ROBUST ADAPTIVE IDENTIFICATION OF NONLINEAR SYSTEM USING NEURAL NETWORK

Q. Song, W.J.Hu and Y.C. Soh

ABSTRACT

It is well known that disturbance can cause divergence of neural networks in the identification of nonlinear systems. Sufficient conditions using so-called modified algorithms are available to provide guaranteed convergence for adaptive system. They are dead zone scheme, adaptive law modification, and $\sigma$-modification. These schemes normally require knowledge of the upper bound of the disturbance. In this paper, a robust weight-tuning algorithm is used to train the multi-layered neural network with an adaptive dead zone scheme. The proposed robust adaptive algorithm does not require knowledge of either the upper bound of the disturbance or the bound on the norm of the estimate parameter. A complete convergence proof is provided based on Lyapunov theorem to deal with the nonlinear system. Simulation results are presented to show good performance of the algorithm.

KeyWord: Neural networks, adaptive deadzone, nonlinear function identification.

I. INTRODUCTION

Recently, there has seen increasing interesting in the field of using neural networks for the identification and control of nonlinear systems. Adaptive and robust identification algorithms are particularly interesting being considered in the area of system identification and signal processing to cope with time variation of disturbance and system parameters [1-4]. Some theoretical results have been applied to convergence and stability analysis of neural networks for identification of nonlinear systems [5-7]. A major problem in the identification of nonlinear system is to deal with disturbance. It has been shown that unmodelled dynamics or even a small disturbance can cause most of the identification algorithms to become divergent [8]. Therefore, robustness of the identification algorithm has been an important topic in adaptive filter and control community for the past decade [4-8]. Most of the results focus on the modification of adaptive algorithms by eliminating the pure integral action of the adaptive laws to guarantee the boundedness of all signals in the adaptive loop [8]. One important method is called dead zone scheme, which simply turns off the adaptive algorithm when the identification error is smaller than a certain threshold [4]. This method has been extended to design robust adaptive algorithm of neural networks recently to overcome the disturbance problem because the neural network itself is sensitive to disturbance [9-10]. However, a fixed dead zone scheme needs the selection of an appropriate size of the dead zone, which requires the bound of the disturbance must be known a priori [11]. An over-sized dead zone would lead to a large identification error, while, the adaptive system may become divergent if the range of the dead zone is selected too small [9] (also refer to Remark 2 in section 2 and simulation results in section 4 of this paper).

In this paper, a robust adaptive dead zone scheme is proposed for the identification of nonlinear system using the multi-layered neural network. There are two major advantages of the new adaptive algorithm over the fixed dead zone scheme. Firstly, the selection of the dead zone does not require knowledge of the upper bound of the disturbance, which can include the reconstruction error of the neural network. Furthermore, the identification algorithm is particularly suitable for the adaptive system in term that the size of the dead zone is estimated on line. Any change of the system parameters or disturbance will not degrade the system performance because the size of the dead zone is updated by the adaptive algorithm accordingly.

II. MODELING OF THE NONLINEAR SYSTEM AND ROBUST ADAPTIVE ALGORITHM OF THE OUTPUT LAYER

Consider a class of nonlinear discrete-time system, the output $y(k) \in \mathbb{R}^m$ is given by:
y(k) = f(x^n(k)) + d(k) \tag{1}

with the input vector

\[ x^n(k) = [x^n_1(k) \ x^n_2(k) \ldots x^n_{m_1}(k)]^T \]

\[ = [y(k-1) \ y(k-2) \ldots y(k-p)] \]

\[ u(k-d) \ u(k-d-10 \ldots u(k-d-q)]^T. \tag{2} \]

where \( m_1 = p + q + 1 \) is the number of input neurons in the network, \( f(x^n(k)) \in \mathbb{R}^n \) is a nonlinear function, \( d(k) \in \mathbb{R}^n \) denotes a disturbance vector, \( p, q, \) and \( d \) are known integers that determine the system order and the time delay. Note that we use the notation \( \| \cdot \| \) for the Euclidean norm of vector and the Frobenius norm of matrix throughout this paper [10].

