Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005 (pp260-267)

Design of a Robust and Adaptive Sensorless Speed Controller for Induction Motor Drives Using General regression Neural Network Based Fuzzy Approach

FARZAN RASHIDI
Faculty of Engineering
Islamic Azad University of Bushehr
Bushehr, IRAN

Abstract: The main purpose of this paper is to apply the Fuzzy based General Regression Neural Network (FGRNN) to the speed control of induction motor. A General Regression Neural Network (GRNN) is adopted to estimate the motor speed and thus provide a sensorless speed estimator system. The performance of the proposed FGRNN speed controller is evaluated for a wide range of operating conditions for induction motor. These include startup and parameters variations. Obtained results show that the GRNN provides a very satisfactory speed estimation under the above mentioned operation conditions and also the sensorless FGRNN speed controller can achieve very robust and satisfactory performance and could be used to get the desired performance levels. The response time is also very fast despite the fact that the control strategy is based on bounded rationality. To evaluate the usefulness of the proposed method, we compare the response of this method with PID controller. The simulation results show that our method has the better control performance than PID controller.

Key-Words: - Induction motor derives, general regression neural network, Fuzzy logic, Speed Control

1 Introduction

Variable speed motor drives play an important role in modern industries because they are utilized extensively in factory automation to store energy or to meet stringent load requirements. The use of variable speed motor drives is ever increasing and will maintain its momentum for several decades to come [1]. Among all different kinds of electric motor drives, the induction motor has become the subject of a large body of research in the field of electric motor drives. This is partly because the motor has an intrinsically simple and rugged structure and low manufacturing cost. Moreover, induction motor drives have the wide speed range, high efficiencies, and robustness [2]. All these merits make the motor a good candidate for the industrial applications.
Sensorless control of induction motor drives is now receiving wide attention. The main reason is that the speed sensor spoils the ruggedness and simplicity of induction motor. In a hostile environment, speed sensors can not even be mounted. However, due to the high order and nonlinearity of the dynamics of an induction motor, estimation of the angle speed and rotor flux without the measurement of mechanical variables becomes a challenging problem [3]. The advantages of speed sensorless induction motor drives are reduced hardware complexity and lower cost, reduce size of drive machine, eliminate of sensor cable, better noise immunity, increasing reliability and less maintenance requirements [4]. Various speed control algorithms for induction motor drives have been devised in the literature. Among them, PID controllers [5], optimal [6], nonlinear [7,8] and robust [9,10] control strategies, and neural and/or fuzzy [3,11] approaches are to be mentioned. The purpose of this paper is to suggest another control approach, based on FGRNN to achieve faster response with reduced overshoot and rise time. To evaluate the usefulness of the proposed method, we compare the response of this method with PID controller. The simulation results show that our method has the better control performance than PID controller. In the next subsections we discusses the mathematical model of the induction motor derive, our proposed method, obtained results and some conclusion and remarks.

2 Mathematical Model of Induction Motor Drive

Many schemes based on simplified motor models have been devised to sense the speed of the induction motor from measured terminal quantities for control purposes. In order to obtain an accurate dynamic representation of the motor speed, it is necessary to base the calculation on the coupled circuit equations of the motor [5]. Since the motor voltages and currents are measured in a stationary frame of reference, it is also convenient to express these equations in that stationary frame [5]. From the stator voltage equations in the stationary frame it is obtained [6]:

$$\begin{aligned} \frac{d\lambda_{qr}}{dt} &= \frac{L_r}{L_m}\left(V_{qs} - (R_s + \sigma L_s \frac{d}{dt})i_{ds}\right) \\ \frac{d\lambda_{dr}}{dt} &= \frac{L_r}{L_m}\left(V_{ds} - (R_s + \sigma L_s \frac{d}{dt})i_{qs}\right) \end{aligned} \tag{1}$$

