is assumed that $\left[y(k) - B_{i,k}^T\theta_i *\right]$ belongs to an ellipsoidal set $S_k$,

$$S_k = \left\{B_{i,k}^T\theta_i^* \left| \frac{1}{\bar{\varsigma}^2}\left\|y(k) - B_{i,k}^T\theta_i^*\right\|^2 \le 1\right.\right\} \tag{14}$$

where $\bar{\varsigma} > 0$ is the known upper bound of the uncertainty $\varsigma_k$, $\left|\varsigma_k\right| < \bar{\varsigma}$. We can also rewrite (14) as

$$y(k) = B_{i,k}^T\theta_i^* + \varsigma_k \tag{15}$$

**A2** It is assumed that the initial weight errors are inside an ellipsoid $E_1$

$$E_1 = \left\{\theta_i(1) \left| \tilde{\theta}_i^T(1) P_1^{-1} \tilde{\theta}_i(1) \le 1\right.\right\} \tag{16}$$

where $P_1 = P_1^T > 0$, $P_1 = \in R^{2Mx2M}$, $\tilde{\theta}_i(1) = \theta_i^* - \theta_i(1)$, $\theta_i^*$ is the unknown optimal weights.

**Remark 1** *The assumption A1 requires that* $\left[y(k) - B_{i,k}^T\theta_i^*\right]$ *is bounded by* $\bar{\varsigma}$. *In this paper we are only interested in open-loop identification, we assume that the plant (1) is bounded-input and bounded-output (BIBO) stable, i.e.,* $x(k)$, $f[\cdot]$ *and* $y(k)$ *in (1) are bounded all of data are bounded, so* $\left[y(k) - B_{i,k}^T\theta_i *\right]$ *is bounded. The assumption A2 requires the initial weights of the neural networks are bounded, it can be satisfied by choosing suitable* $P_1$ *and* $\hat{\theta}_i(1)$.

We can see that the common part of the sets $S_1, S_2, \cdots$ is $\theta_i^*$, so

$$\left\{\theta_i^*\right\} \subset \bigcap_{j=1}^{k} \varsigma_j \tag{17}$$

Finding $\left\{\theta_i^*\right\}$ is an intractable task since the amount of information in (17) grows linearly in *k*. Moreover, evaluating a value of $\theta_i(k)$ in (17) involves the solution of *2k n-th* order inequalities. From the definition of $E_k$ in

(13), $\theta_i^*$ is the common center of the sets $E_1, E_2, \cdots$ is $\theta_i^*$, so

$$\{\theta_i^*\} \subset \bigcap_{j=1}^{k} E_j, \quad \{\theta_i^*\} \subset E_k \tag{18}$$

Thus the problem of identification is to find a set $E_k$ which satisfies (18). We will construct a recursive identification algorithm such that $E_{k+1}$ is the set corresponding to the set $E_k$ and the data $[y_i(k), B_{i,k}]$. Because the two ellipsoids satisfy (13) and (14), we calculate the ellipsoid intersection $(1-\lambda_{i,k})E_k + \lambda_{i,k}S_k$, it satisfies

$$\begin{aligned}
&(1-\lambda_{i,k})\tilde{\theta}_i^T(k)P_k^{-1}\tilde{\theta}_i(k) \le (1-\lambda_{i,k}) \\
&\frac{1}{\varsigma^2}\lambda_{i,k}\left\|y(k) - B_{i,k}^T\theta_i^*\right\|^2 \le \lambda_{i,k} \\
&(1-\lambda_{i,k})\tilde{\theta}_i^T(k)P_k^{-1}\tilde{\theta}_i(k) + \frac{1}{\varsigma^2}\lambda_{i,k}\left\|y(k) - B_{i,k}^T\theta_i^*\right\|^2 \le 1
\end{aligned} \tag{19}$$

The next theorem shows the propagation process of the ellipsoids.

**Theorem 1** If $E_k$ in (13) is an ellipsoidal set, we use the following recursive algorithm to update $P_k$ and $\theta_i(k)$

$$\begin{aligned}
&P_{k+1} = \frac{1}{1-\lambda_{i,k}}\left[P_k - P_k B_{i,k}\frac{\lambda_{i,k}}{(1-\lambda_{i,k})\varsigma^2 + \lambda_{i,k}B_{i,k}^T P_k B_{i,k}}B_{i,k}^T P_k\right] \\
&\theta_i(k+1) = \theta_i(k) + \frac{\lambda_{i,k}}{\varsigma^2}P_{k+1}B_{i,k}e(k) \\
&\lambda_{i,k} = \begin{cases} \dfrac{\lambda\varsigma^2}{1 + B_{i,k}^T P_k B_{i,k}} & if\, e^2(k) \ge \dfrac{\varsigma^2}{1-\lambda} \\ 0 & if\, e^2(k) < \dfrac{\varsigma^2}{1-\lambda} \end{cases}
\end{aligned}$$
(20)

where $P_1$ is a given diagonal positive definite matrix, $0 < \lambda < 1$, $0 < \lambda\overline{\varsigma}^2 < 1$, $0 < \lambda_{i,k} < 1$ and $(1-\lambda_{i,k}) > 0$, then $E_{k+1}$ is an ellipsoidal set and satisfies

$$E_{k+1} = \left\{\theta_i(k+1) \,\middle|\, \tilde{\theta}^T{}_i(k+1)P_{k+1}{}^{-1}\tilde{\theta}_i(k+1) \le 1 - \frac{\lambda_{i,k}}{\varsigma^2}[1-\lambda]e^2(k) \le 1\right\} \tag{21}$$

where $\tilde{\theta}_i(k) = \theta_i * - \theta_i(k)$ and $e(k)$ is given in (12)

