Artificial Neural Network Based Speed Controller for Induction Motors

F. Lftisi, G.H. George, A. Aktaibi, C.B, Butt, M. A. Rahman
Department of Electrical & Computer Engineering,
Faculty of Engineering & Applied Science, Memorial University of Newfoundland
1 Arctic Ave. St. John's, NL, Canada, A1B3X5
fl55101@mun.ca, glyn@mun.ca, aama38@mun.ca, cbutt@gmail.com, arahman@mun.ca

Abstract— **This paper presents an intelligent indirect field oriented control (IFOC) technique for saturated induction motor (IM) drives in order to achieve high dynamic performance and wide operating range. The IFOC of IM drives has been traditionally carried out using linear proportional-integral (PI) controllers. As an IM is a nonlinear device due to the saturation phenomenon, conventional PI-IFOC methods provide poor performance, limited disturbance rejection capability and longer convergence time. The artificial neural network (ANN) has been widely used as an intelligent controller for nonlinear systems. ANN provides an adaptive learning ability to the controllers to better characterize the system dynamics for achieving accurate and fast responses. However, due to the iterative nature of neural networks, training of the ANN is excessively slow for saturated IM drives. In this paper, a novel neural network map (NNM) is developed to find input weights of the neurons; without the need for any recurrent training process. The proposed technique is applied on a 3-phase 4-pole 208V ¼ hp IFOC-IM drives. Both the simulation and experimental investigation have been carried out for the same motor drive, and the results are depicted and analyzed in this paper. A relative comparison between the PI controller and the proposed NNM based ANN controller indicates that the ANN mapping controller yields superior performance.**

Keywords – Induction motor control, PI, artificial neural network, ANN mapping, saturation-contour.

I. INTRODUCTION

A high performance induction motor (IM) drive system is characterized by an efficient intelligent speed controller, causing the drive to follow the command speed over a range of varying operating conditions. This tracking of command speed is achieved rapidly and precisely regardless of system parameter variations and sudden load excursions. The conventional PI model is widely used for steady state performance results. For transient performances this model does not yield good results due to saturation effects, parameter variations, load disturbances, temperature variations, etc. The consequence is that these PI controllers typically perform well for limited range of specific conditions. On the other hand, the traditional artificial neural network controllers require extensive off-line training, incur high computation time and rely on the past characteristic behavior of the motor drive for a specific system. However, if the performance is unpredictable, the drive encounters parameters outside the training set. [1-2,5-6,9-10]. For the saturated IM drive, an improved ANN mapping model is needed to develop a nonlinear polynomial types PI with more parameters that can give a complex map and provide better fit to responses.

To train a controller based on the integrated squared difference over a set period of time, it is preferable to develop a desired model and train it before uploading the controller to the actual system. For online training it would be best to select the most sensitive parameters based on simulation and adjust only those parameters step by step in time. With fixed weights and biases, these would seem to produce control which is proportional-like and integral type, respectively. However, due to the speed-error based self-tuning of the ANN and the interconnections between layers, the drive, when in closed-loop, will seek command speed even if one of the input weights is fixed. This occurs because the non-fixed weights and biases of any interconnections and neurons remain adaptive and will be continually adjusted as long as speed error exists within limits, if the ANN output command torque differs from the reference torque command. The more complex the network, the less appropriate such a comparison to PI-based control becomes. A network consisting of two neurons is compared to a precise and adaptive PI controller which continually adjusts itself in real-time based on speed error and by comparison to reference torque command.

In applying this technique to this paper, ANN is used only to mimic PI controllers which means fixed suitable weights are needed to follow the desired speed precisely. For comparative evaluation purposes, a PI controller-based IM drive system has also been employed and tested. Furthermore, simulations have been run to assess the performances of both drives at different dynamic operating conditions. Simulation and laboratory results are provided and discussed to validate the efficacy of the proposed improved method.