Where λ is the flux linkage; L is the inductance; V is the voltage; R is the resistance; i is the current, and $\sigma = 1 - \frac{L_m^2}{L_r L_s}$ is the motor leakage coefficient. The subscripts r and s denotes the rotor and stator values respectively refereed to the stator, and the subscripts d and q denote the dq-axis components in the stationary reference frame. The rotor flux equations in the stationary frame are [6]:

$$\begin{aligned} \frac{d\lambda_{dr}}{dt} &= \frac{L_m}{\tau_r} i_{ds} - \omega_r \lambda_{qr} - \frac{1}{\tau_r}\lambda_{dr} \\ \frac{d\lambda_{qr}}{dt} &= \frac{L_m}{\tau_r} i_{qs} + \omega_r \lambda_{dr} - \frac{1}{\tau_r}\lambda_{qr} \end{aligned} \tag{2}$$

Where ω_r is the rotor electrical speed and $\tau_r = \frac{L_r}{R_r}$ is the rotor time constant. The synchronous frequency in stationary frame is defined as follows [6]:

$$\omega_e = \frac{\lambda_{dr}\frac{d\lambda_{qr}}{dt} - \lambda_{qr}\frac{d\lambda_{dr}}{dt}}{\lambda_{dr}^2 + \lambda_{qr}^2} \tag{3}$$

Substituting the equation (2) in the equation (3) it is obtained:

$$\omega_e = \omega_r - \frac{L_m}{\tau_r}\left(\frac{\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}}{\lambda_{dr}^2 + \lambda_{qr}^2}\right) \tag{4}$$

Then substituting the equations (3) in the equation (4), and finding ω_r we obtain [6]:

$$\omega_r = \frac{1}{\lambda_{dr}^2 + \lambda_{qr}^2}\begin{bmatrix} \lambda_{dr}\frac{d\lambda_{qr}}{dt} - \lambda_{qr}\frac{d\lambda_{dr}}{dt} \\ -\frac{L_m}{\tau_r}\left(\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}\right) \end{bmatrix} \tag{5}$$

Therefore, given a complete knowledge of the motor parameters, the instantaneous speed ω_r can be calculated from the eq. (5) where the stator currents and voltages are known along with the machine parameters, and the rotor flux linkages are obtained from equation (1).

3 Speed Estimation of Induction Motor Using General Regression Neural Network

Neural Networks are a family of intelligent algorithms which can be used for time series prediction, classification, control and identification. Neural networks have an ability to train with various parameter of induction motor. As a nonlinear function, they can be used for identifying the extremely nonlinear system parameters with high accuracy. Recently, the use of neural networks to identify and control nonlinear dynamic systems has been proposed because they can approximate a wide range of nonlinear functions to any desired degree of accuracy. Moreover, they have the advantages of extremely fast parallel computation, immunity from input harmonic ripples, and fault tolerance characteristics. Also there have been some investigations into the application of NNs to power electronics and ac drives, including speed estimation. This technique gives a fairly good estimate of the speed and is robust to parameter variation. However, the neural network speed estimator should be trained sufficiently with various patterns to get good performance. In this paper, a new speed estimation method of an induction motor is proposed. This method is based on general regression neural network (GRNN).

The GRNN is an extension of the Probabilistic Neural Network (PNN) to estimate continuous variable and one-pass learning algorithm [7]. In this network each layer is connected together. Hidden and output layers are the same as PNN. Rather than categorizing data like PNN, however, GRNN applications are able to produce continuous valued outputs. GRNNs are reported to respond better than backpropagation in many types of problems. The GRNN is implemented by feedforward architecture has 4 layers; Input, Pattern (Gaussian Kernals), Summation/Division, and Output layer. The input layer is the external buffer to the X training data. The connection weights from the input layer to the i^{th} PE in the pattern layer store the center $X^{(i)}$ of the i^{th} Gaussian kernel. PEs in the pattern layer are referred to as "pattern units." The summation function for the i^{th} pattern unit calculates the "City Block" distance C_i between the input vector and stored center $X^{(i)}$. The transfer function in the pattern layer transforms the calculated summation values C_i through the exponential transfer function. The *B coefficients* are encoded into the connection weights from the pattern layer to the first PE in the summation/division layer. The *A coefficients* are encoded into the connection weights from the pattern layer to the remaining PEs in the summation/division layer. The summation function of the summation/division layer is the standard weighted sum function. The output layer is the external buffer which outputs the estimated conditional means given in following equation and receives desired output information during learning [8].