**Proof.** First we apply matrix inversion lemma [9] to calculate $P_{k+1}^{-1}$ from (20) we have:

$$P_{k+1}^{-1} = (1-\lambda_{i,k}) * \left[P_k - P_k B_{i,k}\frac{\lambda_{i,k}}{(1-\lambda_{i,k})\varsigma^2 + \lambda_{i,k}B_{i,k}^T P_k B_{i,k}}B_{i,k}^T P_k\right]^{-1} \tag{21}$$

the matrix inversion lemma is [27]:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(DA^{-1}B + C^{-1}\right)^{-1}DA^{-1}$$

where A, B, C and D denote matrices of the correct size. It can be rewritten as:

$$\left(A^{-1} - A^{-1}B\left(DA^{-1}B + C^{-1}\right)^{-1}DA^{-1}\right)^{-1} = A + BCD$$

Specifically, $A \in \Re^{2M\times 2M}$, $B \in \Re^{2M\times 1}$, $C \in \Re^{1\times 1}$ and $D \in \Re^{1\times 2M}$, $A^{-1} = P_k$, $B = \frac{\lambda_{i,k}}{\varsigma^2}B_{i,k}$, $C^{-1} = (1-\lambda_{i,k})$ and $D = B_{i,k}^T$; then $P_{k+1}^{-1}$ is:

$$\begin{aligned}
P_{k+1}^{-1} &= (1-\lambda_{i,k})\left[P_k^{-1} + \frac{\lambda_{i,k}}{\varsigma^2}B_{i,k}\frac{1}{(1-\lambda_{i,k})}B_{i,k}^T\right] \\
&= (1-\lambda_{i,k})P_k^{-1} + \frac{\lambda_{i,k}}{\varsigma^2}B_{i,k}B_{i,k}^T
\end{aligned} \tag{22}$$

Now we calculate $\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)$ when $e^2(k) \ge \frac{\varsigma^2}{1-\lambda}$ by (20) or $\tilde{\theta}_i(k+1) = \tilde{\theta}_i(k) + \frac{\lambda_{i,k}}{\varsigma^2}P_{k+1}B_{i,k}e(k)$

$$\begin{aligned}
&\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1) = \tilde{\theta}_i^T(k)P_{k+1}^{-1}\tilde{\theta}_i(k) - \\
&2\frac{\lambda_{i,k}}{\varsigma^2}\tilde{\theta}_i^T(k)B_{i,k}e(k) + \frac{\lambda_{i,k}^2}{\varsigma^4}B_{i,k}^T P_{k+1}B_{i,k}e^2(k)
\end{aligned} \tag{23}$$

Substituting (22) into (23), it gives

$$\begin{aligned}
&\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1) = \\
&\tilde{\theta}_i^T(k)\left[(1-\lambda_{i,k})P_k^{-1} + \frac{\lambda_{i,k}}{\varsigma^2}B_{i,k}B_{i,k}^T\right]\tilde{\theta}_i(k) - \\
&2\frac{\lambda_{i,k}}{\varsigma^2}\tilde{\theta}_i^T(k)B_{i,k}e(k) + \frac{\lambda_{i,k}^2}{\varsigma^4}B_{i,k}^T P_{k+1}B_{i,k}e^2(k) \\
&= (1-\lambda_{i,k})\tilde{\theta}_i^T(k)P_k^{-1}\tilde{\theta}_i(k) + \frac{\lambda_{i,k}}{\varsigma^2}\tilde{\theta}_i^T(k)B_{i,k}B_{i,k}^T\tilde{\theta}_i(k) - \\
&2\frac{\lambda_{i,k}}{\varsigma^2}\tilde{\theta}_i^T(k)B_{i,k}e(k) + \frac{\lambda_{i,k}^2}{\varsigma^4}B_{i,k}^T P_{k+1}B_{i,k}e^2(k)
\end{aligned} \tag{24}$$

By the intersection property (19) of the ellipsoidal sets, we have:

$$(1-\lambda_{i,k})\tilde{\theta}_i^T(k)P_k^{-1}\tilde{\theta}_i(k) \le 1 - \frac{\lambda_{i,k}}{\varsigma^2}\left\|y(k) - B_{i,k}^T\theta_i^*\right\|^2$$

