Performances Comparison of Neural Architectures for On-Line Speed Estimation in Sensorless IM Drives

K.Sedhuraman, S.Himavathi and A.Muthuramalingam

***Abstract*—**The performance of sensor-less controlled induction motor drive depends on the accuracy of the estimated speed. Conventional estimation techniques being mathematically complex require more execution time resulting in poor dynamic response. The nonlinear mapping capability and powerful learning algorithms of neural network provides a promising alternative for on-line speed estimation. The on-line speed estimator requires the NN model to be accurate, simpler in design, structurally compact and computationally less complex to ensure faster execution and effective control in real time implementation. This in turn to a large extent depends on the type of Neural Architecture. This paper investigates three types of neural architectures for on-line speed estimation and their performance is compared in terms of accuracy, structural compactness, computational complexity and execution time. The suitable neural architecture for on-line speed estimation is identified and the promising results obtained are presented.

***Keywords*—**Sensorless IM drives, rotor speed estimators, artificial neural network, feed- forward architecture, single neuron cascaded architecture.

I. Introduction

INDUCTION motors are applied today, to a wide range of applications requiring variable speed. Accurate speed measurement is necessary to realize high performance and high-precision speed control of an induction motor. The speed is obtained by using mechanical sensors as resolver or pulse encoders. However, these sensors are usually expensive, bulky, and subject to failures under hostile industrial environments. Therefore, the cost and size of the drive systems are increased. Speed sensorless closed loop control of induction motor drives, leads to cheaper and reliable control. Therefore sensorless control of induction motor drives has become an active area of research. Advances in digital technology have made the sensorless control realizable by industries for high performance variable speed applications.

Since the late 1980s, speed-sensorless control methods of induction motors using the estimated speed instead of the measured speed have been reported. They have estimated speed from the instantaneous values of stator voltages and currents using induction motor model. Other approaches to estimate speed use Rotor Slot Harmonic [1], [2], Extended Kalman Filter (EKF), Extended Luenbergern Observer (ELO) [3], Saliency Techniques [4], [5] and Model Reference Adaptive System (MRAS) [6], [7]. The saliency techniques attempt to be parameter independent, but secondary magnetic effects do lead to complications in their implementation. Rotor slot harmonic speed estimation will work successfully if the rotor is about a minimum speed. The problems related to EKF or ELO are the large memory requirement, computational intricacy, and the constraint such as treating all inductances to be constant in the machine model. MRAS schemes are also direct dependent on motor parameters. However, an induction motor is highly coupled, non-linear dynamic plant, and its parameters vary with time and operating conditions. Therefore, it is very difficult to obtain good performance for the entire speed range using previous methods. Recently, the use of Neural Network (NN) 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 [8]-[10]. Moreover, they have the advantages of extremely fast parallel computation and immunity to noise. A two layer NN was used for the speed estimation [11], [12]. Here, the learning techniques of NN were used to update the estimation parameter namely speed in the equation. This method has used the machine equations and hence not robust to variations in motor parameters. Thus, NN trained form input/output data based estimator is promising alternative for on-line speed estimation in sensorless controlled induction motor drives.The major issues in NN based speed estimation are; the NN model should be accurate, simpler in design, structurally compact and computationally less complex to ensure faster execution time in real time implementation for effective control. This in turn to a large extent depends on the type of neural architecture which is the method of interconnection between the neurons. This paper investigates three types of neural architectures for on-line speed estimation and their performance is compared in terms of accuracy, structural compactness, computational complexity and execution time. The Neural architectures considered for investigations are Single Layer Feed-Forward (SLFF) Architecture, Multilayer Layer Feed-Forward (MLFF) Architecture and Single Neuron Cascaded (SNC) Architecture.The neural models for on-line speed estimation are obtained using input/output data and Levenberg-Marquardt (LM) training algorithm.The paper is organized as follows. Section II details the sensor-less IM drives. Section III gives the discussion about Feed-Forward architecture, Single Neuron Cascaded architecture and presents learning algorithms. Section IV describes the NN based speed estimation. Comparison of NN models for on-line speed estimation is carried out and discussed in section V. Section VI concludes the paper.

II. Speed Sensorless Vector Controlled IM Drives

The speed sensorless vector control presented here is indirect field oriented control (rotor flux oriented control). Fig. 1 shows the overall block diagram of the speed-sensorless

K.Sedhuraman, Research Scholar in the department of Electrical and Electronics Engineering, Pondicherry Engineering College, Puducherry-605014, India. (e-mail:sedhu_k@pec.edu).
S.Himavathi, Professor and Head in the department of Electrical and Electronics Engineering, Pondicherry Engineering College, Puducherry-605014, India. (e-mail: himavathi@pec.edu).
A.Muthuramalingam, Professor in the Department of Electrical and Electronics Engineering, Pondicherry Engineering College, Puducherry-605014, India. (e-mail:amrlingam@hotmail.com).

drive system of an induction motor using a speed estimator. The system consists of a solid state IM drive system, rotor flux oriented control, flux and speed estimator. Rotor flux oriented control consists of a PI speed controller, a current controller, and PWM generator. In sensor-less vector control IM drives the torque command is generated as a function of the speed error signal. The error in estimated speed will reflect in the torque command. Thus an error in speed estimation will lead to ineffective control and sometimes instability. So, the design of suitable NN model for on-line flux and speed estimation is inevitable for high performance drive.