To simplify our analysis, a three-layered neural network is used for identification of the nonlinear system in this paper, which could also be extended to other multi-layered cases. The output of the neural network may be presented as

\[ y^n_0(k) = v^{m^T}(k)H(w^{m^*}(k), x^n(k)), \tag{3} \]

where \( v^{m^*}(k) = [v^{m^*}_1(k), v^{m^*}_2(k), \ldots v^{m^*}_{m_1}(k)]^T \in \mathbb{R}^{m_1 \times m} \) is the weight matrix of the output layer of the neural network, \( m_1 \) is the number of neurons in the hidden layer of the network, and \( H(w^{m^*}(k), x^n(k)) \in \mathbb{R}^m \) is the nonlinear activation function

\[ H(w^{m^*}(k), x^n(k)) = [h_{m_1}^{m^*}(k) \ h_{2}^{m^*}(m) \ldots h_{1}^{m^*}(k)]^T \]

\[ = [h(w^{m^T}(k)x^n(k)) \ h(w^{m^T}_2(k)x^n(k)) \ldots h(w^{m^T}_{m_1}(k)x^n(k))]^T. \tag{4} \]

The adjustable weight matrix \( w^{m}(k) \in \mathbb{R}^{m \times m^*} \) of the hidden layer

\[ w^{m}(k) = [w^{m}_{11}(k), w^{m}_{12}(k), \ldots w^{m}_{m_1 m_2}(k)]^T, \]

where \( w^{m}_{ji}(k) = [w^{m}_{j1}(k) \ w^{m}_{j2}(k) \ldots w^{m}_{jm_2}(k)]^T \in \mathbb{R}^{m_2} \) with \( j = 1, 2, \ldots m_1 \). \( w^{m}_{ji}(k) \) is the adjustable weight from the \( ij \)th neuron of the input layer to the \( j \)th neuron of the hidden layer, and \( h(.) \) is the nonlinear activation function

\[ h(x) = \frac{1}{1 + e^{-\lambda x}} \tag{6} \]

where \( \lambda > 0 \) is the so-called gain parameter of the threshold function.

Define a bounded ideal weight \( v^{m^*} \in \mathbb{R}^{m_1 \times m_2} \) in the output layer of the neural network, we have the desired output of the network

\[ y^{*}(k) = v^{m^T}(k)H(w^{m^*}, x^n(k)), \tag{7} \]

where \( w^{m^*} \in \mathbb{R}^{m_1 \times m_2} \) is also a bounded desired weight matrix of the hidden layer as

\[ w^{m^*} = [w^{m^*}_1, w^{m^*}_2, \ldots w^{m^*}_{m_1}]^T, \tag{8} \]

with \( w^{m}_{ji} \in \mathbb{R}^{m_2} \)

\[ w^{m}_{ji}(k) = [w^{m}_{j1}(k), w^{m}_{j2}(k) \ldots w^{m}_{jm_2}(k)]^T (j = 1, 2, \ldots m_1). \tag{9} \]

Such that \( f(x^n(k)) \) can be presented by the neural network as

\[ f(x^n(k)) = y^{*}(k) + \varepsilon(k) = v^{m^T}H(w^{m^*}, x^n(k)) + \varepsilon(k), \tag{10} \]

where \( \varepsilon(k) \in \mathbb{R}^{m} \) is a reconstruction and measurement error vector.

**Remark 1.** The reconstruction error \( \varepsilon(k) \) is usually assumed to be a disturbance with a known bound according to the well-known universally approximate property of neural networks. However, the bounds of the reconstruction error, and even the disturbance \( d(k) \), may be difficult to decide in the adaptive system design. Therefore, in this paper we do not use knowledge of the upper bound in the robust adaptive algorithm.

