Comparative Investigation of DSP-Based Speed Control of PMSM Using Proportional Integral and Takagi-Sugeno Fuzzy Logic Controller

Rachid Askour*, Hatim Jbari, and Badr Bououlid Idrissi
National High School of Arts and Crafts, Moulay Ismail University, Meknes, Morocco
Email: r.askour@umi.ac.ma (R.A.), ha.jbari@edu.umi.ac.ma (H.J.), b.bououlid@umi.ac.ma (B.I.)

Abstract—**This paper presents the simulation and experimental results of speed control of a three-phase Permanent Magnet Synchronous Motor (PMSM). The control techniques employed in this work include a Proportional Integral (PI) controller and a Takagi-Sugeno Fuzzy Logic Controller (TS-FLC). While theory and simulation help in predicting the PI controller settings for real drive implementation, further adjustment of the PI controller coefficients is still needed for optimal performance. However, the complexity of PI controller tuning can be overcome by employing a fuzzy logic controller, which is less affected by variations in model parameters and load torque. Two variants of the proposed TS-FLC are implemented: a simple algorithm with 9 rules and a standard algorithm with 49 rules. The drive system utilizes Field Oriented Control (FOC). The control program is developed using the C/C++ programming language and executed on the Texas Instruments TMS320LF28335 Digital Signal Processor (DSP), known for its intelligent motor control capabilities. The drive system evaluation was performed through simulations and experimentation using the MCK28335 professional development kit provided by Technosoft Company. The simulation and experimental results of the proposed controllers are compared under various conditions, including speed reversal and load torque variations.**

Index Terms—**Digital Signal Processor (DSP) real implementation, field oriented speed control, fuzzy logic controller, Permanent Magnet Synchronous Motor (PMSM)**

I. Introduction

The Permanent Magnet Synchronous Machine (PMSM) is widely used in various industrial applications, including robotics, electric vehicles, wind turbines, and renewable energy generation systems. To optimize the performance of these machines, it is essential to implement robust control and understand their behavior through the establishment of more precise models. The objective of robust control of a PMSM is to ensure stable and accurate functioning, combined with high resistance to external disturbances and load variations. It aims to ensure that the machine can reliably follow speed, torque, or position setpoints, keeping energy efficiency high and minimizing undesirable vibrations. One of the main challenges in controlling a PMSM is its complex and non-linear behavior. The presence of permanent magnets introduces non-linearities into the system, requiring specific control approaches to fully exploit their potential. In addition, PMSMs are often affected by variable operating conditions and unexpected disturbances, which makes the development of robust control strategies more crucial.

Robust PMSM speed control relies on advanced techniques such as predictive control, sliding mode control, adaptive control, computational intelligence and reinforcement learning, etc. [1, 2]. These approaches provide compensation for uncertainties and parametric variations, reduce the effects of external disturbances, and improve the stability of the system in comparison with classical control techniques. According to the literature, various techniques for achieving more robust control in PMSMs have been proposed by researchers using different approaches. These approaches include the use of fuzzy controller [3–5] and Finite Control Set-Model Predictive Controller (FCS-MPC) [6]. Another more promising approach to achieve robust control of synchronous machines is to integrate Neural Networks with other control techniques such as PI and fuzzy controllers [7]. Some researchers have also combined two types of controllers, namely fuzzy logic combined with Sliding Mode Variable Structure Control (SMVSC) [8], fuzzy logic and the Proportional Integral Derivative components (PID) like Proportional-Integral (PI) controller [9, 10] and PID controller [11]. An alternative method enables the development of an optimized fuzzy Proportional-Integral (PI) controller for Brushless DC Motors (BLDCMs) through the utilization of the Particle Swarm Optimization (PSO)–Gravitational Search Algorithm (GSA) [12]. These approaches aim to enhance control robustness and improve the dynamic performance compared to a conventional controller under various load conditions, by providing faster response speed, shorter stabilization time with less overshoot, and less error in the steady state.

A study in [13] presents a fuzzy PI controller designed for the speed control of a PMSM. The system comprises a Speed Loop Controller (SLC) and a Current Loop Controller (CLC). The SLC incorporates a fuzzy PI controller and a standard model. An algorithm is developed using VHDL and integrated into

Manuscript received August 8, 2023; revised October 24, 2023; accepted November 15, 2023.

*Corresponding author

doi: 10.18178/ijeetc.13.2.135-147

MATLAB/Simulink for simulation. Controller performance is evaluated under varying load conditions, with adjustments based on PMSM parameters. Both simulation and experimental results confirm the PMSM's ability to maintain stable speed even with changing loads. However, the study compares PI and fuzzy PI controllers based on simulation and experimental results without quantifying their performance. Another work in [14] proposes the design of an adaptive Fuzzy Logic Controller (FLC) to improve the PMSM dynamics in steady state and in the flux-weakening region. The performance of the proposed control scheme has been validated using experimental results based on the dSPACE (DS1006) platform showing enhanced performance in torque, d/q-axis currents, and DC-link voltage in comparison with conventional PI controller.

The objective of the study in [15] was to improve the performance of a Surface Mounted Permanent Magnet Synchronous Motor (SPMSM) by implementing a modified Direct Torque Control system with a Fuzzy Logic Controller. This algorithm includes a torque hysteresis controller and stator voltage sectors, eliminating the requirement for a flux controller. The study concentrated on evaluating the motor's performance using this approach in contrast to a traditional PI Controller, and it was validated under various speeds and load disturbances. The simulation results clearly demonstrated the superior performance of the SPMSM control system with the fuzzy speed loop controller compared to the PI controller. Reference [16] presents a PMSM control system for electric vehicles based on two distinct controller variants: the fuzzy PI controller and the sliding mode controller. Simulation results reveal that the fuzzy PI controller exhibits a quicker dynamic response and superior dynamic performance compared to the sliding mode controller.

