Journal Européen des Systèmes Automatisés
Vol. 54, No. 4, August, 2021, pp. 575-589
Journal homepage: http://iieta.org/journals/jesa

Intelligent Control of Uncertain PMSM Based on Stable and Adaptive Discrete-Time Neural Network Compensators

Aziz El Janati El Idrissi*, Mohsin Beniysa, Adel Bouajaj, Mohammed Réda Britel

Innovative Technology Laboratory, ENSAT, Abdelmalek Essaadi University, Tetouan 93000, Morocco

Corresponding Author Email: ajanatiidrissi@uae.ac.ma

https://doi.org/10.18280/jesa.540407

Received: 10 December 2020
Accepted: 10 August 2021

Keywords:
uncertain PMSM, adaptive neural network, compensator, parameter uncertainties, Lyapunov stability

ABSTRACT

In this paper, stable and adaptive neural network compensators are proposed to control the uncertain permanent magnet synchronous motor (PMSM). Firstly, the overall uncertainties caused by mathematical modelling, parameters variation during operation and external load torque disturbances are modelled. Secondly, a new motion control scheme, where (d-q) current loops are dotted by two on-line tuning neural network compensators (NNCs), is used to compensate these uncertainties. As a result, the speed control loop is processed easily by proportional integral (PI) controller. Stability of the closed-loop system is also designed according to the Lyapunov stability. Compared to classical vector control, the simulations of PMSM system at different speeds including nominal, low and high speed, with and without uncertainties, show the effectiveness of the proposed control scheme.

1. INTRODUCTION

In the past two decades, many problems of uncertain nonlinear systems such as PMSM, manipulators, robots etc. are intensively studied in the automatic research field. PMSM is increasingly used in industrial applications due to its high efficiency, high speed, high power density, low cost and large torque to inertia ratio [1, 2]. But it is still a challenging domain of research to control these types of motors perfectly, because of their mathematical modelling, multi-variability, nonlinearities, parameters uncertainties, high degrees of freedom and external disturbances. These problems can cause undesired behavior of PMSM that can extremely destroy the stabilization of the motor drive. However, traditional control techniques cannot be used in this case [3]. In order to overcome these drawbacks, many adaptive techniques are proposed in recent years. The most popular of them are backstepping control with disturbance observer [4], unknown disturbance estimation with a hybrid sliding mode observer [5], and adaptive sliding model control [6, 7]. However, these mentioned control methods require the exact mathematical model of the dynamical system, but the model is approximated by ignoring some physical phenomena. To overcome this problem others modern optimized and intelligent techniques are proposed in the literatures [8-12].

In recent years, neural networks (NN) have always been considered as an efficient method in control of linear and nonlinear systems, due to its high accuracy and robustness against internal and external disturbances [13-16]. The NN is a set of the neuron's structures using training algorithm to adjust the NN weight's [17]. The online weight updating mechanism is an extended version of training algorithm with a simple structure by adding an adaptive term to guarantee the robustness of uncertainty. Extended version of neural networks control has been found one of the most popular and conventional tools in engineering control, identification and functional approximations [18-20].

The NN control has strong ability of handling uncertain information and can be easily used in the engineering control systems that are too complex to have an exact mathematical model. Actually, NN techniques have been developed in both continuous and discrete-time, but the techniques implemented in discrete-time have the favor that they can be directly implemented in digital hardware [21-32]. Unfortunately, one of the most important problems of NN control system analysis is the stability, and especially when the system is in discrete-time [21, 22]. Over the past decade, the stability and stabilization of NN systems have been considered as an essential and necessary area of researches [21-27]. However, stability analysis of NN systems was studied mainly by Lyapunov stability theory and others different kinds of its extended versions [28-32].

The main objective of this work is to use an intelligent dynamic approach based on two on-line tuning neural networks compensators (NNCs), which can adapt their weights dynamically to any uncertainty in the PMSM drives. This adaptation allows the NNCs to eliminate all disturbances related to nonlinearities uncertainties caused by mathematical modelling, external disturbance affected by the external load and parameters uncertainty. The discrete Lyapunov function is also considered to verify the stability of the PMSM under these uncertainties. In order to validate the effectiveness of the proposed approach we used several inputs references data for respective simulation tests. The simulation results are discussed and compared to those obtained in classical vector control. The results indicate that the proposed approach is successful and simple to control PMSM systems in spite of the uncertainty problems.

2. MATHEMATICAL MODEL DESCRIPTION AND PROBLEM FORMULATION OF PMSM

By considering the simplifying conditions, assumptions and physical laws, the three-phase mathematical model of PMSM can be expressed easily in three-phase (*abc*) frame [5-17]. Then, by developing this coupled three-phase mathematical model of PMSM, the (*dq*) axis of current, voltage and flux can be also obtained from Park and Concordia transformations [33]. So, according to the above steps the electromechanical behavior of PMSM system can be described by the following first order differential equations [33]:

$$\begin{cases} L_d \dfrac{di_d}{dt} = u_d - Ri_d + pL_q\omega_r i_q \\ L_q \dfrac{di_q}{dt} = u_q - Ri_q - pL_d\omega_r i_d - p\varphi_v\omega_r \\ J\dfrac{d\omega_r}{dt} = T_e - T_r - B\omega_r \\ \dfrac{d\theta_r}{dt} = \omega_r \\ \omega_e = p\omega_r \\ T_e = 1.5p\left(\varphi_v i_q + \left(L_d - L_q\right)i_d i_q\right) \end{cases} \tag{1}$$

where, R represents the stator resistance, θ_r represents the rotor position, u_d and u_q represent the stator voltages at rotational reference frame, i_d and i_q represent the direct and quadrature stator currents, ω_r and ω_e represent the electrical and mechanical rotor speeds, φ_v represents the stator flux linkage due to the permanent magnet, T_e represents the electrical torque, p represents the number of pole pairs, J represents the rotor inertia, B represents the friction, L_d and L_q represent the direct and quadrature stator inductances and T_r represents the mechanical load torque.

Furthermore, by using the following notations: $i_d(k) = x_1(k)$, $i_d(k+1) = x_2(k)$, $i_q(k) = x_3(k)$, $i_q(k+1) = x_4(k)$, $\omega_r(k) = x_5(k)$, $\omega_r(k+1) = x_6(k)$, $\theta_r(k) = x_7(k)$, $\theta_r(k+1) = x_8(k)$, $T_e(k) = x_9(k)$, $T_e(k+1) = x_{10}(k)$ the dynamic of the PMSM can be rewritten in the Brunovsky discrete-time domain form by the following system of equations, according to Newton discretization method:

$$\begin{cases} x_1(k+1) = x_2(k) \\ x_2(k+1) = x_2(k) + (u_d(k) - Rx_2(k) + pL_q x_4(k)x_6(k))\left(\dfrac{T_s}{L_d}\right) \\ x_3(k+1) = x_4(k) \\ x_4(k+1) = x_4(k) + (u_d(k) - Rx_4(k) - pL_q x_2(k)x_6(k) - p\varphi_v x_6(k))\left(\dfrac{T_s}{L_d}\right) \\ x_5(k+1) = x_6(k) \\ x_6(k+1) = x_6(k) + (x_{10}(k) - T_r - Bx_6(k))\left(\dfrac{T_s}{J}\right) \\ x_7(k+1) = x_8(k) \\ x_8(k+1) = x_8(k) + x_6(k)T_s \\ x_9(k+1) = x_{10}(k) \\ x_{10}(k+1) = 1.5\mathrm{p}\left(\varphi_v x_4(k+1) + \left(L_d - L_q\right)x_2(k+1)x_4(k+1)\right) \end{cases} \tag{2}$$

where, T_s is the sampling time. In the fact, the parameters φ_v, R, L_d, L_q, B, J, and load torque T_r are uncertain during the operation, then:

$$\begin{array}{lll} \hat{L}_d = L_d + \tilde{L}_d & \hat{J} = J + \tilde{J} & \hat{T}_r = T_r + \tilde{T}_r \\ \hat{L}_q = L_q + \tilde{L}_q & \hat{R} = R + \tilde{R} & \hat{\varphi}_v = \varphi_v + \tilde{\varphi}_v \\ \hat{B} = B + \tilde{B} & & \end{array} \tag{3}$$

where, the parameters defined by cap depict the parameter actual values, the tiled parameters depict the individual errors and the parameters without accent depict constant (nominal) values. Substituting nominal parameters in Eq. (2) by those caped in Eq. (3), the uncertain PMSM can be modelled by the following equations:

$$\begin{cases} x_1(k+1) = x_2(k) \\ x_2(k+1) = x_2(k) + \left(u_d(k) - Rx_2(k) + pL_q x_4(k)x_6(k)\right)\left(\dfrac{T_s}{L_d}\right) \\ \qquad + \left(\begin{array}{c} -\tilde{R}x_2(k) + p\tilde{L}_q x_4(k)x_6(k) - \\ \left(u_d(k) - Rx_2(k) + pL_q x_4(k)x_6(k)\right)\left(\dfrac{\tilde{L}_d}{L_d}\right) \end{array}\right)\left(\dfrac{T_s}{L_d + \tilde{L}_d}\right) = f_d\left(x(k)\right) + \tau_d(k) + d_d(k) \\ x_3(k+1) = x_4(k) \\ x_4(k+1) = x_4(k) + \left(\begin{array}{c} u_q(k) - Rx_4(k) \\ -pL_q x_2(k)x_6(k) - p\varphi_v x_6(k) \end{array}\right)\left(\dfrac{T_s}{L_q}\right) \\ \qquad + \left(\begin{array}{c} -\tilde{R}x_4(k) - p\tilde{L}_d x_2(k)x_6(k) - p\tilde{\varphi}_v x_6(k) - \\ \left(u_q(k) - Rx_4(k) - pL_d x_2(k)x_6(k) - p\varphi_v x_6(k)\right)\left(\dfrac{\tilde{L}_q}{L_q}\right) \end{array}\right)\left(\dfrac{T_s}{L_q + \tilde{L}_q}\right) = f_q\left(x(k)\right) + \tau_q(k) + d_q(k) \\ x_5(k+1) = x_6(k) \\ x_6(k+1) = x_6(k) + \left(x_{10}(k) - T_r - Bx_6(k)\right)\left(\dfrac{T_s}{J}\right) \\ \qquad + \left(-\tilde{T}_r - \tilde{B}x_6(k) - \left(x_{10}(k) - T_r - Bx_6(k)\right)\left(\dfrac{\tilde{J}}{J}\right)\right)\left(\dfrac{T_s}{J + \tilde{J}}\right) = x_6(k) + \left(x_{10}(k) - T_r - Bx_6(k)\right)\left(\dfrac{T_s}{J}\right) + d_\omega(k) \\ x_7(k+1) = x_8(k) \\ x_8(k+1) = x_8(k) + x_6(k)T_s \\ x_9(k+1) = x_{10}(k) \\ x_{10}(k+1) = 1.5\mathrm{p}\left(\varphi_v x_4(k+1) + \left(L_d - L_q\right)x_2(k+1)x_4(k+1)\right) \\ \qquad + 1.5\mathrm{p}\left(\tilde{\varphi}_v x_4(k+1) + \left(\tilde{L}_d - \tilde{L}_q\right)x_2(k+1)x_4(k+1)\right) 1.5\mathrm{p}\left(\varphi_v x_4(k+1) + \left(L_d - L_q\right)x_2(k+1)x_4(k+1)\right) + d_T(k) \end{cases} \tag{4}$$

where, $x_i(k) \in \mathbb{R}^{10}, i = \{1,2 \dots ,10\}$, and $\tau_i(k) \in \mathbb{R}^2, i = \{\mathrm{d}, \mathrm{q}\}$. The vector $d_i(k) \in \mathbb{R}^4, i = \{\mathrm{d}, \mathrm{q}, \omega, \mathrm{T}\}$ denotes an unknown disturbance acting on the system at the k^{th} time with known constant.

