Nova Scientia
Revista de Investigación de la Universidad De La Salle Bajío

Nº 22, Vol. 11 (1), 2019. **ISSN 2007 – 0705**, pp.: 142 - 170
doi.org/10.21640/ns.v11i22.1614

Controlador de Velocidad Adaptativo para un Motor Síncrono de Imanes Permanentes
Adaptive Speed Controller for a Permanent Magnet Synchronous Motor

Omar Aguilar Mejía[1], Rubén Tapia Olvera[2], Iván Rivas Cambero[3] y Hertwin Minor Popocatl[1]

Palabras clave: lógica difusa; redes neuronales artificiales; control de máquinas; modos deslizantes; máquinas de imanes permanentes
Keywords: fuzzy logic; artificial neural network; motor drives; sliding mode; permanent magnet machines

Recepción: 08-08-2018 / Aceptación: 10-01-2019

Resumen

En este trabajo se presenta el desempeño de un controlador adaptativo utilizando una red neuronal B-spline para ajustar la velocidad del rotor del motor síncrono de imanes permanentes. El análisis incluye una comparación de tres estrategias de control convencionales y no lineales: control proporcional integral, modos deslizantes y lógica difusa. Además, se proporciona una forma sistemática de determinar las ganancias óptimas del controlador para mejorar el desempeño del error de seguimiento. Dentro del trabajo se explica el diseño del controlador adaptativo y la metodología para realizar su entrenamiento. La eficiencia del controlador propuesto se demuestra mediante simulaciones en el tiempo donde el motor se somete a diferentes perturbaciones y cambios en el valor de la señal de referencia. El análisis de los resultados muestra que la técnica de control propuesta exhibe mejor rendimiento porque puede adaptarse a todas las condiciones operativas, presentando un bajo esfuerzo computacional para una operación en línea y considerando las no linealidades del sistema.

Abstract

This paper presents a controller performance that is develop employing an adaptive B-spline neural network algorithm for adjusting the rotor speed of the permanent magnet synchronous motor. It includes a comparative analysis with three control strategies: conventional proportional integral, sliding mode and fuzzy logic. Also, gives a systematic way to determine the optimal control gains and improve the tracking error performance. A methodology for the adaptive controller and its training procedure are explained. The efficacy of the proposed method is analyzed using time simulations where the motor is subjected to disturbances and reference

[1] Universidad Popular Autónoma del Estado de Puebla, Departamento de Posgrado. E-mail: omar.aguilar@upaep.mx
[2] Universidad Nacional Autónoma de México, Departamento de Energía Eléctrica
[3] Universidad Politécnica de Tulancingo, Departamento de Investigación y Posgrado

changes. The proposed control technique exhibits the best performance because it can adapt to every condition, demanding low computational effort for an on-line operation and considering the system nonlinearities.

Introduction

The permanent magnet synchronous motors (PMSM) has paid and increased popularity in industrial and aerospace applications due the rapid advance in the development of semiconductor technology, new theories of electrical machines control and new magnetic materials that using rare earth. These machines are used in high performance applications for its excellent efficiency, no reactive power demand from the source, compact structure, low maintenance, low inertia and better dynamic performance compared with the induction motor (Omrane *et al.*, 2015, Liu *et al.*, 2018, Englert and Graichen, 2018). These benefits and performance are applied in electric vehicles, airplanes, nuclear power plants, submarines, robots, computer numerical control machines, elevators and propellers of ships (Englert and Graichen, 2018, Jun-Jien *et al.,* 2015, Qutubuddin and Narri Yadaiah, 2018, Kumar *et al.*, 2014). However, in some situations the PMSM dynamic performance is diminished significantly by some parametric disturbances (variations in temperature, magnetic saturation, etc.), external load torque variations and non-linearities in the dynamic model. Recently, in the literature there are several studies, they use different control techniques to ensure robust performance despite this problem (Liu *et al.*, 2018, Jun-Jien *et al.,* 2015, Englert and Graichen, 2018, Kumar *et al.*, 2014). Several advanced controllers use non-linear control techniques (Verrelli *et al.*, 2017), adaptive schemes (Hu *et al.*, 2017, Iqbal *et al.*, 2014), linear control theory (Ortega *et al.*, 2018), neural networks (Jon *et al.*, 2017), sliding modes (Lu *et al.*, 2017). These methods have been used in different applications where the main purpose is for regulating the PMSM rotor speed. In the industry, the proportional-integral-derivative (PID) controllers are widely utilized to manipulate the PMSM performance due to its simplicity and effectiveness. However, the linear controllers are sensitive to parametric variations, noise, and external disturbances, in most cases tuning is very difficult and tedious due that its performance is guarantee for some operation conditions. It is worth mentioning that the performance and efficiency of the entire control system depends not only on the efficient hardware software platform, also the control law robustness (Kumar *et al.*, 2014).

To increase the robustness of the proportional-integral (PI) controllers it is possible to incorporate one or two innovative control techniques, such as: adaptive control, sliding mode, predictive control, robust control and intelligent control. Some problems in design procedure are the need to incorporate internal or external disturbances in the model, causing that this process become complex and difficult (Jun-Jien *et al.*, 2015). Intelligent control techniques provide a good dynamic performance when parametric variations and external disturbances occur but its implementation requires known prior of several parameters before starting his training (Choi *et al.*, 2015). In the literature, there are different techniques of robust control and auto-tuning used to improve system response and increase the robustness of the linear controllers (Jin-Woo *et al.*, 2014). The auto-tuning techniques can be used in some applications where the dynamic plant model cannot represent in detail the physical system.

On many occasions, it is a common practice use expert systems when the model plant is incompletely modeled, for instance neural networks, fuzzy logic or a combination within the control schemes to regulate non-linear systems. However, during the design process of a fuzzy system there are significant problems associated with specialist expertise, accuracy and the correct application of the rules to regulate the plant (Chang *et al.*, 2016). In (Choi *et al.*, 2015) a fuzzy logic for updating the gains of a PID controller to regulate the PMSM rotor speed subject to different load torque conditions is used, but the details of how select the membership functions of the fuzzy system are not explained. Besides, artificial neural networks (ANN) are widely used in pattern identification, adaptive filtering, modeling and controlling complex systems due they are able of learning. Recently, (Beltran-Carbajal *et al.*, 2018) is presented an adaptive PID controller using ANN to update on-line the gains of the controller structure, but the training process does not ensure stability in closed loop. The on-line learning process of the ANN is to update weights and base functions continuously throughout the operation of the control scheme. In most cases this process can take too many multiplications and divisions demanding hard computational effort or must have a microprocessor too fast (Nizami *et al.*, 2017). To avoid these problems in literature there are ANN called B-spline (BSNN), which are a type of ANN that utilizes polynomial basis functions, known as B-spline basis functions. The BSNN has the ability to update only one or more weights of the network structure according to the input and desired output, resulting faster speed of convergence and lower computational complexity compared than backpropagation algorithm (Tapia-Olvera *et al.*, 2016).

In this paper the dynamic performance of four different control schemes to regulate the rotor speed of a PMSM subject to different disturbances are compared. The first control scheme consists in tuning two PI controllers using a first order dynamic model of the PMSM. The simplified model is obtained disengaging the dynamic equations of the stator currents using two control variables, one for each of the stator currents in dq reference frame. The second controller is an adaptive regulator based in BSNN to update on-line the PI controller gains is proposed, using the first controller architecture. The controller gains are updated on-line using the gradient descent method and BSNN. The following proposed control system is based on fuzzy logic that depends on the membership and distribution functions, and user experience. Finally, a non-linear controller based on the technique of control by sliding modes is used. A nonlinear observer to estimate the corresponding external disturbance is applied, due that the control law require the load torque.

The paper is organized as follows: section 2 gives an introduction to the mathematical model of the PMSM in dq reference frame; the problem formulation, PI controller and estimator design are presented in section 3; in section 4 the conventional PI controller tuning, a proposed neural network controller, fuzzy logic and sliding mode control are analyzed. The comparison results of the four control schemes are discussed in section 5, they were performed in Matlab/Simulink®.

Dynamic model of PMSM

The vector control technique is used in the PMSM control, this scheme allows to decoupling the motor currents. Therefore, it is possible to consider its characteristics similar to a direct current (DC) motor because it works only with two currents, typically called excitation and torque currents.

