6

Discrete-Time Output Trajectory Tracking

In this chapter, two schemes for trajectory tracking based on the backstepping and the block control techniques, respectively, are proposed, using an RHONO. This observer is based on a discrete-time recurrent high-order neural network (RHONN), which estimates the state of the unknown plant dynamics. The learning algorithm for the RHONN is based on an EKF. Once the neural network structure is determined, the backstepping and the block control techniques are used to develop the corresponding trajectory tracking controllers. The respective stability analyzes, using the Lyapunov approach, for the neural observer trained with the EKF and the controllers are included. Finally, the applicability of the proposed design is illustrated by an example: output trajectory tracking for an induction motor.

Nonlinear trajectory tracking is an important research subject ([1, 3, 4, 6, 8], and some references cited therein; mostly for continuous-time systems). In the recent literature on adaptive and robust controls, numerous approaches have been proposed for nonlinear trajectory tracking; among them the backstepping and the block control strategies provide well-suited design methodologies [2]. For most nonlinear control designs, it is usually assumed that the whole system state are measurable. In practice, however, it is very difficult to measure all the state variables.

For this reason, nonlinear state estimation remains an important topic for study in the nonlinear systems theory [9]. Recurrent neural-network observers have also been proposed, and they do not require a precise plant model. This technique is therefore attractive and actually has been successfully applied to state estimation [9, 10].

E.N. Sanchez et al.: *Discrete-Time High Order Neural Control – Trained with Kalman Filtering*, Studies in Computational Intelligence (SCI) **112**, 59–72 (2008)
www.springerlink.com

6.1 Backstepping Control Using an RHONO

In this section, an RHONO is used to estimate the plant state as in Sect. 5.1, and based on the backstepping technique developed in Sect. 3.1 the trajectory tracking problem is solved. The proposed control scheme is shown in Fig. 6.1. The main result of this chapter is established in the following proposition

Proposition 6.1. *Given a desired output trajectory* y_{d}*, a dynamic system with output* y*, and a neural network with output* $\widehat{y}$*, the following inequality holds* [2]:

$$\|y_{\mathrm{d}} - y\| \leq \|\widehat{y} - y\| + \|y_{\mathrm{d}} - \widehat{y}\|,$$

where $y_{\mathrm{d}} - y$ *is the system output tracking error,* $\widehat{y} - y$ *is the output estimation error, and* $y_{\mathrm{d}} - \widehat{y}$ *is the output tracking error of the nonlinear observer.*

Based on this proposition, it is possible to divide the tracking objective into two parts [2]:

1. Minimization of $\widehat{y} - y$, which can be achieved by the proposed online nonlinear observer algorithm trained with the EKF as shown in Theorem 5.1.
2. Minimization of $y_{\mathrm{d}} - \widehat{y}$. For this, a tracking algorithm is developed on the basis of the nonlinear observer (5.3). This minimization is obtained by designing the control law (3.5), as shown in Theorem 3.1.

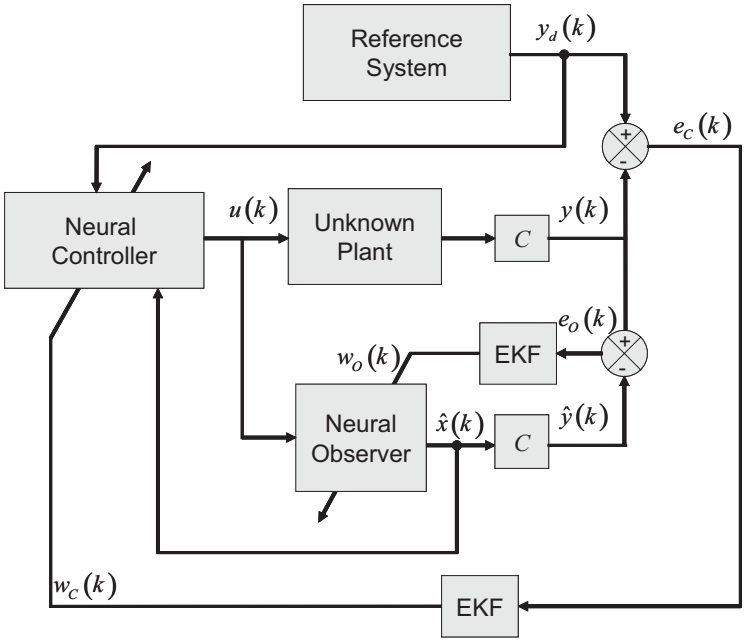

Fig. 6.1. Backstepping control scheme using an RHONO

It is possible to establish Proposition 6.1 due to the separation principle for discrete-time nonlinear systems [7], as stated in Theorem 2.2 and Corollary 2.1.

6.1.1 Application to an Induction Motor

In this section, the control objective is to achieve velocity and flux amplitude tracking for the discrete-time induction motor model (3.33), using the backstepping technique control algorithm developed in Chap. 3 and the RHONO (5.17), as is shown in Fig. 6.1.

