Direct Field Oriented Neural Control of a Three Phase Induction Motor

Ieroham S. Baruch[1], Irving P. de la Cruz[1], and Boyka Nenkova[2]

[1] CINVESTAV-IPN, Department of Automatic Control, Av. IPN No 2508,
Col. Zacatenco, A.P. 14-740, 07360 Mexico D.F., Mexico
{baruch,idelacruz}@ctrl.cinvestav.mx
[2] IIT-BAS, 1113 Sofia, Bulagaria
nenkova@riskeng.bg

Abstract. 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 and recurrent neural networks learned by Levenberg-Marquardt or real-time BP algorithms with data taken by PI-control simulations. The graphical results of modelling shows a better performance of the NN control system with respect to the PI controlled system realizing the same general control scheme.

Keywords: Angular velocity control, direct field oriented neural control, recurrent and feedforward neural networks, backpropagation and Levenberg-Marquardt learning, three phase induction motor.

1 Introduction

The application of Neural Networks (NN) for identification and control of electrical drives became very popular in last decade. In [1], [2], a multilayer feedforward and a recurrent neural networks are applied for a DC motor drive high performance control. In the last decade a great boost is made in the area of Induction Motor (IM) drive control. The induction machine of cage type is most commonly used in adjustable speed AC drive systems, [3]. The control of AC machines is considerably more complex than that of DC machines. The complexity arises because of the variable-frequency power supply, the AC signals processing, and the complex dynamics of the AC machine, [3], [4]. 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, [3]-[7]. There exist two methods for PWM inverter current control – direct and indirect vector control, [3]. 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 [8] a feedforward NN is used for vector PW modulation,

D. Dicheva and D. Dochev (Eds.): AIMSA 2010, LNAI 6304, pp. 241–250, 2010.

resulting in a faster response. In [9] an ADALINE NN is used for cancellation of the integration DC component during the flux estimation. In [10] a fuzzy-neural uncertainty observer is integrated in a FOC system, using an estimation of the rotor time constant. In [11] an Artificial NN is used for fast estimation of the angle ρ used in a FOC system. In [12] a flux and torque robust NN observer is implemented in a FOC system. In [13], an ADALINE-based-filter and angular-velocity-observer are used in a FOC system. In [14], a NN velocity observer is used in FOC high performance system for an IM drive. In [15] a Feedforward-NN (FFNN)-based estimator of the feedback signals is used for IM drive FOC system. The paper [16] proposed two NN-based methods for FOC of IM. 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 showed that the application of NN offers a fast and improved alternative of the classical FOC schemes, [17]. The present paper proposed a neural solution of a direct FOC. The system achieved adaptation to a variable load applying real-time learned neural controllers of IM velocity, flux, and torque. In our early work, [17], the phase (a,b,c), the (q,d,0) model, and the coordinate transformation between them has been completely described, so here we may skip those parts.

2 Field Orientation Conditions and Flux Estimation

The flux and torque decoupling needs to transform the stator flux, current and voltage vectors from (a, b, c) reference frame into (q-d,s) reference frame and than to stationary and synchronous reference frames, [17]. In the next equations, the following notation is used: v –voltage, i-current, λ-flux, r-resistance, L-inductance, ω-velocity; the sub-indices are r- rotor, s-stator, q, d- components of the (q, d, 0) model; the upper index s means stator reference frame and e means synchronous reference frame; the prime means relative rotor to stator value. The Fig. 1a illustrates the current and voltage vector representations in stator and rotor synchronous frames and also the magnetic FO, 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. 1b. So, the FO conditions are:

$$\lambda'^{e}_{qr} = 0;\ p\lambda'^{e}_{qr} = 0;\ \lambda_r = \lambda'^{e}_{dr} \tag{1}$$

Taking into account that the rotor windings are shortcut, (the rotor voltage is zero), also the given up field orientation conditions, and the (q, d, 0) model, [17], we could write:

$$0 = r'_r i'^{e}_{qr} + (\omega_e - \omega_r)\lambda'^{e}_{dr};\ 0 = r'_r i'^{e}_{dr} + p\lambda'^{e}_{dr} \tag{2}$$