The neural network estimate error \( e(k) \in \mathbb{R}^{m} \) is

\[ e(k) = y(k) - y^n_0(k) = y^{*}(k) - y^n_0(k) + \varepsilon(k) + d(k) \]

\[ = v^{m^T}(k)H(w^{m^*}, x^n(k)) + v^{m^T}H^{\varepsilon}(k) + \varepsilon(k) + d(k), \tag{11} \]

with \( H^{\varepsilon}(k) \in \mathbb{R}^{m_1} \)

\[ H^{\varepsilon}(k) = H(w^{m^*}, x^n(k)) - H(w^{m^*}, x^n(k)), \tag{12} \]

and \( v^{m^*} = v^{m^*} - \varepsilon(k) \in \mathbb{R}^{m_1 \times m_2} \) is the weight error matrix of the output layer.

To achieve global convergence we can define an adaptive dead zone function as

\[ g(e(k), \Delta^n_{w}(k)) = \begin{cases} e(k) - \| \Delta^n_{w}(k) \| & \text{if } \| e(k) \| > \Delta^n_{w}(k) \\ 0 & \text{if } \| e(k) \| \leq \Delta^n_{w}(k) \end{cases} \tag{13} \]

where \( \Delta^n_{w} \) is the adaptive dead zone parameter to be estimated.

We define the cost function as

\[ E(k) = e^{T}(k)e(k)/2. \tag{14} \]
To update the weight a normalized gradient descent algorithm can be written in the matrix form

\[ v^m(k+1) = v^m(k) - \frac{\partial E(k)}{\partial v^m(k)} \alpha(k) \left( 1 + \left| H(w^m(k), x^m(k)) \right|^2 \right)^{-1}, \]

where \( \alpha(k) \) is a variable learning rate based on the dead zone scheme defined as

\[ \alpha(k) = \begin{cases} 
  g(e(k), \Delta^m(k)) & \text{if } e(k) > \Delta^m(k) \\
  0 & \text{if } e(k) \leq \Delta^m(k).
\end{cases} \]

Note the initial value of \( \Delta^m(k) \) is selected as a positive number in this paper to make sure that \( \Delta^m(k) \) is a positive variable.

**Theorem 1.** The weight-tuning algorithm in equation (15) and (16) using the dead zone scheme has the following properties

\[ |v^m(k+1)| \leq |v^m(k)| \text{ for } k = 0, 1, \ldots. \]

\[ |\Delta^m(k+1)| \leq |\Delta^m(k)| \text{ for } k = 0, 1, \ldots. \]

\[ \limsup_{k \to \infty} |e(k)| \leq \Delta^m. \]

where \( \Delta^m \) is a positive constant, and \( \Delta^m(k) = \Delta^m - \Delta^m(k) \).

**Proof.** From Equation (11) we have

\[ |e(k)|^2 = e^T(k) \sigma^m(k) H(w^m(k), x^m(k)) + e^T(k) \tau^m(k) \]

\[ \leq \text{tr} \{ e^T(k) \sigma^m(k) H(w^m(k), x^m(k)) \} + |e(k)| \left| \tau^m(k) \right|. \]

with

\[ \left| \tau^m(k) \right| = \left| v^m H^T(k) + d(k) + e(k) \right| \leq \Delta^m. \]

Define a Lyapunov function

\[ V^m(k) = \left| v^m(k) \right|^2 + \left| \Delta^m(k) \right|^2 = \text{tr} \{ \sigma^m(k) \sigma^m(k) \} + \Delta^m (k) \Delta^m(k). \]

Then combine equations (15), (16) and (23) we have

\[ V^m(k+1) - V^m(k) \leq \frac{(2 \alpha(k) \left| e(k) \right|^2 - \left( -2 \alpha(k) \right) \left| e(k) \right| \left| \Delta^m(k) - \Delta^m(k) \right|)}{1 + \left| H(w^m(k), x^m(k)) \right|^2} \]

\[ + \frac{\left| H(w^m(k), x^m(k)) \right|^2}{1 + \left| H(w^m(k), x^m(k)) \right|^2} \left( \left| e(k) \right|^2 - \left| \Delta^m(k) - \Delta^m(k) \right|^2 \right) \]

\[ \leq \frac{-2g^m(e(k), \Delta^m(k)) \left| e(k) \right|^2 - g^m(e(k), \Delta^m(k))}{1 + \left| H(w^m(k), x^m(k)) \right|^2} + \frac{g^m(e(k), \Delta^m(k))}{1 + \left| H(w^m(k), x^m(k)) \right|^2} \]

\[ = - \frac{g^m(e(k), \Delta^m(k))}{1 + \left| H(w^m(k), x^m(k)) \right|^2}. \]

Note that the first inequality above is from equation (21), the second inequality is from equation (22), and the last inequality is from the definition in equation (13).