Block-Strict-Feedback-Form (BSFF) for an Induction Motor

Let us define the following states:

$$\begin{aligned} x^1(k) &= \begin{bmatrix} \widehat{x}_1(k) \\ \Psi(k) \end{bmatrix}; \quad x^2(k) = \begin{bmatrix} \widehat{x}_4(k) \\ \widehat{x}_5(k) \end{bmatrix}, \\ u(k) &= \begin{bmatrix} u^{\alpha}(k) \\ u^{\beta}(k) \end{bmatrix}; \quad y_{\mathrm{d}}(k) = \begin{bmatrix} \omega_{\mathrm{d}}(k) \\ \Psi_{\mathrm{d}}(k) \end{bmatrix}, \\ y(k) &= x^1(k), \end{aligned} \tag{6.1}$$

where $\Psi(k) = \widehat{x}_2^2(k) + \widehat{x}_3^2(k)$ is the rotor flux magnitude, $\omega_{\mathrm{d}}(k)$ and $\Psi_{\mathrm{d}}(k)$ are the reference signals. The control objective is to force the output $y(k)$ to track the reference $y_{\mathrm{d}}(k)$. Using (6.1), the system (3.33) can be represented in the BSFF consisting of two blocks

$$\begin{aligned} x^1(k+1) &= f^1(x^1(k)) + g^1(x^1(k))x^2(k) + d^1(k), \\ x^2(k+1) &= f^2(\overline{x}^2(k)) + g^2(\overline{x}^2(k))u(k), \end{aligned}$$

where $f^1(x^1(k))$, $g^1(x_1(k))$, $f_2(\overline{x}_2(k))$, and $g_2(\overline{x}_2(k))$ are assumed to be unknown and $d_1(k)$ is an unknown bounded disturbance; in this case, this disturbance is the load torque. Now we use the HONN to approximate the desired virtual controls and the ideal practical controls described as

$$\begin{aligned} \alpha^{1*}(k) \triangleq x^2(k) &= \varphi^1(x^1(k), y_{\mathrm{d}}(k+2)), \\ u^*(k) &= \varphi^2(x^1(k), x^2(k), \alpha^{1*}(k)), \\ y(k) &= x^1(k). \end{aligned}$$

The HONN proposed for this application is as follows:

$$\begin{aligned} \alpha^1(k) &= {w^1}^{\top} z^1(\varrho^1(k)), \\ u(k) &= {w^2}^{\top} z^2(\varrho^2(k)), \end{aligned}$$

with

$$\varrho^1(k) = [x^1(k), y_\mathrm{d}(k+2)]^\top,$$
$$\varrho^2(k) = [x^1(k), x^2(k), \alpha^1(k)]^\top.$$

The weights are updated using the EKF:

$$w^i(k+1) = w^i(k) + \eta^i K^i(k) e^i(k), \qquad i = 1,\ 2,$$
$$K^i(k) = P^i(k) H^i(k) \left[R^i(k) + H^{i^\top}(k) P^i(k) H^i(k) \right]^{-1},$$
$$P^i(k+1) = P^i(k) - K^i(k) H^{i^\top}(k) P^i(k) + Q^i(k),$$

with

$$e^1(k) = y_\mathrm{d}(k) - y(k),$$
$$e^2(k) = x^2(k) - \alpha^1(k),$$

The training is performed online, using a parallel configuration. All the NN states are initialized in a random way. The associated covariances matrices are initialized as diagonals, and the nonzero elements are $P_1(0) = P_2(0) = 10{,}000$; $Q_1(0) = Q_2(0) = 5{,}000$; and $R_1(0) = R_2(0) = 10{,}000$, respectively.

Simulation Results

The simulations are performed for the system (3.33) using the following parameters: $R_\mathrm{s} = 14\,\Omega$; $L_\mathrm{s} = 400\,\mathrm{mH}$; $M = 377\,\mathrm{mH}$; $R_\mathrm{r} = 10.1\,\Omega$; $L_\mathrm{r} = 412.8\,\mathrm{mH}$; $n_\mathrm{p} = 2$; $J = 0.01\,\mathrm{Kg\,m^2}$; $T = 0.0001\,\mathrm{s}$. To estimate the state of system (3.33), we use the RHONO (5.3) with $n = 5$ trained with the EKF (5.7).

The tracking results are presented in Figs. 6.2 and 6.3. There the tracking performance can be verified for the two plant outputs. Figure 6.4 displays the load torque applied as an external disturbance. Figure 6.5 portrays a parametric variation introduced in the rotor resistance (R_r) as an increment. Figure 6.6 shows the weights evolution. Figures 6.7 and 6.8 portray the fluxes and their estimates.

6.2 Block Control Using an RHONO

In this section, an RHONO is used to estimate the plant state as in Sect. 5.1, and based on the block control technique developed in Sect. 4.2 the trajectory tracking problem is solved on the same basis of Proposition 6.1, dividing the tracking objective into two parts, as in Sect. 6.1 [2]. The proposed control scheme is shown in Fig. 6.9.