Using the (q, d, 0) model, [17], for the q-component of the rotor flux, it is obtained:

$$\lambda'^{e}_{qr} = L_m i^{e}_{qs} + L'_r i'^{e}_{qr} = 0;\ L'_r = L'_{lr} + L_m;\ i'^{e}_{qr} = -(L_m / L'_r) i^{e}_{qs};\ \tag{3}$$

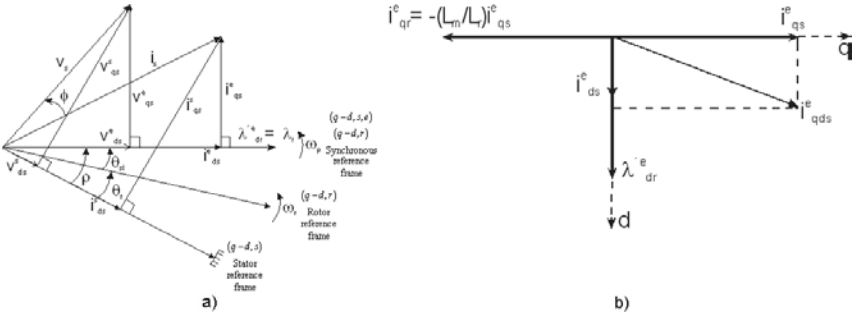

Fig. 1. Vector diagrams of the stator current, voltage and the rotor flux. a) The current and voltage vector representations in stator and in rotor synchronous reference frames. b) The stator current and the rotor flux vector representations in synchronous reference frame.

Using (1) and (3), the torque equation, [17], could obtain the form:

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r'}\lambda_{dr}'^{e} i_{qs}^{e} \tag{4}$$

The equation (4) shows that if the flux of the rotor is maintained constant, so the torque could be controlled by the q-component of the stator current in synchronous reference frame. From the second equation of (2), taking into account (3) it is easy to obtain the slipping angular velocity as:

$$\omega_e - \omega_r = (r_r' L_m / L_r')(i_{qs}^{e} / \lambda_{dr}'^{e}) \tag{5}$$

The final equations (3), (4), (5) gives us the necessary basis for a direct decoupled field oriented (vector) control of the AC motor drive, where (see Fig. 1b) the q- component of the stator current produced torque and its d-component produced flux. Following [17], we could write:

$$\lambda_{qs}^{s} = (1/p)\left(v_{qs}^{s} - r_s i_{qs}^{s}\right); \lambda_{ds}^{s} = (1/p)\left(v_{ds}^{s} - r_s i_{ds}^{s}\right) \tag{6}$$

$$i_{qr}'^{s} = (\lambda_{qs}^{s} - L_s i_{qs}^{s})/L_m; i_{dr}'^{s} = (\lambda_{ds}^{s} - L_s i_{ds}^{s})/L_m \tag{7}$$

$$\lambda_{qr}'^{s} = (L_r'/L_m)\left(\lambda_{qs}^{s} - L_s' i_{qs}^{s}\right); \lambda_{dr}'^{s} = (L_r'/L_m)\left(\lambda_{ds}^{s} - L_s' i_{ds}^{s}\right); L_s' = [L_s - (L_m^2/L_r')] \tag{8}$$

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, which are:

$$\lambda_r' = \sqrt{(\lambda_{qr}'^{s})^2 + (\lambda_{dr}'^{s})^2}; sin\rho = \lambda_{qr}'^{s}/\lambda_r'; \cos\rho = \lambda_{dr}'^{s}/\lambda_r' \tag{9}$$

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r'}\lambda_r' i_{qs}^{e} \tag{10}$$

3 General Control Scheme and NN Realization of the IM Control

A general block diagram of the direct vector control of the Induction Motor drive is given on Fig. 2a. The direct control scheme contains three principal blocks. They are: G1, G2, G3 – blocks of PI controllers; block of coordinate (abc) to (q-d,s,e) transformation, [17]; block of vector estimation, performing the field orientation and the torque, flux and angle computations (see equations (9), (10)); 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 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. The Fig. 2b illustrates the flux and angle estimation for field orientation, computing (6), (8), (9). The rotor flux computations block (see Fig. 2b) performs computations given by (6), (8), illustrated by the Fig. 2c. The rotor flux, the angle, and the *sin, cos* -functions computations are given by equation (9). The torque estimation is computed by equation (10).

