Linear Quadratic State Feedback and Robust Neural Network Estimator for Field-Oriented-Controlled Induction Motors

Pompeo Marino, Michele Milano, and Francesco Vasca, *Member, IEEE*

***Abstract*—A field-oriented control scheme for an induction motor with a linear quadratic optimal regulator and a robust neural network estimator is proposed. The state feedback is designed by using the synchronous frame motor model. The number of the states is increased in order to take into account the presence of two integrators on the flux and torque errors. The resulting model is suitably simplified and the corresponding approximations are discussed. The procedure proposed is shown to be suitable also for the design of the state feedback via pole placement technique. A comparison with standard proportional integral regulators is provided. The rotor flux is estimated by using a robust neural network observer. The network training set is suitably designed in order to preserve the drive effectiveness also in the presence of large parameter uncertainties. The robust neural observer is compared with an extended Kalman filter and a standard neural network observer. Using a 250-kW induction motor as a case study, the simulation results show the effectiveness of the proposed solution, both during transient and steady-state operating conditions.**

***Index Terms*—Induction motor drives, linear quadratic control, neural network applications, parameter estimation.**

I. Introduction

In recent years, field-oriented control (FOC) has allowed drastically improved dynamic performance of induction motor drives. As is well known, the implementation of this control technique needs the knowledge of the rotor fluxes, which are, in general, difficult to measure. In order to overcome this problem, a rotor flux observer, usually based on the measurements of stator voltages, stator currents, and rotor speed, can be designed. Once the estimation problem has been solved, all motor state variables (measured and/or estimated), i.e., the stator currents, the rotor fluxes and the rotor speed, can be used in order to design a feedback controller. In the recent literature, different solutions for the observer/controller design, aimed to avoid the lack of performance due to parameter variations (or uncertainties) and measurement noise, have been proposed. These problems become more significant for medium- and high-power induction motors, where the inverter switching frequency is limited by the electronic devices available and the motor parameters undergo larger variations.

The classical Luenberger observer was first proposed in order to reconstruct the motor variables which are difficult to measure [1]. Then, the extended Kalman filter (EKF) algorithm has been widely considered for the estimation of both the rotor fluxes and the rotor resistance [2]. A possible solution for the problem of the large computational burden related to the EKF algorithm has been proposed in [3], with a filter order reduction, and in [4] with a suitable interlacement between a Kalman filter and a least-square algorithm. However, the application of these algorithms for the simultaneous estimation of more than one motor parameter leads to much more complicated structures. A possible solution has been presented in [5], where, by assuming a slowly varying rotor speed, a different least-square algorithm allowed the simultaneous estimation of the rotor fluxes and the motor parameters. Recently, neural networks have been proposed as a valuable alternative to the above-mentioned solutions [6]. Some of the advantages offered by the use of neural networks as observers for induction motors are as follows:

- their almost instantaneous response;
- the simple modifications needed in order to be adapted to parameter variations;
- their availability on integrated chips;
- the good robustness properties also achievable without on-line adaptation mechanisms, when employed for vector control schemes [7].

On the other hand, the use of neural networks may be complicated by some drawbacks, such as the choice of a realistic (for simulations) or general enough (for experiments) training set, the time needed for the training procedure, and the lack of performance in the presence of unexpected operating conditions or parameters variations.

Once a robust observer is available, i.e., by assuming that the motor state variables are available, the controller of the motor can be designed. Most of the research effort expended on the design of the torque and flux control of the field-oriented-controlled motor has been directed toward the well-known solution of two feedback loops with standard proportional integrator (PI) controllers [8]. These controllers ensure the regulation of the torque and the flux to their constant reference values and provide the synchronous reference currents. The currents are then controlled by means of further feedback loops, thus obtaining, after the inverse of the synchronous

Manuscript received April 6, 1997; revised December 21, 1997. Abstract published on the Internet October 26, 1998.

P. Marino is with the Dipartimento di Ingegneria dell'Informazione, Seconda Università di Napoli, Aversa (CE), Italy.

M. Milano is with the Fachbereich Maschinenbau und Verfahrenstechnik, ETH, 8092 Zurich, Switzerland.

F. Vasca is with the Facoltà di Ingegneria di Benevento, Università degli Studi del Sannio, 82100 Benevento, Italy (e-mail: vasca@disna.dis.unina.it).

Publisher Item Identifier S 0278-0046(99)00493-1.

0278–0046/99$10.00 © 1999 IEEE

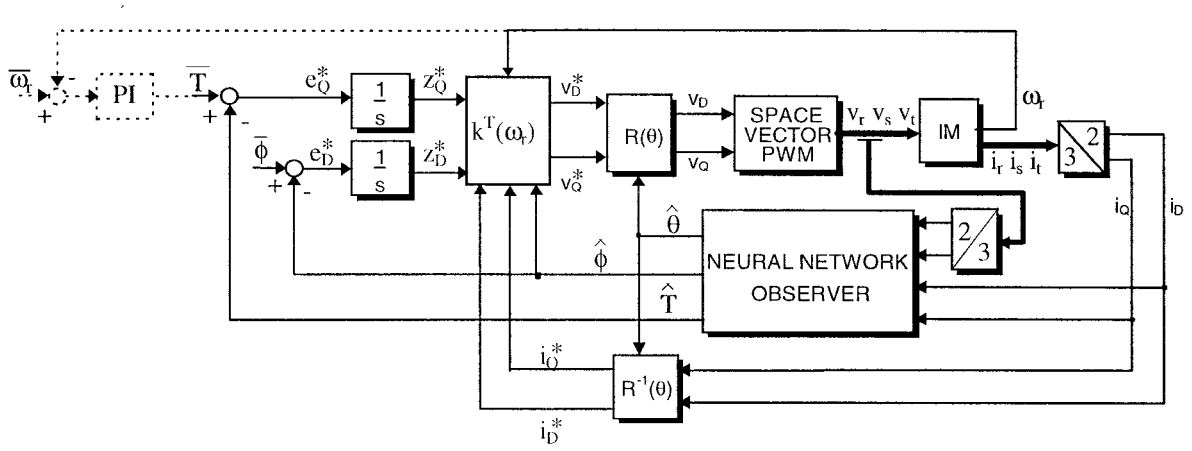

Fig. 1. Closed-loop scheme of the FOC induction motor with state feedback.

transformation, the input voltages for the inverter. The main drawback of this solution is that the PI parameters are difficult to select in order to achieve the torque and flux control with the minimum transient and overshoot. Moreover, often, the controller parameters need to be adapted when the rotor speed varies and, in this case, a suitable adaptation mechanism must be designed.

In this paper, a linear quadratic (LQ) feedback controller on the field-oriented-controlled induction motor and a robust neural network observer are proposed. In particular, the LQ state feedback is designed by using a dynamic model which is obtained by a suitable increase of the number of the state variables of the induction motor model represented in a synchronous reference frame. The controller provides the synchronous stator voltages, whereas the neural network observer estimates the flux needed for the state feedback and for the FOC. The feedback gains are calculated off-line for different constant rotor speed values and then interpolated for on-line operation. The same approach is shown to be useful in order to obtain the state feedback gains designed via pole placement technique. For the neural network estimator design, a suitable training set, based on the simulations of the LQ-controlled induction motor, is generated in order to also preserve the estimation accuracy in the presence of parameters variations or uncertainties. More specifically, the Price algorithm is used to estimate the locus of the possible model parameter variations which is coherent with a process noise, the variance of which is less than a given reasonable one [7].

The effectiveness of the proposed scheme is checked via simulations on a 250-kW induction motor driven by a space-vector voltage-source pulsewidth modulation (PWM) inverter, in the presence of large parameters uncertainties. The corresponding results are compared with those achievable with the standard PI control, with a pole placement state feedback controller, and by using as an observer either an EKF or the neural network presented in [6].

This paper is organized as follows. In Section II, the whole closed-loop scheme is described. In Section III, the state feedback design procedure is detailed. Section IV presents the neural network structure and the Price algorithm used for the generation of the training set. In Section V, the effectiveness of the proposed scheme is discussed via simulation results.

II. Modeling

A. The System Under Investigation

The whole closed-loop scheme of the considered field-oriented-controlled induction motor with state feedback is represented in Fig. 1. The errors on the flux and torque with respect to their reference values $\bar{\Phi}$ and $\bar{T}$ are the inputs of two integrators which ensure zero steady-state errors in the presence of constant reference signals. A suitable feedback gain matrix K^T, which depends on the rotor speed ω_r,[1] multiplies the outputs of the integrators, the synchronous currents i_D^* and i_Q^*, and the estimated flux. The obtained synchronous reference voltages v_D^* and v_Q^* are transformed to the stationary ones by means of the rotation matrix $R(\theta)$, which uses the instantaneous phase of the rotor flux estimated by an observer which includes a neural network. The inputs of this observer are the stationary stator voltages and currents. A possible further feedback loop can be considered in order to regulate the rotor speed.