Then \( V^m(k) \) defined in equation (23) is bounded blow by zero and is not increasing. The limit of \( V^m(k) \) exists and the last item in equation (24) will go to zero. Since \( H(w^m(k), x^m(k)) \) is a bounded signal, we have

\[ \limsup_{k \to \infty} |e(k)| = \limsup_{k \to \infty} |\Delta^m(k)| \leq \Delta^m. \]

This completes the proof.

**Remark 2.** From the second inequality, the Lyapunov convergence condition can be re-written as:

\[ \alpha(k) \left| e(k) \right| \geq \alpha(k) \left| e(k) \right| \]

This is not generally true for a fixed learning rate \( \alpha(k) \). Therefore, a variable learning rate like the one defined in equation (17) is used in this paper to provide the guaranteed convergence. Please refer to [9] for further details.
Remark 3. A major advantage of the robust adaptive dead zone algorithm, compared to the fixed dead zone scheme in [9-10], is that knowledge of the upper bound $\Delta^n$ is not used in the design of the adaptive system. Furthermore, since the adaptive dead zone parameter $\Delta^n(k)$ is converged to $\Delta^n$, the estimate error $e(k)$ is approximating to a minimum value in the presence of disturbance.

III. ROBUST TRAINING ALGORITHM FOR THE HIDDEN LAYER

To update $w^n(k)$ a robust back-propagation algorithm can be written in the matrix form

$$w^n(k+1) = w^n(k) - \frac{\partial E(k)}{\partial w^n(k)} + \frac{\beta(k)}{1 + \|H^n(k)w^n(k)\|e^T(k)}$$

$$= w^n(k) + \frac{\beta(k)H^n(k)w^n(k)e(k)x^n(k)}{1 + \|H^n(k)w^n(k)\|e^T(k)}$$

$$\Delta^n(k+1) = \Delta^n(k) + \frac{\beta(k)e^T(k)}{1 + \|H^n(k)w^n(k)\|e^T(k)}$$

(26)

(27)

where $\beta(k)$ is a variable learning rate based on the dead zone scheme defined as

$$\beta(k) = \begin{cases} g(e(k), \Delta^n(k)) \| e(k) \| \quad \text{if} \quad \frac{\Delta^n}{\Delta_{\min}} \geq e(k) > \Delta^n(k) \\ 0 \quad \text{if} \quad \frac{\Delta^n}{\Delta_{\min}} \leq e(k) \leq \Delta^n(k) \end{cases}$$

(28)

with

$$H^n(k) = \text{diag} [h^n_1(k), \ldots, h^n_{m_1}(k)]$$

$$= \text{diag} [h(w^n_1(k), x^n(k)), \ldots, h(w^n_{m_1}(k), x^n(k))] \in \mathbb{R}^{m_1 \times m},$$

where $h^n_i(k)$ is the first derivative of the threshold function in equation (6), and $h_{\min} = \min[h^n_1(k), \ldots, h^n_{m_1}(k)] > 0$ is a positive constant.

Lemma 1. We have

$$\|e(k)\|^2 \leq \frac{\Delta^n}{\Delta_{\min}} \text{tr} \{H^n(k)w^n(k)e(k)x^n(k)w^n^T(k)\} + e^T(k)e(k),$$

(29)

where $e(k)$ is an equivalent disturbance signal.

