A modified type-2 neuro-fuzzy SVM-based inverter fed IM drive

G. Srinivas*

Vignan's Foundation for Science,
Technology and Research University,
Guntur, 522213, India
Email: gadde.cnu@gmail.com
*Corresponding author

G. Durga Sukumar

Vignan Institute of Technology and Science,
Deshmukhi (V), Yadadri (dist), Telangana, 508284, India
Email: durgasukumar@gmail.com

Abstract: In indirect vector control, the conventional speed and current controllers operate satisfactorily when the operating point is constant. But the operating point is always dynamic; the reference voltages obtained in a closed-loop system feeding to the inverter contains more harmonics. Due to this, the pulses which are going to produce are uneven and in turn, understudy creates the inverter output voltages produced with more harmonics. In order to produce a better output voltage from the inverter, this paper presents the neuro-fuzzy type-2 (NFT2) space vector modulation technique. Performance comparison of the inverter is done with conventional space vector modulation (SVM) and neuro-fuzzy type-1 (NFT1) system using MATLAB simulation and experimental validation. The performance parameters of the induction motor based on current, torque, and speed with neuro-fuzzy type-2 space vector modulation is compared with conventional and type-1 neuro-fuzzy SVM. The % THD of inverter output voltages are also compared. The experimental validation of a DSpace-1104 is used to analyse the performance of the induction motor which is obtained by the simulation. The experimental validations are carried out considering 2HP induction motor in the lab.

Keywords: indirect vector control; neuro-fuzzy type-2; NFT2; neuro-fuzzy type-1; NFT1; space vector modulation; SVM; induction motor; IM; third harmonic distortion; THD.

Reference to this paper should be made as follows: Srinivas, G. and Sukumar, G.D. (2022) 'A modified type-2 neuro-fuzzy SVM-based inverter fed IM drive', *Int. J. Power Electronics*, Vol. 15, Nos. 3/4, pp.267–289.

Biographical notes: G. Srinivas received his BTech in Electrical and Electronics Engineering from the JNTU Hyderabad, in 2006 and MTech in Power and Industrial Drives from JNTU Anantapur, in 2010. He is currently pursuing his PhD from the Department of Electrical and Electronics Engineering in Vignan's Foundation for Science, Technology and Research University, with the specialisation of power electronics and drives. His research interests are in power electronics and drives, neural networks and fuzzy logics and new research eras in renewable energy sources.

G. Durga Sukumar received his Bachelor's and Master's in Electrical Engineering from the JNTU, Hyderabad, India, and PhD in Electrical Engineering from Indian Institute of Technology, Roorkee, India. Currently, he is working as a Professor in the Vignan Institute of Technology and Science, TS, India. His research interests include power electronics, AC drives and soft computing techniques.

1 Introduction

The various pulse width modulation (PWM) techniques are used to control output voltages generated from voltage source inverter (VSI). But in this mode of operation gives more switching losses and produce more harmonics in output waveform with less efficiency. To avoid above drawback, this paper introduces space vector modulation (SVM) controller with neuro-fuzzy (NF) techniques. In this method, the linear operation of SVPWM increased with modulation index ≤ 0.907.

The turning points are generated by space vector instants that are used for applying the pulse width modulated VSI. Due to this, the switching times are reduced and current and torque ripple is minimised as compared to the basic sinusoidal method (Van Der Broeck et al., 1988). The digital implementation is a method that is used to develop a transient simulation for both linear and nonlinear modes of operation. SVM is a technique to implement proper utilisation of bus voltage and support to harmonic spectrum used for modern applications (Mehrizi-Sani and Filizadeh, 2006). The adaptive neuro-fuzzy interference system (ANFIS)-based MPPT is proposed for induction motor (IM) drive to reduce ripple in torque and current in MATLAB (Simulink) and it is verified with the experimental setup using the hardware Dspace(1104) (Pakkiraiah and Sukumar, 2017). The type-2 fuzzy logic direct torque control (DTC) technique is implemented with the replacement of proportional-integral controllers. The control method is used to improve the response in both transient and steady-state. It also reduces the flux distortion and torque ripple as compared to the conventional DTC under different operating conditions (Nai and Singh, 2014). The adaptive NFIS current controllers that are employed for an indirect vector of inverter fed IM is used to reduce the torque ripple without any filter as compared to conventional PI controllers. The performance of the drive is simulated under different operating conditions (Sukumar and Pathak, 2012a). The variable frequency modulation to VSI fed IM for minimising the switching losses and output voltage distortion from the developed SVM algorithm (Attaianese et al., 2007). The NF torque controller is used to reduce torque ripple instead of PI controller without any filter. The SVM algorithm is also proposed but there is need information for calculation of sector and angle (Sukumar and Pathak, 2012b). The *n*-level VSI fed inverter with current error, a space vector hysteresis as the current controller is employed for the error within the boundary. The hysteresis controller has an advantage of a smooth transition from linear to over modulation and it is also been verified under experimental validation with steady and transient performances (Dey et al., 2013). The type-2 fuzzy-based method has been employed for VSI fed IM drive. This method is independent of switching frequency and performance has been compared with conventional SVM (Sukumar et al., 2012c). The ANFIS-based SVM does not depend on switching frequency and needed training error, which can be estimated by using the SVM

algorithm. Due to this, the dynamic performance of an IM is improved with a smaller number of iterations as compared to other optimisation techniques like genetic, neural and fuzzy (Sukumar and Pathak, 2011a). The control of torque in steady and dynamic state can be achieved by developing a technique that is constant switching frequency. It uses flux error vector-based SVM for estimating the torque ripple &angular velocity (Tripathi and Ashwin, 2005). The ANN SVM-based fed VSI is used for implementation of fast switching frequency resulting in dynamic operation of IM drive under linear region to square wave (Mondal et al., 2000). The NF-based SVM with three-level inverter fed VSI is used to implementation of improved steady and dynamic performance of the IM drive. In the proposed method, the input is given as a space vector angle and change of space vector angle to generate output with desired duty ratios. The neural network (NN) is used for fast implementation of SVM algorithm with specific IC chip (Sukumar et al., 2014; Mondal et al., 2002). The ANN is proposed with SVM-based VSI fed IM drive is independent of switching frequency and different outputs have been estimated with total neurons that are constant (Muthuramalingam et asl., 2005). The NF SVM-based inverter fed IM with a five-layer network, receives inputs V_{ds} and V_{qs} and generates output, which is trained duty ratios. For different switching frequencies, the third harmonic distortion (THD) calculation can be estimated by simulation and experimental validation (Sukumar and Pathak, 2011b).

Section 2 explains the IM mathematical modelling. Section 3 discusses the two-level inverter with mathematical modelling. Section 4 demonstrates the neuro-fuzzy type-2 (NFT2)-based SVM. In Section 5, the results and discussion are compared with the conventional SVM. Section 6 presents the conclusions of the study.

2 IM mathematical modelling

The mathematical modelling of a three-phase squirrel cage IM developed in direct-quadrature (d-q) stationary reference as follows.

$$\frac{di_{ds}}{dt} = -\frac{1}{\sigma L_s}\left(R_s + \frac{L_m^2}{L_r^2}R_r\right)i_{ds} + \frac{1}{\sigma L_s}\frac{L_m R_r}{L_r^2}\Psi_{dr} + \frac{PL_m}{\sigma L_s L_r}\omega_r\Psi_{qr} + \frac{V_{ds}}{\sigma L_s} \tag{1}$$

$$\frac{di_{qs}}{dt} = -\frac{1}{\sigma L_s}\left(R_s + \frac{L_m^2}{L_r^2}R_r\right)i_{qs} + \frac{1}{\sigma L_s}\frac{L_m R_r}{L_r^2}\Psi_{qr} - \frac{PL_m}{\sigma L_s L_r}\omega_r\Psi_{dr} + \frac{V_{qs}}{\sigma L_s} \tag{2}$$

$$\frac{d\Psi_{dr}}{dt} = -\frac{R_r}{L_r}\Psi_{dr} - P\omega_r\Psi_{qr} + \frac{L_m R_r}{L_r}i_{ds} \tag{3}$$

$$\frac{d\Psi_{qr}}{dt} = -\frac{R_r}{L_r}\Psi_{qr} - P\omega_r\Psi_{dr} + \frac{L_m R_r}{L_r}i_{qs} \tag{4}$$

$$\frac{d\omega_r}{dt} = -\frac{B}{J}\omega_r + \frac{1}{J}(\tau_e - \tau_l) \tag{5}$$

where

$\sigma = \left(1 - \frac{L_m^2}{L_s L_r}\right)$ represents the leakage coefficient

i_{ds} and i_{qs} indicates the d-q component of stator currents

R_s and R_r indicates stator and rotor resistances

V_{ds} and V_{qs} indicates the d-q component of stator voltages

L_s and L_r indicates stator and rotor inductances

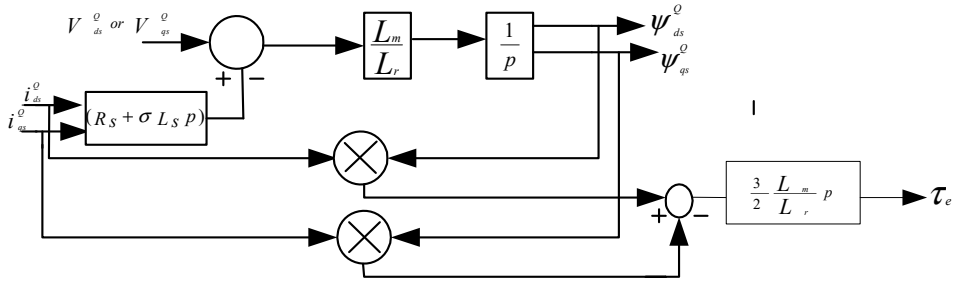
Ψ_{dr} and Ψ_{qr} indicates the d-q component of rotor fluxes

L_m, ω_r and P indicates magnetising inductance, rotor speed and no. of poles.

