

NEURAL NETWORK BASED SYSTEM IDENTIFICATION OF A PMSM UNDER LOAD FLUCTUATION

MODELAMIENTO BASADO EN REDES NEURONALES DE UN PMSM BAJO FLUCTUACIONES DE CARGA

JABID QUIROGA

Profesor asociado, Universidad Industrial de Santander, Bucaramanga- Colombia, jabib@uis.edu.co

DAVID CARTES

Profesor asociado, Ingeniería Mecánica, Florida State University, Tallahassee, United States, e-mail:dave@eng.fsu.edu

CHRIS EDRINGTON

Profesor, Ingeniería Eléctrica y Computacional, Florida State University, Tallahassee, United States, edrinch@eng.fsu.edu

Recibido para revisar julio 25 de 2008, aceptado mayo 21 de 2009, versión final junio 19 de 2009

RESUMEN: La técnica de redes neuronales es usada para modelar un PMSM. Una red recurrente multicapas predice el componente fundamental de la señal de corriente un paso adelante usando como entradas el componente fundamental de las señales de voltaje y la velocidad del motor. El modelo propuesto de PMSM puede ser implementado en un sistema de monitoreo de la condición del equipo para realizar labores de detección de fallas, evaluación de su integridad o del proceso de envejecimiento de éste. El modelo se valida usando un banco de pruebas para PMSM de 15 hp. El sistema de adquisición de datos es desarrollado usando Matlab®/Simulink® con dSpace® como interfase con el hardware. El modelo mostró capacidades de generalización y un desempeño satisfactorio en la determinación de las componentes fundamentales de las corrientes en tiempo real bajo condiciones de no carga y fluctuaciones de esta.

PALABRAS CLAVE: Identificación de Sistemas, PMSM, Redes Neuronales, Redes Recurrentes.

ABSTRACT: A neural network based approach is applied to model a PMSM. A multilayer recurrent network provides a near term fundamental current prediction using as an input the fundamental components of the voltage signals and the speed. The PMSM model proposed can be implemented in a condition based maintenance to perform fault detection, integrity assessment and aging process. The model is validated using a 15 hp PMSM experimental setup. The acquisition system is developed using Matlab®/Simulink® with dSpace® as an interface to the hardware, i.e. PMSM drive system. The model shows generalization capabilities and a satisfactory performance in the fundamental current determination on line under no load and load fluctuations.

KEYWORDS: System, Identification, PMSM, Neural Network, Recurrent Networks.

1. INTRODUCTION

The number of applications of Permanent Magnet Synchronous Machines (PMSM) is steadily increasing as a result of the advantages attributed to this type of motor. PMSMs are found in power and positioning applications such as ship propulsion systems, robotics, machine tools, etc. The main reason PMSM is so

attractive is due to its physical construction, which consist of permanent magnets mounted onto the rotor. This arrangement improves the efficiency and performance. PMSM presents several advantages compared with the induction motor, the most popular electromechanical actuator, such as: high power density, high air-gap flux density, high-torque/inertia ratio, low package weight, less copper losses, high

efficiency, and a small rotor for the same power output.

Many real-world applications, such as adaptive control, adaptive filtering, adaptive prediction and Fault Detection and Diagnosis systems (FDD) require a model of the system to be available online while the system is in operation. The NN based PMSM model proposed in this study can be implemented particularly as a component of a FDD model based system to monitoring the electric condition of a PMSM and evaluate motor aging.

The basic idea of a model based FDD is to compare measurements with computationally obtained values of the corresponding variables, from which residual signals can be constructed. The residuals provide the information to detect the fault.

In fact, in a FDD system the residuals are generated by the comparison between a computational variable multiple time steps ahead (MSP) into the future with the present value of the variable. This time ahead is required to consider the computational time spent by the model to produce on line the signal to be compared. MSP is performed using a recursive approach based on a dynamic recurrent neural network [1]. This recursive approach is followed in this study and it is one of the advantages of the neural networks compared with other techniques such as support vector machines which training is a batch algorithm and does not exist a recursive algorithm [2].

The modeling process for complex systems such as PMSM under load fluctuation demands methods which deal with high dimensionality, nonlinearity, and uncertainty. Therefore, alternative techniques to traditional linear and nonlinear modeling methods are needed.