From the Eq. (4) we conclude that the output $\tau_i(k)$ is related to the uncertain control input $u_i(k)$, $i = \{d, q\}$ through the following nonlinear expression:

$$\begin{cases} \tau_d(k) = u_d(k)\left(\dfrac{T_s}{L_d}\right) + \left(\begin{array}{c} -\tilde{R}x_2(k) + p\tilde{L}_q x_4(k) x_6(k) - \\ \big(u_d(k) - Rx_2(k) + pL_q x_4(k) x_6(k)\big)\left(\dfrac{\tilde{L}_d}{L_d}\right) \end{array}\right)\left(\dfrac{T_s}{L_d + \tilde{L}_d}\right) \\ \tau_q(k) = u_q(k)\left(\dfrac{T_s}{L_q}\right) + \left(\begin{array}{c} -\tilde{R}x_4(k) - p\tilde{L}_d x_2(k) x_6(k) - p\tilde{\varphi}_v x_6(k) - \\ \big(u_q(k) - Rx_4(k) - pL_d x_2(k) x_6(k) - p\varphi_v x_6(k)\big)\left(\dfrac{\tilde{L}_q}{L_q}\right) \end{array}\right)\left(\dfrac{T_s}{L_q + \tilde{L}_q}\right) \end{cases} \tag{5}$$

Since a backlash is a dynamic nonlinearity, the uncertainties in Eq. (5) can be modelled by the following backlash in discrete time domain [21, 22]:

$$\begin{aligned} \tau_i(k) &= Backlash\big(\tau_i(k-1), u_i(k-1), u_i(k)\big) \\ &= \begin{cases} m_i u_i(k) & if\ \big((u_i(k) > 0)\ and\ u_i(k-1) = m_i(\tau_i(k-1) - d_{i+})\big) \\ & if\ \big((u_i(k) < 0)\ and\ u_i(k-1) = m_i(\tau_i(k-1) - d_{i-})\big) \\ 0 & Otherwise \end{cases} \end{aligned} \tag{6}$$

where, $u_i(k-1)$ and $u_i(k)$ are the inputs of backlash function, $\tau_i(k-1)$ is the state, and m_i, d_{i+} and d_{i-} are unknown values, $i = \{d, q\}$.

The control law of this nonlinear discrete-time control of PMSM system is based on the following assumptions [21, 22]:

Assumption 1: The desired trajectory is assumed to be bounded in the sense that $\left\|x^*_{(i,1)}(k)\right\|^T \le X_{(i,1)}$ and $\left\|x^*_{(i,2)}(k)\right\|^T \le X_{(i,2)}$ for a known bound $X_{(i,1)}$ and $X_{(i,2)}$, where $x^*_{(i,1)}(k)$ is the delayed value of $x^*_{(i,2)}(k)$, $i = \{d, q\}$.

Assumption 2: The nonlinear function $f_i\big(x(k)\big)$ is assumed to be unknown, but a fixed estimate $\hat{f}_i\big(x(k)\big)$ is assumed known such that the functional estimation error $\tilde{\varepsilon}_i\big(x(k)\big) = \hat{f}_i\big(x(k)\big) - f_i\big(x(k)\big)$ satisfies $\left\|\tilde{\varepsilon}\big(x(k)\big)\right\| \le f_{iM}\big(x(k)\big)$ for some known bounding function $f_{iM}\big(x(k)\big)$, $i = \{d, q\}$.

Assumption 3: The unknown disturbance is also assumed to satisfy: $\|\tau_i\| \le \tau_{iM}$, with $\tau_{iM}(k)$ are a known positive constants, $i = \{d, q\}$.

Assumption 4: The unknown values m_i, d_{i+} and d_{i-} are bounded by known constants M_i, D_{i+} and D_{i-} as: $\|m_i\| \le M_i$, $\|d_{i+}\| \le D_{i+}$ and $\|d_i\| \le D_{i-}$.

Assumption 5: The dynamics of the backlash compensator is invertible. Its dynamic inverse is indicated by the following equation:

$$u_i(k) = Backlash^{-1}\big(u_i(k-1), w_i(k-1), w_i(k)\big) \tag{7}$$

where, $w_i(k-1)$ and $w_i(k)$ are the signals input of the backlash inverse that generates the signal $u_i(k)$, which is subsequently sent into the backlash to generate $\tau_i(k)$.

Assumption 6: The parameters m_i, d_{i+} and d_{i-} used in the backlash inverse are exactly corrects.

Our objective is to control the speed and currents of uncertain PMSM such that the tracking errors of $x_i(k)$, $i = \{1,2, \dots ,10\}$ closely converge to zero in the presence of any uncertainty. To achieve this objective, the stable discrete-time adaptive neural networks compensators are used. To prove the stability of the system the Lyapunov function is applied.

3. STRATEGY OF THE PROPOSED CONTROL DESIGN

3.1 Main scheme of the proposed technique

The tracking control system is presented schematically in Figure 1. The overall system consists of the uncertain PMSM coupled to variable external load disturbance, space vector pulse width modulation (SVPWM), voltage-source inverter (VSI), two backlash models, two estimated functions, two filters, two proportional-plus-derivative (PD) controllers, two NNCs and three essential loops.

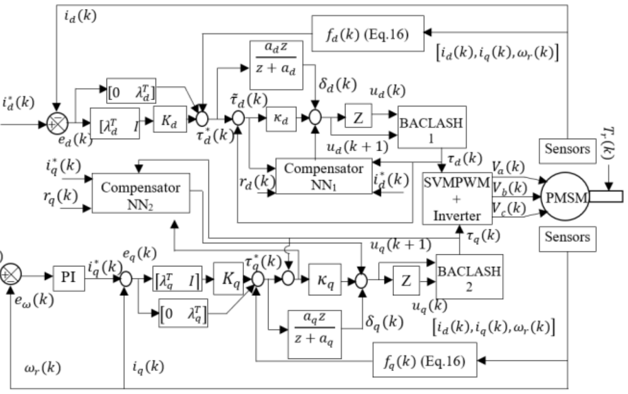

Figure 1. Stable and adaptive scheme for uncertain PMSM control

The controllers employ a structure of cascade control loop including a speed loop and two inner current loops. Here, the controllers, which are used to stabilize the (dq)-axis currents error, are adopted according to λ_i Hurwitz gains, which are followed by two inner PD controllers with gains (K_i and κ_i, $i = \{d, q\}$). The PI controller is also used to track the desired speed. As it can be seen from Figure 1, the electrical angular speed ω_r can be obtained from the position sensor, The currents i_d and i_q can be calculated from measured three-phase currents i_a and i_b by Park transformations.

According to the actual value of the angle θ_r, the backlash outputs can be transformed, from the rotating $(d,\, q)$ frame to the stator-fixed coordinates (α, β), to be the inputs of SVPWM that activates the transistors state of the inverter. Finally, the suitable stator voltages are applied on the motor with respect to the amplitude and the desired frequency. The uncertainty of PMSM destroys the tracking, and then the classical PD controllers are not capable to compensate it. To correct the errors due to this uncertainty, neural network compensators are used.

More details about the design procedure for the proposed control strategy (Figure 1) are presented step-by-step in the following subsection.

3.2 Discrete-time PI controller of the speed

To handle speed trajectory a PI controller is considered. The expressions of the proportional and integral controller gain respectively K_P and K_I are determined by the following equations [20]:

$$\begin{cases} K_I = \frac{4JR^2}{L_q^2} \\ K_P = \frac{K_I L_q}{R} \end{cases} \tag{8}$$

This controller compares, at each iteration, the actual speed value with the desired speed value. Then, it processes the speed errors defined by the Eq. (9) by the Eq. (10) to track the desired speed perfectly.

$$\begin{cases} e_i(k) = x_i^*(k) - x_i(k) \\ e_i(k+1) = x_i^*(k+1) - x_i(k+1) \end{cases}; \quad i = \{5{,}6\} \tag{9}$$

where, $e_i(k)$ and $e_i(k+1)$, represent the speed errors at discrete-times k and $(k+1)$.

$$\begin{cases} x_9^*(k) = x_9^*(k-1) + K_P e_5(k) + (K_I T_s - K_P) e_5(k-1) \\ x_{10}^*(k) = x_{10}^*(k-1) + K_P e_6(k) + (K_I T_s - K_P) e_6(k-1) \\ x_9^*(k+1) = x_9^*(k) + K_P e_5(k+1) + (K_I T_s - K_P) e_5(k) \\ x_{10}^*(k+1) = x_{10}^*(k) + K_P e_6(k+1) + (K_I T_s - K_P) e_6(k) \end{cases} \tag{10}$$

For $i_d^* = 0$ strategy, the desired trajectory of i_q is related to the desired trajectory of the torque with the following equations:

$$\begin{cases} x_3^*(k) = \frac{2}{3p\varphi_v} x_9^*(k) \\ x_3^*(k+1) = \frac{2}{3p\varphi_v} x_9^*(k+1) \\ x_4^*(k) = \frac{2}{3p\varphi_v} x_{10}^*(k) \\ x_4^*(k+1) = \frac{2}{3p\varphi_v} x_{10}^*(k+1) \end{cases} \tag{11}$$

Hence, by substituting Eq. (10) in Eq. (11), the desired trajectory i_q^* is related to the tracking errors of the speed with the following equations:

$$\begin{cases} x_3^*(k) = x_3^*(k-1) + \frac{2}{3p\varphi_v}\big(K_P e_5(k) + (K_I T_s - K_P) e_5(k-1)\big) \\ x_3^*(k+1) = x_3^*(k) + \frac{2}{3p\varphi_v}\big(K_P e_5(k+1) + (K_I T_s - K_P) e_5(k)\big) \\ x_4^*(k) = x_4^*(k-1) + \frac{2}{3p\varphi_v}\big(K_P e_6(k) + (K_I T_s - K_P) e_6(k-1)\big) \\ x_4^*(k+1) = x_4^*(k) + \frac{2}{3p\varphi_v}\big(K_P e_6(k+1) + (K_I T_s - K_P) e_6(k)\big) \end{cases} \tag{12}$$

3.3 Discrete-time adaptive neural networks compensators

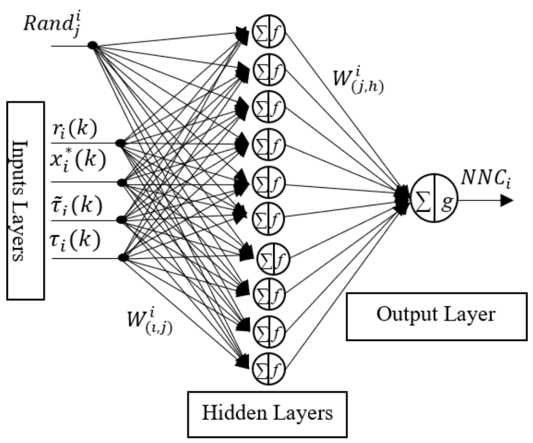

Figure 2. The structure of neural network compensators

The multi-layer perceptron neural networks (MLPNNs), used as compensators in this work (Figure 2), have dynamic weights between hidden and output neurons and static weights between input and hidden layers neurons. Consequently, they possess lesser number of weights to be updated and can be trained quickly and easily.

