Robust Backstepping Control of Induction Motors Using Neural Networks

C. M. Kwan and F. L. Lewis

Automation and Robotics Research Institute
The University of Texas at Arlington
7300 Jack Newell Blvd. S.
Fort Worth, TX 76118.

Research supported by NSF grant IRI-9216545

Nomenclature

R_s	stator resistance
R_r	rotor resistance
i_s	stator current
ψ_s	stator flux linkage
i_r	rotor current
ψ_r	rotor flux linkage
ψ_d	magnitude of rotor flux linkage
u	voltage input
ω	angular speed
ρ	angle between the flux linkages in the rotor
n_p	number of pole pairs
δ	angle of rotation
L_s	stator inductance
L_r	rotor inductance
M	mutual inductance
σ	$1- M^2/(L_sL_r)$
J	rotor inertia
T_L	load torque
T	electric motor torque
$(\bullet)_d, (\bullet)_q$	$(\bullet)$ in the (d,q) frame
$(\bullet)_a, (\bullet)_b$	$(\bullet)$ in the (a,b) frame
$(\bullet)^r$	reference trajectory
$\hat{(\bullet)}$	estimate of $(\bullet)$
σ_1, σ_2	sliding variables
$(\bullet)_N$	nominal values of $(\bullet)$
α	R_r/L_r
β	$M/((\sigma L_sL_r)$
γ	$M^2R_r/(\sigma L_sL_r^2)+ R_s/(\sigma L_s)$
μ	$n_pM/(JL_r)$

Abstract

In this paper, we present a new robust control technique for induction motors using neural networks (NN). New tuning schemes are proposed which *can guarantee the boundedness of tracking error and weight updates*. A main advantage of our method is that we *do not require the regression matrix*, so that no preliminary dynamical analysis is needed. Another salient feature of our NN approach is that *no off-line learning phase is needed.* Full state feedback is needed for implementation. Load torque and rotor resistance can be unknown but bounded.

1. Introduction

The induction motor is quite popular for fixed-speed applications. Since rotor currents are induced, no brushes and slip rings are needed. It is maintenance free, simple in operation, rugged and generally less expensive than either DC or synchronous motors [2]. On the other hand its model is much more complicated than other machines and because of this, it is considered as "**the benchmark problem in nonlinear systems**" by the Editorial Board of IEEE Transactions on Automatic Control in comments on a recent paper [16].

There are many approaches to induction motor control such as [1] [16] [21] and references therein.

Neural networks (NN) have been applied to system identification [4] or identification-based control [3] [18], little about the use of NN in direct closed-loop controllers that yield guaranteed performance. Some results on the application of NN to robot are presented in [9] [12] [14] [15]. Recently many NN controllers have been proposed for various control applications [19] [20].

Problems that remain to be addressed in NN research include ad hoc controller structures and the inability to guarantee satisfactory performance of the system in terms of small tracking errors and bounded NN weights. Uncertainty on how to initialize the NN weights leads to the necessity for "preliminary off-line tuning" [3] [5].

In the recent adaptive and robust control literature there has been a tremendous amount of activity on a special control scheme known as "backstepping" [10] [11]. When used under some mild assumptions, many existing robust and adaptive control techniques can be extended to wide classes of applications. Dawson *et al.* [7] have applied such techniques to various kind of robotic control schemes with the inclusion of motor dynamics. A major problem with backstepping approaches is that certain functions must be "linear in the unknown parameters", and some very tedious analysis is needed to determine a "regression matrix".

In this paper, we will use neural nets at each stage of the backstepping procedure to estimate certain nonlinear functions. This means that *linearity in the parameters is not needed, and no regression matrix need be found.* Thus, a major problem with backstepping is cured. Compared with other NN approaches, the NN weights here are tuned on-line, with *no learning phase required.* Most importantly, we can *guarantee the boundedness of tracking error and weight updates.* When compared with adaptive controllers, we *do not require persistent excitation conditions.*

The paper is organized as follows. We will first describe a fifth-order model in Marino's paper [16] and then an equivalent field oriented model in Section 2. In Section 3 we will develop a robust NN control scheme without the PE requirement. We first treat certain signals in the system as fictitious control signals to a simpler subsystem. Two-layer NN is used in this stage to design the fictitious controllers. Then we apply a second two-layer NN to robustly realize the fictitious NN signals. Our method is modular in nature and hence can be applied to other nonlinear systems with similar structures. Load torque and rotor resistance can be unknown but bounded. Full state is needed for implementation. Theory as well as simulations in Section 3.2 and 3.3 show that our NN control scheme works very well.

2. Models of Induction Motor

2.1 Induction Motor Model

A fifth-order model, which includes rotor dynamics, under the assumptions of equal mutual inductance and linear magnetic circuit, is given by Marino *et al.* [16]