The produced instantaneous electromagnetic torque expression for the IM is given as follows:

$$\tau_e = \frac{3}{2}\frac{L_m}{L_r} P\left(\Psi_{dr} i_{qs} - \Psi_{qr} i_{ds}\right) \tag{6}$$

Figure 1 IM toque and flux estimator flow diagram



3 Two-level inverter with mathematical modelling

Space vector is a technique is used to control the PWM used in variable speed drives.

Figure 2(a) shows that the connection diagram of the two-level inverter and Figure 2(b) shows eight switching vectors with space vector rotation. These different vectors are used for switching states of a two-level inverter. The resultant voltage is zero when V_0 and V_7 vectors are selected. The effective voltage is fed to the induction machine by selecting the remaining vectors that are V_1 to V_6.

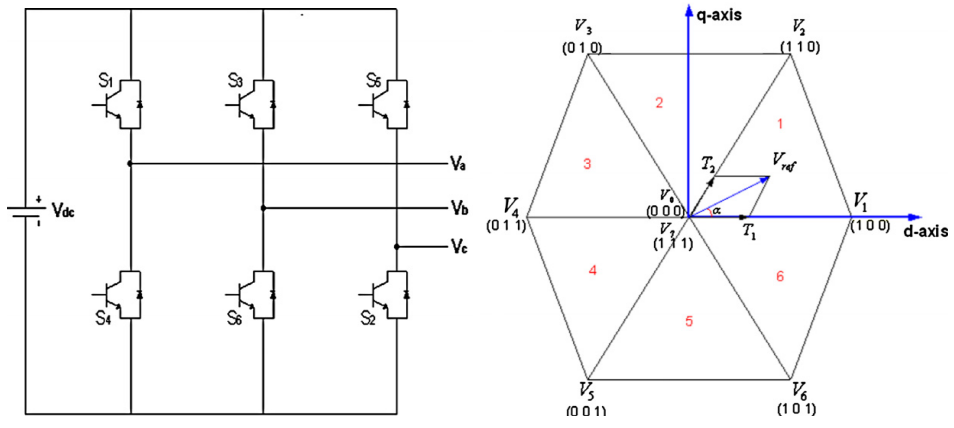
The reference voltage (V_{ref}) with a constant value is generated at an angle of α by using zero vectors (V_0 and V_7), in the combination of the nearest two active vectors V_n, V_{n+1}. Effective vectors are used to obtain the desired output by using two active vectors.

From the above concept, the mean voltage criteria, the reference vector in unit sampling time can be stated as

$$V_{ref} = \left(T_1.V_n + T_2.V_{n+1}\right) / T_s \tag{7}$$

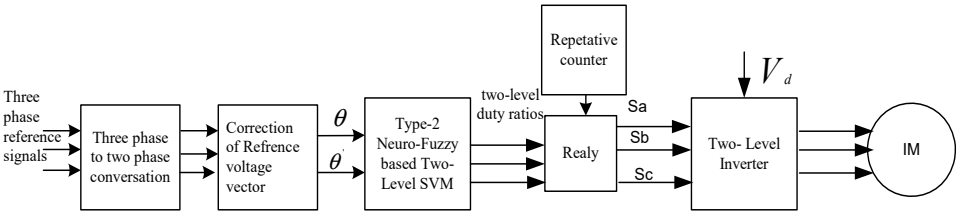
T_1, T_2 are represents the V_1–V_6 sector's active times.

Figure 2 (a) Two-level inverter 2 (b) Space vector diagram with active vectors (see online version for colours)



(a) (b)

Figure 3 Proposed NFT2-based SVM for inverter control



Along the direct axis, while equating the equations is

$$V_{ref}\cos\alpha.Ts = V_{dc}.T_1 + \left(V_{dc}.\cos\frac{\pi}{3}\right)T_2 \tag{8}$$

Along the quadrature axis, while equating the equations is

$$V_{ref}\sin\alpha.Ts = V_{dc}.T_1 + \left(V_{dc}.\sin\frac{\pi}{3}\right)T_2 \tag{9}$$

where α is the angle of V_{ref} in a 600 sector concerning the beginning of the sector, and V_{dc} is the magnitude of each active vector (V_1–V_6).

$$T_1 = M.\frac{\mathrm{Sin}\left(\frac{\pi}{3}-\alpha\right)}{\mathrm{Sin}\left(\frac{\pi}{3}\right)}\cdot T_s \tag{10}$$

$$T_2 = M.\frac{\mathrm{Sin}(\alpha)}{\mathrm{Sin}\left(\frac{\pi}{3}\right)}\cdot T_s \tag{11}$$

$$T_o = T_s - T_1 - T_2 \tag{12}$$

The duration of zero vector with the sampling period represents time *Ts*.*M* represents modulation index $M = \frac{V_{ref}}{V_{dc}}$. When time is equally distributed, then the ripple value is reduced to zero.

4 NFT2-based SVM

The IF-THEN rules are used to implement a NFT2 system, and it involves rules of antecedent and consequent parts with type-2 fuzzy values. In Gaussian type-2 fuzzy sets, uncertainties can be associated with the mean and the standard deviation. Gaussian type-2 fuzzy sets with uncertain standard deviation and uncertain mean are shown in the structure of the type-2 fuzzy neural system.

Layer 1 *Input layer:* This layer gives inputs to the fuzzification layer, and this layer connects as many number of nodes as the no. of facial features *m*. The input layer of output is given as

$$u_i^t = x_{i(\theta)}^t, i = 1, 2, \ldots, m \tag{13}$$

Layer 2 *Fuzzification layer:* Each node calculates the membership of input data with the rule of antecedents by employing the interval type-2 Gaussian membership function. The membership of the i^{th} input feature with k^{th} rule is given thereby.

$$\varnothing_{ik(\theta)}^t = \exp\left(-\frac{\left(\mu_1^t - \mu_{ik(\theta)}\right)^2}{2\sigma_{k(\theta)}^2}\right) = \varnothing\left(\mu_{ik(\theta)}, \sigma_{k(\theta)}, \mu_i^t\right) \tag{14}$$

$$\varnothing_{1k(\theta)}^{lo} = -\exp\left(\frac{\left(\mu_1^t - \mu_{1k(\theta)}\right)^2}{2\sigma_{k(\theta)}^2}\right), \quad \varnothing_{2k(\theta)}^{lo} = -\exp\left(\frac{\left(\mu_2^t - \mu_{2k(\theta)}\right)^2}{2\sigma_{k(\theta)}^2}\right) \tag{15}$$

$$\varnothing_{k(\theta)}^{lo} = -\left(\frac{\left(\mu_1^t - \mu_{1k(\theta)}\right)^2 + \left(\mu_2^t - \mu_{2k(\theta)}\right)^2 + \cdots + \left(\mu_n^t - \mu_{nk(\theta)}\right)^2}{2\sigma_{k(\theta)}^2}\right) \tag{16}$$

$$\varnothing_{k(\theta)}^{up} = -\left(\frac{\left(\mu_{1k(\theta)} - \mu_1^t\right)^2 + \left(\mu_{2k(\theta)} - \mu_2^t\right)^2 + \cdots + \left(\mu_{nk(\theta)} - \mu_n^t\right)^2}{2\sigma_{k(\theta)}^2}\right) \tag{17}$$

where $\mu_{ik(\theta)} \in [{}^1\mu_{ik(\theta)}, {}^2\mu_i^t]$ are the left and right limits of the centre and σ_k is the width of k^{th} rule.