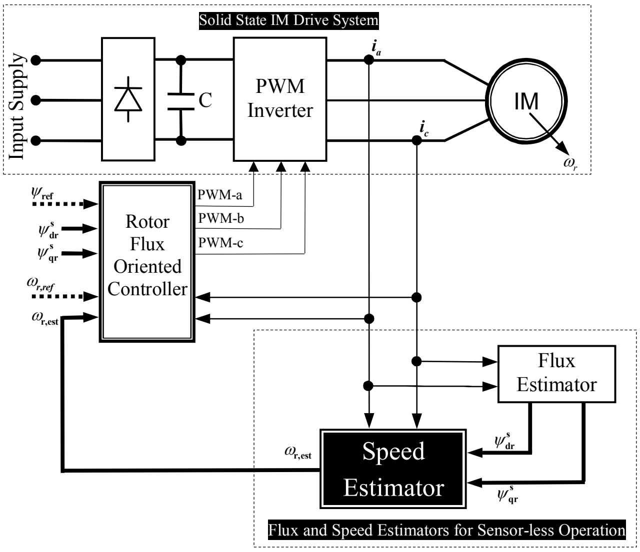


Fig. 1 Sensor-less Vector Controlled IM Drives showing the Estimators for Flux and Speed

III. NN Architectures & NN Learning Algorithms

The design of neural network to a large extent depends on the type of neural architectures and learning algorithms. The brief discussion about neural architectures considered for investigation and learning algorithms are presented.

A. Feedforward Architecture-Single and multi-layer

Feed-forward architecture has an input layer, one or more hidden layers and an output layer. The signal flows only in the forward direction. Each neuron model in the architecture includes a nonlinear activation function. The commonly used activation functions such as tan-sigmoid/log-sigmoid is used for hidden layers while pure-linear function is used for output layer. Feed-Forward architecture with single hidden layer is called as "Single Layer Feed-Forward architecture and with more than one hidden layer is called as "Multilayer Layer Feed-Forward architecture. Extensive work using this architecture has established the nonlinear mapping capability of Neural Networks.

B. Cascade Architecture-Single Neuron cascading

The Cascade architecture consists of an input layer, hidden layers and an output layer. The first hidden layer receives only external signals as inputs. Other layers (M) receive external inputs and outputs from all previous (M-1) layers. It is called cascade because the input to a neuron consists of system inputs and outputs of all preceding layers/neurons. This is in contrast to the feed-forward architecture where inputs to a neuron are only from previous layer. Cascading single neuron in every hidden layer results the "Single Neuron Cascaded" (SNC) architecture which greatly simplifies the design process and can be self-organized which aids design automation similar to SLFF-NN.

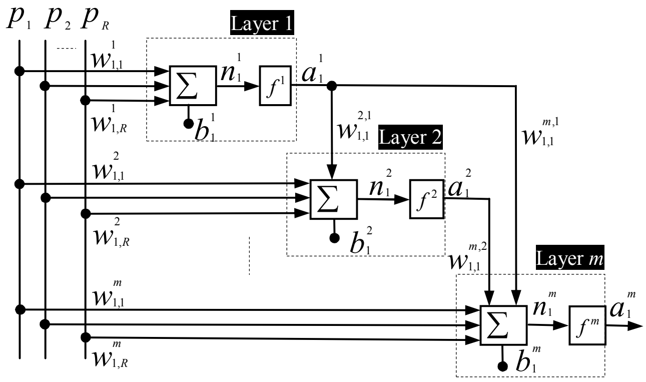


Fig. 2 SNC-NN with multiple inputs/single output

where,

- p - Input vector, p=[1,2,...R]
- S^m - Number of neurons in the layer 'm' where m=[1,2,...M] and $S^o = p$
- $w^m_{i,R}$ - Input weight of neuron 'i' of layer 'm' for external input 'R'.
- $w^{m,k}_{i,j}$ - Link weight of neuron 'i' of layer 'm' for input from neuron 'j' of layer 'k'.
- b^m_i - Bias for neuron 'i' of layer 'm'.
- f^m - Activation functions of all neurons in a layer 'm'.
- a^m_i - Output of neuron 'i' of layer 'm'

The Single Neuron Cascaded (SNC) architecture with multiple inputs/single output is shown in Fig. 2. Each neuron in the architecture includes weights, bias and a nonlinear activation function. The weights of interconnections to the previous layer are called as "input weights" and the weights of interconnections between the layers are called "link weights". The commonly used tan-sigmoid/log-sigmoid activation function is used for all hidden layers while pure-linear function is used for output layer. Initially, a hidden layer with only one neuron between the input and output is trained. To create a multilayer structure similar to MLFF-NN, hidden layers are added one by one and the whole network trained repeatedly using the concept of moving weights so as to obtain compact network [13]. This process continues, till the performance index is reached. Thus SNC-NN combines the advantage of SLFF-NN and MLFF-NN.

C. NN Learning Algorithms

There are different types of learning algorithms reported in the literature to train neural network [14]-[16]. Directed search algorithm use steep descent method. The first order approach and its variants are simple and effective. For better accuracy the second order approach namely Levenberg-Marquardt (LM) algorithm is used [15]. The higher accuracy is obtained at the cost of increased complexity of update laws. In this paper, NN models are trained off-line using LM algorithm to obtain higher accuracy for on-line speed estimation.