$$\hat{Y}(X) = \frac{\sum_{j=1}^{n} A^i \exp(-\frac{C_i}{\sigma})}{\sum_{j=1}^{n} B^i \exp(-\frac{C_i}{\sigma})} \quad , \quad C_i = \sum_{j=1}^{p} |X_j - X^i_j|$$

$A^i(k) = A^i(k-1) + Y^j$, $B^i(k) = B^i(k-1) + 1$;

The basic problem in training a NN to recognize induction motor speed is that the functional relationship between the speed and stator parameters. As seen in equation (5) the ω_r is a function of the flux linkage and current, so if these two terms are considered as inputs to GRNN, it should be able to estimate the speed with high accuracy.

4 Fuzzy Based General Regression Neural Network

The fundamental approach with FGRNN is dividing the input space into small linear subspaces with fuzzy validity functions. Any produced linear part with its validity function can be described as a fuzzy neuron. Thus the total model is a neurofuzzy network with one hidden layer, and a linear neuron in the output layer which simply calculates the weighted sum of the outputs of locally linear models (LLMs).

$$\hat{y}_i = \omega_{i_0} + \omega_{i_1} u_1 + \omega_{i_2} u_2 + \ldots + \omega_{i_p} u_p$$
$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i \phi_i(\underline{u}) \qquad (7)$$

This structure is depicted in figure 1, where $\underline{u} = [\, u_1 u_2 \cdots u_p \,]^T$ is the model input, M is the number of LLM neurons, and ω_{ij} denotes the LLM parameters of the *i*th neuron. The validity functions are chosen as normalized Gaussians; normalization is necessary for a proper interpretation of validity functions.

$$\phi_i(\underline{u}) = \frac{\mu_i(\underline{u})}{\sum_{j=1}^{M} \mu_j(\underline{u})} \qquad (8)$$

$$\mu_i(\underline{u}) = \exp\left(-\frac{1}{2}\left(\frac{(u_1 - c_{i1})^2}{\sigma_{i1}^2} + \ldots + \frac{(u_p - c_{ip})^2}{\sigma_{ip}^2}\right)\right) =$$
$$\exp\left(-\frac{1}{2}\frac{(u_1 - c_{i1})^2}{\sigma_{i1}^2}\right) \times \ldots \times \exp\left(-\frac{1}{2}\frac{(u_p - c_{ip})^2}{\sigma_{ip}^2}\right) \qquad (9)$$

Each Gaussian validity function has two sets of parameters, centers (c_{ij} s) and standard deviations (σ_{ij} s) which are the $M.p$ parameters of the nonlinear hidden layer. Optimization or learning methods are used to adjust both the parameters of local linear models (ω_{ij} s) and the parameters of validity functions (c_{ij} s and σ_{ij} s). Global optimization of linear parameters is simply obtained by Least squares technique. The complete parameter vector contains $M.(p+1)$ elements:

$$\underline{\omega} = [\omega_{10} \quad \omega_{11} \quad \ldots \quad \omega_{1p} \quad \omega_{20} \quad \omega_{21} \quad \ldots \quad \omega_{M0} \quad \ldots \quad \omega_{Mp}]$$

And the associated regression matrix $\underline{X}$ for N measured data samples is:

$$\underline{X} = [\underline{X}_1 \quad \underline{X}_2 \quad \ldots \quad \underline{X}_M]$$
$$\underline{X}_i = \begin{bmatrix} \phi_i(\underline{u}(1)) & u_1(1)\phi_i(\underline{u}(1)) & \ldots & u_p(1)\phi_i(\underline{u}(1)) \\ \phi_i(\underline{u}(2)) & u_1(2)\phi_i(\underline{u}(2)) & \ldots & u_p(2)\phi_i(\underline{u}(2)) \\ \vdots & \vdots & & \vdots \\ \phi_i(\underline{u}(N)) & u_1(N)\phi_i(\underline{u}(N)) & \ldots & u_p(N)\phi_i(\underline{u}(N)) \end{bmatrix} \qquad (10)$$