Another work in [17] proposes a novel hybrid algorithm called H-GA-GSA, which combines the strengths of the Genetic Algorithm (GA) and Gravitational Search Algorithm (GSA) for optimizing the FLC for PMSM speed control. The study outlines a practical method for the adjustment and design of FLC optimization using heuristic algorithms. To assess the effectiveness of H-GA-GSA, the proposed hybrid algorithm is compared to GA and GSA in terms of convergence rate, PMSM speed control performance, and electromagnetic torque variations. The optimization performance and results obtained from simulation studies confirm that the proposed hybrid H-GA-GSA surpasses GA and GSA in terms of performance._The authors in [18] present a control system of a PMSM based on a PI Fuzzy controller to enhance the machine dynamics in a steady state. the proposed control system integrates the neural network of Extreme Learning Machine (ELM) with optimized parameters based on the Artificial Bee Colony (ABC) algorithm. The ELM is used to define the optimal parameters of the PI Fuzzy controller. The comparison with classical and heuristic controllers shows that the proposed control scheme ensures high dynamic performances, especially the accuracy and the response speed of the machine.

Reference [19] present a servo control system used as a servomotor position controller, using a fractional order Proportional-Integral-Derivative (PID) controller with parameter autotuning by a fuzzy logic algorithm. The proposed system combines the accuracy of the PID with the advanced adaptive capacity of the fuzzy controller. Simulation and experimentation results show high robustness when compared with a classical PID-based controller. In [20], the proposed control scheme is based on the PSO mechanism to determine the optimal values of a fuzzy-PI controller. The power drive includes a conventional Voltage Source Inverter (VSI) supplemented with a Current Source Inverter (CSI) supplied by a buck converter. The proposed system integrates a PMSM overcurrent protection circuit and ensures a low Total Harmonic Distortion (THD). In the same context, another approach introduced in [21] aims to enhance the flexibility of the Takagi-Sugeno Fuzzy Logic PID (TSFL-PID) controller by optimizing its fuzzy rules for speed control of PMSM, in contrast to the conventional TSFL-PID controller. In this method, the TSFL-PID control parameters are tuned using the PSO method, and for comparison, the TSFL-PID controller is also optimized using Genetic Algorithms (GA). The simulation results obtained from this study demonstrate that the PSO algorithm outperforms the GA in terms of controller performance. However, it's worth noting that these algorithms have not undergone experimental testing.

In [22], the authors focus on designing and implementing speed control for a Surface-Mounted Permanent-Magnet Synchronous Motor (SPMSM) using Takagi-Sugeno (T-S) fuzzy models. They introduce a speed controller based on the T-S fuzzy model, which improves tracking and load regulation. The study outlines a speed control strategy that incorporates defuzzification and parallel distributed compensation. Additionally, the research confirms the system's ability to reduce disturbances and maintain Lyapunov stability. Controller gains are determined with the Linear Matrix Inequality (LMI) Toolbox, and all speed control processes are executed via the Renesas RX62T microcontroller. Experimental results validate the system's tracking performance and load regulation capability.

On the other hand, PI controllers are widely used in PMSM speed control due to their simplicity, low cost of implementation and effectiveness in achieving stable and precise speed control. However, the performance of the PI speed controller in a wide range of motor speeds is highly dependent on the values of these gains $K_p$and K_i, necessitating careful tuning for optimal results. Higher PI gains are necessary for motor acceleration and fast rejection of load disturbances, but this can lead to undesirable overshoot in motor speed. Therefore, there is a growing interest in replacing these controllers with more sophisticated control strategies. Indeed, the state of the art in PMSM speed control is characterized by the incorporation of intelligent algorithms, which can range from simpler to more complex solutions, offering adaptability, efficiency, and precision in control. Recent developments have placed a significant emphasis on advanced control methodologies, including Fuzzy Logic Controllers (FLCs) and Takagi-Sugeno FLCs, to enhance speed control capabilities. However, it's important to note that these advanced approaches often come with substantial computational requirements, raising concerns regarding practical implementation and potential hardware costs. Additionally, research has been dedicated to the simplification and optimization of rule-bases within FLCs [23]. This approach aims to facilitate the design and implementation of complex control systems while minimizing the computational burden.

In this paper, the speed control of a PMSM is examined using two variants of T-S FLCs: one with a standard algorithm featuring 49 rules and another with a simpler design comprising just 9 rules. The novelty of this research lies in comparing the dynamic performances of these controllers and a classical PI controller under varying speed and load torque conditions through both digital simulation and experimental implementation. The objective is to determine whether the simplified rules can maintain the drive's performance and exhibit behavior like the standard rule-based design.

The structure of the paper is organized as follows: Section II presents an overview of the mathematical model and the FOC technique applied to PMSM. Section III discusses the design of speed controllers, including the PI controller and the TS-FLCs. Section IV presents an evaluation of the simulation results. The experimental setup and the implementation of the DSP control algorithm are described in Section V. Section VI focuses on the experimental validation of the proposed controllers. Finally, the conclusion is presented in the last section.

II. PMSM Control Structure

A. PMSM Mathematical Model

To simplify the modelling of the PMSM, the usual assumptions given in most references are adopted as follows:

- The rotor's damping effect is neglected.
- The machine's magnetic circuit is not saturated.
- The induced EMF is sinusoidal.
- The capacitive couplings between the windings are neglected.
- Eddy currents and hysteresis losses are omitted.

The mathematical model of the PMSM in the dq-rotating coordinate system, utilizing Clark and Park transformations, is employed for analysis under the given conditions.