IV. Rotor Speed Estimation using Neural Networks

The chosen three architectures are used to model the on-line flux estimator. The systematic design process for SNC-NN is to add a hidden layer with single neuron at a time between the inputs/outputs till the target MSE is reached where as the systematic design procedure for SLFF-NN is to add single neuron in a single hidden layer at a time till the target MSE is reached. In MLFF-NN, the choice of number of layers and number of neurons in each layer is decided by trial and error. The design of MLFF-NN is more of an art than a science. Therefore, in this paper, MLFF-NN with two hidden layers is designed by trial and error method.

Around 46,525 data sets were obtained through simulation. The obtained data set is used as the training data set. The inputs to estimator are current and flux, whose components are direct and quadrature axis stator currents measured at (k-1)$^{\text{th}}$ sample $\{ i^s_{ds}(k-1),\ i^s_{qs}(k-1)\}$ and fluxes measured at k^{th} and (k-1)$^{\text{th}}$ sample $\{\Psi^s_{dr}(k),\ \Psi^s_{dr}(k-1),\ \Psi^s_{qr}(k),\ \Psi^s_{qr}(k-1)\}$. The output is the estimated rotor speed $\{\omega_r\}$ at k^{th} sample. The activation function for hidden and output layers is chosen as tan-sigmoid and pure linear function respectively. The inputs and outputs of NN based rotor speed estimator is shown in Fig. 4. For comparison, all the three NN models are trained with the same input/output data using LM algorithm for the same target mean square error (MSE) of 1×10^{-7}. The obtained SNC-NN, MLFF-NN and SLFF-NN model for on-line flux estimation are 6-15(h)-1 (h-hidden layer with single neuron), 6-15-15-1 and 6-75-1 respectively. The three models trained with same accuracy are tested for on-line estimation of rotor speed.

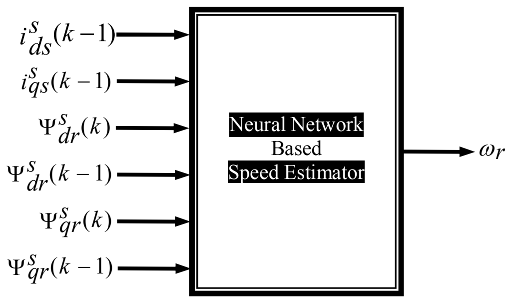


Fig. 4 The Inputs and Outputs of NN based Speed Estimator

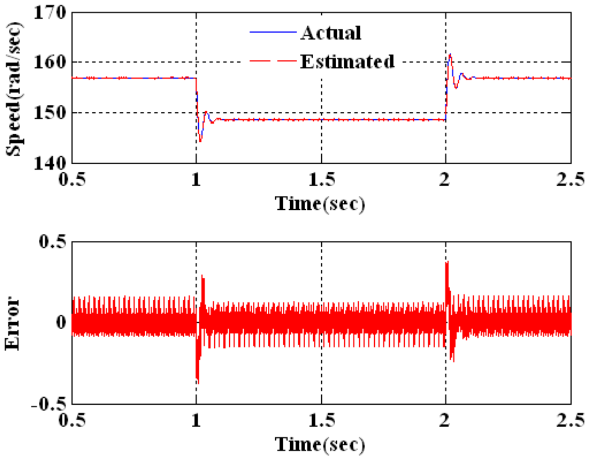
V. Comparison of NN Models for Rotor Speed Estimation

A. Steady State and Dynamic Performance of NN models for On-Line Speed Estimation

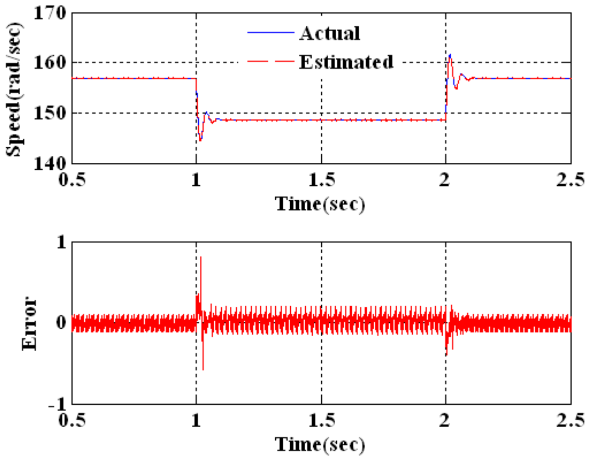
The performance of all the three NN models is compared in terms of accuracy. The off-line trained three NN models are tested for on-line estimation of rotor speed for various operating conditions extensively. The sample results for major operating conditions are presented. The operating conditions are explained in terms of operating speed and load for convenience. The operating conditions are: (1) transient load changes, (2) transient speed changes, and (3) very low speed.

The operating condition-I examines the performance of all the three NN models for transient load disturbance. The step and ramp change load torque response results are presented at rated speed. The motor is initially operated at rated speed under 0% loaded condition and 100% step change in load torque is applied at 1 s and rejected at 2 s. The rotor speed estimated using all the three NN models for step change in load torque are shown in the Fig. 5 respectively. The speed estimator performance for ramp change in load is presented in Fig. 6. The motor is loaded gradually from no load (at 1 s) to full load (at 2 s). Similarly the load is gradually decreasing from full load (3 s) to no load (4 s). The error curves between the actual and the estimated for all the models are shown. From the results obtained, it is observed that the load change capability of SNC-NN and MLFF-NN model is found to be similar and excellent where as the load change capability of SLFF-NN model is poor as compared to SNC-NN and MLFF-NN models.