Thus

$$\hat{\underline{y}} = \underline{X}.\hat{\underline{\omega}} \qquad ; \qquad \hat{\underline{\omega}} = (\underline{X}^T \underline{X})^{-1} \underline{X}^T \underline{y}$$

The remarkable properties of locally linear neurofuzzy model, its transparency and intuitive construction, lead to the use of least squares technique for rule antecedent parameters and incremental learning procedures for rule consequent parameters.

5 Learning algorithm of the FRGNN

The most important property of FGRNN model is that one can use some intuitive algorithms in training. The model starts as an optimal least squares estimation, and the new local linear models are created to reduce the prediction error. It implements a heuristic search for the rule premise parameters and avoids a time consuming nonlinear optimization. The FGRNN algorithm is described in five steps as follows:

1. *Start with an initial model:* start with a single LLM, which is a global linear model over the whole input space with $\Phi_1(\underline{u})=1$ and set $M=1$. If there is a priori input space partitioning it can be used as the initial structure.

2. *Find the worst LLM*: Calculate a local loss function e.g. MSE for each of the $i=1,\ldots,M$ LLMs, and find the worst performing LLM.

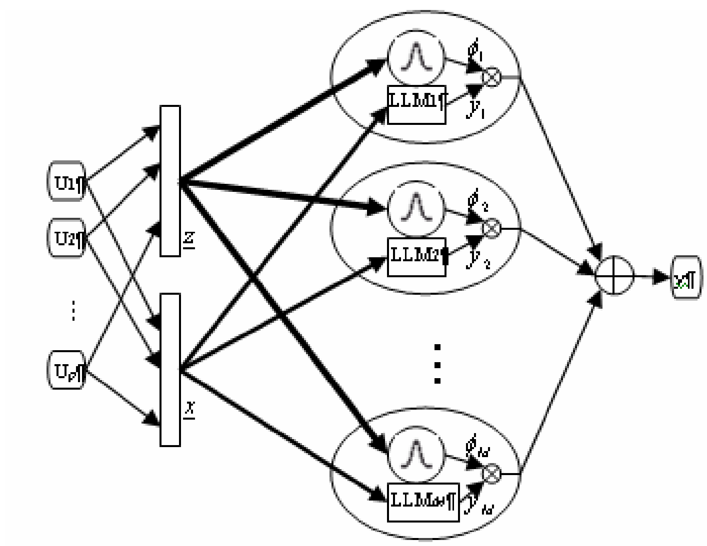

Figure 1: The structure of the FGRNN

3. *Check all divisions*: The worst LLM is considered for further refinement. The hyper rectangle of this LLM is split into two halves with an axis orthogonal split. Divisions in all dimensions are tried, and for each of the p divisions the following steps are carried out:

a. Construction of the multi-dimensional membership functions for both generated hyper rectangles;

b. Construction of all validity functions.

c. Local estimation of the rule consequent parameters for both newly generated LLMs.

d. Calculations of the loss function for the current overall model.

4. *Find the best division*: The best of the p alternatives checked in step 3 is selected, and the related validity functions and LLMs are constructed. The number of LLM neurons is incremented $M=M+1$.

Test the termination condition: If the termination condition is met, then stop, else go to step 2.