3.1 Neural Network Realization of the Control Scheme

The simplified block-diagram of the direct neural vector control system, given on Fig. 2a 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.

The FFNN1. The first NN1 is an angular velocity neural PI controller with two inputs (the velocity error, and the total sum of velocity errors) and one output (the torque set point). The weights learning is done in real – time using the Backpropagation (BP) algorithm. The FFNN1 function is given by the following equation:

$$T^*(k) = \varphi[g_p(k)e_{vel} + g_i(k)e_{vel}^{sum}(k)] \tag{11}$$

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:

$$e_{vel}^{sum}(k) = \sum_{k=0}^{n} e_{vel}(k) \tag{12}$$

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

$$\begin{aligned} g_p(k+1) &= g_p(k) + \eta e_{vel}(k)[1-(T^*(k))^2]e_{vel}(k) \\ g_i(k+1) &= g_i(k) + \eta e_{vel}(k)[1-(T^*(k))^2]e_{vel}^{sum}(k) \end{aligned} \tag{13}$$

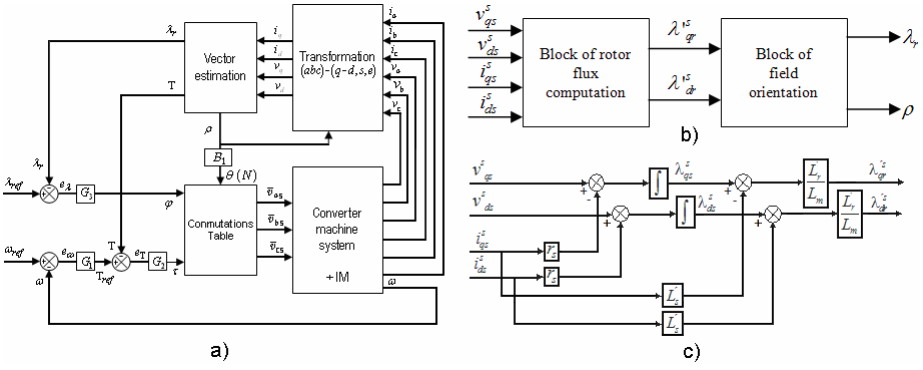

Fig. 2. Block diagrams. a) General BD of a direct IM vector control. b) BD of the vector estimation computations. c) BD of the flux estimation computations.

The FFNN2. The second FFNN2 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 of this FFNN2 are given by:

$$i_{qs}^{e*}(k) = \phi[g_p(k)e_T(k)] \tag{14}$$

$$g_p(k+1) = g_p(k) + \eta e_T(k)[1-(i_{qs}^{e*}(k))^2]e_T(k) \tag{15}$$

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.

The FFNN3. The third FFNN3 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 of this NN3 are given by:

$$i_{ds}^{e*}(k) = \varphi[g_p(k)e_{flux} + g_i(k)e_{flux}^{sum}(k)] \tag{16}$$

$$\begin{aligned} g_p(k+1) &= g_p(k) + \eta e_{flux}(k)[1-(i_{ds}^{e*}(k))^2]e_{flux}(k) \\ g_i(k+1) &= g_i(k) + \eta e_{flux}(k)[1-(i_{ds}^{e*}(k))^2]e_{flux}^{sum}(k) \end{aligned} \tag{17}$$

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 during n iterations is:

$$e_{flux}^{sum}(k) = \sum_{k=0}^{n} e_{flux}(k) \tag{18}$$

The FFNN4. The fourth FFNN4 is a torque off-line trained neural estimator (realizing (10) equation computation) which has two inputs and one output (the rotor flux, the stator q-current, and the estimated torque). The topology of this FFNN4 is (2-10-1).

