Científica, Vol. 13 Núm. 4, pp. 183-192.

A Direct Adaptive Vector Neural Control of a Three-Phase Induction Motor

Ieroham S. Baruch
Irving Pavel de-la-Cruz

Department of Automatic Control,
CINVESTAV-IPN, Av. IPN núm. 2508
México, DF.
MÉXICO.

Tel. 5747 3800 ext. 4229, fax: 5747 3812.

Correo electrónico: baruch@ctrl.cinvestav.mx

Recibido el 6 de diciembre de 2007; aceptado el 10 de agosto de 2009.

1. Abstract

The paper proposes a complete neural solution to the direct vector control of three phase induction motor including real-time trained neural controllers for velocity, flux and torque, which permitted the speed up reaction to the variable load. The basic equations and elements of the direct field oriented control scheme are given. The control scheme is realized by nine feedforward neural networks learned by real-time Backpropagation or off-line Levenberg-Marquardt algorithms with data taken by PI-control simulations. The graphical results of modelling show a better performance of the neural control system with respect to the PI controlled system realizing the same general control scheme.

Key words: Modelling and Simulation, Induction Motor, Field Oriented Control, Direct Vector Control, Neural Networks, Levenberg - Marquardt Learning, Backpropagation.

2. Resumen (Control vectorial neuronal directo adaptable de un motor trifásico de inducción)

El artículo propone una completa solución neuronal al control vectorial directo de un motor trifásico de inducción que incluye controladores neuronales de velocidad, flujo y torque entrenados en tiempo real, lo cual permite acelerar la reacción a carga variable. Las ecuaciones básicas y los elementos del esquema de control directo de orientación del campo magnético, están dados. El esquema de control esta realizado a partir de nueve redes neuronales con conexiones hacia delante, entrenadas en tiempo real con el algoritmo de retropropagacion del error o el algoritmo fuera de línea de Levenberg-Marquardt con datos tomados de la simulación usando el control PI. Los resultados gráficos de modelación muestran un mejor comportamiento del control neuronal con respecto al control PI realizando el mismo esquema de control general.

Palabras clave: modelación y simulación, motor de inducción, control de orientación del campo magnético, control vectorial directo, redes neuronales, aprendizaje de Levenberg-Marquardt, retropropagación.

3. Introduction

The Neural Networks (NN) applications for identification and control of electrical drives became very popular in last decade. In [1], an adaptive neuro-fuzzy system is applied for a stepping motor drive control. In [2], a multilayer perception-based-neural-control is applied for a DC motor drive. In [3], a recurrent neural network is applied for identification and adaptive control of a DC motor drive mechanical system. In the last decade a great boost is made in the area of induction motor drive control. The induction machine, particularly the cage type, is most commonly used in adjustable speed AC drive systems [4]. The control of AC machines is considerably more complex than that of DC machines. The complexity arises because of the variable-frequency power supply, AC signals processing, and complex dynamics of the AC machine [4], [5]. In the vector or Field-Oriented Control (FOC) methods, an AC machine is controlled like a separately excited DC machine, where the active (torque) and the reactive (field) current components are orthogonal and mutually decoupled so they could be controlled independently [4]-[8]. There exists two methods for PWM current controlled inverter –direct and indirect vector control [4]. This paper applied the direct control method, where direct AC motor measurements are used for field orientation and control. There are several papers of NN application for AC motor drive direct vector control. In [9] a feedforward NN is used for vector PW modulation, resulting in a faster response. In [10] an ADALINE NN is used for cancellation of the integration DC component during the flux estimation. In [11] a fuzzy-neural uncertainty observer is

integrated in a FOC system, using an estimation of the rotor time constant. In [12] an Artificial NN is used for fast estimation of the angle ρ used in a FOC system. In [13] a flux and torque robust NN observer is implemented in a FOC system. In [14], an ADALINE-based-filter and angular-velocity-observer are used in a FOC system. The NN training is online using the Total Least Squares (TLS) algorithm. In [15], the authors proposed a NN velocity observer used in FOC high performance system for an induction motor drive. In [16] a Feedforward-NN (FFNN)-based estimator of the feedback signals is used for induction motor drive FOC system. The paper [17] proposed two NN-based methods for FOC of induction motors. The first one used a NN flux observer in a direct FOC. The second one used a NN for flux and torque decoupling in an indirect FOC. The results and particular solutions obtained in the referenced papers shows that the application of NN offers a fast and improved alternative of the classical FOC schemes. The present paper proposed a total neural solution of a direct FOC problem which assures fast response and adaptation to a variable load. This is achieved applying real-time learned P/PI neural controllers of IM velocity, flux and torque.

4. Models of the Induction Machine

4.1. A Phase (*a*, *b*, *c*) Model

The Induction Motor (IM) equations [6], [7], for stator and rotor voltages in vector-matrix form are given as:

$$\mathbf{v}_{abcs} = \mathbf{r}_s \mathbf{i}_{abcs} + p\boldsymbol{\lambda}_{abcs} \quad (1)$$

$$\mathbf{v}_{abcr} = \mathbf{r}_r \mathbf{i}_{abcr} + p\boldsymbol{\lambda}_{abcr} \quad (2)$$

$$\mathbf{r}_s = r_s \mathbf{I}_3 \; ; \; \mathbf{r}_r = r_r \mathbf{I}_3 \quad (3)$$

where:

$$\begin{aligned} &\mathbf{v}_{abcs} = (v_{as}, v_{bs}, v_{cs})^T \; ; \; \mathbf{v}_{abcr} = (v_{ar}, v_{br}, v_{cr})^T \\ &\mathbf{i}_{abcs} = (i_{as}, i_{bs}, i_{cs})^T \; ; \; \mathbf{i}_{abcr} = (i_{ar}, i_{br}, i_{cr})^T \\ &\boldsymbol{\lambda}_{abcs} = (\lambda_{as}, \lambda_{bs}, \lambda_{cs})^T \; ; \; \boldsymbol{\lambda}_{abcr} = (\lambda_{ar}, \lambda_{br}, \lambda_{cr})^T \end{aligned} \quad (4)$$

Are: voltage, current, and flux, stator and rotor, three dimensional (*a*, *b*, *c*) vectors, with given up phase components; r_s and r_r are stator and rotor winding resistance diagonal matrices, with given up equal elements r_s and r_r, respectively; $\mathbf{I}_3$ is an identity matrix with dimension three, and *p* is a Laplacian differential operator. The vector-matrix block-form representation of the flux leakage is given by the equation:

$$\begin{bmatrix} \boldsymbol{\lambda}_{abcs} \\ \boldsymbol{\lambda}'_{abcr} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{ss}^{abc} & \mathbf{L}'^{abc}_{sr} \\ (\mathbf{L}'^{abc}_{sr})^T & \mathbf{L}'^{abc}_{rr} \end{bmatrix} \begin{bmatrix} \mathbf{i}_{abcs} \\ \mathbf{i}'_{abcr} \end{bmatrix} \quad (5)$$