The nominal parameters of the 250-kW induction motor considered in this paper are the same as the one used in [9]

$$R_s = 0.102\ \Omega, \quad R_r = 0.115\ \Omega, \quad L_1 = 1.56\ \text{mH}$$
$$L_2 = 1.43\ \text{mH}, \quad M = 41.4\ \text{mH}, \quad N_p = 2$$
$$J = 60\ \text{kg}\cdot\text{m}^2, \text{ and } \quad T_L = 100\ \text{N}\cdot\text{m}$$

where L_1 and L_2 are the leakage inductances, N_p is the number of pole pairs and T_L is the load torque. Moreover, we consider a space-vector PWM inverter, the dc input voltage of which is equal to 1.5 kV, and it operates in asynchronous mode with a switching frequency of 300 Hz.

In this section, we describe the dynamic model of the induction motor. In particular, the approximations introduced

[1] For the sake of simplicity, we will not indicate explicitly the dependence on time of the state and input variables.

on the FOC motor model in order to design the state feedback controllers are discussed.

B. The Motor Dynamic Model

The dynamic model of the induction motor is written by choosing as state variables the stator currents i_D and i_Q, the rotor fluxes φ_d and φ_q,[2] and the rotor speed ω_r. Therefore, by assuming a *stationary* reference frame, we obtain

$$\frac{di_D}{dt} = -a_1 i_D + a_2\varphi_d + a_3\omega_r\varphi_q + a_4 v_D \tag{1}$$

$$\frac{di_Q}{dt} = -a_1 i_Q + a_2\varphi_q - a_3\omega_r\varphi_d + a_4 v_Q \tag{2}$$

$$\frac{d\varphi_d}{dt} = a_5 i_D - a_6\varphi_d - \omega_r\varphi_q \tag{3}$$

$$\frac{d\varphi_q}{dt} = a_5 i_Q - a_6\varphi_q + \omega_r\varphi_d \tag{4}$$

$$\frac{d\omega_r}{dt} = \frac{N_p}{J}[a_7(i_Q\varphi_d - i_D\varphi_q) - T_L] \tag{5}$$

where $L_s = M + L_1$, $L_r = M + L_2$, the positive constants a_i are

$$a_1 = \frac{R_r M^2}{\sigma L_s L_r^2} + \frac{R_s}{\sigma L_s}, \quad a_2 = \frac{R_r M}{\sigma L_s L_r^2}, \quad a_3 = \frac{M}{\sigma L_s L_r},$$
$$a_4 = \frac{1}{\sigma L_s}, \quad a_5 = \frac{R_r M}{L_r}, \quad a_6 = \frac{R_r}{L_r}, \quad a_7 = \frac{3N_p M}{2L_r}$$

and the parameter σ is given by $\sigma = 1 - M^2/(L_s L_r)$.

In what follows, we assume that the rotor speed is measurable, and then, we will consider ω_r as a time-varying parameter in the reduced-order dynamic model (1)–(4). The FOC model of the induction motor can be simply obtained by rewriting (1)–(4) in a *rotating* reference frame, synchronous and instantaneously in phase with the rotor flux component φ_d. In particular, by indicating with asterisks the variables in the rotating reference frame and, since under FOC $\varphi_q^* = 0$, (1)–(3) can be rewritten as follows:

$$\frac{di_D^*}{dt} = -a_1 i_D^* + a_2\varphi_d^* + \omega i_Q^* + a_4 v_D^* \tag{6}$$

$$\frac{di_Q^*}{dt} = -\omega i_D^* - a_1 i_Q^* - a_3\omega_r\varphi_d^* + a_4 v_Q^* \tag{7}$$

$$\frac{d\varphi_d^*}{dt} = a_5 i_D^* - a_6\varphi_d^* \tag{8}$$

where ω is the time derivative of the instantaneous phase of the rotor flux vector in the stationary reference frame. Using (4), rewritten in the rotating reference frame, ω can be expressed as

$$\omega = \omega_r + a_5\frac{i_Q^*}{\varphi_d^*}. \tag{9}$$

It should be noted that (9) is usually computed using the reference values $\bar{I}_Q^*$ and $\bar{\Phi}$ of the current i_Q^* and flux φ_d^*, respectively, thus neglecting the dependence on the instantaneous values of the current and flux [8]. However, for a more rigorous analysis, (9) must be substituted into (6) and (7), thus providing the following nonlinear third-order dynamic model:

$$\frac{di_D^*}{dt} = -a_1 i_D^* + a_2\varphi_d^* + \omega_r i_Q^* + a_5\frac{{i_Q^*}^2}{\varphi_d^*} + a_4 v_D^* \tag{10}$$

$$\frac{di_Q^*}{dt} = -\omega_r i_D^* - a_1 i_Q^* - a_3\omega_r\varphi_d^* - a_5\frac{i_D^* i_Q^*}{\varphi_d^*} + a_4 v_Q^* \tag{11}$$

$$\frac{d\varphi_d^*}{dt} = a_5 i_D^* - a_6\varphi_d^*. \tag{12}$$

In order to ensure zero steady-state errors on the torque and flux, an integrator is inserted in each of the corresponding feedback loops. The inputs of these integrators are the errors $z_D^* = \bar{\Phi} - \varphi_d^*$ and $z_Q^* = \bar{T} - a_7 i_Q^*\varphi_d^*$ between the reference values $\bar{\Phi}$ and $\bar{T}$ of the flux and the torque and their corresponding actual values, which will be shortly replaced with the estimated flux and torque. In order to also consider analytically this situation, the system state is enlarged by considering as further state variables z_D^* and z_Q^*. Therefore, the following two state equations must be added to (10)–(12):

$$\frac{dz_D^*}{dt} = -\varphi_d^* + \bar{\Phi}, \tag{13}$$

$$\frac{dz_Q^*}{dt} = -a_7 i_Q^*\varphi_d^* + \bar{T}. \tag{14}$$

C. The Linearized and Approximated Models

In order to design a linear state feedback controller on the FOC dynamic model, (10)–(14) must be linearized in the neighborhood of the corresponding equilibrium point. In particular, from (12)–(14) in steady state, we obtain

$$I_D^* = \frac{a_6\bar{\Phi}}{a_5} \tag{15}$$

$$\Phi_d^* = \bar{\Phi} \tag{16}$$

$$I_Q^* = \frac{\bar{T}}{a_7\bar{\Phi}} \tag{17}$$

where capital letters indicate steady-state values. Moreover, using (10) and (11), we obtain

$$V_D^* = \frac{1}{a_4}\left(a_1\frac{a_6\bar{\Phi}}{a_5} - a_2\bar{\Phi} - \omega_r\frac{\bar{T}}{a_7\bar{\Phi}} - a_5\frac{\bar{T}^2}{a_7^2\bar{\Phi}^3}\right) \tag{18}$$

$$V_Q^* = \frac{1}{a_4}\left(\omega_r\frac{a_6\bar{\Phi}}{a_5} + a_1\frac{\bar{T}}{a_7\bar{\Phi}} + a_3\omega_r\bar{\Phi} + \frac{a_6\bar{T}}{a_7\bar{\Phi}}\right). \tag{19}$$

Then, the linearized dynamic model of the FOC induction motor with the torque and flux integrators can be written as

$$\dot{\hat{x}}^* = A(\omega_r)\hat{x}^* + B\hat{v}^* \tag{20}$$

where

$$\hat{x}^* = (i_D^* - I_D^*, i_Q^* - I_Q^*, \varphi_d^* - \Phi_d^*, z_D^* - Z_D^*, z_Q^* - Z_Q^*)^T \tag{21}$$

$$\hat{v}^* = (v_D^* - V_D^*, v_Q^* - V_Q^*)^T \tag{22}$$

[2] Note that capital (small) pedices are used for stator (rotor) quantities.

the system matrices are shown as (23) and (24), at the bottom of the page, and Z_D^* and Z_Q^* are the steady-state outputs of the integrators. Note that, under closed-loop control , $Z_D^* = V_D^*$ and $Z_Q^* = V_Q^*$. Therefore, (18) and (19) can also be used as the feedforward action for the synchronous reference voltages.