II. OPTIMAL PI PARAMETERS COMPUTATION METHOD

The speed control of the indirect vector controller in Fig. 1 is performed using a Proportional-Integral (PI) regulator. The outer speed loop is governed by the following equation of motion:

$$\frac{d\omega_r}{dt} + B\omega_r = \frac{P}{2}(T_e - T_l) \qquad (1)$$

The output of the speed controller is the reference torque to the motor. The reference torque is transferred into the inner

978-1-5090-3474-1/16/$31.00 ©2016 IEEE

current control loop. However, while designing the outer speed loop, the inner current control loop can be treated as a unity gain system shown in Fig.1. This is due to the mechanical time constant being very high compared to the electrical time constant. Equation (1) presents a mechanical motor model, which is a first order system. Transforming (1) into the s-domain yields the following transfer function of the system:

$$\omega_r(s) = \frac{K_m}{1+\tau_m s}(T_e - T_l) \tag{2}$$

where the mechanical time constant $\tau_s = \frac{J}{B}$, and gain $k_m = \frac{P}{2B}$. Equation (2) has a settling time of about 5 τ_m . The characteristic equation of the closed-loop system shown in Fig. 1 can be obtained from the following transfer function:

$$\frac{\omega_r}{\omega_r^*} = \frac{K_m K_p s + K_m K_i}{\tau_m s^2 + (1+K_m K_p)s + K_m K_i} \tag{3}$$

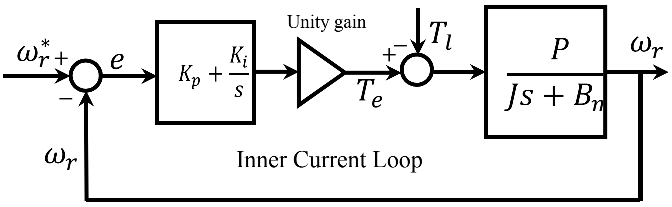

Fig. 1. Inner current loop considered as a unity gain system

The system is assumed to have 1st order response described by,

$$\frac{\omega_r}{\omega_r^*} = \frac{1}{\tau_c s + 1} \tag{4}$$

Where τ_c is a time constant of the closed-loop system. Using (3) and (4), the following parameters of the PI-controller are obtained:

$$K_P = \frac{\tau_m}{K_m \tau_s}, \qquad K_I = \frac{1}{K_m \tau_s} \tag{5}$$

Equation (5) calculates the PI controller parameter with regard to the system time constant, and requires a closed-loop time constant. As a result of the settling time of a first order system is about five times of its time constant. Hence, (5) can be expressed in terms of the settling time as follows:

$$K_P = \frac{5\tau_m}{K_m \tau_s}, \qquad K_I = \frac{5}{K_m \tau_s} \tag{6}$$

where, τ_s is the settling time of the closed loop system. With a settling time of 0.2 seconds and the desired speed of 150 rad/sec, it is clear from the resulting motor speed shown in Fig. 6(a) that the motor speed converges to the desired speed without overshoot [11]. However, these gain parameters give a good performance when only light load is applied to the motor. It will be affected by heavy load. Ziegler-Nichols techniques for adjusting controller parameters are used when the motor is overloaded.

III. CONTROL OF IM WITH ANN CONTROLLER

Fig. 2 shows the schematic block diagram of applying ANN controller based speed drives for field oriented control of induction motors (IM).

The speed sensor on the rotor shaft relays rotor speed information; it is added to the slip in order to calculate rotor position. Along with this and the desired command speed, the input calculator uses phase current readings to determine the appropriate reference command torque. Command speed and speed error are then fed to the ANN-based speed controller. The ANN controller generates the appropriate command torque from which the i_{qs}^* component can be produced. Then the correct i_{ds}^* component is determined from the flux command which is assumed to be constant at the saturation level, and applied to the motor stator via vector rotation calculations and the hysteresis current controller through a voltage source inverter [4, 7-8].