$$\frac{d\omega}{dt} = \frac{n_pM}{JL_r}(\psi_{ra}i_{sb} - \psi_{rb}i_{sa}) - \frac{T_L}{J},$$
$$\frac{d\psi_{ra}}{dt} = -\frac{R_r}{L_r}\psi_{ra} - n_p\omega\psi_{rb} + \frac{R_r}{L_r}Mi_{sa},$$
$$\frac{d\psi_{rb}}{dt} = -\frac{R_r}{L_r}\psi_{rb} + n_p\omega\psi_{ra} + \frac{R_r}{L_r}Mi_{sb}, \qquad (2.1)$$
$$\frac{di_{sa}}{dt} = \frac{MR_r}{\sigma L_sL_r^2}\psi_{ra} + \frac{n_pM}{\sigma L_sL_r}\omega\psi_{rb} - \left(\frac{M^2R_r+L_r^2R_s}{\sigma L_sL_r^2}\right)i_{sa} + \frac{1}{\sigma L_s}u_{sa},$$
$$\frac{di_{sb}}{dt} = \frac{MR_r}{\sigma L_sL_r^2}\psi_{rb} - \frac{n_pM}{\sigma L_sL_r}\omega\psi_{ra} - \left(\frac{M^2R_r+L_r^2R_s}{\sigma L_sL_r^2}\right)i_{sb} + \frac{1}{\sigma L_s}u_{sb}.$$

i, ψ, u_s denote current, flux linkage and stator voltage input to the machine. The meaning of other symbols in (2.1) are listed in the Nomenclature.

2.2 Field Oriented Model

This technique was introduced by Blaschke [1]. It involves a transformation from the stator fixed frame (a,b) to a frame (d,q), which rotates along the flux vector (ψ_a,ψ_b). The transformations between currents and flux magnitudes in different frames are given by

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\rho & \sin\rho \\ -\sin\rho & \cos\rho \end{bmatrix}\begin{bmatrix} i_a \\ i_b \end{bmatrix}, \qquad (2.2)$$
$$\begin{bmatrix} \psi_d \\ \psi_q \end{bmatrix} = \begin{bmatrix} \cos\rho & \sin\rho \\ -\sin\rho & \cos\rho \end{bmatrix}\begin{bmatrix} \psi_a \\ \psi_b \end{bmatrix} \qquad (2.3)$$

where

$$\rho = \tan^{-1}\left(\frac{\psi_b}{\psi_a}\right).$$

A state transformation was suggested by Blaschke [1] as

$$\omega = \omega, \qquad \psi_d = \sqrt{\psi_a^2+\psi_b^2},$$
$$\rho = \tan^{-1}\left(\frac{\psi_b}{\psi_a}\right), \qquad i_d = \frac{\psi_a i_a+\psi_b i_b}{\psi_d},$$
$$i_q = \frac{\psi_a i_b-\psi_b i_a}{\psi_d}. \qquad (2.4)$$

Substituting (2.2), (2.3) and (2.4) into (2.1) yields the field oriented model

$$\frac{d\omega}{dt} = -\frac{T_L}{J} + \mu\psi_d i_q, \qquad (2.5a)$$
$$\frac{d\psi_d}{dt} = -\alpha\psi_d + \alpha M i_d, \qquad (2.5b)$$
$$\frac{di_d}{dt} = -\gamma i_d+\alpha\beta\psi_d+n_p\omega i_q+\alpha M\frac{i_q^2}{\psi_d}+\frac{1}{\sigma L_s}u_d, \qquad (2.5c)$$
$$\frac{di_q}{dt} = -\gamma i_q-n_p\omega\beta\psi_d-n_p\omega i_d-\alpha M\frac{i_q i_d}{\psi_d}+\frac{1}{\sigma L_s}u_q, \qquad (2.5d)$$
$$\frac{d\rho}{dt} = n_p\omega + \alpha M\frac{i_q}{\psi_d} \qquad (2.5e)$$

with

$$\alpha = R_r/L_r, \qquad \gamma = M^2R_r/(\sigma L_sL_r^2)+R_s/(\sigma L_s),$$
$$\beta = M/((\sigma L_sL_r), \qquad \mu = n_pM/(JL_r).$$

Since R_r is unknown, α, γ are also unknown. However, β and μ are known. Blaschke [1] also developed a feedback linearization plus PI controller to control (2.5). Marino *et al.* [16] went a step further to use adaptive input-output decoupling technique to tackle the control problem. In the next section, we will make use of a special structure of the above model to perform our NN controller design.

3. Robust Control of Induction Motor Using Neural Networks

By looking at (2.5), one will notice that there exists a very special structure in it. (2.5a) to (2.5d) can be considered as two nonlinear systems in cascade. i_d, i_q can be treated as the outputs of subsystem (2.5c)-(2.5d). At the same time, they can also be treated as fictitious inputs to the subsystem (2.5a)-(2.5b). It is this special structure that we will exploit in our NN controller design. Before we go into the details, two general assumptions and some mathematical preliminaries are needed:

Assumption 1: The reference trajectories ω^r and ψ_d^r are differentiable and bounded.

Assumption 2: The load torque T_L and rotor resistance R_r are unknown but bounded.

3.1 Preliminaries

Neural Networks

Given an input vector x in R^{N_1}, a three-layer neural net (NN) has an output given by

$$y_i = \sum_{j=1}^{N_2}\left[w_{ij}\sigma\left[\sum_{k=1}^{N_1} v_{jk}x_k + \theta_{vj}\right] + \theta_{wi}\right] \qquad i = 1, \ldots, N_3 \qquad (3.1)$$

with $\sigma(\bullet)$ the activation function, v_{jk} the first-to-second layer interconnection weights, and w_{ij} the second-to-third layer interconnection weights. θ_{vm}, θ_{wm}, m = 1,2,..., are called the threshold offsets and the number of neurons in layer l is N_l, with N_2 the number of hidden-layer neurons. A three layer neural network is shown in Fig. 3.1.