In the rotating Park frame, the equation for the stator voltage is as follows (1) and (2):

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q \tag{1}$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \varphi_m) \tag{2}$$

where (v_d, v_q) and (i_d, i_q) are respectively the stator voltages and the stator currents in the dq-axis, R_s is the stator resistance, (L_d, L_q) are stator inductances in the dq-axis, ω_e is the rotor electrical angular velocity and φ_m is the permanent magnet flux linkage.

The electromagnetic torque is described in (3) as follows:

$$T_{\text{em}} = \frac{3}{2} p \left[\varphi_m i_q - (L_q - L_d) i_q i_d \right] \tag{3}$$

In the case of a smooth pole synchronous motor, the expression of T_{em} becomes:

$$T_{\text{em}} = \frac{3}{2} p \varphi_m i_q \tag{4}$$

where T_{em} is the electromagnetic torque and p is the number of pole pairs.

The motor's motion equation is represented by the following expression (5):

$$J \frac{d\omega_m}{dt} = T_{em} - T_L - B\omega_m \tag{5}$$

where J is the motor moment of inertia, ω_m is the mechanical speed, T_L is the load torque and B is the friction coefficient.

B. Field Oriented Control of PMSM

The FOC scheme for a three phase PMSM with TS-FLC is depicted in Fig. 1. It is mainly based on the measurement of the stator currents (i_a and i_b) and the rotor position of the machine. The "Calculation of position and speed" block calculates the electrical angular position of the rotor θ_e, and the speed ω of the machine by determining the difference in rotor position over one speed sampling period. The measured stator currents (i_a and i_b) are transformed into the fixed Clark frame (i_α and i_β), and then into the rotating Park frame (i_d and i_q) based on the machine's position θ_e. Indeed, this strategy realizes the decoupling between the components (i_d and i_q) of the stator current which respectively control the flux and the torque, thus making the control of the PMSM like that of a DC machine.

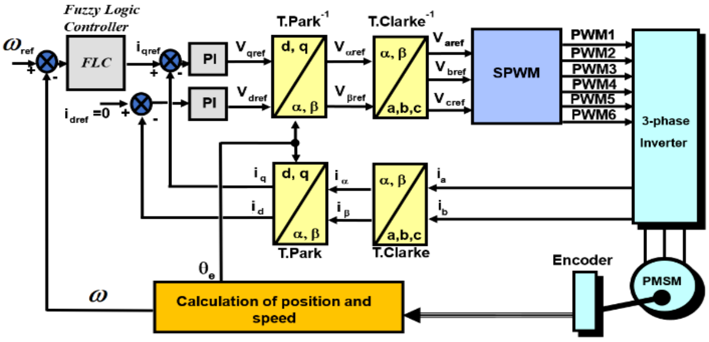


Fig. 1. Field oriented control scheme for a three-phase PMSM with TS-FLC.

In this work, the speed controller is achieved using either a PI controller or a Fuzzy Logic Controller. The stator current control is implemented using discrete PI controllers. By employing inverse Park and Clarke transformations, the reference voltages ($V_{a,\text{ref}}$, $V_{b,\text{ref}}$ and $V_{c,\text{ref}}$) needed for generating the PWM signals are calculated from the outputs of the current controllers. These reference voltages are then utilized by the sinusoidal PWM signal generation block to produce control signals for the three-phase inverter.

The FOC involves independent control of the current components, i_q and i_d. The value of i_q is controlled based on the desired speed, while the value of i_d is set to achieve the desired flux. For permanent magnet motors, the reference current i_d is typically maintained at zero. The effectiveness of FOC implementation depends not only on the accurate computation of the electrical angular position, θ_e , but also on the proper tuning of the parameters for the speed and current controllers.

III. Design of Speed Controllers

A. PI controller with Basic Anti-Windup

A Proportional Integral (PI) controller is suitable for adjusting the reference torque value, i_q^{ref} . The PI controller's variable parameters include K_i, K_p, Sampling Time $T_{s_{\text{spd}}}$, Output Saturation Sat_{spd}, and Integral Limit $I_{\max}$. The behavior of the PI controller is determined by the gains K_p and K_i, which control the sensitivity to speed error and the steady-state speed error. In the proposed implementation, the controller operates in a discrete form.

In the context of a PI controller, windup represents an extensively researched issue [24, 25]. This problem typically arises when the input error to the controller is significant, or when the input error remains non-zero for an extended period. Controller output saturation can occur due to either a substantial input error or persistent non-zero error, causing the integrator output to continually accumulate. The integrator's output remains significantly elevated, leading to a pronounced overshoot of the controlled signal beyond the commanded value while the integrator's output is being reset. Throughout this period, the trajectory of the controlled signal is effectively unregulated and could potentially lead to system damage if not constrained. This phenomenon is akin to tightly winding a spring and then abruptly releasing it. This is the reason why this phenomenon is referred to as "windup" in PI controllers.

Fig. 2 illustrates the structure of the basic anti-windup PI controller with an integral limiter used in this study.

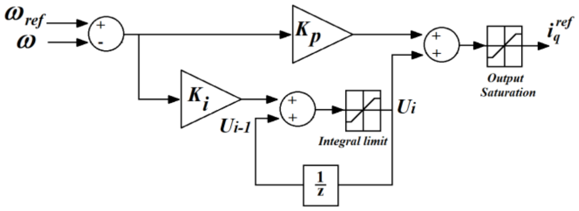


Fig. 2. Block diagram of a discrete PI speed controller with basic anti-windup.

This reduction in the involvement of the integral component occurs when the limits are reached. It assists in mitigating windup while endeavoring to maintain improved control tracking whenever possible.

