International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1 (2018) pp. 80-85

Speed Control of Induction Motor Drive Using Artificial Neural Networks- Levenberg-Marquardt Backpropogation Algorithm

V. Raveendra Reddy

Research Scholar, Sri Venkateswara University, Tirupati, and Associate professor of Department of Electrical and Electronics Engineering, RamiReddy Subbarami Reddy Engineering College, Kadanuthala, NH-5, S.P.S.R. Nellore District, Andhra Pradesh, India.

Orcid Id: 0000-0001-9942-0889

Dr. V.C. Veera Reddy

Professor, Department of Electrical and Electronics Engineering, Annamacharya Institute of Tech & Science, Tirupati, Andhra Pradesh, India.

Dr. V.Chandra Jagan Mohan

Professor, Department of Electrical and Electronics Engineering, Institute of Aeronautical Engineering, Dundigal, Hyderabad-500043, Andhra Pradesh, India.

Abstract

This paper affords an artificial neural network (ANN) primarily based space vector pulse width modulated direct torque control (DTC) scheme to control speed and torque of IM drive. The Levenberg-Marquardt back propagation (LMBP) technique has been used to train the neural network.

A neural network controller is proposed to replace the conventional PID controllers to enhance the drive's performance since the performance of an electric drive genuinely relies upon on the excellent of a speed controller. The neural network controller was trained and realizes for a speed controller. The neural community controller changed into educated and realizes for a speed controller. The controller become applied within the feed-forward back propagation algorithm to test its performance. the network is trained the use of multi layer feed forward back propagation algorithm to check its performance.A simulation model representing the complete neural network based direct torque control scheme of induction motor drive using svpwm is developed and verified using MATLAB/Simulink block program. The results of ANN fed DTC based speed control of induction motor drive compared with the results of space vector pulse with modulator (SVPWM) controlled induction motor (I.M) drive. Time analysis (rise time, delay time, peak time and over shoot), total harmonic distortion(THD)of both (DTC SVPWMIM and ANNDTCIM) models has been done and results are compared.

Keywords: Direct Torque Control (DTC), Induction Motor (I.M), Artificial Neural Network (ANN), SpaceVector PulseWidth Modulation (SVPWM), Levenberg-Marquardt back propagation (LMBP).

INTRODUCTION

Artificial Neural Networks (ANN) is known for its capability in providing main features, such as: flexibility, competency of learning by instances, and capability to generalize and solve problems in pattern classification, optimization, function approximation, pattern matching and associative memories [1, 2]. Due to their powerful capability and functionality, ANN provides an unconventional approach for many engineering problems that are difficult to solve by normal methods, and extensively used in many extents such as control, speech production, signal processing, speech recognition and business [1]. Among numerous neural network models, the Multilayer Feed-Forward Neural Networks (MLFF) have been primarily used due to their well-known universal estimation proficiencies [3]. For MLFF training back-propagation (BP) algorithm and Levenberq-Marquardt (LM) which are based on gradient descent are mostly used [4]. The most frequently used method to train an ANN is based on BP [5-6]. The BP learning has become the standard method and process in adjusting weights and biases for training an ANN in many domains [7]. The most effective method is Levenberg Marquardt (LM) algorithm [8], which is a derivative of the Newton method. This is quite multifaceted algorithm since not only the gradient but also the Jacobian matrix is calculated. The LM algorithm was developed only for layer-by-layer ANN topology, which is far from optimal [9]. LM algorithm is coined as one of the most successful algorithm in increasing the convergence speed of the ANN with MLP architectures [10]. It is a good combination of Newton's method and steepest descent [11]. It Inherits speed from Newton method but it also has the convergence capability of steepest descent method. It suits specially in training neural network in which the performance index is calculated in Mean Squared Error (MSE) [12].

NEURAL NETWORKS

A neural network is a massively parallel distributed processor made up of simple Processing units that have a natural tendency for storing experiential knowledge and making it available for us. artificial neural community (ANN) is a sort of synthetic intelligence method that mimics the conduct of the human brain (Haykin, 2009).ANNs have the ability to model linear and non-linear systems without the want to make assumptions implicitly as in maximum traditional statistical methods. They have been applied in various aspects of science and engineering.

