Model Reference Neural Predictive Controller for Induction Motor Drive

M. OUHROUCHE, A. MERABET and R.T. BUI
Electrical Machines Identification and Control Laboratory (EMICLab)
Department of Applied Sciences
University of Quebec at Chicoutimi
555 Blvd de l'Université, Chicoutimi (Qc), G7H 2B1
CANADA

Abstract: - In this paper an accurate nonlinear model of induction motor using an artificial neural network (ANN) is given. This modeling technique is done by using the data from the system inputs/outputs information without requiring the knowledge about machine parameters. The ANN training is carried out off-line using the Levenberg-Marquardt algorithm. Then, the proposed neural network model is used as predictor for predictive control with reference model to track speed and flux profiles, where the cost function is minimized by Newton-Raphson method. Results of simulation show that the proposed model is accurate under both transient and steady state conditions.

Key-Words: - Input-Output Modeling, ANN, Predictive Control, Reference Control Model, Induction Motor.

1 Introduction

The induction machine is the workhorse of industry. It is more rugged, reliable, compact, efficient and cheap in comparison to other motors used in similar applications. With the advancement of power electronics and DSP technology advanced control techniques for induction motor which were once thought to be impractical can now be implemented in real-time. This is particularly true in the aluminum industry for which part of this work will be applied, where a host of processes require efficient control of speed and high performance torque response. The problem is often compounded by the fact that the varying machine parameters are difficult to determine.

The induction motors (IM) constitute a theoretically interesting and practically important class of nonlinear systems. The control task is further complicated by the fact that induction motors are subject to unknown (load) disturbances and change in values of parameters during its operation. The control engineering community is faced then with the challenging problem of controlling a highly nonlinear system, with varying parameters, where the regulated outputs, besides some of them being not measurable, are perturbed by an unknown disturbance signal [1-3].

In recent years, the use of artificial neural networks for nonlinear system identification and control has proved to be extremely successful because of their adaptability to a changing environment, which allows them to emulate time varying behavior of nonlinear plant, and their robustness with respect to noise. A few studies have been reported on the induction motor modeling using neural networks for obtaining speed or rotor flux model [4-6] but not for both.

Among the applications of the neural network model in control is neural predictive control (NPC). The predictive control method is traditionally used for industrial process control and a large number of implementation algorithms have been presented in literature such as extended prediction self adaptive control, generalized predictive control and unified predictive control [7]. Most of these control algorithms use an explicit process model to predict the future behavior of the plant and because of this, the term Model based predictive control (MBPC) is often used. The classical MBPC algorithms use linear models for prediction, but when the process is nonlinear, the use of linear model becomes impractical, and identification of nonlinear model for control becomes a necessity.

In this paper, we present an input-output modeling for the induction motor using an artificial neural network for obtaining a model with flux and speed as outputs and its application in predictive control to track speed and flux profiles. The paper is organized as follows: in section 2, we give the mathematical input-output model for the induction motor by differentiating the outputs with Lie derivative [2, 3] and expressing all states and inputs in terms of these outputs [1, 8]. In section 3, the system modeling by neural networks is given. In section 4, the application of the ANN model as predictor in predictive control is detailed. Finally, the simulation results using the neural networks model are presented.

2 Induction Motor Mathematical Model

Under the assumption of linearity of the magnetic circuit, the well-known (α-β) induction motor model established in the stationary reference frame is given by the following fifth-order continuous time model

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}\mathbf{u} \tag{1}$$

where

$$\mathbf{x} = \begin{bmatrix} i_{s\alpha} & i_{s\beta} & \phi_{r\alpha} & \phi_{r\beta} & \omega \end{bmatrix}^T, \ \mathbf{u} = \begin{bmatrix} u_{s\alpha} & u_{s\beta} \end{bmatrix}^T$$

The state $\mathbf{x}$ belongs to the set

$$\mathbf{\Omega} = \left\{ \mathbf{x} \in \Re^5 : \phi_{r\alpha}^2 + \phi_{r\beta}^2 \neq 0 \right\}$$

Vector function $\mathbf{f}(\mathbf{x})$ and constant matrix $\mathbf{g}$ are defined as follows

