Neural Network Based Direct Adaptive Control of Permanent Magnet Synchronous Motor

Ashraf A. HAGRAS
Nuclear Research Centre, Egyptian Atomic Energy Authority
Abo Zaabal, Cairo, 13759, ashrafa1973@yahoo.com

Abstract: *This paper proposes neural networks based direct adaptive state feedback control for the uncertain model of permanent magnet synchronous motor (PMSM). The proposed method used two types of neural networks (NN) to approximate the unknown model of PMSM; Backpropagation and Radial Basis Function (RBF) neural networks. Backpropagation (BP) and RBF neural network with single hidden layer were trained using MATLAB/NN Toolbox. The simulation results proved the effectiveness of the proposed method at various operating conditions. It exhibits considerable amount of torque ripples, adequate dynamic torque performance and improved speed response. Also, RBF neural network proved its superiority over backpropagation neural network in terms of faster speed and torque response at start up, steady state, step loaf disturbance and speed variations.*

Key words: *RBF neural Network, PB neural Network, Adaptive Control, Permanent Magnet Synchronous Motor.*

1. Introduction.

Permanent magnet synchronous motors have gained popular interest in the industry area due to their high efficiency, high power density, large torque to inertia ratio, wide speed operation, absence of rotor windings and absence of external rotor excitation. Due to these advantages, many control methodologies were applied to achieve high performance requirements such as adaptive backstepping, feedback linearization, fuzzy and neural networks based controllers. Some of them don't make the performance of the system at the optimum level due to the high nonlinearities present in the motor or because some of the motor parameters isn't possible to measure them. Therefore, an adaptive controller is required to maintain the improved system performance in the presence of parameters uncertainties or external disturbance.

Input-output linearization and feedback linearization design may lead to cancellation of some useful nonlinearities in the design stage. This requires the known of exact values of the plant parameters to compensate for the parameters uncertainties, load torque disturbance and magnet flux. This make the controllers may give unsatisfactory results as the plant parameters vary because of different operating conditions such as variation of temperature, saturation and external disturbances.

Therefore, adaptive input-output linearization with on-line estimation of the stator resistance, flux and motor inductance [1], adaptive backstepping and dynamic surface controllers [2-4] have employed to overcome these operating conditions. Although they compensated the uncertainties and various operating conditions, have asymptotic stability, improved response and preserve and take all the nonlinearities into account in the design of the controller, they require the knowledge the exact motor model.

Model reference adaptive control (MRAC) is based on the parameter update law which is based on the error difference between two models; reference model and adjustable model. Recent research tried to avoid depending on the stator resistance because it may suffer from integrator related problems like drift and saturation. Therefore, reactive power based (MRAC) for speed estimation is more popular as it is less parameter sensitive depending only on the stator inductance [5, 6].

Many methods were proposed to estimate the parameters of the PM motor especially the speed to eliminate the sensors cost, temperature sensitivity and reliability decrease. These methods like back-emf based method which is highly sensitive to machine parameters and state observer-based method which employ Extended Kalman Filter (EKF), Extended Luenburger observer (ELO) and sliding mode observer. But EKF requires long computational time because of the several matrix operations and suffers from the computational complexity and parameters sensitivity. Moreover, the sliding mode observer suffers from the chattering problem [7-11].

In these controllers, the estimated parameters were used directly in the controllers or to calculate the controller parameters. These methods can be considered parameters dependent control methods. But in this paper, a simple method of the adaptive controller is based on the uncertain model of the motor not on the estimated parameters. This method doesn't depend on any equation based on model parameters calculations. Neural networks were known as universal function approximators of highly nonlinear models. Therefore, they were used to model the PM motor without prior knowledge of the PM parameters [16].

Neural Networks can save lots of efforts on system modeling. It doesn't require the mathematical model of the plant. Therefore, it can be used to approximate continuous unknown nonlinear systems using previous measured data of the plant where they were used to reproduce the dynamics of the plant by building a model from the experimental observations of the system.

In this paper, the model was investigated under different values of supply voltages and was trained off line. The model was created by NNTOOL command in the MATLAB package.

2. Mathematical Modeling of Permanent Magnet Synchronous Motor.

The d and q-axis stator voltages for PMSM referred to rotor reference frame may be expressed as:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R + pL_d & -\omega_s L_q \\ \omega_s L_d & R + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_s \lambda_{af} \end{bmatrix} \quad (1)$$

where v_d and v_q are the d-q axis stator voltages, R is the stator resistance, L_d and L_q denote the stator inductances in the d-q axes, P is the number of pole pairs, i_d and i_q are the d-q axis stator currents, ω_s is the rotor speed, λ_{af} is the rotor permanent magnetic flux [12, 13].