$$i_{qs}^{e*} = \frac{4 L_r T_e^*}{3 P L_m \psi_{r(est)}} \tag{7}$$

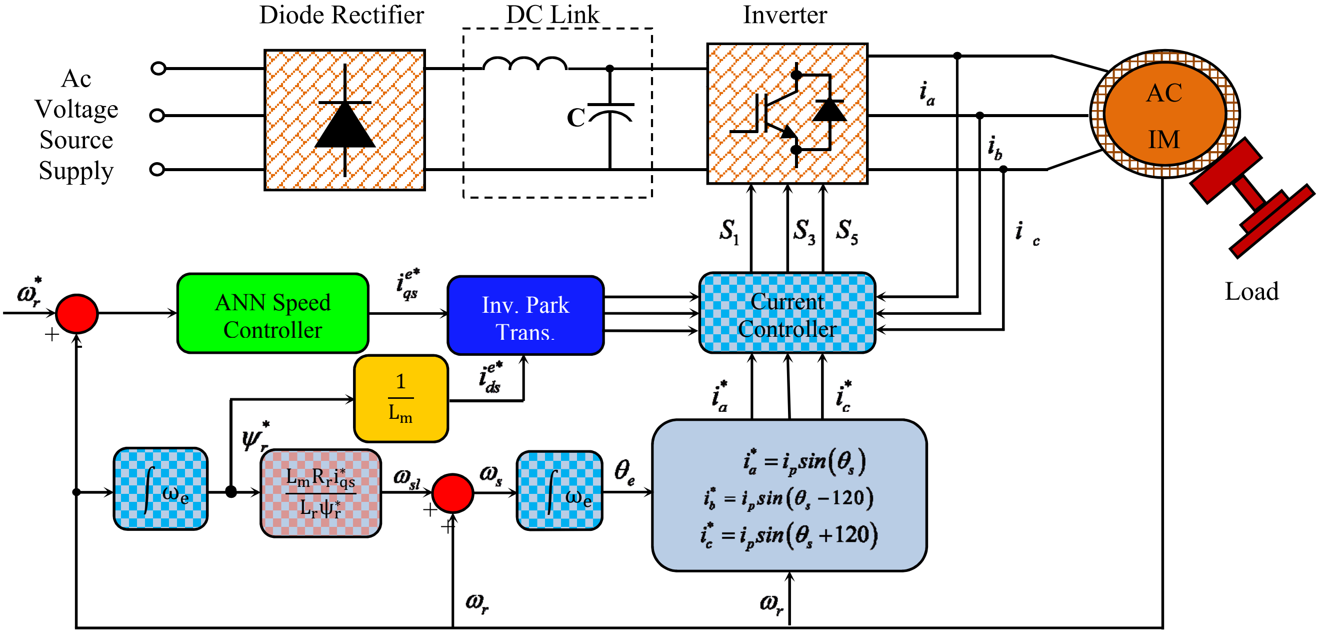

Fig. 2: Illustration of field oriented control scheme using ANN [1-2].

The stator direct-axis current reference i_{ds}^{e*} can be attained from the rotor flux reference ψ_r^* as:

$$\psi_r^* = L_m i_{ds}^{e*} \tag{8}$$

The rotor flux position θ_r required for coordinate transformation is generated from the rotor speed ω_r and slip frequency ω_{sl}.

$$\theta_r = \int(\omega_e + \omega_{sl})dt \tag{9}$$

The slip frequency can be calculated as:

$$\omega_{sl} = \frac{L_m R_r}{\psi_{r(est)} L_r} i_{qs}^{e*} \tag{10}$$

The speed control of the field oriented control in Fig. 1 is achieved using an ANN regulator. The conventional PI controller is replaced by the ANN and the fixed weights of the input neurons are calculated based on corresponding PI gains to mimic PI control.

IV. ANN Structure

The amount of nodes in the input layer depends on the number of inputs to the ANN. Since the aim of using the ANN in this application is to mimic the PI controller, the number of nodes required in the input layer is two (i.e. *i* and *j*); where one is used to represent proportional gain, and the other is used to represent integral gain. The number of hidden layers and the number of neurons in the hidden layer(s) were chosen with the goal of achieving the optimal balance between drive performance and computational complexity. The output of the ANN is command torque, requiring only one neuron in the output layer. Fig. 3 shows the ANN structure based speed drive for the field oriented control of IM. This type of structure was chosen because it is the standard one which allows the creation of the controller with dead band and saturation effect in one input direction and saturation in the other direction. This contrasts with other structures that are not as efficient. For example, one neuron with one input would have allowed the creation of the controller with saturation in one input direction like proportional or integral; or two and more neurons with one input would have allowed the creation of the controller with dead band and saturation effect in one input direction [1,3].

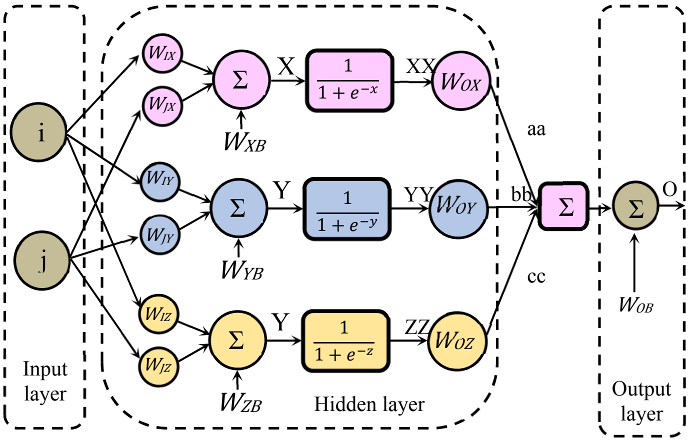

Fig. 3: ANN Structure.