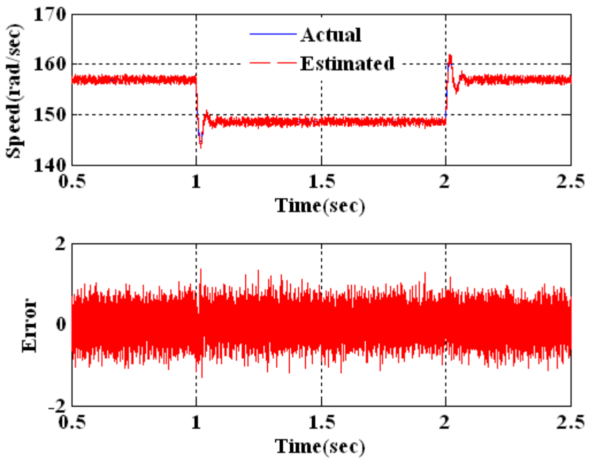
The operating condition-II test the performance of NN models for change in speed under no load condition. The tracking performance of the all three NN model is observed in Fig. 7 for a ramp speed command. Rotor speed is gradually decreased from 100% to 50% during 1 s to 2 s. Thereafter, the speed is maintained constant at 50% up to 3 s. Then rotor speed is gradually increased from 50% to 100% during 3 s to 4 s. The performance of the NN models during the step change of rotor speed is shown in Fig. 8. Machine operation starts at t = 0 s and a constant reference speed of 100% is considered. A step change in the reference speed occurs at 1sec and the new reference value is 50%. At t = 2 s, another step change in the reference speed occurs, which goes back to 100%. The error curves between the actual and the estimated for step and ramp response of all the three NN models are shown. The performance of NN model at very low speed is shown in Fig. 9 for operating condition-III.



(a)

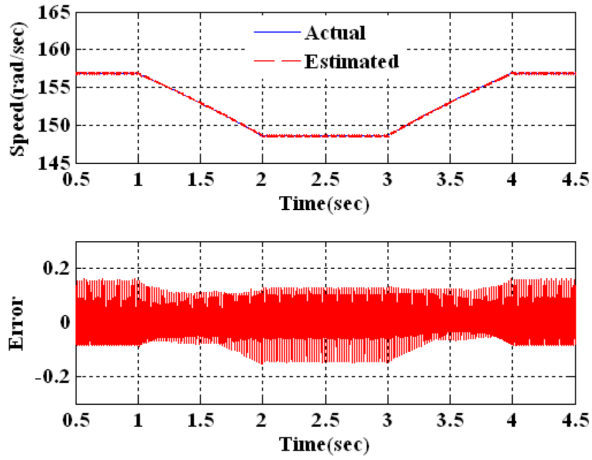


(b)

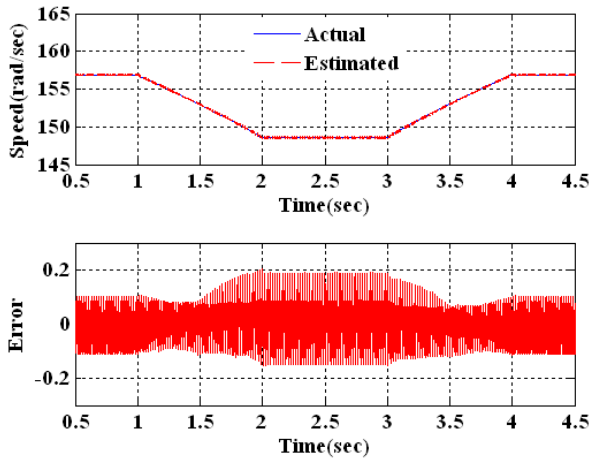


(c)

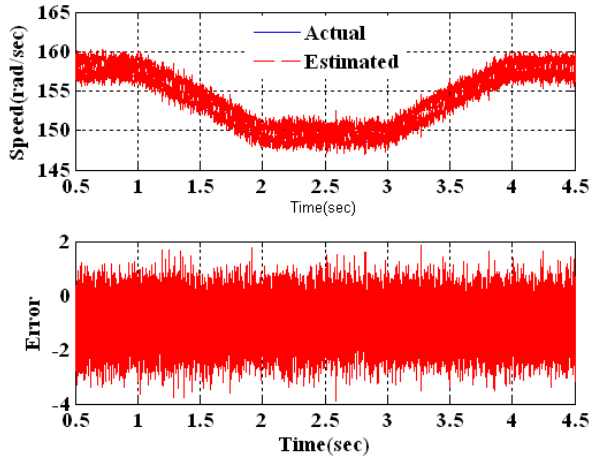
Fig. 5 Operating Condition-I for Step Change in Load: (a) SNC-NN (b) MLFF-NN (c) SLFF-NN



(a)



(b)



(c)

Fig. 6 Operating Condition-I for Ramp Change in Load: (a) SNC-NN (b) MLFF-NN (c) SLFF-NN