Proposed System

The block diagram of induction motor using ANN controller is shown in fig below figure(1). The paper [14] presents simple structured neural networks for flux position estimation, sector selection and stator voltage vector selection for induction motors using DTC method. The simple structure network facilitates a short training and processing times. the concept and implementation of a new simple space vector pulse width modulated direct-torque neuro-fuzzy control (SVPWMDTNN) scheme for pulse width-modulation-inverter-fed induction motor drive is presented. An adaptive neuro-fuzzy inference system is applied to achieve high-performance decoupled flux and torque control.

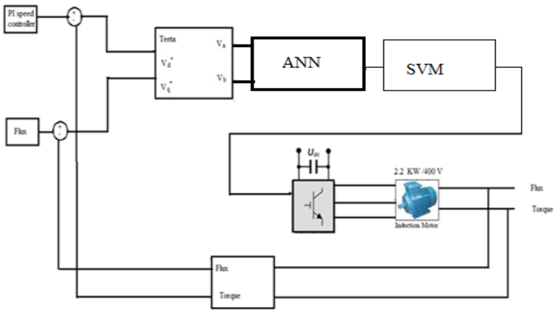

Figure 1: Block diagram of the proposed system

Designing and programming ANN models

Designing ANN models follows a number of systemic procedures as shown in figure (2). In general, there are five basics steps: (1) collecting data, (2) preprocessing data, (3) building the network, (4) train and (5) test performance of the model.

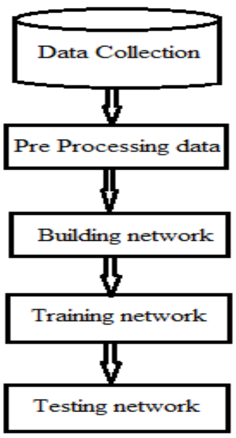

Figure 2: Basic flow for designing artificial neural network model

- *Data collection:* Collecting and preparing sample data is the first step in designing ANN models.
- *Data pre-processing:* After data collection, three data preprocessing procedures are conducted to train the ANNs more efficiently. These procedures are: (1) solve the problem of missing data, (2) normalize data and (3) randomize data.
- *Building the network:* At this stage, the designer specifies the number of hidden layers as shown in fig(3),it is a three input (Vd,Vq &theta) and two output (Va,Vb)structure, neurons in each layer, transfer function in each layer, training function, weight/bias learning function, and performance function

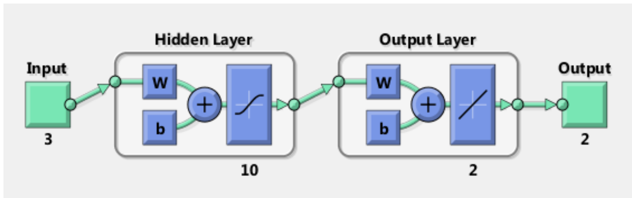

Figure 3: Neural network diagram of theproposed model

- *Training the network:* During the training process, the weights are adjusted in order to make the actual outputs (predicated) close to the target (measured) outputs of the network. The screen print of the neural network training state is shown in fig(4).