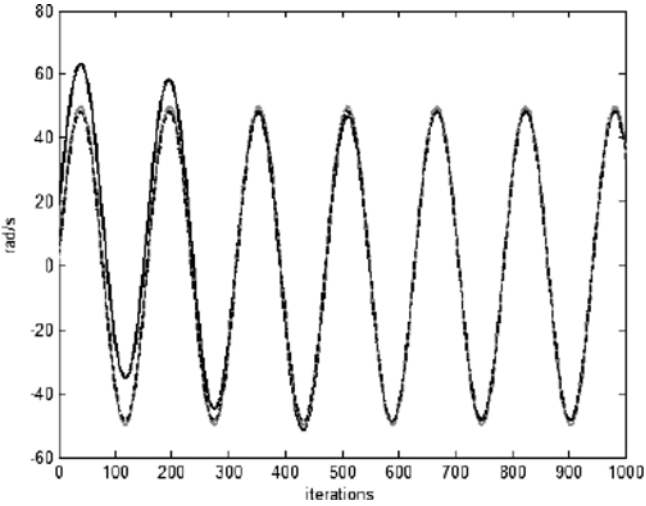

Fig. 6.2. Tracking performance $\omega(k)$ (*solid line*), $\widehat{x}_1(k)$ (*dash-dot line*), and $\omega_{\mathrm{r}}(k)$ (*dashed line*)

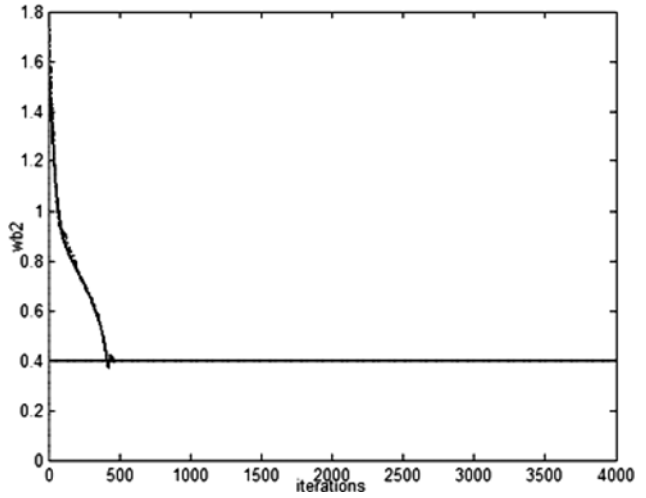

Fig. 6.3. Tracking performance $\Psi(k)$ (*solid line*), $\widehat{x}_2^2(k)+\widehat{x}_3^2(k)$ (*dash-dot line*), and $\Psi_{\mathrm{r}}(k)$ (*dashed line*)

6.2.1 Application to an Induction Motor

To this end we use the RHONO developed for the discrete-time induction motor model, developed in Sect. 5.2, which is described as

$$\begin{aligned}\widehat{x}_1(k+1) = {} & w_{11}(k)S(\widehat{x}_1(k)) + w_{12}(k)S(\widehat{x}_1)S(\widehat{x}_3(k))\widehat{x}_4(k) \\ & + w_{13}(k)S(\widehat{x}_1)S(\widehat{x}_2(k))\widehat{x}_5(k) + g_1 e(k),\end{aligned}$$

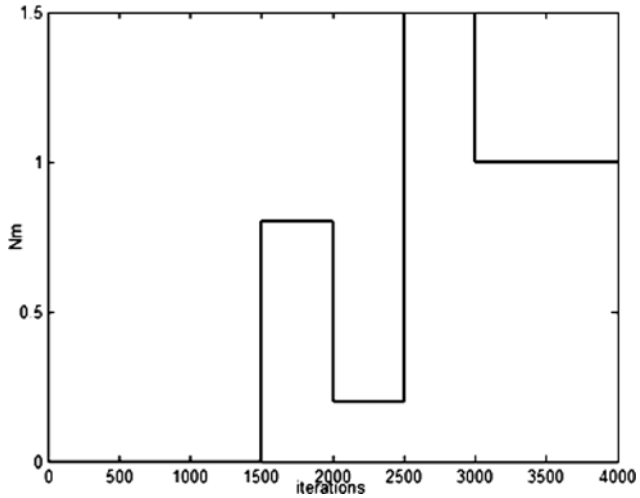

Fig. 6.4. Load torque $T_{\mathrm{L}}(k)$

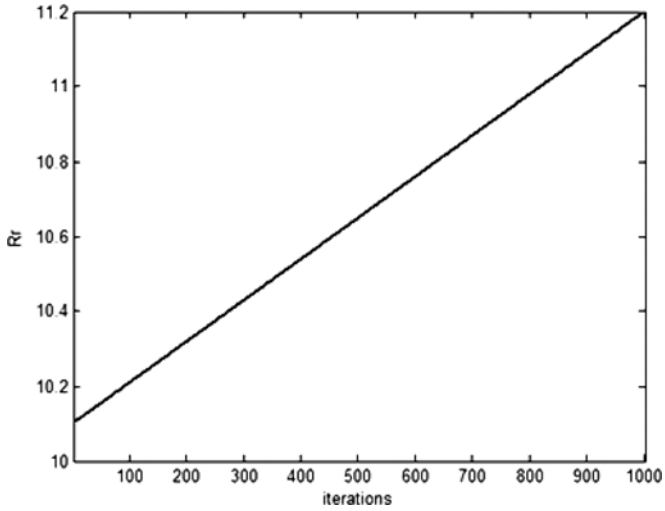

Fig. 6.5. Rotor resistance variation (R_{r})