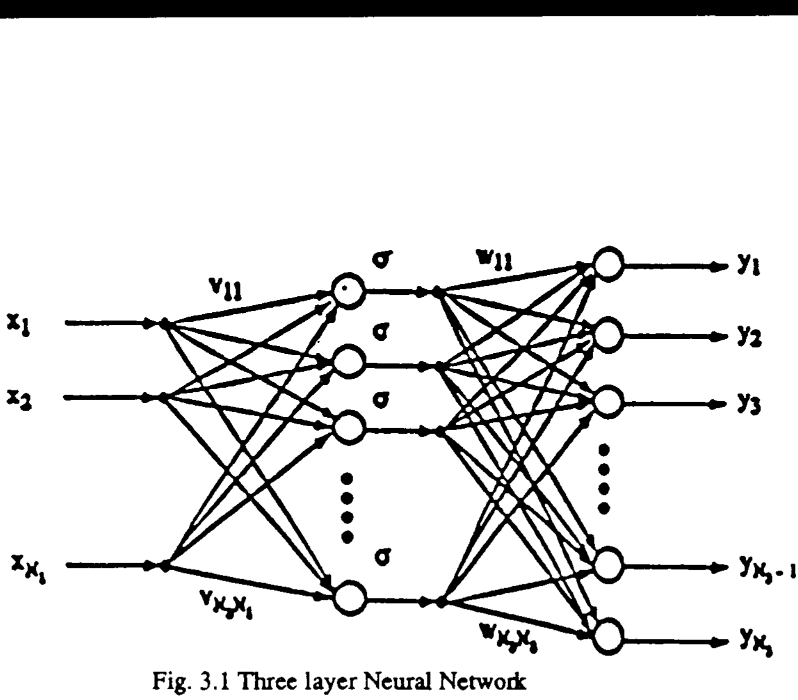

Fig. 3.1 Three layer Neural Network

The NN equation may be conveniently expressed in matrix format by defining $x = [x_0\ x_1\ ...\ x_{N_1}]^T$, $y = [y_1\ y_2\ ...\ y_{N_3}]^T$, and weight matrices $W^T = [w_{ij}]$, $V^T = [v_{jk}]$. Including $x_0 = 1$ in x allows one to include the threshold vector $[\theta_{v1}\ \ \theta_{v2}\ \ ....\ \theta_{vN_1}]$ as the first column of V^T. Hence the NN outputs can be compactly written as

$$y = W^T\sigma(V^Tx) \qquad (3.2)$$

where, if $z = [z_1\ \ z_2\ \ ...]^T$ is a vector we define $\sigma(z) = [\sigma(z_1)\ \ \sigma(z_2)...\ ]^T$. Including 1 as a first term in the vector $\sigma(V^Tx)$ allows one to incorporate the thresholds θ_{wj} as the first column of W^T. Any tuning of W and V then includes tuning of the thresholds as well. Basically the functional approximation property of NN is used in this paper [6] [8].

Stability of Systems

Consider the following nonlinear system

$$\dot{x} = f(x,u,t), \qquad y = h(x,t). \qquad (3.3)$$

with state $x(t) \in R^n$. We say the solution is uniformly ultimately bounded (UUB) if there exists a compact set $U \subset R^n$ such that for all $x(t_0) = x_0 \in U$, there exists an $\varepsilon > 0$ and a number $T(\varepsilon, x_0)$ such that $\|x(t)\| < \varepsilon$ for all $t \geq t_0 + T$.

3.2 Robust Backstepping Controller Design

In the remainder of the paper we consider the NN for the case of fixed first-layer weights V. The use of two-layer NN with no backstepping has been applied to several occasions [15] [20]. Here, we consider general basis functions $\phi(x)$ and propose various weight tuning algorithms with and without the requirement of persistent excitation (PE) on certain signals in the system. The tuning algorithms generally need persistent excitation for suitable performance. A modified tuning algorithm is then proposed to make the NN controller *robust* so that PE is not needed.

Define $\phi(x) = \sigma(V^Tx)$ so that the net output is

$$y = W^T\phi(x). \qquad (3.4)$$

Then, for suitable NN approximation properties, $\phi(x)$ must satisfy some conditions. Take $N_1 = n$, $N_3 = m$.

Definition 3.1 [20]

Let S be a compact simply connected set of R^n, and $\phi(x)$: $S \rightarrow R^{N_2}$ be integrable and bounded. Then ϕ is said to provide a **basis** for $C^m(s)$ if

1. A constant function on S can be expressed as (3.4) for finite N_2.
2. The functional range of NN (3.4) is dense in $C^m(s)$ for countable N_2.

The issue of selecting σ and V so that ϕ provides a basis, as well as the further issue of selecting N_2 for a given $S \subset R^n$ and ε_N, are topics of current research. One possibility is to use the radial basis functions.

3.2.1 Controller Structure

We first treat i_d, i_q as the ideal fictitious control signals for a subsystem consisting of (2.5a) - (2.5b). Then we use a second 2-layer NN to realize these fictitious signals.