The controller design needs that the dynamic equations must be transformed from the stator reference frame, *abc*, to a synchronous rotor reference frame, *dq*. First, the three-phase *abc* coordinate system is transformed to a two-phase reference frame $\alpha\beta$, where α-axis coincides with the a-axis phase and the β-axis leads by 90° relative to the α-axis in the clockwise direction. The permanent magnet mounted on the rotor of PMSM and the vector of flows are in the same position as the magnetic pole. Then, the $\alpha\beta$ coordinate system is transformed to *dq* rotor reference frame using the Park transformation. With this transformation the *d*-axis is aligned in the same direction to the magnetic flux of the rotor and the q-axis leads by 90° to *d*-axis. The angle between d-axis and a-axis is the rotor electrical angle, θ_e. In this paper, the nonlinear dynamic model of the PMSM is used in a dq rotor stationary reference frame (Yazdani & Iravani, 2010, Krishnan, 2010). This model consist of a set of nonlinear differential equations that relate the electrical variables: currents, voltages and flow links with mechanical variables: torque and rotor speed. The electrical equations of the motor in dq rotor reference frame are:

$$\begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} -r_s & \omega_e L_q \\ -\omega_e L_d & -r_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} - \begin{bmatrix} 0 \\ -\omega_e \varphi_m \end{bmatrix} + \begin{bmatrix} v_d \\ v_q \end{bmatrix} \tag{1}$$

$$T_e = 1.5 \frac{n}{2} \left[\varphi_m i_q + \left(L_d - L_q \right) i_d i_q \right] \tag{2}$$

Where v_d and v_q are the voltages applied to the stator in dq reference frame, respectively; i_d and i_q are the currents in dq; r_s is the stator resistance; L_d and L_q are the stator inductances in dq axis; ω_e is the electrical angular speed of the rotor; T_e is the electromagnetic torque; n is the number of pole pairs and; φ_m is the magnetic flux in the rotor produced by the permanent magnet. The dynamic equations of the rotor position and rotor speed are:

$$\frac{d\omega_r}{dt} = \frac{T_e - T_l}{J} \tag{3}$$

$$\frac{d\theta_r}{dt} = \omega_r \tag{4}$$

Where ω_r is the mechanical speed of the rotor; J is the inertia moment; θ_r is the rotor position and; T_l is the load torque applied to the motor. This paper considers a nonsalient PMSM with magnets mounted on the rotor surface (Omrane *et al.*, 2015), whereby the d-axis and q-axis inductances are equal, consequently the electromagnetic torque now can be calculated by:

$$T_e = 0.75 n \varphi_m i_q \tag{5}$$

Three control loops with negative feedback for speed regulation are used, this allows obtaining the desired rotor speed and improving steady state and transient performance of the motor. The control system includes a loop for the rotor speed, a control loop in d-axis current and one in q-axis. Figure 1 shows the block diagram of overall control system for PMSM drive system. The control system consists of an AC-DC voltage source converter, a PMSM, a DC-AC voltage source inverter, an automatic current control, a position sensor that determines the speed of the rotor and a speed regulator. Multiplying the rotor mechanical angle by the number of poles, the electric angle is calculated.

The speed reference ω^*; the desired direct axis current i_d^*; and the rotor speed ω_r, are the inputs to the speed control system, Figure 1. The desired current i_q^* is determined from the electromechanical torque, which is proportional to the current in q-axis.

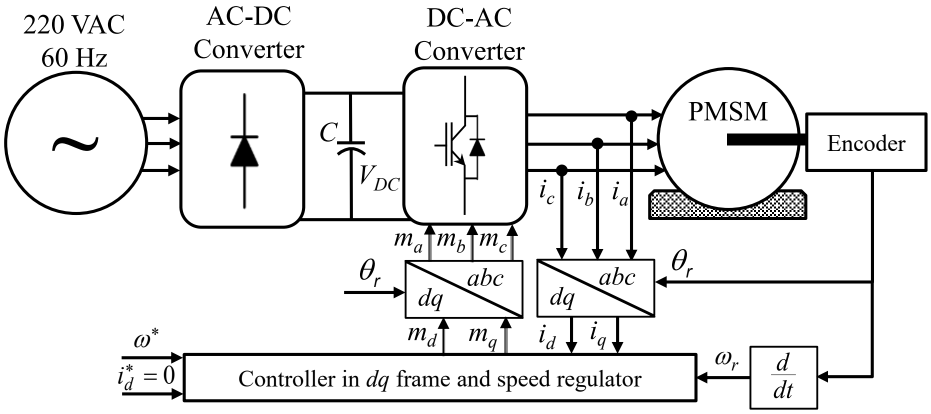

Figure 1. Block diagram of overall control system for PMSM drive system

Field Oriented Control for PMSM

Design of the control scheme: The PMSM model (1), (3)-(5) is a dynamic system with two control inputs, v_d and v_q. If in some way can make the component of the direct axis current is equal to zero, a similar behavior to a DC motor with excitation constant flux is obtained (Choi *et al.*, 2015). The equations indicate that the PMSM is a nonlinear system due to coupling between the different state variables (currents and rotor speed). We can also observe that the parameters depend on the operating conditions, mainly load torque and operating temperature. The control objective is to design an asymptotically stable speed controller for PMSM to make the rotor speed track the reference trajectory correctly under different parameter perturbations and load torque disturbances. Therefore, the main signal error can be defined as:

$$e_\omega = \omega_r^* - \omega_r \tag{6}$$

Where e_ω is the rotor speed error, ω_r^* is the desired rotor speed. Equation (1) has two independent control variables v_d and v_q, if the control signal v_q regulates the motor speed; we have a freedom degree available, v_d, and can take advantage. One advantage is to prevent the demagnetization of the permanent magnets, which is accomplished if the direct axis current is zero (Yazdani & Iravani, 2010, Krishnan *et al.*, 2010). If the above condition is known, the stator current is equal to the q-axis current, and then the electromagnetic torque is proportional to i_q. Another advantage is that the constant flow links have a linear relationship with the current in the stator; thereby its values can be estimate in an easy way. We can define the tracking error of the d-axis current as:

$$e_d = i_d^* - i_d \tag{7}$$

Where i_d^* is the d-axis current reference value. In theory, the proposed controller forces the current i_d to have a value of zero in a finite time, so that the electric torque is proportional to current i_q. Therefore, the current error in q-axis is defined as follows:

$$e_q = i_q^* - i_q \tag{8}$$

Where i_q^* is the q-axis current reference value. The proposed control system is based on a sinusoidal pulse width modulation (SPWM) scheme; m_q and m_d are the modulating signals in dq reference frame, Figure 1. The modulation signals are transformed to *abc* system using the Park transformation and rotor position θ_r, for comparison with high-frequency carrier signal.

Calculation of the desired current i_q^*: The desired q-axis current requested for closed loop control system is determined by differentiating (6) with respect to time and substituting the derivative of ω_r:

$$\dot{e_\omega} = \dot{\omega}_r^* - \dot{\omega}_r = \dot{\omega}_r^* - \left[0.75 n\varphi_m i_q - T_l\right]\frac{1}{J} \tag{9}$$

The speed tracking error can be defined as:

$$\dot{e_\omega} = -ce_\omega \tag{10}$$

Where $c>0$, and the convergence speed of (10) depends of the constant c. Equaling (9) and (10) and solving for i_q, the desired current $i_q{}^*$ is obtained for the quadrature axis current controller:

$$i_q^* = \frac{J}{0.75n\varphi_m}\left(\dot{\omega}_r^* + \frac{T_l}{J} + ce_\omega\right) \tag{11}$$

We can see that the desired current value is a function of three variables: derivative of the desired reference speed trajectory, the rotor speed and the load torque, so it is necessary to measure or estimate.

Estimated load torque: In practice, the load torque is unknown, therefore, it is necessary to design an estimator to know its value. The nonlinear estimator is described by (Moreno & Osorio, 2008):

$$\frac{d\widehat{\omega}_r}{d_t} = \frac{i_q k_\omega - \hat{T}_l}{J} - l_1 e_s \tag{12}$$

$$\frac{d\hat{T}_l}{dt} = l_2 e_s \tag{13}$$

Where e_s is the estimation error, $\widehat{\omega}_r$ is the estimated mechanical rotor speed; $\hat{T}_l$ is the estimated mechanical torque; e_s is the estimated error speed; l_1 and l_2 are the estimator gains; $k_\omega = 0.75n\varphi_m$. Combining (12) and (13) and the dynamic equation of the rotor speed, we can obtain a set of equations describing the dynamic tracking error estimator:

$$\begin{bmatrix} \dot{\bar{\omega}}_r \\ \dot{\bar{T}}_l \end{bmatrix} = \begin{bmatrix} -l_1 & -\frac{1}{J} \\ l_2 & 0 \end{bmatrix} \begin{bmatrix} \bar{\omega}_r \\ \bar{T}_l \end{bmatrix} \tag{14}$$

The system (14) is an autonomous linear system, asymptotically stable with the appropriate choice of the l_1 and l_2 parameters (Aguilar-Mejía *et al.*, 2016).

The control scheme for PMSM is exposed in previous section, however, a proper operation requires robust strategies or adaptive control that ensure safe and reliable performance also must respond quickly and appropriately to various scenarios that it can endure. In this paper, three schemes for tuning and implementation of field-oriented control are proposed, by comparing the performance with an adaptive PI controller.