System identification is an experimental approach for determining the dynamic of a system from measured input/output data sets. It includes: experimental data sets, a particular model structure, the estimation of the model parameters and finally the validation of the identified model. A complete system

identification process must cover the items mentioned above [3].

One such approach is Neural Networks (NN) modeling. NN are powerful empirical modeling tools that can be trained to represent complex multi-input multi-output nonlinear systems. NN have many advantageous features including parallel and distributed processing and an efficient non-linear mapping between inputs and outputs [4].

NN have been also used in control applications. In [4-6] A multilayer feedforward artificial neural networks speed PID controller for a PMSM are presented. In [4] on-line NN self tuning is developed and the NN is integrated with the vector control scheme of the PMSM drive. In [6-8] an on-line adaptive NN based vector control of a PMSM are proposed. In this application the NN play both roles, system identification and speed control.

Various types of NN structures have been used for modeling dynamic systems. Multilayers NN are universal approximator and have been utilized to provide an input-output representation of complex systems. Among the available multilayer NN architectures the recurrent network has shown to be more robust than a plain feedforward network when taking the accumulative error into account [8].

In [1] is proposed to use a dynamic recurrent network in the form of an IIR filter as a multi-step predictor for complex systems. Present and delayed observations of the measured system inputs and outputs, are utilized as inputs to the network. The proposed architecture includes local and global feedback.

Some NN structures for modeling electrical motors have been proposed previously. Most of them have been focused in modeling induction machines. In [9] a NN model of an induction motor based on a NARX structures is used to simulate the speed using as input the voltage signal. In [10] a NN model for simulating the three phase currents in an induction motor is proposed based on multilayer recurrent NN; however, the load fluctuation is not addressed.

In this paper, a near term fundamental current predictor of a PMSM is proposed using a recurrent (global and local feedback) multilayer network with delayed connections of the voltage and the speed signals as inputs. This architecture provides to the neural network the ability to capture the complex dynamic associated to the operation of the PMSM under load fluctuation.

2. METHODOLOGY

2.1 NN for System Identification

In the last decade there has been a growing interest in identification methods based on neural networks [11]. The recent success of dynamic recurrent neural networks as semiparametric approximators for modeling highly complex systems offers the potential for broadening the industrial acceptance of model-based system identification methods [12]. Neural networks are universal approximators in that a sufficiently large network can implement any function to any desired degree of accuracy. By presenting a network with samples from a complex system and training it to output subsequent values, the network can be trained to approximate the dynamics, which underlie the system. The network, once trained, can then be used to generalize and predict states that it has not been exposed to.

The use of NN as a modeling tool involves some issues such as: NN architecture, the number of neurons and layers, the activation functions, the appropriate training data set and the suitable learning algorithm.

Recurrent networks are multilayer networks which have at least one delayed feedback loop. This means an output of a layer feeds back to any proceeding layer. In addition, some recurrent networks have delays inputs (Recurrent dynamic network). These delays give the network partial memory due to the fact that the hidden layers and the input layer receive data at time t but also at time $t-p$, where p is the number of delayed samples. This makes recurrent networks

powerful in approximating functions depending on time.

From the computational point of view, a dynamic neural structure that contains feedback may provide more computational advantages than a static neural structure, which contains only a feedforward neural structure. In general, a small feedback system is equivalent to a large and possibly infinite feedforward system [11]. A well-known example is an infinite *order finite impulse response* (FIR) filter is required to emulate a single-pole *infinite impulse response* (IIR).

A. The effect of load change on a Synchronous Motor

If a load is attached to the shaft of a synchronous motor, the motor will develop enough torque to keep the motor and its load turning at a synchronous speed. If the load on the shaft is increased, the rotor will initially slow down and the induced torque increases. The increase in induced torque eventually speeds the rotor back up, and the motor again turns at synchronous speed but with a large torque. Figure 1 shows the behavior of the speed under a load fluctuation in the PMSM used in this study. The load is applied using a ramp of 2 seconds from no load condition to 20% of the rated torque and a ramp down of 2 seconds from 20% of the rated torque to no load condition. It should be noted, the settling time of the speed control implemented in the motor is around 2 seconds. This value is of great importance in the determination of the training set for the training of the neural network. The raise in the torque at constant speed yields the increasing of the input power to the machine via current increase.