The state feedback control should be designed on the linearized model (20). However, in order to implement the state controller, the steady-state values of the state variables should be computed using (15)–(19) and, consequently, they will depend on the reference torque and flux and on the motor parameters. In order to design a more simple controller, an approximated model can be used instead of (20). In particular, we assume that we can compute the frequency ω from (9) by using the steady-state variables I_Q^* and Φ_d^* instead of their corresponding instantaneous values i_Q^* and φ_d^*, i.e., we use the following equation:

$$\omega = \omega_r + \bar{\omega}_{sl} \tag{25}$$

with $\bar{\omega}_{sl} = a_5 \frac{\bar{T}}{a_7 \bar{\Phi}^2}$. Moreover, we assume that the flux reaches its steady-state value before a reference torque is applied, so that, in (14), which models the integrator on the torque error, we can substitute φ_d^* with $\bar{\Phi}$. A possible analytical justification of the last approximation might be obtained by searching for dynamic separations among the state variables of the dynamic model (10)–(14).

From the considerations given above, for the controller design, we will consider the following dynamic model:

$$\dot{x}^* = F(\omega_r)x^* + Bv^* + Gu \tag{26}$$

where

$$x^* = (i_D^*, i_Q^*, \varphi_d^*, z_D^*, z_Q^*)^T \tag{27}$$

$$v^* = (v_D^*, v_Q^*)^T \tag{28}$$

$$u = (\bar{\Phi}, \bar{T})^T \tag{29}$$

and the system matrices are

$$F(\omega_r) = \begin{pmatrix} -a_1 & \omega_r + \bar{\omega}_{sl} & a_2 & 0 & 0 \\ -\omega_r - \bar{\omega}_{sl} & -a_1 & -a_3\omega_r & 0 & 0 \\ a_5 & 0 & -a_6 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -a_7\bar{\Phi} & 0 & 0 & 0 \end{pmatrix} \tag{30}$$

$$G = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}^T. \tag{31}$$

It is interesting to note that, from (23) and (30), we can write

$$A(\omega_r) = F(\omega_r) + \delta F \tag{32}$$

where

$$\delta F = \begin{pmatrix} 0 & a_5\frac{\bar{T}}{a_7\bar{\Phi}^2} & -a_5\frac{\bar{T}^2}{a_7^2\bar{\Phi}^4} & 0 & 0 \\ -a_5\frac{\bar{T}}{a_7\bar{\Phi}^2} & -a_6 & \frac{a_6\bar{T}}{a_7\bar{\Phi}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\bar{T}}{\bar{\Phi}} & 0 & 0 \end{pmatrix}. \tag{33}$$

If we neglect the nonzero term on its last row, the matrix δF is in the range of the input matrix B and, therefore, the "uncertainties" on the dynamic model can be compensated by a suitable state feedback action. As regards the term $-\bar{T}/\bar{\Phi}$, we can say that the faster the rotor flux is with respect to the torque, the less the influence of the approximation in considering $F(\omega_r)$ instead of $A(\omega_r)$ it will be.

A further justification of the approximation done can be obtained by analyzing the eigenvalues of the matrices $A(\omega_r)$ and $F(\omega_r)$. In particular, these eigenvalues, for each value of the rotor speed and of the nominal slip frequency $\bar{\omega}_{sl}$, can be determined as the roots of the following equations obtained from the characteristic polinomials of $A(\omega_r)$ and $F(\omega_r)$, respectively, by neglecting the two zero eigenvalues due to the integrators:

$$\begin{aligned} 0 = {} & s^3 + (\delta_1 + a_6)s^2 + \left[\delta_2(\omega_r, \bar{\omega}_{sl}) + a_1a_6 + a_6^2 \right. \\ & \left. + 4\bar{\omega}_{sl}^2 + 2\omega_r\omega_{sl}\right]s + \delta_3(\omega_r, \bar{\omega}_{sl}) + a_1a_6^2 - a_2a_5a_6 \\ & + (2a_6 + a_1)\bar{\omega}_{sl}^2 + (a_6 + a_3a_5)\omega_r\bar{\omega}_{sl} \end{aligned} \tag{34}$$

$$0 = s^3 + \delta_1 s^2 + \delta_2(\omega_r, \bar{\omega}_{sl})s + \delta_3(\omega_r, \bar{\omega}_{sl}) \tag{35}$$

where

$$\begin{aligned} \delta_1 &= 2a_1 + a_6 \\ \delta_2(\omega_r, \bar{\omega}_{sl}) &= 2a_1a_6 + a_1^2 - a_2a_5\omega_r^2 + \bar{\omega}_{sl}^2 + 2\omega_r\bar{\omega}_{sl} \\ \delta_3(\omega_r, \bar{\omega}_{sl}) &= a_1^2a_6 - a_1a_2a_5 + (a_6 + a_3a_5)\omega_r^2 \\ &\quad + a_6\bar{\omega}_{sl}^2 + (2a_6 + a_3a_5)\omega_r\bar{\omega}_{sl}. \end{aligned}$$

Fig. 2 shows a plot of the roots of (34) and (35) for $\omega_r \in [0, 270]$ rad/s and $\bar{\omega}_{sl} \in [0, 7.1]$ rad/s. In particular, $\bar{\omega}_{sl} = 7.1$ rad/s corresponds to a maximum reference torque of $\bar{T} = 600$ N·m with a reference flux $\bar{\Phi} = 1.8$ Wb. The eigenvalues of $A(\omega_r)$ do not drastically depend on the

$$A(\omega_r) = \begin{pmatrix} -a_1 & \omega_r + 2a_5\frac{\bar{T}}{a_7\bar{\Phi}^2} & a_2 - a_5\frac{\bar{T}^2}{a_7^2\bar{\Phi}^4} & 0 & 0 \\ -\omega_r - 2a_5\frac{\bar{T}}{a_7\bar{\Phi}^2} & -a_1 - a_6 & -a_3\omega_r + \frac{a_6\bar{T}}{a_7\bar{\Phi}^2} & 0 & 0 \\ a_5 & 0 & -a_6 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -a_7\bar{\Phi} & -\frac{\bar{T}}{\bar{\Phi}} & 0 & 0 \end{pmatrix} \tag{23}$$

$$B = \begin{pmatrix} a_4 & 0 & 0 & 0 & 0 \\ 0 & a_4 & 0 & 0 & 0 \end{pmatrix}^T \tag{24}$$

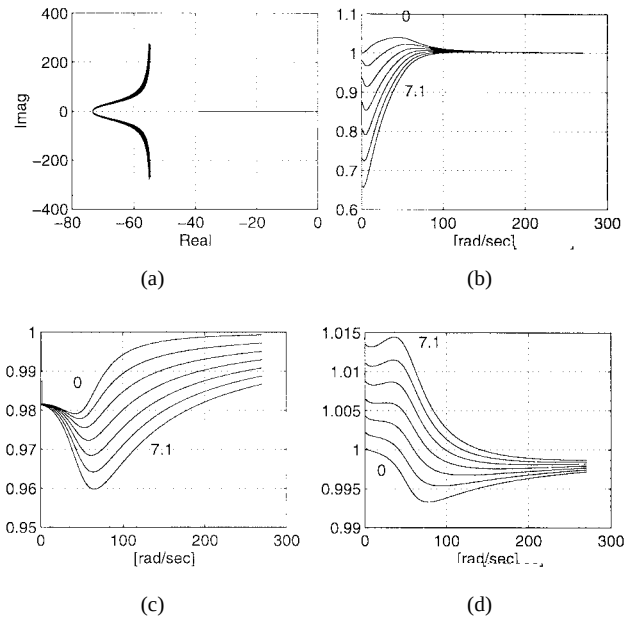

Fig. 2. Eigenvalues of $A(\omega_r)$ and $F(\omega_r)$. (a) Root locus of the eigenvalues of $A(\omega_r)$ with respect to ω_r and for constant 1 rad/s steps of $\bar{\omega}_{sl}$. (b) Ratio between the real eigenvalue of $A(\omega_r)$ and that of $F(\omega_r)$ as a function of ω_r for the same range of $\bar{\omega}_{sl}$. (c) Ratio between the magnitude of the complex eigenvalues of $A(\omega_r)$ and that of $F(\omega_r)$ with respect to ω_r. (d) Ratio between the phase of the complex eigenvalues of $A(\omega_r)$ and that of $F(\omega_r)$.

slip frequency. On the other hand, the eigenvalues are much more sensitive to ω_r; the complex eigenvalues have a larger imaginary part for larger values of ω_r, and the real eigenvalue has a larger modulus, which corresponds to a faster dynamic evolution, when the rotor speed increases. The other subplots show a comparison between the eigenvalues of the linearized and the approximated models. It is evident that the errors on the complex eigenvalues remain very small, less than 5% on the magnitude, and less than 2% on the phase. As regards the real eigenvalue, the differences between the two systems increase with the ratio between the slip frequency and the rotor speed. In other words, larger errors are made by approximating the eigenvalues of $A(\omega_r)$ with those of $F(\omega_r)$ when the rotor speed reduces and the reference torque increases. It is important to note that the dynamics of the linearized system are always faster than those of the approximated one. The simulation results reported below will also show the effectiveness of the proposed scheme in the worst operating conditions, i.e., small speed and large torque.