The controller's general functional equations can be expressed as in (6), (7), (8), (9) and (10):

$$E = \omega_{\text{ref}} - \omega \tag{6}$$

$$U_i = U_{i-1} + K_i E T_{s_{\text{spd}}} \tag{7}$$

$$\text{if } |U_i| \geq I_{\max} \Rightarrow U_i = \text{sgn}(U_i) I_{\max} \tag{8}$$

$$i_q^{\text{ref}} = K_p E + U_i \tag{9}$$

$$\text{if } \left|i_q^{\text{ref}}\right| \geq \text{Sat}_{\text{spd}} \Rightarrow i_q^{\text{ref}} = \text{sgn}\left(i_q^{\text{ref}}\right) \cdot \text{Sat}_{\text{spd}} \tag{10}$$

where ω_{ref} represents the reference speed of the PMSM, ω is the feedback speed measured from the encoder, and U_i, U_{i-1} are the integral outputs of the PI controller at the current and previous sampling, respectively.

By incorporating the Integral Limit parameter, the output of the integral term $I_{\max}$ can be limited to a specified range. In practice, $I_{\max}$ is determined as a fixed percentage between 0% and 100%. Its value depends on the desired reference speed and the load torque. To accurately estimate its value, simulations are required to achieve the desired performance of the control system. This approach helps prevent overshooting and other undesirable effects during system control. Moreover, the Output Saturation parameter enables saturation of the overall controller output.

The speed control system is characterized by incorporating the mechanical properties of the motor. Specifically, this study is focused on the case where a single lumped sum inertia is connected tightly to the motor shaft, and no viscous damping is present. The torque equation for the mechanical system can be expressed as in (11):

$$J \frac{d\omega_m}{dt} = \frac{3}{2} p \varphi_m i_q - T_L \tag{11}$$

By employing both the mechanical and electrical torque equations, the speed of the system (mechanical speed) can be described as in (12):

$$\omega_m = \frac{K}{Js} i_q; \; K = \frac{3}{2} p \varphi_m i_q \tag{12}$$

If an ideal current controller is assumed, the block diagram of the speed control model can be simplified to a feedback control system using PI control. This is depicted in Fig. 3.

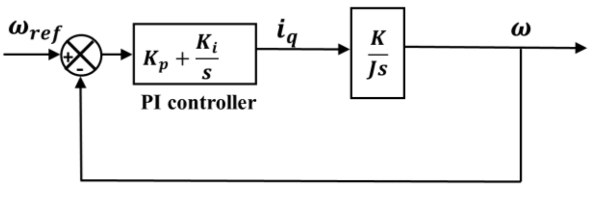


Fig. 3. Simplified block diagram of the speed loop.

The closed-loop transfer function of this system can be derived as described in (13):

$$G_S(s) = \frac{\frac{KK_i}{J}(1+\frac{s}{K_i/K_p})}{s^2+\frac{KK_p}{J}s+\frac{KK_i}{J}} \tag{13}$$

When the closed-loop characteristic polynomial $G_S(s)$ is compared with that of a second-order system with zero point, the following Eq. (14) can be derived.

$$\omega_n^2 = \frac{KK_i}{J}, 2\xi\omega_n = \frac{KK_p}{J}, \omega_z = \frac{K_i}{K_p} \tag{14}$$

where ω_n is the natural frequency of the closed loop system, ξ is the damping factor and ω_z is the zero-point frequency.

From above equations, the parameters ω_n, ξ and ω_z are written as in (15):

$$\omega_n = \sqrt{\frac{KK_i}{J}}, \xi = \frac{1}{2}\frac{KK_p}{\sqrt{JKK_i}}, \omega_z = \frac{\omega_n}{2\xi} \tag{15}$$

Speed PI control gains (K_p, K_i) are written as shown in (16) and (17):

$$K_p = \frac{2\xi_{\mathrm{D\_S}}\omega_{n,\mathrm{D\_S}}J}{K} \tag{16}$$

$$K_i = \frac{J\omega_{n,\mathrm{D\_S}}^2}{K} \tag{17}$$

where $\omega_{n,\mathrm{D\_S}}$: desired natural frequency of speed control system and $\xi_{\mathrm{D\_S}}$: desired damping ratio of speed control system.

Hence, the design of PI control gains for the speed control system can be achieved by considering the parameters $\omega_{n,\mathrm{D\_S}}$ and $\xi_{\mathrm{D\_S}}$. The controller parameters should consider also the current and speed scaling factors ($K_{\mathrm{cr}}[\mathrm{A/bit}]$ and $K_{\mathrm{spd}}[\mathrm{bit/rad/s}]$). These scaling factors depend on specific sensors parameters. Therefore, the digital speed controller is tuned using the next equations as illustrated in (18) and (19):

$$K_p = \frac{2\xi_{\mathrm{D\_S}}\omega_{n,\mathrm{D\_S}}J}{KK_{\mathrm{cr}}K_{\mathrm{spd}}} \tag{18}$$

$$K_i = \frac{J\omega_{n,\mathrm{D\_S}}^2}{KK_{\mathrm{cr}}K_{\mathrm{spd}}} \tag{19}$$

By using a discretizing step T_S the discrete speed controller can be obtained as shown in (20):

$$R(z) = K_p + \frac{K_iT_S}{1-z^{-1}} \tag{20}$$

where T_S is sampling period of the speed loop.

It is important to note that the parameters of the PI controller are configured to ensure the PMSM's proper operation under nominal conditions. The values for the gains K_p and K_i of the PI controller are $K_p = 393.73$ and $K_i = 49.22$.