Proof. Note that the activation function $h(.)$ in Equation (6) is non-decreasing function. There exists a unique positive number $\mu(.)$ with $\Lambda \geq \mu(.) \geq 0$ so that the system estimation error $e(k)$ in equation (11) can be rewritten as the following

$$e(k) = f(x^n(k)) - y_{m}(k) + \epsilon(k) + d(k)$$

$$= v^n_1(k)(H(w^n_1, x^n(k))) - H(w^n_1, x^n(k)) + \epsilon(k)$$

$$= v^n_1(k)[h(w^n_1, x^n(k)) - h(w^n_1, x^n(k)) + \epsilon(k)]$$

$$\ldots h(w^n_{m_1}, x^n(k)) - h(w^n_{m_1}, x^n(k)) + \epsilon(k)]$$

$$\ldots \mu(w^n_{m_1}, x^n(k))(w^n_{m_1} - w^n_{m_1}(k)x^n(k)) + \epsilon(k)$$

$$= v^n_1(k)[w^n_{m_1}(k)x^n(k)] + \epsilon(k)$$

(30)

where $w^n_{m_1} = w^n_1 - w^n_1(k) \in \mathbb{R}^{m \times m_1}$ is the weight error matrix, and $\mu(k) \in \mathbb{R}^{m \times m_1}$ is a unique positive matrix

$$\mu(k) = \text{diag} [\mu_1^n, \ldots, \mu_{m_1}^n]$$

$$= \text{diag} [\mu(w^n_{m_1}, x^n(k)), \ldots, \mu(w^n_{m_1}, x^n(k))].$$

(31)

Then from equation (30) we have

$$e^T(k)e(k) = e^T(k)v^n_1(k)x^n(k)\mu(k)v^n_1(k)x^n(k) + e^T(k)e(k)$$

$$= [\mu_1^n e^T(k)v^n_1(k)x^n(k)\mu_1^n(k) + \ldots ]$$

$$+ [\mu_1^n e^T(k)v^n_1(k)x^n(k)\mu_1^n(k) + \ldots ]$$

$$+ [\mu_1^n e^T(k)v^n_1(k)x^n(k)\mu_1^n(k) + \ldots ]$$

$$+ e^T(k)e(k)$$

(32)

where $v^n_1(k) = [v_1^n(k), \ldots, v_{m_1}(k)] \in \mathbb{R}^{m \times m_1}$ with $v^n_{m_1}(k) \in \mathbb{R}^m$.

Rewrite the definition in (3) as $y_{m}(k) = [y_1^n(k), \ldots, y_{m_1}(k)]^T \in \mathbb{R}^m$, and use the differential rule for vector, we have

$$-\frac{\partial E(k)}{\partial v_1^n(k)} = -e^T(k)\frac{\partial e(k)}{\partial v_1^n(k)} = e^T(k)\frac{\partial y_1^n(k)}{\partial v_1^n(k)} = e^T(k)\frac{\partial y_1^n(k)}{\partial v_1^n(k)}$$

(33)
\[
\begin{bmatrix}
    v_1''(k)h_1''(k)x_1''(k) & \cdots & v_m''(k)h_m''(k)x_m''(k) \\
    \vdots & \ddots & \vdots \\
    v_1''(k)h_1''(k)x_1''(k) & \cdots & v_m''(k)h_m''(k)x_m''(k)
\end{bmatrix}
\]

\[
= \beta(k) \left( \begin{array}{c}
    v_1''(k)h_1''(k)x_1''(k) \\
    \vdots \\
    v_m''(k)h_m''(k)x_m''(k)
\end{array} \right)
\]

\[
= h_j''(k)\varepsilon(k)v_m''(k)x_m''(k)
\]

where \[\frac{\partial v_m''(k)}{\partial w_{ij}''(k)}\] is Jacobian matrix with \(1 \leq i \leq m_z\), \(1 \leq j \leq m_1\), and \(1 \leq o \leq m\). Replace \(h_j''(k)\varepsilon(k)v_m''(k)x_m''(k)\) in the last item of (32) with \(-\varepsilon(k)\), we have

\[
e^T(k)e(k) = \mu(k) \left( \frac{-\varepsilon(k)}{h_{1j}(k)} \right) \cdot w_m''(k) + \ldots
\]

\[
+ \mu(k) \left( \frac{-\varepsilon(k)}{h_{m1}(k)} \right) \cdot w_m''(k) + \ldots
\]

\[
+ \left( \frac{-\varepsilon(k)}{w_{ij}''(k)} \right) \cdot w_m''(k) + \ldots
\]

\[
= \beta \cdot \text{tr} \left[ H''(k)w''(k)e(k)x_m''(k)w_m''(k) \right] + e^T(k)e(k)
\]

Note that the inequality above is due to the fact \(-\varepsilon(k)\) is assumed to be a scalar.