Step 1: Selection of desired i_d and i_q to control subsystem (2.5a) and (2.5b)

Our control objective is to regulate the rotor speed and the magnetic flux magnitude. Denote ω^r and $\psi_d{}^r$ as the desired reference levels of ω and ψ_d respectively. First, we rewrite (2.5a-b) as

$$\frac{de_1}{dt} = -\frac{T_L}{J} + \mu\psi_d i_q, \qquad (3.5a)$$

$$\frac{de_2}{dt} = -\alpha\psi_d + \alpha M i_d \qquad (3.5b)$$

where

$$e_1 = \omega - \omega^r, \qquad e_2 = \psi_d - \psi_d^r.$$

Since α in (3.5b) is unknown, this will cause some difficulties of using the results of backstepping NN [13]. The difficulty is due to that fact that the stability analysis will be much more complicated if α is unknown. To alleviate this difficulty, we apply the following trick. Dividing both sides of (3.5b) by αM yields

$$\frac{1}{\alpha M}\frac{de_2}{dt} = -\frac{1}{M}\psi_d + i_d. \qquad (3.6)$$

Now the coefficient of i_d is unity and known. Then we can express (3.5) as

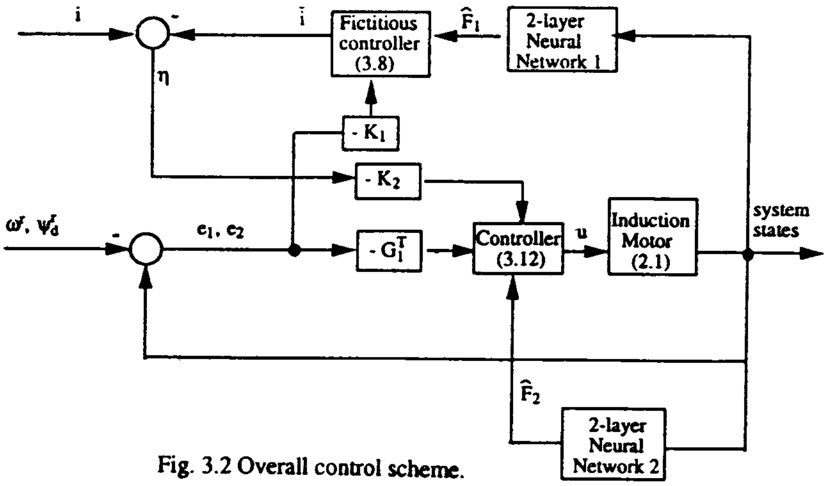

Fig. 3.2 Overall control scheme.

$$D\dot{e} = F_1 + G_1 i \tag{3.7}$$

where

$$e = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}, \quad i = \begin{bmatrix} i_q \\ i_d \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \\ 0 & 1/(\alpha M) \end{bmatrix},$$

$$F_1 = \begin{bmatrix} -T_L/J \\ 0 \end{bmatrix}, \quad G_1 = \begin{bmatrix} \mu\psi_d & 0 \\ 0 & 1 \end{bmatrix}.$$

It should be noted that G_1 is exactly known and invertible. By treating i as a fictitious input, we design a controller for the ideal $\bar{i}$ as

$$\bar{i} = G_1^{-1}[-\hat{F}_1 - E - K_1 e], \qquad K_1 > 0 \quad (3.8)$$

with K_1 a design parameter, $E = [0 \quad \psi_d/M]^T$, $\hat{F}_1$ the estimate of F_1. Substituting (3.8) into (3.7) gives

$$D\dot{e} = F_1 - \hat{F}_1 - K_1 e + G_1\eta \tag{3.9}$$

where $\eta = i - \bar{i}$. The form of $\hat{F}_1$ will be discussed in the next section. The usual approach is by assuming F_1 to be linear parametrizable (LP) so that standard adaptive control can be used in this stage [10] [11]. As we will see in a moment, we will use a two-layer NN method to approximate F_1. The advantage is that no linearity in the unknown parameters is needed and no regression matrix need be found.

Step 2: Realization of the desired signals in (3.8).

In order to achieve the desirable result in Step 1, i.e. the ideal fictitious control signal in (3.8), we need to find the error dynamics of η which is defined as

$$\eta = i - \bar{i}. \tag{3.10}$$

Differentiating (3.10) and using the dynamics in (2.5) yields

$$\dot{\eta} = F_2 + F_2^{known} + G_2 u \tag{3.11}$$

where

$$F_2^{known} = \begin{bmatrix} -\gamma i_q - n_p\omega\beta\psi_d - n_p\omega i_d \\ -\gamma i_d + n_p\omega i_q \end{bmatrix} + G_1^{-1}\dot{\hat{F}}_1,$$

$$F_2 = \begin{bmatrix} -\alpha M \dfrac{i_q i_d}{\psi_d} \\ \alpha\beta\psi_d + \alpha M \dfrac{i_q^2}{\psi_d} \end{bmatrix} + \dot{G}_1^{-1}(\hat{F}_1 + K_1 e) + G_1^{-1}K_1 D^{-1}[F_1 - \hat{F}_1 - K_1 e + G_1\eta],$$

$$G_2 = \frac{1}{\sigma L_s}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

To make η as small as possible, the following control u is chosen as

$$u = G_2^{-1}[-F_2^{known} - \hat{F}_2 - K_2\eta - G_1^T e]. \tag{3.12}$$

Note that $\hat{F}_2$ is the estimate of the unknown function F_2. Similar to Step 1, the usual way of design is to assume F_2 to be LP. However, in our controller design here, we will use another two-layer NN to approximate F_2 which means no regression matrix requirement is needed. Also note that a term $-G_1^T e$ is added in (3.12) which is necessary to cancel the effect of $G_1\eta$ in (3.9) so that we will be able to prove the closed-loop stability.

Step 3: Closed-loop Stability Analysis and Weight Tuning Algorithms

We will perform a detailed treatment of stability and performance analysis of two weight tuning algorithms in the next few sections. The overall control scheme is shown in Fig. 3.2.