B. Fuzzy Logic Controller

Fig. 4 illustrates the operational mechanism of the fuzzy logic controller employed to control the speed of a permanent magnet synchronous motor. Indeed, the speed error E_ω and its variation CE_ω have a significant impact on the machine's dynamics. Therefore, these variables are chosen as the two inputs of the speed fuzzy controller. As for the output, it is practical to use the control signal increment $\Delta I_q^{\mathrm{ref}}$ corresponding to the reference stator current i_q^{ref}.

As illustrated in Fig. 4, the inputs of the TS-FLC are calculated at time instant k as shown in (21) and (22):

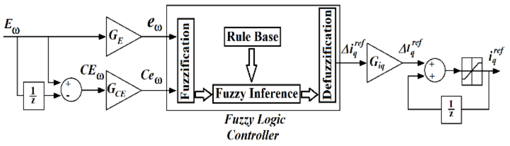


Fig. 4. Structure block diagram of fuzzy logic controller.

$$E_\omega(k) = \omega_r^{\mathrm{ref}}(k) - \omega(k) \tag{21}$$

$$CE_\omega(k) = E_\omega(k) - E_\omega(k-1) \tag{22}$$

The integration of the TS-FLC output yields the control signal associated with the stator torque component $\mathrm{i}_q^{\mathrm{ref}}$ as illustrated in (23):

$$i_q^{\mathrm{ref}}(k) = i_q^{\mathrm{ref}}(k-1) + \Delta I_q^{\mathrm{ref}}(k) \tag{23}$$

The TS-FLC block typically comprises three main components: fuzzification, fuzzy inference, and defuzzification, as depicted in Fig. 4.

1) Fuzzification

In this work, the inputs $(E_\omega, \mathrm{CE}_\omega)$ of the simplified TS-FLC are categorized into three conventional designations: negative (NG), zero (EZ), and positive (PG). On the other hand, the inputs of the standard TS-FLC are distinguished into seven conventional designations: negative large (NL), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM), and positive large (PL). As indicated in Fig. 5 and Fig. 6, triangular shapes are used for membership functions (With the exception of the endpoints of each membership function, trapezoidal shapes are used). In addition, the fuzzy partitioning follows a symmetrical and equidistant approach, ensuring that the divisions are uniformly spaced and balanced.

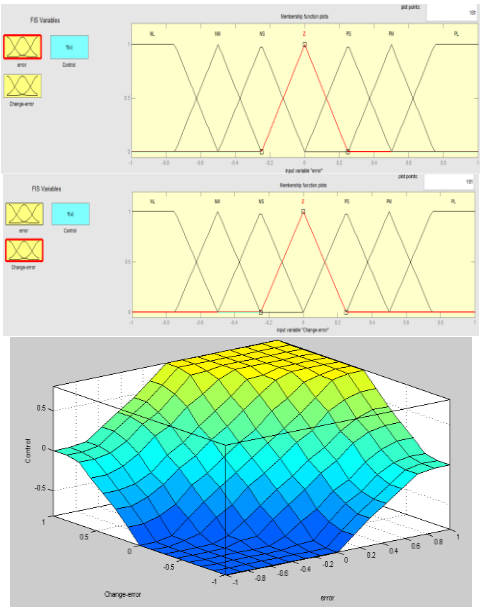


Fig. 5. Membership function of standard TS-FLC for speed error, change in speed error and output surface of fuzzy inference system.

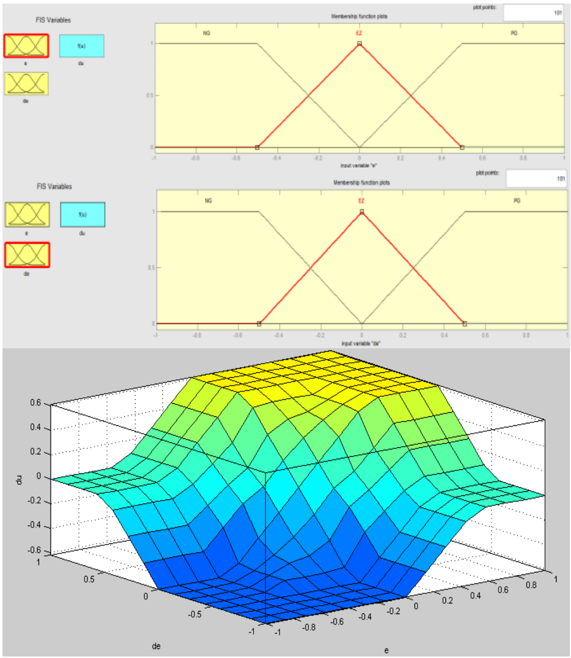


Fig. 6. Membership function of simplified TS-FLC for speed error, change in speed error and output surface of fuzzy inference system.

It is apparent that the utilization of value ranges in an FLC varies from one process to another. To develop a universal FLC, it is necessary to transform these value ranges into normalized ranges. This normalization process allows for a consistent and standardized representation of input and output variables across different applications.

The value ranges of E_ω, CE_ω and $\Delta I_q^{\mathrm{ref}}$ are normalized and defined in the per unit system within the range of (−1, 1), respectively, using the following quantities e_ω, Ce_ω and $\Delta i_q^{\mathrm{ref}}$, given as (24):

$$G_E = \frac{e_\omega(k)}{E_\omega(k)},\ G_{CE} = \frac{Ce_\omega(k)}{CE_\omega(k)},\ G_{i_q} = \frac{\Delta I_q^{\mathrm{ref}}(k)}{\Delta i_q^{\mathrm{ref}}(k)} \qquad (24)$$

Therefore, input and output gains are introduced to adjust the scaling and sensitivity of the TS-FLC (25). These gains allow for fine-tuning the influence and responsiveness of the input and output variables within the control system [26]. By adjusting these gains, the TS-FLC can be optimized to achieve the desired control performance and dynamic behavior in a specific application. Indeed, the choice and experimentation with these scaling factors play a fundamental role in the stability, oscillations, and damping of the system.