2.2 Neural Network Model Development

Because of the complexity of the dynamic behavior in the PMSM under load fluctuation and the difficulty associated in establishing an exact mathematical formulation to develop an explicit model of the PMSM with conventional methods, a nonlinear empirical model using a NN is developed. In this paper, it is proposed to utilize a multi-layer dynamic recurrent NN with local feedback of the hidden nodes and global feedback, as shown in Figure 2. Local feedback

implies use of delayed hidden node outputs as hidden node inputs, whereas global feedback is produced by the connection of delayed networks outputs as network inputs. This architecture provides a network in the form of a nonlinear infinite impulse response (IIR) filter.

The operation of a recurrent NN predictor that employs global feedback can be represented as (1):

$$\hat{y} = \Phi \left(\begin{array}{c} u(k), u(k-1), u(k-2), \dots, u(k-p) \\ \hat{y}(k-1), \dots, \hat{y}(k-q), W \end{array} \right) \quad (1)$$

where $\Phi(\bullet)$ represents the nonlinear mapping of the NN, u is the inputs, \hat{y} is the simulated values and W is the parameters associated to the NN.

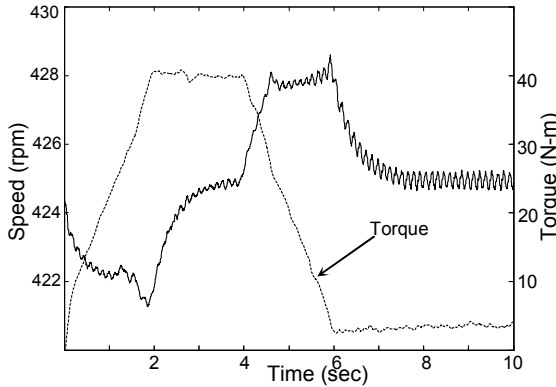


Figure 1. Effect of the load in the speed in PMSM (Torque=20% of rated value)

This NN architecture provides the capability to predict the output several steps into the future without the availability of actual outputs. Empirical models with predictive capabilities are desirable in fault monitoring and diagnosis applications. The implemented NN consists of an input layer, a hidden layer, and an output layer. Each of the processing elements of a MLP network is governed by (2).

$$x_{[l,i]} = \sigma_{[l,i]} \left(\sum_{j=1}^{N_{[l-1]}} w_{[l-1,j][l,i]} x_{[l-1,j]} + b_{[l,i]} \right) \quad (2)$$

for $i = 1, \dots, N_{[l]}$ (the node index), and $l = 1, \dots, l$ (the layer index), where $x_{[l,i]}$ is the i^{th} node output

of the l^{th} layer, $b_{[l,i]}$ is the bias, and $\sigma_{[l,i]}(\bullet)$ is the activation function of the i^{th} node in the l^{th} layer. The relationship between inputs and outputs in a multilayer NN can be expressed using a general nonlinear input-output model, (3):

$$\hat{y}(k, W) = f(u(k); W) \quad (3)$$

where W is the weight matrix determined by the learning algorithm, $f(\cdot)$ represents the nonlinear mapping of the vector input using any activation function. In this study, the *tansig* function is used in the hidden layer and *purelin* is used in the output layer. The input vector is defined as:

$$U(t) = \begin{bmatrix} V_f^{NS}(t), V_f^{NS}(t-1), V_f^{NS}(t-2), \\ v^{NS}(t-1), v^{NS}(t-2), \hat{I}_f^{NS}(t-1), \\ \hat{I}_f^{NS}(t-2) \end{bmatrix} \quad (4)$$

where NS represents a non-stationary signal, V_f^{NS} are the actual normalized values of the 3 phases line voltages:

$$V_f^{NS} = [V_{ab_f}^{NS}, V_{bc_f}^{NS}, V_{ca_f}^{NS}] \quad (5)$$

The normalized values of currents and voltages are obtained through the relation between the present current and voltage data and the maximum values of current and voltage respectively. The limit values of current and voltage are getting from the PMSM data sheet used in the test bench. The vector \hat{I}_f^{NS} is the three normalized predicted phase currents:

$$\hat{I}_f^{NS} = [\hat{I}_{a_f}^{NS}, \hat{I}_{b_f}^{NS}, \hat{I}_{c_f}^{NS}] \quad (6)$$

The variable v^{NS} is the normalized rotational velocity of the rotor with respect to the maximum values indicated in the data sheet. The hidden layer is composed of 6 neurons with delayed local feedback employed in each neuron. The hidden layer node number is chosen considering the balance of accuracy and network

size. The output layer, with global feedback, has 3 nodes, which correspond to the three phase-current predictions as shown in (7).