Finally $u = [u_q \quad u_d]^T$ is related to the actual control u_a, u_b in (2.1) through the following relation

$$\begin{bmatrix} u_a \\ u_b \end{bmatrix} = \frac{1}{\psi_d}\begin{bmatrix} \psi_a & -\psi_b \\ \psi_b & \psi_a \end{bmatrix}\begin{bmatrix} u_d \\ u_q \end{bmatrix}. \tag{3.13}$$

3.2.2 Bounding Assumptions

Assume that the nonlinear functions F_1, F_2 in (3.7) and (3.11) can be represented as two-layer neural nets for some constant "ideal" weights W, i.e.

$$F_1 = W_1^T\phi_1 + \varepsilon_1, \qquad \|\varepsilon_1\| < \varepsilon_{1N} = \text{constant}$$
$$F_2 = W_2^T\phi_2 + \varepsilon_2, \qquad \|\varepsilon_2\| < \varepsilon_{2N} = \text{constant} \tag{3.14}$$

where $\phi_1(x)$, $\phi_2(x)$, provide suitable basis functions. The net reconstruction error $\varepsilon_i(x)$ is bounded by a known constant ε_{iN}, i =1, 2.

Define the NN functional estimate of F_1 in (3.7) by

$$\hat{F}_1 = \hat{W}_1^T\phi_1, \tag{3.15}$$

Then the error dynamics (3.9) becomes

$$D\dot{e} = \tilde{W}_1^T\phi_1 - K_1 e + G_1\eta + \varepsilon_1. \tag{3.16}$$

Similarly, define the NN functional estimate of F_2 in (3.11) by

$$\hat{F}_2 = \hat{W}_2^T\phi_2. \tag{3.17}$$

The error dynamics of η in (3.11) is then given by

$$\dot{\eta} = \tilde{W}_2^T\phi_2 - K_2\eta - G_1^T e + \varepsilon_2. \tag{3.18}$$

Note that there is a term $G_1\eta$ in (3.16) and a term $-G_1^Te$ in (3.18). This means there are couplings between the error dynamics (3.16) and (3.18). The closed-loop stability analysis and the weight tuning algorithms will be discussed in next section.

An additional standard assumption, which is quite common in the neural networks literature [14] [15], is stated next.

Assumption 3: The ideal weights are bounded by known positive values so that

$$\|W_1\|_F \le W_{1M},\ \|W_2\|_F \le W_{2M},\ \text{or}\ \|Z\|_F \le Z_M$$

where $Z = \mathrm{diag}\{W_1,\ W_2\}$.

3.2.3 Weight Updates

The result is summarized in the following theorem.

Theorem 3.1: Let the desired trajectories be bounded. Take the control input (3.12) with NN weight tuning be provided by

$$\dot{\hat{W}}_1 = \Gamma_1\phi_1 e^T - k_v\Gamma_1\|\zeta\|\hat{W}_1 \quad (3.19a)$$

$$\dot{\hat{W}}_2 = \Gamma_2\phi_2\eta^T - k_v\Gamma_2\|\zeta\|\hat{W}_2 \quad (3.19b)$$

with any constant matrices $\Gamma_1 = \Gamma_1^T > 0$, $\Gamma_2 = \Gamma_2^T > 0$, and scalar positive constant k_v. Then the errors e(t), $\eta(t)$ are UUB. NN weight estimates are bounded. The errors e(t), $\eta(t)$ can be kept as small as desired by increasing gains K in (3.12).

Proof: Define

$$\zeta = [e^T\ \eta^T]^T, \qquad \tilde{W}_i = W_i - \hat{W}_i,$$

$$\tilde{Z} = \mathrm{diag}\{\tilde{W}_1\ \tilde{W}_2\}, \qquad \Gamma = \mathrm{diag}\{\Gamma_1, \Gamma_2\}.$$

Let the NN approximation property (i.e. equations (3.14) holds with given accuracy ε_{iN}'s for all ζ in the compact set $U_\zeta \equiv \{\zeta \mid \|\zeta\| \le b_\zeta\}$ with b_ζ a positive constant. Let $\zeta(0) \in U_\zeta$. Now consider the following Lyapunov function candidate for error systems (3.9) and (3.11)

$$V = \frac{1}{2}\zeta^T D\zeta + \frac{1}{2}\mathrm{tr}(\tilde{Z}^T\Gamma^{-1}\tilde{Z}) \quad (3.20)$$

where $D > 0$ is defined in (3.7). Differentiating (3.20) and using (3.12), (3.16), (3.18), (3.19) we have

$$\dot{V} = -\zeta^T K\zeta + k_v\|\zeta\|\,\mathrm{tr}\{\tilde{Z}^T(Z - \tilde{Z})\} + \zeta^T\varepsilon. \quad (3.21)$$