The scaling factor G_E can be calculated (25) by considering the range of speed references from minimum to maximum [−200, 200] Hz flux rotation frequency ([−6000, 6000] rpm):

$$G_E = \frac{1}{400} \cong 0{,}002 \qquad (25)$$

The calculation of the scaling factor (27), G_{CE}, for the change in speed error involves considering the motor's rated inertia J, nominal torque T_{nom} (26), and the sampling time T_s:

$$T_{\mathrm{nom}} = \frac{J}{p}\left(\frac{\Delta\omega}{T_s}\right) \qquad (26)$$

$$G_{\mathrm{CE}} = \frac{1}{\Delta\omega} \approx 0{,}017 \qquad (27)$$

The output scaling factor G_{i_q} is defined as demonstrated in (28):

$$G_{i_q} = 2^{15} - 1 \qquad (28)$$

by measuring the rated current under rated conditions.

2) *Rule bases and rule inference establishment*

The control strategy and desired objective are expressed in the rule base using linguistic control rules. These rules are obtained from a variety of sources, including expert knowledge, control experience, the actions of control operators, and the learning capabilities of the controller. By studying the system's response under various conditions and analyzing the relationships between input variables and the resulting output, these control rules are derived. In this work, a Takagi-Sugeno type fuzzy controller of order zero is implemented. In fact, this type of controller, like the Mamdani controller, is built based on an "If... Then..." rule base. The premises are always expressed linguistically and like those used in a Mamdani fuzzy controller, while the conclusions are of a numerical nature. A Takagi-Sugeno (T-S) fuzzy system uses rules written as follows:

If antecedent 1 **is** *x* **and** antecedent 2 **is** *y*,
then *z* = *f*(*x*, *y*)

where *z*=*f*(*x*, *y*) is a crisp function in the consequent, commonly represented as a polynomial function of the input variables *x* and *y*. For a zero-order Sugeno-type model, *f*(*x*, *y*) is a constant. In this model, where real numbers are employed for the output, the fuzzy set of consequence inference will be represented as a discrete fuzzy set containing a finite number of points. This characteristic significantly simplifies the calculation process of the defuzzification algorithm.

In the case of a standard zero-order T-S FLC, there are 7 fuzzy sets, indicating 49 possible combinations of these inputs, resulting in 49 rules. The rules are of the following form:

Rule 1: if $e_\omega = \mathrm{NL}$ and $Ce_\omega = \mathrm{NL}$, then $\Delta i_q^{\mathrm{ref}} = u$ where u is a constant that depends on the inputs of the FLC.

The values of output membership functions of the zero-order TS-FLC correspond to the centroids of the equivalent Mamdani FLC output membership functions. The values of these outputs for the simplified and standard TS-FLCs are provided in Table I. The 9 rules for the simple TS-FLC and the 49 rules for the standard TS-FLC can be presented in an inference matrix, referred to as the inference matrix, as shown in Table II and Table III, respectively.

TABLE I: OUTPUTS OF TS-FLCS

Output membership functions of standard zero-order TS-FLC						
$u_{\mathrm{PL}}^{49} = 0.81$	$u_{\mathrm{PM}}^{49} = 0.5$	$u_{\mathrm{PS}}^{49} = 0.25$	$u_{Z}^{49} = 0$	$u_{\mathrm{NS}}^{49} = -0.25$	$u_{\mathrm{NM}}^{49} = -0.5$	$u_{\mathrm{NL}}^{49} = -0.81$
Output membership functions of simplified zero-order TS-FLC						
$u_{\mathrm{PG}}^{9} = 0.62$			$u_{\mathrm{EZ}}^{9} = 0$		$u_{\mathrm{NG}}^{9} = -0.62$	

TABLE II: RULE TABLE FOR THE SIMPLIFIED TS-FLC

$\boldsymbol{\Delta i_q^{\mathrm{ref}}}$		**Speed error e_ω**		
		N	**EZ**	***P***
Change speed error Ce_ω	NG	u_{NG}^{9}	u_{NG}^{9}	u_{EZ}^{9}
	EZ	u_{NG}^{9}	u_{EZ}^{9}	u_{PG}^{9}
	PG	u_{EZ}^{9}	u_{PG}^{9}	u_{PG}^{9}

TABLE III: RULE TABLE FOR THE STANDARD TS-FLC

$\boldsymbol{\Delta i_q^{\mathrm{ref}}}$		**Speed error e_ω**						
		NL	**NM**	**NS**	**ZE**	**PS**	**PM**	**PL**
Change speed error Ce_ω	NL	u_{NL}^{49}	u_{NL}^{49}	u_{NL}^{49}	u_{NL}^{49}	u_{NM}^{49}	u_{NS}^{49}	u_{Z}^{49}
	NM	u_{NL}^{49}	u_{NL}^{49}	u_{NL}^{49}	u_{NM}^{49}	u_{NS}^{49}	u_{Z}^{49}	u_{PS}^{49}
	NS	u_{NL}^{49}	u_{NL}^{49}	u_{NM}^{49}	u_{NS}^{49}	u_{Z}^{49}	u_{PS}^{49}	u_{PM}^{49}
	Z	u_{NL}^{49}	u_{NM}^{49}	u_{NS}^{49}	u_{Z}^{49}	u_{PS}^{49}	u_{PM}^{49}	u_{PL}^{49}
	PS	u_{NM}^{49}	u_{NS}^{49}	u_{Z}^{49}	u_{PS}^{49}	u_{PM}^{49}	u_{PL}^{49}	u_{PL}^{49}
	PM	u_{NS}^{49}	u_{Z}^{49}	u_{PS}^{49}	u_{PM}^{49}	u_{PL}^{49}	u_{PL}^{49}	u_{PL}^{49}
	PL	u_{Z}^{49}	u_{PS}^{49}	u_{PM}^{49}	u_{PL}^{49}	u_{PL}^{49}	u_{PL}^{49}	u_{PL}^{49}