The footprint of the uncertainty of this membership function can be represented as a bounded interval in terms of upper membership function $\varnothing^{up}$ and lower membership function $\varnothing^{lo}$ as given below

$$\varnothing_{ik(\theta)}^{up,t} = \begin{cases} \varnothing\left({}^{1}\mu_{ik(\theta)}, \sigma_{k(\theta)}\right), \mu_i^t \mu_i^t < {}^{1}\mu_{ik(\theta)} \\ 1/2 \quad {}^{1}\mu_{ik(\theta)} \le \mu_i^t \le {}^{2}\mu_{ik(\theta)} \\ \varnothing\left({}^{2}\mu_{ik(\theta)}, \sigma_{k(\theta)}\right), \mu_i^t \mu_i^t < {}^{2}\mu_{ik(\theta)} \end{cases} \tag{18}$$

$$\varnothing_{ik(\theta)}^{lo,t} = \begin{cases} \varnothing\left({}^{1}\mu_{ik(\theta)}, \sigma_{k(\theta)}\right), \mu_i^t \mu_i^t \le \dfrac{\left({}^{1}\mu_{ik(\theta)} + {}^{2}\mu_{ik(\theta)}\right)}{2} \\ \varnothing\left({}^{2}\mu_{ik(\theta)}, \sigma_{k(\theta)}\right), \mu_i^t \mu_i^t > \dfrac{\left({}^{1}\mu_{ik(\theta)} + {}^{2}\mu_{ik(\theta)}\right)}{2} \end{cases} \tag{19}$$

The output of each node can be represented by the interval

$$\varnothing_{ik(\theta)}^{t} = \left[\varnothing_{ik(\theta)}^{lo,t}, \varnothing_{ik(\theta)}^{up,t}\right]. \tag{20}$$

Layer 3 *Firing layer:* In this layer, each node calculates the firing strength of a rule. It consists of 2 × *K* nodes where *K* nodes represent upper and lower firing strength of *K* rule. Then, the algebraic product of operation is applied as the rule antecedent to calculate the firing strength of a rule, and it is given by

$$\left[F_{k(\theta)}^{lo,t}, F_{k(\theta)}^{up,t}\right]; k = 1, \ldots\ldots, K = \min\left[F_{k(\theta)}^{lo,t}, F_{k(\theta)}^{up,t}\right] \tag{21}$$

Substitute equation (21)

$$F_{k(\theta)}^{t} = \sum\nolimits_{i=1}^{m} \frac{1}{2\sigma_{k(\theta)}^2}\left[\left(\mu_i^t - \mu_{ik(\theta)}\right)^2, \left(\mu_{ik(\theta)} - \mu_i^t\right)^2\right] \tag{22}$$

Layer 4 *Type reduction layer:* This layer consists of *K* nodes. In this layer, each node performs type reduction of interval type-1 fuzzy set to a fuzzy number employing a variant of Nie-Tan type-reduction. This approach is closed-form approximation of the well-known Karnik-Mendel algorithm. The output of each node is given as

$$F_{k(\theta)}^{t} = \alpha F_{k(\theta)}^{lo,t} + (1-\alpha) F_{k(\theta)}^{up,t}, \quad k = 1, \ldots\ldots, K \tag{23}$$

$$F_{k(\theta)}^{t} = \alpha\left[-\left(\frac{\left(\mu_1^t - \mu_{1k(\theta)}\right)^2}{2\sigma_{k(\theta)}^2}\right)\sum\nolimits_{i=1}^{m} + (1-\alpha)\sum\nolimits_{i=1}^{m} \frac{1}{2\sigma_{k(\theta)}^2}\left(\mu_{ik(\theta)} - \mu_1^t\right)^2\right] \tag{24}$$

$$F_{k(\theta)}^{t} = 1 - \alpha\left[\sum\nolimits_{i=1}^{m} \frac{\left(\mu_i^t - \mu_{1k(\theta)}\right)^2 + \left(\mu_{1k(\theta)} - \mu_1^t\right)^2}{2\sigma_{k(\theta)}^2}\right] \tag{25}$$

where α is the weighted measure of uncertainty and is set as 0.5.

Layer 5 *Normalisation layer:* This layer consists of *K* nodes, and each node normalises the firing strengths of a rule that is generated by the type reduction layer:

$$\overline{F_{k(\theta)}^t} = \frac{F_{k(\theta)}^t}{\sum_{p=1}^{K} F_{k(\theta)}^t}, \quad k = 1, \ldots\ldots, K. \tag{26}$$

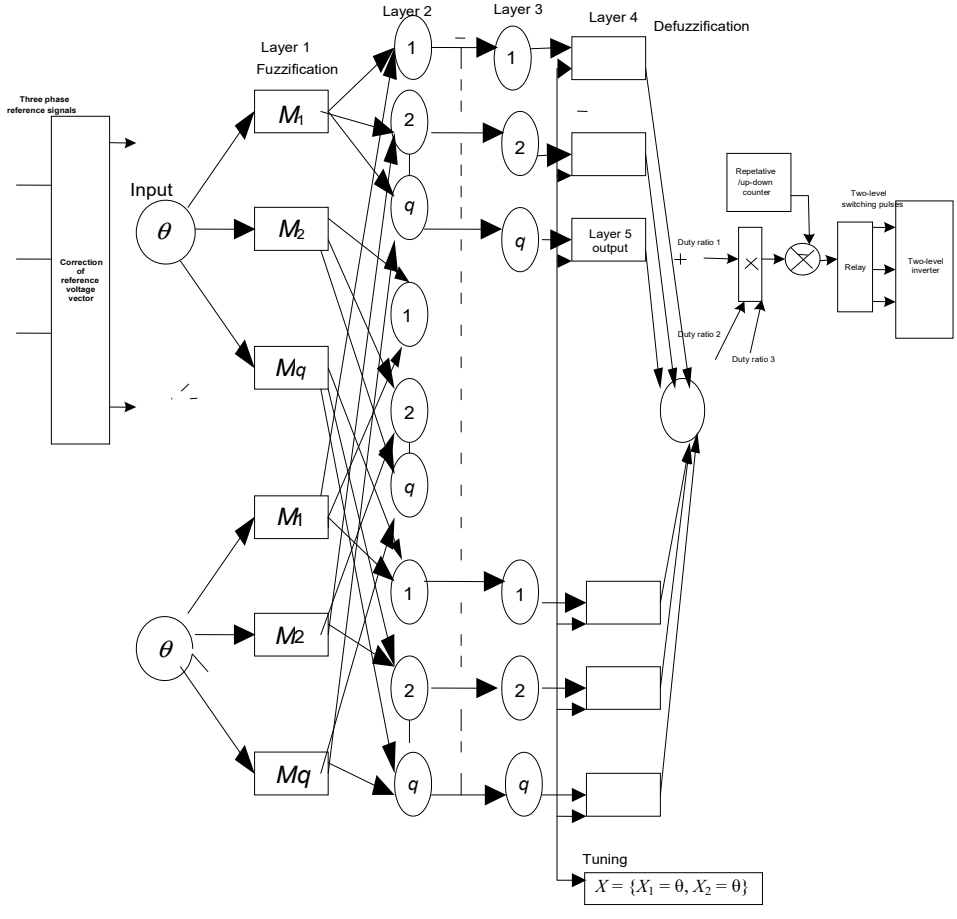
Layer 6 *Output layer:* This layer computes the output of the network by the normalised weighted sum of firing strengths of rules and is given as:

$$\widetilde{y_J^t} = \frac{\sum_{k=1}^{K} w_{kj} F_k^t}{\sum_{p=1}^{K} F_p^t}, \; j = 1, 2, \ldots\ldots, n \tag{27}$$

$$\widetilde{y_J^t} = \frac{\sum_{k=1}^{K} w_{kj} \left[1 - \alpha \left[\sum_{i=1}^{m} \frac{\left(\mu_i^t - \mu_{1k(\theta)}\right)^2 + \left(\mu_{1k(\theta)} - \mu_1^t\right)^2}{2\sigma_k^2} \right] \right]}{\sum_{p=1}^{K} F_p^t} \tag{28}$$

where w_{kj} is the output weight connecting k^{th} rule with the j^{th} output node.

Figure 4 Structure of NFT2 Takagi-Sugeno system



4.1 Learning algorithm

To calculate the weights of the network, learning algorithm is used as a regularised form of the project-based learning algorithm. To find the minimum weight of network with reduced hinge error what is proposed is the project-based learning algorithm. However, minimising only the hinge-error may lead to over fitting. Hence, in this proposed work, the sum of squared hinge-error is regularised for all samples with a regularising factor.