The concept of the adaptation technique law is to track precisely the desired trajectories of the uncertain PMSM. The tracking current errors of the desired trajectories $i^*_{(i,2)}(k)$ and their first delayed values $i^*_{(i,1)}(k)$, $i = \{d, q\}$ are expressed as:

$$\begin{cases} e_{(i,1)}(k) = i^*_{(i,1)}(k) - i_{(i,1)}(k) \\ e_{(i,2)}(k) = i^*_{(i,2)}(k) - i_{(i,2)}(k) \end{cases} \tag{13}$$

Eq. (14) defines a filtered tracking error $r_i(k)$ calculated from $e_{(i,2)}(k)$ tracking errors of currents and their delayed values $e_{(i,1)}(k)$, $i = \{d, q\}$.

$$r_i(k) = e_{(i,2)}(k) + \lambda_i e_{(i,1)}(k) \tag{14}$$

where, λ_i are chosen according to Hurwitz stability. Eq. (14) is used also to calculate $r_i(k+1)$ for discrete-time $(k+1)$. From Eq. (2), (13) and (14) we obtain the expression of the filtered tracking errors $r_i(k+1)$, $i = \{d, q\}$ as follows:

$$\begin{aligned} r_i(k+1) = f_i\big(x(k)\big) - i^*_{(i,1)}(k+1) + \lambda_i e_{(i,1)}(k) \\ + \tau_i(k) + d_i(k) \end{aligned} \tag{15}$$

where, the nonlinear function $f_d(x(k))$ and $f_q(x(k))$ can be estimated successively by the terms $\hat{f}_d(x(k))$ and $\hat{f}_q(x(k))$ as follows:

$$\begin{cases} \hat{f}_d\big(x(k)\big) = x_2(k) + \big(-Rx_2(k) + pL_q x_4(k) x_6(k)\big)\left(\frac{T_s}{L_d}\right) \\ \hat{f}_q\big(x(k)\big) = x_4(k) + \begin{pmatrix} -Rx_4(k) - pL_q x_2(k) x_6(k) \\ -p\varphi_v x_6(k) \end{pmatrix}\left(\frac{T_s}{L_d}\right) \end{cases} \tag{16}$$

An adaptive compensation scheme for the unknown term $f(x(k))$ is provided by selecting the following tracking controller:

$$\tau_i^*(k) = K_i r_i(k+1) - \hat{f}_i\big(x(k)\big) + x_{(i,2)}^*(k+1) + \lambda_i e_{(i,1)}(k) \tag{17}$$

The term $\tau_i^*(k)$ is the actuator output, which is also the desired control signal. The feedback gain matrix K_i is often selected diagonal and greater than zero [8].

The complete error system dynamics can be expressed by the following equation:

$$\begin{cases}\tilde{\tau}_d(k) = \tau_d^*(k) - \tau_d(k)\\ \tilde{\tau}_q(k) = \tau_q^*(k) - \tau_q(k)\end{cases} \tag{18}$$

Using the Eq. (17), the Eq. (15) can be rewritten as follows:

$$r_i(k+1) = K_i r_i(k) + \tilde{\varepsilon}_i\big(\iota_i(k)\big) + d_i(k) - \tilde{\tau}_i(k) \tag{19}$$

where, $i = \{d, q\}$, The substitution of $\tau_i(k+1)$ by $B_i(\tau_i(k), u_i(k), u_i(k+1))$ in equation 18 gives the following equation:

$$\begin{aligned}\tilde{\tau}_i(k+1) &= \tau_i^*(k+1) - \tau_i(k+1)\\ &= \tau_i^*(k+1) - B_i(\tau_i(k), u_i(k), u_i(k+1))\end{aligned} \tag{20}$$

Furthermore, the ideal backlash inverse and its approximation is given by:

$$\begin{cases}u_i(k+1) = B_i^{-1}(\tau_i(k), u_i(k), u_i(k+1))\\ \hat{u}_i(k+1) = \hat{B}_i^{-1}(\hat{\tau}_i(k), \hat{u}_i(k), \hat{u}_i(k+1))\end{cases} \tag{21}$$

where, $i = \{d, q\}$. We define also the backlash errors by the following equation:

$$\begin{aligned}\tilde{B}_i(\tau_i(k), u_i(k), u_i(k+1)) &= B_i(\tau_i(k), u_i(k), u_i(k+1))\\ &\quad - \hat{B}_i(\tau_i(k), u_i(k), u_i(k+1))\end{aligned} \tag{22}$$

Thus,

$$\begin{aligned}\tau_i(k+1) &= \hat{\tau}_i(k+1) + \tilde{\tau}_i(k+1)\\ &= \hat{B}_i(\tau_i(k), u_i(k), u_i(k+1))\\ &\quad + \tilde{B}_i(\tau_i(k), u_i(k), u_i(k+1))\end{aligned} \tag{23}$$

where, $i = \{d, q\}$. In order to design a stable closed-loop system with backlash compensation, we select a nominal backlash inverse $\hat{u}_i(k+1)$. Then, the system dynamics can be represented as follows:

$$\hat{u}_i(k+1) = -\kappa_i \tilde{\tau}_i(k) + \delta_i(k) + \widehat{W}^T(k)\sigma\big(V^T X_{NN}(k)\big) \tag{24}$$

The term $\widehat{W}^T(k)\sigma(V^T X_{NN}(k))$ is used to compensate the backlash inverse and the filtered error by using the NN approximation property, where $\widehat{W}(k)$ and $V(k)$ are the weights of NN. The term $\delta_i(k)$ is the filter dynamic of $\tau_i^*(k)$ processed by a discrete time filter $\frac{a_i z}{z + a_i}$. The discrete-time filter output $\delta_i(k)$, $i = \{d, q\}$ is calculated by the following equation:

$$\delta_i(k) = a_i \delta_i(k-1) - a_i \tau_i^*(k) \tag{25}$$

where, a_i and K_i represent respectively constant values of discrete-time filter and gain vector, and σ represents sigmoidal function.

Each NNC is composed of: input, hidden and output layers (Figure 2).

The input layer vector of NNC_i, $i = \{d, q\}$ is expressed by:

$$X^i(k) = [1 \quad r_i^T(k) \quad (r_i^*)^T(k) \quad \tilde{\tau}_i^T(k) \quad \tau_i^T(k)]^T \tag{26}$$

The expression of the hidden output j of NNC_i, $i = \{d, q\}$ is expressed by the following sigmoidal function:

$$Y_j^i(k) = \frac{1}{1 + exp\big(\sum_{\iota=1}^{n_1} W_{(\iota,j)}^i(k) X_\iota^i(k) + Rand_j^i\big)} \tag{27}$$

where, $Rand_i^j$ is the j^{th} normalized random bias value of the compensator NNC_i , $i = \{d, q\}, \iota = \{1, 2, \dots, n_1\}$, $j = \{1, 2, \dots, n_2\}$, n_1 and n_2 are respectively the numbers of the input layer neurons and the hidden layer neurons.

The output layer of each NNC_i, $i = \{d, q\}$ has one output neuron which is expressed by:

$$Z^i(k) = \sum_{j=1}^{n_2} W_{(j,h)}^i(k) Y_j^i(k) \tag{28}$$

Then, the expression of $u_i(k+1)$ is depicted by:

$$u_i(k+1) = \kappa_i(\tau_i^*(k) - \tau_i(k)) + \delta_i(k) + Z^i(k) \tag{29}$$

where, κ_i is a design parameter, which is always selected greater than zero [8]. Finally $\tau_i(k+1)$ is showed by:

$$\tau_i(k+1) = B_i(\tau_i(k), u_i(k), u_i(k+1)) \tag{30}$$

And hence,

$$\tau_i(k+1) = \begin{cases} m_i u_i(k+1) & if \ \ (u_i(k+1) > 0.0)\\ & and\ (u_i(k) = m_i \tau_i(k) - m_i d_{i+})\\ & or\ (u_i(k+1) < 0.0)\\ & and\ (u_i(k) = m_i \tau_i(k) - m_i d_{i+})\\ 0 & Otherwise \end{cases} \tag{31}$$

where, $i = \{d, q\}$, d_{i+} , d_{i-} and m_i are the backlash parameters. The right choice of the values of these parameters provides a correct backlash.

Notice that the weights vector $W_{(\iota,j)}^i$ between neurons of layers ι and j of NNC_i , $i = \{d, q\}$ are not time dependent since they are selected randomly at the initial time to provide a basis [18], and then they are kept constant through the tuning process. For the hidden layer, the NN weights are on-line adjusted in real time with no preliminary off-line learning required. The on-line tuning rule of hidden layer weights of the network are updated by the following expression [21]:

$$\begin{aligned}\widehat{W}^i(k+1) =& \widehat{W}^i(k) + \alpha_i Y^i(k)[r_i^T(k+1) + \tilde{\tau}_i(k+1)]\\ & - \gamma_i \| I - \alpha_i Y^i(k) Y^{iT}(k) \| \widehat{W}^i(k)\end{aligned} \tag{32}$$

where, $i = \{d, q\}$, α_i and γ_i are the networks learning rates [21], I is the identity matrix of the size $(n2+1)(n2+1)$ and $Y^i Y^{iT}$ is expressed by:

$$Y^{i}(k)Y^{iT}(k) = \sqrt{\sum_{\iota=1}^{n_1}\sum_{j=1}^{n_2}\left(Y^{i}_{(\iota+1)}(k)Y^{i}_{(j+1)}(k)\right)^2} \tag{33}$$

The convergence of Eq. (32) in such a manner that closed-loop stability is guaranteed, can be ensured by Lyapunov stability theory [22], in the following section.