This algorithm trains the model automatically, and requires just one user defined parameter: the termination condition. To avoid over fitting and to provide maximum generalization, which is the most important property of a controller, the termination condition is defined by checking the error index on validation sets. The optimal number of neurons is achieved when the error index on validation sets starts to increase**.**

6 Simulation Results

In this section we illustrate the performance of the proposed method by simulations using the MATLAB software package. Figure 2 shows the Block diagram of speed-sensorless controller using FGRNN. The induction motor considered in this paper has the following data and parameters:

3 phase, 380V, 15KW, 31A, 2 poles ,2895rpm

$$R_r=1.46\Omega,\ R_s=0.603\Omega,\ L_{lr}=4.72mH,$$

$$L_{ls}=4.72mH,\ L_m=330.2mH$$

Using the algorithm developed in the previous sections, the dynamic performance of the FGRNN has been simulated for a wide range of operating conditions. The results of these tests will be detailed below. In addition, the important effects of parameter variations are examined in detail in order to verify the robustness of the system. In first simulation the speed command is set as 50rad/sec. The speed response when the induction motor is started from standstill has been given in Fig. 3 that compared with PID controller. The performance levels achieved via the two alternative approaches are outlined in table 1. As can be seen from this table the neural based fuzzy regression approach has the better control performance than PID controller in terms of settling time, overshot and rise time.

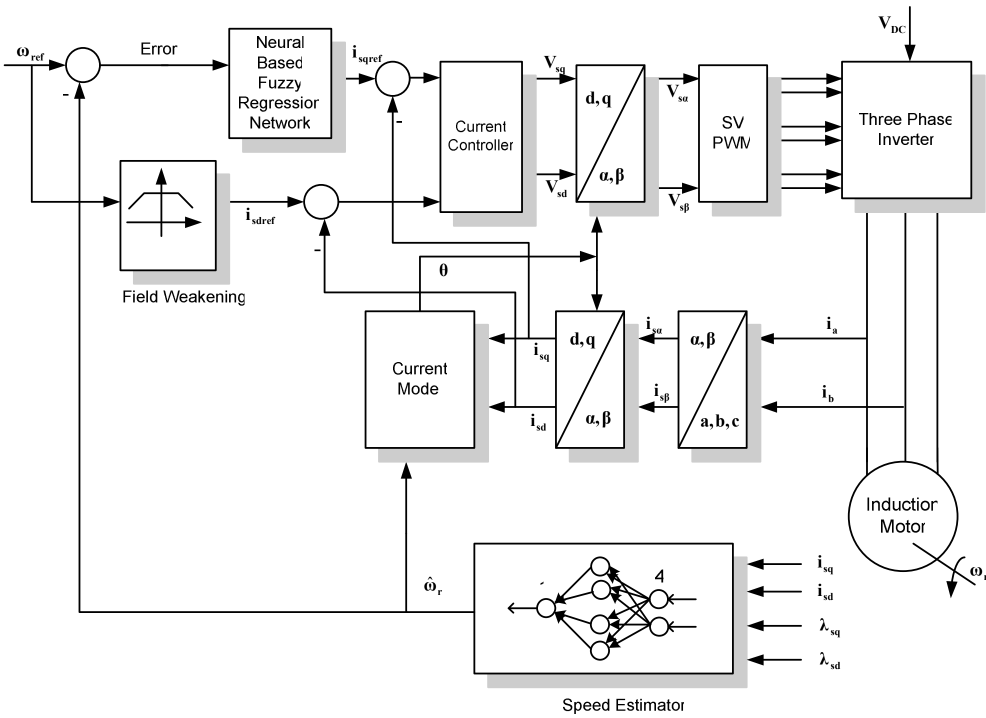

Figure 2: Block diagram of speed-sensorless controller using FGRNN

The speed response when the speed command is set as 190rad/s is given in Fig. 4 for proposed method and PID controller. As can be seen from this figure the FGRNN has the better control performance than PID controller in terms of settling time, overshot and rise time. Now we test the robustness of the adaptive fuzzy-neural controller toward modeling errors.

The mechanical parameters (*J, B*) can not always be obtained accurately and they also vary during the induction motor operation. However, the uncertainties on the mechanical parameters can be well handled by the process of load torque estimation [3]. Thus in this section only the test under electrical parameter variations is done here. We illustrate the robustness toward electrical parameters by using rotor resistance as an example. During the induction motor operation, R_r will increase due to temperature rise. At the beginning of this test, the measured R_r =1.46Ω. Suppose this value changes to R_r =1.8Ω when t= 0.2 sec. The speed command is 190rad/s. The result of Fig. 5 shows the robustness against this resistance variation. It is shown in Fig. 5 that the estimated speed can still track the reference speed very well without noticeable deviation. Hence in this test, we demonstrated that the developed GRNN-based sensorless fuzzy general regression-neural network is robust to the electrical parameter variations.