The hinge-error is defined as

$$J_t = \sum_{j=1}^{n} \begin{cases} \left(\widetilde{y_J^t} - \sum_{k=1}^{K} w_{kj} F_k^t\right)^2 & \text{if } \widetilde{y_J^t} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

where

$$\overline{F_{k(\theta)}^t} = \frac{F_k^t}{\sum_{p=1}^{K} F_k^t} \quad k = 1, \ldots, K \tag{30}$$

$$\overline{F_{k(\theta)}^t} = \frac{1-\alpha\left[\sum_{i=1}^{m} \frac{\left(\mu_i^t - \mu_{ik(\theta)}\right)^2 + \left(\mu_{ik(\theta)} - \mu_i^t\right)^2}{2\sigma_{k(\theta)}^2}\right]}{\sum_{p=1}^{K} F_p^t} \tag{31}$$

$$J_t = \sum_{j=1}^{n} \left[\frac{\sum_{k=1}^{k} w_{kj} F_k^t}{\sum_{p=1}^{K} F_p^t} - \sum_{k=1}^{k} w_{kj} F_k^t\right] \tag{32}$$

$$J_t = \sum_{j=1}^{n} \left\{\frac{\sum_{k=1}^{k} w_{kj}\left[1-\alpha\left[\sum_{i=1}^{m} \frac{\left(\mu_i^t - \mu_{ik(\theta)}\right)^2 + \left(\mu_{ik(\theta)} - \mu_i^t\right)^2}{2\sigma_{k(\theta)}^2}\right]\right]}{\sum_{p=1}^{K} F_p^t}\right\} \tag{33}$$

The amalgamation of mean least-squares optimisation (LSO) and gradient descent back-propagation algorithms gives rise as well as tuning to the fuzzy model. An error measure is reduced by adding the squared difference between actual and desired output. Obtaining of error rate or the pre-determined epoch number prevents training. The nonlinear premise parameters are fine-tuned by implementing the gradient descent algorithm on one hand and in minimising the linear consequent parameters theme of least-square on the other. Both forward pass as well as backward pass has been used in the learning algorithm at each period.

4.2 Forward pass

In the forward pass, each node output is calculated in the NFT2 controller by training set of input patterns space vector angle (θ) and change in space vector angle (θ'), and the least-squares estimator is used to identify the rule of linear consequent parameters. The

duty ratio or output vector is a linear function in the Takagi-Sugeno model. The Gaussian membership function enriched with three parameters a, b and c training set of input (space vector and change in space vector angle) – output (duty ratio) patterns are linear equations, (m_{1j}, n_j and r_i) as

$$D_{1-p} = \overline{\omega_1}(1) f_1(1) + \overline{\omega_1}(1) f_2(1) + \cdots \overline{\omega_n}(1) f_n(1) \tag{34}$$

$$D_{1-p} = \overline{\omega_1}(2) f_1(2) + \overline{\omega_1}(2) f_2(2) + \cdots \overline{\omega_n}(2) f_n(2) \tag{35}$$

$$D_{1-p} = \overline{\omega_1}(m) f_1(m) + \overline{\omega_1}(m) f_2(m) + \cdots \overline{\omega_n}(m) f_n(m) \tag{36}$$

$$\begin{aligned} D_1(1) = {} & \overline{\omega_1}(1)\left[p_1\theta(1) + q_1\theta(1) + r_1\right] + \overline{\omega_2}(1)\left[p_2\theta(1) + q_2\theta(1) + r_2\right] + \cdots \\ & + \overline{\omega_n}(1)\left[p_n\theta(1) + q_n\theta(1) + r_n\right] \end{aligned} \tag{37}$$

$$\begin{aligned} D_1(2) = {} & \overline{\omega_1}(2)\left[p_1\theta(2) + q_1\theta(2) + r_1\right] + \overline{\omega_2}(2)\left[p_2\theta(2) + q_2\theta(2) + r_2\right] + \cdots \\ & + \overline{\omega_n}(2)\left[p_n\theta(2) + q_n\theta(2) + r_n\right] \end{aligned} \tag{38}$$

$$\begin{aligned} D_1(m) = {} & \overline{\omega_1}(m)\left[p_1\theta(m) + q_1\theta(m) + r_1\right] + \overline{\omega_2}(m)\left[p_2\theta(m) + q_2\theta(m) + r_2\right] + \cdots \\ & + \overline{\omega_n}(m)\left[p_n\theta(m) + q_n\theta(m) + r_n\right] \end{aligned} \tag{39}$$

where N represents the input-output patterns, in the rule layer (= 25), n represents a number of nodes, D_{1-N} is the expected duty ratio of the NFT2.

Equation (39) represented as follows:

$$D_{1-p} = Ak \tag{40}$$

where D_{1-N} is an $m \times 1 = 10{,}000 \times 1$

$$D_{1-p} = \begin{bmatrix} D_{1-p(1)} \\ D_{1-p(2)} \\ . \\ . \\ . \\ D_{1-p(m)} \end{bmatrix} \tag{41}$$

$$A = \begin{bmatrix} \varpi_1(1) & \varpi_1\theta(1) & \varpi_1\theta'(1) & . & . & \varpi(1) & \varpi_n(1)\theta(1) & \varpi_n\theta'(1) \\ \varpi_1(2) & \varpi_1\theta(2) & \varpi_1\theta'(2) & . & . & \varpi(2) & \varpi_n(2)\theta(2) & \varpi_n\theta'(2) \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \varpi_1(m) & \varpi_1\theta(m) & \varpi_1\theta'(m) & . & . & \varpi(m) & \varpi_n(m)\theta(m) & \varpi_n\theta'(m) \end{bmatrix} \tag{42}$$

A can be represented by $M \times X(1 + \text{number of inputs parameters}) = 10{,}000 \times 75$ matrix.

And Y is an $X(1 + \text{input variables} \times 1) \times 1 = 75 \times 1$ vector unidentified consequent parameters as

$$k = \left[m_1 n_1 r_1, m_2 n_2 r_2, \ldots\ldots, m_n n_n r_n\right]^T \tag{43}$$

In the above case, in training, 10,000 input-output patterns are used, which gives complexity, so the solution may not exist to equation (40). As an alternative, a least-squared estimate of K should be identified to find out the exact solution with minimising squared error as: $Ak - D_{1-N}^2$.

By using the pseudo-inverse technique, the least-squares estimate is achieved which is as follows:

$$k^* = (A^T A)^{-1} A^T D_{1-m} \tag{44}$$

where $(A^TA)^1\ A^T$ is the pseudo-inverse of A. After the establishment of consequent parameters, the error vector e and output vector D_1 can be calculated as

$$e = D_{1-N} - D_1. \tag{45}$$

4.3 Backward pass

The back-propagation algorithm is used in the backward pass. By using the chain rule, the antecedent parameters are updated, and error signals are propagated. The chain rule can be represented in equation form as

$$\Delta a = -\eta \frac{\partial E}{\partial a} = -\eta \frac{\partial E}{\partial e} \times \frac{\partial e}{\partial D_1} \times \frac{\partial D_1}{\partial (\varpi_i f_i)} \times \frac{\partial (\varpi_i f_i)}{\partial \varpi_i} \times \frac{\partial \varpi_i}{\partial \omega_i} \times \frac{\partial \omega_i}{\partial \omega_{A_1}} \times \frac{\partial \omega_{A_1}}{\partial a} \tag{46}$$

$$\begin{aligned} E &= \frac{1}{2} e^2 = \frac{1}{2} (D_{1-P} - D_1) \\ \Delta a &= -\eta (D_{1-P} - D_1)(-1) f_i \times \frac{\varpi_i (1 - \varpi_i)}{\varpi_i} \times \frac{\omega_i}{\omega_{Ai}} \times \frac{\partial \omega_{Ai}}{\partial a} \end{aligned} \tag{47}$$

where

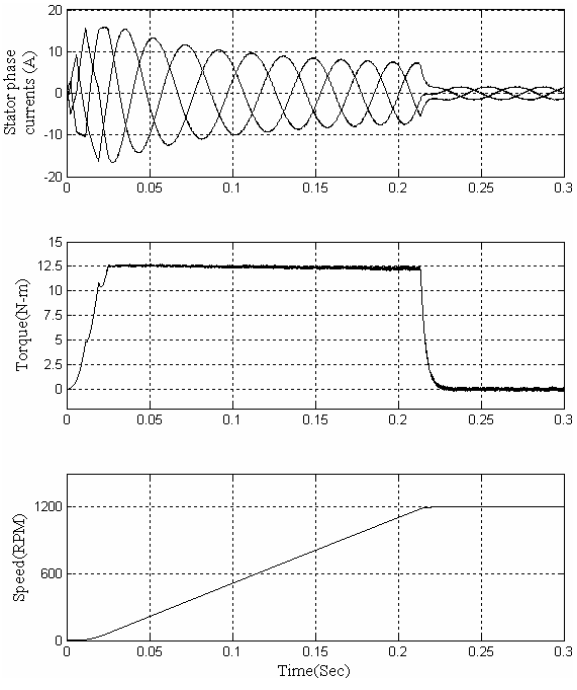
$$\begin{aligned} \frac{\partial \omega_{Ai}}{\partial a} &= \frac{1}{\left[1 + \left(\frac{\theta - a}{c}\right)^{2b}\right]} 2 \times \frac{1}{c^{2b}} \times 2b \times (\theta - a)^{2b-1} \times (-1) \\ &= \omega_{Ai}^2 \times \frac{2b}{c} \times \left(\frac{\theta - a}{c}\right)^{2b-1}. \end{aligned} \tag{48}$$

by using corrections, b and c parameters can be estimated.