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} -\gamma i_{s\alpha} + \dfrac{K}{T_r}\phi_{r\alpha} + pK\omega\phi_{r\beta} \\ -\gamma i_{s\beta} + \dfrac{K}{T_r}\phi_{r\beta} - pK\omega\phi_{r\alpha} \\ \dfrac{M}{T_r} i_{s\alpha} - \dfrac{1}{T_r}\phi_{r\alpha} - p\omega\phi_{r\beta} \\ \dfrac{M}{T_r} i_{s\beta} - \dfrac{1}{T_r}\phi_{r\beta} + p\omega\phi_{r\alpha} \\ \dfrac{pM}{JL_r}\left(\phi_{r\alpha} i_{s\beta} - \phi_{r\beta} i_{s\alpha}\right) - \dfrac{1}{J}\left(f\omega + T_L\right) \end{bmatrix}$$

$$\mathbf{g} = \begin{bmatrix} g_1 & g_2 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\sigma L_s} & 0 & 0 & 0 & 0 \\ 0 & \dfrac{1}{\sigma L_s} & 0 & 0 & 0 \end{bmatrix}^T$$

$$K = \frac{M}{\sigma L_s L_r}; \sigma = 1 - \frac{M^2}{L_s L_r}; \gamma = \frac{R_s}{\sigma L_s} + \frac{R_r M^2}{\sigma L_s L_r^2}$$

$i_{s\alpha}$, $i_{s\beta}$ denote the stator currents, $\phi_{r\alpha}, \phi_{r\beta}$ the rotor fluxes, ω the rotor speed and $u_{s\alpha}$, $u_{s\beta}$ the stator voltages.

The information about the speed and the rotor flux norm is known by estimation; we choose the outputs of the system as

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \text{with} \begin{cases} y_1 = h_1(\mathbf{x}) = \omega \\ y_2 = h_2(\mathbf{x}) = \sqrt{\phi_{r\alpha}^2 + \phi_{r\beta}^2} = \phi \end{cases} \tag{2}$$

The functions $\mathbf{f}(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are sufficiently differentiable.

Input-output modeling formulation is carried out by using Lie derivative until we get relations between the outputs and the inputs. In order to alleviate the burden of input-output model computation with Lie derivative, the second output is transformed as

$$\tilde{y}_2 = \phi^2 = y_2^2 \tag{3}$$

The following notation is used for the Lie derivative of state function $h(x) : \Re^n \rightarrow \Re$ along a vector field $f(x) = (f_1(x), \ldots, f_n(x))$

$$L_f h = \sum_{i=1}^{n} \frac{\partial h}{\partial x_i} f_i(x) \tag{4}$$

Iteratively, we have $L_f^i h = L_f(L_f^{(i-1)} h)$

For the proposed model, we get the input-output relations by the second derivative.

$$\begin{aligned} \ddot{y}_1 &= L_f^2 h_1 + (L_{g_1} L_f h_1) u_{s\alpha} + (L_{g_2} L_f h_1) u_{s\beta} \\ &= \frac{pM}{JL_r}\left(\gamma + 1 + \frac{f}{J}\right)\left(i_{s\alpha}\phi_{r\beta} - i_{s\beta}\phi_{r\alpha}\right) \\ &- \frac{p^2 MK}{JL_r}\omega\left(\phi_{s\alpha}^2 + \phi_{s\beta}^2\right) - \frac{p^2 M}{JL_r}\omega\left(i_{s\alpha}\phi_{r\alpha} + i_{s\beta}\phi_{r\beta}\right) + \frac{f}{J^2}\left(f\omega + T_L\right) \\ &- \frac{pM}{\sigma L_s L_r J}\phi_{r\beta} u_{s\alpha} + \frac{pM}{\sigma L_s L_r J}\phi_{r\alpha} u_{s\beta} \\ \ddot{\tilde{y}}_2 &= L_f^2 h_2 + (L_{g_1} L_f h_2) u_{s\alpha} + (L_{g_2} L_f h_2) u_{s\beta} \\ &= -\frac{2M}{T_r}\left(\gamma + \frac{3}{T_r}\right)\left(i_{s\alpha}\phi_{r\alpha} + i_{s\beta}\phi_{r\beta}\right) - \frac{2pM}{T_r}\omega\left(i_{s\alpha}\phi_{r\beta} - i_{s\beta}\phi_{r\alpha}\right) \\ &+ \frac{2M^2}{T_r^2}\left(i_{s\alpha}^2 + i_{s\beta}^2\right) + \frac{(4 + 2MK)}{T_r^2}\left(\phi_{s\alpha}^2 + \phi_{s\beta}^2\right) + \frac{2M}{\sigma L_s T_r}\phi_{r\alpha} u_{s\alpha} + \frac{2M}{\sigma L_s T_r}\phi_{r\beta} u_{s\beta} \end{aligned} \tag{5}$$