III. Controller Design

In this section, we detail the design procedures for three different controllers which are applied to the field-oriented-controlled induction motor.

A. LQ State Feedback

The LQ state feedback control is designed by using the approximated model (26). As is well known, in order to apply the LQ technique to a linear time-invariant system, the system must be controllable or, at least, stabilizable [11]. By using the Symbolic Toolbox of MATLAB, we verified that, for each value of the rotor speed, the pair $(F(\omega_r), B)$ is controllable. Therefore, the closed-loop control is obtained by applying to the dynamic model (26) the following linear state feedback:

$$v^* = -K^T(\omega_r)x^*. \tag{36}$$

For each value of the rotor speed, the feedback gain is obtained off-line as the steady-state LQ solution which minimizes the performance index

$$H = \int_0^\infty (x^{*T}Qx^* + v^{*T}Rv^*)\,dt \tag{37}$$

with the constant weighting matrices Q and R suitably chosen indipendently from the rotor speed. Therefore, the feedback gain matrix is given by $K^T(\omega_r) = R^{-1}B^TP(\omega_r)$ where $P(\omega_r)$ is the solution of the steady-state Riccati equation

$$\begin{aligned} &F(\omega_r)^TP(\omega_r) + P(\omega_r)F(\omega_r) \\ &\quad - P(\omega_r)BR^{-1}B^TP(\omega_r) + Q = 0. \end{aligned} \tag{38}$$

In order to implement the controller, the discrete feedback gain is computed, for each value of the rotor speed, by using the *lqrd* command of the Control System Toolbox of MATLAB, thus considering the discretization of the control at the 300-Hz sampling rate (see [10] for more details).

The weighting matrices have been assumed with a diagonal form. Their first trial values have been selected so that, by considering the rated values of the state variables, all terms in the perormance index (37) had the same magnitude order. Then, a further refinement was done by looking at the desired transient performance of the nonlinear closed-loop system, i.e., through simulations of the system represented in Fig. 1 with the motor model (1)–(5). For instance, the decrease of the third term in Q (which weighs the flux) and the increase of the fourth one (which weighs the flux error) allowed a faster response of the rotor flux. Moreover, the currents overshoot was limited by varying the first two terms of the Q matrix, according to the desired constraint on the maximum currents (in our case, we assumed 1.2 times the rated current). Following these guidelines, the trial-and-error procedure led to the weighting matrices

$$\begin{aligned} Q &= \mathrm{diag}(10^{-3}, 10^{-5}, 10^{-4}, 10^4, 1) \\ R &= \mathrm{diag}(10^{-3}, 10^{-3}). \end{aligned}$$

The discrete feedback gains as a function of the rotor speed are shown in Fig. 3. The adaptation of the feedback gains has been implemented via two different interpolation procedures. First, by using third-order interpolating polynomials and, second, by building a multilayer feedforward neural network (MFFN) architecture [12] with three hidden neurons and one output neuron for each component of the matrix gain. In simulations, both techniques have shown to provide identical performance.

B. Pole Placement State Feedback

Since the dynamic model (26) has been verified to be controllable for each value of the rotor speed, the state

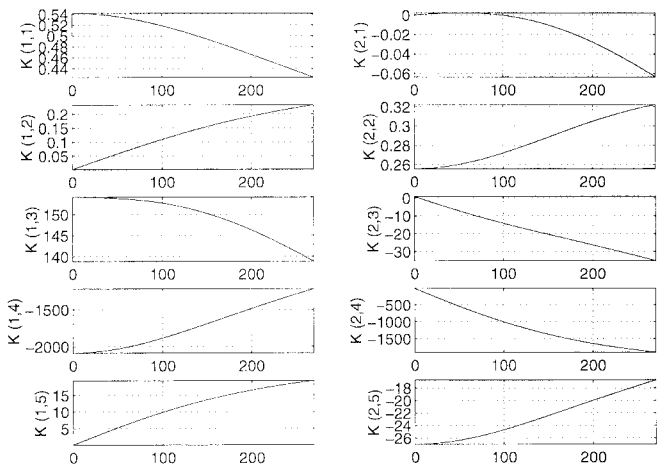

Fig. 3. Variations of the discrete LQ feedback gains with respect to the rotor speed.

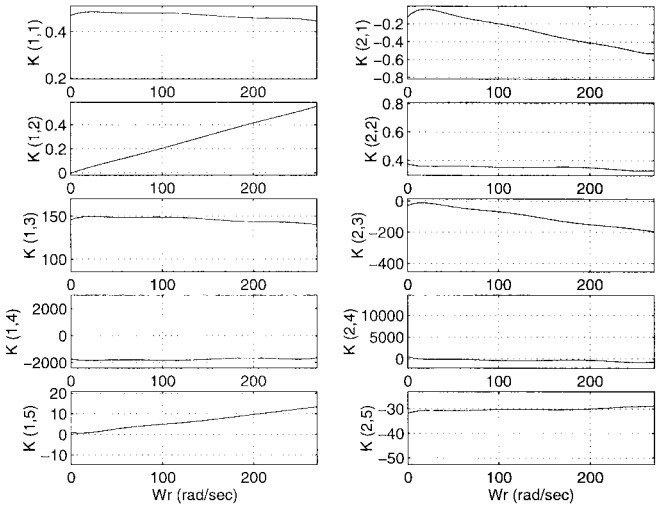

Fig. 4. Variations of the discrete pole placement feedback gains with respect to the rotor speed.

feedback gain $K^T(\omega_r)$ in (36) can also be designed by using the pole placement technique. The rotor-speed-dependent feedback gains are designed so that the closed-loop poles, given by the eigenvalues of the matrix $F(\omega_r) - K^T(\omega_r)B$, are fixed independently from ω_r. In other words, the state feedback counteracts the variation of the system poles due to the rotor speed variation.

The feedback gain matrix is simply designed; the desired closed-loop poles are fixed and then, for suitably chosen values of the rotor speed, the feedback gains which verify this constraint are obtained by using the place MATLAB command. These values are interpolated as described above in the case of the LQ controller. In order to let the approximated model verify the previously given constraints on the rotor flux settling time and on the currents overshoot, a good tradeoff for the choice of the closed-loop poles of the discretized system (with sampling frequency 300 Hz) has been found to be 0.36, $0.69 \pm j0.07$, and $0.95 \pm j0.04$. The corresponding discrete rotor-speed-dependent feedback gains are represented in Fig. 4.

C. PI Controller

In order to compare the state feedback controllers presented above with a more standard closed-loop scheme, let us consider the drive represented in Fig. 5. To compare fairly the PI-controlled system with that obtained via state feedback designs, the parameters of the PI were chosen looking at the simulations carried out using the nonlinear model (1)–(5). In particular, the same constraints imposed for the Q and R matrices selection in the LQ design procedure were considered.

The PI transfer fuctions have been assumed to have the form $k_p + k_i/s$, s being the complex variable. Following the described procedure, the PI parameters have been chosen equal to $k_p = 0.05$ and $k_i = 1$ for the torque error controller, $k_p = 250$ and $k_i = 230$ for the flux error controller, and $k_p = 0.1$ and $k_i = 1$ for the i_D current error.

IV. Observer Design

As shown in Fig. 1, the state feedback controller needs the rotor flux amplitude and the motor torque; moreover, the FOC loop needs the instantaneous rotor flux phase. Since the stator currents are assumed to be measurable, only the stationary rotor flux components must be estimated. To this aim, we propose an observer structure which is similar to that presented in [6]; given the stator currents and voltages, the stator flux components are estimated by a suitably designed observer and then used as inputs, together with the stator currents, for the neural network estimator. The network outputs are the rotor flux components and the torque. As will be detailed in this section, both parts of the observer are designed in order to also provide good estimation performance in the presence of parameters' variations.