The developed electromagnetic torque can be expressed as;

$$T_e = J\frac{d\omega_s}{dt} + B\omega_s + T_L$$

The governing electromechanical equation is:

$$T_e = \frac{3P}{2}\left(\lambda_{af} i_q + \left(L_d - L_q\right) i_d i_q\right)$$

Where T_e is the developed torque, J is the rotor moment of inertia, B is the viscous friction factor and T_L also represents the applied load torque disturbance. If $L_d = L_q$ for surface mounted PMSM, Then, the torque will be function of the q-axis current component only as follows:

$$T_e = \frac{3P}{2}\left(\lambda_{af} i_q\right) = k_t i_q$$

where k_t is the torque constant

Using the PM model represented by Eq. (1), the motor was operated under various values of the supply voltage and measuring the various corresponding values of the d-q axis stator currents i_d and i_q of the PM motor, the training data were obtained and used in training the neural networks using MATLAB/NN Package.

3. New Direct Adaptive robust State Feedback Controller.

The PM motor can take the form as shown below:

$$\begin{bmatrix} \dot{i}_d \\ \dot{i}_q \end{bmatrix} = \begin{bmatrix} \frac{-R}{L_d} & \omega_s \frac{L_q}{L_d} \\ -\omega_s \frac{L_d}{L_q} & \frac{-R}{L_q} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-\omega_s \lambda_{af}}{L_q} \end{bmatrix} \quad (2)$$

Considering the model of nonlinear uncertain system is represented by:

$$y^{(n)} = f(y \ldots y^{(n-1)}) + g(y \ldots y^{(n-1)})u + d \quad (3)$$

Where y is the system output, d models the disturbance, f and g are unknown continuous functions with g bounded away from zero. Without loss of generality, we assume that $g(x)$ is strictly positive.

Let $x = [y, \ldots\ldots, y(n-1)]$, then the system (2) can be represented in canonical controllable form as

$$\begin{aligned} \dot{x}_i &= Ax_i + b(f(x_i) + g(x_i)u + d) \\ y &= Cx_i \end{aligned} \quad (4)$$

Where $x_i = [x_1, x_2, x_3, \ldots]$, u and y are the state variables, system input and output, respectively, which are all assumed to be available for measurement; $f_i(x_i)$ and $g_i(x_i)$, $i = 1, \ldots, n$, are smooth nonlinear functions that contain both parametric and nonparametric uncertainties. $g_i(x_i)$ is usually referred to as the gain function.

The main control objective is to design a control input u that forces the system output y to track a given desired trajectory y_d that has bounded derivatives up to the n-th order.

Let the system tracking error $e = x - x_d$, then the tracking error dynamics can be expressed as:

$$\begin{aligned} \dot{e} &= Ae + b(y^{(n)} - y_d^{(n)}) \\ &= Ae + b(f(x) + g(x)u - y_d^{(n)} + d) \end{aligned} \quad (5)$$

Then, an appropriate control law can be presented [5-7]:

$$u = \frac{1}{\hat{g}(x)}\left[-\hat{f}(x) + y_d^{(n)} - ke\right] \quad (6)$$

Where $\hat{f}(x)$ and $\hat{g}(x)$ are approximations of $f(x)$ and $g(x)$ which will be obtained by neural network, $k^T = \left[k_n, k_{n-1}, \ldots, k\right]^T$ and k is selected such that the polynomial $e^{(n)} + k_1 e^{(n-1)} + k_2 e^{(n-2)} + \ldots + k_n e$ is Hurwitz.

The controller in (6) is composed of two terms; feed-farward term $-\hat{f}(x) + y_d^{(n)}$ to compensate for error and model uncertainties and linear term ke to stabilize the system.

By applying the control law (6) to the motor model (2) in the synchronous reference frame where v_d and v_q are the control inputs. Then, the control law in the d and q axis can be expressed as:

$$v_d = L_d\left[-v_{d(neural)} + \dot{i}_{dref} - ke_d\right]$$

and $v_q = L_q\left[-v_{q(neural)} + \dot{i}_{qref} - ke_q\right]$

where $e_d = i_{dref} - i_d$, $e_q = i_{qref} - i_q$, $v_{d(neural)}$ and $v_{q(neural)}$ are the outputs of the neural models as mentioned in section 2.

Then, the block diagram of direct adaptive state feedback controller implemented by MATLAB/SIMULINK package can be represented as shown in fig. 1.