Design of conventional PI controller for PMSM

For regulating the PMSM rotor speed and improve its dynamic performance first a conventional PI control scheme is designed, Figure 2. Considering (1) as in (Heng *et al.*, 2010) the following new control signals are defined:

$$v_d = V_q - L_q i_q \omega_e \tag{15}$$

$$v_q = V_q - L_d i_d \omega_e + \omega_e \varphi_m \tag{16}$$

Substituting (15) and (16) in (1), the following dynamic first order system is obtained:

$$L_d \frac{di_d}{dt} + r_s i_d = V_d \tag{17}$$

$$L_q \frac{di_q}{dt} + r_s i_q = V_q \tag{18}$$

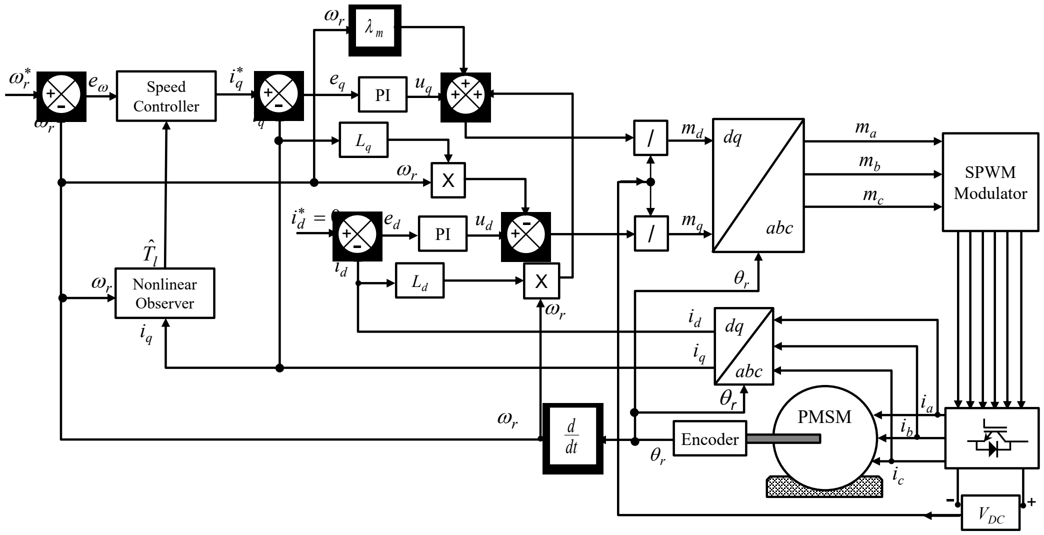

Figure 2. Block diagram for regulating the PMSM speed using two conventional PI controllers

The PMSM speed control using two conventional PI controllers, two different closed loop control systems are necessary to regulate i_d and i_q, Figure 2. The controller consists of two inner loops, which regulate the stator currents in the dq stationary reference frame. Figure 2 shows that the controller for i_d handle the error signal $e_d = i_d^* - i_d$ and calculates v_d. The desired current value (i_d^*) is to improve the performance and avoid physical damage to the magnetic components of the machine (Krishnan, 2010).

Similarly, the controller for i_q, process the error signal $e_q = i_q^* - i_q$ and computes v_q. The desired current value (i_q^*) is determined by the electromagnetic torque control loop, that it is a function of the reference speed.

Assuming that the transfer functions of the closed loop (17) and (18) are the first degree and similar, the PI controllers are:

$$\frac{V_d(s)}{E_d(s)} = k_p + \frac{k_i}{s} \tag{19}$$

Where $V_d(s)$ is the Laplace transform of the control signal, $E_d(s)$is the Laplace transform of the error signal, $k_p = L_d/\tau$ and $k_i = k_p(r_s/L_d)$. The τcoefficient is the time delay of the controller response. The control signals that provide the current controllers are transformed to *abc* reference frame, for comparison with a triangular signal. The pulses generated by the two signals comparison are sent to the power semiconductor devices in the voltage source inverter for feeding the PMSM.

Adaptive control using B-spline neural networks

Proposition: The controller design is one of the major advantages of the artificial neural networks. In this section an adaptive velocity trajectory tracking control scheme for nonlinear PMSM subjected to variable mechanical load torque disturbances is used. Artificial neural networks is employed in the synthesis of the adaptive motor control scheme based on velocity tracing. Thus, the tracking control approach has the capability to adapt or learn on-line from diverse operating scenarios. A different perspective for control synthesis based on output signal modeling into arbitrary small intervals of time is adopted. The independence of some detailed mathematical model of the nonlinear electromechanical system is also achieved by the B-spline neural networks (BSNN) control approach. In addition, a priori knowledge or real time estimation of disturbances is unnecessary. Furthermore, BSNN have important desirable features for efficiently controlled physical systems due to them can update their response on-line, considering a simple structure and a small number of basic mathematical operations. Their adaptive nature allows learning by means of examples, where a representative set of measured data, providing a relationship between inputs and outputs, is enough. This characteristic makes them suitable in application domains where there is little or incomplete understanding of the system to be controlled, but where training data are available.

Neural Network Structure: The (BSNN) are a particular case of neural networks useful for on-line controlling and modeling, taking into account the system non-linearities. Some control parameters must be specified, such as the shape of the basic functions, Figure 3, and the learning rate. Despite the fact that the BSNN structure is specified, it can be adapted to achieve a satisfactory performance over a wide range of operating conditions, since its weighting vector is updated on-line over each sampling time. That is, the BSNN can adaptively be updated for output tracking when the system operating point is modified or an external disturbance arises.

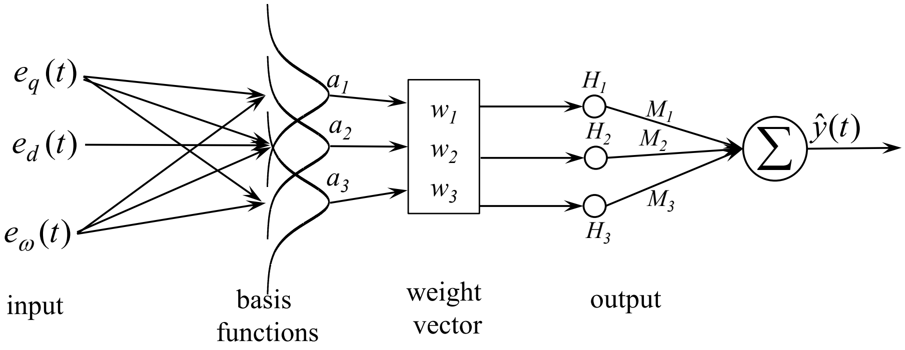

Figure 3. Proposed B-spline neural network structure

This ANN exhibits better characteristics compared with other intelligent neural network, for example: the ANN backpropagation and ANN radial base function. BSNN in this case consists of three layers, namely: 1) a l-dimensional input space standard; 2) a set of basis functions p; and 3) the output function of the network, Figure 3.

Through parameters k_p and k_i, the major task of the adaptive controller is to regulate the rotor speed control, Figure 3. Three error signals are employed to calculate k_p and k_i. $e_{d,q}$ are the difference between the dq-axis currents references and the measured currents i_d and i_q, e_ω is the difference between desire and actual speed, finally, its derivative, Figure 3. A similar structure for each control gain is used. For determining the output value has as inputs: past and actual inputs and outputs, based on those values determine the optimal value of the controller gains, so that the error approaches to zero.

The BSNN output can be described by (Brown & Harris, 1995), Figure 2:

$$\hat{y} = \sum_{i=1}^{P} H_i M_i = \mathbf{H}^T \mathbf{M}(\mathrm{t}) \tag{20}$$

Where h_i is the weight connecting the i-th hidden neuron to the output neuron, m_i is the output of the i-th hidden neuron and p is the number of weights. **M**(t) can be expressed as:

$$\mathbf{M}_{\boldsymbol{i}}(\mathrm{t}) = \mathrm{b}_i + \sum_{i=1}^{p} w_i e_j + w_{i+1}\mathbf{M}_{\boldsymbol{i-1}} \tag{21}$$

Where b_i is the bias of *i*th hidden neuron, w_i is the weight connecting ith input to the *j*th hidden neuron and e_j is the *j*th input. The proposed adaptive controller is composed by two BSNN, one to update k_p and another to update k_i, for each PI controller. Thus, the networks are described as follows:

$$k_p^d = NN_i\left(e_q, e_d, e_\omega, w_j, H_i, M_i\right) \tag{22}$$

$$k_i^d = NN_i\left(e_q, e_d, e_\omega, w_j, H_i, M_i\right) \tag{23}$$

Where i=1,2,3; NN_i denotes the BSNN which is used to calculate k_p and k_i in *d-q* axis, respectively. **Learning Rule:** To design the proposed controller it is decided that the learning rule is a gradient descent that is suitable for BSNN, instead of recursive algorithms based on least squares that are more complicated. For updating the weights Δ**w** an instantaneous learning rule is used (Brown & Harris, 1995):

$$\Delta\mathbf{w}(t-1) = \frac{\gamma e_x(t)}{\|\mathbf{a}(t)\|_2^2}\mathbf{a}(t) \tag{24}$$

Where γ is the learning ratio; **a** is the vector containing the outputs of the basis functions; **w** is the weight vector and, e_x is the desired B-spline output and the actual one difference.