Fig. 4 shows the ANN map controller map, where *i* and *j* represent the proportional and integral errors, respectively, and O is the control signal. Alternately stated, *i* and *j* represent the inputs of the neuron and O is the output of the neuron. The summed inputs into hidden layer neurons are processed by a nonlinear sigmoid function as they pass "through" the neuron. The sigmoid function, or 'S' shape, is used as the squashing function for all neurons and defined as follows:

$$f(i) = \frac{1}{(1+e^{-i})} \tag{11}$$

Information flows forward through the net from input to output. Note that as *i* tends to +∞, the value of $f(i)$ tends to +1, while as *i* tends to -∞, $f(i)$ tends to 0. When *i* is equal to 0 the value of $f(i)$ is 0.5. The sigmoid function $f(i)$ and its derivative with respect to *i*, $f'(i)$, are shown in Fig. 5 and expressed as follows:

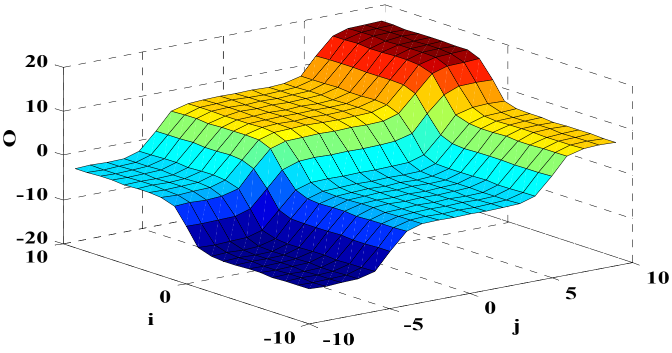

Fig. 4: ANN controller map with saturation.

$$f'(i) = \frac{e^{-i}}{(1+e^{-i})^2} \tag{12}$$

Note that as *i* tends to +∞ or -∞ the value of $\frac{df}{di}$ tends to 0, and when *i* is equal to 0 the value of $\frac{df}{di}$ is ¼. A plot of the $f(t)$ function which illustrates these trends is given below in Fig. 4.

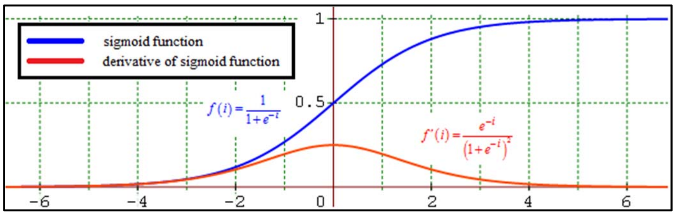

Fig. 5: Sigmoid function and derivative of sigmoid function.

The output of the proposed ANN is expressed as:

$$O = W_{OB} + W_{OX} f\left(W_{IX}I + W_{JX}J + W_{XB}\right) + W_{OY} f\left(W_{IY}I + W_{JY}J + W_{YB}\right) \tag{13}$$

where *I* is the speed error and *J* is the integral of the speed error. The output of a classical PI controller is:

$$O = K_P I + K_I J \tag{14}$$

This controller gives zero output when *I* and *J* are both zero. To get a neural network controller which mimics the PI controller, the conditions in (9) will need to be met:

$$W_{JX} = W_{IY} = 0, \qquad W_{XB} = W_{YB} = 0$$

$$W_{OB} = -(W_{OX} + W_{OY})/2 \tag{15}$$

The saturation limits of the controller are set by the values of W_{OX} by W_{OY} and would be determined by hardware, meaning that W_{OB} would generally be known. The output of the neural network controller reduces to:

$$O = W_{OX} f(W_{IX} I) + W_{OY} f\big(W_{JY} J\big) + W_{OB} \tag{16}$$

Differentiation of the PI controller output with respect to I gives the proportional gain K_P and differentiation of the output with respect to J gives the integral gain K_I. Differentiating the neural network controller output with respect to I when I is zero, and recalling that f is a sigmoid function, gives:

$$\frac{dO}{dI} = W_{OX} \frac{df(W_{IX} I)}{dI} = W_{OX} \frac{1}{4} W_{IX} \tag{17}$$