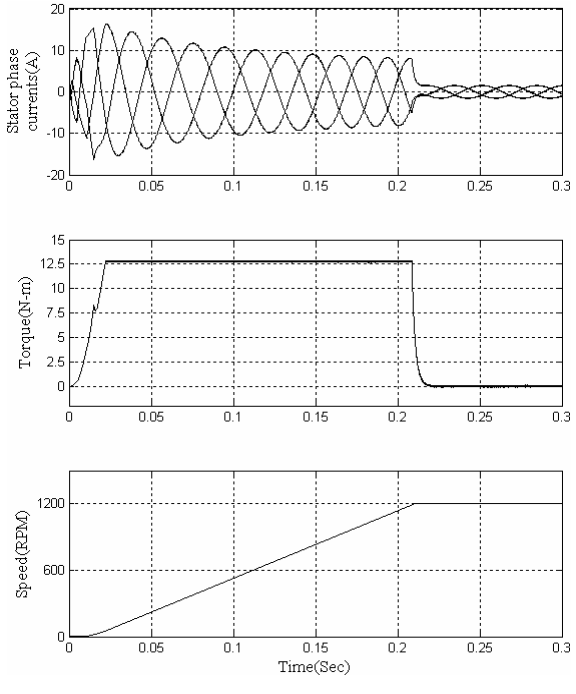
5 Results and discussion

The performance characteristics of an IM have been observed under various operating conditions with PI, neuro-fuzzy type-1 (NFT1) and NFT2 SVM-based controllers, i.e., during the start, during the step change in load torque, during steady-state and during speed reversal operation. The command speed of the 2 HP IM is 125.6 rad/sec (1,200 rpm). The appendix parameters of the IM shown below.

Figure 5 Performance of IM during starting, (a) conventional SVM (b) NFT1-based SVM (c) NFT2-based SVM

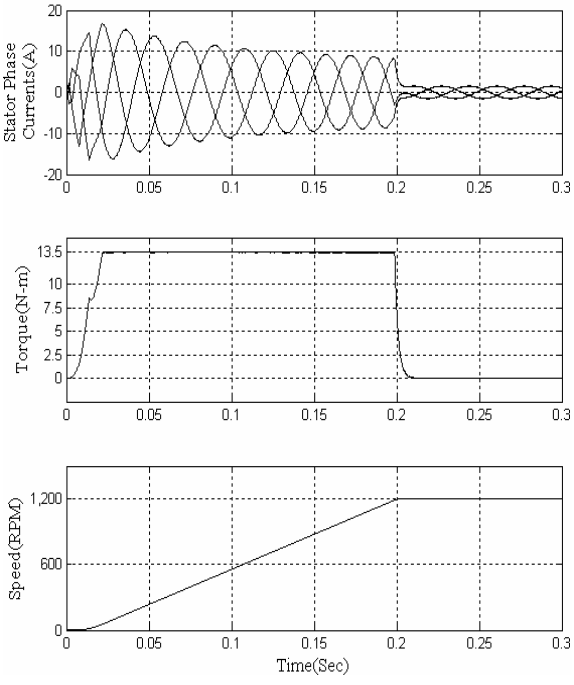


(a)



(b)

Figure 5 Performance of IM during starting, (a) conventional SVM (b) NFT1-based SVM (c) NFT2-based SVM (continued)



(c)

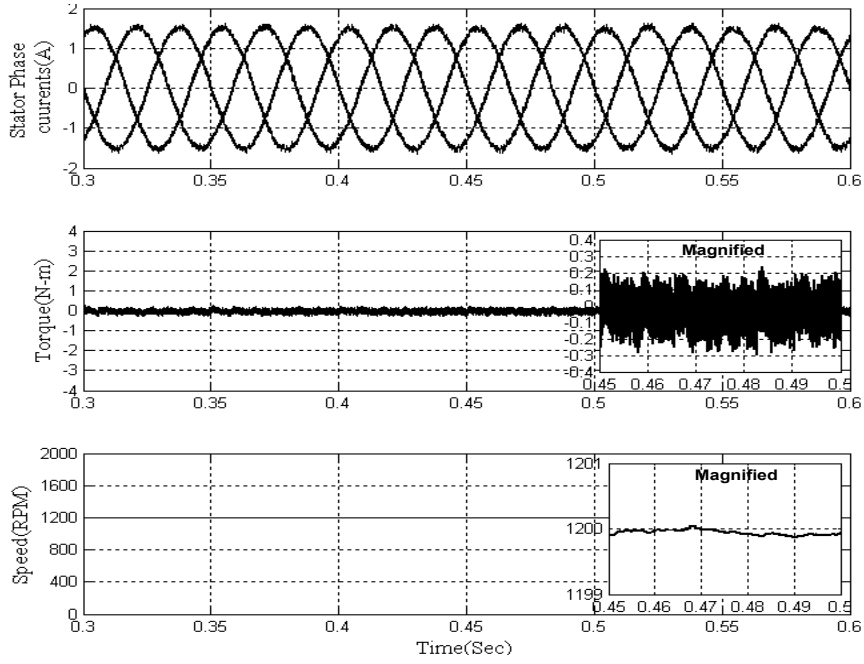
5.1 During starting

Figure 5 shows speed response of IM during starting reached early in NFT2-based SVM controller as compared to conventional and NFT1-based SVM controllers in IM within the time interval between 0.2 sec to 0.25 sec and ripple in torque is also reduced with NFT2 compared to conventional and NFT1-based SVM control method. Due to reduced ripple in the NFT2 controller, the torque is improved from 12.5 N-m to 13.5 N-m as compared with NFT1 and conventional SVM control method.

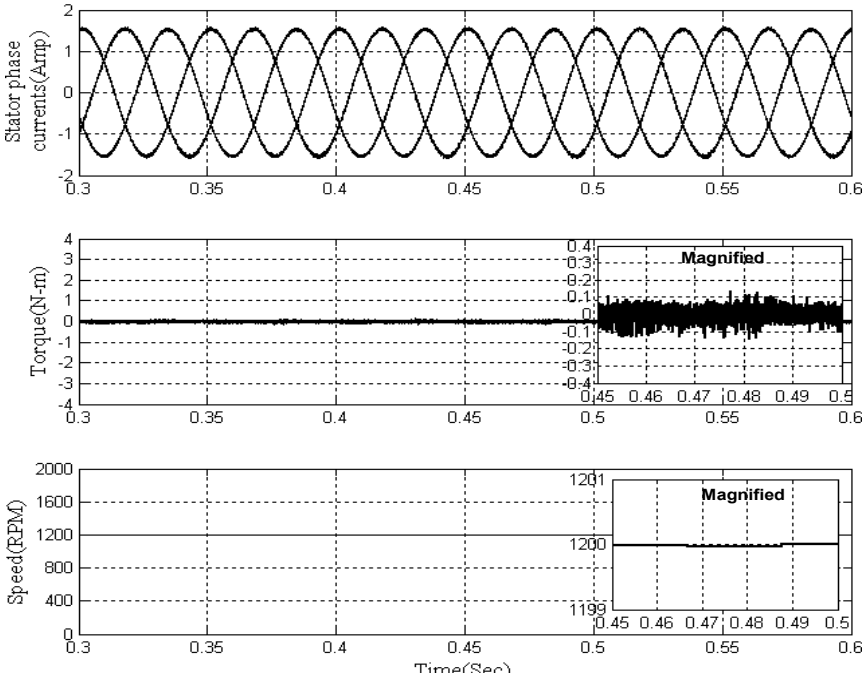
5.2 During steady-state

Figure 6 shows the response of induction during steady-state, where the ripple in torque is reduced with the NFT2-based SVM controller as compared to the conventional and NFT1-based SVM controller. The speed response of the IM is magnified within an interval time of 0.5 sec to 0.6 sec as compared to NFT1 and conventional SVM-based controllers. The torque ripple in conventional and NFT1-based SVM is 0.15 N-m and 0.5 N-m, respectively. In the proposed NFT2-based SVM, torque ripple reduced from 0.15 N-m to 0.025 N-m within the same internal time of operation.