To get the outputs (2), we use this transformation

$$\ddot{y}_2 = \frac{1}{\sqrt{\tilde{y}_2}}\left(\frac{1}{2}\ddot{\tilde{y}}_2 - \frac{1}{4\tilde{y}_2}\dot{\tilde{y}}_2^2\right) \tag{6}$$

In order to have the relation between the inputs and outputs from (5) and (6), we need to express all the machine states in terms of outputs using the motor equations. For sake of simplicity we prefer to work in complex form.

The electro-mechanical equation is given by

$$J\frac{d\omega}{dt} = p\,\mathrm{Im}(\phi_r i_r^*) - f\omega - T_L \tag{7}$$

where Im (.): Imaginary part and $\phi_r = \phi e^{j\delta}$

The rotor electrical equation is given by

$$\frac{d\phi_r}{dt} - jp\,\omega\phi_r + R_r i_r = 0 \tag{8}$$

The torque produced by the motor is given by

$$T_{em} = p\,\mathrm{Im}\left(\phi_r i_r^*\right) = \frac{p}{R_r}\phi^2\left(\dot{\delta} - p\,\omega\right) \tag{9}$$

set $\alpha = \delta - p\theta$

Substituting T_{em} from (9) in (7), the electro-mechanical equation becomes

$$J\frac{d\omega}{dt} = \frac{p}{R_r}\phi^2\dot{\alpha} - f\omega - T_L \tag{10}$$

then, with assuming that the disturbance $T_L = 0$

$$\alpha = \frac{R_r}{p}\int_0^t\left(\frac{J\dot{\omega}}{\phi^2} + \frac{f\omega}{\phi^2}\right)d\tau \tag{11}$$

and with (2)

$$\alpha = \frac{R_r}{p}\int_0^t\left(\frac{J\dot{y}_1}{y_2^2} + \frac{fy_1}{y_2^2}\right)d\tau \tag{12}$$

so, for the rotor flux, we have

$$\phi_r = y_2\exp\left(j\frac{R_r}{p}\int_0^t\left(\frac{J\dot{y}_1}{y_2^2} + \frac{fy_1}{y_2^2} + \frac{p^2}{R_r}y_1\right)d\tau\right) \tag{13}$$

Using (8) and (13), the rotor current can be expressed as

$$\begin{aligned} i_r &= -\frac{1}{R_r}\left(\frac{d\phi_r}{dt} - jp\omega\phi_r\right) \\ &= -\frac{1}{R_r}\left[\dot{y}_2 + j\frac{R_r}{p}\left(\frac{J\,\dot{y}_1}{y_2} + \frac{fy_1}{y_2}\right)\right]e^{j(\alpha+p\theta)} \end{aligned} \tag{14}$$

Using (13), (14) and the following rotor flux equation:

$$\phi_r = Mi_s + L_r i_r \tag{15}$$

we obtain the following stator current:

$$i_s = \frac{1}{M}\left[y_2 + T_r A\right]e^{j(\alpha+p\theta)} \tag{16}$$

with

$$A = \dot{y}_2 + j\frac{R_r}{p}\left(\frac{J\dot{y}_1}{y_2} + \frac{fy_1}{y_2}\right) = \dot{y}_2 + j\dot{\alpha}\,y_2 \tag{17}$$

From (14), (16) and the following rotor flux

$$\phi_s = L_s i_s + Mi_r \tag{18}$$

The stator flux is given by

$$\phi_s = \left(\frac{L_s}{M}y_2 + \frac{L_s L_r - M^2}{MR_r}A\right)e^{j(\alpha+p\theta)} \tag{19}$$

Finally, the stator voltage can be expressed as

$$\begin{aligned} u_s &= u_{s\alpha} + ju_{s\beta} = R_s i_s + \frac{d\phi_s}{dt} \\ &= \frac{R_s}{M}\left(y_2 + T_r A\right)e^{j(\alpha+p\theta)} \\ &\quad + \frac{d}{dt}\left\{\left(\frac{L_s}{M}y_2 + \frac{L_s L_r - M^2}{MR_r}A\right)e^{j(\alpha+p\theta)}\right\} \end{aligned} \tag{20}$$

The mathematical input-output model (5), with all states expressed in terms of the outputs, is difficult to solve and to implement in advanced control strategies like nonlinear predictive control. Therefore, we try to approximate it to a nonlinear model less complicated using ANN as explained in the next section.