Where: the stator, rotor and mutual block-inductance (3x3) matrices are described in [6], [7]. The relative leakage inductance depends on the winding turn stator/rotor ratio *n*, and on the angular rotor position θ_r respectively [6], [7]. Finally, the voltage equations (1), (2) could be expressed with respect to the stator in the (*a*, *b*, *c*) model form:

$$\begin{bmatrix} \mathbf{v}_{abcs} \\ \mathbf{v}'_{abcr} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_s + p\mathbf{L}_{ss}^{abc} & p\mathbf{L}'^{abc}_{sr} \\ (p\mathbf{L}'^{abc}_{sr})^T & \mathbf{r}_r + p\mathbf{L}'^{abc}_{rr} \end{bmatrix} \begin{bmatrix} \mathbf{i}_{abcs} \\ \mathbf{i}'_{abcr} \end{bmatrix} \quad (6)$$

Where the relative rotor voltage, current, flux and resistance values are:

$$\begin{aligned} &\mathbf{v}'_{abcs} = n\mathbf{v}_{abcs} \; ; \quad \mathbf{i}'_{abcr} = (1/n)\mathbf{i}_{abcr} \\ &\boldsymbol{\lambda}'_{abcr} = n\boldsymbol{\lambda}_{abcr} \; ; \quad \mathbf{r}'_r = n^2 \mathbf{r}_r \end{aligned} \quad (7)$$

4.2. A (*q*, *d*, 0) Model

The (*a*, *b*, *c*) model is very complicated for control, so it could be simplified using a transformation to the (*q*, *d*, 0) form. The AC motor equations for the stator and rotor voltages in vector-matrix form are given as follows:

$$\mathbf{v}_{qd0s} = \mathbf{r}_s \mathbf{i}_{qd0s} + \boldsymbol{\Omega}\boldsymbol{\lambda}_{qd0s} + p\boldsymbol{\lambda}_{qd0s} \quad (8)$$

$$\mathbf{v}'_{qd0s} = \mathbf{r}_s \mathbf{i}'_{qd0s} + \boldsymbol{\Delta\Omega}\boldsymbol{\lambda}'_{qd0s} + p\boldsymbol{\lambda}'_{qd0s} \quad (9)$$

where:

$$\begin{aligned} &\mathbf{v}_{qd0s} = (v_{qs}, v_{ds}, v_{0s})^T \; ; \; \mathbf{v}'_{qd0s} = (v'_{qr}, v'_{dr}, v'_{0r})^T \\ &\mathbf{i}_{qd0s} = (i_{ds}, i_{ds}, i_{0s})^T \; ; \; \mathbf{i}'_{qd0s} = (i'_{qr}, i'_{dr}, i'_{0r})^T \\ &\boldsymbol{\lambda}_{qd0s} = (\lambda_{ds}, \lambda_{ds}, \lambda_{0s})^T \; ; \; \boldsymbol{\lambda}'_{qd0s} = (\lambda'_{qr}, \lambda'_{dr}, \lambda'_{0r})^T \end{aligned} \quad (10)$$

Are: voltage, current, and flux, stator and rotor, three dimensional (*q*, *d*, 0) vectors, with given up components; $\mathbf{r}_s$ and $\mathbf{r}_r$ are stator and rotor resistance diagonal matrices, given by (3); Ω, and ΔΩ are diagonal angular velocity matrices, given by:

$$\boldsymbol{\Omega} = \begin{bmatrix} \omega & 0 & 0 \\ 0 & -\omega & 0 \\ 0 & 0 & 0 \end{bmatrix} \; ; \; \boldsymbol{\Delta\Omega} = \begin{bmatrix} \omega - \omega_r & 0 & 0 \\ 0 & -(\omega - \omega_r) & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (11)$$

The vector-matrix block-form representation of the flux leakage is given by the equation:

Científica, Vol. 13 Núm. 4 (2009), pp. 183-192. ISSN 1665-0654, ESIME-IPN.

$$\begin{bmatrix} \boldsymbol{\lambda}_{qd0s} \\ \boldsymbol{\lambda}'_{qd0s} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{ss}^{qd0} & \mathbf{L}'^{qd0}_{sr} \\ (\mathbf{L}'^{qd0}_{sr})^T & \mathbf{L}'^{qd0}_{rr} \end{bmatrix} \begin{bmatrix} \mathbf{i}_{qd0s} \\ \mathbf{i}'_{qd0s} \end{bmatrix} \tag{12}$$

Where: the stator, rotor and mutual block-inductance (3x3) matrices are given in [6], [7]. The (q, d, 0) model could be written in the stationary and synchronous frames taking the angular velocity equal to: $\omega=0$ and $\omega=\omega_e$, where ω_e, corresponds to the angular velocity of the stator field. Now we could write the scalar electromagnetic torque equation which could be expressed in the following basic forms used:

$$T_{em} = \frac{3}{2}\frac{P}{2}\boldsymbol{\lambda}'^{T}_{q-d,r}\mathbf{i}'_{q-d,s} \tag{13}$$

$$T_{em} = \frac{3}{2}\frac{P}{2}\mathbf{i}'^{T}_{q-d,r}\mathbf{i}_{q-d,s} \tag{14}$$

Where: P is a number of poles.

$$\begin{aligned} &\mathbf{i}_{q-d,s} = (i_{qs}, i_{ds})^T ;\ \mathbf{i}'_{q-d,r} = (i'_{qr}, -i'_{dr})^T \\ &\boldsymbol{\lambda}'_{q-d,r} = (\lambda'_{qr}, \lambda'_{dr})^T \end{aligned} \tag{15}$$

4.3. Field Orientation Conditions

The flux and torque equations decoupling needs to transform the stator flux, current and voltage vectors from (a, b, c) reference frame to (q–d,s) reference frame and than to stationary and synchronous reference frame. The Fig. 1 illustrates the current and voltage vector representations in stator and rotor synchronous frames. The Fig. 1 shows also the magnetic field orientation, where the rotor flux vector is equal to the d-component of the flux vector, represented in a synchronous reference frame ($\lambda'^{e}_{dr} = \lambda_r$), which is aligned with the d-component of the current in this frame. For more clarity, the current and flux orientation in the synchronous reference frame are shown on Fig. 2. So, the field orientation conditions are the following, [7]:

$$\lambda'^{e}_{qr} = 0 ;\ p\lambda'^{e}_{qr} = 0 ;\ \lambda_r = \lambda'^{e}_{dr} \tag{16}$$