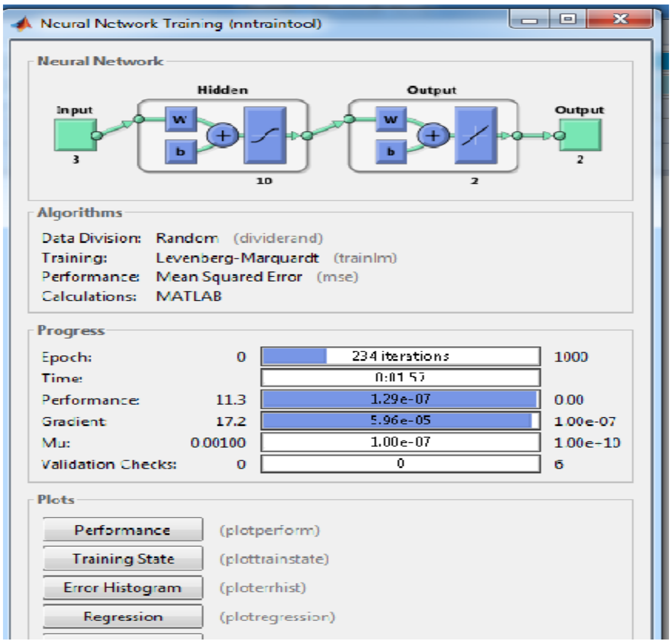

Figure 4: Neural network training state

Testing the network: The following step is to check the overall performance of the advanced model. At this level unseen data are uncovered to the model. In order to evaluate the overall performance of the advanced ANN fashions quantitatively and verify whether there may be any underlying trend in performance of ANN models, statistical evaluation involving the the root mean square error (RMSE), and the mean bias error (MBE) were conducted. RMSE provides information on the short term performance which is a measure of the variation of predicated values around the measured data. The lower the RMSE, the more accurate is the estimation. MBE is an indication of the average deviation of the predicted values from the corresponding measured data and can provide information on long term performance of the models; the lower MBE the better is the long term model prediction.

Back propagation learning algorithm based on Levenber Marquardt Algorithm(LM)

Levenberg – Marquardt algorithm is specifically designed to minimize sum-of-square error functions [28], of the form.

$$E = \tfrac{1}{2} \sum k (e_k)^2 = \tfrac{1}{2} \|e\|^2$$

Where ek is the error in the kth exemplar or pattern and e is a vector with element ek. If the difference between the pervious weight vector and the new weight vector is small, the error

vector can be expanded to first order by means of a Taylor series.

$$e(j+i) = e(j) + \partial e_k / \partial w_i \left(w(j+1) - w(j)\right)$$

As a consequence, the error function can be expressed as

$$E = \tfrac{1}{2} \left\| e(j) + \partial e_k / \partial w_i \left(w(j+1) - w(j)\right) \right\|^2$$

Minimizing the error function with respect to the new weight vector, gives

$$w(j+1) = w(j) - \left(Z^T Z\right)^{-1} Z^T e(j)$$

Where

$$(Z)_{ki} \equiv \partial e_k / \partial w_i$$

Since the Hessian for the sum-of-square error function is

$$(H)_{ij} = \partial^2 E / \partial w_i \partial w_j = \sum \left\{ \left(\partial e_k / \partial w_i\right)\left(\partial e_k / \partial w_i\right) + e_k \partial^2 e_k / \partial w_i \partial w_j \right\}$$

Neglecting the second term, the Hessian can be written as $H = Z^T Z$

Updating of the weights so involves the inverse Hessian or an approximation therefrom for nonlinear networks. The Hessian is comparatively easy to compute, since it is based on first order derivatives with respect to the network weights that are easily accommodated by back propagation. Though the change formula can be applied iteratively to attenuate the error perform, this might lead to an outsized step size, which might invalid the linear approximation on that the formula is predicated.

In the Levenberg-Marquardt algorithm, the error function is minimized, while the step size is kept small in order to ensure the validity of the linear approximation. This is accomplished by use of a modified error function of the form.

$$E = \tfrac{1}{2} \left\| e(j) + \partial e_k / \partial w_i \left(w(j+1) - w(j)\right) \right\|^2 + \lambda \left\| w(j+1) - w(j) \right\|^2$$

Where λ is a parameter governing the step size. Minimizing the modified error with respect to W(j+1) gives very large values of λ amount to standard gradient descent, while very small values λ of amount to the Newton method.

$$w(j+1) = w(j) - \left(Z^T Z + \lambda I\right)^{-1} Z^T e(j)$$

Performance evaluation using Levenberg-Marquardt Algorithm

Levenberg-Marquardt back propagation algorithm is used for training the network as shown in below figure(5) [7,8]. Training automatically stops when generalization stops improving, as indicated by an increase in the Mean Square Error (MSE) of the validation samples. The Mean Squared Error (MSE) is the average squared difference between outputs and targets. Lower values are better while zero means no error. Regression R analysis is performed to measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship. The performance plot of the Levenberg-Marquardt back propagation algorithm is shown in figures (6) &(7).