$$
\begin{aligned}
\widehat{x}_2(k+1) &= w_{21}(k)S(\widehat{x}_1(k))S(\widehat{x}_3(k)) + w_{22}(k)\widehat{x}_5(k) + g_2e(k),\\
\widehat{x}_3(k+1) &= w_{31}(k)S(\widehat{x}_1(k))S(\widehat{x}_2(k)) + w_{32}(k)\widehat{x}_4(k) + g_3e(k),\\
\widehat{x}_4(k+1) &= w_{41}(k)S(\widehat{x}_2(k)) + w_{42}(k)S(\widehat{x}_3(k)) + w_{43}(k)S(\widehat{x}_4(k))\\
&\quad + w_{44}(k)u^{\alpha}(k) + g_4e(k),\\
\widehat{x}_5(k+1) &= w_{51}(k)S(\widehat{x}_2(k)) + w_{52}(k)S(\widehat{x}_3(k)) + w_{53}(k)S(\widehat{x}_5(k))\\
&\quad + w_{54}(k)u^{\beta}(k) + g_5e(k),
\end{aligned}
\tag{6.2}
$$

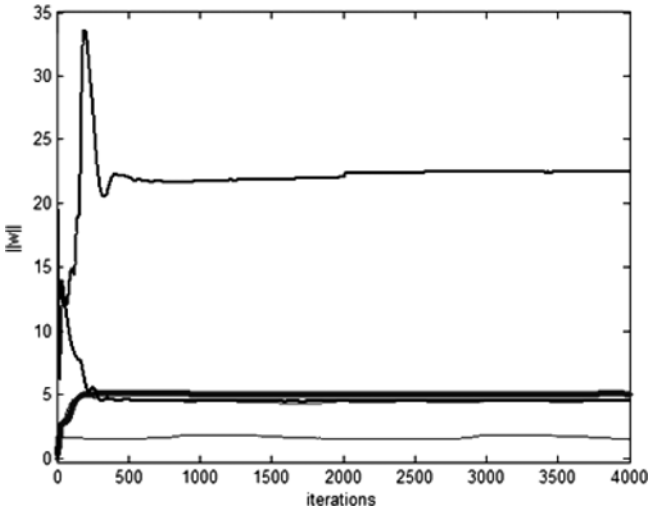

Fig. 6.6. Weights evolution

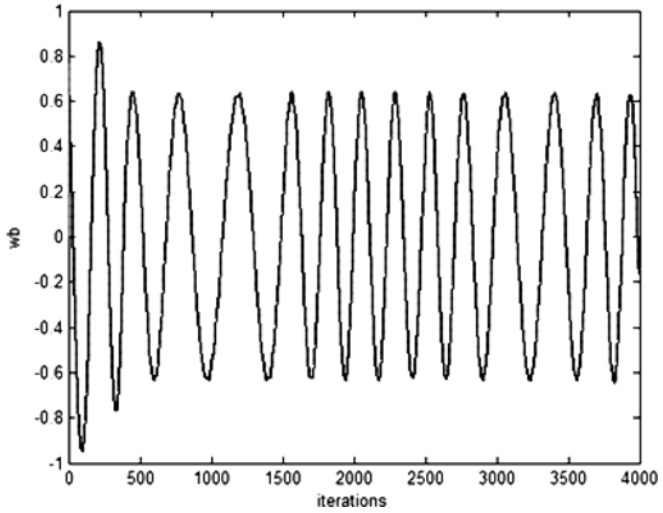

Fig. 6.7. Time evolution of $\psi^{\alpha}(k)$ (*solid line*) and its estimated $\widehat{x}_2(k)$ (*dashed line*)

where $\widehat{x}_1$ estimates the angular speed ω; $\widehat{x}_2$ and $\widehat{x}_3$ estimates the fluxes ψ^{α} and ψ^{β}, respectively; $\widehat{x}_4$ and $\widehat{x}_5$ estimates the currents i^{α} and i^{β}, respectively. The training is performed online, using a parallel configuration. All the NN states are initialized in a random way as well as the weights vectors. It is important to remark that the initial conditions of the plant are completely different from the initial conditions for the NN.