A. The Stator Flux Observer

By introducing the stator flux components φ_D and φ_Q, and considering as inputs the stator voltages and currents, we can write

$$\varphi_D = \int (v_D - R_s i_D)\, dt \tag{39}$$

$$\varphi_Q = \int (v_Q - R_s i_Q)\, dt. \tag{40}$$

These equations could, in principle, be implemented by using analog integrators, as is shown in [6]. However, this solution suffers from errors on the estimated fluxes in the presence of biased measured input voltages and uncertainities on the stator resistance. In order to mitigate this problem, we used the integration algorithm with adaptive magnitude compensation proposed in [13]. This algorithm is able to counteract only small variations of the stator resistance. In the presence of large detuning conditions, the estimated stator fluxes will be different from the actual ones. However, as detailed below, a suitable design of the neural network estimator (and, in particular, of its training set) will allow us to also obtain good rotor fluxes estimations in the presence of errors on the estimated stator fluxes.

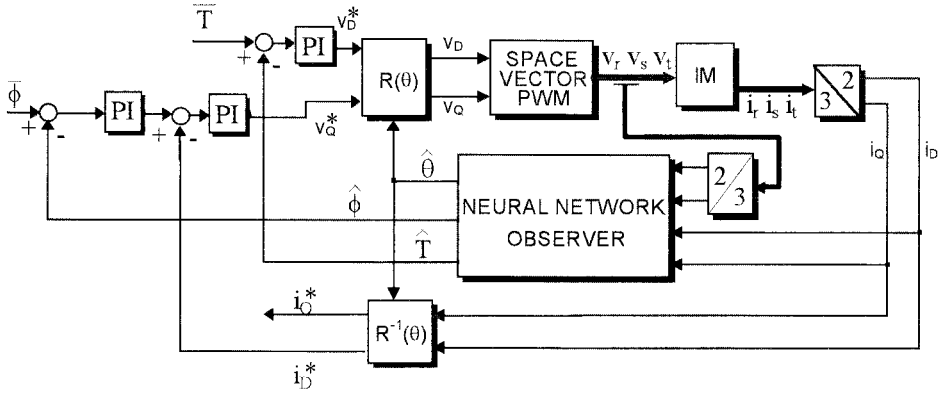

Fig. 5. Closed-loop scheme of the FOC induction motor with PI regulators.

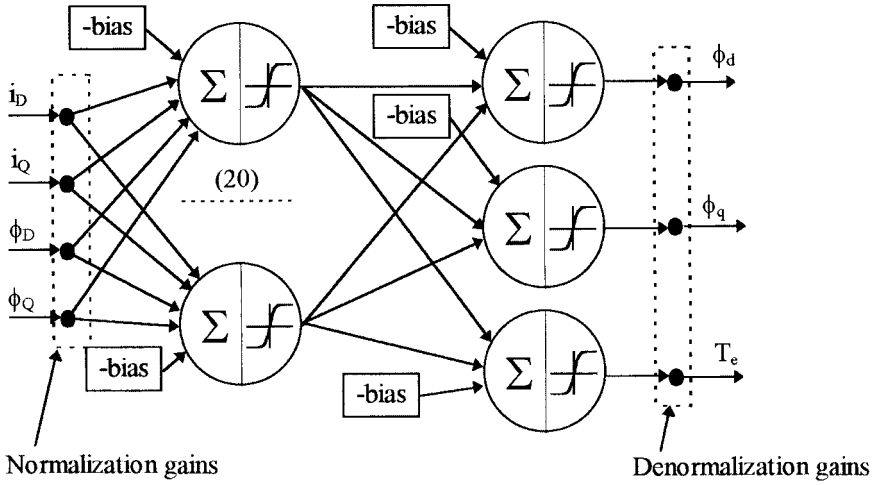

Fig. 6. Neural network structure.

B. The Neural Network

A neural network is used to obtain the rotor flux and the torque, given the stator currents (measured) and the stator fluxes (estimated). This can be done by exploiting the following well-known relations:

$$\varphi_d = \frac{L_r}{M}\varphi_D - \sigma L_s i_D \tag{41}$$

$$\varphi_q = \frac{L_r}{M}\varphi_Q - \sigma L_s i_Q \tag{42}$$

$$T_e = \frac{3}{2}N_p(\varphi_D i_Q - \varphi_Q i_D). \tag{43}$$

An MFFN has been used to implement (41)–(43). The neural network structure is shown in Fig. 6. As usual, the basic structure of the neurons consists of a summing device, that performs a weighted and biased sum of its inputs, and an output function, which is a nonlinear monotonically increasing function. The weights and biases of all the neurons in the whole network are adjustable in the training phase. In what follows, we assume that the nonlinear function used in the network is the so-called hyperbolic-tan function, whereas the training procedure, differently from [6], is the Levenberg–Marquardt backpropagation algorithm [14].

As shown in Fig. 6, the neural network observer considered in this paper has four inputs, three output neurons, and a single hidden layer with 20 neurons.

In order to avoid a possible lack of precision of the estimated variables due to parameters uncertainties and/or variations, in what follows, we propose a suitable design of the network training set.

C. The Training Set Design

In order to allow the neural network to generalize properly, its training set should take explicitly into account the input/output pairs resulting from reasonable parameter variations. These kinds of input/output pairs are generated by suitable simulations of the ideal LQ FOC induction motor. In other words, we simulated the whole scheme presented in the previous section in an ideal environment, in which the rotor fluxes are assumed to be measurable.

Some slight modifications are inserted in these simulations to accomodate the training set richness and the parameter variations. In particular, following the procedure detailed in [15], random signals uniform in the interval of $\pm 10\%$ of the reference voltages, are added to the stator voltages in order to

ensure the richness of the training set in the neighborhood of the desired operating conditions. Moreover, at fixed time steps, the motor parameters are varied within a suitable designed region in the parameter space. In what follows, we detail how the selection of this region has been done, which is the key point of the training set design.

By assuming that the rotor speed and the stator currents are measurable, and the corresponding errors to be independent Gaussian random variables with zero mean and given standard deviations σ_{ω_r} and σ_i, an identification problem can be set up. The aim of this problem is to find the parameter vector $\pi = (R_s, R_r, L_s, L_r, M)$ which minimizes the following so-called reduced χ^2 function [16]:

$$G(\pi) = \frac{1}{3N-5} \sum_{n=1}^{N} \left\{ \frac{[\tilde{\omega}_r(n) - \omega_r(n,\pi)]^2}{\sigma_{\omega_r}^2} + \frac{[\tilde{i}_D(n) - i_D(n,\pi)]^2}{\sigma_i^2} + \frac{[\tilde{i}_Q(n) - i_Q(n,\pi)]^2}{\sigma_i^2} \right\} \tag{44}$$

where the term $\frac{1}{3N-5}$ is the inverse of the number of degrees of freedom (the number of measured points minus the number of parameters to be estimated), the symbol "˜" is used to denote measured variables, and n identifies the nth sampling time of measured and simulated variables. It is important to stress that, by neglecting the measurement errors, we can say that the only reason for differences between measured and simulated variables is due to the parameters mismatch between the actual motor and the simulated model.

In order to stop the recursive minimization procedure at a "good" point, we use the χ^2 test for the goodness of fit [16]. In other words, once a step of suitable minimization procedure has been completed, said $G_{\min}$ the minimum value of the objective function (44), if $G_{\min}$ will be less than 1.5, this will be indicative of a good fitting of the measured points and the recursive minimization procedure will be stopped.

Therefore, once σ_{ω_r} and σ_i have been chosen, we have to determine the region in the parameter space which contains all points π such that $G(\pi) < 1.5$, i.e., the region of all parameters yielding a good fit for the measurements according to the hypotheses made on the errors. To this aim, the so-called Price algorithm has been used.

D. The Price Algorithm

The Price algorithm is a direct, global optimization algorithm [17], [18]. It works with the objective function values calculated in a set of points of the parameters search volume. This set of points is called a grid. In a first phase of the algorithm, the grid points are initially chosen as random points uniformly distributed in the interior of the search volume. Said L the number of parameters (the size of the vector π), and S the number of grid points, in order to ensure a successful Price algorithm the inequality $S \gg L$ must be verified. After the first phase is completed, the algorithm works as detailed in the following.