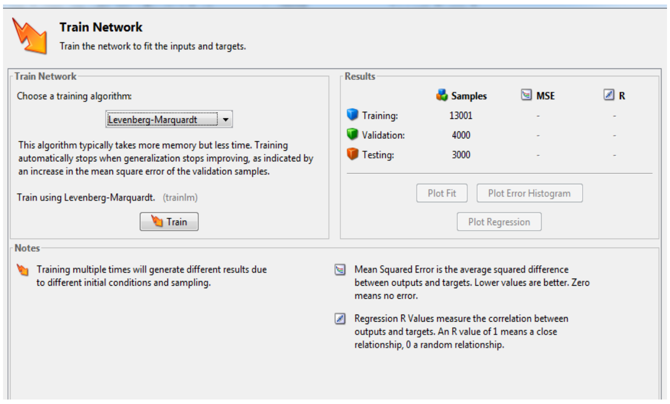

Figure 5: Neural netwok training using Levenberg-Marquardt back propagation algorithm

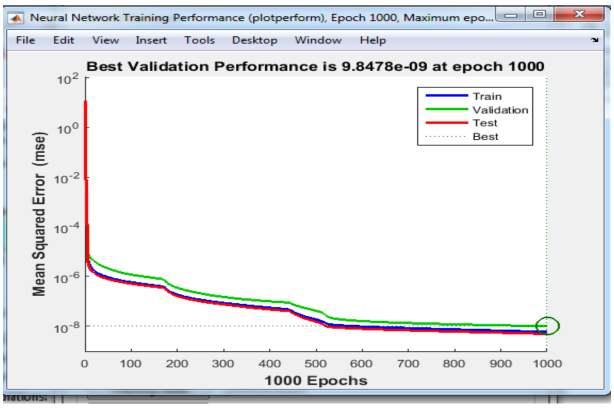

Figure 6: Performance of Levenberg-Marquardt Backpropagation Algorithm

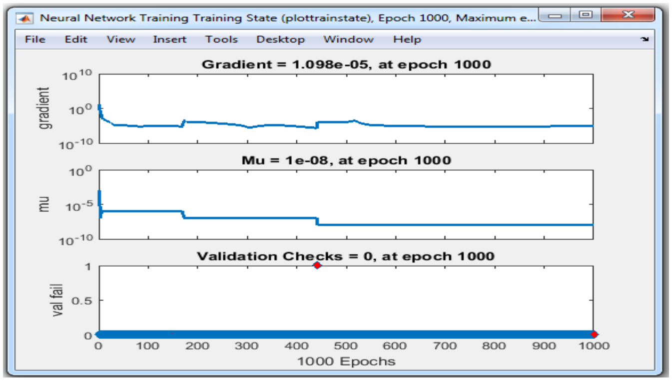

Figure 7: Training state and Performance plot of Levenberg-Marquardt Backpropagation Algorithm

From Figure (6) it is observed that the best validation performance0.0000000098478 at epoch 1000 is obtained. The Regression plot shown in Figure (8) shows the perfect correlation between the outputs and the targets. Figure (9) shows error histogram of the proposed neural network.

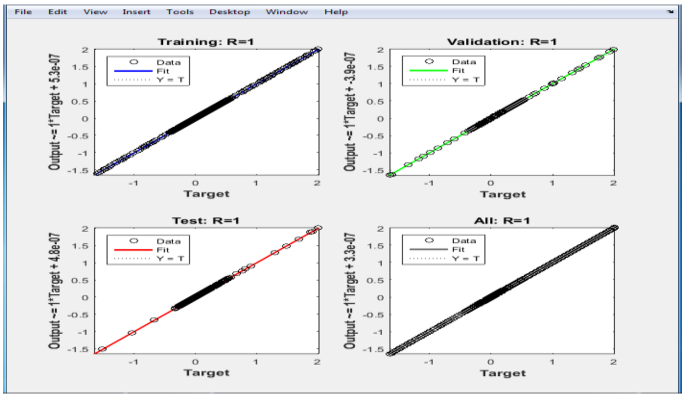

Figure 8: Regression Analysis Plot - Levenberg- Marquardt Backpropagation Algorithm

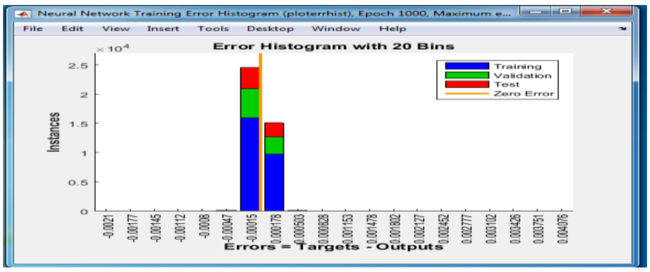

Figure 9: Error Histogram of the svpwm fed nndtc I.M drive

Table1: The Mean Square Error (MSE) and Regression (R) values for the Training, Validation