Then (24) becomes

$$\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)$$
$$\leq 1-\frac{\lambda_{i,k}}{\bar{\varsigma}^2}\left\|y(k)-B_{i,k}^T\theta_i^*\right\|^2+\frac{\lambda_{i,k}}{\bar{\varsigma}^2}\tilde{\theta}_i^T(k)B_{i,k}B_{i,k}^T\tilde{\theta}_i(k)-$$
$$2\frac{\lambda_{i,k}}{\bar{\varsigma}^2}\tilde{\theta}_i^T(k)B_{i,k}e(k)+\frac{\lambda_{i,k}^2}{\bar{\varsigma}^4}B_{i,k}^TP_{k+1}B_{i,k}e^2(k)$$
$$\leq 1+\frac{\lambda_{i,k}^2}{\bar{\varsigma}^4}B_{i,k}^TP_{k+1}B_{i,k}e^2(k)+$$
$$\frac{\lambda_{i,k}}{\bar{\varsigma}^2}\left[-\left\|y(k)-B_{i,k}^T\theta_i^*\right\|^2+\tilde{\theta}_i^T(k)B_{i,k}B_{i,k}^T\tilde{\theta}_i(k)-2\tilde{\theta}_i^T(k)B_{i,k}e(k)\right]$$

Now we use $\tilde{\theta}_i^T(k)=\theta_i^*-\theta_i(k)$; $e(k)=y(k)+B_{i,k}^T\theta_i(k)$ as in (12), the second term can be calculated as

$$-\left\|y(k)-B_{i,k}^T\theta_i^*\right\|^2+\tilde{\theta}_i^T(k)B_{i,k}B_{i,k}^T\tilde{\theta}_i(k)-2\tilde{\theta}_i^T(k)B_{i,k}e(k)=$$
$$-\left\|y(k)-B_{i,k}^T\theta_i^*\right\|^2+\left[\theta_i^*-\theta_i(k)\right]^TB_{i,k}B_{i,k}^T\left[\theta_i^*-\theta_i(k)\right]-$$
$$2\left[\theta_i^*-\theta_i(k)\right]^TB_{i,k}\left[y(k)-B_{i,k}^T\theta_i(k)\right]$$
$$=-\left[y^2(k)-2\theta_i^T(k)B_{i,k}^Ty(k)+\theta_i^T(k)B_{i,k}B_{i,k}^T\theta_i(k)\right]$$
$$=-\left[y(k)-B_{i,k}^T\theta_i(k)\right]^2=-e^2(k)$$

So

$$\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)\leq 1+\frac{\lambda_{i,k}^2}{\bar{\varsigma}^4}B_{i,k}^TP_{k+1}B_{i,k}e^2(k)-\frac{\lambda_{i,k}}{\bar{\varsigma}^2}e^2(k)$$

From (22) we know $P_{k+1}^{-1}=\left(1-\lambda_{i,k}\right)P_k^{-1}+\frac{\lambda_{i,k}}{\bar{\varsigma}^2}B_{i,k}B_{i,k}^T$. Because $\lambda_{i,k}=\frac{\lambda\bar{\varsigma}^2}{1+B_{i,k}^TP_kB_{i,k}}>0$; $P_{k+1}>0$ from $P_1>0$. By the updating algorithm (20):

$$\frac{\lambda_{i,k}^2}{\bar{\varsigma}^4}B_{i,k}^TP_{k+1}B_{i,k}e^2(k)=\frac{\lambda_{i,k}}{\bar{\varsigma}^4}\frac{\lambda\bar{\varsigma}^2}{1+B_{i,k}^TP_kB_{i,k}}B_{i,k}^TP_{k+1}B_{i,k}e^2(k)$$
$$\leq\frac{\lambda_{i,k}}{\bar{\varsigma}^2}\lambda e^2(k)$$

So

$$\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)\leq 1-\frac{\lambda_{i,k}}{\bar{\varsigma}^2}\left[1-\lambda\right]e^2(k) \quad (25)$$

(21) is established. Because $\lambda<1$ and $\lambda_{i,k}>0$

$$\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)\leq 1 \quad (26)$$

when $e^2(k)\geq\frac{\bar{\varsigma}^2}{1-\lambda}$ we have that $\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)$ is an ellipsoidal set. Now we consider $e^2(k)<\frac{\bar{\varsigma}^2}{1-\lambda}$, then $\lambda_{i,k}=0$, substituting in (20) we have $P_{k+1}=\frac{1}{1-0}\left[P_k-0\right]=P_k$, then $P_{k+1}^{-1}=P_k^{-1}$, $\theta_i(k+1)=\theta_i(k)+0$, then $\tilde{\theta}_i(k+1)=\tilde{\theta}_i(k)$, substituting in $\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)$ we have:

$$\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)=\tilde{\theta}_i^T(k)P_k^{-1}\tilde{\theta}_i(k)\leq 1$$

when $e^2(k)<\frac{\bar{\varsigma}^2}{1-\lambda}$ we have that $\tilde{\theta}_i^T(k+1)P_{k+1}^{-1}\tilde{\theta}_i(k+1)$ is an ellipsoidal set. Thus $E_{k+1}$ is an ellipsoidal set.