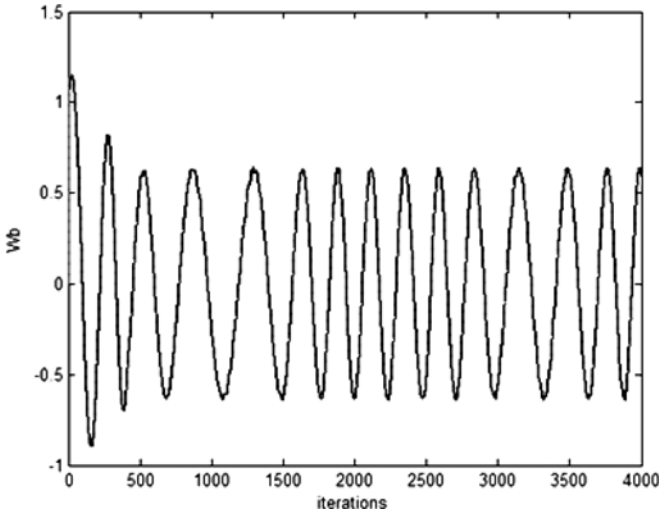

Fig. 6.8. Time evolution of $\psi^{\beta}(k)$ (*solid line*) and its estimated $\widehat{x}_3(k)$ (*dashed line*)

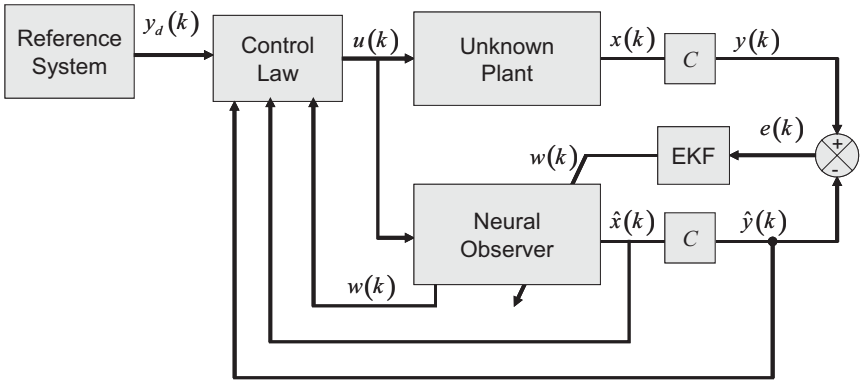

Fig. 6.9. Block control scheme using an RHONO

Comment 6.1. It is important to remark that as in Chap. 4, to apply the Block control and the sliding modes techniques it is necessary to use the modified EKF (4.5), to avoid the zero-crossing for $w_{44}(k)$ and $w_{54}(k)$. The proof for the RHONN trained with the modified EKF is similar to the proof of Theorem 4.1.

Neural Block Controller Design

The control objective is to achieve velocity and flux amplitude tracking for the discrete-time induction motor model (3.33), using the discrete-time block control and sliding mode techniques control algorithm developed in Chap. 4. Let us define the following states as

$$x^1(k) = \begin{bmatrix} \widehat{x}_1(k) - \omega_{\rm r}(k) \\ \Psi(k) - \Psi_{\rm r}(k) \end{bmatrix}, \quad x^2(k) = \begin{bmatrix} \widehat{x}_4(k) \\ \widehat{x}_5(k) \end{bmatrix}, \tag{6.3}$$

where $\Psi(k) = \widehat{x}_2^2(k) + \widehat{x}_3^2(k)$ is the rotor flux identify magnitude, $\Psi_{\rm r}(k)$ and $\omega_{\rm r}(k)$ are reference signals. Then

$$\begin{aligned}
\Psi(k+1) = {} & w_{21}^2(k) S^2(\widehat{x}_1(k)) S^2(\widehat{x}_3(k)) + w_{22}^2(k) \widehat{x}_5^2(k) + w_{32}^2(k) \widehat{x}_4^2(k) \\
& + w_{31}^2(k) S^2(\widehat{x}_1(k)) S^2(\widehat{x}_2(k)) \\
& + 2 w_{21}(k) S(\widehat{x}_1(k)) S(\widehat{x}_3(k)) w_{22}(k) \widehat{x}_5(k) \\
& + 2 w_{31}(k) S(\widehat{x}_1(k)) S(\widehat{x}_2(k)) w_{32}(k) \widehat{x}_4(k) \\
& + 2 w_{22}(k) \widehat{x}_5(k) g_2 e(k) + (g_2 e(k))^2 \\
& + 2 w_{21}(k) S(\widehat{x}_1(k)) S(\widehat{x}_3(k)) g_2 e(k) + (g_3 e(k))^2 \\
& + 2 w_{32}(k) \widehat{x}_4(k) g_3 e(k) + 2 w_{31}(k) S(\widehat{x}_1(k)) S(\widehat{x}_2(k)) g_3 e(k).
\end{aligned}$$

Using (6.3), (6.2) can be represented in the block control form consisting of two blocks

$$\begin{aligned}
x^1(k+1) &= f_1(x^1(k)) + B_1(x^1(k)) x^2(k), \\
x^2(k+1) &= f_2(x^1(k), x^2(k)) + B_2(k) u(k),
\end{aligned} \tag{6.4}$$