	MSE	**R**
1. Training	5.80557e-9	9.99999e-1
2. validation	9.84777e-7	9.99999e-1
3. Testing	4.81392e-9	9.99999e-1

The result for training, validation and testing samples is illustrated in Table 4.1. It is observed that the value of R is closest to 1 indicating the accurate prediction. When the data set was trained in Levenberg–Marquardt algorithms the performance obtained was in 1000 epochs. Levenberg–Marquardt algorithm (LM) is the most widely used optimization algorithm. LM algorithm is an iterative technique that locates a local minimum of a multivariate function that is expressed as the sum of squares of several non-linear, real-valued functions. It has become a standard technique for non linear least-square problems, widely adopted in various disciplines for dealing data-fitting applications.

This paper presented a Levenberg-Marquardt back propogation optimization algorithm, applied to the conventional Direct torque control using space vector pulse width modulated induction motor drive. The results indicated that the Marquardt algorithm is very efficient when training networks which have up to a few hundred weights. Although the computational requirements are much higher for each iteration of the Marquardt algorithm, this is more than made up for by the increased efficiency. This is specially true when high accuracy is essential. It is also found that in many cases the Marquardt algorithm converged when the conjugate gradient and variable learning rate algorithms failed to converge.The time analysis (peak time, rise time, slew rate, settling time and overshoot) ,total harmonic distortion of the NNDTC I.M drive is compared with the SVPWM fed DTC I.M drive and it proved that the reduced peak time, settling times were obtained with Neural Network Controller and results are tabulated in below table 2.

Table 2: Comparison table of peak time, rise time, slew rate, settling time and overshoot, THD of SV PWM and SVPWM fed NNDTC I.M drive.

	Speed (rpm)	t_p (sec)	t_r (sec)	Slew rate(v/sec)	t_s(sec)	Over shoot %	THD
SVPWM fed NNDTC	1500	0.0033	0.001065	663.017	0.006551	------	**20.94**
SVPWM	1500	0.01	0.007	921.036	0.0110	5.851	**63.3**

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1 (2018) pp. 80-85

Matlab/simulink model

Figure(10) shows matlab/simulink model of the space vector direct torque controlled induction motor using neural network controller.

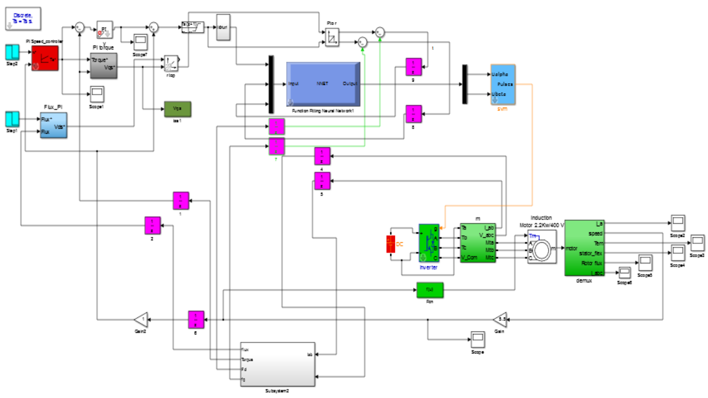

Figure 10: simulink diagram of the SVPWM DTC I.M using NNC

SIMULATION RESULTS:

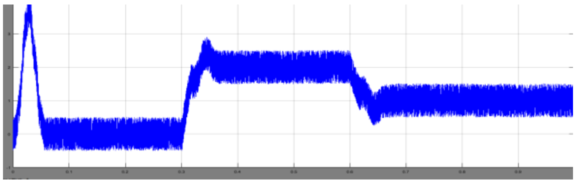

Figure 11: Steady state plot torque for DTC I.M drive

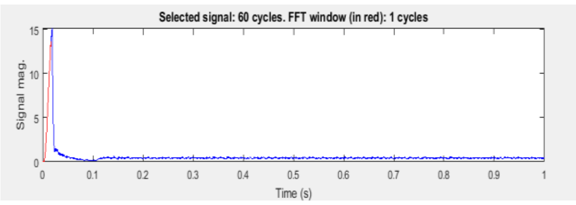

Figure 12: Steady state plot torque waveform of spwm

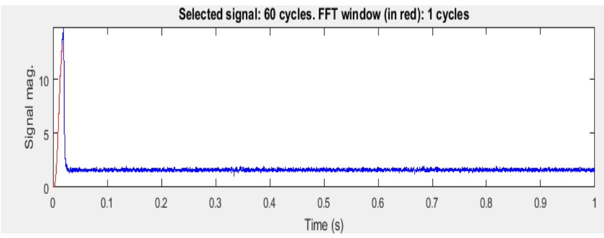

Figure 13: Steady state plot torque waveform of svpwm

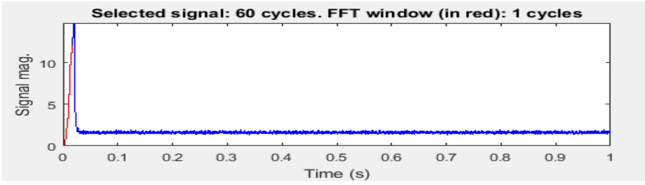

Figure 14: Steady state plot torque waveform of svpwm fed NNDTC I.M drive

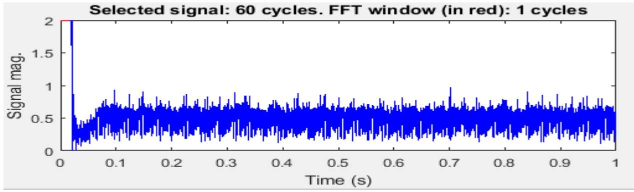

Figure15: Stator voltage waveform of quadrature axis for svpwm fed NNDTC I.M drive