Since $\mathrm{tr}\{\tilde{Z}(Z-\tilde{Z})\} = \langle\tilde{Z},Z\rangle_F - \|\tilde{Z}\|_F^2 \le \|\tilde{Z}\|_F\|Z\|_F - \|\tilde{Z}\|_F^2$, there results

$$\dot{V} \le -K_{min}\|\zeta\|^2 + k_v\|\zeta\|\,\|\tilde{Z}\|_F(Z_M - \|\tilde{Z}\|_F) + \varepsilon_N\|\zeta\|$$

$$= -\|\zeta\|\,[K_{min}\|\zeta\| + k_v\|\tilde{Z}\|_F(\|\tilde{Z}\|_F - Z_M) - \varepsilon_N] \quad (3.22)$$

which is negative as long as the term in square bracket is positive. Completing the square yields

$$K_{min}\|\zeta\| + k_v\|\tilde{Z}\|_F(\|\tilde{Z}\|_F - Z_M) - \varepsilon_N$$

$$= k_v(\|\tilde{Z}\|_F - Z_M/2)^2 - k_vZ_M^2/4 + K_{min}\|\zeta\| - \varepsilon_N$$

which is guaranteed positive as long as

$$\|\zeta\| > \left[k_vZ_M^2/4 + \varepsilon_N\right]/K_{min} \quad (3.23)$$

or

$$\|\tilde{Z}\|_F > Z_M/2 + \sqrt{Z_M^2/4 + \varepsilon_N/k_v}. \quad (3.24)$$

Thus, $\dot{V}$ is negative outside a compact set. The form of the right-hand side of (3.23) shows that the control gain K can be selected large enough so that

$$\left[k_vZ_M^2/4 + \varepsilon_N\right]/K_{min} < b_\zeta.$$

Therefore, any trajectory $\zeta(t)$ beginning in U_ζ evolves completely within U_ζ. According to a standard Lyapunov theorem extension [17], this demonstrates the UUB of both ζ and $\tilde{Z}$.

Q.E.D.

Note also that the problem of ***neural net weight initialization*** occurring in other approaches in the literature does not arise, since if is taken as zero the terms $-K_1e$, $-K_2\eta$ stabilizes the plant on an interim basis. A formal proof reveals that K or K_1, K_2 should be large enough and the initial error $\zeta(0)$ small enough [15].

A comparison with the results of Narendra *et al.* [17] shows that the NN reconstruction error ε_N increases the bounds on $\|\zeta\|$ and $\|\tilde{Z}\|_F$ in a very interesting way. Note, however, that arbitrarily small tracking error bounds may be achieved by selecting large control gains K. On the other hand, the NN weight error is fundamentally bounded by Z_M, the known bound on the ideal weights Z. The parameter k_v offers a design tradeoff between the relative eventual magnitudes of $\|\zeta\|$ and $\|\tilde{Z}\|_F$; a smaller k_v yields a smaller $\|\zeta\|$ and a larger $\|\tilde{Z}\|_F$, and vice versa.

Note that PE is not needed to establish the bounds on $\tilde{Z}$ with the modified weight tuning algorithm (3.31).

3.3 Simulation Results

Using the controller described in Section 3.2, we performed some simulation studies. In these simulations, we assume states are available. Using the data in [16], we simulate our robust NN controller without PE in Section 3.2. Fig. 3.3 shows the performance of PD control. The system goes unstable when there is a change in reference. There also exists steady-state error when the load disturbance is on. Fig. 3.4 shows the performance of PD + NN 1. The load disturbance is removed. However, due to the lack of compensation for the nonlinear dynamics by NN 2, the system becomes unstable when there is sudden change in speed reference. Fig. 3.5 shows the performance of PD + NN 2. Althought the system is stable now, the load disturbance effects still exists. Fig. 3.6 shows the performance of the complete controller. Now the system is both stable and clear of load disturbance. We used 4 and 10 units in the two NN's which approximate F_1, F_2, respectively. The reference trajectories are the same as those in Marino's paper [16]. ω^r is zero from 0 to 0.3 s., 220 r/s from 0.3 to 5 s., and 350 r/s from 5 sec. onwards. ψ_d^r is 1.3 Wb from 0 to 5 s. and 0.8 Wb after 5 s. The discontinuities are smoothed by linear interpolations. A load disturbance of 40 Nm is added at t = 2 s. We set $R_{rN} = 0.15$, $T_{LN} = 0$, $K_1 = \mathrm{diag}\{35,25\}$, $K_2 = \mathrm{diag}\{25,25\}$, $k_v = 1$, $\Gamma_i = 10\,I$, i = 1, 2. The applied voltage u_a has the same magnitude as that of Marino's paper [16] and is well within inverter limits.

It should be noted that the plots of u_a, u_b are not due to switching in sliding mode control as it appears to be. Similar waveforms have also been observed in Marino's adaptive input-output feedback method [16].

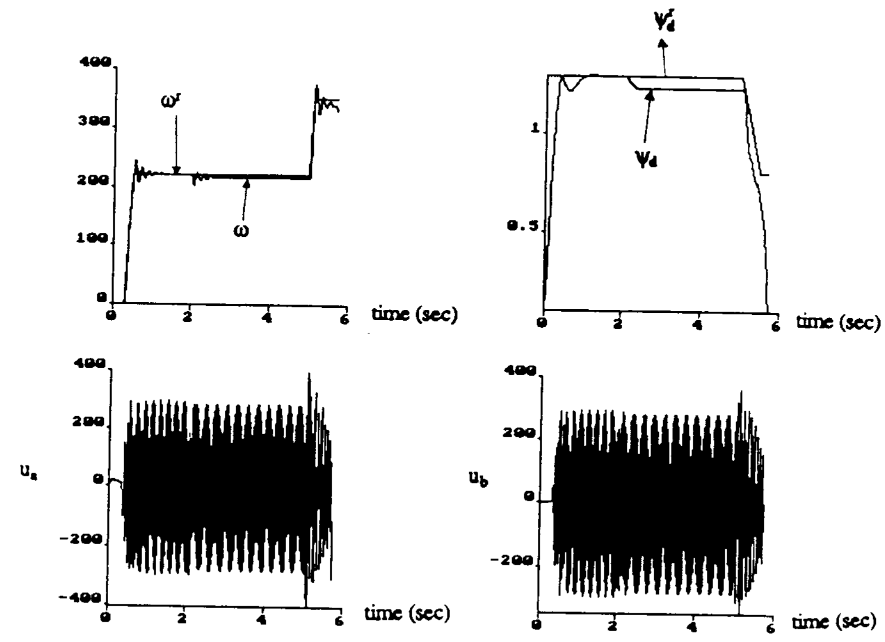

Fig. 3.3 Performance of PD controller.