3 Neural Network Modeling

The goal of modeling a nonlinear system by neural networks is to build a mathematical model which emulates its dynamic behavior, given some prior knowledge about the system and information on inputs and outputs. For an induction motor, it is possible to estimate the various machine quantities in both the steady state and transient state by using an ANN. The ANN can accurately describe the nonlinear behavior of the machine without requiring the knowledge of machine parameters [4-6].

The objective is to use a NNARX (Neural Network AutoRegressive eXternal input) $\hat{g}$[.] to approximate the input/output model of the induction motor. This is essentially a one-step ahead prediction structure in which we use past inputs and outputs to predict the current output The feedforward structure is used for application in neural-predictive control [9], where [.] contains data from the plant. The regression vector for this network is given by

$$\begin{aligned} \varphi(k) = [&\mathbf{y}(k-1),\ldots,\mathbf{y}(k-n_a), \\ &\mathbf{u}(k-n_k),\ldots,\mathbf{u}(k-n_b-n_k+1)] \end{aligned} \tag{21}$$

The predicted output is parameterized in terms of network weights $\mathbf{w}$ by

$$\hat{\mathbf{y}}(k) = \hat{g}[\varphi(k),\mathbf{w}] \tag{22}$$

The ANN used for induction motor modelling is a feedforward multilayer network as shown in Fig. 1, with one hidden layer activated by *tanh* (hyperbolic tangent) function and output layer activated by linear function. The orders and delay of this structure are chosen intuitively where $n_a = n_b = 1$ and $n_k = 1$. The training of ANN is done off-line by using the Levenberg-Marquardt (LM) optimization technique which aims to minimize the mean square error between the plant output and the ANN output over the training data set. The procedure is based on batch mode, where the weights updating is performed only after the entire training set has been applied to the network [10]. The training data set (inputs and desired outputs data) is taken from vector controlled induction motor drive. In order to have a good training, the data must contain sufficient information about the system dynamics.

For LM algorithm, the performance index to be optimized is defined as

$$E(\mathbf{w}) = \sum_{k=1}^{P}\left[\sum_{j=1}^{O}\left(d_{jk} - \hat{y}_{jk}\right)^2\right] \qquad (23)$$

where **w** is the vector of all weights of the network, d_{jk} is the desired value of the j^{th} output and the k^{th} pattern, $\hat{y}_{jk}$ is the actual value of the j^{th} output and the k^{th} pattern, P is the number of patterns and O is the number of network outputs.

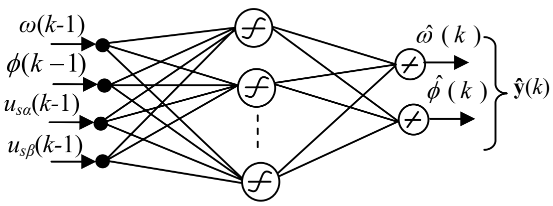

Fig.1. NNARX model for induction motor.

4 Neural Predictive Control

The neural predictive control is basically a type of model based predictive control, where the model for prediction is a neural network. The model predictive control strategy is based on the use of a model to predict the future output trajectory of the process. Next, the algorithm computes the future control actions to minimize a performance index at each sample k, and the first action is applied. The procedure is repeated at sample k+1.

The j-step ahead prediction of the system output by using the NNARX model is shown in Fig. 2 and has this form

$$\hat{\mathbf{y}}(k+j) = \hat{g}[\,\hat{\mathbf{y}}(k+j-1), \ldots, \hat{\mathbf{y}}(k+j-n_a), \mathbf{u}(k+j-n_k), \ldots, \mathbf{u}(k+j-n_b-n_k+1)\,] \qquad (24)$$

The initial inputs to the ANN are given by the plant.