Fig. 6 shows the behavior of the proposed controller under these step changes in speed command. It can be seen that the motor speeds exhibit smooth performance under different reference speed, and a rapid transient response. Therefore in this test it is shown clearly that the adaptive FGRNN controller can track variable reference speed rapidly and smoothly.

Table 1- Performance characteristics of Induction motor with two PID and FGRNN controllers

	S-S Error	Rise Time	POS
FGRNN	0.00%	0.24	0.00%
PID	0.00%	0.32	10%

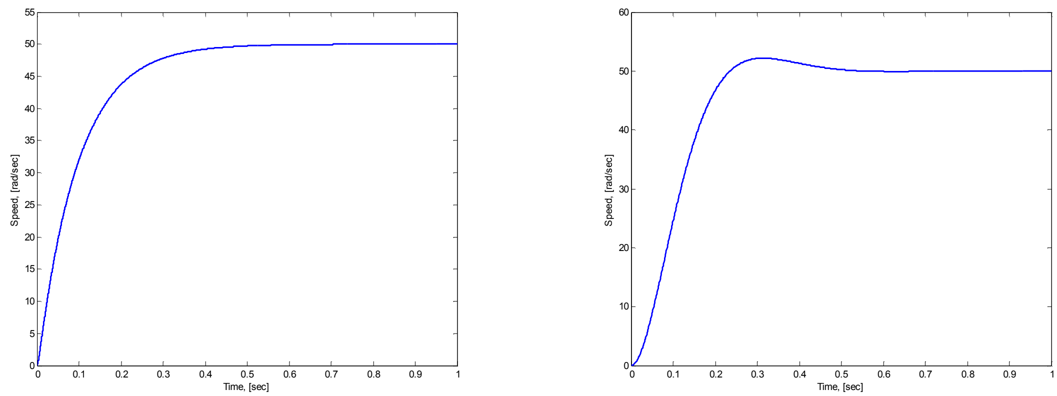

Fig.3. Speed response when speed command is set as 50rad/s (Left: FGRNN, Right: PID controller)

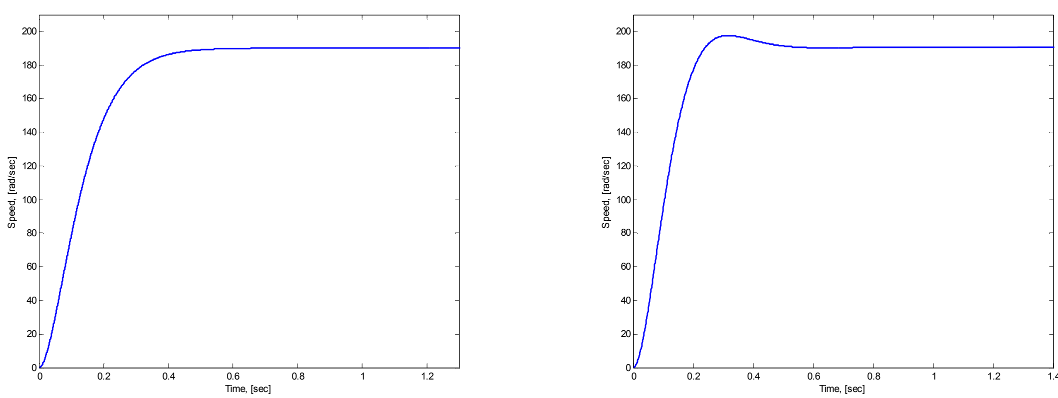

Fig.4. Speed response when speed command is set as 190rad/s (Left: FGRNN, Right: PID controller)

It is common that in many cases the induction motor may be operated with an unknown load torque. Thus, the performance of the developed scheme under unknown load torque is examined here. In this test, the external load torque is set to 5Nm during 0-0.2sec, 7Nm during 0.2-0.4sec, and 9Nm during 0.4-0.6sec. The reference speed is 100rad/s and the initial estimated load torque is set as 0Nm. In Fig. 7 we can see the speed responses under these large step changes in the load torque. We can see even with such fast changes in load the observer-based controller also exhibits very good performance with a fast response. Hence, the results above have shown the robustness of the adaptive FGRNN speed controller toward variable load torque, which is the common case in practice.