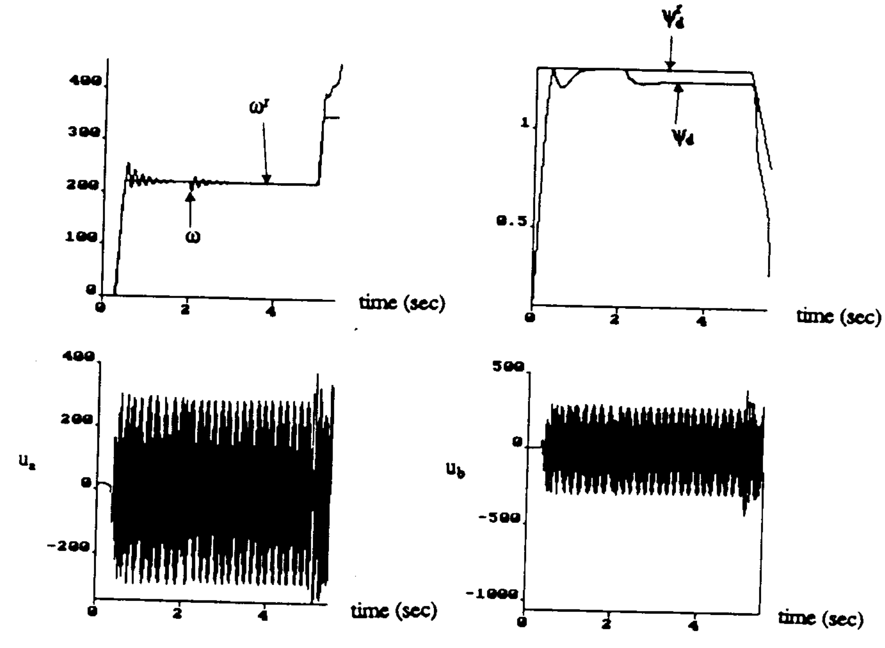

Fig. 3.4 Performance of controller (PD + NN 1).

4. Conclusion

There are four important contributions in this paper. The first one is that we have derived a novel robust NN control scheme to a "**nonlinear benchmark problem**" known as induction motors. The scheme involves the backstepping technique. Our method has

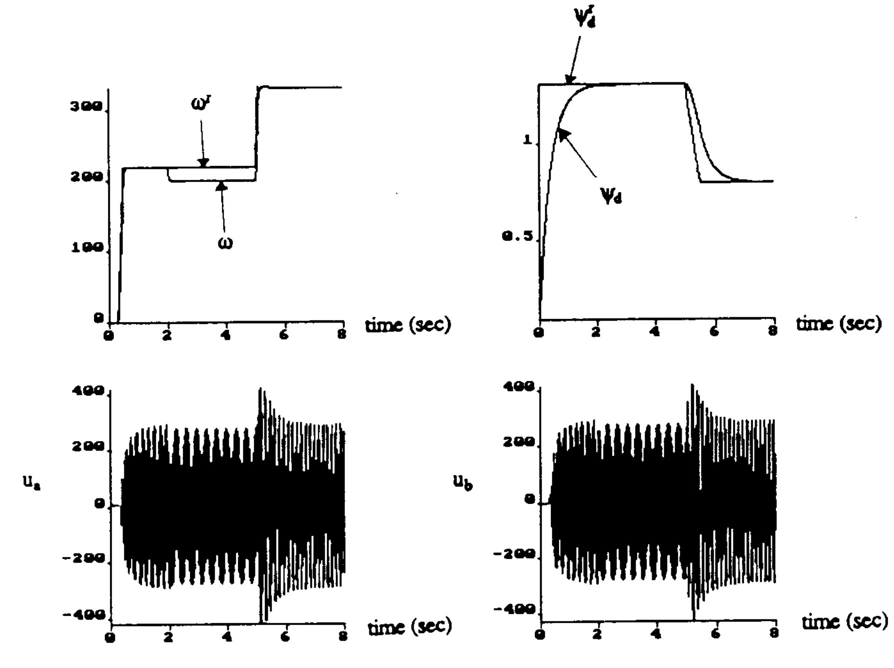

Fig. 3.5 Performance of controller (PD + NN 2).