**Remark 2** *The ellipsoid intersection of $E_k$ and $S_k$ in (19), it is an ellipsoid also, defined as*

$$\Pi_k=\left\{z_i(k)\,\middle|\,\left[z_i(k)-\theta_i^*\right]^T\Sigma^{-1}\left[z_i(k)-\theta_i^*\right]\leq 1\right\}$$

*where $z_i(k)$ is an unknown variable. The ellipsoid intersection of $E_{k+1}$ and $S_{k+1}$ with the algorithm (20) is defined as*

$$(1-\lambda_{i,k+1})\tilde{\theta}_i^T(k+1)P_{k+1}{}^{-1}\tilde{\theta}_i(k+1)+\frac{1}{\bar{\varsigma}^2}\lambda_{i,k+1}\left\|y(k+1)-B_{i,k+1}^T\theta_i^*\right\|^2$$
$$\leq(1-\lambda_{i,k+1})\left[1-\frac{1}{\bar{\varsigma}^2}\lambda_{i,k+1}\left[1-\lambda\right]e^2(k)\right]+\lambda_{i,k+1}$$
$$=1-\frac{1}{\bar{\varsigma}^2}\lambda_{i,k}(1-\lambda_{i,k+1})\left[1-\lambda\right]e^2(k)$$

so

$$\Pi_{k+1}=\left\{z_i(k+1)\,\middle|\,\left[z_i(k+1)-\theta_i^*\right]^T\Sigma^{-1}\left[z_i(k+1)-\theta_i^*\right]\leq 1-\frac{(1-\lambda_{i,k+1})\lambda_{i,k}(1-\lambda)}{\bar{\varsigma}^2}e^2(k)\right\}$$

The volume of $\Pi_k$ is defined as [21],[9].

$$Vol(\Pi_k)=\sqrt{\det(\Sigma)U}$$

*where $U$ is constant represents the volume of the unit ball in $R$. Because $\lambda_{i,k}(1-\lambda_{i,k+1})(1-\lambda)e^2(k)>0$, the volume of $\Pi_{k+1}$ is less than the volume of $\Pi_k$ when $e(k)\neq 0$. From (17) and (18) we know, the common part of $\Pi_k$ and $\Pi_{k+1}$ are $\left\{\theta_i^*\right\}$. Thus the set $\Pi_k$ will convergent to the set $\left\{\theta_i^*\right\}$ when $e(k)\neq 0$, see Figure 3.*

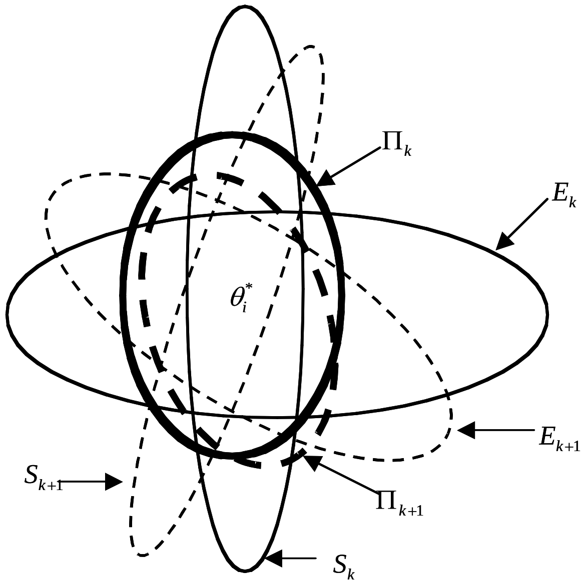


Figure 3: The convergence of the intersection $\Pi_k$.

**Remark 3** *The algorithm (20) is for each subsystem. This method can decrease computational burden when we estimate the weights of the recurrent neural network, the similar idea can be found in* [1] and [7]. *By (6) we know the data matrix* $B_k$ *depends on the parameters* $V_k^T$, *this will not effect parameter updating algorithm (20), because the unknown parameter* $\theta_i(k+1)$ *is calculated by the known parameters* $\theta_i(k)$ *and data* $B_k$. *For (20), we have*

$$
\begin{aligned}
V_{k+1} &= V_k + \frac{\lambda_k}{\overline{\varsigma}^2} P_{k+1} \sigma\left[W_k x(k)\right] e(k) \\
W_{k+1} &= W_k + \frac{\lambda_k}{\overline{\varsigma}^2} P_{k+1} \sigma'\left[W_k x(k)\right] V_k^T x^T(k) e(k)
\end{aligned}
\tag{27}
$$

*It has the same form as the backpropagation* [1], [23], [25], [26], [28], [29], [33], *but the learning rate is not positive constant, it is a matrix* $\frac{\lambda_k}{\overline{\varsigma}^2} P_{k+1}$ *which changes through time. That may be the reason why OBE algorithm is faster*.