3.4 Stability analysis of the proposed control

To show the boundedness of all closed-loop signals, the following positive discrete-time Lyapunov function is selected:

$$V(k) = V_1(k) + V_2(k) \tag{34}$$

The positive $V_1(k)$ represents the closed speed loop Lyapunov function which is used to weight the speed error $e_\omega(k)$. The positive $V_2(k)$ is used to weight the filtered tracking errors $r_i(k)$, the NN weights estimation errors $\widetilde{W}_i(k)$ and the backlash errors $\tilde{\tau}_i(k)$, $i = \{d, q\}$. These functions are defined by the following equations:

$$V_1(k) = \frac{1}{2}e_\omega^2(k) \tag{35}$$

$$\begin{aligned} V_2(k) = {} & 2r_d^T(k)r_d(k) + 2r_q^T(k)r_q(k) + 2r_d^T(k)\tilde{\tau}_d(k) \\ & +2r_q^T(k)\tilde{\tau}_q(k) + 2\tilde{\tau}_d^T(k)\tilde{\tau}_d(k) + 2\tilde{\tau}_q^T(k)\tilde{\tau}_q(k) \\ & +\frac{1}{\alpha_d}\left(W_d^T(k)\widetilde{W}_d(k)\right) + \frac{1}{\alpha_q}\left(W_q^T(k)\widetilde{W}_q(k)\right) \end{aligned} \tag{36}$$

The variation of the Eq. (35) gives the following equation:

$$\begin{aligned} \Delta V_1(k) \quad & = \frac{1}{2}e_\omega^2(k+1) - \frac{1}{2}e_\omega^2(k) \\ & \frac{1}{2}\left(\omega_r^*(k+1) - \omega_r(k+1)\right)^2 - \frac{1}{2}e_\omega^2(k) \end{aligned} \tag{37}$$

By using the PI speed loop design method, according to Euler integration, we obtain the virtual desired q-axis current i_q^* and by using the strategy law the desired d-axis current is $i_d^* = 0$. Then, the expression of input currents is:

$$\begin{cases} i_d^* = 0 \\ i_q^* = \dfrac{2}{3p\varphi_v}\left(\begin{array}{c} B\omega_r(k) + T_r \\ +J\left(K_P e_\omega(k) + K_I \sum e_\omega(k)\right) \end{array}\right) \end{cases} \tag{38}$$

where, K_I and K_P are a positive constant gains of PI controller. Substituting Eqs. (2) and (38) in Eq. (37) the variation of $V_1(k)$ can be expressed by the following equation:

$$\begin{aligned} \Delta V_1(k) \quad = \quad & \frac{1}{2}\left(\begin{array}{c} \omega_r^*(k) + \left(T_e(k) - T_r - B\omega_r^*(k)\right)\left(\frac{T_s}{J}\right) \\ -\omega_r^*(k) + (T_e^*(k) - T_r - B\omega_r^*(k))\left(\frac{T_s}{J}\right) \end{array}\right)^2 - \frac{1}{2}e_\omega^2(k) \\ = \quad & -\frac{BT_s}{J}\left(2 - \frac{BT_s}{J}\right)e_\omega^2(k) \\ & +2\left(1 - \frac{BT_s}{J}\right)\frac{T_s}{J}(T_e^*(k-1) - T_e(k-1) + K_P e_\omega(k) \\ & +(K_I T_s - K_P)e_\omega(k-1)\big)e_\omega(k) \\ & +\left(T_e^*(k-1) - T_e(k-1) + K_P e_\omega(k) + (K_I T_s - K_P)e_\omega(k-1)\right)^2 \\ & \left(\frac{T_s}{J}\right)^2 \end{aligned} \tag{39}$$

Substituting Eq. (2) in Eq. (39), we obtain the following equation:

$$\begin{aligned} \Delta V_1(k) = {} & -\frac{BT_s}{J}\left(2 - \frac{BT_s}{J}\right)e_\omega^2(k) \\ & +2\left(1 - \frac{BT_s}{J}\right)\frac{T_s}{J}\left(\begin{array}{c} \left(\frac{J}{T_s} + B\right)e_\omega(k) + \frac{J}{T_s}e_\omega(k-1) \\ +K_P e_\omega(k) + (K_I T_s - K_P)e_\omega(k-1) \end{array}\right)e_\omega(k) \\ & +\left(\frac{T_s}{J}\right)^2\left(\begin{array}{c} \left(\frac{J}{T_s} + B\right)e_\omega(k) + \frac{J}{T_s}e_\omega(k-1) + K_P e_\omega(k) \\ +(K_I T_s - K_P)e_\omega(k-1) \end{array}\right)^2 \end{aligned} \tag{40}$$

The Eq. (40) can be represented as follows:

$$\begin{aligned} \Delta V_1(k) = {} & -\frac{BT_s}{J}\left(2 - \frac{BT_s}{J}\right)e_\omega^2(k) \\ & +2\left(1 - \frac{BT_s}{J}\right)\frac{T_s}{J}\left(\left(\frac{J}{T_s} + B + K_P\right)e_\omega(k) + \left(\frac{J}{T_s} + K_I T_s - K_P\right)e_\omega(k-1)\right)e_\omega(k) \\ & +\left(\frac{T_s}{J}\right)^2\left(\left(\frac{J}{T_s} + B + K_P\right)e_\omega(k) + \left(\frac{J}{T_s} + K_I T_s - K_P\right)e_\omega(k-1)\right)^2 \end{aligned} \tag{41}$$

Finally, the expression of Lyapunov function variation $\Delta V_1(k)$ is:

$$\begin{aligned}\Delta V_1(k)= & -\frac{BT_s}{J}\left(2-\frac{BT_s}{J}\right)e_\omega^2(k)\\ & +2\left(1-\frac{BT_s}{J}\right)\frac{T_s}{J}\big(e_\omega(k)+K_0e_\omega(k-1)\big)e_\omega(k)+\left(\frac{T_s}{J}\right)^2\big(e_\omega(k)+K_0e_\omega(k-1)\big)^2\end{aligned} \tag{42}$$

where, $K_0=\frac{\left(\frac{J}{T_s}+(K_IT_S-K_P)\right)}{\left(\frac{J}{T_s}+B+K_P\right)}$ is a positive constant and $-B\frac{T_s}{J}\left(2-B\frac{T_s}{J}\right)<0$ according to the parameter values, which will be declared in the simulation. Then, according to Slotine [34], if K_0 is chosen properly for the controller, then $\big(e_\omega(k)+K_0e_\omega(k-1)\big)\rightarrow 0$. So, it is evident that the equation 42 is negative, meaning that the speed tracking errors is guaranteed to converge to zero. On the other hand, the variation of $V_2(k)$ is expressed by:

$$\begin{aligned}\Delta V_2(k)= & \ 2r_d^T(k+1)r_d(k+1)+2r_q^T(k+1)r_q(k+1)+2r_d^T(k+1)\tilde{\tau}_d(k+1)\\ & +2r_q^T(k+1)\tilde{\tau}_q(k+1)+2\tilde{\tau}_d^T(k+1)\tilde{\tau}_d(k+1)+2\tilde{\tau}_q^T(k+1)\tilde{\tau}_q(k+1)\\ & +\frac{1}{\alpha_d}\left(W_d^T(k+1)\widetilde{W}_d(k+1)\right)+\frac{1}{\alpha_q}\left(W_q^T(k+1)\widetilde{W}_q(k+1)\right)-2r_d^T(k)r_d(k)\\ & -2r_q^T(k)r_q(k)-2r_d^T(k)\tilde{\tau}_d(k)-2r_q^T(k)\tilde{\tau}_q(k)-2\tilde{\tau}_d^T(k)\tilde{\tau}_d(k)-2\tilde{\tau}_q^T(k)\tilde{\tau}_q(k)\\ & -\frac{1}{\alpha_d}\left(W_d^T(k)\widetilde{W}_d(k)\right)-\frac{1}{\alpha_q}\left(W_q^T(k)\widetilde{W}_q(k)\right)\end{aligned} \tag{43}$$

To not burden this paper, we note that the proof of $\Delta V_2(k)<0$ is similar to that determined in [21] (appendix (section 5.A)). So, according to the standard Lyapunov theorem extension, it can be concluded that the tracking error $r_i(k)$, the actuator error $\tilde{\tau}_i(k)$, and the errors of estimated NN weights $\widetilde{W}_i(k)$, $i=\{d,q\}$ are globally, uniformly and ultimately bounded (GUUB). So the uncertain PMSM is globally stable in the presence of internal and external uncertainties.

4. SIMULATION RESULTS

In this simulation the inverter is simulated by its ideal switching frequency 20 KHz. The motor is represented by its dynamic model in the Park frame. Thus, after several simulation tests we found that the average execution time of our program is around 0.0117s for 30000 simulation cycles. We deduce that the execution time needed for one cycle is around 3.9 μs. Bearing in mind that practically the sensors, the analog/digital converters, the switching elements of the inverter and the algorithm processing in DSP are time consuming. It is practically difficult to achieve such system with small sampling period. Thus, in practice, convenient sampling periods, such as $100\,\mu s$ or larger is normally selected for processing. So, to make our proposed adaptive method feasible for the uncertain PMSM with parameters shown below, it was simulated with the sampling time $T_s=100\ \mu s$, meaning that for 30000 samples the time of simulation is $T=3\ s$.

The values of PMSM parameters used in this study are given in Table 1 [9].

The two gains $K_P=0.3669$ and $K_I=88.5612$, of PI controller are given according to the expressions of [20]. In order to show the effectiveness of the proposed method the two backlash are considered differently with the parameters $m_d=0.5$, $m_q=0.55$, $d_{d+}=0.2$, $d_{d-}=0.2$, $d_{q+}=0.2$ and $d_{q-}=0.2$.

After several simulation tests, the appropriate values of the following parameters are chosen as: The Hurwitz and PD controller parameters gains are $\lambda_d=2.5$, $\lambda_q=0.23$, $K_d=0.0001285$; $K_q=0.1285$, $\kappa_d=2.25$ and $\kappa_q=2.0$. The number of hidden layer neurons, the bounded weights values of hidden layer with sigmoidal activation are respectively chosen as: $n_2=10$, $W_{(j,h)}^i\in[-0.1,+0.1]$. The first layer weights and bias are initialized randomly and uniformly distributed in the $[-0.1,+0.1]$ and $Rand_j^i\in[-100,+100]$. The filters that generate the signals δ_d and δ_q are implemented respectively with the gains $a_d=0.05006$ and $a_d=0133425$.

Table 1. Parameters of PMSM

Location	Symbols	Values	Units
Stator resistance	R	1.4	Ω
d-axis inductance	L_d	6.6	mH
q-axis inductance	L_q	5.8	mH
Magnetic Flux constant	φ_v	0.1546	Wb
Friction coefficient	B	0.00038	$N.mrad^{-1}s^{-1}$
Motor inertia	J	0.00176	Kgm^2
Nominal speed	ω_n	1430	rpm

In order to demonstrate the high performance of the proposed technique, numerous simulation tests are performed at the different operating conditions indicated below:

- The motor is started running under no load,
- The motor is suddenly under load torque disturbances variation: $T_r=2\,N.m$ applied between 0.25s and 1.75s,
- The motor is suddenly under load torque disturbances variation: $T_r=4\,N.m$ applied between 1.75s and 3.00s.
- The desired speeds with and without parameters uncertainty: $\omega_r=+150\,rad.s^{-1}$, $\omega_r=-150\,rad.s^{-1}$, $\omega_r=+5\,rad.s^{-1}$, $\omega_r=-5\,rad.s^{-1}$, $\omega_r=+200\,rad.s^{-1}$ and $\omega_r=-200\,rad.s^{-1}$ respectively between $[0-0.5]\,s$, $[0.5-1.0]\,s$, $[1.0-1.5]\,s$, $[1.5-2.0]\,s$, $[2.0-2.5]\,s$ and $[2.5-3.0]\,s$

- The desired sinusoidal speeds under parameters uncertainty: $\omega_r = +150\,rad.s^{-1}$, $\omega_r = -150\,rad.s^{-1}$, $\omega_r = +5\,rad.s^{-1}$, $\omega_r = -5\,rad.s^{-1}$, $\omega_r = +200\,rad.s^{-1}$ and $\omega_r = -200\,rad.s^{-1}$ respectively between $[0-0.5]\,s$, $[0.5-1.0]\,s$, $[1.0-1.5]\,s$, $[1.5-2.0]\,s$, $[2.0-2.5]\,s$ and $[2.5-3.0]\,s$ with $f = 4\,Hz$.