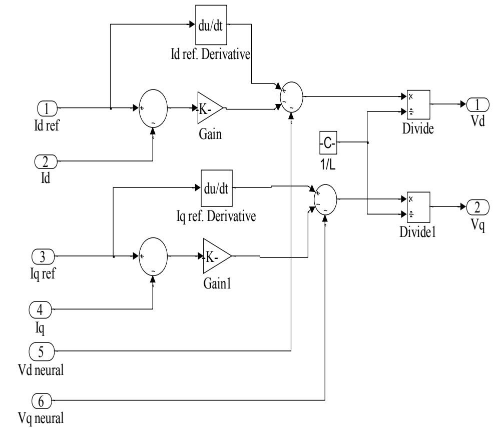

Fig. 1 Block diagram of direct adaptive state feedback controller using MATLAB/SIMULINK

4. Stability analysis of the controller.

Let the system tracking error $e=x-x_d$. For the convenience of stability analysis in the following steps, k is selected as: $k=\frac{1}{4}+k'+k^*$

Replacing the controller (5) in (4) results the following errors:

$$\dot{e}+ke=\hat{f}(x)-f(x)+\hat{g}(x)u-g(x)u=\hat{f}(x)-f(x) \qquad (6)$$

Where $g(x)$ in our case is a constant gain in the model (2) and the term $\hat{g}(x)u-g(x)=0$. Since the nonlinear model will be approximated and identified by neural network for example RBF NNs that can be expressed as following:

$$f(z)=\theta^T\xi(z)+\varepsilon$$

Where $\varepsilon(z)$ is the so-called NN functional approximation error; $z\in R^m$ is the input of NNs; $\theta=[\theta_1,\theta_2,.........\theta_l]^T\in R^l$ is the weight collection to be determined, l is the node number of RBF NNs; $\xi=[\xi_1,\xi_2,\xi_3,........,\xi_l]^T$ is the basis function vector. $\xi(z)$ is usually chosen as the Gaussian function:

$$\xi_i(z)=\exp\left(\frac{\|z-\mu_i\|_2^2}{\eta_i^2}\right), i=1,2,...l$$

Where $\|\cdot\|_2$ denotes the Euclidian norm, $\mu_i=[\mu_{i1},\mu_{i2},.........,\mu_{im}]^T$, $\mu_{ij}\in R^l$, $j=1,\ldots,m$, is the center of $\xi(z)$, and η_i is the width, which are usually selected according to the priori information about $f(z)$. Then, the equation (6) can be written as follows:

$\dot{e}+ke=\hat{f}(x)-f(x)=\theta^T\xi(x)-f(x)$ Where $\xi(x)$ is the basis function vector in approximating $f(x)$ and abbreviated to ξ and θ is the corresponding weight parameter.

Then, $\dot{e}+ke=\phi^T\xi-\sigma(x)$.

Where $\sigma(x)$ is the functional approximation error using the optimal parameters. $\phi=\theta-\theta^*$, θ^* denote the optimal parameters in approximating $f(x)$. Suppose Δ_f is the universal minimax functional approximation error in approximating $f(x)$ where $\Delta_f=\left\|f(x)-\xi^T\theta^*\right\|_\infty$.

Choosing a suitable Lyupanov function;

$V=\frac{1}{2}e^2+\frac{1}{2}\beta\phi^T\phi$, where β is a random positive constant. We propose the following adaptation law:

$\dot{\theta}=-\frac{\xi}{\beta}\left[\dot{e}+ke+\Delta_f\right]$, where $\dot{\theta}=\dot{\phi}$ then,

$$\dot{\phi}=-\frac{\xi}{\beta}\left[\phi^T\xi-\sigma+\Delta_f\right]$$

Differentiating the Lyupanov function as follows:

$$\dot{V}=e\dot{e}+\frac{1}{2}\phi\dot{\phi}=-ke^2+e\left[\phi^T\xi-\sigma\right]-\phi^T\xi\left[\phi^T\xi-\sigma+\Delta_f\right]$$

$$=-k'e^2-k^*e^2-\left(\frac{e^2}{4}-e\left(\phi^T\xi-\sigma\right)+\left(\phi^T\xi-\sigma\right)^2\right)$$

$$-\left(\phi^T\xi-\sigma\right)^2-\phi^T\xi\left[\phi^T\xi-\sigma+\Delta_f\right]$$

$$=-k'e^2-\left(\frac{e}{2}-\left(\phi^T\xi-\sigma\right)\right)^2-k^*e^2+\sigma^2-\phi^T\xi(\sigma+\Delta)$$

Since $|\sigma|\le|\Delta|$ if $\left|\phi^T\xi\right|\ge|\sigma|$, we have

$\left|\phi^T\xi(\sigma+\Delta)\right|\ge\sigma^2$ then, $\dot{V}<0$. On the other hand, if

$\left|\phi^T\xi\right|<|\sigma|$ then, $\left|\phi^T\xi(\sigma+\Delta)\right|<3|\sigma|^2$, Thus if

$k^*e^2\ge4\sigma^2$ then, $\dot{V}<0$

Since $|\sigma|$ is very small, k can be chosen as a large positive constant. Then, the system will converge to a very small neighborhood of the reference signal [14-15].