This completes the proof.

**Remark 4.** The nonlinear gradient descent algorithm in (26) will search for the edge of an attractor basin randomly as shown in the figure below [12]. The proposed adaptive dead zone scheme can provide guaranteed convergence as long as a local minimum point is concerned. For further improvement, the higher order gradient algorithm can be used as shown in [13], which may provide a global convergence point and is an interesting topic for future research.

\[\left[ \begin{array}{c} w_m''(k) \\
\end{array} \right] \leq \left[ \begin{array}{c} \tilde{w}_m''(k) + 1 \\
\end{array} \right] \quad \text{for} \quad k = 0, 1, \ldots
\]

\[\left[ \begin{array}{c} \tilde{\Delta}_m''(k) \\
\end{array} \right] \leq \left[ \begin{array}{c} \Delta_m''(k) + 1 \\
\end{array} \right] \quad \text{for} \quad k = 0, 1, \ldots
\]

\[\lim_{k \to \infty} e(k) \leq \Delta_m''.
\]

where \(\tilde{w}_m''(k) = w_m'' - w_m(k), \tilde{\Delta}_m''(k) = \Delta_m'' - \Delta_m''(k).

**Proof.** Define a Lyapunov function

\[V_m''(k) = \left[ w_m''(k) \right]^2 + \left[ \Delta_m''(k) \right]^2
\]

\[= \text{tr} \left[ \tilde{w}_m''(k)\tilde{w}_m''(k) \right] + \tilde{\Delta}_m''(k)\tilde{\Delta}_m''(k)
\]

where \(\tilde{\Delta}_m''(k) = \Delta_m'' - \Delta_m''(k).

According to Lemma 1 we have the similar proof procedure as in Theorem 1, i.e.

\[V_m''(k + 1) - V_m''(k)
\]

\[\leq -2\beta(k)h_m'' \left\{ \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} + \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right. \]

\[+ \left. \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right\} + \Delta_m'' - \Delta_m''(k)
\]

\[+ \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right\} + \Delta_m'' - \Delta_m''(k)
\]

\[\leq -2\beta(k)\left\{ \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} + \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right. \]

\[+ \left. \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right\} + \Delta_m'' - \Delta_m''(k)
\]

\[+ \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right\} + \Delta_m'' - \Delta_m''(k)
\]

\[\leq -2\beta(k)\left\{ \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} + \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right. \]

\[+ \left. \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right\} + \Delta_m'' - \Delta_m''(k)
\]

\[+ \frac{\varepsilon(k)}{H''(k)v_m''(k)x_m''(k)} \right\} + \Delta_m'' - \Delta_m''(k)
\]
\[ \begin{aligned}
&\leq (-2)\frac{h_m^*}{\lambda} \left( g_{\omega m}(k) \left| e(k) \right| - \Delta \omega(k) \lambda / h_m^* \right) \\
&+ \frac{g_{\omega m}(k)}{1 + \left( H_{\omega m}(k) v_m(k) \left\| x_m(k) \right\| \right)^2} \\
&= - \frac{g_{\omega m}(k)}{1 + \left( H_{\omega m}(k) v_m(k) \left\| x_m(k) \right\| \right)^2}
\end{aligned} \tag{37} \]

Furthermore, if the input vector \( x_m(k) \) is bounded, then equation (35) follows similar to the proof in Theorem 1.