Setting this equal to the proportional gain gives:

$$\frac{dO}{dI} = K_P \tag{18}$$

$$K_P = W_{OX} \frac{1}{4} W_{IX} \tag{19}$$

Rearranging (19) gives:

$$W_{IX} = 4K_P / W_{IX} \tag{20}$$

Differentiation of the neural network controller output with respect to J when J is zero is gives:

$$\frac{dO}{dJ} = W_{OY} \frac{df(W_{JY} J)}{dJ} = W_{OY} \frac{1}{4} W_{JY} \tag{21}$$

Setting this equal to the integral gain gives:

$$\frac{dO}{dJ} = K_I \tag{22}$$

$$K_I = W_{OY} \frac{1}{4} W_{JY} \tag{23}$$

Rearranging (23) gives the following:

$$W_{JY} = 4K_I / W_{OY} \tag{24}$$

The above tuning method produces a neural network controller with initial weights and biases that mimic a PI controller, but with the traditional constant gains K_P and K_I replaced with non-linear sigmoid functions. A set of weights was used to construct a suitable map which makes the neural network have more degrees of freedom and a more local character.

V. Simulation Results and Discussion

To validate the efficiency of the ANN based speed controller for the IM drive, hysteresis PWM control algorithms are generated. The aim of using ANN is to create a suitable map. This map has to have more degrees of freedom and a more local character. To train a controller based on the integrated squared difference over a set period of time, it is preferable to develop a model of the system and train it before uploading the controller to the actual system. For online training it is best to select the most sensitive parameters based on simulation and adjust only those parameters step by step in time. However, the ANN controller map must be able to mimic the conventional controller (PI) when there is no change in its parameters.

For comparative evaluation purposes, a PI controller-based IM drive system has been implemented and tested. Further, simulations were run to evaluate the performances of both, the ANN and PI techniques, at different operating conditions. The performances of the IM drive for both the conventional and proposed speed controllers are investigated for step changes in command speeds and load torque.

The simulated performances of the IM drive system are shown in Figs. 6 to 11. Figs. 6 and 7 show the simulated speed responses for ANN and PI based speed controller performances respectively, when the rated load is applied to the IM. It should be noted that the two responses follow the tracking of the command speed and the settling time is fast, within 0.2s. It is evident from Figs. 6, the ANN-based speed drives have a capability to reject the load disturbance applied to the IM, similar to the conventional PI speed controller. Figs. 8 and 9 show the torque responses of both controllers with rated load applied at 1 sec. The corresponding stator currents of the ANN and PI controller-based IM drives are shown in Figs. 10 and 11, respectively.

From these results, it is clear that the ANN controller has been successfully tuned, via the off-line PI-referenced method presented in this paper, to exhibit performances equal to those obtained with the conventional PI controller upon which it is based. The results highlight that under the conditions presented here, the ANN controller is, in fact, producing control signals in agreement with the reference PI controller.

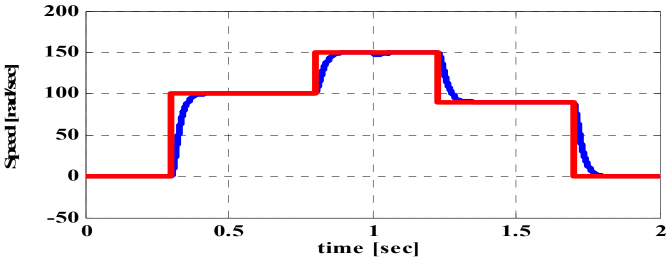

Fig. 6: Simulated response of the ANN maps controller-based IM to step changes of command speed with applying rated load.

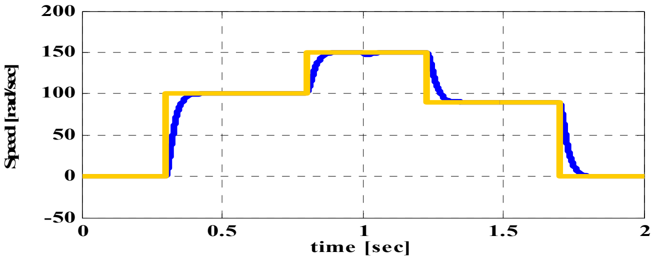

Fig. 7: Simulated response of the PI controller-based IM to step changes of command speed with applying rated load.