1) Compute the grid point $\pi_{\max}$ in which (44) attains the maximum value, i.e., $\pi_{\max} = \arg[\max_{i=1,\cdots,S} G(\pi_i)]$.
2) Set $G_{\max} = G(\pi_{\max})$.
3) Choose $L+1$ different grid points at random, for example, $\pi_1, \cdots, \pi_{L+1}$.
4) Determine the centroid, $\underline{\pi}$, of the first L points chosen at the third step, i.e., $\underline{\pi} = \frac{1}{L} \sum_{i=1}^{L} \pi_i$.
5) Generate the trial point $\pi_t = 2\underline{\pi} - \pi_{L+1}$; if π_t is not contained in the search volume, return to the third step.
6) Calculate $G(\pi_t)$; if $G(\pi_t) < G_{\max}$, then $\pi_{\max}$ is eliminated from the grid, and it is substituted by the trial point π_t.
7) Compute the new $G_{\max}$, if necessary;
8) If a suitable convergence criterion is not satisfied, return to the first step.

The convergence criterion chosen to detect the end of the minimization procedure is the previously described χ^2 test for the goodness of a fit, applied to $G_{\max}$ only. It is important to stress that, because of the structure of the Price algorithm, at each step, $G_{\max} \geq G(\pi_i)$ with $i = 1, \cdots, S$. Therefore, since the χ^2 test is applied at each step to the maximum value $G_{\max}$, we can say that, if the test is verified for $G_{\max}$, it will be verified for all the other grid points too.

It is possible to show that the larger the number of grid points S, the larger is the number of potential different trial points generated at each iteration, thus ensuring completeness of the search. The more complete the search is, the larger is the probability to find a global minimum. In [17], it has been shown that, if the objective function has only one global minimum in the search volume, the distribution of the grid points tends to become a Gaussian distribution, having as the mean the global minimum point and variance decreasing with the increase of the number of iterations.

As a concluding remark, we can say that, at the end of the minimization procedure, the grid points are an approximation of the region of reasonable parameters we were looking for.

In Fig. 7, the projections of this region on different planes are reported for $\sigma_{\omega_r} = 10$ rad/s and $\sigma_i = 10$ A. It is interesting to note that the region obtained is an ellipsoid, which corresponds to the well-known *confidence ellipsoid* [16]. Moreover, the ellipsoid is centered at the nominal parameters, which suggests that the Price algorithm can be used for identification purposes, also.

V. Simulation Results

A. The Simulated Drive

In this section, we present the simulation results carried out by considering the induction motor drive represented in Fig. 1 with the dynamic model (1)–(5). The simulations were performed by using the code written in ANSI C language and presented in [9].

All simulations were obtained by considering the presence of a space-vector PWM voltage-source inverter with dc supply of 1.5 kV and constant switching frequency of 300 Hz [19]. Moreover, the constant reference flux has been chosen equal to 1.8 Wb and the trapezoidal reference torque starts at 0.5 s, increases to 400 N·m in 1 s, is held constant for 2 s,

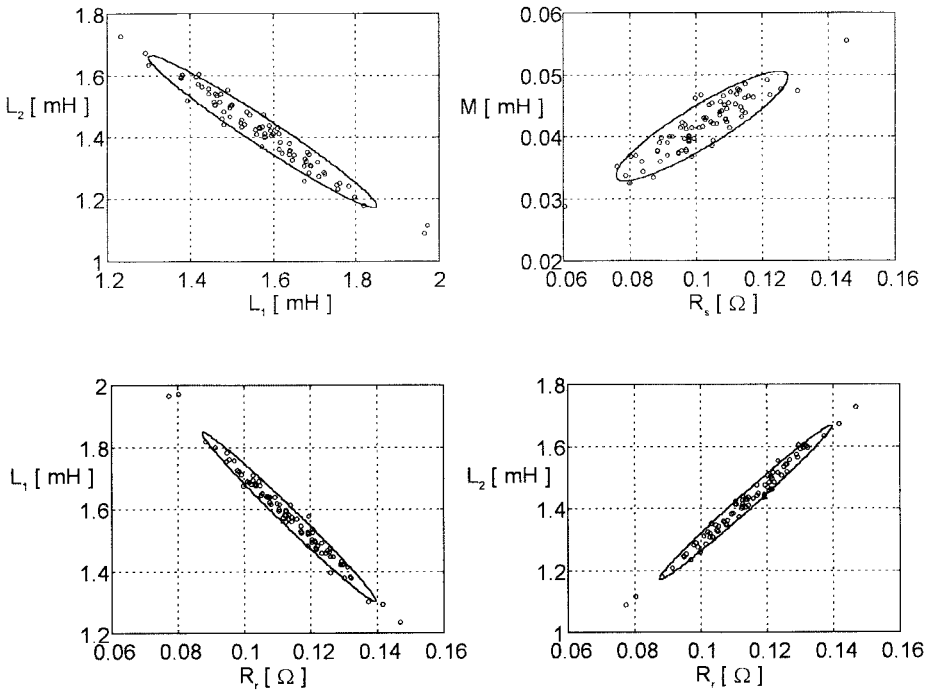

Fig. 7. Confidence ellipsoid represented on different parameter space planes.

decreases to −400 N·m in 2 s and, again after 2 s, increases with the same slope. It is important to stress that these reference signals yield larger model approximation errors, due to the corresponding small speed and large torque operating conditions (see Section II).

Although not shown in this paper, similar results have been obtained in the presence of a speed regulation loop which has been implemented by adding a further PI controller, the output of which provides the reference torque as shown with dashed line in Fig. 1. In order to also consider critical operating conditions, the startup transient has been included in the simulations. Therefore, the initial conditions of the stator currents, the rotor fluxes, and the rotor speed have been set to zero.

Under closed-loop operating conditions and in the presence of matched and detuned parameters, the performance of the robust neural network observer is compared with that achieved with an EKF [2] and with the observer proposed in [6]. In particular, in order to construct the EKF model, the induction motor states have been enlarged with the rotor time constant, and the rotor speed has been assumed to be measurable. The intial conditions of the EKF observer have been set to zero, except for the rotor time constant, which has been set to its nominal value. For the EKF algorithm, the covariance matrices of the process and measurements noises have been chosen, respectively, equal to

$$Q_{\mathrm{EKF}} = \mathrm{diag}(10^2, 10^2, 10^{-4}, 10^{-4}, 10)$$
$$R_{\mathrm{EKF}} = \mathrm{diag}(10^{-2}, 10^{-2}).$$

In what follows, we detail the results obtained with LQ, pole placement, and PI control, in three different situations corresponding to ideal parameters matching, slight detuning, and large detuning. To perform the simulations in the worst case, we assumed that the considered detuning held from the initial time instant, without introducing further parameters variations during the simulations.

B. Ideal Parameters Matching

Let us assume that the actual induction motor parameters coincide with the nominal ones, the values of which have been presented above. The different observers will then be designed by using the nominal induction motor parameters.

In the simulations, all the estimators worked well, and only slight differences were checked among the corresponding estimated variables. Therefore, we will present the results by considering only the robust neural network observer presented in this paper. Fig. 8 shows the electrical and mechanical variables obtained when using the LQ controller. Similar results are obtained with the pole placement controller. To verify the improvements obtained with the state feedback controllers with respect to the PI one, Fig. 9 shows the flux and currents obtained with the PI and a comparison with the corresponding variables previously represented. It is interesting to note that the LQ controlled drive, although it provides a smaller current overshoot, ensures faster response with smaller overshoot on the rotor flux magnitude. This advantage will be maintained in all the other simulations, even when not explicitly highlighted.

Since several simulations have shown that the LQ controller is superior to the others considered here, in what follows, we will present only the results obtained in the presence of the LQ controller.

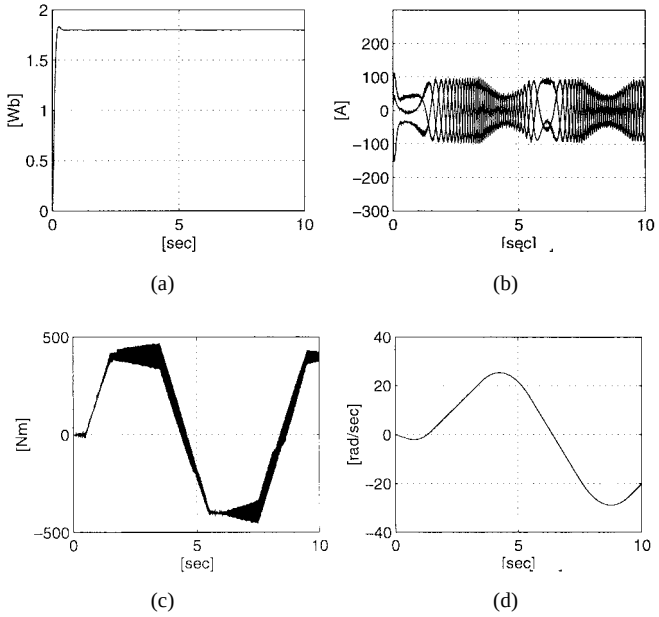

Fig. 8. Electrical and mechanical variables in the presence of robust neural network observer and LQ controller. (a) Actual flux magnitude. (b) Three-phase stator currents. (c) Torque. (d) Rotor speed.