Taking into account that the rotor windings are shortcut, (the rotor voltage is zero) and the field orientation conditions, given by (16), the first two components of the equation (9), obtained the form:

$$\begin{aligned} 0 &= r'_r i'^{e}_{qr} + (\omega_e - \omega_r)\lambda'^{e}_{dr} \\ 0 &= r'_r i'^{e}_{dr} + p\lambda'^{e}_{dr} \end{aligned} \tag{17}$$

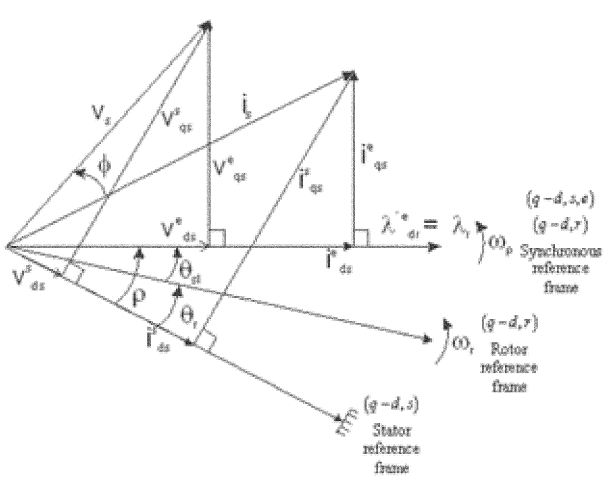

Fig. 1. The current and voltage vector representations in stator and in rotor synchronous reference frames.

From (12), we could obtain the q-component of the rotor flux, as it is:

$$\begin{aligned} &\lambda'^{e}_{qr} = L_m i^{e}_{qs} + L'_r i'^{e}_{qr} = 0 ;\quad L'_{lr} + L_m \\ &i'^{e}_{qr} = -(L_m / L'_r) i^{e}_{qs} \end{aligned} \tag{18}$$

Taking into account the condition (16), the torque equation (13) could be written in the form:

$$T_{em} = -\frac{3}{2}\frac{P}{2}\lambda'^{e}_{dr} i'^{e}_{qr} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L'_r}\lambda'^{e}_{dr} i'^{e}_{qs} \tag{19}$$

The equation (19) shows that if the flux of the rotor is maintained constant, so the torque could be controlled by

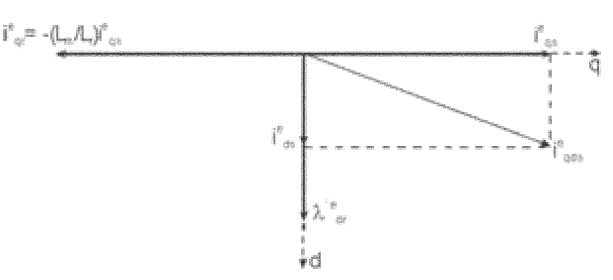

Fig. 2. The stator current and the rotor flux vector representations in synchronous reference frame.

the q-component of the stator current in synchronous reference frame. From the second equation of (17) and (18) it is easy to obtain the slipping angular velocity as:

$$\omega_e - \omega_r = (r'_r L_m / L'_r)(i^e_{qs} / \lambda'^e_{dr}) \tag{20}$$

The final equation (20) gives us the necessary basis for a direct decoupled field oriented (vector) control of the AC motor drive, where following the Figure 2, the q-component of the stator current produced torque and the d-component of the stator current produced flux.

4.4. Coordinate Transformations

The combined stator current transformation from (a, b, c) to (q–d,s,e) synchronous reference frame [6], [7], is given by the equation:

$$\begin{bmatrix} i^e_{qs} \\ i^e_{ds} \end{bmatrix} = \begin{bmatrix} \cos\rho & f_1 & f_2 \\ \sin\rho & f_3 & f_4 \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \tag{21}$$

$$\begin{aligned} f_1 &= [-(1/2)\cos\rho - (\sqrt{3}/2)\sin\rho] \\ f_2 &= [-(1/2)\cos\rho + (\sqrt{3}/2)\sin\rho] \\ f_3 &= [-(1/2)\sin\rho + (\sqrt{3}/2)\cos\rho] \\ f_4 &= [-(1/2)\sin\rho - (\sqrt{3}/2)\cos\rho] \end{aligned} \tag{22}$$

4.5. Flux and Torque Estimation

From the equation (8), written for the stationary reference frame (ω=0), we could obtain:

$$\begin{aligned} \lambda^s_{qs} &= (1/p)\,(v^s_{qs} - r_s i^s_{qs}) \\ \lambda^s_{ds} &= (1/p)\,(v^s_{ds} - r_s i^s_{ds}) \end{aligned} \tag{23}$$

The stator flux part of the equation (12) could be resolved for the rotor currents, as it is:

$$\begin{aligned} i'^s_{qr} &= (\lambda^s_{qs} - L_s i^s_{qs}) / L_m \\ i'^s_{dr} &= (\lambda^s_{ds} - L_s i^s_{ds}) / L_m \end{aligned} \tag{24}$$

Substituting (24) back in the rotor flux part of (12) we could obtain:

$$\lambda'^s_{qr} = (L'_r / L_m)(\lambda'^s_{qs} - L_s i^s_{qs})\,; \tag{25}$$

$$\lambda'^s_{qr} = (L'_r / L_m)(\lambda'^s_{qs} - L_s i^s_{qs}) \tag{25}$$

$$L'_s = [L_s - (L^2_m / L'_r)] \tag{26}$$

Where: L_s, L_r, L_m are stator, rotor and mutual inductances, and the prime supper index signifies its relative values. Now it is easy to compute the angle ρ needed for field orientation, the rotor flux, and the sin, cos functions of this angle, needed for flux control, torque estimation, and coordinate transformations, as it is:

$$\begin{aligned} \lambda'_r &= \sqrt{(\lambda'^s_{qr})^2 + (\lambda'^s_{dr})^2}\ ; \\ \rho &= \tan^{-1}(\lambda'^s_{qr} / \lambda'^s_{dr})\ ; \\ \sin\rho &= \lambda'^s_{qr} / \lambda'_r\ ;\ \ \cos\rho = \lambda'^s_{qr} / \lambda'_r \end{aligned} \tag{27}$$

The torque equation (19) could be rewritten in the form:

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L'_r}\lambda'_r\, i^e_{qs} \tag{28}$$

5. Direct Vector Control of the IM

5.1. A General Control Scheme

A general block diagram of the direct vector control of the Induction Motor drive is given on Figure 3.