7 Conclusion

In this paper, a novel FGRNN based rotor speed estimation algorithm was presented for induction motor drives. Furthermore GRNN was designed to estimate velocity in the whole speed range to provide a sensorless speed estimator system. To demonstrate the effectiveness and applicability of the proposed method, simulation results under the whole range of the operation conditions was presented and compared with PID controller, including low and high speed and parameters variations as well. In conclusion, the proposed FGRNN presented a very good performance under the whole range of operation conditions. Moreover it was also robustness to the parameters variations.

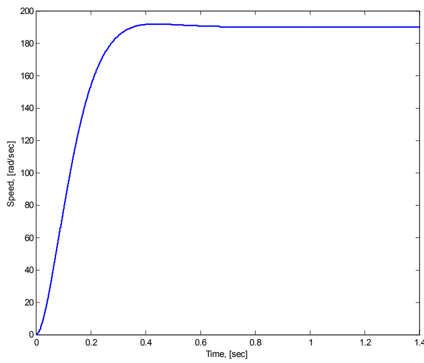

Fig. 5. Speed response when the rotor resistance was changed at t=0.2 sec

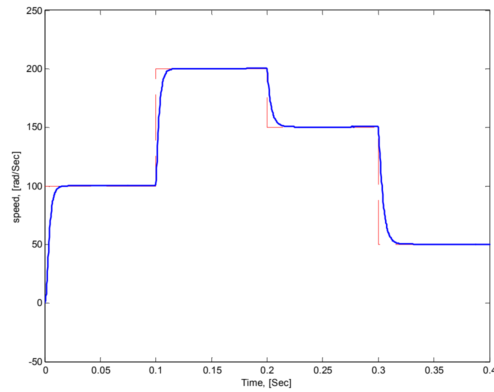

Fig. 6. Speed response when speed command is changed

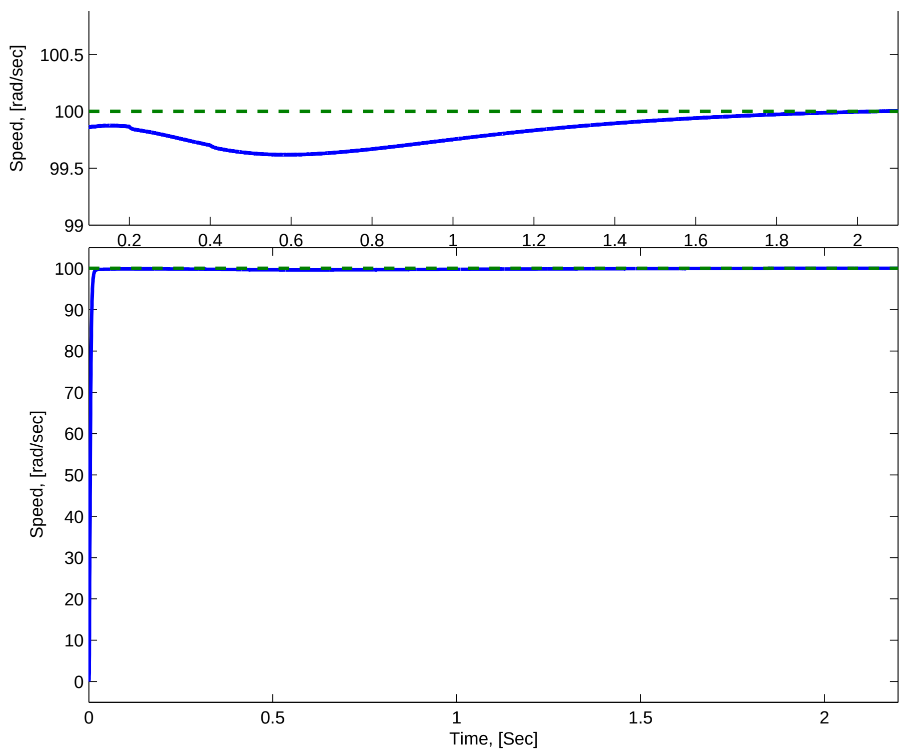

Fig. 7. Speed response when the external load torque is changed

Reference

[1]- Rashidi. F., C. Lucas, “Applying CBBEL (Context Based Brain Emotional Learning) to SR Motors”, WSEAS Transactions on Systems, 2003, 1082-1086.

[2]- Bottura, C.P., Neto, M.F.S.; Filho, S.A.A., ”Robust speed control of an induction motor: an /spl Hscr/sub /spl infin/ control theory approach with field orientation and /spl mu/-analysis”, IEEE Transactions on Power Electronics, Vol. 15, pp. 908-915, 2000

[3]- Rashidi, F., Rashidi, M., “Design of robust sliding mode speed control with fuzzy approach for induction motors”, IEEE International Conference on Industrial Technology, ICIT03, 2003, pp. 27-30

[4]- Joachim Holtz, “Sensorless speed and position control of induction motor derives”, IEEE international conference on Industrial Electronics, IECON03, 2003,

[5]- Ho-Seok Lee; Taeck-Kie Lee; Soon-Bong Cho; Dong-Seok Hyun, “Speed control of induction motor using fuzzy algorithm with hierarchical structure”, IEEE Conf., TENCON '93, vol.5, 551-554, 1993