5. Back propagation Neural Network (BP NN).

BP Neural Network is a multilayer fully connected network. It consists of three layers; input layer, hidden layer and output layer. The number of neurons in the input layer is equal to the number of inputs. The neurons in the hidden layer are nonlinear where in the output layer are linear ones. The number of hidden neurons is sufficient to achieve better accuracy of the neural network.

The modeling of PM motor was carried out by taking various values of the supply voltages and measuring the stator currents at certain value of load torque. The input pattern of the neural network in the proposed method is time delayed series of the stator currents in the d-q axes;

i_d and i_q and the output of the network are the *d-q* axis stator voltages; v_d and v_q. As shown in fig. 2, by off line training, the NN can be adjusted with the hidden layer containing 10 neurons with a tanh activation function and an offset, whereas the output layer contains two neurons with a linear activation function (Purelin. function). The network was trained with the modified Levenberg-Marquardt backpropagation algorithm given by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T\mathbf{J} + \mu\mathbf{I}]^{-1}\mathbf{J}^T\mathbf{e}$$

Where **J** is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, **e** is a vector of network errors, **I** is the identity matrix and μ is a scalar constant.

This method of training is more suitable for BP neural network, faster than Gauss-Newton method or variable learning rate, required few training epochs and took only a few seconds to achieve the goal using the MATLAB/NN toolbox. The transfer function of the k-th neuron (Tansig.) in the hidden layer is

$$V_k = \frac{1}{1+e^{-z}}$$

The adaptation learning rule (learnGDM) calculates the weight change dW for a given neuron from the neuron's input P and error E, the weight (or bias) W, learning rate lr, and momentum constant mc,

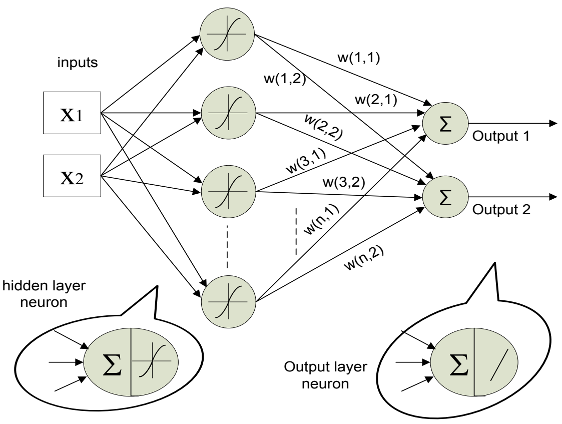

Fig. 2 The structure of back propagation (BP) neural network

according to Gradient Descent with Momentum:
*dW = mc*dWprev + (1-mc)*lr*gW*

The previous weight change *dWprev* is stored and read from the learning state LS. The structure of BP neural network is shown in fig. 2.

6. Radial Basis Function Neural Network (RBF NN).

The RBF neural network consists also of three layers. But in this work, it is consists of two layers hidden layer and output layer. The input data to the hidden layer is the Euclidean distance of the input and the output is the radial basis function of the neuron input. It implements a gaussain function in the hidden layer. The mapping between the input and the output of the hidden layer is nonlinear and between the hidden layer and the output layer is linear. A partial persistency of excitation (PE) condition can be satisfied by local radial basis functions for any periodic trajectory. Then, the exponential stability can be achieved and also accurate approximation of the NN in a local region along the periodic or recurrent trajectory [17-19].

While RBF network model has the advantage of much less training time as compared to the back propagation network model, it suffers from the drawback of requiring more number of neurons than the BP network model for the same error goal, Compared to BP NN, RBF NN has faster rate convergence and better learning capability without local minima. The output of the hidden neuron can be expressed as:

$$\xi_i(z) = e^{\left(-\left\|z-\mu_k\right\|/\eta_k^{\,2}\right)}$$

μ_k is the center of *k-th hidden neuron*, and η_k is the width, which are usually selected according to the priori information about *f(z)*. The structure of BP neural network is shown in fig. 3.

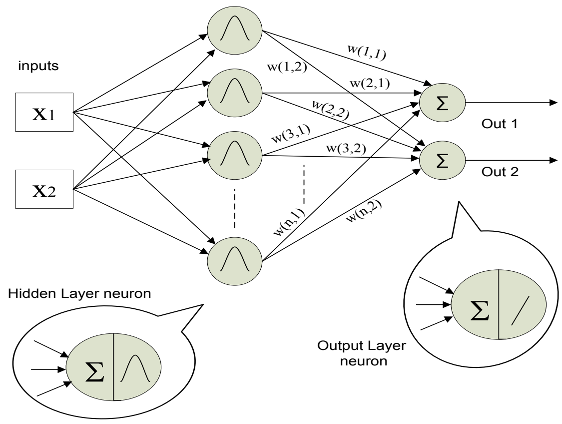

Fig. 3 The structure of radial basis function (RBF) neural network

7. Simulation Results

In order to validate the effectiveness of the proposed approach, we investigated it by MATLAB/SIMULINK and used the following PMSM parameters as follows: $L_d = L_q = 0.012$ H, $R = 2.875\ \Omega$, $J = 8 \times 10{-}4$ kg.m^2, B= 0.001 N.m.s/rad, k_t = 1.15 N.m/A, P = 4, and λ_{af} = 0.19167 V.s.