The direct control scheme contains three principal blocks. They are: G_1, G_2, G_3 blocks of PI controllers; block of coordinate (abc) to (q–d,s,e) transformation (see equation (21)); block of vector estimation, performing the field orientation and the torque, flux and angle computations (see equations (27, (28)); block of inverse (q–d,s,e) to (a,b,c) transformation; block of the converter machine system and induction motor. The block of the converter machine system contains a current three phase hysteresis controller; a three phase bridge ASCI DC-AC current fed inverter; an induction motor model; a model of the whole mechanical system driven by

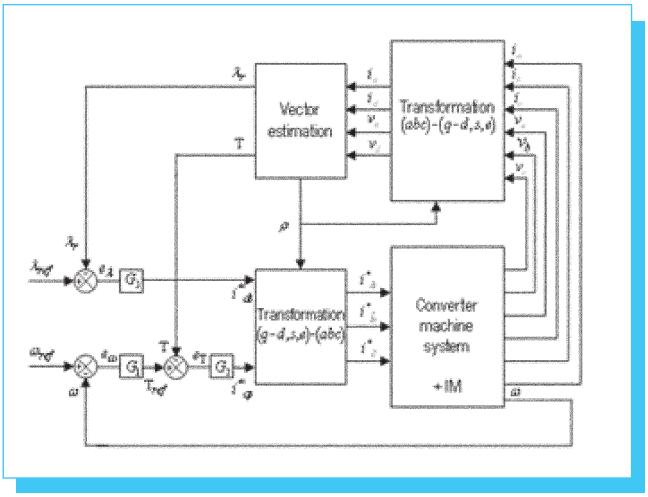

Fig. 3. General block-diagram of a direct IM vector control.

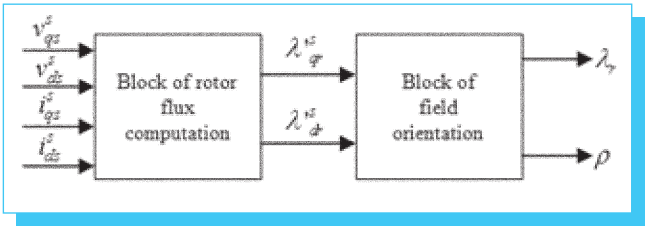

Fig. 4. Block-diagram of the vector estimation computations.

the IM ($(2/P)J(d\omega_r/dt) = T_{em} - T_L$, where J is the moment of inertia, T_L is the load torque). The block of vector estimation performed rather complicated computations, so it contains various blocks, illustrated by the next figures. The Figure 4 illustrates the flux and angle estimation for field orientation, computing (23), (25), (26), (27). The rotor flux computations block (see Figure 4) performs computations given by (23), (25), (26), illustrated by the Figure 5. The rotor flux, the angle, and the sin, cos functions computations, given by equation (27) are illustrated by the Figure 6.

The Figure 7 illustrates the torque estimation computed by equation (28). The next paragraph describes the feedforward neural network realization of these computations.

5.2. A Feedforward Neural Network (FFNN) Realization of the Direct Vector Control Scheme

The simplified block-diagram of the direct neural vector control system, given on Figure 3 is realized by nine FFNNs. We will describe in brief the function, the topology and the learning of each FFNN. The main contribution here is the introduction of the neural P/PI velocity, flux and torque controllers which are capable to adapt the control system to load changes.

FFNN1: The first NN1 is an angular velocity neural PI controller with two inputs (the velocity error, and the total

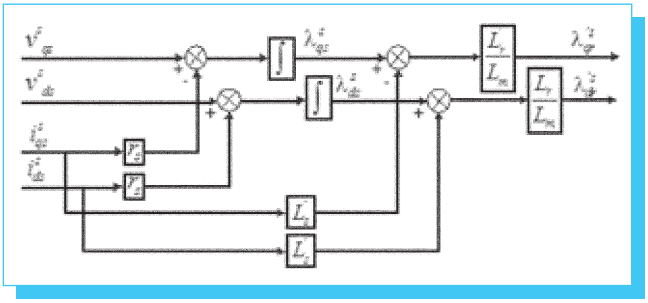

Fig. 5. Block-diagram of the flux estimation computations.

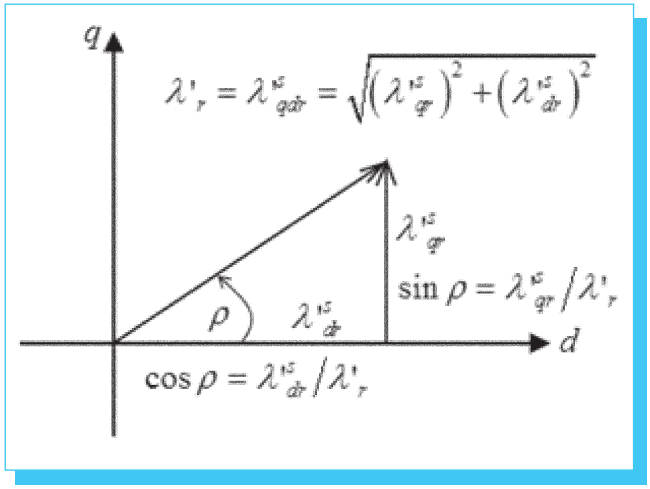

Fig. 6. Illustration of the trigonometric functions computation necessary for flux and angle estimations.

sum of velocity errors) and one output (the torque set point). The weights learning is done in real-time using the Backpropagation (BP) algorithm, [3]. The FFNN1 function is given by the following equation:

$$T^*(k) = \varphi[g_p(k)e^{sum}_{vel} + g_i(k)e_{vel}(k)] \qquad (29)$$

Where: g_p and g_i are proportional and integral FFNN1 weights; φ is a tanh activation function; e_{vel} is a velocity error; ; T^* is the torque set point –output of the FFNN1. The integration sum of errors is:

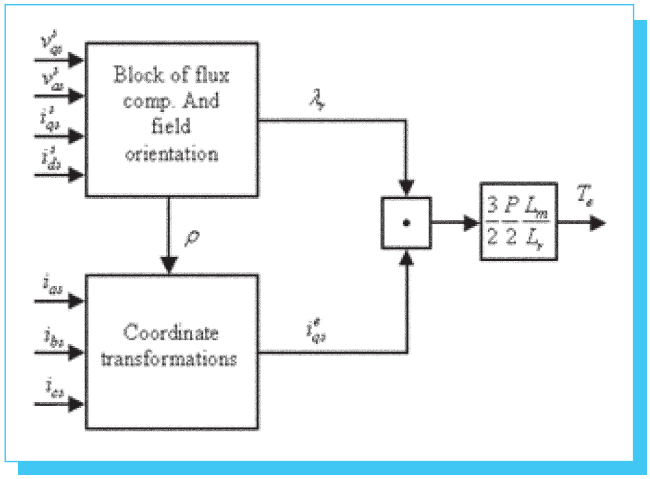

Fig. 7. Block-diagram of the torque estimation computations.

$$e_{vel}^{sum}(k) = \sum_{k=0}^{n} e_{vel}(k) \tag{30}$$

Where n is the total number of iterations. The BP algorithm for this FFNN1 is given by:

$$\begin{aligned} g_p(k+1) &= g_p(k) + \eta e_{vel}(k)[1 - (T^*(k))^2]e_{vel}(k) \\ g_i(k+1) &= g_i(k) + \eta e_{vel}(k)[1 - (T^*(k))^2]e_{vel}^{sum}(k) \end{aligned} \tag{31}$$

FFNN2: The second NN2 is a torque neural P controller with one input and one output (the torque error and the stator q-current set point). The function and the real-time BP learning, [3], of this FFNN2 are given by:

$$i_{qs}^{e*}(k) = \phi[g_p(k)e_T(k)] \tag{32}$$

$$g_p(k+1) = g_p(k) + \eta e_T(k)[1 - (i_{qs}^{e*}(k))^2]e_T(k) \tag{33}$$

Where: g_p is a proportional weight; ϕ is a tanh activation function; e_T is a torque error; η is a learning rate parameter; i_{qs}^{e*} is a current set point –output of FFNN2.