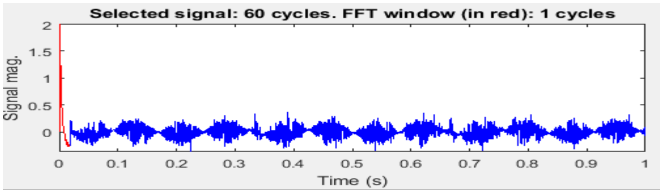

Figure16: Va of for svpwm fed NNDTC I.M drive

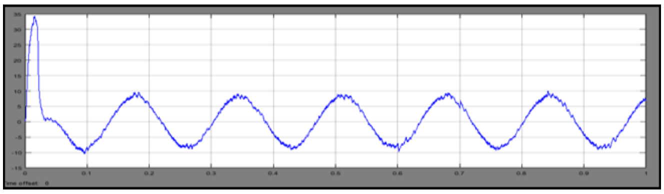

Figure17: Steady state plot of current for svpwm fed NNDTC I.M drive

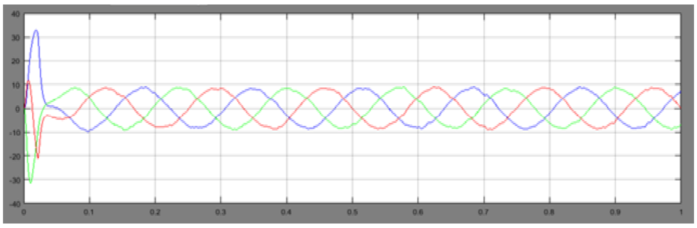

Figure18: Steady state plot of Stator current for svpwm fed NNDTC I.M drive

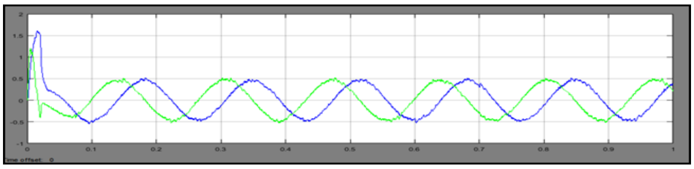

Figure 19 : Steady state plot of Stator flux for svpwm fed NNDTC I.M drive

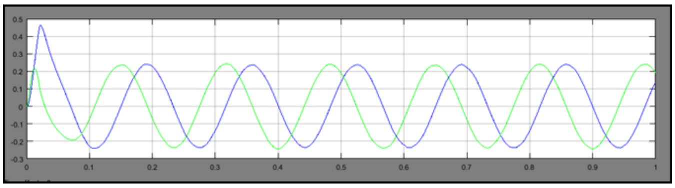

Figure20: Steady state plot of rotor flux for svpwm fed NNDTC I.M drive

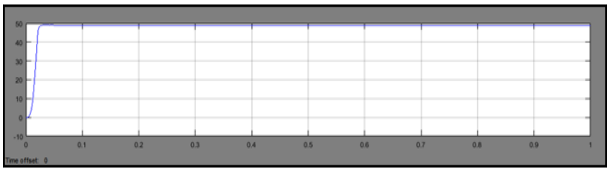

Figure 21: Steady state plot of rotor flux for svpwm fed NNDTC I.M drive

CONCLUSION

Speed control of induction motor by direct torque control, sinusoidal pulse width modulation,space vector modulation and direct torque control with neural network controller has been done by simulating matlab/simulink models.The results of simulink models are compared from figures (11-20). Time analysis peak time, rise time, slew rate, settling time and overshoot,THD of SV PWM and SVPWM fed NNDTC I.M drive models are compared in table 4.2 and also it show the improved performance and reduced distortion in case of DTC using Levenberg-Marquardt back propogation optimization algorithm.

REFERENCES

[1] José Manuel Ortiz-Rodríguez, Ma. del Rosario Martínez-Blanco, José Manuel Cervantes Viramontes and Héctor René Vega-Carrillo “Robust Design of Artificial Neural Networks Methodology in Neutron Spectrometry”.

[2] M. K., M. C., and R. S., Elements of Artificial Neural Networks. Cambridge, MA: MIT Press, 1997.

[3] S. Haykin, Neural Networks a Comprehensive Foundation. Prentice Hall, New Jersey, 1999.

[4] Ozturk, C., &Karaboga, D. (2011). Hybrid Artificial Bee Colony algorithm for neural network training.2011 IEEE Congress of Evolutionary Computation (CEC) (pp. 84–88). IEEE. doi:10.1109/CEC.2011.5949602

[5] Abid, S., Fnaiech, F. and Najim, M.: A fast feedforward training algorithm using a modified form of the standard back propagation algorithm, IEEE Transactions on Neural Networks 12 (2001), 424–430.

[6] Yu, X., Onder Efe, M. and Kaynak, O.: A general back propagation algorithm for feed forward neural networks learning, IEEE Transactions on Neural Networks 13 (2002), 251–259.

[7] Chronopoulos AT, Sarangapani J, (2002) A distributed discrete time neural network architecture for pattern allocation and control Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS’02), Florida, USA pp,204-211.

[8] M.T. Hagan. M.B. Menhaj. "Training feed forward networks with the Marquardt algorithm." IEEE Trans. on Neural Networks; NN-5:989-993. Sciences, 1994;Vol.23, pp 899–916.

[9] Bogdan M. Wilamowski, Nicholas Cotton, OkyayKaynak “Neural Network Trainer with Second Order Learning Algorithms” INES 2007 11th International Conference on Intelligent Engineering Systems Budapest, Hungary 2007 IEEE.

[10] M. T. Hagan and M. B. Menhaj, “Training feed forward networks with the Marquardt algorithm,” IEEE Trans. Neural Netw., 1994;vol. 5, no. 6, pp. 989–993.

[11] CAO Xiao-ping, HU Chang-hua, ZHENG Zhi-qiang, LV Ying-jie “Fault Prediction for Inertial Device Based on LMBP Neural Network”, Electronics Optics & Control, 2005; Vol.12 No.6, pp.38-41.

[12] Simonhaykin, neural networks, China Machine Press, Beijing, 2004.

[13] Aldrich .C (2002), “Exploratory Analysis of Metallurgical Process Data with Neural Networks and Related Methods”, Elsevier Science, British Library Cataloguing in Publication Data, Netherlands, Vol.1, pp.56-57.M.H. Meskon, M. Albert and F.Hedouri. “Fundamentals of Management”. Moscow: Delo, 1998.

[14] Rajesh Kumar, R.A.Gupta, S.V.Bhangale, Himanshu Gothwal, “Artificial neural network based direct torque control of induction motor drives”, *IETECH Journal of Electrical Analysis,* Vol.2, No.3, pp.159-165, 2008.