By setting the reference d-axis current i_{dref}=0 to achieve the maximum torque per ampere and obtaining the reference torque from the speed controller which employs conventional PI controller, the block diagram of the proposed system with the new proposed control method can be implemented as depicted in fig. 4.

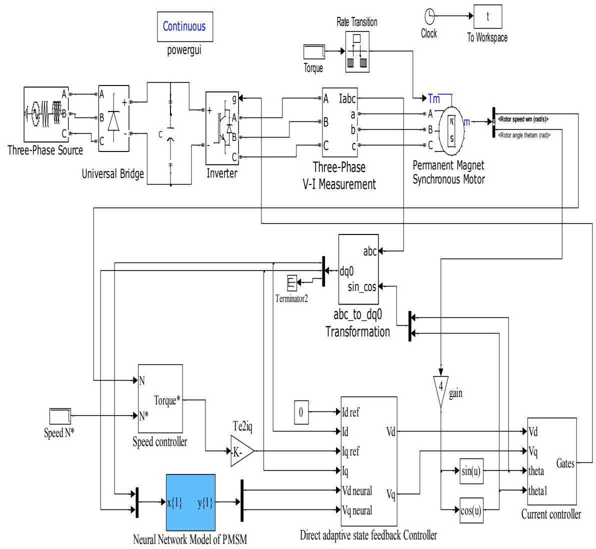

Fig. 4 Block diagram of the proposed method using MATLAB/SIMULINK

To study the robust performance of the proposed method, many tests were carried out at various operating conditions.

7.1 Comparison of step up response for RBF and BP NN's

Fig. 5 shows the start up speed response of the proposed method with RBF and BP neural networks. It is evident that RBF neural network has faster response than BP NN with the proposed method.

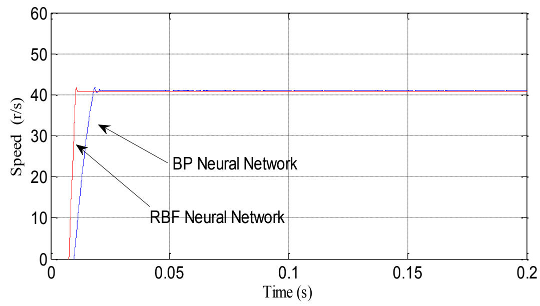

Fig. 5 The speed response of the proposed method with RBF and BP neural networks

Fig. 6 depicts the start up torque response of the proposed method with RBF and BP neural networks. It is clear the torque of the proposed method with RBF neural network settles at 13.5 ms while it settles at 21.5 ms for the proposed method with BP neural network. Also, the torque has minimum ripples with the proposed method.

Figs. 7 and 8 display the stator currents for the proposed method with RBF and BP neural networks where the two types of neural networks achieve ripples free stator currents.

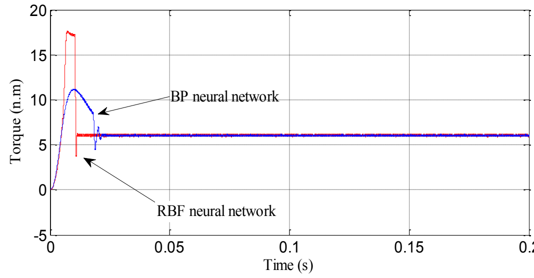

Fig. 6 The torque response of the proposed method with RBF and BP neural networks