**Remark 4** *The OBE algorithm (20) has the similar structure as the extended Kalman filter training algorithm* [6],[3] and [4]. *The extended Kalman filter algorithm is* [6], [27]:

$$
\begin{aligned}
\theta_i(k+1) &= \theta_i(k) + P_k B_{i,k} (R_2 - B_{i,k}^T P_k B_{i,k})^{-1} e(k) \\
P_{k+1} &= R_1 + \left[P_k - P_k B_{i,k} (R_2 - B_{i,k}^T P_k B_{i,k})^{-1} B_{i,k}^T P_k\right]
\end{aligned}
\tag{28}
$$

*where* $e(k)$ *is the same as in (12),* $R_1$ *can be chosen as* $\alpha I$, *where* $\alpha$ *is small and positive,* $R_2$ *is the covariance of `process noise'* . *When* $R_1 = 0$, *it becomes the least square algorithm* [27]. *If* $R_1 = 0$ *in (23), then* $(R_2 - B_{i,k}^T P_k B_{i,k})^{-1}$ *corresponds to* $\frac{\lambda_{i,k}}{(1-\lambda_{i,k})\overline{\varsigma}^2 + \lambda_{i,k} B_{i,k}^T P_k B_{i,k}}$ *in (20). There is a big difference, the OBE algorithm is for deterministic case and the extended Kalman filter is stochastic case.*

The following steps show how to train the weights of recurrent neural networks with the OBE algorithm:

1. Construct a recurrent neural networks model (2) to identify an unknown nonlinear system (1). The matrix A is selected such that it is stable.

2. Rewrite the neural network in linear form

$$
\begin{aligned}
&\hat{y}(k) = \sigma^T(\cdot) V_k^T + V_k \sigma'(\cdot) W_k x(k) \\
&\theta_k = \left[\theta_1(k), \cdots \theta_n(k)\right]^T, \quad \theta_i(k) = \left[V_k W_{i,k}^T\right]^T \\
&B_k^T = \left[B_{1,k}, \cdots B_{n,k}\right], \quad B_{i,k}^T = \left[\sigma, x_i V_k \sigma'\right] \in R^{1x2M}
\end{aligned}
$$

3. Train the weights as

$$
\theta_i(k+1) = \theta_i(k) + \frac{\lambda_{i,k}}{\overline{\varsigma}^2} P_{k+1} B_{i,k} e(k)
$$

$$
\lambda_{i,k} = \begin{cases} \dfrac{\lambda \overline{\varsigma}^2}{1 + B_{i,k}^T P_k B_{i,k}} & if \quad e^2(k) \geq \dfrac{\overline{\varsigma}^2}{1-\lambda} \\ 0 & if \quad e^2(k) < \dfrac{\overline{\varsigma}^2}{1-\lambda} \end{cases}
$$

4. is changed as OBE algorithm :

$$
P_{k+1} = \frac{1}{1-\lambda_{i,k}} \left[ P_k - P_k B_{i,k} \frac{\lambda_{i,k}}{(1-\lambda_{i,k})\overline{\varsigma}^2 + \lambda_{i,k} B_{i,k}^T P_k B_{i,k}} B_{i,k}^T P \right]
$$

With initial conditions for the weight $\theta_i(1)$ and $P_1 > 0$, we can start the system identification with the feedforward neural networks.

# 3 Simulation

In this section, the suggested on-line optimal bounded ellipsoid algorithm proposed is applied to nonlinear system identification.

**Example 1** Consider the nonlinear system given in [24] and [25]:

$$
y(k) = 0.52 + 0.1x_1 + 0.28x_2 - 0.06x_1x_2 \tag{29}
$$

with $x_1(k) = \sin^2(10/k)$ and $x_2(k) = \cos^2(10/k)$. We use the neural network given in (10) to identify the this nonlinear system, we use 10 nodes in the hidden layer, i.e.,

$v_k \in \Re^{1x10}$, $W_{1,k}, W_{2,k} \in \Re^{10x2}$, $\sigma = [\sigma_1, \sigma_2,, \cdots, \sigma_{10}]^T$. The initial weights $W_1$ and $V_1$ are chosen in random between (0,1) and (0,0.2) respectively. We select $P(1) = diag(100) \in \Re^{20x20}$, λ=0.9, $\bar{\varsigma}^2$=1x10$^{-6}$ such that λ $\bar{\varsigma}^2$<1. We compare the OBE training algorithm (20) with the standard backpropagation algorithm [1], [28]-[29], the learning rate for the backpropagation is 0.2. The identification results for $y(k)$ are shown in Figure 4. If we define the mean squared error for finite time as $J(T) = \frac{1}{2T}\sum_{k=1}^{T} e^2(k)$, the comparison results for the identification error are shown in Figure 5.

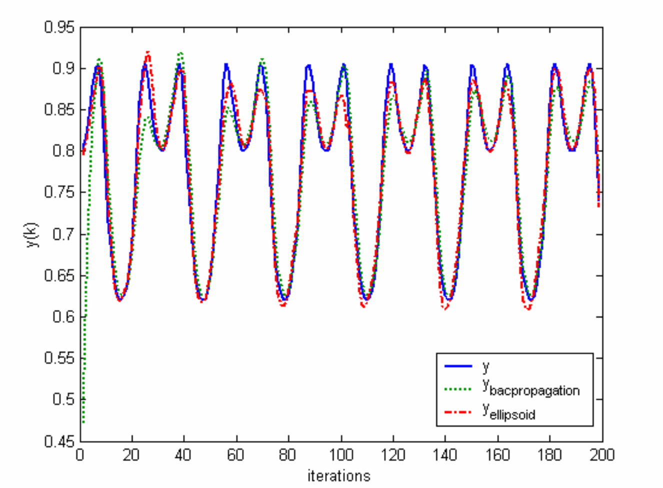


Figure 4: Identification results in example 1.