The FFNN5. The fifth FFNN5 performed a stator current (a,b,c) to (q-d,s,e) transformation, [17]. The FFNN5 topology has five inputs (three i_{as}, i_{bs}, i_{cs} –stator currents; $sin\rho$, $cos\rho$), 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 FFNN6. The sixth FFNN6 performed an inverse stator current (q-d,s,e) to (a,b,c) transformation using the transpose of the transformation matrix, [17]. The FFNN6 topology is (4-30-10-3) (four inputs -two i_{qs}^{e}, i_{ds}^{e} –stator currents; $sin\rho$, $cos\rho$; three outputs- i_{as}, i_{bs}, i_{cs} – stator currents; two hidden layers of 30 and 10 neurons).

The FFNN7. The seventh FFNN7 performed rotor flux estimation using equation (9). The rotor (q-d,r) flux components $\lambda_{qs}^{s}, \lambda_{ds}^{s}$ are previously computed using equation (6) (see Fig. 2c), 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: $\lambda_{r}^{'}$. The FFNN7 topology is (4-30-10-1).

The FFNN8 and FFNN9. The FFNN8, and FFNN9 are similar to FFNN7 and performed separately the q and d rotor flux components estimation using equations (8). The FFNN8, FFNN9 topologies are: (2-10-5-2). The values of $sin\rho$, $cos\rho$, (9), needed for the coordinate transformations are obtained dividing the outputs of FFNN8, FFNN9 by the output of FFNN7.

All the FFNN4-FFNN9 are learned by 2500 input-output patterns (half period) and generalized by another 2500 ones (the other half period). The FFNN4-FFNN9 learning is off-line, applying the Levenberg-Marquardt algorithm [18], [19] during 61, 29, 32, 35, 47 and 49 epochs of learning, respectively. The final value of the MSE reached during the learning is of 10^{-10} for all the FFNN4-FFNN9.

4 Graphical Results of the Control System Modeling

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 Fig. 3a, b showed the angular velocity set point vs. the IM angular velocity in the general case of velocity control and particularly with load torque changes (PI and NN control). The results show that the angular velocity control system has a fast speed up response and satisfactory behaviour in the case of load change. The Fig. 4 a, b showed the flux graphics of control system with hysteresis control applying the PI control scheme and NNs. The results show a faster and better

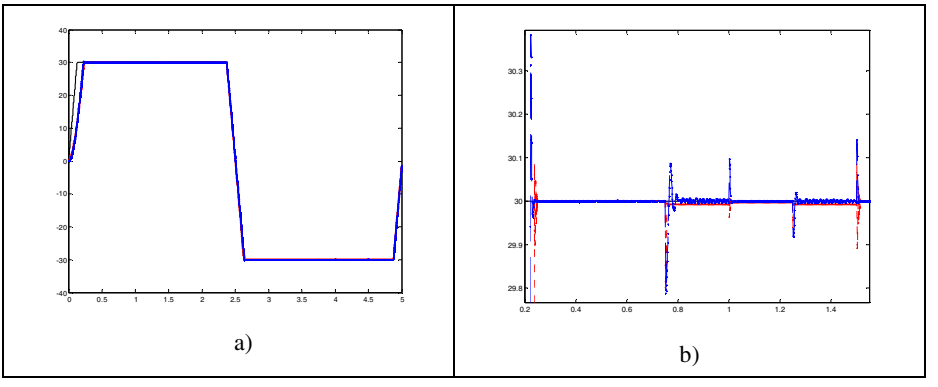

Fig. 3. Graphical results of the IM velocity control. a) General graphics of the angular velocity control; b) Graphical results of angular velocity control with load changes.