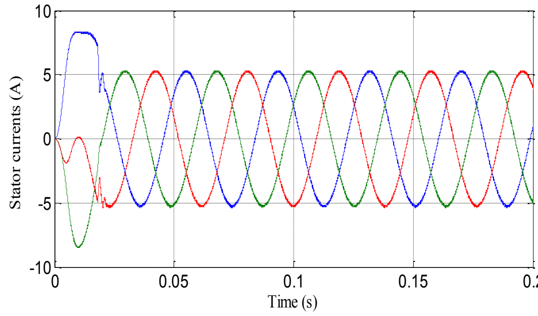

Fig. 7 The stator currents of PMSM for the proposed method with BP neural NN

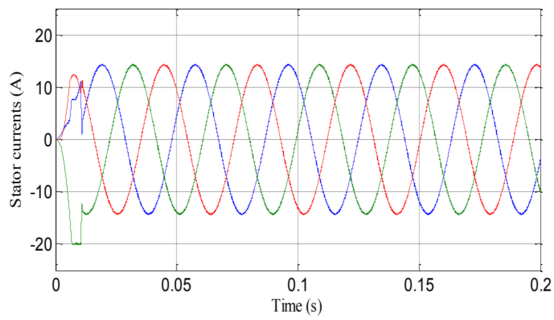

Fig. 8 The stator currents of PMSM for the proposed method with RBF NN

7.2 The proposed method with RBF NN

7.2.1 The effect of changing the load torque

Fig. 9 displays the dynamic torque response of the proposed method with RBF NN tested with load torque (1 n.m) added at t= 0.1 to the starting load torque (T_L=6 n.m) and removing it at t= 0.12. Fig. 10 shows the torque response for removing the load completely at t=0.15 s.

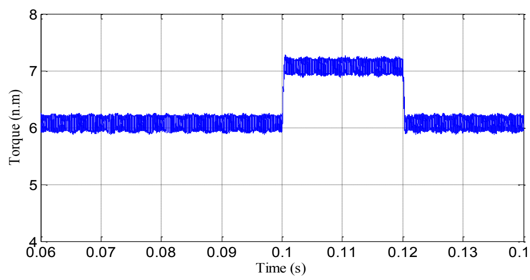

Fig. 9 The step up and down torque response of the proposed method with RBF NN at t= 0.1s

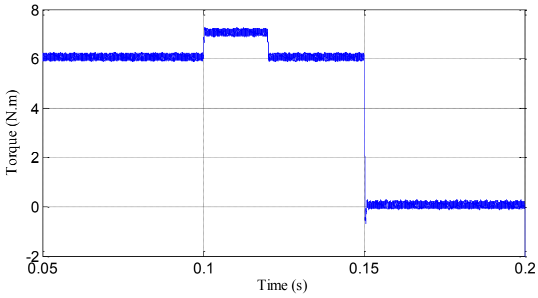

Fig. 10 The torque response of the proposed method with RBF NN for removing the load completely at t= 0.15s

As shown in figs. 9-11, It was clear that the torque changes softly at step up, step down and very small undershoot when completely removing the load torque at t= 0.15 s.

Fig. 11 shows that the speed can track the reference speed without steady state arror for step up load torque at t= 0.1, step down load torque at t= 0.12 and completely removing it at t=0.15 s.

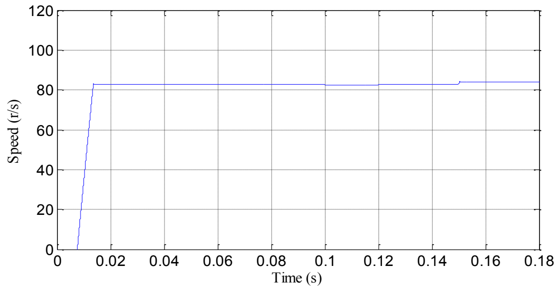

Fig. 11 The speed response of the proposed method with RBF NN for step up and step down load torque

7.2.2 The effect of changing the reference rotor speed

Fig. 12 displays the reference speed (80 r/s) and decreasing it to 40 r/s at t= 0.1 s and setting it to zero at t= 0.12 to show the torque response as shown in fig. 13. As seen in fig. 13, the torque maintains its reference value (4 n.m) except large undershoot at the instants of changing the rotor speed or setting it to zero.

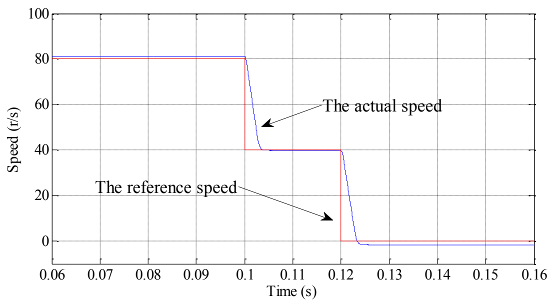

Fig. 12 The step down speed response and braking the motor at t= 0.1 s and t=0.12 respectively

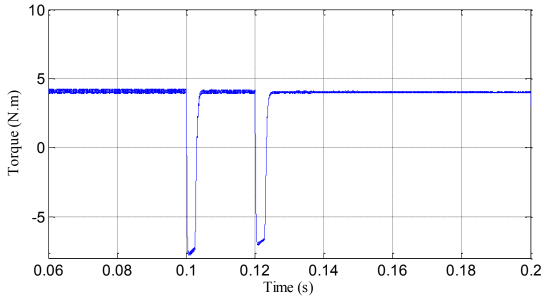

Fig. 13 The torque response for changing the rotor speed

7.3 The proposed method with BP NN

7.3.1 The effect of changing the load torque

The proposed method with PB NN was tested at t= 0.1 s by adding load torque (1 n.m), removed at t=0.12 s and completely removing the load at t=0.15s. Fig. 14 shows a small overshoot of the torque response at the instants of adding the load (1 n.m) and removing it and larger overshoot compared to the RBF NN when completely removing the load at t=0.15 as shown in fig. 15.