This completes the proof. \( \blacksquare \)

**VI. SIMULATION RESULTS**

The simulation is based on a single-input single-output nonlinear time-series model with the delayed output signal as the input as proposed in [14]:

\[ \begin{aligned}
y(k) &= (0.8 - 0.5\exp(-y^2(k - 1)))y(k - 1) \\
&\quad - (0.3 + 0.9\exp(-y^2(k - 1)))y(k - 2) \\
&+ 0.1\sin(3.1415926y(k - 1)) + \eta(k)
\end{aligned} \tag{38} \]

where \( \eta(k) \) is a normally distributed random number with the bound \( \left| \eta(k) \right| \leq \Delta = 0.2 \).

A three-layered feed-forward neural network with two linear input neurons, five hidden nonlinear neurons and one linear output neuron was set up for the simulation. A neural network trained with the fixed dead zone scheme [10] is used for the identification of the nonlinear plant. Assume that knowledge of the disturbance is unknown and the size of the fixed dead zone is set to be very small. This makes the fixed dead zone scheme is effectively equivalent to a standard back-propagation algorithm [10]. After 300 epoch training, the time response of the neural network is shown in Fig. 1, which does not match the output of the plant closely. This is explained by the plot of cost function of the neural network, also shown in Figure 1, that the network does not converge to an optimal point due to the disturbance.

In contrast to the fixed dead zone algorithm, the adaptive robust training algorithm proposed in sections 2 and 3 can be used to train the neural network objectively to reject disturbance. Figure 2 shows the time response of the neural network, using the adaptive robust training algorithm in the presence of the disturbance with the amplitude \( \Delta = 2 \). The performance is improved in terms of both the time response and the quadratic error cost function using the adaptive robust training algorithm as shown in Fig. 2, compared to the results of the fixed dead zone algorithm in Fig. 1.

![Fig. 1. Time response and cost function of the neural network using the mismatched fixed dead zone training algorithm: Dashed line - output of neural network; Solid line - output of the plant.](image1)

![Fig. 2. Time response and cost function of the neural network using the adaptive robust training algorithm: Dashed line - output of neural network; Solid line - output of the plant.](image2)

**V. CONCLUSION**

The robust adaptive training algorithm provides guaranteed convergence of the neural network for the identification of nonlinear system without knowledge of the upper bound of disturbance. The gradient descent algorithm has been modified with a variable learning rate, which is referred as the adaptive dead zone scheme. Unlike a conventional dead zone scheme, the range of dead zone is determined by adaptive iteration algorithm on line. A complete convergence proof of the robust adaptive
algorithm for the three-layered neural network is provided based on the Lyapunov stability theory. Simulation results shown that the robust adaptive algorithm guarantees the convergence property of the proposed algorithm for identification of the nonlinear system in the presence of disturbance.

REFERENCES


Qing Song received the B.S. and the M.S. degrees from Harbin Shipbuilding Engineering Institute and Dalian Maritime University, China in 1982 and 1986, respectively. He obtained the Ph.D. degree from the Industrial Control Centre at Strathclyde University, U.K. in 1992. He was an assistant professor at the Dalian maritime University from 1986 to 1987. He worked as a research engineer at the System and Control pte Ltd, U.K., before joined the School of Electrical and Electronic Engineering, Nanyang Technological University as a lecturer in 1992. He is now a senior lecturer.

Dr. Qing Song is the first inventor of a neural network patent and was the recipient of the Tan Chin Tuan Academic Exchange Fellowship in 1997. He is currently and active industrial consultant. He has been the principal investigator of a few neural network related research programs targeted for industry, such as nonlinear control and pattern recorgnition systems.

Wenjie Hu was born in P.R. China. He received the B.S. degree from Zhe Jiang University, P.R. China. He is currently supported by a research scholarship to pursue Ph.D. degree in the school of EEE, Nanyang Technological University (NTU), Singapore. His research interests include support vector machine, neural networks and their applications.

Yeng Chai Soh received the B. Eng. (Hons. I) degree in electrical and electronic engineering from the University of Canterbury, new Zealand, in 1983, and the Ph.D. degree in electrical engineering from the University of Newcastle, Australia, in 1987.