$$\hat{y}(t+1|t) = [\hat{I}_f^{NS}(t+1|t)] \quad (7)$$

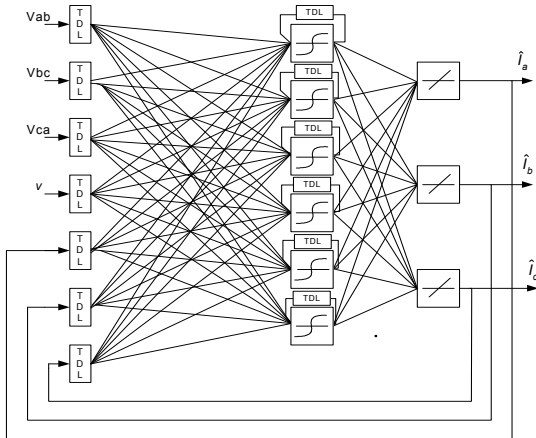


Figure 2. General structure of multilayer NN

2.3 Model Training and Validation

Generally, training recurrent dynamic networks is computationally intensive and in this work has been difficult due to the time dependencies present in their architectures. Recurrent networks exhibit complex error surfaces characterized by very narrow valleys which bottoms are often cusps. Additionally, initial conditions assigned in the training stage and variations in the input sequence can produce spurious valleys in the error surface [14].

The goal of the NN training is to produce a network, which yield small errors on the training set, but which will also respond properly to novel inputs (regulation). Therefore, in order to provide appropriate training of the model consideration of issues such as: regulation, initial values of the parameters, as well as the need to train the NN several times, must be addressed in order to achieve optimal results.

The neural network model proposed is trained using Bayesian regulation conveniently implemented within the framework of the Levenberg-Marquardt algorithm.

Regulation is used to avoid an over fitted network and to produce a network that

generalizes well [15]. This approach constrains the size of the network weights, adding a penalty term proportional to the sum of the squares of the weights (*msw*) and biases to the performance function. As can be seen, the objective function becomes a maximum penalty likelihood estimation procedure as shown in (8).

$$msereg = \beta mse + \alpha msw \quad (8)$$

where α and β are objective function parameters. This approach provides to the neural network a smooth response. The values of α and β determine the response of the NN. When $\alpha \ll \beta$, over fitting of the NN occurs. If $\alpha \gg \beta$ the NN does not adequately fit the training data. In [16] one approach is proposed to determine the optimal regulation parameter based on a Bayesian framework. In this framework, the weights and biases of the network are assumed to be random variables with specified distributions. The regularization parameters are related to the unknown variances associated with these distributions. These parameters can be estimated using statistical techniques.

The Levenberg-Marquardt algorithm is a variation of Newton's method and was designed for minimizing functions that are sums of squares of other nonlinear functions [17]. The algorithm speeds up the training by employing an approximation of the Hessian matrix (9). The gradient is computed via (10),

$$H = J^T J \quad (9)$$

$$g = J^T e \quad (10)$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors.

The Jacobian matrix is much less complex than computing the Hessian matrix. The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (11)$$

where x_k is a vector of current weights and biases. The Levenberg-Marquardt algorithm is an accommodative approach between Gauss-Newton's method (faster and more accurate near an error minimum) and gradient descent method (guaranteed convergence) based on the adaptive value of μ . If scalar μ is zero then the update process described by (11) resembles Newton's method using the approximate Hessian matrix. If μ is large, then this process becomes gradient descent method with a small step size.

A complete description of the Levenberg-Marquardt Backpropagation (LMBP) algorithm can be found in [15]. A detailed discussion about the implementation of Bayesian Regulation in combination with Levenberg-Marquardt training is presented in [15].

During the NN model training each layer's weights and biases are initializing according to the method proposed for Nguyen and Widrow in [18]. This method for setting the initial weights of hidden layers of a multilayer neural network provides a considerable reduction in training time. Using the Nguyen and Widrow initialization algorithm, the values of weights and bias are assigned at the beginning of the training, then the network is trained and each hidden neuron still has the freedom to adjust its own values during the training process.