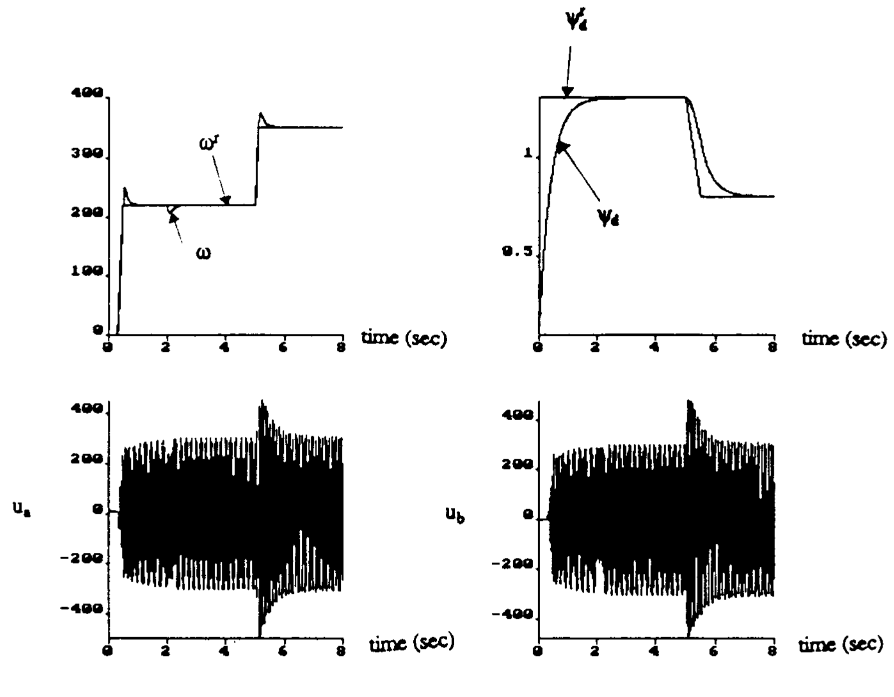

Fig. 3.6 Performance of complete NN controller.

achieved the same performance as that of Marino's adaptive input-output feedback method [16]. The design is modular and systematic and hence can be easily applied to nonlinear systems with similar structures. The second contribution is that our method does not require the linear parametrizability of the unknown parameters (LP). No regression matrix is needed, so that no preliminary dynamical analysis is needed. We believe this advantage is very useful for some systems where the LP property does not hold. The third one is that new tuning schemes, which *do not require any persistent excitation conditions*, are proposed which *can guarantee the boundedness of* *tracking error and weight updates*. The fourth one is that we do not need motor accelerations and the derivative of flux magnitude, as compared to some conventional sliding control schemes such as the works of Soto and Yeung [21].

References

[1] F. Blaschke, "The principle of Field Orientation Applied to the New Transvector Closed-loop control system for rotating field machines," *Siemens-Rev*, **39**, 217-220, 1972.

[2] D. Brown and E.P. Hamilton, *Electromechanical Energy Conversion*, Macmillan Publishing Co., 1984.

[3] F.-C. Chen and C.-C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural networks," *IEEE Trans. Autom. Control*, vol. 39, pp. 1306-1310, 1994.

[4] S.R. Chu and R. Shoureshi, "Neural-based adaptive nonlinear system identification," *Intelligent Control Systems*, DSC-vol. 45, ASME Winter Meeting, 1992.

[5] X. Cui and K.G. Shin, "Direct control and coordination using neural networks," *IEEE Trans. Systems, Man, and Cybernetics*, vol.23, no.3, 1993.

[6] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303-314, 1989.

[7] D. Dawson, Z. Qu, and J. Hu, "Robust Tracking Control of an Induction Motor," *Proc. American Control Conference*, pp. 648-652, 1993.

[8] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, pp. 359-366, 1989.

[9] Y. Iiguni, H. Sakai, and H. Tokumaru, "A nonlinear regulator design in the presence of system uncertainties using multilayer neural networks," *IEEE Trans Neural Networks*, vol. 2, no.4, pp. 410-417, 1991.

[10] I. Kanellakopoulos, P.V. Kokotovic, and A.S. Morse, "Systematic Design of Adaptive Controllers for Feedback Linearizable Systems, *IEEE Trans. Automat. Control*, vol. 36, pp. 1241-1253, 1991.

[11] P. V. Kokotovic, "Bode Lecture: The Joy of Feedback," *IEEE Control Systems Magazine*, no. 3, pp.7-17, June, 1992.

[12] L.G. Kraft and D.P. Campagna, "A summary comparison of CMAC Neural Network and Traditional adaptive control systems," *Neural Networks and Control*, pp. 143-169, MIT Press, 1991.

[13] C. Kwan and F.L. Lewis, "Robust Backstepping Control of Nonlinear Systems Using Neural Networks," submitted for publication.

[14] F. Lewis, K. Liu and A. Yesildirek, "Multilayer Neural Net Robot Controller With Guaranteed Tracking Performance," IEEE Trans. Neural Networks, May 1995.

[15] F.L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *Proc. IEEE Int. Symp. Intell. Cont.* pp. 225-231, 1993.

[16] R. Marino, S. Peresada, S., and P. Valigi, "Adaptive input-output linearization control of Induction motor, *IEEE.Transactions on Automatic Control*, **38**, 208-221, 1993.

[17] K.S. Narendra and A. M. Annaswamy, "A New adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Control*, vol. 32, pp. 134-145, 1987.

[18] K.S. Narendra, "Adaptive control using neural networks," *Neural Networks for Control*, pp. 115142, MIT Press, 1991.

[19] G.A. Rovithakis and M.A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 24, no. 3, pp.400-412, 1994.

[20] N. Sadegh, "Nonlinear identification and control via neural networks," *Control Systems with Inexact Dynamic Models*, DSC-vol.33, ASME Winter Annual Meeting, 1991.

[21] R. Soto and K.S. Yeung, "Sliding-Mode control of an Induction Motor without flux measurement," *Proc. 27th Annual Meeting and Conference of the IEEE Industry Application Society*, 1992.