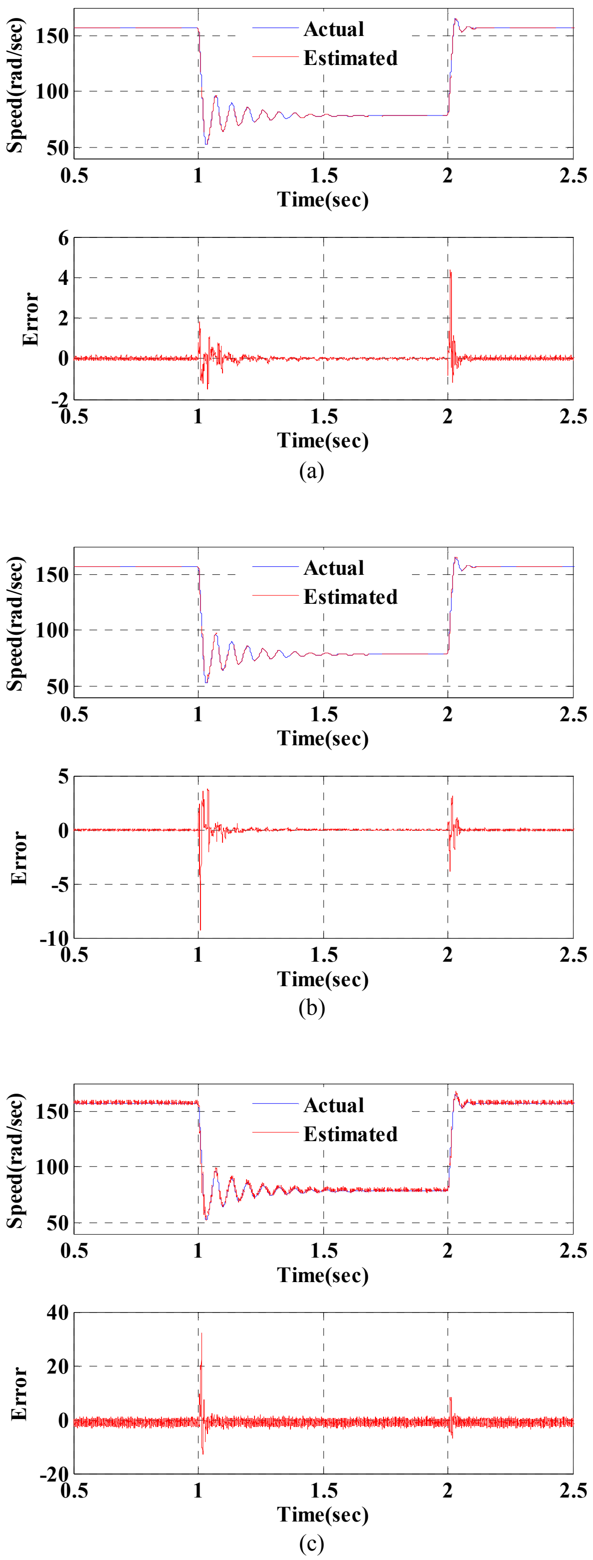


Fig. 7 Operating Condition-II for Step Change in Speed: (a) SNC-NN (b) MLFF-NN (c) SLFF-NN

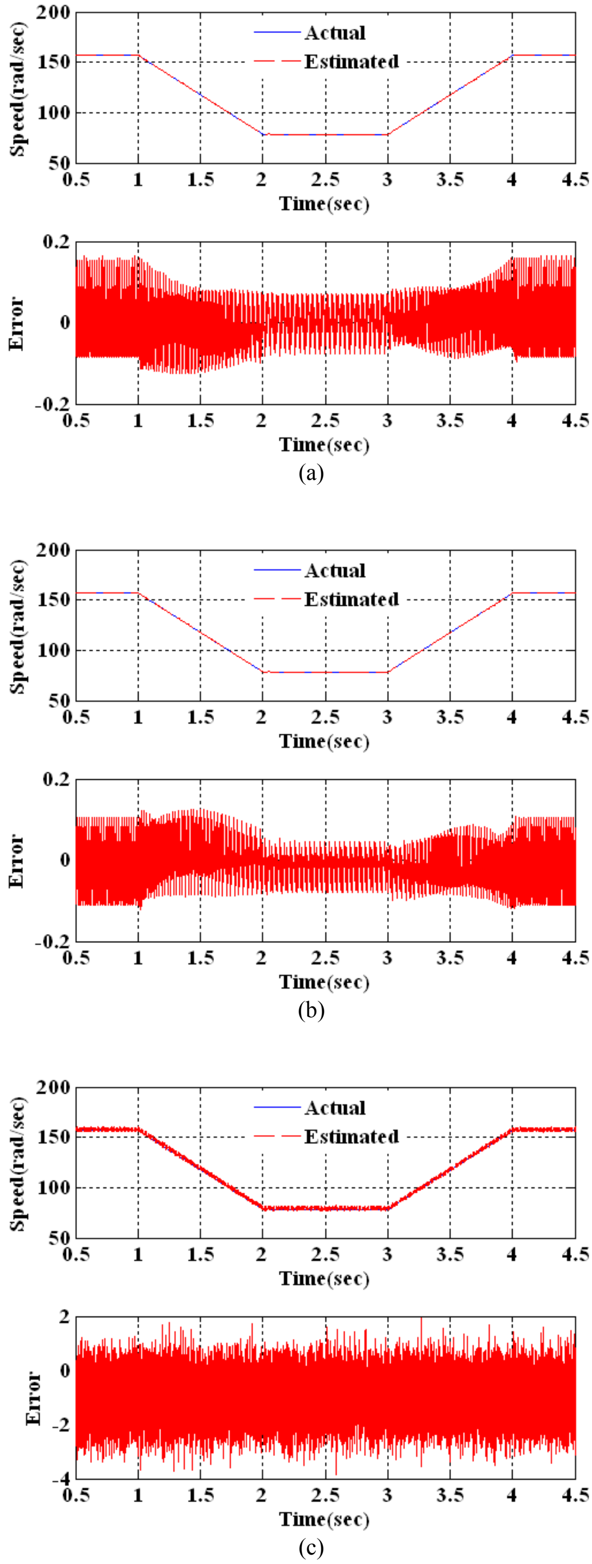


Fig. 8 Operating Condition-II for Ramp Change in Speed: (a) SNC-NN, (b) MLFF-NN (c) SLFF-NN