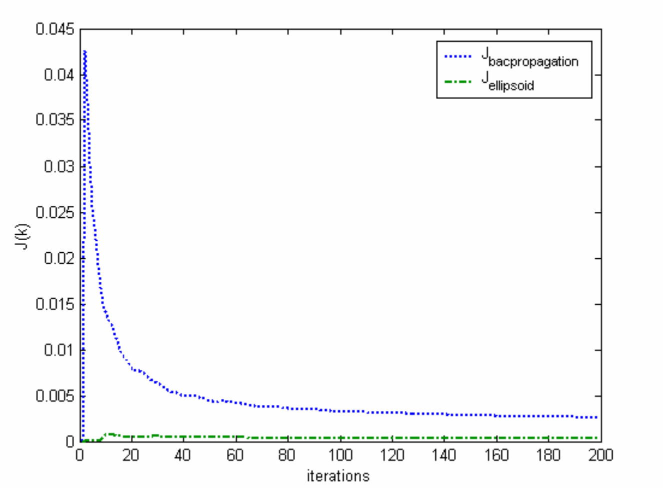


Figure 5: Identification error in example 1.

**Example 2** Consider the nonlinear system:

$$y(k) = \frac{y^2(k-3) + y^2(k-2) + y^2(k-1) + \tanh(u(k)) + 1}{y^2(k-3) + y^2(k-2) + y^2(k-1) + 1} \quad (30)$$

where $U(k) = 0.6\sin(9\pi kTs) + 0.2\sin(12\pi kTs) + 1.2\sin(3\pi kTs)$, $Ts = 0.01$, $x_1(k) = y(k-1)$, $x_2(k) = y(k-2)$, $x_3(k) = y(k-3)$, $x_4(k) = u(k)$. We use the neural network given in (10) to identify the this nonlinear system, we use 10 nodes in the hidden layer, i.e., $v_k \in \Re^{1x10}$, $W_{1,k}, W_{2,k} \in \Re^{10x4}$, $\sigma = [\sigma_1, \sigma_2,, \cdots, \sigma_{10}]^T$. The initial weights $W_1$ and $V_1$ are chosen in random between (0,1) and (0,0.2) respectively. We select $P(1) = diag(100) \in \Re^{20x20}$, λ=0.9, $\bar{\varsigma}^2$=1x10$^{-6}$ such that λ $\bar{\varsigma}^2$<1. We compare the OBE training algorithm (20) with the standard backpropagation algorithm [1], [28]-[29], the learning rate for the backpropagation is 0.1. The identification results for $y(k)$ are shown in Figure 6. If we define the mean squared error for finite time as $J(T) = \frac{1}{2T}\sum_{k=1}^{T} e^2(k)$, the comparison results for the identification error are shown in Figure 7.

We have that the OBE algorithm has better behavior than the backpropagation.

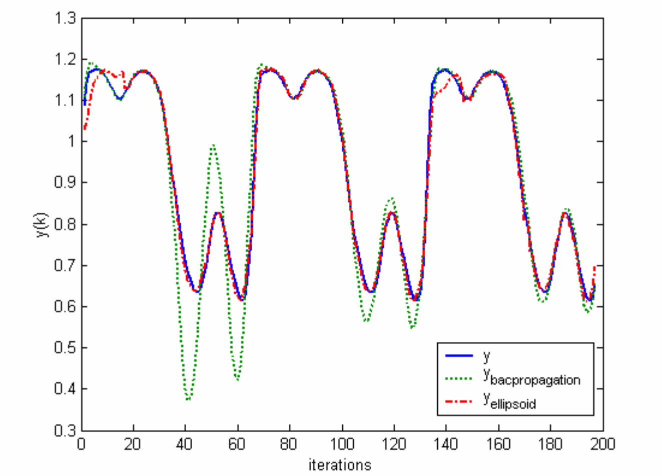


Figure 6: Identification results in example 2.

In the future we pretend to apply this identification algorithm for some real systems as the robotic systems [30] and the mechatronic systems [31], [33] or for pattern recognition [23], [32], [34].

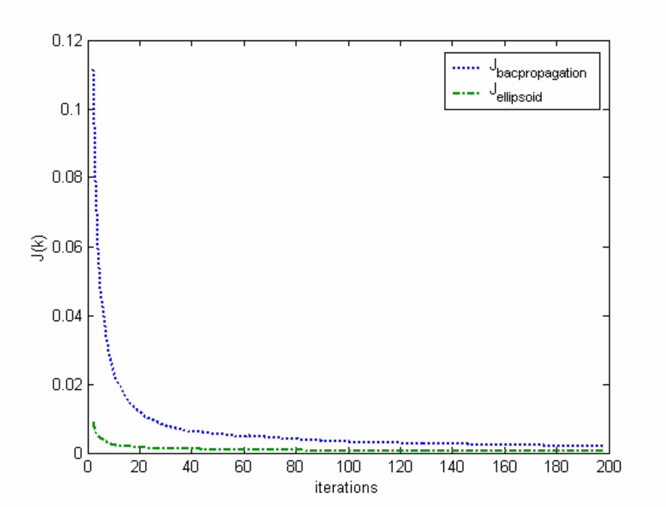

Figure 7: Identification error in example 2.