FFNN3: The third NN3 is a flux neural PI controller with two inputs and one output (the flux error and its sum, and the stator d-current set point). The function and the real-time BP learning, [3], of this NN3 are given by:

$$i_s^{e*}(k) = \varphi[g_p(k)e_{flux} + g_p(k)e_{flux}^{sum}(k)] \tag{34}$$

$$\begin{aligned} g_p(k+1) &= g_p(k) + \eta e_{flux}(k)[1 - (i_{ds}^{e*}(k))^2]e_{flux}^{sum}(k) \\ g_i(k+1) &= g_i(k) + \eta e_{flux}(k)[1 - (i_{ds}^{e*}(k))^2]e_{flux}^{sum}(k) \end{aligned} \tag{35}$$

Where: g_p and g_i are proportional and integral FFNN3 weights; φ is a tanh activation function; e_{flux} is a flux error; η is a learning rate parameter; i_{ds}^{e*} is a current set point –output of FFNN3. The integration sum of errors is:

$$e_{vel}^{sum}(k) = \sum_{k=0}^{n} e_{flux}(k) \tag{36}$$

Where n is a total number of iterations;.

FFNN4: The fourth FFNN4 is a torque off-line trained neural estimator (realizing (28) equation computation) which has two inputs and one output (the rotor flux, the stator q-current, and the estimated torque). The topology of this multilayer FFNN4 is (2-10-1) and the off-line algorithm of its learning is the Levenberg-Marquardt (LM) one [18], [19]. The FFNN4 is learned by 2500 input-output patterns (half period) and generalized by another 2 500 ones (the other half period) during 61 epochs. The final value of the MSE reached during the learning is of 10^{-10}.

FFNN5: The fifth NN5 performed a stator current (a,b,c) to (q–d,s,e) transformation (using (21), (22) equations). The FFNN5 topology has five inputs (three i_{as}, i_{bs}, i_{cs} stator currents; sinρ, cosρ), two outputs (i_{qs}^{e}, i_{ds}^{e} stator currents) and two hidden layers of 30 and 20 neurons each (5-30-20-2). The FFNN5 learning is off-line, applying the Levenberg-Marquardt algorithm [18], [19]. The final value of the MSE reached during the learning is of 10^{-10}.

The FFNN5 is learned by 2500 input-output patterns and generalized by 2500 ones during 29 epochs of learning.

FFNN6: The sixth NN6 performed an inverse stator current (q–d,s,e) to (a,b,c) transformation (using the transpose of the transformation matrix in (21), (22) equations). The NN6 topology has four inputs (two i_{qs}^{e}, i_{ds}^{e} stator currents; sinρ, cosρ), three outputs (i_{as}, i_{bs}, i_{cs} stator currents) and two hidden layers of 30 and 10 neurons each (4-30-10-3). The NN6 learning is off-line, applying the Levenberg-Marquardt algorithm [18], [19]. The final value of the MSE reached during the learning is of 10^{-10}. The NN6 is learned by 2500 input-output patterns and generalized by 2500 ones during 32 epochs.

FFNN7: The seventh NN7 performed rotor flux estimation using (27) equation. The rotor (q–d,r) flux components λ_{qs}^{s}, λ_{ds}^{s} are previously computed using eqn. (23) (see Fig.5), and they are inputs of FFNN7. The other two inputs are the stator currents: i_{qs}^{s}, i_{ds}^{s}. The FFNN7 output is the rotor flux: λ'_r. The FFNN7 topology is (4-30-10-1). The FFNN7 learning is off-line, applying the Levenberg-Marquardt algorithm [18], [19]. The final value of the MSE reached is of 10^{-10}. The FFNN7 is learned and generalized by 2500 input-output patterns in 35 epochs.

FFNN8, FFNN9: The FFNN8, and FFNN9 are similar to FFNN7 and performed separately the q and d rotor flux components estimation using (25), (26) equations. The FFNN8, FFNN9 topologies are: (2-10-5-2).The FFNN8, FFNN9 learning is off-line, applying the Levenberg-Marquardt algorithm [18], [19]. The final value of the MSE reached during the learning is of 10^{-10}. The FFNN8, FFNN9 are learned and generalized by 2500 input-output patterns during 47 and 49 epochs of learning, respectively. The values of sinρ, cosρ, needed for the coordinate transformations (see eqns. (27)) are obtained dividing the outputs of NN8, NN9 by the output of NN7.

6. Graphical Results of Control System Modelling

The parameters of the IM used in the control system modelling are: power 20Hp; nominal velocity N=1800 Rev.pm; pole number P = 4; voltage 220 volts; nominal current 75 A; phase number 3; nominal frequency 60 Hz; stator resistance r_s = 0,1062 ohms; rotor resistance referenced to stator r_r' = 0.0764 ohms; stator inductance L_s = 0.5689. 10-3 Henry; rotor inductance referenced to stator L_r' = 0.5689. 10^{-3} Henry; magnetizing inductance L_m = 15.4749. 10-3 Henry; moment of inertia J = 2.8 kg.m^2. The control system modeling is done changing the load torque in different moment of time. The Figure 8 and Figure 9 show the angular velocity set point *vs.* the IM angular velocity control (PI and NN control) with load torque changes. The results show that the angular velocity neural control system has a fast speed up response and satisfactory behaviour in the case of load change. This type of control gives better results with respect to the PI control (inferior overshoot in load changes). The Figure 10 shows the flux graphics of control system with PI control *vs.* neural control without and with load changes. The results show a faster and better response of the neural system, which try to maintain the flux constant in the case of load changes. The Figure 11 and Figure 12 show the graphical results of IM torque control for system with PI and neural control without and with load changes. The comparison of the Fig.11 a,b shows that the neural control has smaller overshoots than the PI control. The same could be observed in the Fig.11 c for the IM system start. In the Fig.12 the results also show a faster and better response (smaller overshoots in load changes) of the neural system in both of the cases. The Figure

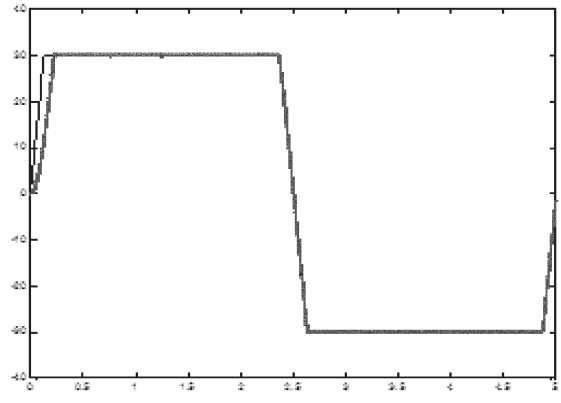

Fig. 8. Graphics of trajectory tracking angular velocity control (set point, PI-control, neural control).