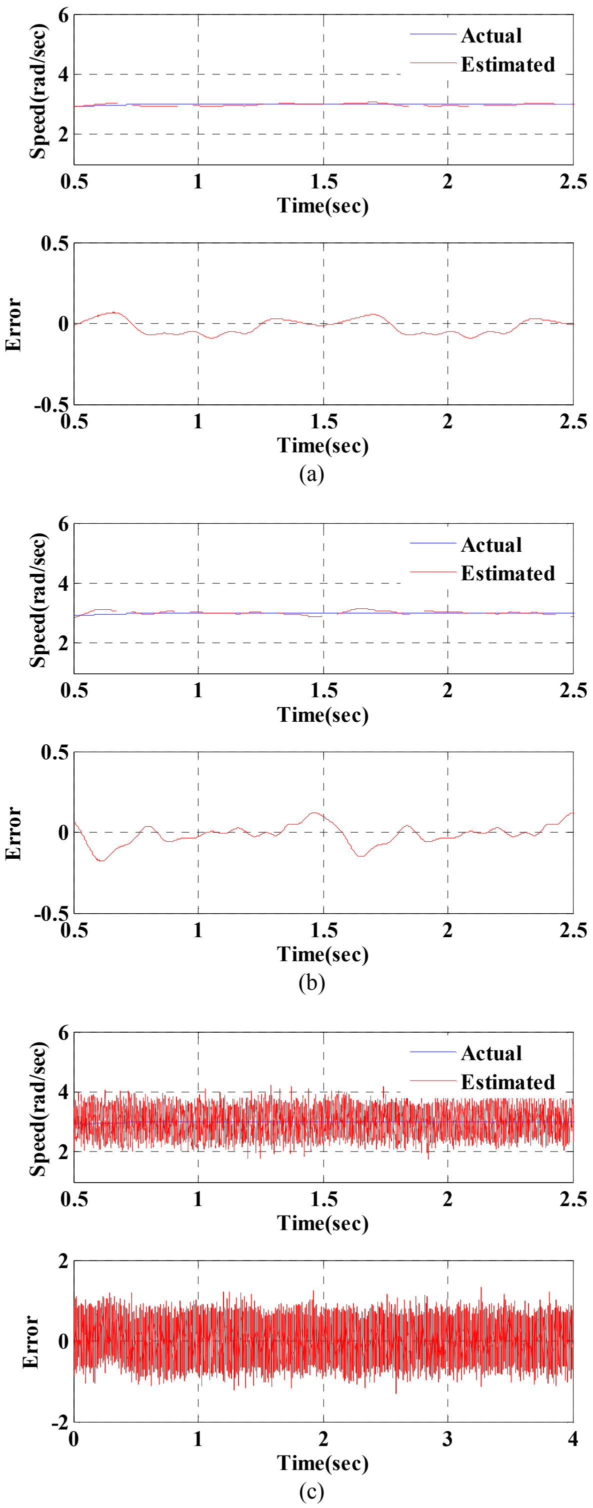


Fig. 8 Operating Condition-III for Low Speed: (a) SNC-NN (b) MLFF-NN (c) SLFF-NN

From the results obtained, it is observed that both the SNC-NN and MLFF-NN model exhibit similar and excellent dynamic performance for transient speed changes where as SLFF-NN model shows poor dynamic performance as compared to SNC-NN and MLFF-NN models.

The test MSE for various operating conditions for all the three NN models is evaluated and the maximum test MSE for all the three NN models are presented in the Table I. From the Table I, it is observed that the test MSE for SNC-NN and MLFF-NN is found to be similar and minimum as compared to SLFF-NN model. From the above analysis, it is understood that SNC-NN and MLFF-NN model have excellent mapping capability as they have multilayer structure when compared to SLFF-NN model. The SLFF-NN model has poor nonlinear mapping capability as it lacks the multilayer structure.

TABLE I
PERFORMANCE COMPARISON OF NN MODELS FOR FLUX ESTIMATION IN TERMS OF ACCURACY

NN Architecture	NN Model	Test MSE		
		Low Speed	Speed Change	Load Change
SNC	**6-15(h)-1**	**0.0023**	**0.0421**	**0.0044**
MLFF	**6-15-15-1**	**0.0042**	**0.2541**	**0.0052**
SLFF	6-75-1	0.1017	3.2381	0.1078

The structural compactness, computational complexity assumes importance in real time implementation to ensure faster execution time for effective control. This motivated the comparison of NN models in terms of structural compactness, computational complexity and execution time.

B. Structural Compactness and Computational Complexity of NN models for On-Line Speed Estimation

The structure of neural network model depends on the number of inputs, number of outputs and the degree of nonlinearity of the system.