## 4 Conclusions

In this paper a novel training method for neural identification is proposed. We give a modified optimal bounded ellipsoid (OBE) algorithm for feedforward neural networks training. Both hidden layers and output layers of the neural network can be updated. From a dynamic system point of view, such training can be useful for all neural network applications requiring real-time updating of the weights. In the future we will prove the stability of the algorithm and we will apply this algorithm for identification of some nonlinear real systems as are the robotic or the mechatronic systems or for pattern recognition.

## 5 Acknowledgements

The authors like to express their gratitude to the editors for inviting us to participate in this project and to the reviewers for their valuable comments and suggestions which helped to improve this research.

*References:*

[1] S. Haykin, Neural Networks- A Comprehensive Foundation, Macmillan College Publ. Co., New York, 1994.
[2] A. G. Parlos, S. K. Menon and A. F. Atiya, An Algorithm Approach to Adaptive State Filtering Using Recurrent Neural Network, IEEE Transactions on Neural Networks, Vol.12, No.6, 1411-1432, 2001.
[3] D. W. Ruck, S. K. Rogers, M. Kabrisky, P. S. Maybeck and M. E. Oxley, Comparative Analysis of Backpropagation and the Extended Kalman Filter for Training Multilayer Perceptrons, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.14, No.6, 686-691, June 1992.
[4] S. Singhal and L. Wu, Training multilayer perceptrons with the extended Kalman algorithm, Advances in Neural inform. Processing Syst. I, 133-140, 1989.
[5] F. N. Chowdhury, A new approach to real-time training of dynamic neural networks, International Journal of Adaptive Control and Signal Processing, 509-521, 2003.
[6] J. J. Rubio and W. Yu, Nonlinear system identification with recurrent neural networks and dead-zone Kalman filter algorithm, Neurocomputing, Vol.70, 2460-2466, 2007.
[7] G. V. Puskorius and L. A. Feldkamp, Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks, IEEE Transactions on Neural Networks, Vol.5, No.2, 279-297, March 1994.
[8] R. G. Bland, D. Goldfarb, and M. J. Todd, The ellipsoid method: A survey, Operation Research, vol. 29, pp. 1039.1091, 1981.
[9] E. Fogel and Y. F. Huang, On the value of information in system identification: Bounded noise case, Automatica, vol. 18, no.2, pp. 229.238, 1982.
[10] E. Weyer, M. C. Campi, Non-asymptotic confidence ellipsoids for the least squares estimate, 9rd IEEE Conference on Decision and Control, Sydney, Australia, 2688-2693, 2000.
[11] L. Ros, A. Sabater and F. Thomas, An Ellipsoidal Alculus Based on Propagation and Fusion, IEEE Transactions on Systems, Man. And Cybernetics, Vol.32, No.4, 430-442, 2002.
[12] M. V. Correa, L. A. Aguirre, and R. R. Saldanha, Using Steady-State Prior Knowledge to Constrain Parameter Estimates in Nonlinear System Identification, IEEE Transactions on Circuits and Systems, Part I, Vol.49, No.9, 1376-1381, 2002.
[13] R. G. Lorenz, and S. P. Boyd, Robust Minimum Variance Beamforming, IEEE Trannsactions on signal processing, Vol. 53, No. 5, 1684-1696, 2005.
[14] S. A. Nazin, and B. T. Polyak. Limiting Behavior of Bounding Ellipsoids for State Estimation. Proceedings of the 5th IFAC Symposium on Nonlinear Control Systems. St.Petersburg, Russia, July 4-6, pp. 585-589, 2001.
[15] J. A. Dickerson, B. Kosko, Fuzzy function approximation with ellipsoidal rules, IEEE Transactions on Systems, Man. And Cybernetics, Vol.26, 542-560, Aug 1996.

[16] N. S. Kayuri, V. Vienkatasubramanian, Representing bounded fault classes using neural networks with ellipsoidal activation functions, Computers & Chemical Engineering, Vol. 17, no. 2, pp. 139-163. 1993

[17] D. Joachim, and J. R. Deller, Multiweight Optimization in Optimal Bounding Ellipsoid Algorithms, IEEE Transactions on singnal processing, VOL. 54, NO. 2, 679-690, 2006

[18] S. Gazor, and K. Shahtalebi, A New NLMS Algorithm for Slow Noise Magnitude Variation, IEEE Signal Processing Letters, Vol. 9, No. 11, 348-351, 2002.

[19] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou and P. A. Ioannou, High Order Neural Network Structures for Identification of Dynamical Systems, IEEE Transactions on Neural Networks, Vol.6, No.2, 422-431, 1995.