with $u(k) = \left[u^\alpha(k) \; u^\beta(k) \right]^\top$ and

$$\begin{aligned}
f_1(x^1(k)) &= \begin{bmatrix} w_{11}(k) S(x_1(k)) + g_1 e(k) - \omega_{\rm r}(k+1) \\ f_{11}(k) \end{bmatrix}, \\
f_{11}(k) &= w_{21}^2(k) S^2(\widehat{x}_1(k)) S^2(\widehat{x}_3(k)) + w_{31}^2(k) S^2(\widehat{x}_1(k)) S^2(\widehat{x}_2(k)) \\
&\quad + 2 w_{22}(k) \widehat{x}_5(k) g_2 e(k) + (g_2 e(k))^2 \\
&\quad + 2 w_{21}(k) S(\widehat{x}_1(k)) S(\widehat{x}_3(k)) g_2 e(k) \\
&\quad + (g_3 e(k))^2 + 2 w_{32}(k) \widehat{x}_4(k) g_3 e(k) \\
&\quad + (g_3 e(k))^2 + 2 w_{32}(k) \widehat{x}_4(k) g_3 e(k) \\
&\quad + w^2 I_m^2(k) - \Psi_r(k+1), \\
I_m(k) &= \sqrt{w_{22}^2(k) \widehat{x}_4^2(k) + w_{32}^2(k) \widehat{x}_5^2(k)}, \\
B_1(x^1(k)) &= \begin{bmatrix} b_{11}(k) \; b_{12}(k) \\ b_{21}(k) \; b_{22}(k) \end{bmatrix}, \\
b_{11}(k) &= w_{12}(k) S(\widehat{x}_1(k)) S(\widehat{x}_3(k)), \\
b_{12}(k) &= w_{13}(k) S(\widehat{x}_1(k)) S(\widehat{x}_2(k)), \\
b_{21}(k) &= 2 w_{31}(k) w_{32}(k) S(\widehat{x}_1(k)) S(\widehat{x}_2(k)), \\
b_{22}(k) &= 2 w_{21}(k) w_{22}(k) S(\widehat{x}_1(k)) S(\widehat{x}_2(k)), \\
f_2(x^2(k)) &= \begin{bmatrix} f_{21}(k) \\ f_{22}(k) \end{bmatrix}, B_2(k) = \begin{bmatrix} w_{44}(k) & 0 \\ 0 & w_{54}(k) \end{bmatrix}, \\
f_{21}(k) &= w_{41}(k) S(\widehat{x}_2(k)) + w_{42}(k) S(\widehat{x}_3(k)) + w_{43}(k) S(\widehat{x}_4(k)),
\end{aligned}$$

$$f_{22}(k) = w_{51}(k)S(\widehat{x}_2(k)) + w_{52}(k)S(\widehat{x}_3(k)) + w_{53}(k)S(\widehat{x}_5(k)).$$

Applying the block control technique, we define the following vector $\mathbf{z}_1(k) = x^1(k)$. Then

$$\mathbf{z}_1(k+1) = f_1(x^1(k)) + B_1(x^1(k))x^2(k) = \mathbf{K}\mathbf{z}_1(k), \tag{6.5}$$

where $\mathbf{K} = \text{diag}\{\mathbf{k}_1, \mathbf{k}_2\}$, with $|\mathbf{k}_i| < 1$ $(i = 1, 2)$; then the desired value $x^{2\text{d}}(k)$ of $x^2(k)$ is calculated from (6.5) as

$$x^{2\text{d}}(k) = B_1^{-1}(x^1(k))[-f_1(x^1(k)) + \mathbf{K}\mathbf{z}_1(k)].$$

It is desired that $x^2(k) = x^{2\text{d}}(k)$. Hence, second new error vector is defined as

$$\mathbf{z}_2(k) = x^2(k) - x^{2\text{d}}(k).$$

Then

$$\mathbf{z}_2(k+1) = f_3(x^1(k)) + B_2(k)u(k),$$

with

$$f_3(x^1(k)) = f_2(x^2(k)) - B_1^{-1}(x^1(k+1))[-f_1(x^1(k+1)) + \mathbf{K}\mathbf{z}_1(k+1)].$$

Let us select the manifold for the sliding mode as $S_{\text{D}}(k) = \mathbf{z}_2(k)$. To design a control law, a discrete-time sliding mode version is implemented as

$$u(k) = \begin{cases} u_{\text{eq}}(k) & \text{if } \|u_{\text{eq}}(k)\| \le u_0, \\ u_0 \frac{u_{\text{eq}}(k)}{\|u_{\text{eq}}(k)\|} & \text{if } \|u_{\text{eq}}(k)\| > u_0, \end{cases}$$

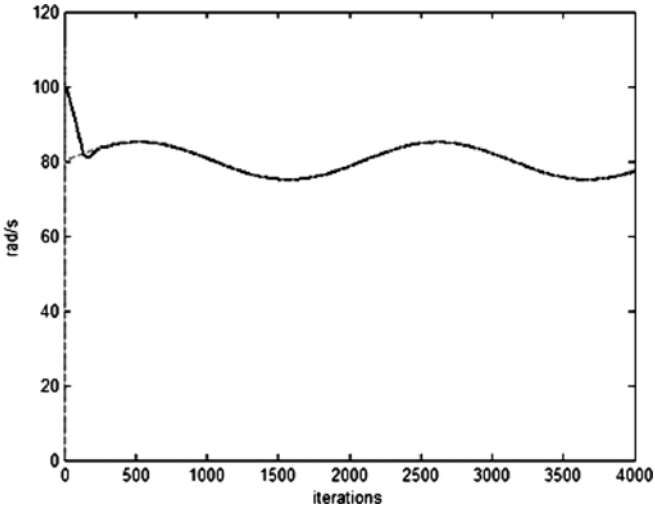

Fig. 6.10. Tracking performance $\omega(k)$ (*solid line*), $\widehat{x}_1(k)$ (*dash-dot line*) and $\omega_r(k)$ (*dashed line*)