Convergence analysis: The instantaneous learning rule requires the minimization of the following cost function, **J** = $\mathbf{E}^2$ (Kumar *et al.*, 2014) where **E(E=R-Y)** is the error between the reference input (**R**) and desired output (**Y**):

$$\|E\|^2 = (R - MH)^T(R - MH) \tag{25}$$

$$\|E\|^2 = R^T R - 2M^T H + M^2 H^T H \tag{26}$$

The objective of the learning rule is to find **H** and **M** so that $\|E\|^2$ is minimized:

$$\frac{\partial\|E\|^2}{\partial M} = -2R^T H + 2MQH^T H \tag{27}$$

$$\frac{\partial\|E\|^2}{\partial w} = -2M\frac{\partial H^T}{\partial w}R^T + 2M^2\frac{\partial H^T}{\partial x}H \tag{28}$$

The optimum value of **H** and **M** is obtained when (27) y (28) are iqual to zero. From (27), the output weight matrix is calculated as $H = \frac{R^T H}{H^T}$H. The calculate of output weights in a single step can make learning speed at least 10–1000 times faster as compared to the gradient based methods where error has to back propagate recursively. Once the parameters of the output layer are obtained, for each incoming data on line least square sequential learning is employed (Serre, 2002). First the layer w_i is calculated, and then the output weight is determined analytically using orthogonal projection method $M_0 = (H^T \mathrm{H})^{-1}\, H^T Y$. For the next portion of data, the following function is minimized:

$$\left\|\begin{bmatrix} H_0 \\ H_1 \end{bmatrix} M - \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix}\right\| \tag{29}$$

From (29) and (24), considering both data samples, the output weight is:

$$M_1 = K_1^{-1}\begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} \tag{30}$$

Where:

$$K_1 = \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \tag{31}$$

$$\begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = K_1 M_0 - H_1^T H_1 M_0 + -H_1^T Y_1 \tag{32}$$

For sequential learning we have to express *M₁* as a function of *M₀*. From (31) *K₁* can be expressed as:

$$K_1 = \begin{bmatrix} H_0^T & H \end{bmatrix}\begin{bmatrix} H_0 \\ H \end{bmatrix} \tag{33}$$

$$K_1 = K_0 + H_1^T H_1 \quad (34)$$

M_1 can be written using (30) and (32):

$$M_1 = M_0 + k_1^{-1} H_1^T (Y_1 - H_1 M_0) \quad \text{Usin} \quad (35)$$

From (34), the value of K for $(k + 1)$th data sample is derives as:

$$K_{k+1}^{-1} = (K_k + H_{k+1}^T H_{k+1})^{-1} \quad (36)$$

Using Woodbury formula (Golub & Loan, 1996), for each incoming data samples the can be updated using the next equation:

$$K_{k+1}^{-1} = K_k^{-1} - K_k^{-1} H_{k+1}^T (I + H_{k+1} K_k^{-1} H_{k+1}^T) H_{k+1} K_k^{-1} \quad (37)$$

Using $V_{k+1} = K_{k+1}^{-1}$, (30) and (31) can be written in a simpler form as:

$$V_{k+1} = V_k - V_k H_{k+1}^T (I + H_{k+1} V_k H_{k+1}^T)^{-1} H_{k+1} V_k \quad (38)$$

$$M_{k+1} = M_k - V_{k+1} H_{k+1}^T (Y_{k+1} - H_{k+1} M_k) \quad (39)$$

Sequential implementation of least square solution the convergence analysis is applied here.

Figure 4 shows the block diagram for the motor control where the control law depends on adaptive controller based on neural networks scheme. The tuning methodology can be developing in two stages off-line and on-line training, Figure 5.

Off-line training: Previously, an off-line training was made to get the initial conditions to the on-line learning, Table 1 presents some experiments of the ANN response, for example: three types of basis functions with different orders were analyzed. As a result we proposed to use monovariable functions of third order, because it presents an adequate performance. Since some values are used in actual units in this system it can be concluded that the interval corresponding to the network input values is -5% ≤ i ≤ 5%; therefore, a good consideration for vector control point is [-1.5 -0.5 0.5 1.5]. Respect to the learning rate, it takes as initial value one point within the interval [0, 2] due to stability purposes. This value is adjusted by trial and error; with a value close to zero the training becomes slow. However, if such value is large, oscillations can occur; in this application it settles down in 0.01, with this value it seeks to have a smooth and robust control as results shown (Brown & Harris, 1995, Tapia-Olvera *et al.*, 2016, Wang & Zou *et al.*, 2014). This training procedure is showed in Figure 5.

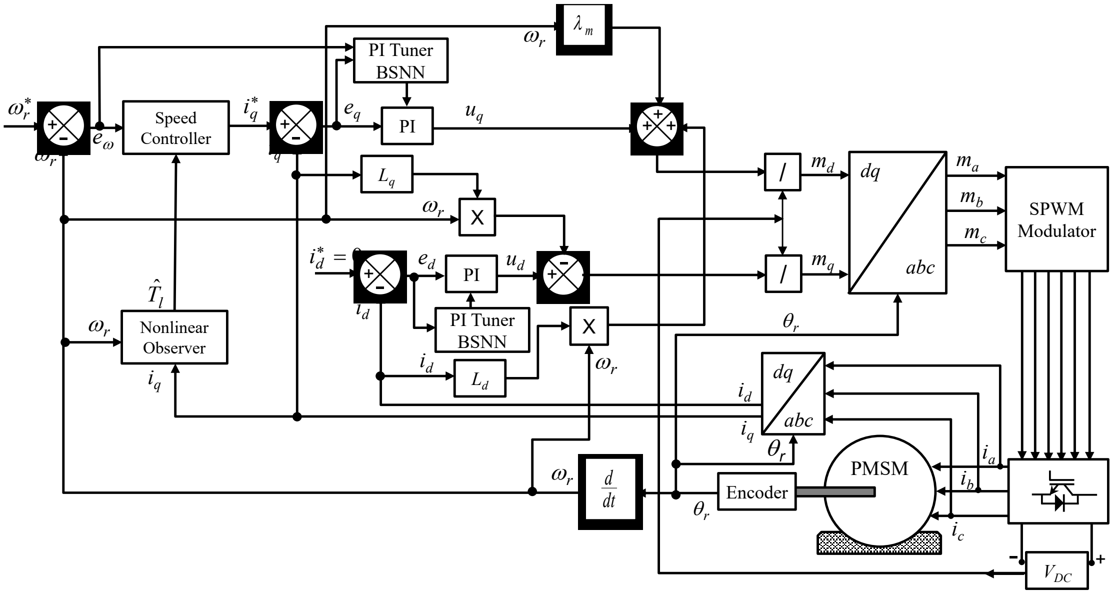

Figure 4. Block diagram of the adaptive controller

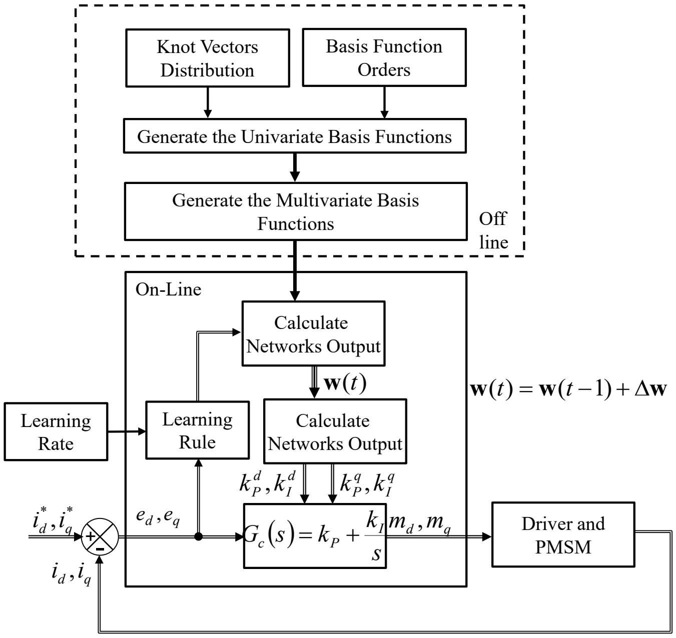
Knot Vectors Distribution
Basis Function Orders
Generate the Univariate Basis Functions
Generate the Multivariate Basis Functions
Off line
On-Line
Calculate Networks Output
$\mathbf{w}(t)$
$\mathbf{w}(t) = \mathbf{w}(t-1) + \Delta\mathbf{w}$
Learning Rate
Learning Rule
Calculate Networks Output
k_P^d, k_I^d
k_P^q, k_I^q
i_d^*, i_q^*
e_d, e_q
$G_c(s) = k_P + \frac{k_I}{s}$
m_d, m_q
Driver and PMSM
i_d, i_q

Figure 5. Training procedure for B-spline neural network

In the proposed scheme, some control parameters have to be specified as the basis functions and the learning rate; but once the BSNN is specified it is adaptive enough to achieve a good performance over a wide range operating conditions. This is because the weights are trained on-line and therefore the output of the ANN adapts to input changes at each time step. BSNN stores and adapts information locally and those weights mainly participating on the output are easily identified.

Table 1. Previous operating conditions to off-line neural network training

Load torque T_l	Rotor speed ω_r
0.95 Nm	30 rpm
3.3 Nm	150 rpm
2.25 Nm	-20 rpm
3.03 Nm	85 rpm
0.32 Nm	33 rpm
1.75 Nm	-74 rpm

Fuzzy logic speed controller for PMSM

A Mamdani scheme with 5 membership functions is employed to design the fuzzy controller (FC). The tracking error *e* and the error change *de* are defined by:

$$e(k) = \omega_r^*(k) - \omega_r(k) \quad (40)$$

$$\frac{de(k)}{dt} = e(k) - e(k-1) \quad (41)$$

The FC in this study uses singleton fuzzifier, triangular and trapezoidal types of membership function, product-inference rule and central average defuzzifier method (Feng, 2010). In Figure 6 some groups are presented: {high negative, low negative, zero, low positive, high positive}. The design procedure of FC algorithm is as follows. Firstly, the linguist values for the membership functions are:

$$N_a = \{(-\infty, A'], \mu_{Na}\}; N_b = \{[B', 0], \mu_{Nb}\}; C = \{[A', A], \mu_c\}; P_b = \{[0, B], \mu_{Pb}\}; P_a = \{[A, \infty), \mu_{Pa}\} \quad (42)$$