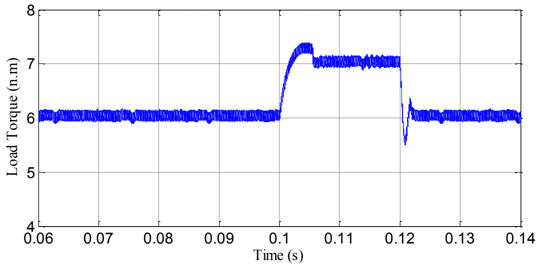

Fig. 14 The step torque response of the proposed method with PB NN with step up load at t= 0.1s and step down load at t=0.12.

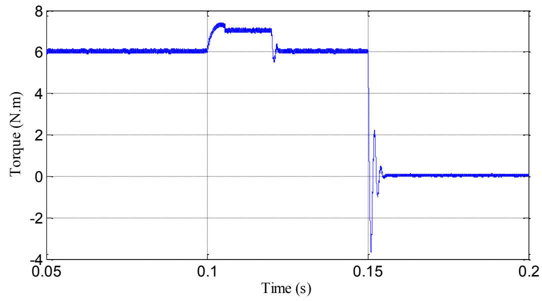

Fig. 15 The torque response of the proposed method with PB NN for removing the load completely at t= 0.15 s.

Fig. 16 depicts the ability of the PMSM drive to recover to the reference speed after stepping down the load at t= 0.12 with small undershoot or completely removing it at t= 0.15 s with small overshoot within 5 ms.

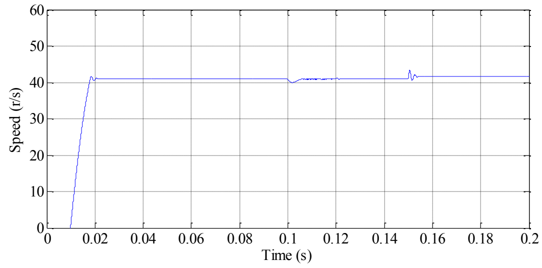

Fig. 16 The speed response of the proposed method with PB NN for step load torque at t=0.1 s

7.3.2 The effect of changing the reference rotor speed

As displayed in fig. 17 the speed has large under shoot at the instants of stepping down the speed to 20 r/s (t=0.1 s) and setting it to zero at t=0.14 s.

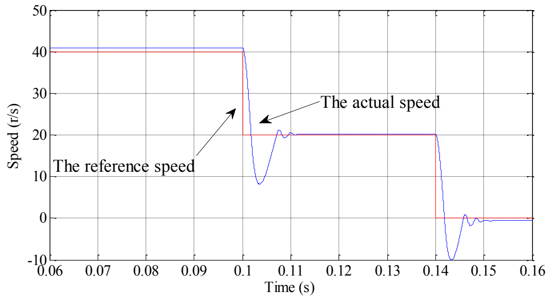

Fig. 17 The step down speed response and braking the motor at t= 0.1 s and t=0.14 respectively

Fig. 18 displays the torque response has large overshoots at the instants of stepping down the speed at t=0.1 s and setting it to zero at t=0.14 s.

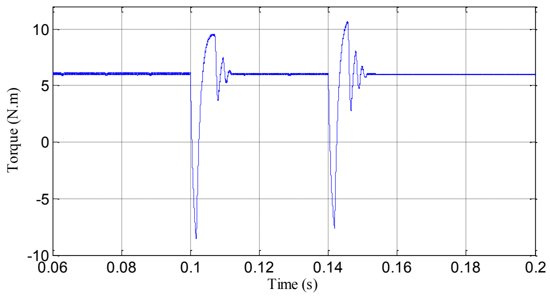

Fig. 18 The torque response for changing the rotor speed

8. Conclusion

A new method of direct adaptive control is proposed for PMSM. The method employed state feedback controller for the uncertain model of PMSM. The controller used two famous types of neural networks to approximate the uncertain nonlinear model of the permanent magnet synchronous motor. The method was tested and validated by MATLAB/SIMULINK Package. The proposed method achives higher performance for the motor with faster speed and torque response and minimized torque ripples without prior knowledge of the motor parameters and independent of the motor model. The RBF neural network provides better performance than BP neural network as it gives faster torque, speed response and more reliable performance for speed variations or load disturbance.