The cost function to be minimized is

$$\Im = \sum_{j=N_1}^{N_2}\left[\mathbf{y}_{ref}(k+j) - \hat{\mathbf{y}}(k+j)\right]^2 + \lambda\sum_{j=1}^{N_u}\left[\Delta\mathbf{u}(k+j-1) - \Delta\mathbf{u}_{ref}(k+j-1)\right]^2 \qquad (25)$$

with the following assumption

$$\Delta\mathbf{u}(k+j) = \Delta\mathbf{u}_{ref}(k+j) \quad \text{for } j \geq N_u \qquad (26)$$

Where

$\hat{\mathbf{y}} = [\hat{\omega} \quad \hat{\phi}]^T$; $\mathbf{y}_{ref} = [\omega_{ref} \quad \phi_{ref}]^T$

$\mathbf{u} = [u_{s\alpha} \quad u_{s\beta}]^T$; $\mathbf{u}_{ref} = [u_{s\alpha ref} \quad u_{s\beta ref}]^T$

N_u is the control horizon, N_1 the minimum prediction horizon, N_2 the prediction horizon, λ is the control weighting factor and Δ is the difference operator $\Delta u(k) = \Delta u(k) - \Delta u(k-1)$.

The reference control signals $u_{s\alpha ref}$ and $u_{s\beta ref}$ are obtained from references ω_{ref}, ϕ_{ref} and the nominal values of the machine parameters in equation (20), this model is the reference control model.

After the training is over the ANN are arranged as shown in Fig. 3 for using as the predictor. After minimization of the cost function is done, the first value $\mathbf{u}(k)$ of the optimal control is applied to the system, whereas the whole control vector **U** is used by NNP according to the receding horizon strategy.

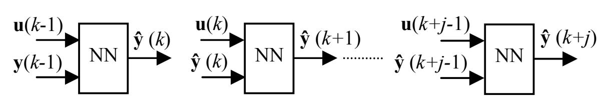

Fig.2. Neural Network Predictor Structure (NNP).

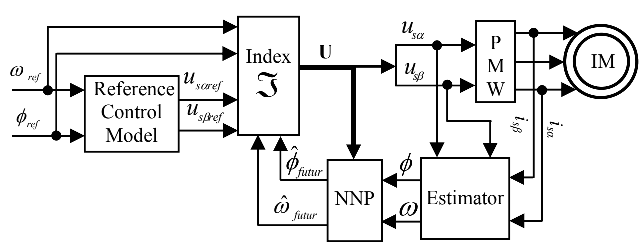

Fig.3. Block Diagram of NPC for IM.

With the Newton-Raphson (NR) method, $\Im$ is minimized iteratively to determine the best control vector [11].

$$\mathbf{U}^{(n)}(k) = [\mathbf{u}(k) \quad \mathbf{u}(k+1) \quad \ldots. \quad \mathbf{u}(k+N_u-1)]^T \qquad (27)$$

The NR update for $\mathbf{U}^{(n+1)}(k)$ is

$$\mathbf{U}^{(n+1)}(k) = \mathbf{U}^{(n)}(k) - \left(\left.\frac{\partial^2\Im(k)}{\partial\mathbf{U}(k)^2}\right|_{\mathbf{U}(k)=\mathbf{U}^{(n)}(k)}\right)^{-1}\left.\frac{\partial\Im(k)}{\partial\mathbf{U}(k)}\right|_{\mathbf{U}(k)=\mathbf{U}^{(n)}(k)} \qquad (28)$$

where n is the number of iterations.

The quantities $\partial \Im(k)/\partial \mathbf{U}(k)$ and $\partial^2 \Im(k)/\partial \mathbf{U}(k)^2$ are the Jacobian and Hessian matrices (the first and second derivatives of cost function) which are calculated from (25), (26) by using NNARX plant model (22) and *tanh* propriety.

$$f(x) = tanh(x) : \begin{cases} \dot{f}(x) = 1-(f(x))^2 \\ \ddot{f}(x) = -2f(x)\left[1-(f(x))^2\right] \end{cases} \quad (29)$$

In order to avoid the computation of the inverse of the Hessian matrix, equation (28) is rewritten in the form of a system of linear equation $\left(\frac{\partial^2 \Im(k)}{\partial \mathbf{U}(k)^2}\right)\left(\mathbf{U}^{(n+1)}(k)-\mathbf{U}^{(n)}(k)\right)=-\frac{\partial \Im(k)}{\partial \mathbf{U}(k)}$ which is solved for $X = \mathbf{U}^{(n+1)}(k)\text{-}\mathbf{U}^{(n)}(k)$ by using LU decomposition.