Secondly, the computation of the membership degree for *e* and *de* are done. The grade of membership for the fuzzy set, is obtained by:

$$\mu_{Na(\varepsilon_f)} = \max\left[\min\left[\frac{A'-\varepsilon_f}{A'-B'},1\right],0\right] \tag{43}$$

$$\mu_{Nb(\varepsilon_f)} = \max\left[\min\left[\frac{\varepsilon_f-B'}{A'-B'},\frac{0-\varepsilon_f}{\varepsilon_f-A'}\right],0\right] \tag{44}$$

$$\mu_{c(\varepsilon_f)} = \max\left[\min\left[\frac{\varepsilon_f-A'}{0-A'},\frac{A-\varepsilon_f}{A-0}\right],0\right] \tag{45}$$

$$\mu_{Pb(\varepsilon_f)} = \max\left[\min\left[\frac{\varepsilon_f-0}{A-0},\frac{B-\varepsilon_f}{B-A}\right],0\right] \tag{46}$$

$$\mu_{Pa(\varepsilon_f)} = \max\left[\min\left[\frac{B-\varepsilon_f}{B-A},1\right],0\right] \tag{47}$$

In the defuzzification stage, the control action described in linguistic form, is transformed into a precise value to calculate a new value for the control output *S*, which is obtained through the centroid of the membership function and it is represented by:

$$S = \frac{\sum_j \mu B_j(\delta_j)\delta_j}{\sum_j \mu B_j(\delta_j)} \text{ for } j \in \{N_a, N_b, N_c, N_{Pb}, N_{Pa},\} \tag{48}$$

Where $\mu B_j(\delta_j)$ is the degree of membership. Thirdly, the selection of the initial FC rules refers to the dynamic response characteristics. The value of parameters $\{a_i,\ b_i\}$ changes according to the error, and accordingly generates the linguistic value of each membership function.

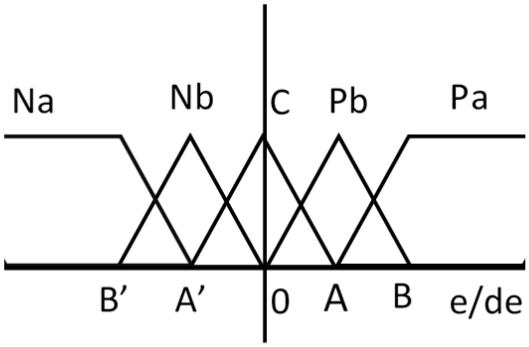

Figure 6. Membership functions for error and its derivative

Figure 7 show the control surfaces for the i_q and i_d current, respectively. The block diagram for regulating the PMSM speed using the fuzzy logic control technique is incorporated in Figure 8, changing only the sliding modes controller.

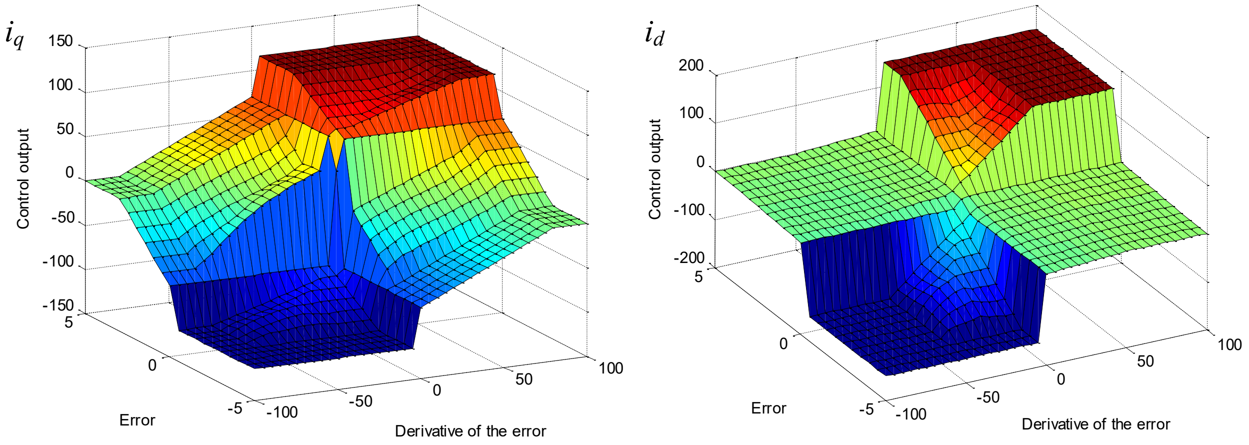

Figure 7. Control surfaces for i_q and i_d current

Design of the sliding mode control

Sliding Mode Current Control: Using Eq. (1), (7), (8) and (11) the switching law by nonlinear sliding mode controller for PMSM is designed (Utkin *et al.*, 2009), being as follows:

$$v_d = v_{d0} sign(e_d) \tag{49}$$

$$v_q = v_{q0} sign\left(e_q\right) \tag{50}$$

Where v_{q0} and v_{d0} are the maximum values of the control signals, respectively. The sliding mode controller block diagram is shown in Figure 8. The condition of existence of sliding mode is equivalent to the stability condition, thus two positive definite Lyapunov functions are selected:

$$V_d = \frac{1}{2} e_d^2 \tag{51}$$

$$V_q = \frac{1}{2} e_q^2 \tag{52}$$

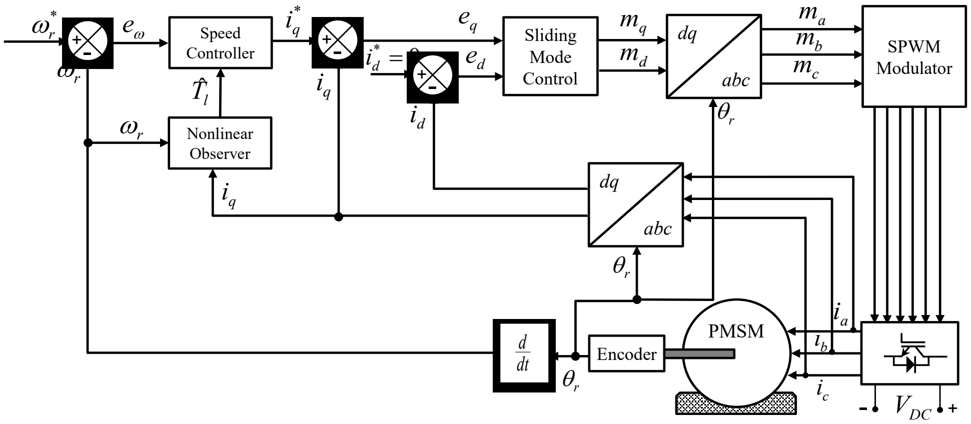

Figure 8. Block diagram for speed control by sliding modes

Existence condition of sliding mode: Differentiating with respect to time on the path of (7-8) and (51-52), respectively, and substituting results:

$$\dot{V}_d = e_d \dot{e}_d = e_d \left(\frac{di_d^*}{dt} + \frac{r_s}{L_d} i_d + \frac{L_q}{L_d} \omega_e i_q \right) - \frac{v_{d0}}{L_d} |e_d| \tag{53}$$

$$\dot{V}_q = e_q \dot{e}_q = e_q \left(\frac{di_q^*}{dt} + \frac{r_s}{L_q} i_q + i_d \omega_e + \frac{\varphi_m}{L_q} \omega_e \right) - \frac{v_{q0}}{L_q} |e_q| \tag{54}$$

Where $|e_i| = e_i sign\ e_i$ for i=d,q. The derivatives of the Lyapunov functions are negative if:

$$\frac{v_{d0}}{L_d} > \left| \frac{di_d^*}{dt} + \frac{r_s}{L_d} i_d + \frac{L_q}{L_d} \omega_e i_q \right| ;\text{for } e_d \tag{55}$$

$$\frac{v_{q0}}{L_q} > \left| \frac{di_q^*}{dt} + \frac{r_s}{L_q} i_q + \omega_e i_d + \frac{\varphi_m}{L_q} \omega_e \right| ;\text{for } e_q \tag{56}$$

Therefore, e_d, e_q and e_ω are zero after a finite time interval.

Results and discussion

In this section the simulation results were performed to verify and compare the performance of different control schemes for rotor speed regulation. These results are used to examine the dynamic response of adaptive control (PI NN) compared with three different control schemes: PI controller (PI FIX), sliding mode controller (SM) and fuzzy controller (Fuzzy). Different operation conditions of the motor, Figure 1, are taking into account. The implementation of the four control laws is performed in Matlab/Simulink, Figure 1. The PMSM parameters are assumed known, Table 2.

A comparative performance in two different cases, considering its dynamic response is analyzed. In all cases the same algorithm to estimate the load torque is used. The convergence rate of the estimator is determined by the choice of system eigenvalues expressed by (12)-(13). The proposed values are $\lambda_{1,2}$=-440, the estimator gains are: l_1=880 and l_2=-6.776; and the speed coefficient of desired dynamics convergence for the error in (10) is c=500.

Table 2. Parameter of dynamic model of the permanent magnet synchronous motor

Parameter	Symbol	Value and units
Inertia coefficient	J	3.5e-5 N.m
Nominal voltage	v	120 volts
Stator resistance	r_s	2.6 Ohms
Stator inductance in d-axis	L_d	6.73 mH
Stator inductance in q-axis	L_q	6.73 mH
Stator leakage inductance	L_{ls}	0.1 L_q
Magnetic flux from the magnet	φ_m	0.319 Wb
Pole pairs	n	2

Case 1: In this case the simulation of the control scheme is done to know their performance to external disturbances, speed requirements and load torque, Table 3. Figures. 9-10 show the dynamic response of the rotor speed and stator currents for the PMSM control scheme. These results present the comparison of the adaptive PI control (PI NN), sliding mode controller (SM), Fuzzy controller and conventional control (PI FIX).