[20] S. Kapoor, S. Gollamudi, S. Nagaraj and Y. F. Huang, Tracking of Time-Varing Parameters using Optimal Bounding Ellipsoid Algorithms, Proc. of 34$^{th}$ Allerton Conference on Comm., Contr. And Computing., Monticello, Oct. 1996.

[21] K. S. Narendra and K. Parthasarathy, Identification and Control of Dynamical Systems using Neural Networks, IEEE Transactions on Neural Networks, Vol.1, No.1, 4-27, 1990.

[22] F. C. Schueppe, Uncertain dynamic systems, Englewood cliffs, Prentice-Hall, 1973

[23] R. Neruda, S. Slusny, Paremeter Genetic Learning of Perceptron Networks, WSEAS Transactions on Systems, Issue 10, Vol.5, 2285-2290, 2006.

[24]L. X. Wang, A Course in Fuzzy Systems and Control, Prentice Hall, Englewood Cliffs, NJ 07458, 1997.

[25] W. Yu and A. Ferreyra, On-line clustering for nonlinear system identification using fuzzy Neural networks, IEEE International Conference on Fuzzy Systems, 678-683, 2005.

[26] W. Yu, Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms, Information Sciences, Vol.158, No.1, 131-147, 2002.

[27] G. C. Goodwin and K. San Sin, Adaptive filtering prediction and control, Prentice Hall, Englewood Cliffis, NJ07632, 1984.

[28] J. R. Hilera, V. J. Martines, Redes neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, Adison Wesley Iberoamericana, U.S.A, 1995.

[29] J. S. R. Jang, C. T. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice Hall, NJ07458, 1996.

[30] J. J. Rubio Avila, R. Alcántara Ramirez, J. Jaimes Ponce, I. I. Siller Alcalá, Design of a low cost electronic control system for an articulated robot arm, International Journal of Mechanics, Issue 2, Vol.7, 21-26, 2007.

[31] J. J. Rubio Avila, R. Alcántara Ramirez, J. Jaimes Ponce, I. I. Siller Alcalá, Design, construction and control of a novel tower crane, International Journal of Mathematics and Computers in Simulation, Issue 2, Vol.7, 119-126, 2007.

[32] W. Wei, X. Wang, W. Neddermeyer, W. Winkler, M. Schnell, Application of an Automatic Fuzzy Logic Based Pattern Recognition Approach for Clustering DNA Microarray Data, WSEAS Transactions on Computers, Issue 5, Vol.5, 858-863, 2006.

[33] A. Bagheri, N. Amanifard, T. Karimi, H. Farahani, S. M. Besarati, Adaptive Neural Network Control of an Underwater Remotely Operated Vehicle (ROV), WSEAS Transactions on Information Science and Applications, Issue 10, Vol.3, 2002-2007, 2006.

[34] C. Aviles, J. Villegas, J. Ocampo, R. Arechiga, Principal Components and Invariant Moments Analysis-Font Recognition Applied, WSEAS Transactions on Computers, Issue 5, Vol.5, 1041-1046, 2006.

**José de Jesús Rubio Avila** was born in México City in 1979. He graduated in electrical engineering from the ESIME IPN in México in 2001. He received the master degree in automatic control in the CINVESTAV IPN in México (2004) and the doctor degree in automatic control in the CINVESTAV IPN in México (2007). Since 2006, he is a full time professor in the Autonomous Metropolitan University - Mexico City where he is a member of the researching group of intelligent systems and signal processing. He has published 5 chapters in books, 6 international papers and he has presented 14 papers in International Conferences. He is a member of the adaptive fuzzy systems task force. His research interests are primarily focused on evolving intelligent systems, nonlinear and adaptive control systems, neural-fuzzy systems, mechatronic and robotic systems, delayed systems.

**Andrés Ferreyra Ramírez** was born in México City in 1968. He received the professional degree of Electrical Engineering with speciality in Electronics and Communications in 1994 from the Escuela Nacional de Estudios Profesionales campus Aragón UNAM, in México. He received the M. S. degree in biomedical engineering in 2001 from the Universidad Autónoma Metropolita, Iztapalapa, in México. He received the Ph.D. in Automatic Control in 2005 from CINVESTAV IPN, in México. He has been a Professor in Electronics Engineering Department of the Universidad Autónoma Metropolitana, Azcapotzalco, Mexico City, since 1996, where he is currently the Coordinator of the researching group of intelligent systems and signal processing. He has published 6 international papers and he has been presented 24 papers in Internationals Conferences and nationals. His research has focused on connectionist evolving systems, neuro-fuzzy system, fuzzy systems, fuzzy control and fuzzy classifiers.

**Carlos Avilés Cruz** was born in Mexico City (march 13,1966). He is graduated in Electronics Engineering (1989) from the Autonomous Metropolitan University, He received the master degree in signal and image processing from the National Institute Polytechnic of Grenoble FRANCE (1993). He received the Doctor degree in signal and image processing from the National Institute Polytechnic of Grenoble- FRANCE (1997). Actually he is full time professor at the Autonomous Metropolitan University - Mexico City. His research interests are: computer vision, digital signal and image processing and High Order Statistics. He is author and co-author of about 30 nationals and internationals articles and also co-author of two books. He is also Member of the National Researcher System.