The proposed NN is trained offline using the collected values at 625 Hz of sampling frequency followed by scaling in the range of [-1:1] of the magnitude of the fundamental components of the voltage and the rotational velocity as inputs. The targets are the normalized values of the magnitudes of the fundamental components of the three phase currents. The training data set consists of 5487 samples; the number of parameters to be calculated (weights and biases) during the training stage is 579 for the neural network proposed.

Tests in the lab showed a dependency between the percentages of the rated load applied using a ramp and the speed settling time. Particularly, for a larger load the settling time is also bigger. PMSM speed settling time plays an important role in the dynamic behavior of the currents. It is observed multiple current values for a unique

value of speed when the load is applied in a ramp using a time below the settling time.

Furthermore, longer time involves more number of samples to take in a loading process of the motor. In addition, the memory requirement of the Levenberg-Marquardt algorithm is relatively large because LM uses the Jacobian matrix which in the case of the network implemented has dimensions of $Q \times n$, where Q is the number of the training sets and n is the number of weights and biases. This large matrix restricts the size of the training set due to limitations in processing capacity of the test bench.

Because the reasons explained above the training set is chosen under size consideration. The training data set is comprised of measurements taken by loading the motor between no-load and 30% of the rated torque applied by ramping for 2 seconds, followed by 2 seconds of constant 30 % of rated torque and then a ramp down for 2 seconds to no-load, as shown in Figure 3. Currently, the torque ramp up and down configuration has been implemented in synchronous machines which startup is executed under torque control. This arrangement has the benefit that the mechanical starting behavior of the equipment driven by the motor will be much softer than when using a step torque for starting and stopping. Additionally the torque ramp is chosen to protect the test bench. Test showed a mechanical impact in the PMSM produced by the quick change between no load and load condition when the torque step is applied.

The criterion established for checking if the network has been trained for a sufficient number of iterations to ensure convergence is obtained when the values of the Sum Squared Error (*SSE*) is low and is relatively constant over at least ten epochs. High variation in this term after each iteration is a clear sign of an unstable convergence. Additionally the training algorithm used provides a measure of how many network parameters (weights and biases) out of the total are being effectively used by the network. This effective number of parameters should remain approximately the same, no matter how large the number of parameters in the network becomes. (This assumes that the network has been trained for a sufficient number of iterations to ensure convergence.) In the case of the performed

training, the neural network achieved convergence when the value of SSE was approximately 1. The use of regulation avoids the need to use a validation data set.

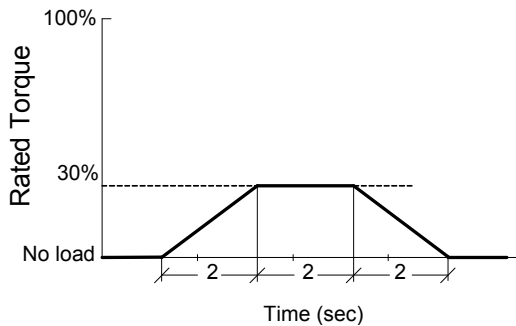


Figure 3. Load applied to obtain the training set

2.4 Experimental Approach

In the proposed system identification system, the data acquisition system allows the sampling of $V^{NS}(t)$, $I_f^{NS}(t)$ and $v(t)$. The signals are sampled at 625 Hz and the voltage and current signals are filtered using bandpass filters to obtain the fundamental components of each phase i.e. $V_f^{NS}(t)$ and $I_f^{NS}(t)$.

The proposed neural network model is experimentally validated using a system which consists of a 28.8 kVA variable frequency drive connected to an 11.25 kW, 640 V, 60 Hz, Y-connected 8-pole PMSM. A dc motor is mechanically coupled to the PMSM to serve as a load (see Figure 4).

During the experiments, the load is changed by varying the armature resistance of the dc motor, in order to emulate a load fluctuation condition, e.g. increasing or decreasing the load from 0% up to 45%.

The system is developed using MATLAB[®]/Simulink with dSPACE[®] as an interface to the data acquisition hardware and PMSM drive system. The fully developed model is applied to the electrical system and performance can be studied in dSPACE[®], which is used to display and record the line voltages, line currents, predicted values of current and the torque signal.

3. EXPERIMENTAL RESULTS

A series of tests are designed to demonstrate the robustness and performance using the proposed

system covering a wide variety of operating conditions at different load levels.