Figure 6 Performance of IM during steady-state, (a) conventional SVM (b) NFT1-based SVM (c) NFT2-based SVM

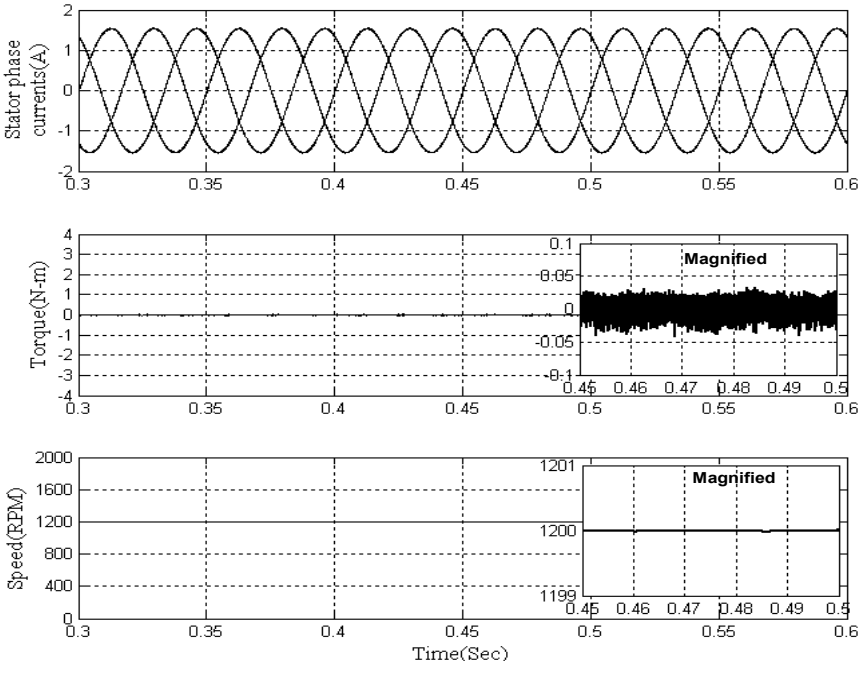


(a)



(b)

Figure 6 Performance of IM during steady-state, (a) conventional SVM (b) NFT1-based SVM (c) NFT2-based SVM (continued)



(c)

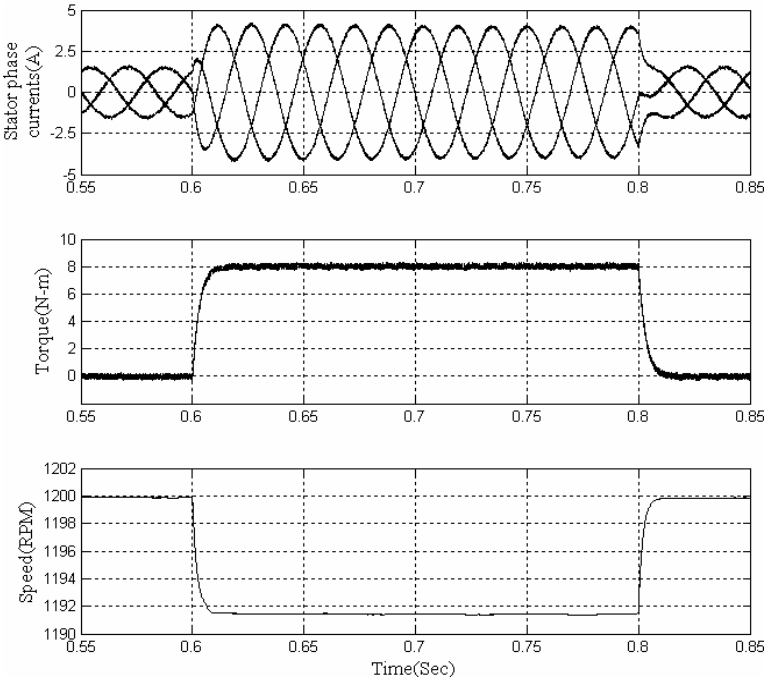
5.3 During step change in load torque

Figure 7 shows ripple in torque during a step change in load torque that is reduced in the NFT2 controller as compared to the conventional and NFT1 SVM-based controllers. The speed response has also increased in the NFT2 controller as compared to the conventional and NFT1 controller. During step-changing interval time, the response in speed has increased from 1,192 to 1,194 rpm in NFT2 controllers as compared to the conventional-based SVM.

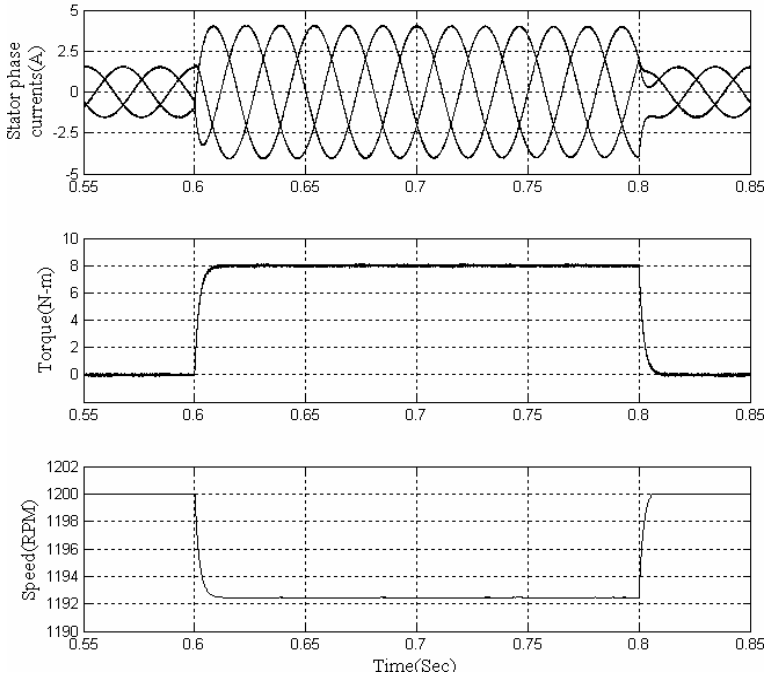
5.4 During speed reversal

Figure 8 shows that the ripple in torque and stator current during speed reversal is reduced in the NFT2 controller as compared with conventional SVM and NFT1 controller-based SVM. The speed response also increased in the NFT2 controller as compared to the conventional and NFT1 controller-based SVM. The speed reversal response in the IM reaches early within the time interval of 1.0 sec to 1.4 sec in NFT2-based SVM controller as compared to the conventional SVM and NFT1-based SVM controllers.

Figure 7 Performance of IM during a step change in load torque, (a) conventional SVM (b) type-1 NF-based SVM (c) NFT2-based SVM

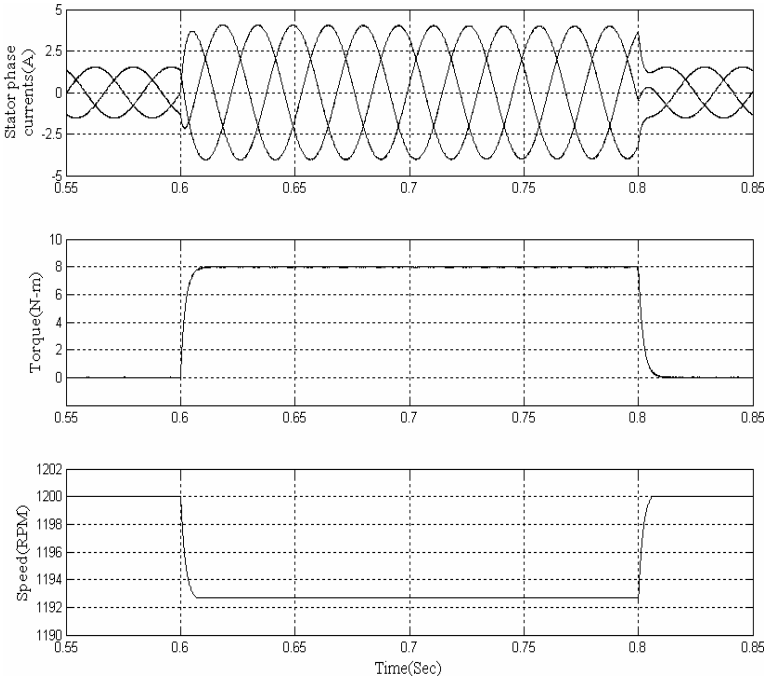


(a)



(b)

Figure 7 Performance of IM during a step change in load torque, (a) conventional SVM (b) type-1 NF-based SVM (c) NFT2-based SVM (continued)



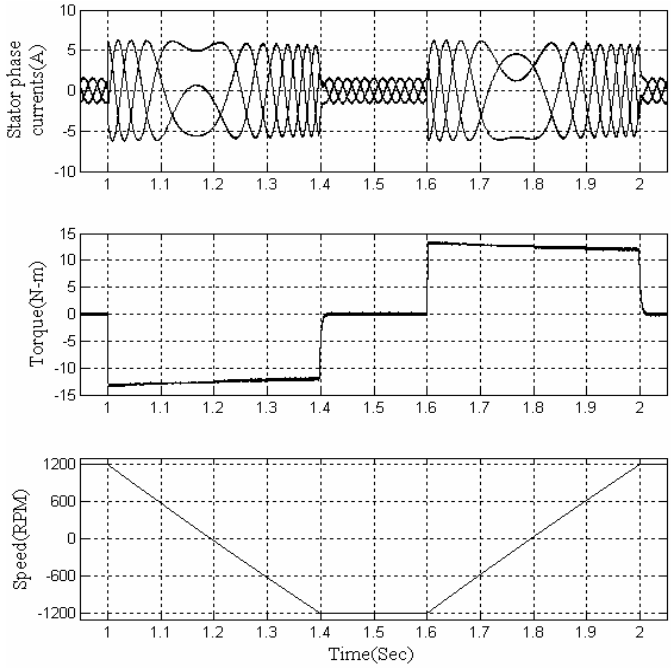
(c)