These conditions allow the testing of motor operation at nominal speed, low speed and high speed with a rotation reversing and under load torque disturbances variation.

The Figure 3 shows the simulation results of the speed response without parameters uncertainty. From the Figure 3(a), we remark that the classical vector control and the proposed adaptive method follow the desired references. In the zoomed Figures 3(b) to 3(i) we can see that the motor actual speed with the proposed adaptive method closely follows the reference speed; at reversing speed and load torque application; without overshoot and undershoot and converges rapidly with shorter settling time (0.025s). It can be seen also that the motor follows the speed reference without oscillation in steady state time. In contrast, the settling time, the overshoot and the undershoot of the classical vector control are noted respectively as (0.125s), (50%) and (25%). In Figure 3(j), speed tracking errors e_ω is plotted, it is clear that the speed errors of the proposed method drop to zero rapidly than the classical vector control. This means that the PMSM tracks the reference speed trajectory with high accuracy in proposed control.

Furthermore, the Figures 4 and 5 show the control currents while the motor is running under load torque disturbance in the same mentioned conditions. As can be seen from the zoomed Figures 4(b) to 4(c) and 5(b) to 5(c), the proposed adaptive method follows the currents reference in steady state time without oscillations. On the other hand, the zoomed Figures 4(d) and 5(d) show the currents responses during speed reversing. We remark clearly that the proposed method tracks rapidly the currents reference.

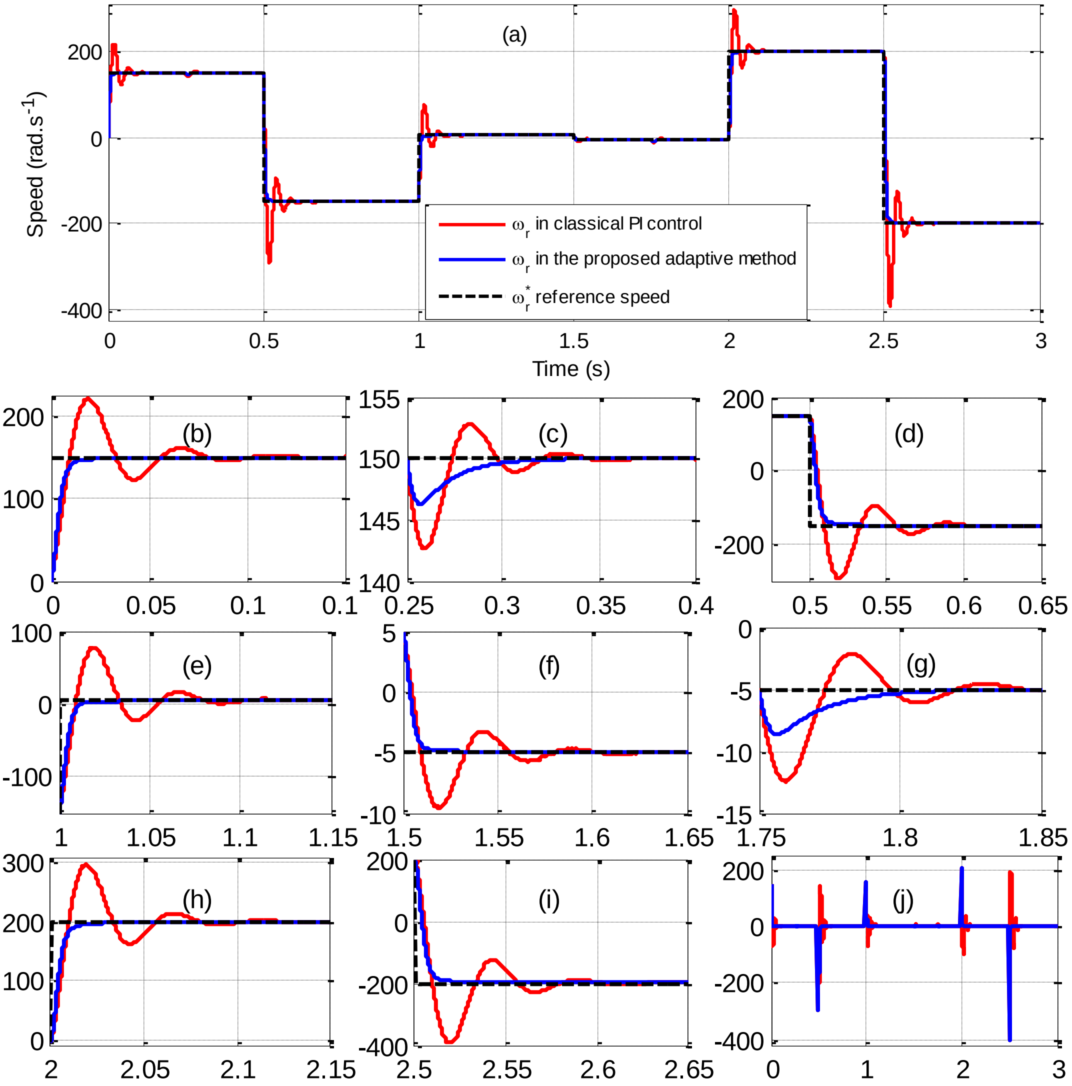

Figure 3. Speed response under load torque disturbance only: (a) Speed response, (b) zoom around 0 s, (c) zoom around 0.25 s, (d) zoom around 0.5 s, (e) zoom around 1.0 s, (f) zoom around 1.5 s, (g) zoom around 1.75 s, (h)zoom around 2.0 s, (i) zoom around 2.5 s, (j) tracking errors

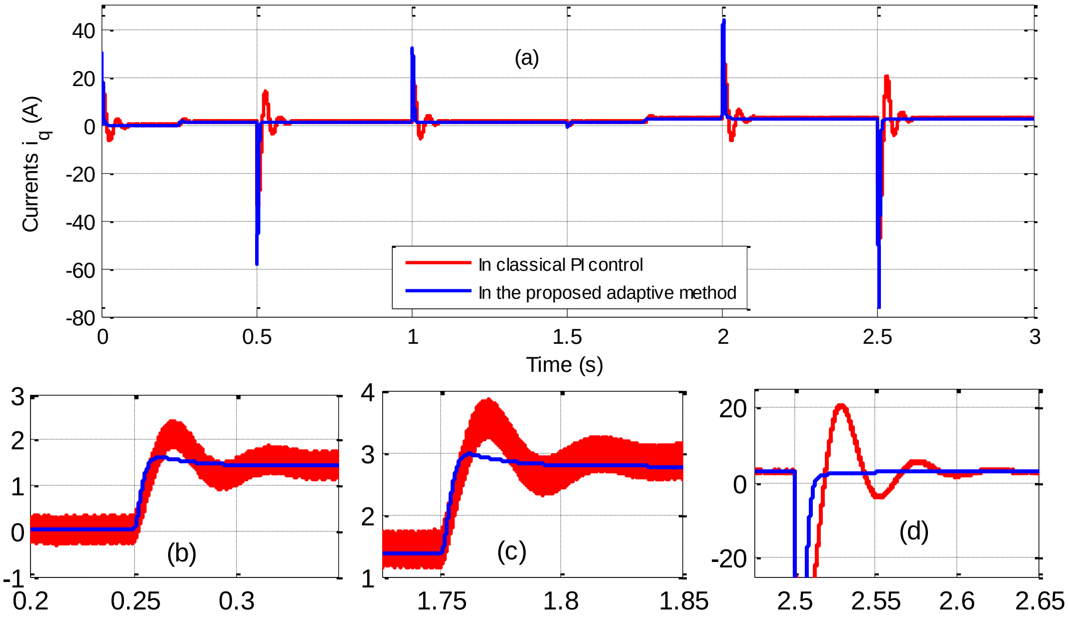

Figure 4. Evolution of the currents i_q response under load torque disturbance: (a) currents response, (b) zoom around 0.25 s, (c) zoom around 1.75 s, (d) zoom around 2.5 s

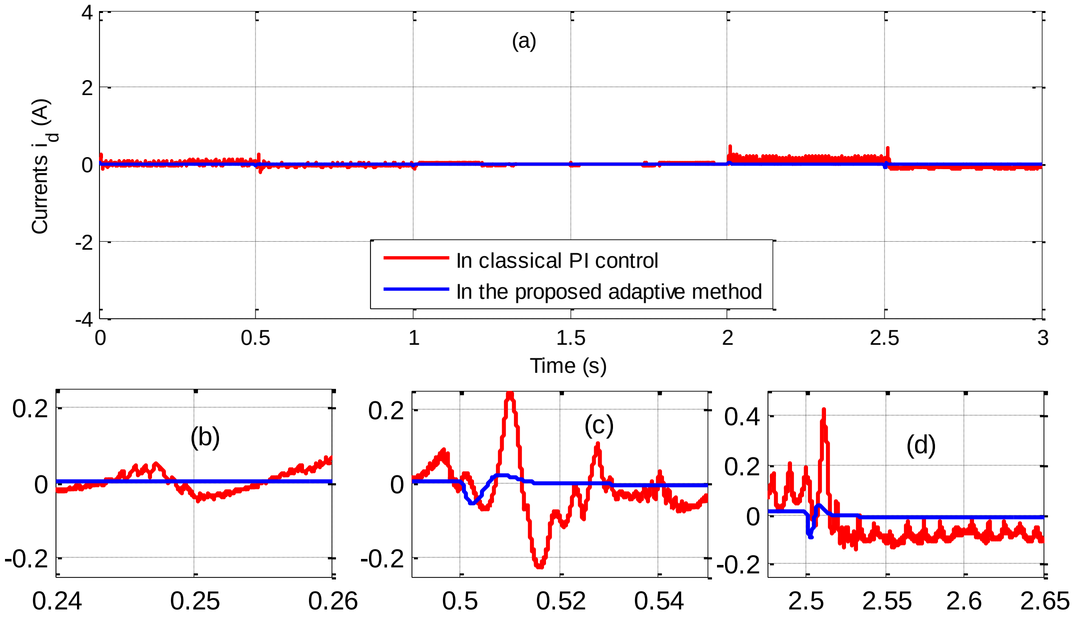

Figure 5. Evolution of the currents i_d response under load torque disturbance: (a) currents response, (b) zoom around 0.25 s, (c) zoom around 0.5 s, (d) zoom around 2.5 s

The Figures 6 presents the evolution of electrical torque responses under the same conditions mentioned above. We can also see that the proposed method follows the desired torque without oscillations in steady state time and drops rapidly to the desired reference during speed reversing.