In testing the performance of the developed network, the normalized mean square error and the absolute mean error are used. The testing data set comprises of measurements obtained from no load to 10%, 20%, 30%, 40%, and 45% of the rated torque respectively, which are entirely different than the ones used in the training data set in order to evaluate the generalization performance of the network. In addition, a series of tests are performed for 45% of the rated torque using different ramp slopes to introduce the load to the PMSM. Although the ramps are different to the ones used in the training stage, all of them are configured to produce ramps in a greater time to the settling time. The results are summarized in Tables I and II in terms of MSE (mean squared error) and $Mean$ error. Tables I and II demonstrate the generalization performance of the network up to 45% of the rated torque.

In Figure 5 is shown the efficacy of the model implemented in tracking the variations of the current in phase A, when the load coupled to the motor is changing. Figure 5 shows the actual value of current I_a , the simulated value of the current I_{asim} , the variation in the torque and the residuals.

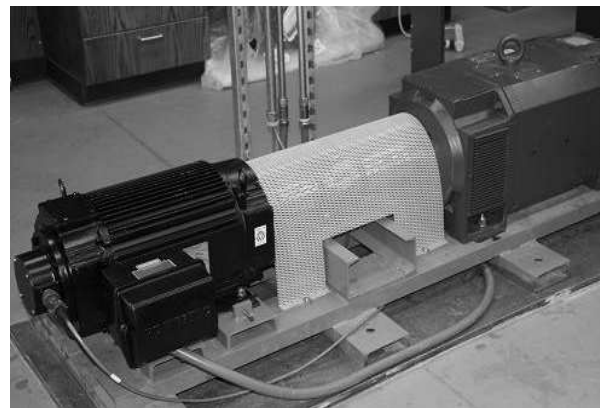


Figure 4. PMSM experimental test bed

Figures 6-11 show the deviation in the simulated current for a load fluctuation condition in each phase via residuals magnitude. The residuals or errors are produced by comparing the three phase current predictions and the actual values of

the three phase currents. The residual for phase A (phases B and C are similar) is expressed in (12):

$$r_a(t) = [I_a(t) - \hat{I}_a(t)] \quad (12)$$

where I_a is the actual value of current in phase A at time t and \hat{I}_a is the predicted value of current in phase A at time t .

As shown in Figures 5-11. The residual magnitudes change depending on the PMSM's load condition. As noted in Figures 7-11, the NN model turn out the largest residuals when the load condition is going up from no load to load condition and going down towards no load condition. This behavior can be attributed to factors such as the time delay generated due to on line operation and the overshooting produced during the change of the variables in the model. Additionally, it can be observed the variation in the magnitude of residuals as a result of the maximum load change. In summary, the performance of the model developed is affected slightly by the variation in the load fluctuation.

Table 1. Generalization performance of the network from no torque up to 45% of the rated torque

Rated torque (%)	MSE (%)			Mean Error (%)		
	$I_A(t)$	$I_B(t)$	$I_C(t)$	$I_A(t)$	$I_B(t)$	$I_C(t)$
0	1.44	1.43	1.47	0.12	0.21	0.01
10	1.13	1.16	1.17	0.10	0.20	0.01
20	0.68	0.71	0.73	0.05	0.11	0.02
30	0.35	0.41	0.35	0.04	0.08	0.04
40	1.16	1.2	1.17	0.14	0.14	0.03
45	2.26	2.20	2.25	0.15	0.16	0.03

Table 2. Generalization performance of the network for 45% of the rated torque using different ramp slopes

Slope	MSE (%)			Mean Error (%)		
	$I_A(t)$	$I_B(t)$	$I_C(t)$	$I_A(t)$	$I_B(t)$	$I_C(t)$
0.175	0.41	0.48	0.41	0.07	0.05	0.01
0.14	0.53	0.57	0.59	0.12	0.12	0.02
0.12	0.36	0.40	0.35	0.01	0.001	0.07
0.0875	0.33	0.35	0.33	0.01	0.01	0.01

4. CONCLUSIONS

The NN based approach to model a PMSM under load fluctuation proposed shows its efficacy in performing current prediction when the PMSM is running under different conditions of load. It is noted, that experimentally the load fluctuation condition does not produce any significant increase in the residuals in each phase studied from no load condition up to 45% of the rated torque.