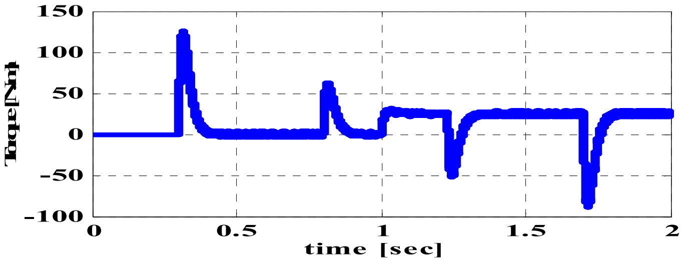

Fig. 8: Simulated torque of the ANN map controller-based IM drive under the application of rated load.

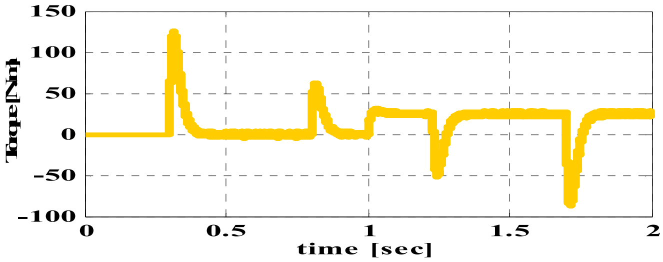

Fig. 9: Simulated torque of the PI controller-based IM drive under the application of rated load.

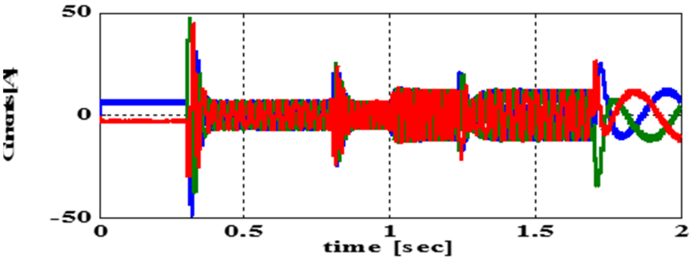

Fig. 10: Simulated currents of the ANN map controller-based IM drive under the application of rated load.

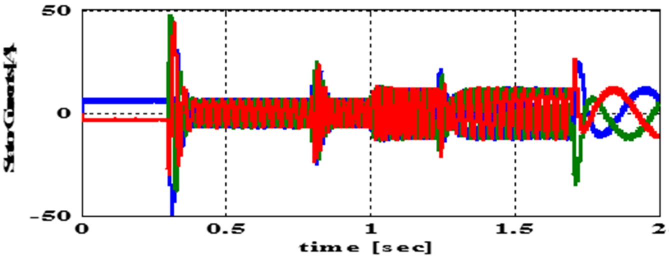

Fig. 11: Simulated currents of the PI controller-based IM drive under the application of rated load.

VI. Experimental Drive Configuration

In this work, Matlab/Simulink was used to simulate the IM drive based on ANN and experimentally implemented using the dSPACE DSP board DSS1104 for a laboratory IM of 1/4 HP. Fig 12. shows a snapshot of the experimental setup. The DSP board DS1104 was incorporated into a computer desktop through a dual port memory. The DS1104 utilized is a standard controller board that was plugged into a PCI slot of the personal computer. The DSP is supplemented by a set of on-board peripherals used in digital control systems including analog to digital (A/D), digital to analog (D/A) converters and digital incremental encoder interfaces. Based on a 603 Power PC floating-point processor, it is a complete real-time system running at 250 MHz. Also, the board includes a slave-DSP subsystem based on the TMS320F240 DSP microcontroller. This is a 16-bit micro controller DSP providing the necessary digital I/O ports and timer function for operations such as PWM generation, input capture, and output capture. The master PPC debug connector (P2) was also used for debugging. Where the PC-based controller produces numerical switching commands to the DSP board, the outputs of the DSP board are sent to the IGBT circuit to drive the VSI inverter. Hall-effect sensors are used to measure the motor current, which is fed back to the DSP board through the A/D channels. The rotor position is detected with the use of an optical incremental encoder that is mounted on the shaft with 2500-line resolution, which is then fed back to the DSP board through an encoder interface.

The D/A channels are used to provide the desired signals outputs to be shown in the oscilloscope. The IM is coupled to a DC machine which is operated as a generator in order to adjust the mechanical load to the tested motor. A Simulink model was developed and downloaded to the DSP board utilizing dSPACE Control Desk software, in order to carry out a real-time testing. Furthermore, so as to perform the vector control algorithm, the hysteresis controller was used as current controllers.