Figures 7, 8, 9 and 10 show the simulations of uncertain PMSM, following the same desired trajectories, indicated above. In addition to mathematical uncertainty, perturbed PMSM parameters introduced in the control are set randomly to the following values: Increasing 300% in the stator resistance, 55% in the rotor moment of inertia and the friction, decreasing 25% in the stator inductances. The perturbed PMSM magnetic flux is expressed by the following function: $\varphi_v = -0.05t + 0.1546$ where t is the time.

The Figure 7 plots the desired and actual speed responses. As is shown in Figure 7(a), it is clear that the uncertain PMSM tracks precisely the reference speed trajectory with the proposed adaptive method, whereas the classical controller follows the reference quietly well at the beginning, after the application of the load torque of approximately $(2\,N.m)$ it loses completely its performance. From the zoomed Figures 7(b) to 7(i) we can also see that the proposed method follows the desired speed without oscillations in steady state time and drops rapidly to the desired reference during speed reversing and load torque application. In addition, the Figure 7(j) displays the tracking errors. The tracking errors in steady state trends to zero with the proposed adaptive method but for the classical vector control the tracking errors rate is greater than 15%, 76% and 36% respectively in nominal, low and high speed simulations. Table 2 resume a qualitative comparison between proposed technique and classical vector control of PMSM.

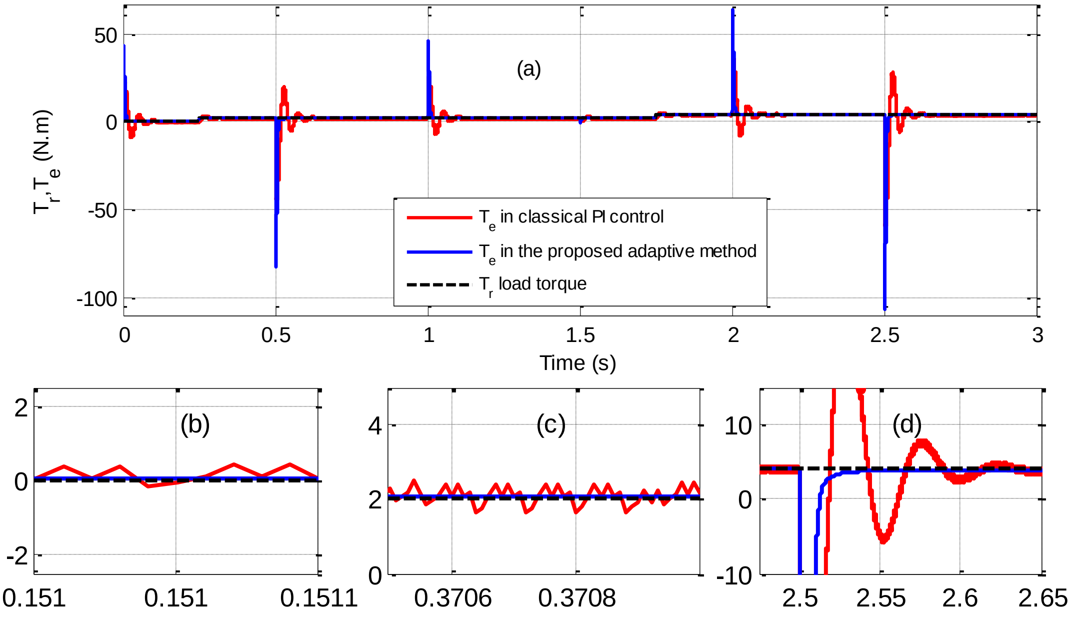

Figure 6. Evolution of the torque responses under load torque disturbance only: (a) Torque responses, (b) zoom around 0.15s, (c) zoom around 0.37s, (d) zoom around 2.5s

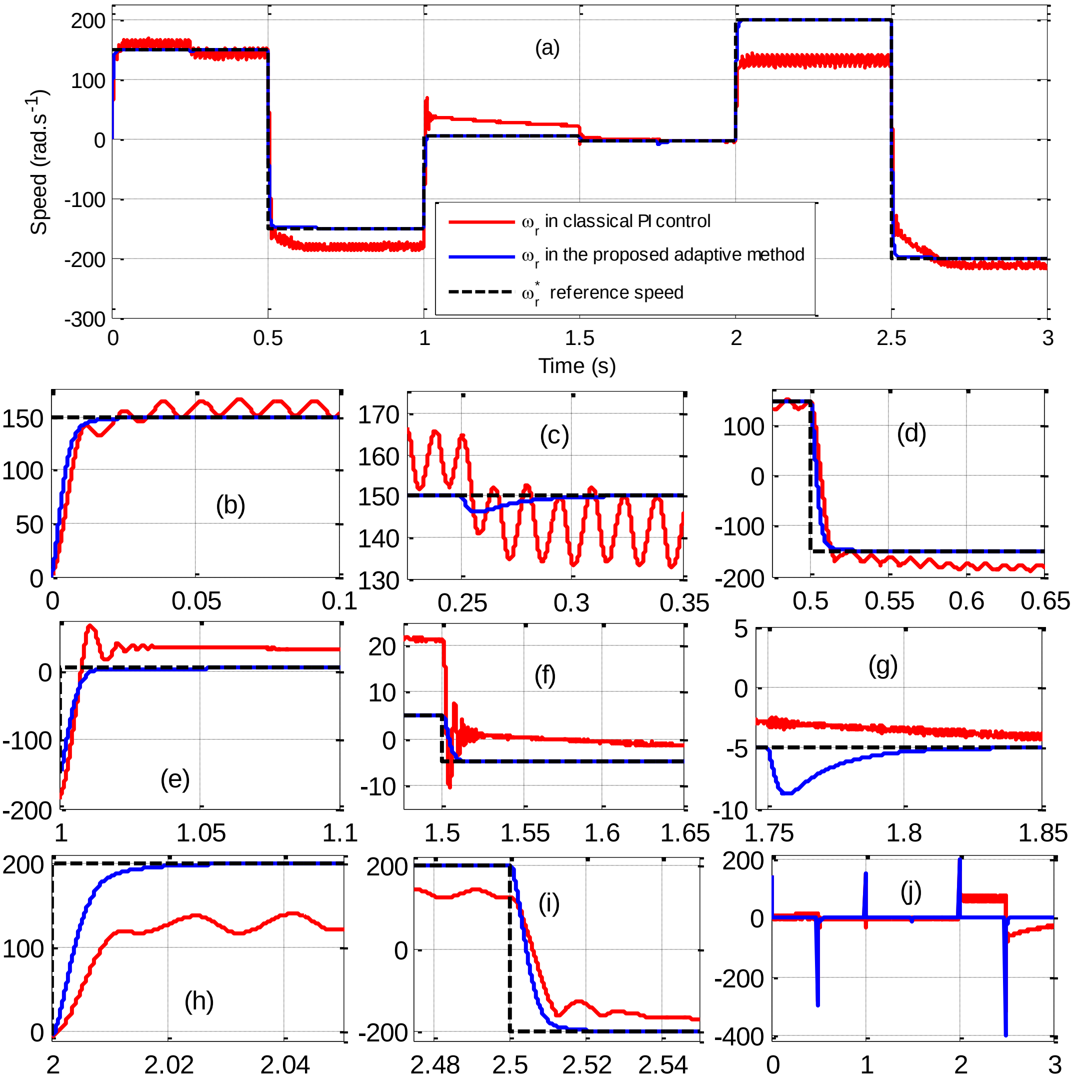

Figure 7. Speed response in the presence of uncertainties: (a) Speed response, (b) zoom around 0s, (c) zoom around 0.25s, (d) zoom around 0.5s, (e) zoom around 1.0s, (f) zoom around 1.5s, (g) zoom around 1.75s, (h) zoom around 2.0s, (i) zoom around 2.5s, (j) tracking errors

Table 2. Performance comparison

	Classical vector control	**Proposed technique**
Settling time (s)	More than 0.045	Less than 0.015
overshoot	More than 16%	0%
undershoot	More than 16 %	0%
oscillation	yes	no
stability	no	yes
Static error	More than 15%	0

Figures 8 and 9 show respectively the evolution of the currents i_q and i_d . The fluctuations that appear at the beginning and during speed inversion can be eliminated easily with the limiters. The simulation results for the currents trajectory tracking control at steady state time are shown in zoomed Figures 8(b) to 8(c) and 9(b) to 9(c). They indicate that in this situation the proposed adaptive method again has superior control performance to classical vector control. During the speed reversion, as can be seen from the zoomed Figures 8(d) and 9(d), it still has the best performance when the classical vector control degrades considerably.

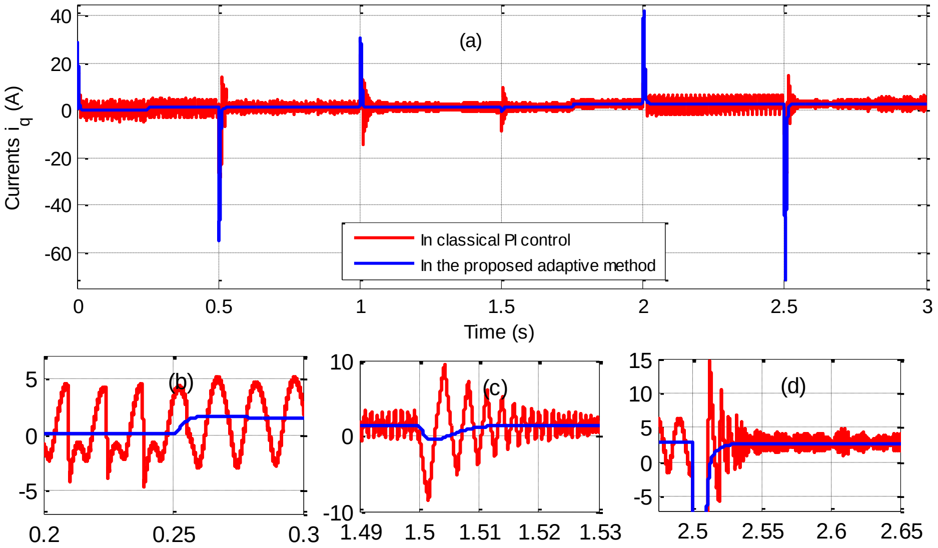

Figure 8. Evolution of the currents i_q response in the presence of uncertainties: (a) currents response, (b) zoom around 0.25s, (c) zoom around 1.5s, (d) zoom around 2.5s