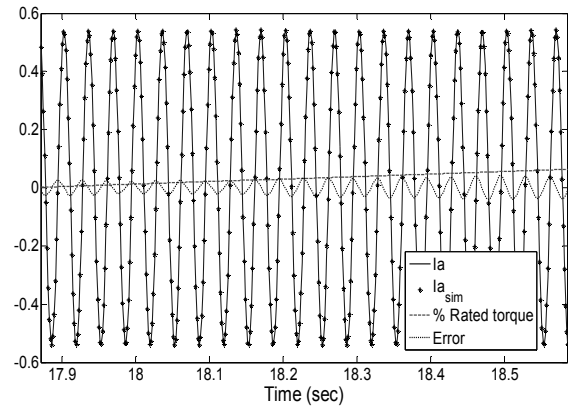


Figure 5. Actual and simulated fundamental component current phase A under load fluctuation

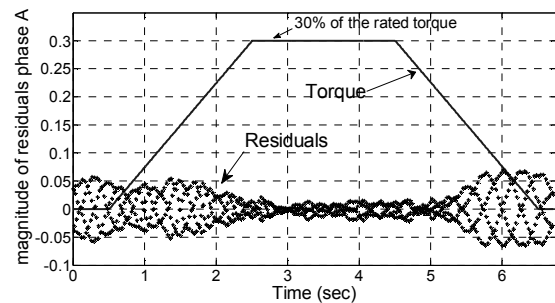


Figure 6. Residuals phase A under load fluctuation (0 to 30% of rated torque)

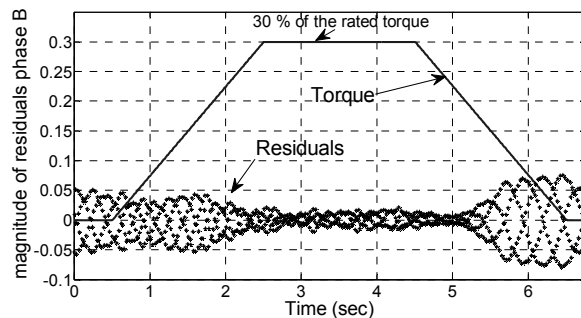


Figure 7. Residuals phase B under load fluctuation (0 to 30% of rated torque)

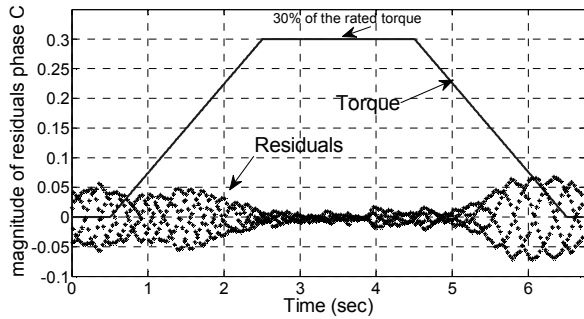


Figure 8. Residuals phase C under load fluctuation (0 to 30% of rated torque)

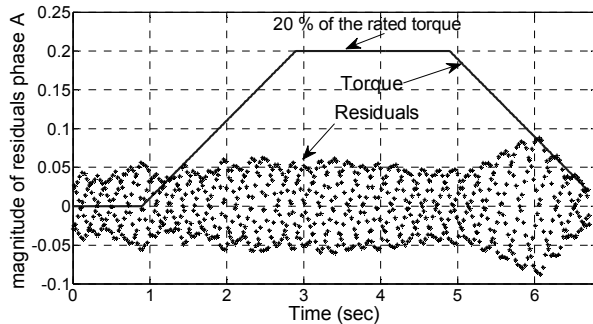


Figure 9. Residuals phase A under load fluctuation (0 to 20% of rated torque)

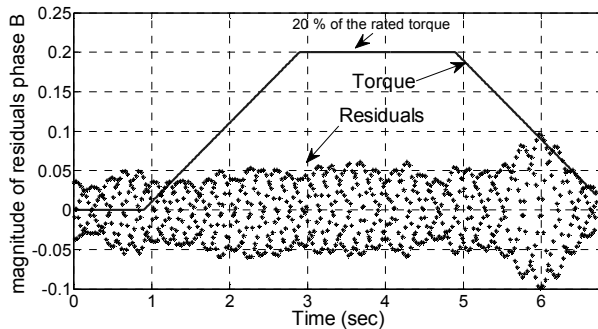


Figure 10. Residuals phase B under load fluctuation (0 to 20% of rated torque)