When solving for X, calculation of each element of the Jacobian and Hessian is needed for each NR iteration.

5 Simulation Results

The simulations have been carried out to verify the performance of neural network based input-output modeling and the neural predictive controller.

For the induction motor modeling, the discrete NNARX model with $n_a = n_b = 1$ and $n_k = 1$ in the structure (21) is selected; so, the neural network used for this model has 4 nodes in the input layer (the previous values of inputs and outputs from the induction motor), 10 nodes in the hidden layer and two nodes in the output layer for rotor speed and rotor flux norm.

To train the ANN, the training data set is collected from vector controlled induction motor (all inputs and outputs are known by measurement and estimation respectively). In order to check the influence of the model parameter variations and disturbances on the ANN, the rotor resistance R_r and the load T_L are varied. The variations of R_r have the values in time intervals (second) as: 2.61 Ω in [0 2], 3.61 Ω in [2 3], 1.61 Ω in [3 4], finally 2.61 Ω in [4 5], and for T_L disturbance: 0 Nm in [0 0.5], 0.2 Nm in [0.5 2.5], 0.38 Nm in [2.5 4], and 0.28 Nm in [4 5].

During training of the ANN, we use a training data set which consists of the inputs to the ANN and desired outputs. The desired outputs are compared with the ANN outputs for updating the weights. The results of simulation during the training are shown in Fig. 4 – 5, where the speed and the rotor flux norm obtained by the ANN model of the motor are compared with those used in training data set.

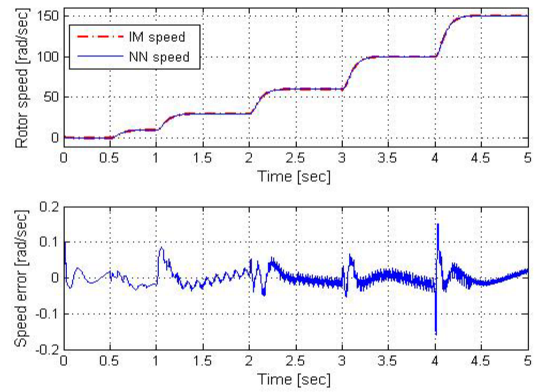

Fig.4.a. Rotor speed from Motor and ANN.
b. Motor and ANN speed error.

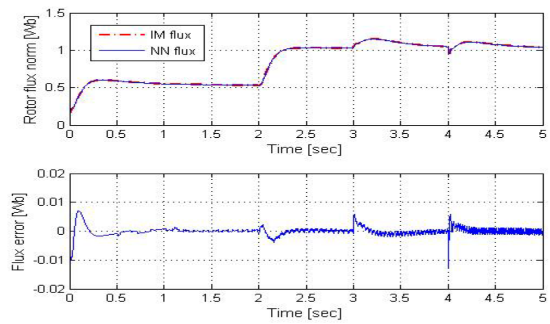

Fig.5.a. Rotor flux norm from Motor and ANN.
b. Motor and ANN flux error.

It is observed from the results of simulation that the proposed model using an ANN gives satisfactory results.

The trained ANN model is placed in the NPC control loop as shown in Fig. 3, the parameters of the index performance (25) for neural predictive controller are chosen by trial and error method in order to produce the desired response.

N_1=1, N_2 = 5, N_u = 2, λ = 2.10^7

In total, five trained ANN are connected as shown in Fig. 2 and work as predictor. The control model reference is carried out using the formula (20) where the reference outputs are shown in Fig. 6 and 7. However, the variations of R_r and T_L are different from the ones considered earlier. These values are (i) for R_r 2.61 Ω in [0 2], 2Ω in [2 2.5] and 2.2Ω in [2.5 5] and (ii) for T_L disturbance: 0 *Nm* in [0 2] and 0.25 *Nm* in [2 5]. They are used to check the performance of the ANN predictor with other variations not used in training.

Fig. 6 and 7 give respectively the responses (speed and flux) of the NPC induction motor drive system and their corresponding errors with references. As shown, the tracking performance of the proposed NPC system is satisfactory achieved, where the controller tries to reduce the influence of the R_r variations when the outputs follow the references.