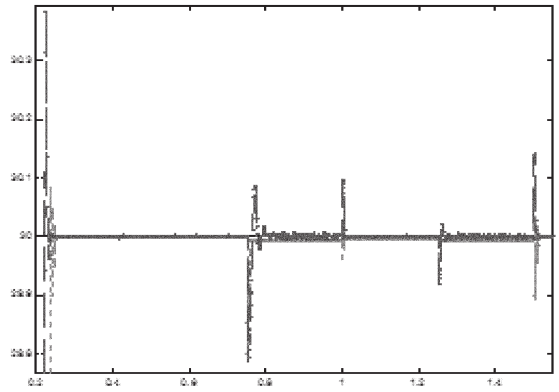

Fig. 9. Graphical results of angular velocity control with load changes (set point, PI-control, neural control).

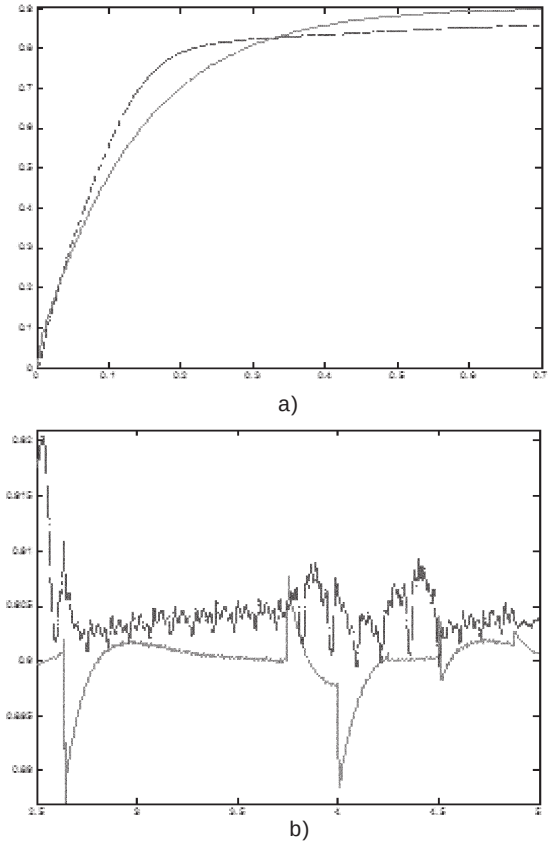

Fig. 10. Graphics of flux PI control vs. neural control. a) IM Start; b) Control with Load Changes.

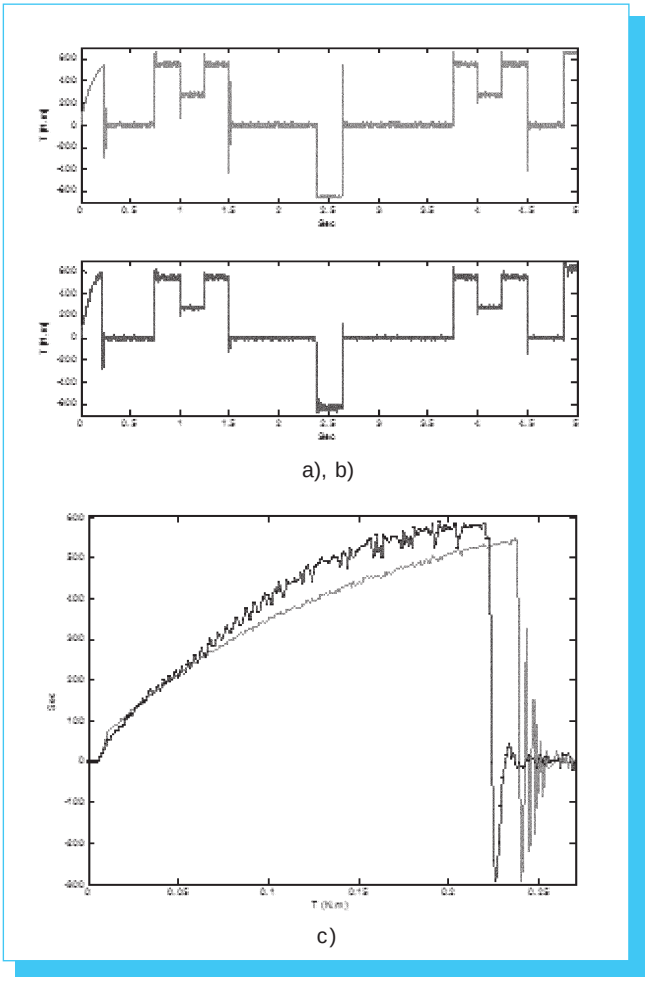

Fig. 11. Graphics of torque (PI control) and torque (neural control). a) Process history of the torque PI control; b) Process history of the torque neural control; c) IM start (PI control vs. neural control).