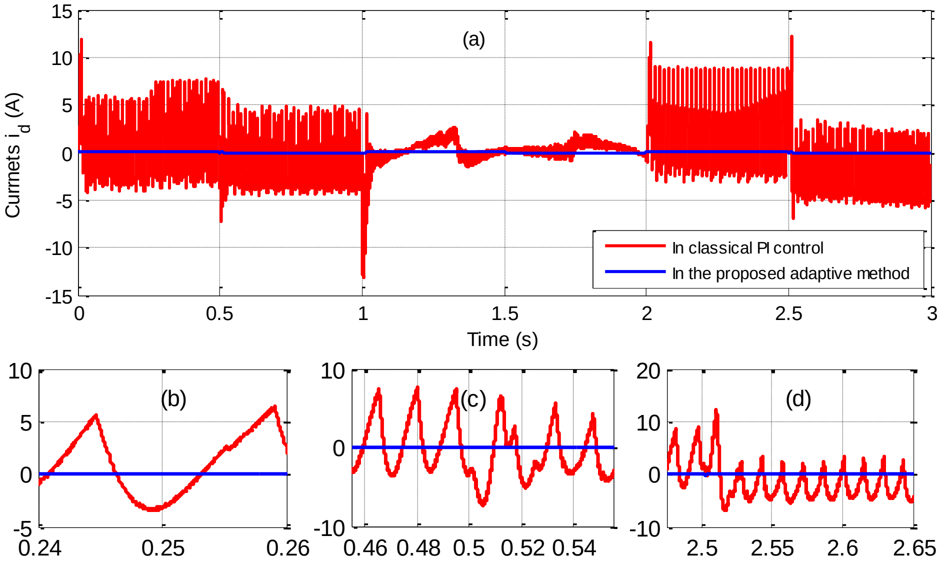

Figure 9. Evolution of the currents i_d response in the presence of uncertainties: (a) currents response, (b) zoom around 0.25s, (c) zoom around 0.5s, (d) zoom around 2.5s

The Figure 10 plots the load torque disturbances variation and the electrical torque. Clearly, the zoomed Figures 10(b) and 10(c) show that the torque in the proposed method follow the load torque variation without ripples in the steady state time, whereas the classical vector control loses its reference and presents more ripples. As can be seen from Figure 10(d) the proposed adaptive method drops with short time to desired values and the classical vector control presents large ripples with long settling time during the speed reversing. Thus, the proposed adaptive method still gains better control performance over the classical vector control.

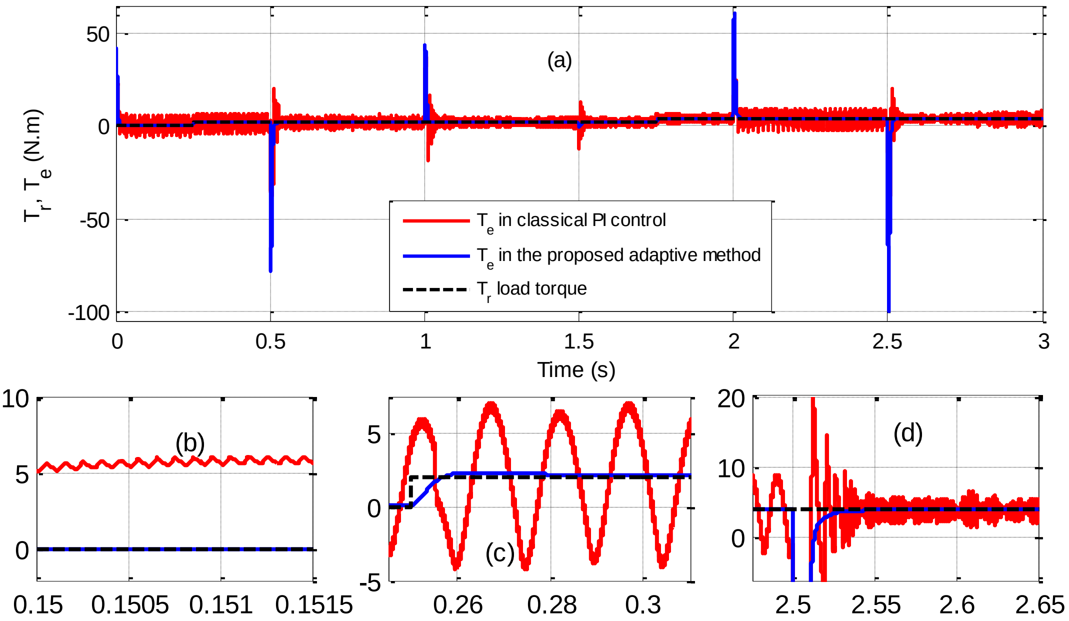

Figure 10. Evolution of the torque responses in the presence of uncertainties: (a) Torque responses, (b) zoom around 0.15s, (c) zoom around 0.37s, (d) zoom around 2.5s

Once the speed and currents control system are designed with and without uncertainties, we proceed to prove the robustness of the proposed technique for uncertain PMSM by applying sinusoidal speed reference. The dynamic reference and actual speed response of the proposed adaptive method are given in Figure 11(a). We can observe that the disturbances rejection capacity of the proposed adaptive method leads to a good speed tracking performance. In addition, the Figure 11(b) presents the tracking errors. It shows that the asymptotic speed tracking objective is obtained with good accuracy under uncertainties.

Furthermore, in the Figure 12 the evolution of the currents i_d and i_q and their tracking errors are presented. Since the reference speed trajectory is in sinusoidal form, the currents, i_d and i_q follow the same form as is shown in Figure 12(a), we note also that the current i_d is closely near to zero according to the control strategy. The Figure 12(b) shows that the tracking errors trends to zero with sinusoidal form due to the sinusoidal form of the speed reference.

The Figure 13 presents the load and electrical torques. We can observe from this figure that the electrical torque has also sinusoidal form with the average equal to the step of load torque disturbance. This is due to the sinusoidal form of the lumped disturbance and the uncertainty effect on the PMSM resulting from the sinusoidal form of the speed reference. As can be seen from this figure, precise control performance is reached during both transient and steady state times. In addition, we can observe that the tracking errors remain near to zero stably, in the presence of any uncertainty.

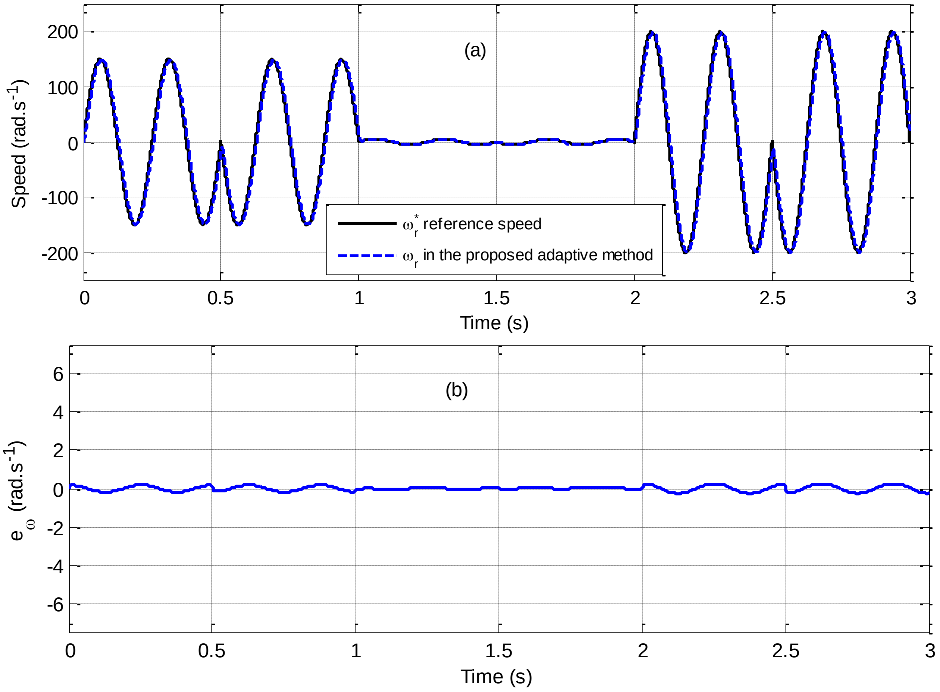

Figure 11. Speed response for sinusoidal reference in the proposed adaptive control in the presence of uncertainties: (a) Speed response and (b) speed tracking errors

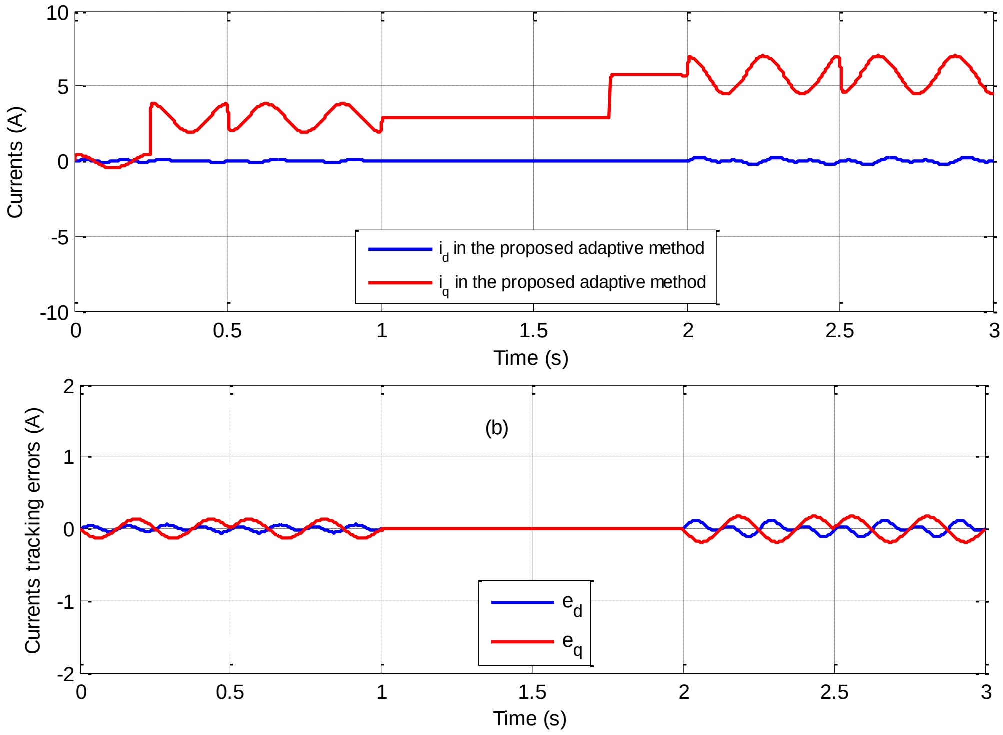

Figure 12. Currents response in the proposed adaptive control in the presence of uncertainties: (a) Currents response and (b) currents tracking error

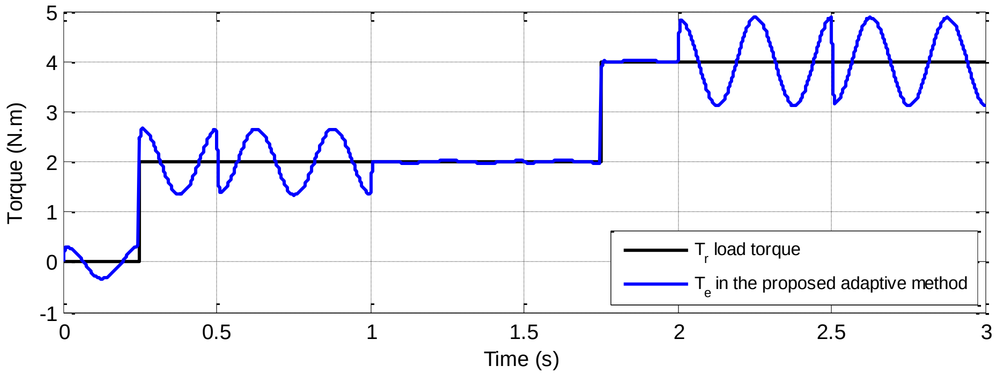

Figure 13. Evolution of the torque in the proposed adaptive control in the presence of uncertainties