where $u_{\mathrm{eq}}(k) = -B_2^{-1}(k) f_3(x^1(k))$ is calculated from $S_{\mathrm{D}}(k) = 0$ and u_0 is the control resources that bound the control. Because of the time varying of RHONO weights, we need to guarantee that $B_1(\bullet)$ and $B_2(\bullet)$ are not singular; then it is necessary to avoid the zero-crossing of the weights $w_{13}(k)$, $w_{22}(k)$, $w_{32}(k)$, $w_{44}(k)$, and $w_{54}(k)$, which are the so-called controllability weights [2]. It is important to remark that in this application only the weights $w_{44}(k)$ and $w_{54}(k)$ tend to cross zero.

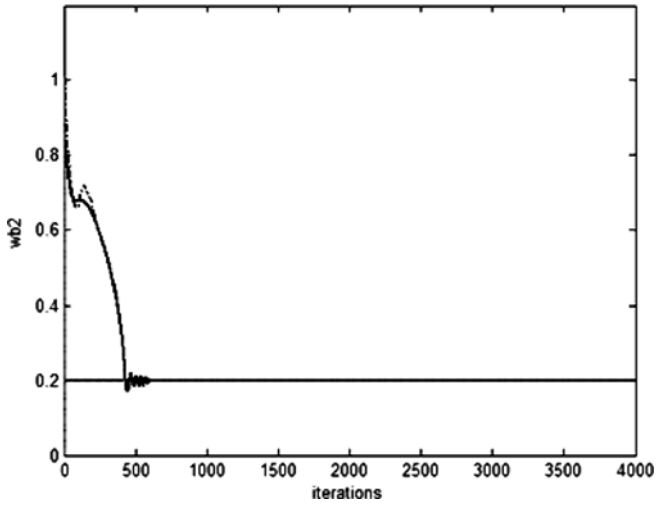

Fig. 6.11. Tracking performance $\Psi(k)$ (*solid line*), $\widehat{x}_2^2(k) + \widehat{x}_3^2(k)$ (*dash-dot line*), and $\Psi_r(k)$ (*dashed line*)

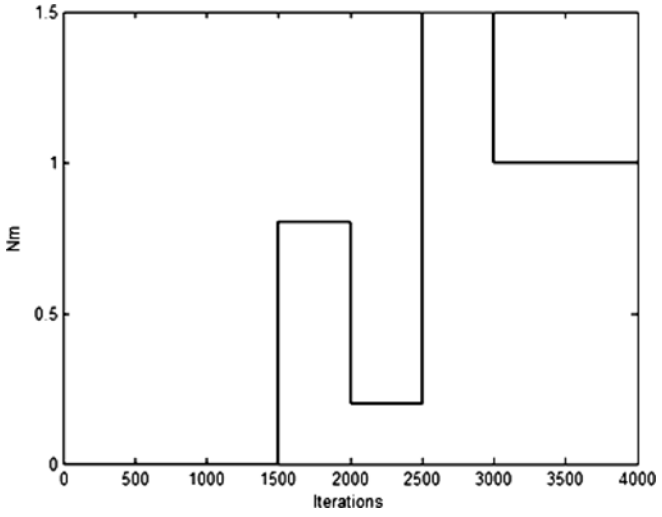

Fig. 6.12. Load torque $T_{\mathrm{L}}(k)$

Simulation Results

Simulations are performed for the system (3.33), using the following parameters: $R_\mathrm{s} = 14\,\Omega$; $L_\mathrm{s} = 400\,\mathrm{mH}$; $M = 377\,\mathrm{mH}$; $R_\mathrm{r} = 10.1\,\Omega$; $L_\mathrm{r} = 412.8\,\mathrm{mH}$; $n_\mathrm{p} = 2$; $J = 0.01\,\mathrm{Kg\,m^2}$; $T = 0.001\,\mathrm{s}$. To estimate the state of system (3.33) we use the RHONO (5.3) with $n = 5$ trained with the EKF (5.7).

The tracking results are presented in Figs. 6.10 and 6.11. There the tracking and state estimation performance can be verified for the two plant outputs.