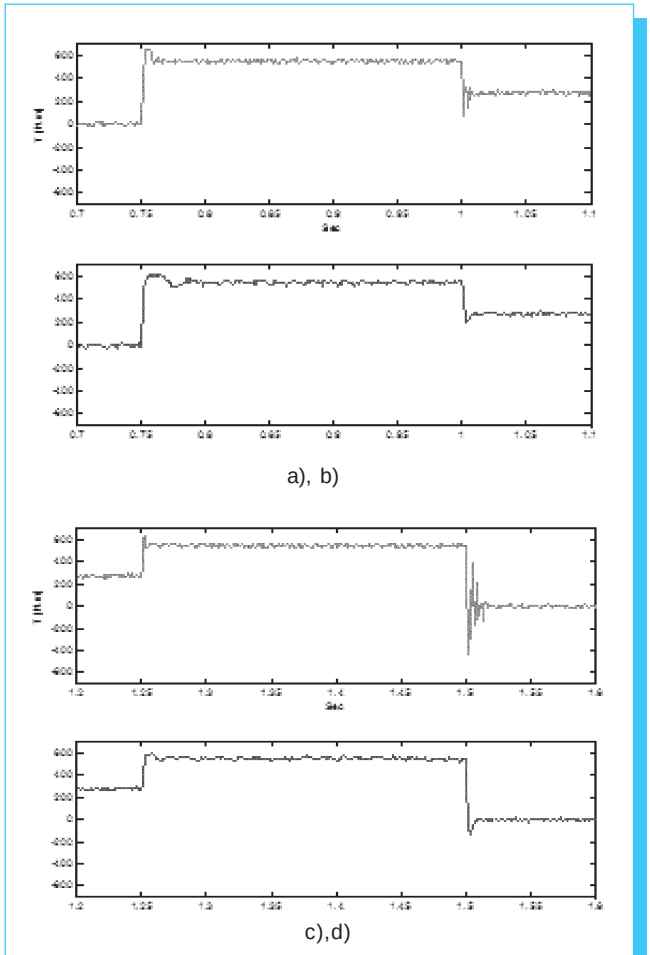

Fig. 12. Graphics of torque (PI control and neural control with load changes). a) PI control (0.7 -1.1 sec.); b) Neural control (0.7 -1.1 sec.); c) PI control (1.2 -1.1.6 sec.); d) Neural control (1.2 -1.6 sec.).

13 shows the (*a*,*b*,*c*) stator current set-points and the (*a*,*b*,*c*) stator currents of current hysteresis controlled system start.

The Figure 14 shows the same variables in the case of load changes. The results show a good performance of the neural control system at all, capable to augment and to decrease the currents in respective load changes.

7. Conclusions

The paper proposed a complete neural solution to the direct vector control of three phase induction motor including real-time trained neural controllers for velocity, flux and torque, which permitted the speed up reaction to the variable load. The basic equations and elements of the direct field oriented control scheme are given. The control scheme is realized by nine feedforward neural networks learned by Levenberg-Marquardt or BP algorithms with data taken by PI-control simulations. The NN PI or P adaptive neural controllers are learned on line using the BP algorithm. The complementary blocks which realized coordinate and computational operations are learned off-line using the LM algorithm with a 10-10 set up error precision. The graphical results of modelling shows a better performance of the adaptive NN control system with respect to the PI controlled system realizing the same computational control scheme with variable load.

Acknowledgements

We want to thank CONACYT-MEXICO for the scholarship given to Irving-Pavel de la Cruz Arguello - MS student at the CINVESTAV-IPN.

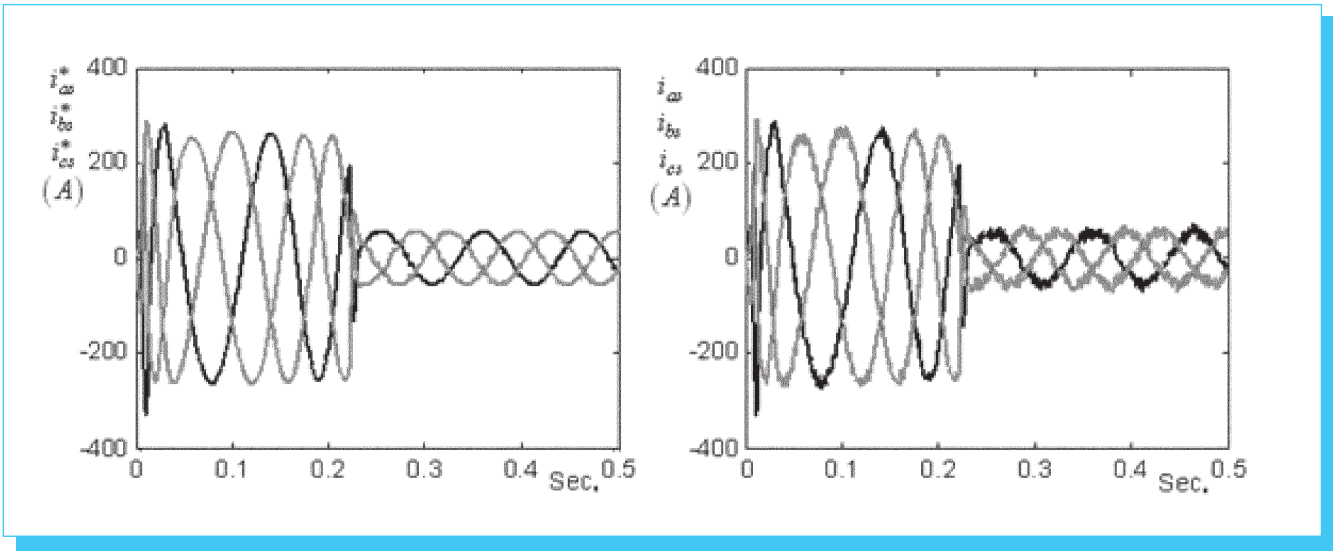

Fig. 13. Graphical results of (a,b,c) stator currents during the start of the IM. a) Current set points (left); b) Currents (right).