5.5 Experimental validation

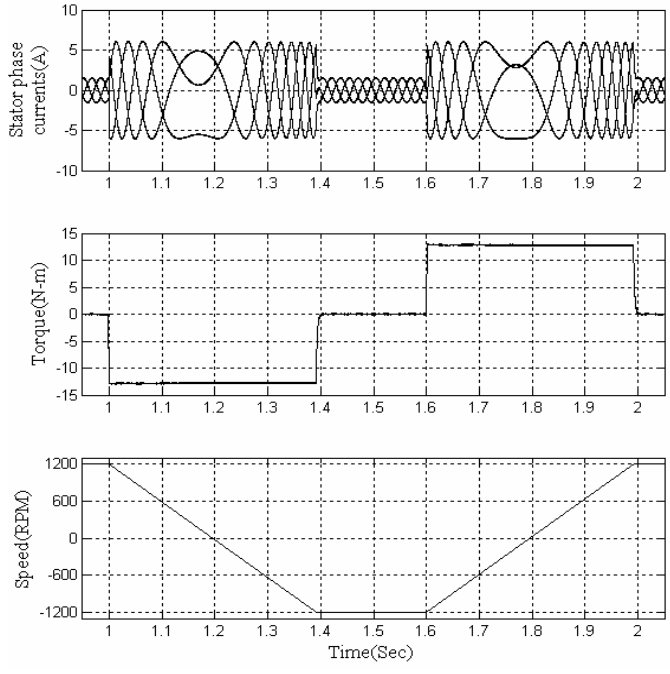
The real-time implementation of a two-level SVM-based inverter with Dspace(1104) has been presented. The MATLAB (Simulink) is used to develop the control algorithm that produces C code for real-time implementation. The controller which is used as an interface between MATLAB and Dspace 1104 to run the hardware setup is the MPC-8240 processor. The learning algorithm based on type-2 neuro-fuzzy (NFT2) has been used for the SVM algorithm. The modulator receives a space vector angle (θ) change of space vector angle (θ) at input and generates duty ratios at the output. The inverter is controlled by a controller (SVM modulator) in real-time simulation. This method is called rapid control prototyping. The analogue and digital converter interfaces use Dspace 1104 on its connector panel with sensed voltages.

Figures 9(a) and (9b) show the achieved experimental validation results with Dspace(1104). It is observed that the speed response reaches early in proposed NFT2-based SVM as compared to conventional-based SVM. The command speed of an IM is 125.6 rad/sec, i.e., 1,200 rpm. The dynamic performance of an IM is improved with proposed NFT2-based SVM without a ripple in actual speed as compared to the conventional SVM.

Figure 8 Performance during speed reversal operation (from +1,200 rpm to –1,200 rpm and from –1,200 to +1,200), (a) conventional SVM (b) NFT1-based SVM (c) NFT2-based SVM

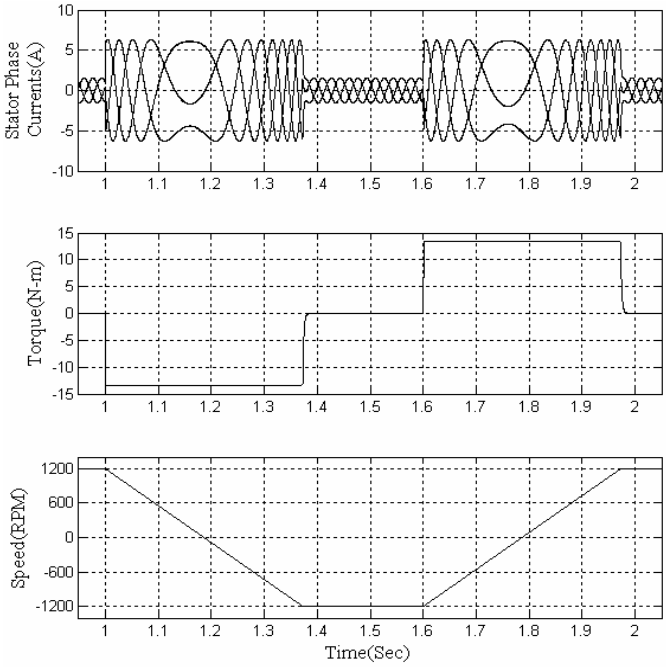


(a)



(b)

Figure 8 Performance during speed reversal operation (from +1,200 rpm to –1,200 rpm and from –1,200 to +1,200), (a) conventional SVM (b) NFT1-based SVM (c) NFT2-based SVM (continued)



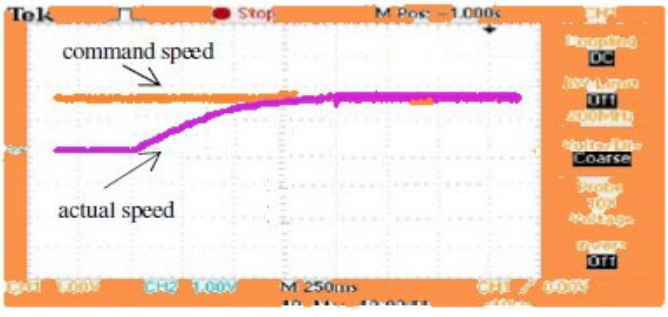
(c)

In real-time simulation, C code is generated from the control algorithm in MATLAB, which is used as an interface with Dspace(1104) and pulses are produced. On the Dspaceboard, master input/outputs (I/O) are used for gating signals for the power switches of the inverter. The connector (CLP1104), is used to achieve connection simplicity between Dspace(1104) and connected device on it. The gating pulse (digital signals) is used on the panel with an array of LED indicators. The opto-isolation circuit board is used to feed gating pulses to IGBT drive circuits.

Figure 9 Experimental speed responses of the IM drive for command speed of 125.6 rad/s, (a) conventional-based SVM (b) proposed NFT2 controller SVM (see online version for colours)

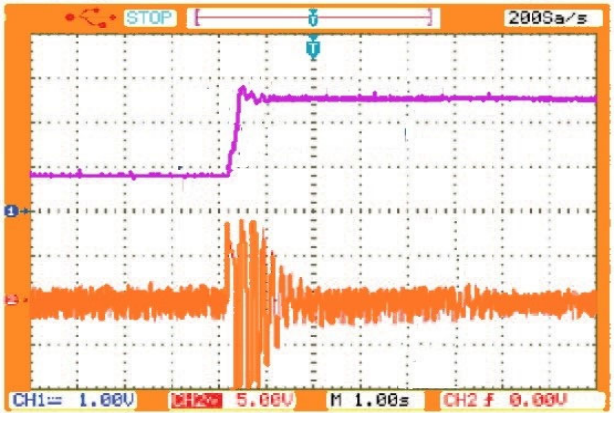
(a)

Figure 9 Experimental speed responses of the IM drive for command speed of 125.6 rad/s, (a) conventional-based SVM (b) proposed NFT2 controller SVM (continued) (see online version for colours)

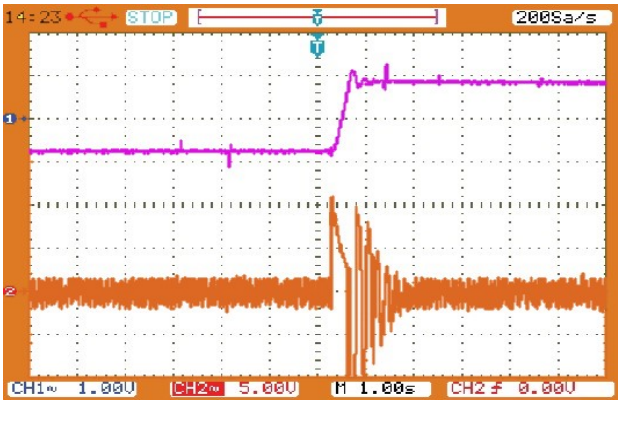


(b)

Figure 10 Experimental response, (a) actual speed and torque in forward motoring (b) actual speed and torque in reverse and forward motoring (brown-torque and 1 Div = 5 N-m) (blue-speed and 1 Div = 10 rad/sec) (c) direct axis and quadrature axis flux at 125.6 rad/sec (1 Div = 0.5 wb) (see online version for colours)