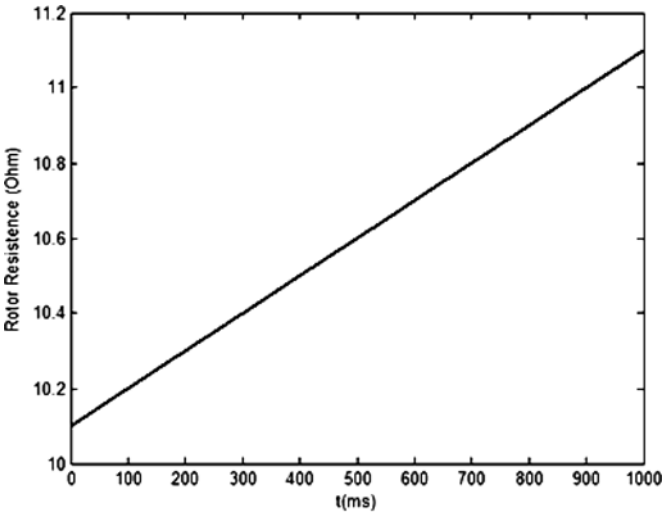

Fig. 6.13. Rotor resistance variation (R_r)

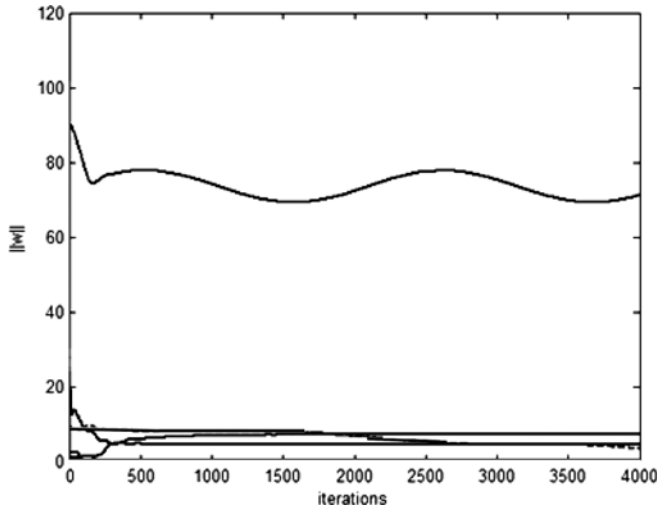

Fig. 6.14. Weights evolution

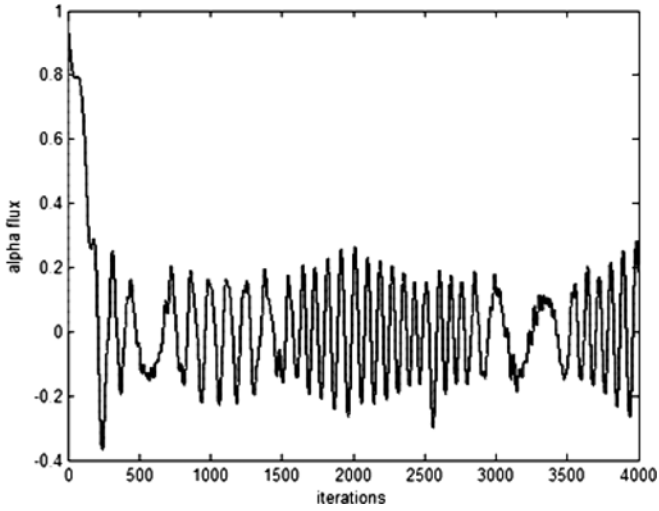

Fig. 6.15. Time evolution of $\psi^{\alpha}(k)$ (*solid line*) and its estimated $\widehat{x}_2(k)$ (*dashed line*)

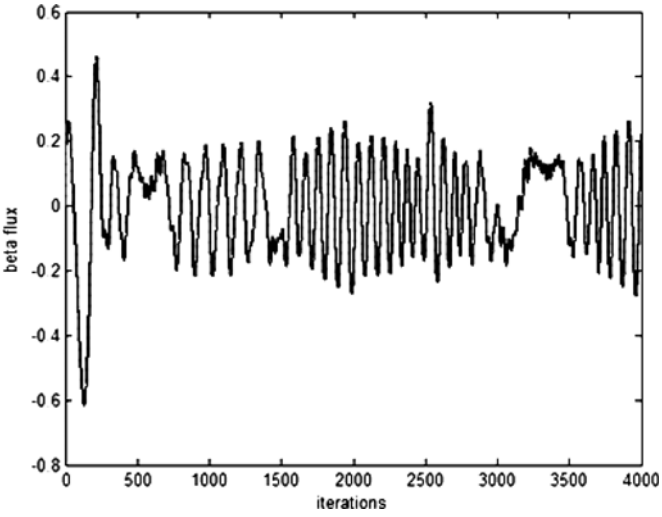

Fig. 6.16. Time evolution of $\psi^{\beta}(k)$ (*solid line*) and its estimated $\widehat{x}_3(k)$ (*dashed line*)

Figure 6.12 displays the load torque applied as an external disturbance. Figure 6.13 presents the parametric variation introduced in the rotor resistance (R_{r}) as a variation of $1\,\Omega\,\mathrm{s}^{-1}$. Figure 6.14 shows the weights evolution. Figures 6.15 and 6.16 portray the fluxes and their estimates.

6.3 Conclusions

In this chapter, the discrete-time output trajectory tracking is solved via the design of two neural controllers based on the backstepping and the block control techniques, respectively. First, a nonlinear observer is designed based on a RHONN trained with a modified EKF-based algorithm, where the training of the nonlinear observer is performed online in a parallel configuration. Then, based on the RHONO, the backstepping and the block control techniques are designed, respectively. Simulation results for an induction motor are included to illustrate the applicability of the proposed control schemes.