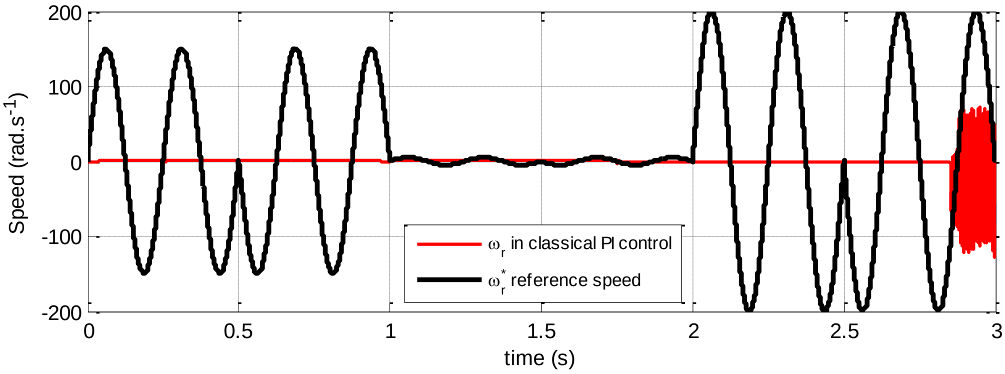

Figure 14. Trajectories of the desired and actual speed with classical vector control in the presence of uncertainties

The proposed adaptive method, for its part, maintains its performance because of the adaptive compensation by reacting quickly to clear the errors occurred by the uncertainty. Whereas the classical vector control loses its reference completely in this case, as is demonstrated in Figure 14.

From the results of these simulations, it is seen clearly that the proposed adaptive method can suppress the uncertain behavior coming from mathematical modelling, parameters uncertainty and load torque disturbance in PMSM drive system and achieve a good tracking performance whatever the references of desired input signals.

5. CONCLUSION

This paper presents a robust speed-control strategy using stable and adaptive neural network compensators for the uncertain permanent magnet synchronous motor drives. The neural network compensators are based on the on-line parameter training methodology and they are designed to stabilize the PMSM despite of the uncertainties. The Hurwitz technique is used to determine the current controller gains, whereas the speed control is processed by PI controller. The stability of the closed-loop system is proven by using the Lyapunov function. Simulations are conducted to demonstrate the effectiveness of the proposed control scheme. Compared precisely to classical vector control, the obtained results show that all signals of the closed-loop system are tracked perfectly under the uncertainties conditions.

REFERENCES

[1] Luo, Y., Chen, Y., Pi, Y. (2010). Cogging effect minimization in PMSM position servo system using dual high-order periodic adaptive learning compensation. ISA Transactions, 49(4): 479-488. https://doi.org/10.1016/j.isatra.2010.05.003

[2] Liu, H., Li, S. (2011). Speed control for PMSM servo system using predictive functional control and extended state observer. IEEE Transactions on Industrial Electronics, 59(2): 1171-1183. https://doi.org/10.1109/TIE.2011.2162217

[3] Quang, N.P., Dittrich, J.A. (2008). Vector control of three-phase AC machines (Vol. 2). Heidelberg: Springer.

[4] Lan, Y.H. (2018). Backstepping control with disturbance observer for permanent magnet synchronous motor. Journal of Control Science and Engineering. https://doi.org/10.1155/2018/4938389

[5] Chen, G., Zhou, Y., Gao, T., Zhou, Q. (2015). Unknown disturbance estimation for a PMSM with a hybrid sliding mode observer. Mathematical Problems in Engineering. https://doi.org/10.1155/2015/316360

[6] Lin, F.J., Chiu, S.L. (1998). Adaptive fuzzy sliding-mode control for PM synchronous servo motor drives. IEE Proceedings-Control Theory and Applications, 145(1): 63-72. https://doi.org/10.1049/ip-cta:19981683

[7] Wai, R.J. (2001). Total sliding-mode controller for PM synchronous servo motor drive using recurrent fuzzy neural network. IEEE Transactions on Industrial Electronics, 48(5): 926-944. https://doi.org/10.1109/41.954557

[8] Elmas, C., Ustun, O., Sayan, H.H. (2008). A neuro-fuzzy controller for speed control of a permanent magnet synchronous motor drive. Expert Systems with Applications, 34(1): 657-664. https://doi.org/10.1016/j.eswa.2006.10.002

[9] Maiti, S., Chakraborty, C., Sengupta, S. (2009). Simulation studies on model reference adaptive controller based speed estimation technique for the vector controlled permanent magnet synchronous motor drive. Simulation Modelling Practice and Theory, 17(4): 585-596. https://doi.org/10.1016/j.simpat.2008.08.017

[10] Rayd, H., El Idrissi, A.E.J., Zahid, N., Jedra, M. (2014). Robust emerged artificial intelligence speed controller for PMSM drive. 2014 Second World Conference on Complex Systems (WCCS), Agadir, Morocco, pp. 510-517. https://doi.org/10.1109/ICoCS.2014.7060952

[11] Yu, J., Gao, J., Ma, Y., Yu, H., Pan, S. (2010). Robust adaptive fuzzy control of chaos in the permanent magnet synchronous motor. Discrete Dynamics in Nature and Society. https://doi.org/10.1155/2010/269283

[12] Ran, J., Li, Y., Wang, C. (2018). Chaos and complexity analysis of a discrete permanent-magnet synchronous motor system. Complexity, 2018. https://doi.org/10.1155/2018/7961214

[13] Beniysa, M., El Idrissi, A. E. J., Bouajaj, A., Britel, M. R. (2020). A robust control of permanent magnet synchronous generator based wind energy conversion system via online-tuned artificial neural network compensators. International Conference on Artificial Intelligence & Industrial Applications, Meknes, Morocco, pp. 47-60. https://doi.org/10.1007/978-3-030-53970-2_5

[14] Aguilar-Mejía, O., Tapia-Olvera, R., Valderrabano-González, A., Rivas Cambero, I. (2016). Adaptive neural network control of chaos in permanent magnet synchronous motor. Intelligent Automation & Soft Computing, 22(3): 499-507. https://doi.org/10.1080/10798587.2015.1103971

[15] Poznyak, A.S., Sanchez, E.N., Yu, W. (2001). Differential neural networks for robust nonlinear control: identification, state estimation and trajectory tracking. World Scientific.

[16] Rivals, I., Personnaz, L. (1998). A recursive algorithm based on the extended Kalman filter for the training of feedforward neural models. Neurocomputing, 20(1-3): 279-294. https://doi.org/10.1016/S0925-2312(98)00021-6

[17] Bose, B.K. (2002). Modern Power Electronics and AC Drives. Prentice Hall PTR, Upper Saddle River, NJ 07458.

[18] Igelnik, B., Pao, Y.H. (1995). Stochastic choice of basis functions in adaptive function approximation and the functional-link net. IEEE transactions on Neural Networks, 6(6): 1320-1329. https://doi.org/10.1109/72.471375

[19] El-Sousy, F.F. (2014). Adaptive hybrid control system using a recurrent RBFN-based self-evolving fuzzy-neural-network for PMSM servo drives. Applied Soft Computing, 21: 509-532. https://doi.org/10.1016/j.asoc.2014.02.027

[20] de Jesús Rubio, J., Yu, W. (2007). Nonlinear system identification with recurrent neural networks and dead-zone Kalman filter algorithm. Neurocomputing, 70(13-15): 2460-2466. https://doi.org/10.1016/j.neucom.2006.09.004

[21] Lewis, F.L., Campos, J., Selmic, R. (2002). Neuro-fuzzy control of industrial systems with actuator nonlinearities. Society for Industrial and Applied Mathematics.

[22] Lewis, F.L. (2006). Neural network control of nonlinear discrete-time systems, CRC Press, Taylor & Francis Group.

[23] Sanchez, E.N., Ornelas-Tellez, F. (2013). Discrete-time inverse optimal control for nonlinear systems, CRC Press Taylor & Francis Group.

[24] Mohamad, S. (2008). Exponential stability preservation in discrete-time analogues of artificial neural networks with distributed delays. Journal of Computational and Applied Mathematics, 215(1): 270-287. https://doi.org/10.1016/j.cam.2007.04.009

[25] Luo, W., Zhong, K., Zhu, S., Shen, Y. (2014). Further results on robustness analysis of global exponential stability of recurrent neural networks with time delays and random disturbances. Neural Networks, 53: 127-133. https://doi.org/10.1016/j.neunet.2014.02.007

[26] Zhang, X.M., Han, Q.L. (2014). Global asymptotic stability analysis for delayed neural networks using a matrix-based quadratic convex approach. Neural Networks, 54: 57-69. https://doi.org/10.1016/j.neunet.2014.02.012

[27] Arik, S. (2014). An improved robust stability result for uncertain neural networks with multiple time delays. Neural Networks, 54: 1-10. https://doi.org/10.1016/j.neunet.2014.02.008

[28] Zhu, Y.H., Cheng, D.Z., Qin, H.S. (2007). Constructing common quadratic Lyapunov functions for a class of stable matrices. Acta Automatica Sinica, 33(2): 202-204. https://doi.org/10.1360/aas-007-0202

[29] Eghbal, N., Pariz, N., Karimpour, A. (2013). Discontinuous piecewise quadratic Lyapunov functions for planar piecewise affine systems. Journal of Mathematical Analysis and Applications, 399(2): 586-593. https://doi.org/10.1016/j.jmaa.2012.09.054

[30] King, C., Nathanson, M. (2006). On the existence of a common quadratic Lyapunov function for a rank one difference. Linear Algebra and its Applications, 419(2-3): 400-416. https://doi.org/10.1016/j.laa.2006.05.010

[31] Griggs, W.M., King, C.K., Shorten, R.N., Mason, O., Wulff, K. (2010). Quadratic Lyapunov functions for systems with state-dependent switching. Linear Algebra and Its Applications, 433(1): 52-63. https://doi.org/10.1016/j.laa.2010.02.011

[32] Dai, D., Hu, T., Teel, A.R., Zaccarian, L. (2009). Piecewise-quadratic Lyapunov functions for systems with dead zones or saturations. Systems & Control Letters, 58(5): 365-371. https://doi.org/10.1016/j.sysconle.2009.01.003

[33] El Idrissi, A.E.J., Zahid, N., Jedra, M. (2012). Optimized DTC by genetic speed controller and inverter based neural networks SVM for PMSM. In Second International Conference on the Innovative Computing Technology (INTECH 2012), Casablanca, Morocco, pp. 392-395. https://doi.org/10.1109/INTECH.2012.6457760

[34] Slotine, J.J.E., Li, W. (1991). Applied Nonlinear Control (Vol. 199, No. 1). Englewood Cliffs, NJ: Prentice Hall.