Table 3. Operating conditions, *case 1*

Load torque T_l	Time	Rotor speed ω_r
0.0 Nm	if $t < 0.15$ sec	50sin(2πt) rpm if t ≤ 1.25 sec
1.0 Nm	if $t > 0.15$ sec & $t \leq 0.37$ sec	
0.5 Nm	if $t > 0.37$ sec & $t \leq 0.60$ sec	
2.0 Nm	if $t > 0.60$ sec & $t \leq 0.80$ sec	
1.0 Nm	if $t > 0.80$ sec & $t \leq 1.00$ sec	50cos(4πt) rpm if t > 1.25 sec
3.5 Nm	if $t > 1.00$ sec & $t \leq 1.30$ sec	
0.0 Nm	if $t > 1.30$ sec & $t \leq 1.60$ sec	
2.0 Nm	if $t > 1.60$ sec	

In Figure 9 different overshoot caused by external disturbance in load torque are observed, first the motor is subject to a load increase from 0 Nm to 1 Nm, Table 3. We can see that the neural network helps decrease the magnitude of the overshoot appearing in the transient speed response, achievement the steady state condition more smoothly and sometimes faster than the response of the system, that only has a conventional PI controller, sliding mode or fuzzy scheme, Figure 9(a). For example, the overshoot is less than 3 rad/sec.; on the other hand, with fixed parameters, it is about 8 rad/sec.; the settling time is 7 msec with respect to 10 msec with PIFIX, in t=0.8sec.

To appreciate in greater detail the change in motor response by different controllers a zoom section of the variable is added when the disturbance is presented. The line in red exhibits the neural network impact in motor speed control. The error evolution reveals how the gains updated in controller PI by BSNN, Figure 10(b). The performance evaluation of each controller is determined according to: ISE, IAE and ITAE. Table 4 shows the performance parameters of the MSIP for case 1.

Table 4. Controller performance indices, *case 1*

Controller	ISE	IAE	ITAE
PIFIX	4.7701	0.9660	0.9303
PINN	2.5842	0.7127	0.7314
FUZZY	6.1231	1.2717	1.4392
SM	8.9817	1.7872	1.9865

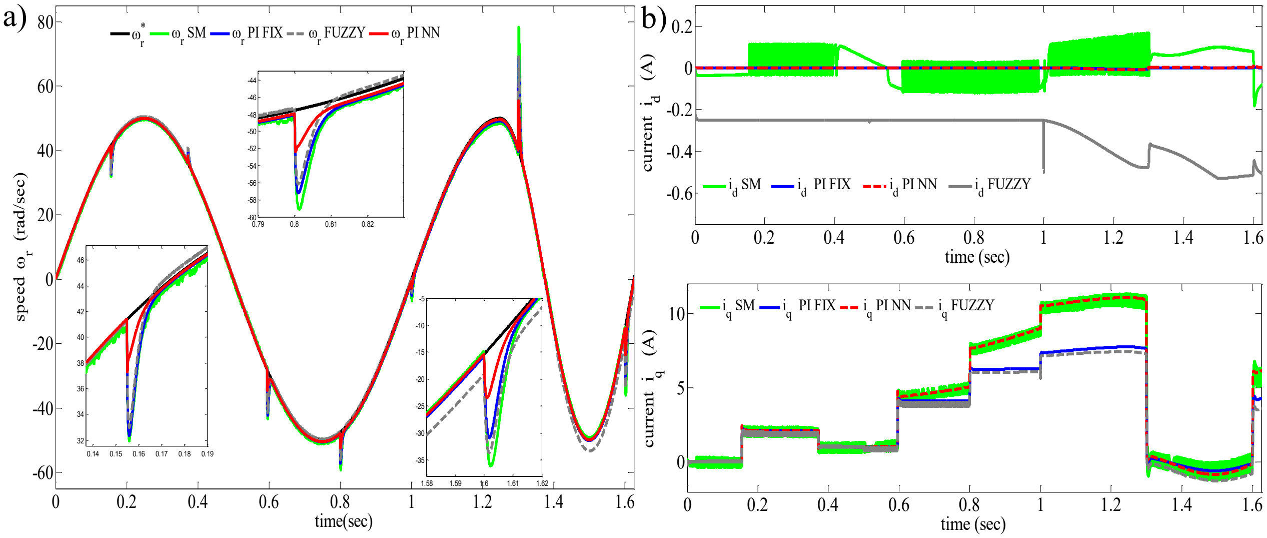

Figure 9. Dynamic performance PMSM for the first case study: a) Closed-loop speed reference trajectory tracking and b) Dynamic performance of the currents in *dq* frame

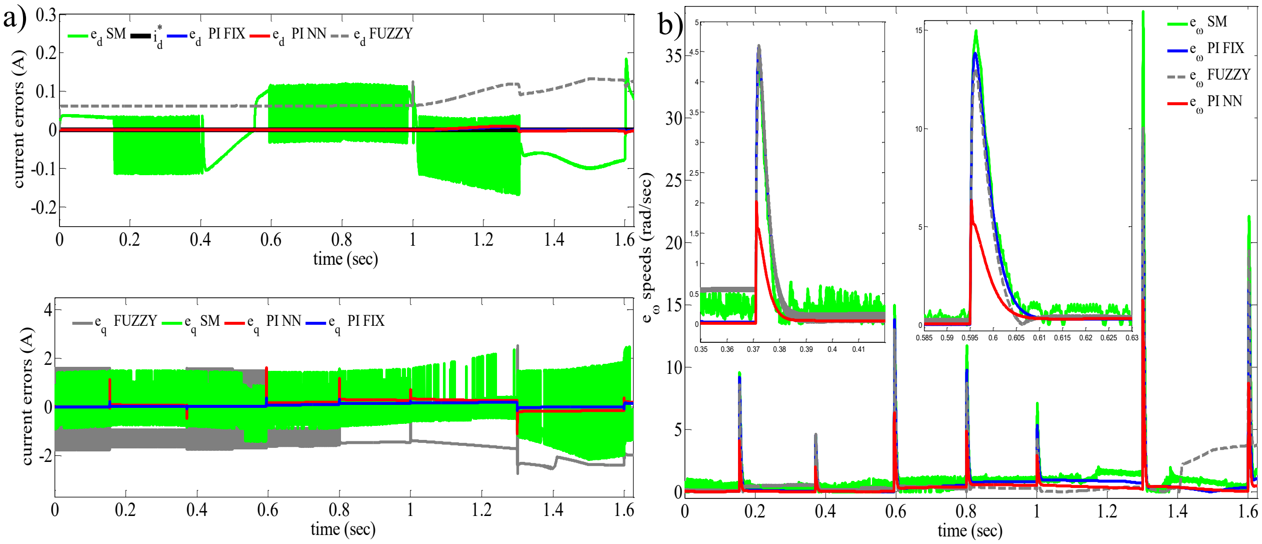

Figure 10. Tracking errors for reference profile under load torque disturbance for the case 1: a) dq axis current tracking error, b) Speed tracking error

Regarding the dynamic performance of the stator currents (i_q, i_d), Figure 9(b) shows that the evolution is suitable for all control strategies. Figure 9(b) exhibit that $i_d = 0$ except in the fuzzy controller and sliding modes techniques. The response i_d obtained by sliding mode controller varies within a range between 0.25 and -0.2 Amp.

The updates gains k_i and k_p of PI controllers provide by the neural networks have the ability to adapt to the new desired operating condition of the PMSM, Figure 11.

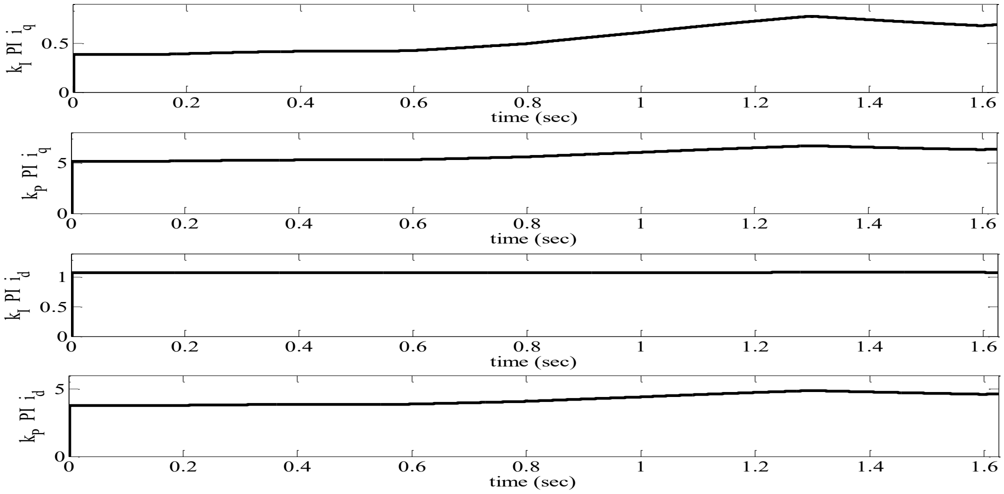

Figure 11. k_i and k_p gains evolution of the PI controllers attained by the neuro-controller, *case 1*

Case 2: In this case the requirements depends of the desired speed (ω^*) under changes in load torque, T_l, Table 5.