Regarding stability of the proposed TS-FLCs, it’s important to note that control systems can exhibit a significant risk of instability when the following conditions are detected: extremely rapid rise time, substantial overshooting, and exceptionally prolonged settling time. As a result, to enhance stability, the aim is to reduce overshooting and minimize settling time. These objectives can be realized through the adjustment of membership functions, rule base modifications, and the fine-tuning of scaling factors of the TS-FLCs.

3) Choice of inference method and defuzzification

Various options exist for implementing operators that act on membership functions. The selection of an inference method determines how these operators are applied in the inference process, facilitating numerical computations. In this fuzzy control, the max-min inference method is commonly employed. Each rule yields a partial membership function, which is determined using the following relationship (29):

$$\mu_{R_i}\left(ci_q^{\mathrm{ref}}\right) = \min\left[\mu_{C_i}, \mu_{O_i}\left(ci_q^{\mathrm{ref}}\right)\right] \quad i = 1, 2, \dots m \quad (29)$$

where μ_{C_i} represents the assigned membership factor for each rule R_i and $\mu_{O_i}\left(ci_q^{\mathrm{ref}}\right)$ denotes the membership function associated with the operation imposed by rule R_i.

The resulting membership function is calculated according to the following expression (30):

$$\mu\left(ci_q^{\mathrm{ref}}\right) = \max\left[\mu_{R_1}\left(ci_q^{\mathrm{ref}}\right), \mu_{R_2}\left(ci_q^{\mathrm{ref}}\right), \dots, \mu_{R_m}\left(ci_q^{\mathrm{ref}}\right)\right] \quad (30)$$

In the case where a zero-order Sugeno inference method is used, defuzzification is achieved through weighted averaging. Individual outputs relative to each rule are calculated using the principle of the average of the maxima, and then their weighted average is computed.

IV. SIMULATION RESULTS

In this section, the simulation is performed to validate the effectiveness of the proposed controllers. The digital control configuration of the PMSM, developed in the MATLAB-Simulink environment, is represented in Fig. 7 and Fig. 8.

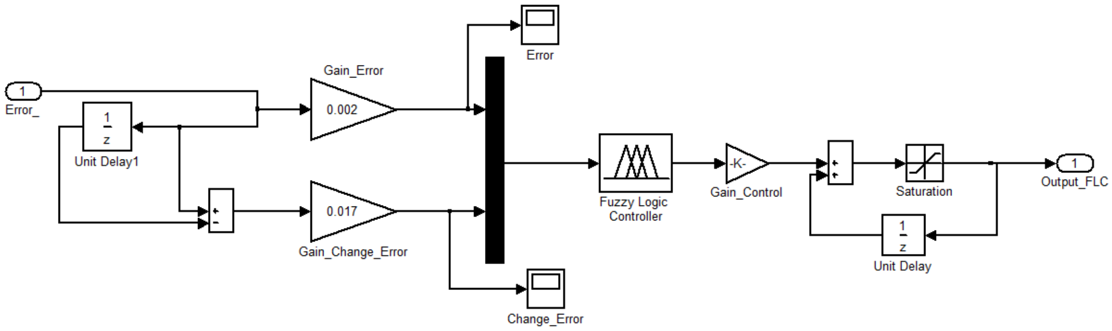


Fig. 7. Simulink block of fuzzy logic controller.

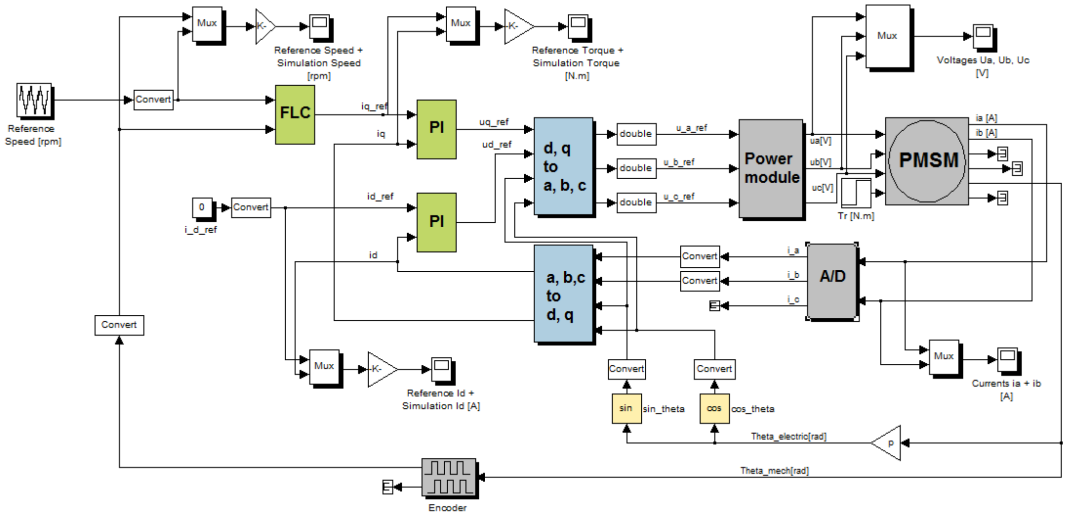


Fig. 8. Simulink model of PMSM digital control configuration with TS-FLC.