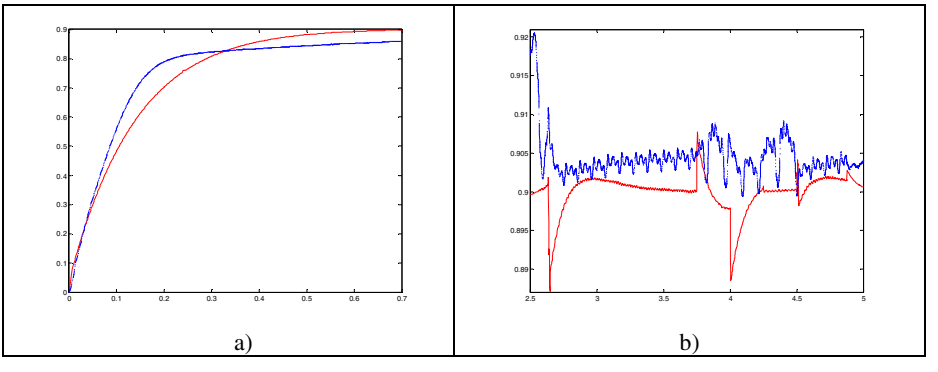

Fig. 4. Graphical results of the IM flux control. a) Graphics of the flux classical control vs. flux neural control; b) Graphics of both (classical vs. NN) flux control with load changes.

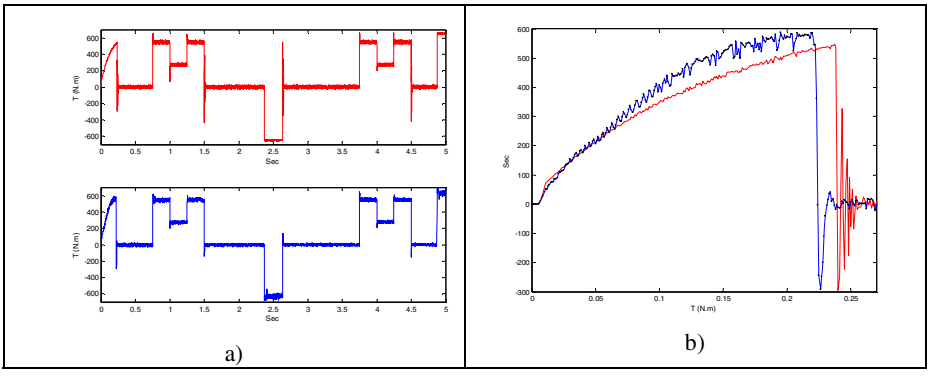

Fig. 5. Graphics of the torque control with load changes. a) Graphics of the torque classical and neural control; b) Graphics of both torque control (PI control vs. neural control) in the IM start.

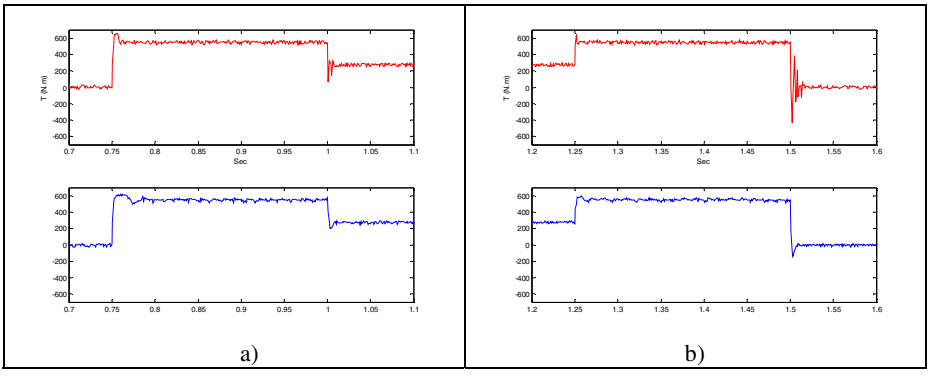

Fig. 6. Detailed graphics of the torque control using both control schemes and load changes. a) Processes from 0.7 sec. to 1.1 sec; b) Processes from 1.2 sec. to 1.6 sec.