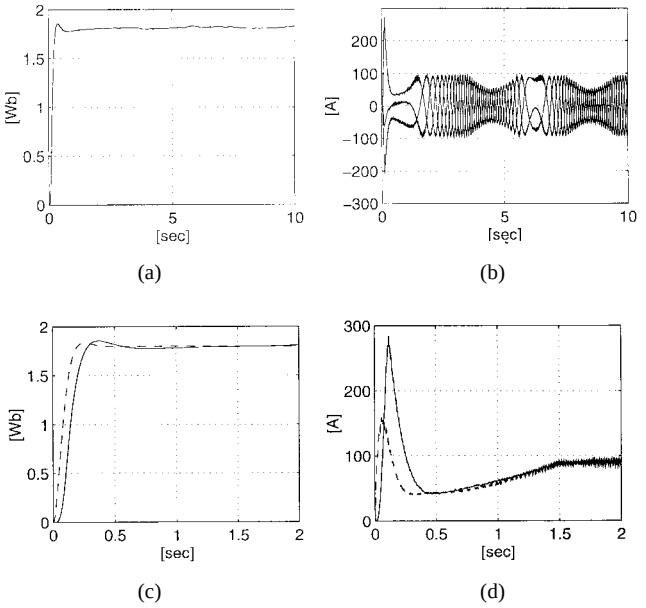

Fig. 9. Electrical variables in the presence of robust neural network observer. (a) Actual flux magnitude with PI. (b) Three-phase stator currents with PI. (c) Rotor flux magnitude with LQ (dashed) and PI. (d) Stator current magnitude with LQ (dashed) and PI.

C. Slight Parameters Detuning

In order to check the robustness of the different observers, we first assume that the actual parameters of the induction motor are slightly detuned with respect to the nominal ones, as follows:

$$\frac{\Delta R_s}{R_s} = -2\%, \quad \frac{\Delta R_r}{R_r} = -10\%$$
$$\frac{\Delta L_1}{L_1} = +5\%, \quad \frac{\Delta L_2}{L_2} = -5\%, \quad \frac{\Delta M}{M} = -3\%.$$

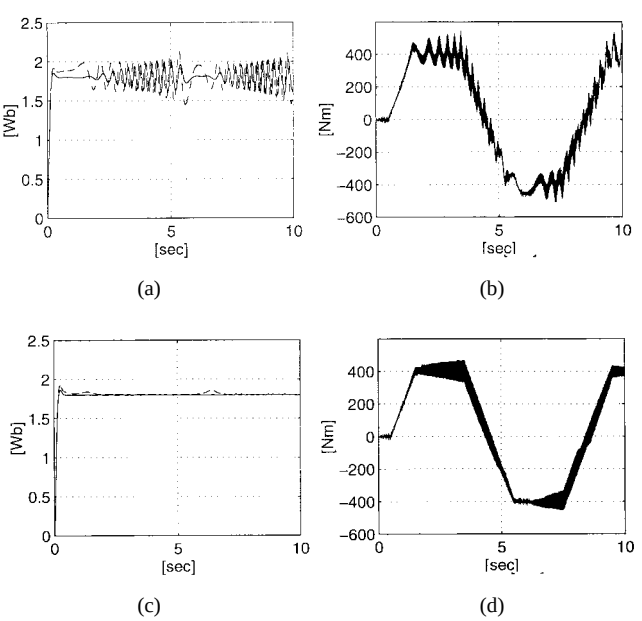

Fig. 10. Electrical variables in the presence of LQ controller. (a) Actual (dashed) and estimated flux magnitude with neural network. (b) Torque with neural network. (c) Actual (dashed) and estimated flux magnitude with EKF. (d) Torque with EKF.

In other words, we designed the observers referring to the nominal parameters of the motor, whereas the simulated motor was assumed to be detuned according to Fig. 7.

The results obtained with our neural network estimator are not reported because they do not change significantly with respect to those presented in the previous subsection.

Fig. 10 shows the flux magnitude and the torque obtained with the EKF and the observer presented in [6]. From this figure, it is evident that the EKF is superior to the neural network. This happens because the EKF algorithm contains an adaptation strategy, whereas the network is designed as a pattern recognition system without any adaptation mechanism. It is important to stress that, with our robust neural network, also, we do not introduce any on-line adaptation mechanism, but its training set itself exploits information on the variation or detuning of the motor parameters, thus allowing us to overcome the problem.

D. Large Parameters Detuning

In many real applications, the motor parameters are known with uncertainties or may vary on ranges which are larger than those considered above. In order to check the robustness of the estimators in the presence of larger detuning, let us consider the following parameter uncertainties:

$$\frac{\Delta R_s}{R_s} = -30\%, \quad \frac{\Delta R_r}{R_r} = -50\%$$
$$\frac{\Delta L_1}{L_1} = +10\%, \quad \frac{\Delta L_2}{L_2} = -10\%, \quad \frac{\Delta M}{M} = -10\%.$$

Under this situation, both the EKF and the neural network work very badly, i.e., they provide large estimation errors or do not converge at all and, consequently, both the PI and the state

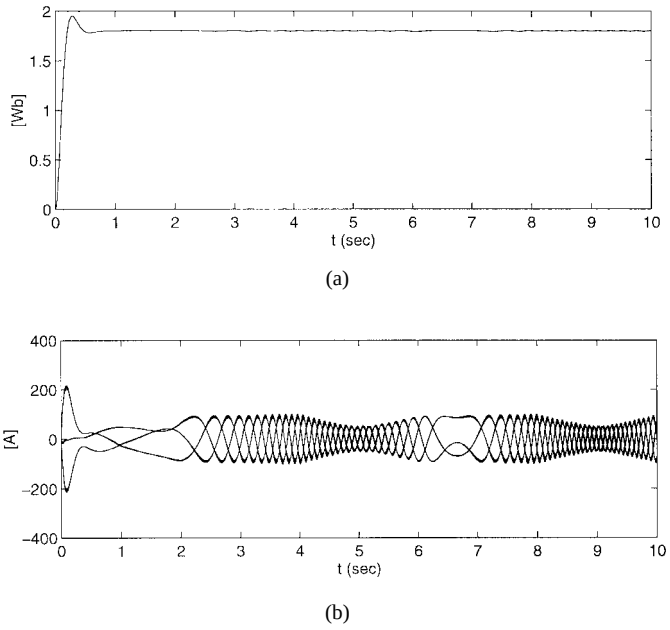

Fig. 11. Electrical variables in the presence of LQ controller and the robust neural network observer. (a) Actual flux magnitude. (b) Three-phase stator currents.

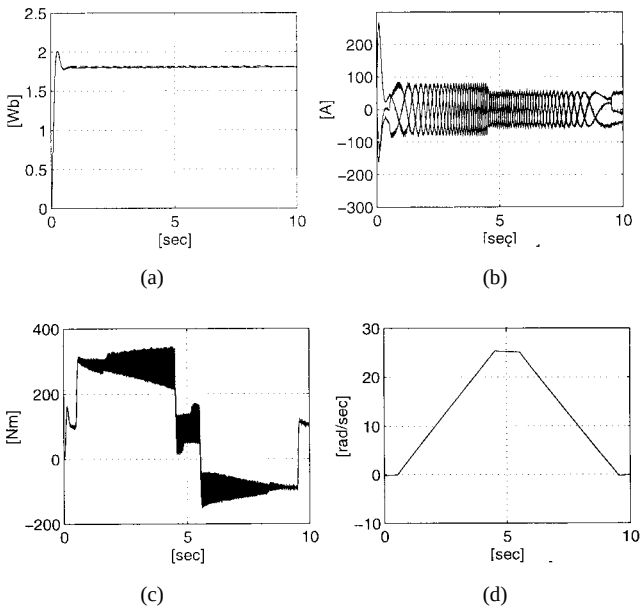

Fig. 12. Electrical variables in the presence of LQ controller and the robust neural network observer with speed control loop. (a) Actual (dashed) and estimated flux magnitude. (b) Three-phase stator currents. (c) Torque. (d) Rotor speed.

feedback control schemes provide completely unsatisfactory performance.

Fig. 11 shows the results obtained with the LQ controller and our robust neural network observer. It can be noted that the performance with this configuration is still satisfactory, despite the quite severe detuning conditions considered.