The command currents were generated from the ANN. These currents are compared with the corresponding actual motor-currents by the hysteresis current controller, thereby generating the logic signals that act as firing pulses for the inverter switches. Therefore, six pulses are produced from the DSP board and fed to the base drive circuit of the inverter power module. The dSPACE real-time interface (RTI) unit has been used to trigger the IGBT gates of the inverter so as to supply the IM with appropriate voltage and frequency.

The sampling frequency utilized is 15 kHz. First, the real-time Simulink graphics model is developed, after which the ANSI 'C' code is generated by the real-time workshop (RTW). The C codes were compiled by the TI C compiler and the program was downloaded to the DSP board using a loader program.

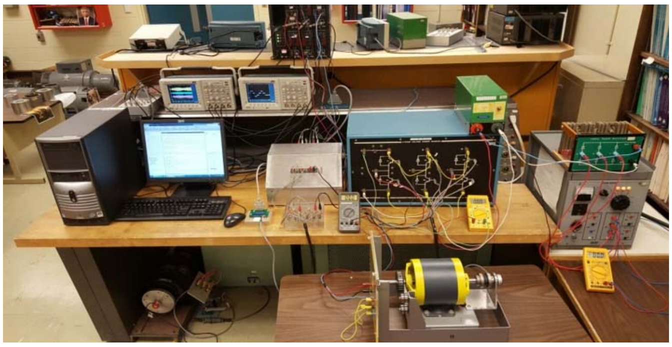

Fig. 12: Response of the ANN controller-based IM to step changes of command speed without load.

I. Experimental Results and Discussion

The control algorithm for the ANN based IM drive was implemented through software by developing a program in ANSI C language. After initializing the required variables, the initial set of weights and biases gotten from the offline training of the ANN structure was downloaded.

This work evaluated the performances of the proposed ANN based drive system. The results obtained for the ANN map controller is shown in Fig. 13 and the result for the PI controller is shown in Fig. 14.

The results show that the proposed drive system has the ability to follow the reference speed under no load conditions. It should be mentioned that the effect of sudden load impact on dynamic speed response and the evaluation of speed with parameter variation of the ANN map based IM drive were performed solely by computer simulation as depicted in Figs. 6, 7. It can be noted for these experimental results that the performance of the ANN map controller is more accurate in following the command speed compared to that of the PI controller. The results show that under the conditions the ANN map controller is producing better control signals by 2%, which is in agreement with the reference PI controller.

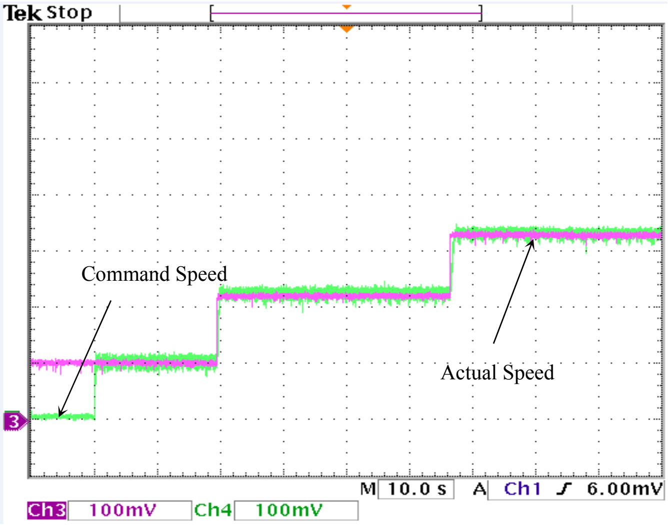

Fig. 13: Experimental speed response of the drive for the step changes of the command speed based on ANN controller, (div=10 rad/sec)

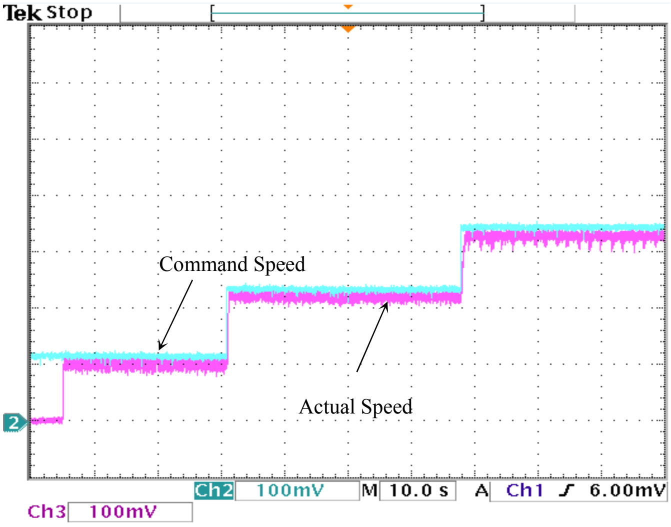

Fig. 14: Experimental speed response of the drive for the step changes of the command speed based on PI controller, (div=10 rad/sec)