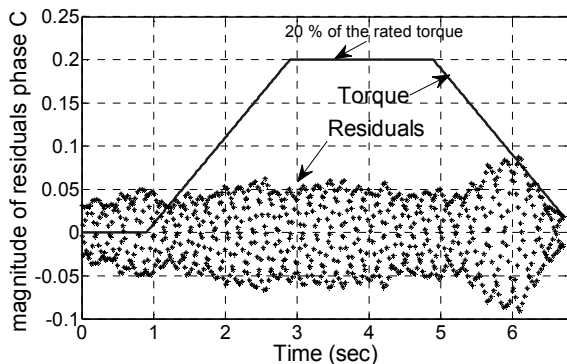


Figure 11. Residuals phase C under load fluctuation (0 to 20% of rated torque)

REFERENCES

[1] ATIYA, A. F. AND PARLOS, A.G. New results on recurrent network training: Unifying the algorithms and accelerating convergence, *IEEE Trans. Neural Networks*, vol. 13, 765–786, 2000.

[2] LI ZHANG AND YUGENG XI. Nonlinear system identification based on an improved support vector regression estimator. In *Advances in Neural Networks, International Symposium on Neural Networks, Dalian, China*, 586-591, August 2004.

[3] IOAN, D.L., *System Identification and control design*, Prentice Hall, 1990.

[4] RAHMAN, M.A., HOQUE, M.A., On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors, *IEEE Transaction on Energy Conversion*, , vol.13, no.4, 311-318, 1998.

[5] WENBIN, W., XUEDIAN, Z., JIAQUN, X., RENYUAN, T. A feedforward control system of PMSM based on artificial neural network. *Proceedings of the Fifth International Conference on Electrical Machines and Systems*, Volume 2, 679 - 682, Aug 2001.

[6] KUMAR, R., GUPTA, R.A., BANSAL, A.K.. Identification and Control of PMSM Using Artificial Neural Network. *IEEE International Symposium on Industrial Electronics*, Volume, Issue, 4-7,30 – 35, June 2007.

[7] FAYEZ F. M. EL-SOUSY. High-Performance Neural-Network Model-Following Speed Controller for Vector-Controlled PMSM Drive System, *IEEE International Conference on Industrial Technology*, Tunisia, December 2004.

[8] SIO, K.C.; LEE, C.K., Identification of a nonlinear motor system with neural networks, *International Workshop on Advanced Motion Control*, vol.1, 287-292, Mar 1996.

- [9] PARLOS, A.G., RAIS, O., AND ATIYA, A., Multi-Step-Ahead Prediction using Dynamic Recurrent Neural Networks, International Joint Conference on Neural Networks, Vol 1, 349 – 352, July 1999.
- [10] MOHAMED, F.A.; KOIVO, H., Modeling of induction motor using non-linear neural network system identification, SICE Annual Conference , vol.2, 977-982, Aug. 2004.
- [11] KIM K. AND PARLOS, A., Induction Motor Fault Diagnosis Based on Neuronpredictors and Wavelet Signal Processing”, IEEE/ASME Transactions on Mechatronics, Vol. 7, No 2, 201-219, June 2002.
- [12] NARENDRA,K.S.; PARTHASARATHY, K., Identification and control of dynamical systems using neural networks," IEEE Transactions on Neural Networks, vol.1, no.1, 4-27, Mar 1990.
- [13] HUSH, D.R. AND HORNE, B.G., Progress in supervised neural networks, Signal Processing Magazine, IEEE , vol.10, no.1, 8-39, Jan 1993.
- [14] DE JESUS, O.; HORN, J.M.; HAGAN, M.T., Analysis of recurrent network training and suggestions for improvements, International Joint Conference on Neural Networks, vol.4, 2632-2637, 2001.
- [15] FORESEE, F.D., AND M.T. HAGAN, Gauss-Newton approximation to Bayesian regularization, Proceedings of the 1997 International Joint Conference on Neural Networks,1930–1935, 1997.
- [16] MACKAY, D.J.C., Bayesian interpolation, Neural Computation, Vol. 4, No. 3, 415–447, 1992.
- [17] HAGAN, M.T., DEMUTH, H.B. AND BEALE, M.H. Neural Network Design, Boston, MA: PWS Publishing, 1996.
- [18] NGUYEN, D., AND WIDROW, B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, Proceedings of the International Joint Conference on Neural Networks, Vol. 3, pp. 21–26, 1990.