Fig. 12 shows the performance of the LQ controller with the robust neural network estimator, in the presence of a speed control loop. The rotor speed is regulated by means of a PI regulator, which provides the reference torque to the LQ regulator. The parameters of this PI controller are $k_p = 150$ and $k_i = 450$. The reference speed profile is similar to that of the reference torque; it varies from 0 to 25 rad/s in 4 s, starting from 0.5 s, then it remains constant for 1 s, returning to 0 in 4 s. The simulation is performed in the specified large detuning conditions. It can be seen that the reference profile is followed with accuracy, without significant overshoots on the rotor speed or on stator currents and rotor flux. This simulation, together with the other results presented, confirms the high-performance quality of the proposed drive system.

E. Some Considerations for Practical Implementation

A low-cost fast implementation of the proposed control system can be simply done by using analog neural network chips. The computation time of these chips is less than 0.1 μs per neuron [12], thus allowing fast processing. The rotor-speed-dependent gains implementation does not require a large number of neurons; four neurons per gain component are necessary to approximate the ten gain components. Thus, the total number of neurons needed for the system implementation, including the neurons of the neural network rotor flux observer is about 70 neurons. This number of neurons can be implemented on a single chip. The computation time of such a structure is much smaller than the turn-off time of a GTO, which can be assumed to be 100 μs. Since the modulation index of the space-vector PWM inverter is limited by this turn-off time, a practically instantaneous control system can be implemented by starting the inverter gating counters exactly 100 μs after the stator currents, voltages, and rotor speed have been sampled. The control calculations can be performed in the requested time limit.

The described analog neural network chip has the drawback of small flexibility. Another way to implement it is by using a digital signal processor (DSP) board. With a DSP, a larger number of neurons can be implemented, and the system is more flexible, because changing network weights is easier. Moreover, there is also the possibility to implement a different rotor flux observer, such as, for instance, an EKF (see [9]).

VI. Conclusion

An LQ regulator has been used as state feedback controller for a field-oriented-controlled induction motor. The flux and torque errors are assumed as further model states, and the rotor speed is assumed to be measurable together with the stator currents. The resulting performance index is quite straightforward to tune, since it penalizes the physically meaningful states directly. Having fixed the performance index, the feedback gains are calculated off-line for different steady-state rotor speeds. The calculated gains are suitably interpolated off-line, thus providing a rotor-speed-dependent gain which is then implemented on-line in order to also obtain good performance during constant acceleration and speed transients. A similar procedure has also been shown to be suitable in order to implement a rotor-speed-dependent feedback gain designed through pole placement technique. The rotor fluxes and the torque are estimated by using a neural-network-based observer.

In order to construct the training set, the LQ-controlled motor and realistic parameters uncertainities, estimated by the Price algorithm, have been considered. The whole system, both the controller and the flux estimator, can be implemented with a neural network, which is easily available as a commercial chip, thus obtaining a very compact, economic, reliable, and fast FOC system. Several numerical investigations have shown the effectiveness of the proposed scheme.

REFERENCES

[1] G. C. Verghese and S. R. Sanders, "Observers for flux estimation in induction machines," *IEEE Trans. Ind. Electron.*, vol. 35, pp. 85–94, Feb. 1988.

[2] L. C. Zai, C. L. De Marco, and T. A. Lipo, "An extended Kalman filter approach to rotor time constant measurement in PWM induction motor drives," *IEEE Trans. Ind. Applicat.*, vol. 28, pp. 96–104, Jan./Feb. 1992.

[3] L. Glielmo, P. Marino, R. Setola, and F. Vasca, "Reduced Kalman filtering for indirect adaptive control of induction motors," *Int. J. Adapt. Control Signal Process.*, vol. 8, no. 6, pp. 527–541, 1994.

[4] P. Marino, V. Mungiguerra, F. Russo, and F. Vasca, "Parameter and state estimation for induction motors via interlaced least squares algorithm and Kalman filter," in *Proc. PESC'96*, Baveno, Italy, 1996, pp. 1235–1241.

[5] J. Stephan, M. Bodson, and J. Chiasson, "Real-time estimation of the parameters and fluxes of induction motors," *IEEE Trans. Ind. Applicat.*, vol. 30, pp. 746–758, May/June 1994.

[6] M. G. Simões and B. K. Bose, "Neural network based estimation of feedback signals for a vector controlled induction motor drive," *IEEE Trans. Ind. Applicat.*, vol. 31, pp. 620–629, May/June 1995.

[7] P. Marino, M. Milano, and F. Vasca, "Robust neural network observer for induction motor control," in *Conf. Rec. PESC'97*, St. Louis, MO, 1997, pp. 699–705.

[8] B. K. Bose, *Power Electronics and AC Drives*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[9] B. Busco, G. De Marco, P. Marino, V. Mungiguerra, M. Porzio, F. Russo, and F. Vasca, "Flux observation and parameter estimation of induction motors for traction drives," in *Conf. Rec. PESC'96*, Baveno, Italy, 1996, pp. 1408–1413.

[10] *Control System Toolobx User's Guide*, The MathWorks Inc., Natick, MA, July 1992.

[11] B. D. O. Anderson and J. B. Moore, *Optimal Control*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[12] S. Haykin, *Neural Networks*. New York: McMillan, 1994.

[13] J. Hu and B. Wu, "New integration algorithms for estimating motor flux over a wide speed range," in *Conf. Rec. 1997*, St. Louis, MO, 1997, pp. 1075–1081.

[14] *Neural Network Toolbox User's Guide*, The MathWorks Inc., Natick, MA, Jan. 1994.

[15] T. S. Low, T. H. Lee, and H. K. Lim, "A methodology for neural network training for control of drives with nonlinearities," *IEEE Trans. Ind. Electron.*, vol. 39, pp. 243–249, Apr. 1993.

[16] P. R. Bevington, *Data Reduction and Error Analysis for the Physical Sciences*. New York: McGraw-Hill, 1969.

[17] L. Milano, F. Barone, and M. Milano, "Time domain amplitude and frequency detection of gravitational waves from coalescing binaries," *Phys. Rev. D*, vol. 55, no. 8, pp. 4537–4554, Apr. 1997.

[18] W. L. Price, "A controlled random search procedure for global optimization," *Comput. J.*, vol. 20, no. 4, pp. 367–370, 1976.

[19] H. W. Van der Broeck, H. C. Skudelny, and G. V. Stanke, "Analysis and realization of a pulsewidth modulator based on voltage space vectors," *IEEE Trans. Ind. Applicat.*, vol. 24, pp. 142–150, Jan./Feb. 1988.

Pompeo Marino was born in Frosinone, Italy, in 1948. He received the "Laurea" degree in electronics engineering in 1973 from the Università di Napoli Federico II, Naples, Italy.

In 1975, he was appointed as a Research Associate in the Dipartimento di Informatica e Sistemistica, Università di Napoli. He was then with the Università di Cosenza from 1977 to 1979. In 1984, he joined the Dipartimento di Informatica e Sistemistica, Università di Napoli, as an Associate Professor of Automatic Control. In 1989, he became an Associate Professor of Industrial Electronics. In 1995, he joined the Dipartimento di Ingegneria dell'Informazione, Seconda Università di Napoli, Aversa, Italy, as an Associate Professor of Industrial Electronics. Since 1995, he has been a Research Group Member of the Sistemi Elettrici per l'Energia of the CNR. His interests include power converters modeling, ac and dc drives motion control, and power quality in electrical systems.

Michele Milano was born in Vico Equense, Napoli, Italy, in 1969. He received the "Laurea" degree in electronics engineering from the Università di Napoli Federico II, Naples, Italy, in 1996.

He is currently an Academic Guest at ETH, Zurich, Switzerland. He is engaged in research on the application of neurocomputing techniques to the control of combustion engines. His research interests include neural networks, fuzzy logic, genetic algorithms, and control theory.

Francesco Vasca (S'94–M'95) was born in Giugliano, Napoli, Italy, in 1967. He received the "Laurea" and Ph.D. degrees from the Università di Napoli Federico II, Naples, Italy, in 1991 and 1995, respectively.

Since 1996, he has been a Research Assistant in Automatic Control with the Facoltà di Ingegneria di Benevento, Università degli Studi del Sannio, Benevento, Italy. His research interests include modeling and control techniques for power converters and induction motors, chaos in switching systems, Kalman filtering for nonlinear systems, and neural network applications.

Dr. Vasca is a member of the IEEE Control Systems, IEEE Industrial Electronics, and IEEE Power Electronics Societies.