CONCLUSION

It is shown that a set of weights has been derived using mathematical reference to a conventional PI controller and used to create a suitable mapping, which makes the artificial neural network with two neurons only have more degrees of freedom. ANN structures for this application are found to produce acceptably fast and accurate command speed tracking. This is due to their precision, ease of implementation, capacity to produce both positive and negative outputs, and quick convergence rates when used in the ANN for IM drives. Simulation results reveal some of these interesting features and show that the ANN network is good for use as an alternative to the conventional PI control of induction motors. Both simulation and laboratory test results are given to verify the validity of this method.

ACKNOWLEDGEMENTS

The authors are grateful to Professor Michael Hinchey, Memorial University, Canada, for his useful suggestions on this approach including nonlinear mapping.

APPENDIX

Parameters of the 3Φ, squirrel-cage induction motor used for Experimental work, it's Rated power $\mathrm{P_{rat}} = 0.25\,\mathrm{Hp}$, Rated stator voltage $\mathrm{V_{rat\,L-L}} = 208\,\mathrm{V}$, $f = 60\,\mathrm{Hz}$, Rated speed $\omega_r = 188.5\,\mathrm{rad/sec}$, Stator Resistance $\mathrm{R_s} = 12.5\,\Omega$, Stator leakage inductance $L_{ls} = 0.03611\,\mathrm{H}$, Rotor Resistance $\mathrm{R_r} = 3.833\,\Omega$, Rotor leakage inductance $L_{lr} = 0.03611\,\mathrm{H}$, Mutual inductance $\mathrm{L_m} = 0.4955\,\mathrm{H}$, Number of pole pairs $\mathrm{P} = 2$.

REFERENCES

[1] C.B. Butt and M. A. Rahman, "Intelligent Speed Control of Interior Permanent Magnet Motor Drives Using a Single Untrained Artificial Neuron," *IEEE Transactions on Industry Applications*, vol. 49, no. 4, pp. 1836-1843, July-August 2013.

[2] M. A. Rahman and M.A. Hoque, "On-Line Adaptive Artificial Neural Network Based Vector Control of Permanent Magnet Synchronous Motors," *IEEE Transactions on Energy Conversion*, vol. 13, no. 4, pp. 311-318, January-February 1998.

[3] M. Hinchey, "Experimental Methods Engineering 9211 Notes", 2012. Memorial University; Newfoundland.

[4] B. K. Bose, "Modem Power Electronics and AC Drives"; Prentice-Hall, 2002.

[5] A. Ba-Razzouk, A. Cheriti, G. Olivier, P. Sicard, "Field-oriented control of induction motors using neural-network decouplers,", *IEEE Trans. on Power Electronics*, vol. 12, no.4, pp.752-763, July 1997.

[6] Apoorva Saxena, Sayak Dutta, A. Chitra,"Artificial Neural Network Controller for Vector Controlled Induction Motor Drive", International Journal of Computer Applications, May 2012.

[7] P.C. Sen, "Electric Motor Drives and Control - Past, Present and Future", *IEEE Trans, on Industrial Electronics*, vol. 37, no. 6, pp. 562-575, Dec. 1990.

[8] R. Krishnan, Electric Motor Drives: modeling, analysis and control. Upper Saddle River: NJ: Prentice-Hall, 2002.

[9] M.H. Jokar, B. Abdi, M. Ardebili, "Vector Control of Induction Motors Using Radial Basis Function Neural Network,". *IEEE International Symposium on in Diagnostics for Electric Machines*, Power Electronics and Drives, pp.215-218, 6-8 Sept. 2007.

[10] M.G. Simoes; B.K. Bose, "Neural network based estimation of feedback signals for a vector controlled induction motor drive," IEEE Transactions on Industry Applications, vol.31, no.3, pp. 620-629, May/Jun 1995.

[11] M. O. Ajangnay, "Optimal PID controller parameters for vector control of induction motors.," SPEEDAM 2010, Pisa, 2010, pp. 959-965.