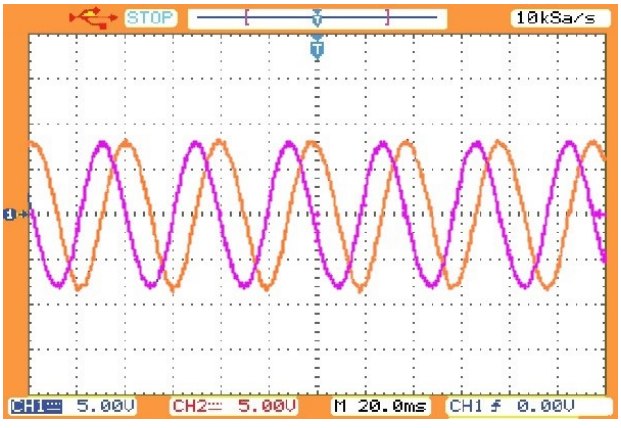


(a)



(b)

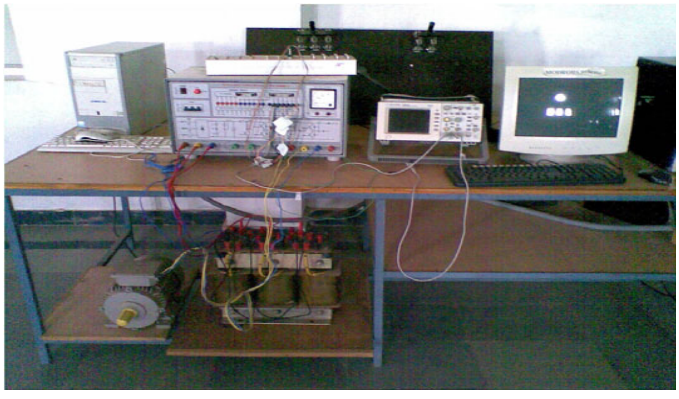
Figure 10 Experimental response, (a) actual speed and torque in forward motoring (b) actual speed and torque in reverse and forward motoring (brown-torque and 1 Div = 5 N-m) (blue-speed and 1 Div = 10 rad/sec) (c) direct axis and quadrature axis flux at 125.6 rad/sec (continued) (1 Div = 0.5 wb) (see online version for colours)



(c)

Figures 10(a) and 10(b) show the actual speed and torque in forward and reverse motoring that has been observed in proposed NFT2-based SVM, speedy responses reaching early and without any ripple content. The torque ripple has also reduced in forward and reversing monitoring in proposed NFT2-based SVM as compared to the conventional SVM. Figure 10(c) shows that the direct and quadrature axis flux at 125.6 rad/sec, i.e., 1,200 rpm. Similarly, Figure 11 shows an experimental setup with an IM (2 HP) using Dspace(1104).

Figure 11 Experimental setup (see online version for colours)



6 Conclusions

The dynamic performance of IM drive has improved with the proposed NFT2 controller as compared to the conventional SVM and NFT1-based SVM controller. The ripple in

torque and stator current has reduced in a proposed NFT2 controller as compared to the conventional DTC and NFT1 controlled-based SVM. The torque has increased by 8% in the NFT2 controller as compared to conventional SVM and NFT1-based SVM at the beginning of the IM. There is a fall in the ripple in torque which is 87.5% in the NFT2-based SVM controller as compared to the conventional-based SVM and NFT1-based SVM within the time interval of 0.5 sec to 0.6 sec. The resultant quick response is smooth and magnified within time interval 0.5 sec to 0.6 sec in NFT2-based SVM as compared to the conventional-based SVM and NFT1-based SVM during steady-state. When a step change in load torque is applied to the drive, the ripple in torque and speed is reduced in NFT2-based SVN as compared to the conventional SVM and NFT1 controller-based SVM. The speed response reaches the highest during speed reversal condition in NFT2 as compared to conventional SVM and NFT1 controller SVM.

The experimental results are validated with Dspace(1104), which has been used to implement the real-time simulation of the controller that is modelled in the Simulink. The NFT2 controller scheme uses a learning algorithm trained data. In the experimental results, the speed response reaches smoothly without any ripple in the NFT2 controller as compared to conventional SVM that has been shown in Figure 8. During speed reversal and forward condition of drive shown in Figures 9(a) and 9(b), torque ripple is reduced, and the fastest speed responses are attained. Figure 9(c) shown d-axis and q-axis flux without any ripple.

The proposed NFT2 SVM-based controller is faster and dynamic response of IM has been observed as compared to conventional-based SVM and NFT1-based SVM controllers.

References

Attaianese, C., Nardi, V. and Tomasso, G. (2007) 'Space vector modulation algorithm for power losses and THD reduction in VSI based drives', *Electric Power Components and Systems*, Vol. 35, No. 11, pp.1271–1283.

Dey, A., Rajeevan, P.P. and Gopakumar, K. (2013) 'A space-vector-based hysteresis current controller for a general n-level inverter-fed drive with nearly constant switching frequency control', *IEEE Transactions on Industrial Electronics*, Vol. 60, No. 5, pp.1989–1998.

Mehrizi-Sani, A. and Filizadeh, S. (2006) *Digital Implementation and Transient Simulation of Space Vector Modulated Converters*, pp.1–7, IEEE.

Mondal, S.K., Pinto, J.O.P and Bose, B.K. (2002) 'A neural-network-based space-vector PWM controller for a three-level voltage-fed inverter induction motor drive', *IEEE Transactions on Industry Applications*, Vol. 38, No. 3, pp.660–669.

Mondal, S.K., Pinto, J.O.P. and Bose, B.K. (2000) 'A neural-network-based space-vector PWM controller for voltage-fed inverter induction motor drive', *IEEE Transactions on Industry Applications*, Vol. 38, No. 3, pp.660–669.

Muthuramalingam, A., Sivaranjani, D. and Himavathi, S. (2005) 'Space vector modulation of a voltage fed inverter using artificial neural networks', *IEEE Indicon Conference*, pp.487–491.

Nai, N.V. and Singh, S.P. (2014) 'Improved torque and flux performance of type-2 fuzzy based direct torque control induction motor using space vector pulse-width modulation', *Electric Power Components, and Systems*, Vol. 42, No. 6, pp.658–669.

Pakkiraiah, B. and Sukumar, G.D. (2017) 'Enhanced performance of an asynchronous motor drive with a new modified adaptive neuro-fuzzy inference system based MPPT controller in interfacing with Dspace DS-1104', *International Journal Fuzzy Systems*, Vol. 19, No. 16, pp.1950–1965.

Sukumar, D., Jithendranatha, J. and Saranu, S. (2014) 'Three-level inverter-fed induction motor drive performance improvement with neuro-fuzzy space vector modulation', *Electric Power Components and Systems*, Vol. 42, No. 15, pp.1633–1646.

Sukumar, G.D. and Pathak, M.K. (2011a) 'Torque ripple minimization of vector controlled VSI induction motor drive using neuro-fuzzy controller', *International Journal of Advances in Engineering Sciences*, Vol. 11, No. 1, pp.40–43.

Sukumar, G.D. and Pathak, M.K. (2011b) 'Comparison of adaptive neuro-fuzzy-based space-vector modulation for two-level inverter', *Electrical Power and Energy System*, Vol. 38, No. 1, pp.9–19.

Sukumar, G.D. and Pathak, M.K. (2012a) 'Neuro-fuzzy-based torque ripple reduction and performance improvement of VSI fed induction motor drive', *International Journal of Bio-Inspired Computation*, Vol. 4, No. 2, pp.63–72.

Sukumar, G.D. and Pathak, M.K. (2012b) 'Neuro-fuzzy-based space vector modulation for THD reduction in VSI fed induction motor drive', *International Journal Power Electronics*, Vol. 4, No. 2, pp.160–180.

Sukumar, G.D., Abhiram, T. and Pathak, M.K. (2012c) *Type-2 Fuzzy Based SVM for Two-level Inverter Fed Induction Motor Drive*, IEEE.

Tripathi, A. and Ashwin, M.K. (2005) 'Torque ripple analysis and dynamic performance of a space vector modulation based control method for AC-drives', *IEEE Transactions on Industrial Electronics*, Vol. 20, No. 2, pp.485–492.

Van Der Broeck, H.W., Skudelny, H-C. and Stanke, G.V. (1988) 'Analysis and realization of a pulse width modulator based on voltage space vectors', *IEEE Transactions on Industry Applications*, Vol. 24, No. 1, pp.142–150.

Appendix

Power supply = three phase.

Machine rating = 3 hp.

Parameters

Stator resistance Rs = 0.55 Ω.

Stator inductance Ls = 93.38 mH.

Rotor resistance R_r = 0.78 Ω.

Rotor inductance L_r = 93.36 mH.

Magnetising inductance L_m = 90.5 mH.

Moment of inertia J = 0.019 Kg-m^2.

Damping coefficient B = 0.000051.

Load torque T_L = 10 N-m.

The inverter switching frequency is 3 kHz and 30 kHz.

DC link voltage is 400 V and 225 V.

The modulation index is 0.8.