Table 5. Operating conditions, *case 2*

Load torque T_l (Nm)	**Rotor speed** ω_r (rpm)
0 Nm if $t < 0.10$ sec	0 rpm if $t < 0.05$ sec
0.8 Nm if $t > 0.10$ & $t \leq 0.19$ sec	150 rpm if $t > 0.05$ & $t \leq 0.15$ sec
1.5 Nm if $t > 0.19$ sec	220 rpm if $t > 0.15$ & $t \leq 0.25$ sec
	60 rpm if $t > 0.25$ sec

Figures 12-14 present satisfactory performance for all control schemes. However, in Figure 12 it's clear that the performance of three controllers is poorer than BSNN. The steady state condition is improved when BSNN is included, Figure 12(a). The picture 12(a) depicts the rotor speed performance when the system is in face with different disturbances, Table 4, which exhibits positive interaction between the BSNN and the conventional PI controller.

In Figure 14 the evolution of the k_i and k_p gains of the two PI controllers updating by the B-spline neural network is presented. For the controller gains of i_q current at the first samples their magnitude radically changes respect to the initial condition, reaching a value close to 1.5 and 0.71 Amp., respectively, (Figure 12(b)). Subsequently, the neural network is still learning about the motor operation, the values are updating to reach other steady state. The clearest changes are presented in the transient period, which helps that the dynamic response has better characteristics. The on-line learning rule operates faster when the parameters of interest are far from the optimal value, Figures 12-13.

The results illustrate the impact of various design and nonlinear control techniques in the PMSM performance, the study was considered high operating requirements. The performance of control techniques is in accordance with case 1 and 2. The BSNN exhibits better performance adapting itself easily to the new circumstances. This helps to maintain the operation within safer regions, preventing from an instability scenario. Table 6 shows the performance parameters of the MSIP for case 2.

Table 6. Controller performance indices, case 2

Controller	**ISE**	**IAE**	**ITAE**
PIFIX	5.8197	0.1122	0.0171
PINN	3.1278	0.0985	0.0150
FUZZY	30.3829	1.5991	1.1748
SM	41.3698	1.8149	1.2014

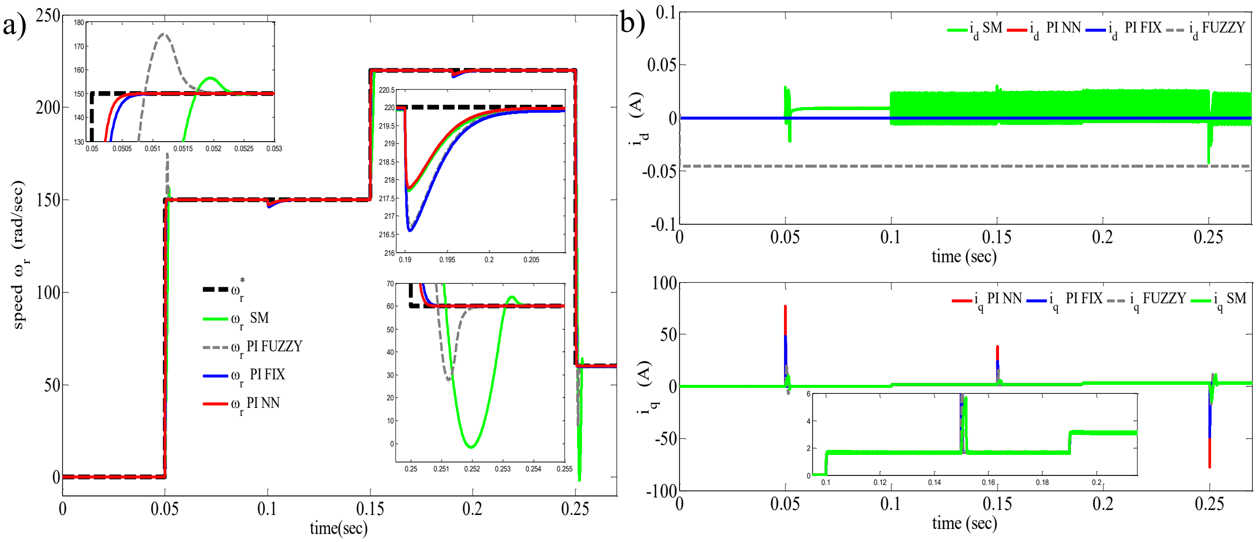

Figure 12. Dynamic performance PMSM for the second case study: a) Closed-loop speed reference trajectory tracking and b) Dynamic performance of the currents in *dq* frame

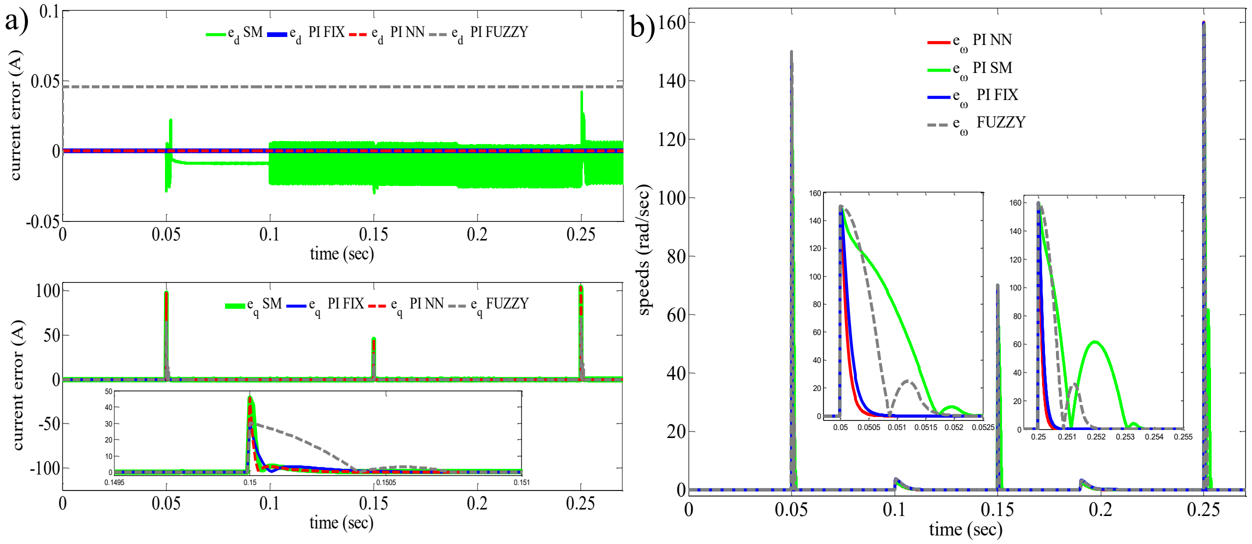

Figure 13. Tracking errors for reference profile under load torque disturbance for the second case: a) *dq* axis current tracking error, b) Speed tracking error

Table 7 presents the initial conditions for k_p and k_i of each current controller for both cases 1 and 2, these were obtained by off-line training. The final values that were reached after the on-line training have also been shown, the upper and lower limits refine the search space in the tuning process for PI controllers.

Table 7. Initial conditions and evolution of the gains of pi controllers

	Signal	Initial		PI + NN					
		k_P	k_I	k_P			k_I		
				max	min	last	max	min	last
Case 1	i_d	3.766	1.078	4.885	3.766	4.651	1.078	1.078	1.082
	i_q	5.152	0.386	6.684	5.152	6.350	0.781	0.386	0.695
Case 2	i_d	1.148	1.576	1.172	0.163	1.131	1.576	1.575	1.576
	i_q	0.784	1.615	0.801	0.111	0.772	1.664	0.224	1.594

The two cases validate the appropriate system evolution under severe and sudden changes in load torque and rotor speed, respectively. The proposed tuning technique based on BSNN has better performance with respect to fixed parameters and other controllers' performance presented in this section, considering the transient and steady-state conditions.

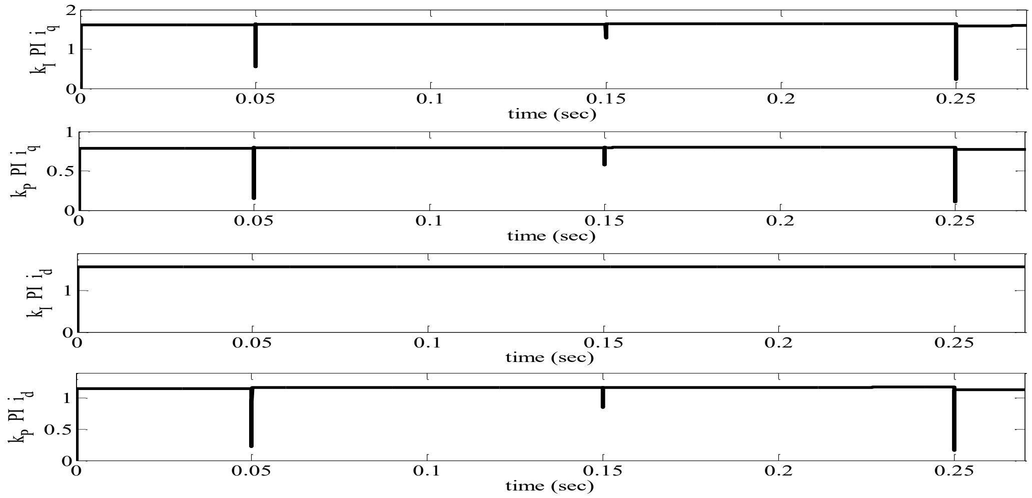

Figure 14. K_P and K_I gains evolution of the PI controllers attained by the neuro-controller, *case 2*

Conclusion

The results exhibit that all controllers are designed and proved for tracking the rotor speed reference. Three control strategies (PI, sliding mode and fuzzy) are compared vs the adaptive B-spline neural network algorithm, it is concluded that while the proposed neural controller can adapt itself to different operating conditions, the conventional controller's performance tends to be poorer under such circumstances, especially for changes in system conditions for which their parameters were tuned. We can conclude that the adaptive neural controller with on-line learning has a good interaction with conventional control techniques achieved superior performance for permanent magnet synchronous motor control. The proposed adaptive control has outperformed the rest of the controllers in analysis.