The number of neurons in the input/output layer is uniquely defined and is equal to that of inputs/outputs of the system to be modeled. The number of hidden layers, hidden neurons and the type of architecture are the choice of the design for a desired accuracy. For the desired accuracy, the number of hidden neurons is used as an index to measure the structural compactness of model. The neural network model with lesser number of hidden neurons is found to be compact and gives ease in real time implementation of the on-line flux estimator.

The number of parameters and nonlinear function extraction in the network indicates its computational complexity. Each parameter warrants some mathematical operations. The number of parameters for SNC-NN can be calculated using (1). As SLFF-NN is a special case of MLFF-NN with one hidden layer, the same formula (2) suits both type of FF-NN.

$$P_{SNC} = \underbrace{\sum_{m=1}^{M} \sum_{q=0}^{m-1} S^{m} S^{q}}_{Weights} + \underbrace{\sum_{m=1}^{M} S^{m}}_{Biases} \tag{1}$$

$$P_{FF} = \underbrace{\sum_{m=1}^{M} S^{m-1} S^{m}}_{weights} + \underbrace{\sum_{m=1}^{M} S^{m}}_{biases} \tag{2}$$

For on-line speed estimation, the complexity of the model assumes importance as the computation/estimation time has to be small enough for effective control of induction motor drives. The mathematical complexity of the model is compared by determining the number of basic operations needed by the NN model. This will depend upon the type of architecture and number of neurons.

Let N_a be the number of additions, N_m be the number of multiplications and N_{nf} be the number of nonlinear function extractions needed for the model. The time taken in real time by a processor for a given model can be easily computed if the time for the basic operations is known. Let t_a, t_m and t_{nf} be the execution time needed for addition, multiplication and nonlinear function extraction. The total execution time T_{total} can be obtained as

$$T_{total} = N_a \times t_a + N_m \times t_m + N_{nf} \times t_{nf} \tag{3}$$

This general approach helps to determine the execution time for any target processor. In this paper, ADSP-TS101 with operating clock frequency of 250 MHz is used for implementing all the NN models for on-line flux estimation. The execution time in micro seconds for the operations namely addition, multiplication, and non-linear function extraction are presented in Table II [17].

TABLE II
TIME TO EXECUTE MATHEMATICAL FUNCTIONS ON ADSP-TS101

Mathematical operation	Execution Time (μs)
Addition	0.004
Multiplication	0.004
Tan-Sigmoid $\{(e^{n}-e^{-n})/(e^{n}-e^{-n})\}$	0.224

TABLE III
PERFORMANCE COMPARISON OF NN MODELS FOR SPEED ESTIMATION IN TERMS OF STRUCTURAL COMPACTNESS, COMPUTATIONAL COMPLEXITY AND EXECUTION TIME

NN Architectures	NN Models	No. of Hidden Neurons	No. of Parameters	Computations			
				No. of Additions	No. of Multiplications	No. of Tan-sigmoids	Execution Time (μs)
SNC	**6-15(h)-1**	**15**	**232**	**216**	**216**	**15**	**05.09**
MLFF	6-15-15-1	30	361	330	330	30	09.36
SLFF	6-75-1	75	601	525	525	75	21.00

The parameters, neurons, computations and execution time required by the SNC-NN, MLFF-NN and SLFF-NN models are tabulated in Table III. From the Table III, it is seen that SNC-NN model requires much lesser number of hidden neurons (15) as compared to MLFF-NN and SLFF-NN that requires 30 and 75 hidden neurons respectively. Hence SNC-NN model results in structurally compact model as compared to SLFF-NN and MLFF-NN model. The total number of parameters and computations required for SNC-NN is found to be lesser as compared to MLFF-NN and SLFF-NN. Hence, SNC-NN model is of lesser complexity as compared to SLFF-NN and MLFF-NN model. Coding the models on ADSP-TS101, it is found that SNC-NN model is 1.83 and 4.12 times faster as compared to MLFF-NN model and SLFF-NN model respectively. Thus, it can be concluded that SNC-NN architecture gives the most compact and mathematically less complex model with faster execution time for on-line speed estimation.

The overall summary of the paper is detailed as follows:

From the above analysis, it is inferred that the steady state and dynamic performance of SNC-NN and MLFF-NN model are found to be similar and superior as compared to SLFF-NN. The SNC-NN model resulted in structurally compact, computationally less complex model with faster execution time as compared to SLFF-NN and MLFF-NN models. The SNC-NN and SLFF-NN model can be self organized which greatly aids design automation where as MLFF-NN lacks the design methodology. Thus the SNC-NN model is observed to derive the advantage of multilayer mapping capability of MLFF-NN model and self-organizing feature of SLFF-NN model.

Thus, it can be concluded that SNC-NN architecture provides the required accuracy, structurally compact, computationally less complex model with faster execution time. Besides, SNC-NN architecture can be self organized which gives ease in design. Hence, SNC-NN model is identified to be most suitable model for on-line speed estimation in sensorless vector controlled IM drives.

VI. CONCLUSION

The suitability of neural architecture for on-line speed estimation is investigated which is the major contribution of this paper. The on-line speed estimator is modeled using three types of Neural Architectures namely SNC-NN, MLFF-NN and SLFF-NN architectures. For comparison, all the three NN models are trained with same training data, algorithm and same accuracy. On testing, the SNC-NN and MLFF-NN models are found to be superior in terms of accuracy as compared to SLFF-NN model.

The SNC-NN model resulted in structurally compact and computationally less complex model as compared to SLFF-NN and MLFF-NN models. Implementing all the models on ADSP-TS101, SNC-NN model is found to be faster as compared to other two models.