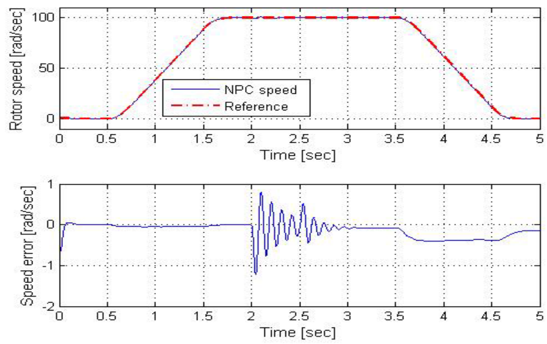

Fig.6.a. Rotor speed from Motor and ANN.
b. Motor and ANN speed error.

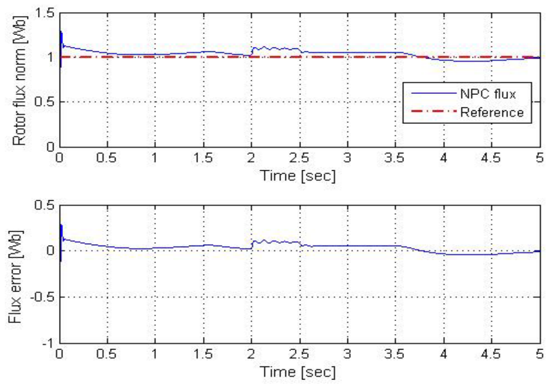

Fig.7.a. Rotor flux norm from Motor and ANN
b. Motor and ANN flux error

6 Conclusion

In this paper, we present an application of neural networks for input-output modeling of an induction motor, which permits to approximate the process by a NNARX model. The choice of neural network structure depends upon the performance needed. The ANN training is done off-line using the Levenberg-Marquardt algorithm. The advantage of this modeling by ANN is the ability to have a nonlinear model with good performance without requiring the knowledge of machine parameters which may vary in time.

The controller development follows the generalized predictive control methodology with the process represented by the NNARX model. A Newton-Raphson method is used for cost function minimization.

The results of simulation show that the performance of the NNARX model and the NPC controller are satisfactory.

References:

[1] E. Delaleau, J.P. Louis and R. Ortega, Modeling and Control of Induction Motors, *International Journal of Applied Mathematics and Computer Science*, Vol.11, No.1, 2001, pp. 105-129.

[2] J. Chiasson, Dynamic Feedback Linearization of the Induction Motor, *IEEE Transactions on Automatic Control*, Vol.38, No.10, 1993, pp. 1588-1594.

[3] R. Marino, S. Peresada and P. Valigi, Adaptive Input-Output Linearizing Control of Induction Motors, *IEEE Transactions on Automatic Control*, Vol.38, No.2, 1993, pp. 208-221.

[4] P. Vas, *Artificial-Intelligence-Based Electrical Machines and Drives: Application of Fuzzy, Neural, Fuzzy Neural and Genetic Algorithm-Based Techniques*, Oxford University Press, Inc. New York, 1999.

[5] L. Constant, R. Ruelland, B. Dagues, I. Rivals and L. Personnaz, Identification and Validation of a Neural Network Estimating the Fluxes of an Induction Machine, *6th International Conference on Modelling and Simulation of Electric Machines, Converters and Systems ELECTRIMACS'99*, 1999.

[6] J. F. Martins, A. J. Pires and J. A. Dente, Automatic Input/Output Modeling of a Squirrel-Cage Induction Motor Drive System using Neural Network, *EPE'97*, 4, 1997, pp. 632-637.

[7] C.E. Garcia, D.M. Prett and M. Morari, Model Predictive Control: Theory and Practice- a Survey, *Automatica*, 3, 1989, pp. 335-348.

[8] M.K. Maaziz, E. Mendes and P. Boucher, A New Nonlinear Multivariable Control Strategy of Induction Motors, *Control Engineering Practice*, 10, 2002, pp. 605-613.

[9] P.H. Sørensen, M. Nørgaard, O. Ravn, and N.K. Poulsen, Implementation of Neural Network based Non-linear Predictive Control, *Neurocomputing*, 28, 1999, pp. 37-51,

[10] M.H. Hassoun, *Fundamentals of Artificial Neural Networks*, Cambridge, Massachusetts: The MIT Press, 1995.

[11] D. Soloway and P. J. Haley, Neural Generalized Predictive Control: A Newton-Raphson Implementation, *Proc.11th IEEE International Symposium on Intelligent Control*, 1996, pp. 277-282.