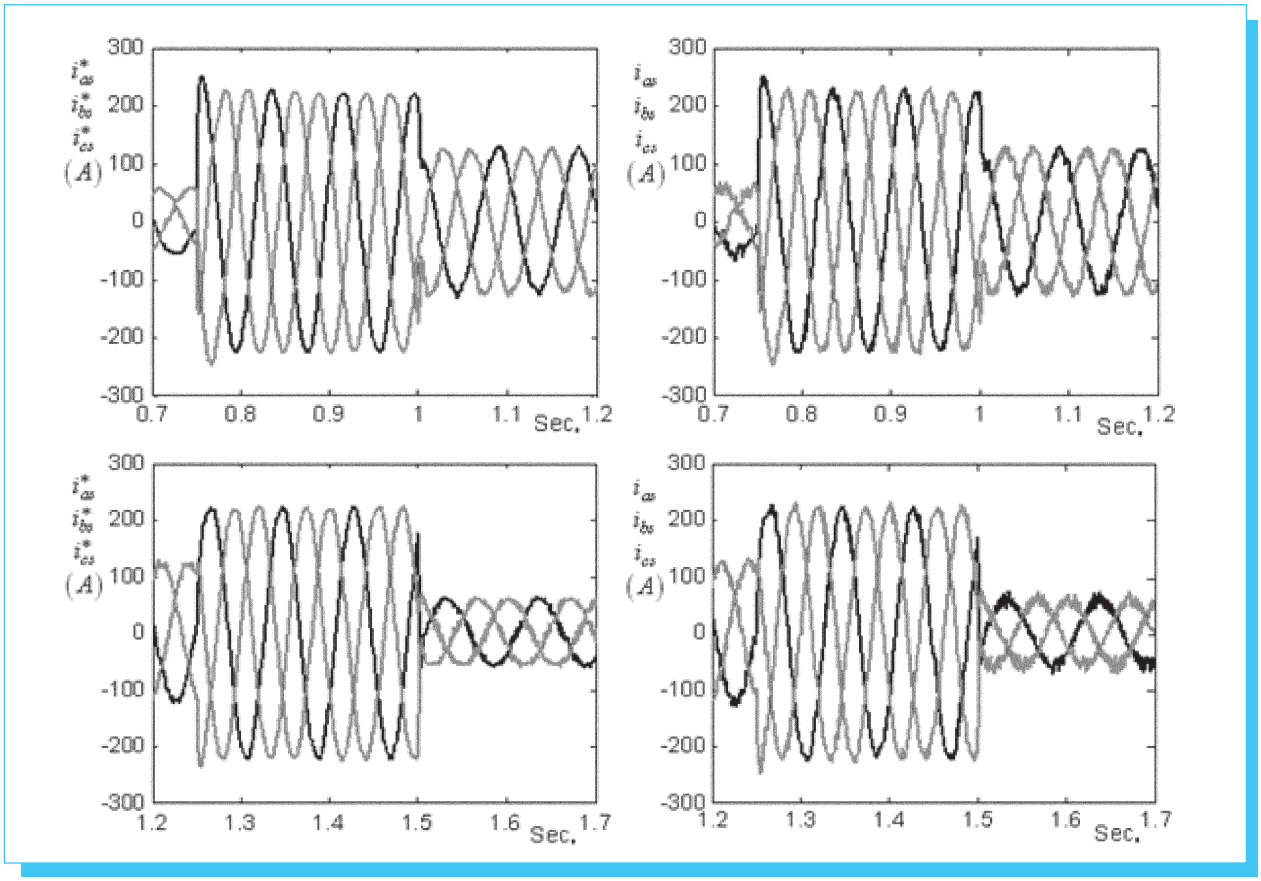

Fig. 14. Graphical results of (a,b,c) stator currents during load changes for different periods of time (0.7-1.2) sec., (1.2-1.7) sec. and different loads. a). c) Current set points (left); b) d) Currents (right).

8. References

[1] P. Melin, and O. Castillo. «Intelligent Control of a Stepping Motor Drive Using an Adaptive Neuro-Fuzzy Inference System». *Information Science,* Vol. 170, No 2-4, 2005, pp. 133-151.

[2] S. Weerasooriya, and M. A. El-Sharkawi. «Adaptive Tracking Control for High Performance DC Drives». *IEEE Trans. on Energy Conversion,* Vol.4, No 3, 1991, pp. 182-201.

[3] I. S. Baruch, J. M. Flores, F. Nava R., I. R. Ramirez P., and B. Nenkova. «An Advanced Neural Network Topology and Learning, Applied for Identification and Control of a D.C. Motor», *Proc. 1-st Int. IEEE Symposium on Intelligent Systems*, IS'02, Varna, Bulgaria, ISBN 0-7803-7601-3, Vol. 1, 2002, pp. 289-295.

[4] B. K. Bose, *Power Electronics and AC Drives.* Prentice-Hall, Englewood Cliffs, New Jersey, 1986, pp. 28-45, pp. 264-291.

[5] R. Ortega, A. Loría, P. Nicklasson, and H. Sira-Ramírez. *Passivity-Based Control of Euler-Lagrange Systems.* Springer-Verlag, London, Berlin, Heidelberg, 1998.

[6] Chee-Mun Ong. «*Dynamic Simulation of Electric Machinery*» . Prentice Hall, New York, 1998.

[7] D. W. Novotny, and T. A. Lipo. *Vector Control and Dynamics of AC Drives.* University Press, New York, Oxford, 1996.

[8] K.M. Woodley, H.Li, and S.Y. Foo. «Neural Network Modeling of Torque Estimation and d–q Transformation for Induction Machine». *Engineering Applications of Artificial Intelligence,* Vol. 18, No 1, 2005, pp. 57-63.

[9] J.O. Pinto, B.K. Bose, and L. E. Borges da Silva. «A Stator-Flux-Oriented Vector-Controlled Induction Motor Drive with Space-Vector PWM and Flux-Vector Synthesis by Neural Networks». *IEEE Transactions on Industry Applications,* Vol. 37, No 5, 2001, pp. 1308-1318.

[10] M. Cirrincione, M. Pucci, and G. Capolino. «A New Adaptive Integration Methodology for Estimating Flux in Induction Machine Drives». *IEEE Transactions on Power Electronics,* Vol.19, No 1, 2004, pp. 25 – 34.

[11] F.J. Lin, R.J. Wai, C.H. Lin, and D.C. Liu. «Decoupled Stator-Flux-Oriented Induction Motor Drive with Fuzzy Neural Network Uncertainty Observer». *IEEE Transactions on Industrial Electronics,* Vol. 47, No 2, 2000, pp. 356 - 367.

[12] W.L. Keerthipala, B.R. Duggal, and M.H. Chun. «Implementation of Field-Oriented Control of Induction Motors Using Neural Networks Observers». *Proc. IEEE International Conference on Neural Networks*, Vol. 3, 1996, pp. 1795 -1800.

[13] P. Marino, M. Milano, and F. Vasca. «Robust Neural Network Observer for Induction Motor Control». *Proc. 28-th Annual IEEE Power Electronics Specialists Conference*, Vol. 1, 1997, pp. 699 -705.

[14] M. Cirrincione, M. Pucci, and G. Capolino. «A New TLS -Based MRAS Speed Estimation UIT Adaptive Integration for High-Performance Induction Machine Drives». *IEEE Transactions on Industry Applications,* Vol. 40, No 4, 2004, pp. 1116 -1137.

[15] M. Cirrincione, M. Pucci, and G. Capolino. «An MRAS Based Sensorless High Performance Induction Motor Drive with a Predictive Adaptive Model». *IEEE Transactions on Industrial Electronics*, Vol. 52, No 2, 2005, pp. 532 -551.

[16] M.G. Simoes, and B. K. Bose. «Neural Network Based Estimation of Feedback Signals for a Vector Controlled Induction Motor Drive». *IEEE Transactions on Industry Applications,* Vol. 31, No 3, 1995, pp. 620- 629.

[17] A. Ba-Razzouk, A. Cheriti, G. Olivier, and P. Sicard. «Field-Oriented Control of Induction Motors Using Neural Network Decouplers». *IEEE Transactions on Power Electronics,* Vol. 12, No 4, 1997, pp. 752 – 763.

[18] M.T. Hagan, and M. B. Menhaj. «Training Feedforward Networks with the Marquardt Algorithm». *IEEE Transaction on Neural Networks, Vol. 5, No* 6, 1994, pp. 989-993.

[19] Howard Demuth, and Mark Beale, «*Neural Network Toolbox User's Guide, Version 4"*. The Math Works, Inc., New York, COPYRIGHT 1992 -2002.