References

Accetta A., Cirrincione M. and Pucci M., (2012), TLS EXIN based Neural Sensorless Control of a High Dynamic PMSM, Control Engineering Practice, (20): 725-732.

Aguilar-Mejía O., Tapia-Olvera R., Valderrabano-González A. & Rivas I., (2016), Adaptive neural network control of chaos in permanent magnet synchronous motor, Intelligent Automation & Soft Computing, (22), 499-507.

Alonge F., D'Ippolito F., Giardina G. and Scaffidi T., (2007). "Design and low-cost implementation of an optimally robust reduced-order rotor flux observer for induction motor control," IEEE Trans. on Industrial Electronics, (54): 3205–3216.

Beltran-Carbajal F., Tapia-Olvera R., Lopez-Garcia I., and Guillen D., (2018), Adaptive dynamical tracking control under uncertainty of shunt DC motors, Electric Power Systems Research, (164), 70-78.

Brown M., Harris C., (1995). Neurofuzzy Adaptive Modeling and Control, Prentice Hall.

Chang Y., Chen C., Zhu Z. and Huang Y., (2016), "Speed Control of the Surface-Mounted Permanent-Magnet Synchronous Motor Based on Takagi–Sugeno Fuzzy Models," IEEE Transactions on Power Electronics, (31), 6504-6510.

Chih-Hong L. and Chih-Peng L., (2013). The hybrid RFNN control for a PMSM drive electric scooter using rotor flux estimator, International Journal of Electrical Power & Energy Systems, (51): 213–223.

Choi H. H., Jin-Woo J., (2011). Takagi–Sugeno fuzzy speed controller design for a permanent magnet synchronous motor, Mechatronics, (21): 1317–1328.

Choi H., Yun H. and Kim Y., (2015), "Implementation of Evolutionary Fuzzy PID Speed Controller for PM Synchronous Motor," IEEE Transactions on Industrial Informaics, (11), 540-547.

Englert T. and Graichen K., (2018), Nonlinear model predictive torque control of PMSMs for high performance applications, Control Engineering Practice, (81), 43–54.

Feng G., (2010). Analysis and Synthesis of Fuzzy Control Systems: A Model-Based Approach, CRC Press.

Golub, G. H., & Loan, C. F. V. (1996). Matrix computations (3rd ed.). Baltimore, MD: The Johns Hopkins Univ. Press.

Hamida M. A., Glumineau A., de Leon J., Loron L., (2014). Robust adaptive high order sliding-mode optimum controller for sensorless interior permanent magnet synchronous motors, Mathematics and Computers in Simulation, (105): 79–104.

Han Ho C., Hong Min Y., Yong K., (2013). Implementation of Evolutionary Fuzzy PID Speed Controller for PM Synchronous Motor, IEEE Trans. on Industrial Informatics, (11): 540-547.

Heng D., Dipti S., Ramesh O., (2010). A B-spline network based neural controller for power electronic applications, Neurocomputing, (73): 593–601.

Hu J., Qiu Y., Lu H., (2016), Adaptive robust nonlinear feedback control of chaos in PMSM system with modeling uncertainty, Applied Mathematical Modelling, (40), 8265-8275.

Iqbal A., Abu-Rub H., Nounou H., (2014). Adaptive fuzzy logic-controlled surface mount permanent magnet synchronous motor drive, Systems Science & Control Engineering, (2): 465–475.

Jin-Woo J., Quoc V., Duc T., Eun-Kyung K., Ho H., (2014). Adaptive PID Speed Control Design for Permanent Magnet Synchronous Motor Drives, IEEE Trans. on Power Electronics, (30): 900-908.

Jon R., Wang Z., Luo C., Jong M., (2017), Adaptive robust speed control based on recurrent elman neural network for sensorless PMSM servo drives, Neurocomputing, (227), 131-141.

Jun-Jien R., Yan-Cheng L., Ning W., Si-Yuan L., (2015). Sensorless control of ship propulsion interior permanent magnet synchronous motor based on a new sliding mode observer, ISA Transactions, (54): 15–26.

Khanchoul M., Hilairet M., Normand-Cyrot D., (2014). A passivity-based controller under low sampling for speed control of PMSM, Control Engineering Practice, (26): 20–27.

Krishnan R., (2010). Permanent Magnet Synchronous and Brushless DC Motor Drives, CRC Press.

Kumar V., Gaur P., Mittal A. P., (2014). ANN based self-tuned PID like adaptive controller design for high performance PMSM position control, Expert Systems with Applications", (41): 7995–8002.

Liu X., Yu H., Yu J. and Zhao L., (2018) Combined Speed and Current Terminal Sliding Mode Control With Nonlinear Disturbance Observer for PMSM Drive, IEEE Access, (6), 29594-29601.

Lu E., Li W., Yang X., Xu S., (2017), Composite Sliding Mode Control of a Permanent Magnet Direct-Driven System for a Mining Scraper Conveyor, IEEE Access, (5), 22399 – 22408.

Marufuzzaman M., Ibne M. B., Farzana L., Gyu T T., (2014). A high speed current dq PI controller for PMSM drive, Tehnički vjesnik – Technical Gazette, (21): 467-470.

Moreno J. A., Osorio M., (2008). Lyapunov Approach to Second-Order Sliding Mode Controllers and Observers, *In Proc. 47th IEEE Conf. on Decision and Control*, 2856–2861.

Nizami T., Chakravarty A. and Mahanta C., (2017), Design and implementation of a neuro-adaptive backstepping controller for buck converter fed PMDC-motor, Control Engineering Practice, (58), 78-87.

Omrane I., Etien E., Dib W., Bachelier O., (2015). "Modeling and Simulation of Soft Sensor Design for Real-Time Speed and Position Estimation of PMSM," ISA Transactions, (57): 329–339.

Ortega R., Monshizadeh N., Monshizadeh P., Bazylev D., and Pyrkin A., (2018), Permanent magnet synchronous motors are globally asymptotically stabilizable with PI current control, Automatica, (98), 296-301.

Öztürk N., and Çelik E., (2012). Speed control of permanent magnet synchronous motors using fuzzy controller based on genetic algorithm, International Journal of Electrical Power & Energy Systems, (43), 889–898.

Qutubuddin M. D. and Yadaiah N., (2018), A new intelligent adaptive mechanism for sensorless control of permanent magnet synchronous motor drive, Biologically Inspired Cognitive Architectures, (24), 47-58.

Serre, D. (2002). Matrices: Theory and applications. New York: Springer.

Tapia-Olvera R., Beltran-Carbajal F., Aguilar-Mejia, O. and Valderrabano-Gonzalez A., (2016), An Adaptive Speed Control Approach for DC Shunt Motors, Energies, (9), 961.

Tapia-Olvera Ruben., Aguilar-Mejía Omar., Minor-Popocatl H., Santiago C., (2012). Power System Stabilizer and Secondary Voltage Regulator Tuning for Multimachine Power Systems, Electric Power Components and Systems, (40): 1751-1767.

Thi-Thuy N. V., Choi H. H., Jung J. W., (2012). “Certainty equivalence adaptive speed controller for permanent magnet synchronous motor,” Mechatronics, (22): 811–818.

Utkin V., Guldner J., Shi J., (2009). Sliding Mode Control in Electro-Mechanical Systems, CRC Press.

Verrelli C. M., Tomei P., Lorenzani E., Migliazza G., Immovilli F. (2017), Nonlinear tracking control for sensorless permanent magnet synchronous motors with uncertainties, Control Engineering Practice, (60), 157-170.

Wang H., Zou Q., (2014). “B-Spline-Decomposition-Based Approach to Multiaxis Trajectory Tracking: Nanomanipulation Example,” IEEE Trans. on Control Systems Technology, (22): 1573-1580.

Yazdani A. and Iravani R., (2010). Voltage-Sourced Converters in Power Systems Modeling, Control, and Applications, Wiley.

Zhang Y., Song J., Song S., Yan M., (2010). Adaptive PID Speed Controller Based on RBF for Permanent Magnet Synchronous Motor System,. *in Proc. 2010 IEEE International Conf. Intelligent Computation Technology and Automation*, Changsha, 425-428.

Zhengyu L., Reay D. S., Williams B. W., Xiangning H., (2006). Torque Ripple Reduction in Switched Reluctance Motor Drives Using B-Spline Neural Networks, IEEE Trans. on Industry Applications, (42), 1445-1453.