[6]- Bose, B.K., 2001, Modern Power Electronics and AC Drives., Prentice Hall, New Jersey.

[7]- Specht, F. Donald, “A General Regression Neural Network,” IEEE Transactions on Neural Networks, Vol. 2, No. 6, November 1991, pp.568-576.

[8]- Specht, F. Donald, “Probabilistic Neural Network and General Regression Neural Network,” 1995 IEEE International Conference on Neural Networks Tutorial, Western Australia, November 1995.

[9]- Guang Feng; Lipei Huang; Dongqi Zhu**;”**A robust controller for improving disturbance rejection in speed control of induction motors”, International Conference on Power Electronics and Drive Systems, Vol. 1, pp. 432-437, 1999

[10]- C. Schauder, “Adaptive speed identification for vector control of induction motor without rotational transducers,” IEEE-IAS., 1989, pp. 493–499.

[11]- I. Ben-Brahim and T. Kudor, “Implementation of an induction motor speed estimator using neural networks,” in Proc. IPEC, 1995, pp. 52–57.

[12]- Rashidi, F., Online short term load forecasting in power systems using emotional critic based fuzzy approach”, LNAI journal, 2004

[13]- Ho-Seok Lee; Taeck-Kie Lee; Soon-Bong Cho; Dong-Seok Hyun, “Speed control of induction motor using fuzzy algorithm with hierarchical structure”, IEEE Conf., TENCON '93, vol.5, 551-554, 1993

[14]- Bose, B.K., 2001, Modern Power Electronics and AC Drives., Prentice Hall, New Jersey.

[15]- Rashidi, F., M.Rashidi, “Robust and Adaptive Tuning of Power System Stabilizers Using Artificial Neural Networks”, lecture notes in artificial intelligence journal, 2004

[16]- Huang, S.J., Huang, C.L. and Lin, Y.S., 1998, Sensorless speed identification of vector controlled induction drives via neural network based estimation, Electric Power System Research, 48, 1-10.

[17]- P. K. Simpson, "Artificial neural systems: Foundations, Paradigms, Applications, and Implementations", Pergamon, Elmsford, NY, 1990.

[18]- J. Stanley, "Introduction to neural networks", California Scientific Software, Siera Madre, CA, 3rd edition, 1990