References

1. Khodabakhsh A. N. and Soltani J.: *MTPA Control of Mechanical Sensorless IPMSM Based on Adaptive Nonlinear Control*. In: ISA Transactions, Vol. 61, 2016, pp. 348-356.
2. Karabacak M., Eskikurt H. I.: *Design, Modelling and Simulation of a New Nonlinear and Full Adaptive Backstepping Speed Tracking Controller for Uncertain PMSM*. In: Applied Mathematical Modeling, Vol. 36, No. 11, 2012, pp. 5199-5213.
3. Yu J., Shi P., Dong W., Chen B. and Lin C.: *Neural Network-Based Adaptive Dynamic Surface Control for Permanent Magnet Synchronous Motors*. In: IEEE Transactions on Neural Networks and Learning Systems, Vol. 26, No. 3, March 2015, pp. 640-645.
4. Chiang S. J. and Lin C.-H.: *Adaptive Backstepping RFNN Control for Synchronous Reluctance Motor Drive*. In: IEEE 2006 37th Power Electronics Specialist Conference (PESC), pp. 1-6.
5. Khlaief A., Boussak M., Châari A.: *A MRAS-Based Stator Resistance and Speed Estimation for Sensorless Vector Controlled IPMSM Drive*. In: Electric Power Systems Research, Vol. 108, No. 1, 2014, pp. 1– 15.
6. Maiti S., Chakraborty C., Sengupta S.: *Simulation Studies on Model Reference Adaptive Controller Based Speed Estimation Technique for the Vector Controlled Permanent Magnet Synchronous Motor Drive*. In: Simulation Modelling Practice and Theory, Vol. 17, No. 4, 2009, pp. 585-596.
7. Rigatos G. G.: *Particle and Kalman filtering for state estimation and control of DC motors*. In: ISA Transactions, Vol. 48, No. 1, 2009, pp. 62-72.
8. Sivanandakumar D. and Ramakrishnan, K.: *LMI Based Digital State Feedback Controller for a Wound Rotor Induction Drive with Guaranteed Closed Loop Stability*. In: IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), Dec. 12-15, 2006.
9. Cheng P.-T., Chen J.-M. and Ni C.-L.: Design of State Feedback Controller for Series Voltage Sag Compensators. In: IEEE Transactions on Industry Applications, Vol. 45, No. 1, 2009, pp. 260-267.
10. Niewiara L., Tarczewski T. and Grzesiak L. M.: *Application of Artificial Neural Network State Feedback Controller to Torque Ripple Minimization of PMSM*. In: IEEE 2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Austria, Sept. 1-3, 2014, Vol. 1, pp. 363-369.
11. Grzesiak L. M. and Ufnalski B.: *Neural-Network Based Programmable State Feedback Controller for Induction Motor Drive*. In: IEEE 2006 International Joint Conference on Neural Networks, Canada, July 16-21, 2006, pp. 1091-1097.
12. Hasanein H. M.: *Torque Ripple Minimization of Permanent Magnet Synchronous Motor Using Digital Observer Controller*. In: Energy Conversion and Mangement, Vol. 51, 2010, pp. 98-104.
13. Elsayed M. T., Mahgoub O. A. and Zaid S. A.: *MTPA of SPMSM-DSP Based*. In: Proceedings of the IEEE 15th Middle East Power System Conference (MEPCON'12), 2012, Alex., Egypt, Paper ID 160.
14. Shi H.: *A Novel Scheme for the Design of Backstepping*

Control for a Class of Nonlinear Systems. In: Applied Mathematical Modeling, Vol. 35, No. 4, 2011, pp. 1893-1903.
15. Lian J., Lee Y., Sudhoff S. D. and Zak S. H.: *Variable Structure Neural Network Based Direct Adaptive Robust Control of Uncertain Systems*. In: Proceedings of IEEE 2008 American Control Conference, Seattle, Washington, USA, Jun. 11-13, 2008, pp. 3402-3407.
16. Yesildirekt A. and F. L. Lewis F. L.: *Feedback Linearization Using Neural Networks*. In: Automatica, Vol. 31, No. 11, Nov. 1995, pp. 1659-1664.
17. Park J. W., Harley R. G. and Ganesh K. Venayagamoorthy G. K.: *Indirect Adaptive Control for Synchronous Generator: Comparison of MLP/RBF Neural Networks Approach With Lyapunov Stability Analysis*. In: IEEE Transactions on Neural Networks, Vol. 15, No. 2, March 2004, pp. 460-464.
18. Li Y., Qiang S., Zhuang X. and Okyay Kaynak: *Robust and Adaptive Backstepping Control for Nonlinear Systems Using RBF Neural Networks*. In: IEEE Transactions on Neural Networks, Vol. 15, No. 3, May 2004, pp. 693-701.
19. Wang C. and Hill D. J.: *Learning From Neural Control*. In: IEEE Transactions on Neural Networks, Vol. 17, No. 1, Jan. 2006, pp. 130-146.