The SNC-NN model combines the advantage of self-organizing feature of SLFF-NN and powerful multilayer non-linear mapping capability of MLFF-NN.

Thus, it can be concluded that SNC-NN model is accurate, simple, self-organizing, structurally compact and computationally less complex and faster in execution time and found to be a promising alternative for online speed estimation in sensor-less IM drives.

ACKNOWLEDGMENT

The research project titled “AI techniques for Electrical Drives” is supported by the grants from the All India Council for Technical Education (AICTE), a statutory body of Government of India. File Number: No 8023/BOR/RID/RPS-79/2007-08 and 8020/RID/TAPTEC-32/2001-02.

APPENDIX

The parameters of the induction machine used for simulation are given in the table shown below.

INDUCTION MOTOR PARAMETERS

Parameters	***Values***	***Parameters***	***Values***
Rated Power	1.1kW	Stator Resistance (R_s)	6.03Ω
Rated voltage	415V	Rotor Resistance (R_r)	6.085Ω
Rated current	2.77A	Magnetizing Inductance (L_m)	0.4893H
Type	3 Ph	Stator Inductance (L_s)	0.5192H
Frequency	50Hz	Rotor Inductance (L_r)	0.5192H
Number of poles	4	Total Inertia (J_T)	0.011787Kgm^2
Rated Speed	1415RPM	Friction Coefficient (B)	0.0027Kgm^2/s

REFERENCES

[1] K. Hurst, T. Habetler, G. Griva, and F. Profumo, “Speed sensorless field-oriented control of induction machines using current harmonic spectral estimation,” *in Conference Record of Industry Applications Society Annual Meeting*, vol. 1, pp. 601–607, *IEEE*, Oct. 1994.

[2] A. Ferrah, P. Hogben-Laing, K. Bradley, G. Asher, and M. Woolfson, “The effect of rotor design on sensorless speed estimation using rotor slot harmonics by adaptive digital filtering using the maximum likelihood approach,” *in Conference Proceedings of the IEEE IAS Annual Meeting*, pp. 128–135, IEEE, Oct. 1997.

[3] M. Rashed and A. F. Stronach, “A stable back-EMF MRAS-based sensorless low speed induction motor drive insensitive to tator resistance variation,” *Proc. Inst. Elect. Eng.—Electr. Power Appl.*, vol. 151, no. 6, pp. 685–693, Nov. 2004.

[4] P. L. Jansen and R. D. Lorenz, “Transducer less field orientation concepts employing saturation-induced saliencies in induction machines*,” IEEE Transactions on Industry Applications*, vol. 32, pp. 1380–1393, Nov/Dec 1996.

[5] M. Schroedl, “Sensorless control of ac machines at low speed and standstill based on the “inform” method,” *in Conference Record of the31st Annual Meeting of the IAS*, vol. 1, pp. 270–277, *IEEE,* Oct. 1996.

[6] S. Maiti, C. Chakraborty, Y. Hori, and M. C. Ta, “Model reference adaptive controller-based rotor resistance and speed estimation techniques for vector controlled induction motor drive utilizing reactive power,” *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 594–601, Feb. 2008.

[7] F. Z. Peng and T. Fukao, “Robust speed identification for speed sensorless vector control of induction machines,” *IEEE Transactions on Industry Applications*, vol. IA-30, no. 5, pp. 1234–1230, 1994.

[8] K.Narendra and K.Part’ms arathy, “Identification and control of dynamical system using neural network” *IEEE Trans. Neural Networks*, vol.1, no.1, pp.4-27, Mar.1990

[9] Karanayil M. Rahman M.F. and Grantham C. “Online Stator and Rotor Resistance Estimation Scheme Using Artificial Neural Networks for Vector Controlled Speed Sensorless Induction Motor Drive” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, Feb 2007.

[10] S.K.Mondal, J.O.P.Pinto and B.K. Bose, “A neural network based space vector PWM controller for a three voltage fed inviter induction motor drive,” *IEEE Trans.Ind.Appl.* vol. 30.no.3.pp 660-669 May 2002.

[11] Maurizio Cirrincione,and Marcello Pucci, “An MRAS-Based Sensorless High-Performance Induction Motor Drive With a Predictive Adaptive Model” *IEEE Transactions On Industrial Electronics*, vol. 52, pp. 532-551, Apr 2005.

[12] P.Vas, “Sensorless and Direct Torque Control”, Oxford University Press, 1998.

[13] Nicholas K.Treadgold and Tramas D. Gedeon, “Exploring Constructive Cascade Networks”, *IEEE Transactions on Neural Networks*, vol. 10, pp.1335-1349, 1999.

[14] K.S.Narendra and K.Parthasarathy, “Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks”, *IEEE Transactions on Neural Networks*, Vol 2, No.2, pp.252-262, March 1991.

[15] Martin T. Hagan and Mohammad B. Menhaj, “Training Feedforward Networks with the Marquardt Algorithm”, *IEEE Transactions on Neural Networks*, Vol 5,No.6, pp.989-993, November 1994.

[16] Martin T.Hagan, Howard B. Demuth and Mark Beale, “Neural Network Design”, Cengage Learning, Pvt.Ltd., India 2008.

[17] Steven W. Smith, “*Digital Signal Processing: a Practical Guide for Engineers and Scientists*”, Cengage Learning, Pvt.Ltd., India 2008.