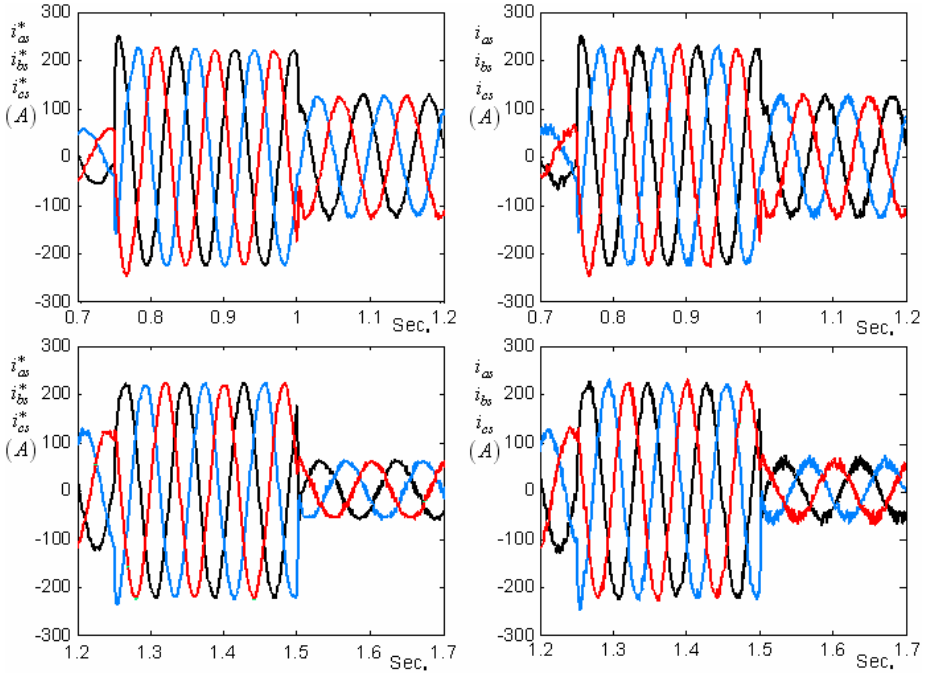

Fig. 7. Graphical results of (a,b,c) stator currents during neural control and load changes for periods of time of (0.7-1.2 sec.) and (1.2-1.7 sec.). a), c) Current set points; b), d) Currents.

response of the neural system which tried to maintain the flux constant in the case of load changes. The Fig. 5 a, b; Fig. 6 a, b; Fig. 7 a, b, c, d show the torque and current graphics with hysteresis control in the same cases and load changes.

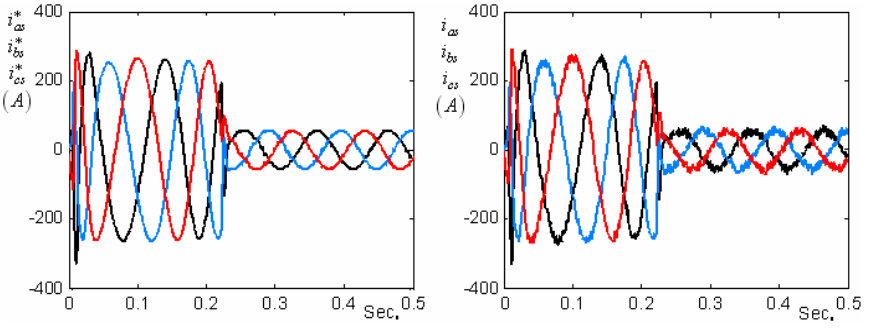

Fig. 8. Graphical Results of (a,b,c) Stator Currents During the Start of the IM. a) Current Set Points; b) Currents.

The Fig. 7 a, b, c, d shows the (a,b,c) stator currents set points and the stator currents of hysteresis controlled system using neural control schemes in load changes conditions for different time intervals. The Fig. 8 a, b shows the same stator current set points and currents during the start of the IM. The results show a good performance of the neural control system at all.

5 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 FOC scheme are given. The control scheme is realized by nine feedforward neural networks learned 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 Levenberg-Marquardt 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. The MS student Irving-Pavel de la Cruz A. is thankful to CONACYT, Mexico for the scholarship received during his studies in the Department of Automatic Control, CINVESTAV-IPN, Mexico City, Mexico.

References

1. Weerasooriya, S., El-Sharkawi, M.A.: Adaptive Tracking Control for High Performance DC Drives. IEEE Trans. on Energy Conversion 4, 182–201 (1991)
2. Baruch, I.S., Flores, J.M., Nava, F., Ramirez, I.R., Nenkova, B.: An Advanced Neural Network Topology and Learning. In: Sgurev, V., Jotsov, V. (eds.) Proc. of the 1-st Int. IEEE Symposium on Intelligent Systems, Applied for Identification and Control of a D.C. Motor, Varna, Bulgaria, pp. 289–295. IEEE, Los Alamitos (2002), ISBN 0-7803-7601-3

3. Bose, B.K.: Power Electronics and AC Drives, pp. 264–291. Prentice-Hall, Englewood Cliffs (1986)
4. Ortega, R., Loría, A., Nicklasson, P., Sira - Ramírez, H.: Passivity – Based Control of Euler – Lagrange Systems. Springer, Heidelberg (1998)
5. Ong, C.M.: Dynamic Simulation of Electric Machinery. Prentice Hall, New York (1998)
6. Novotny, D.W., Lipo, T.A.: Vector Control and Dynamics of AC Drives. Oxford University Press, New York (1996)
7. Woodley, K.M., Li, H., Foo, S.Y.: Neural Network Modeling of Torque Estimation and d-q Transformation for Induction Machine. Engng. Appl. of Artif. Intell. 18(1), 57–63 (2005)
8. Pinto, J.O., Bose, B.K., Da Silva, L., Borges, E.: A Stator – Flux – Oriented Vector – Controlled Induction Motor Drive With Space – Vector PWM and Flux – Vector Synthesis by Neural Networks. IEEE Trans. on Industry Applications 37, 1308–1318 (2001)
9. Cirrincione, M., Pucci, M., Capolino, G.: A New Adaptive Integration Methodology for Estimating Flux in Induction Machine Drives. IEEE Trans. on PE 19(1), 25–34 (2004)
10. Lin, F.J., Wai, R.J., Lin, C.H., Liu, D.C.: Decoupled Stator-Flux-Oriented Induction Motor Drive with Fuzzy Neural Network Uncertainty Observer. IEEE Trans. on Industrial Electronics 47(2), 356–367 (2000)
11. Keerthipala, W.L., Duggal, B.R., Chun, M.H.: Implementation of Field – Oriented Control of Induction Motors using Neural Networks Observers. In: Proc. IEEE International Conference on Neural Networks, vol. 3, pp. 1795–1800 (1996)
12. Marino, P., Milano, M., Vasca, F.: Robust Neural Network Observer for Induction Motor Control. In: Proc. 28-th Annual IEEE PE Specialists Conference, vol. 1, pp. 699–705 (1997)
13. Cirrincione, M., Pucci, M., Capolino, G.: A New TLS Based MRAS Speed Estimation UIT Adaptive Integration for High-Performance Induction Machine Drive. IEEE Trans. on Industry Applications 40(4), 1116–1137 (2004)
14. Cirrincione, M., Pucci, M., Capolino, G.: An MRAS Based Sensorless High Performance Induction Motor Drive with a Predictive Adaptive Model. IEEE Trans. on Industrial Electronics 52(2), 532–551 (2005)
15. Simoes, M.G., Bose, B.K.: Neural Network Based Estimation of Feedback Signals for a Vector Controlled Induction Motor Drive. IEEE Trans. on Ind. Appl. 31, 620–629 (1995)
16. Ba – Razzouk, A., Cheriti, A., Olivier, G., Sicard, P.: Field - Oriented Control of Induction Motors Using Neural Network Decouplers. IEEE Trans. on P. Elec. 12, 752–763 (1997)
17. Baruch, I.S., Mariaca-Gaspar, C.R., De la Cruz, I.P.: A Direct Torque Vector Neural Control of a Three Phase Induction Motor. In: Sossa-Azuela, J.H., Baron-Fernandez, R. (eds.) Research in Computer Science, Special Issue on Neural Networks and Associative Memories, vol. 21, pp. 131–140 (2006), ISSN 1870-4069
18. Hagan, M.T., Menhaj, M.B.: Training Feedforward Networks with the Marquardt Algorithm. IEEE Trans. on Neural Networks 5, 989–993 (1994)
19. Demuth, H., Beale, M.: Neural Network Toolbox User's Guide, version 4, The Math. Works, Inc. COPYRIGHT (1992-2002)