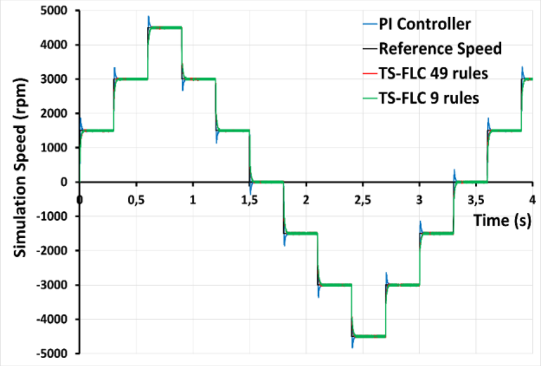
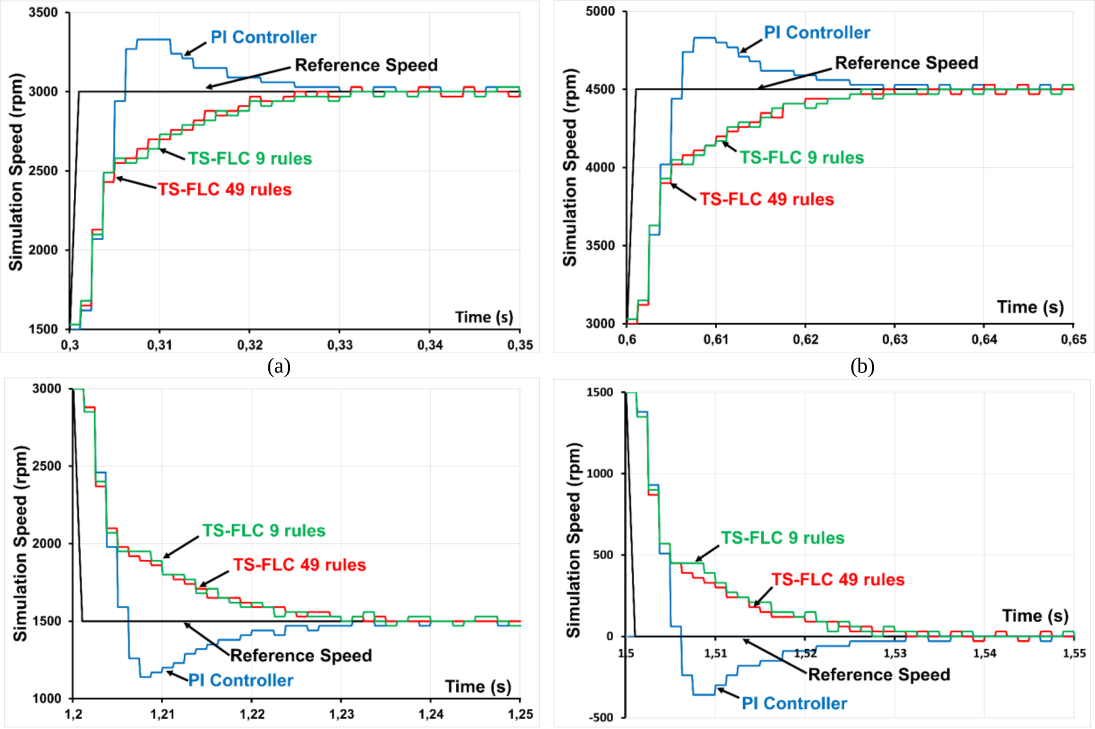


Fig. 9. Simulation rotor speed response (rpm) under no load conditions.

Each component in the system can be described by its functional equations and models, including the analog parts (motor, power converter, sensors) and the digital components (TS-FLC, digital controllers, Analog to Digital, Digital to Analog conversions and computations). By simulating various operating conditions, it becomes possible to assess the system's expected behavior, refine the system model and/or parameters, and achieve improved performance.

The simulations were conducted to evaluate the performance of the proposed controllers under various conditions. Specifically, the speed tracking capabilities of the PI controller, standard TS-FLC, and simplified TS-FLC were investigated. Fig. 9 and Fig. 10 show the results under no-load conditions, while Fig. 11 and Fig. 12 show the results under load conditions with a load torque of 0.6 N·m. The controllers were tested across a wide range of motor speeds, ranging from very low to very high speeds. To assess their performance, a reference speed profile was applied, consisting of multiple up and down steps with amplitudes of 1500 rpm, 3000 rpm, and 4500 rpm, in both directions of rotation.



(a) (b)

(c) (d)

Fig. 10. Simulation rotor speed response (rpm) at (a) 0.3 s (b) 0.6 s (c) 1.2 s (d) 1.5 s under no load conditions.

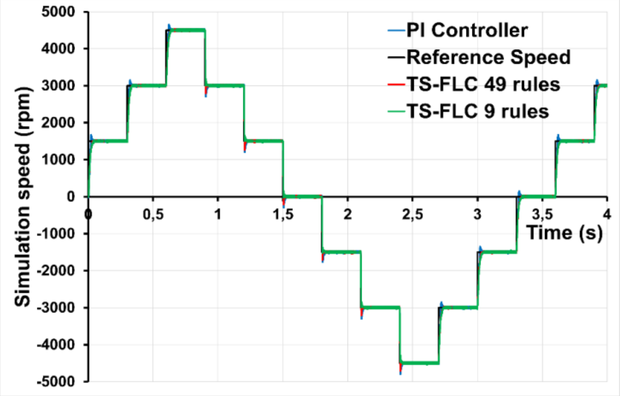


Fig. 11. Simulation rotor speed response (rpm) under load conditions.

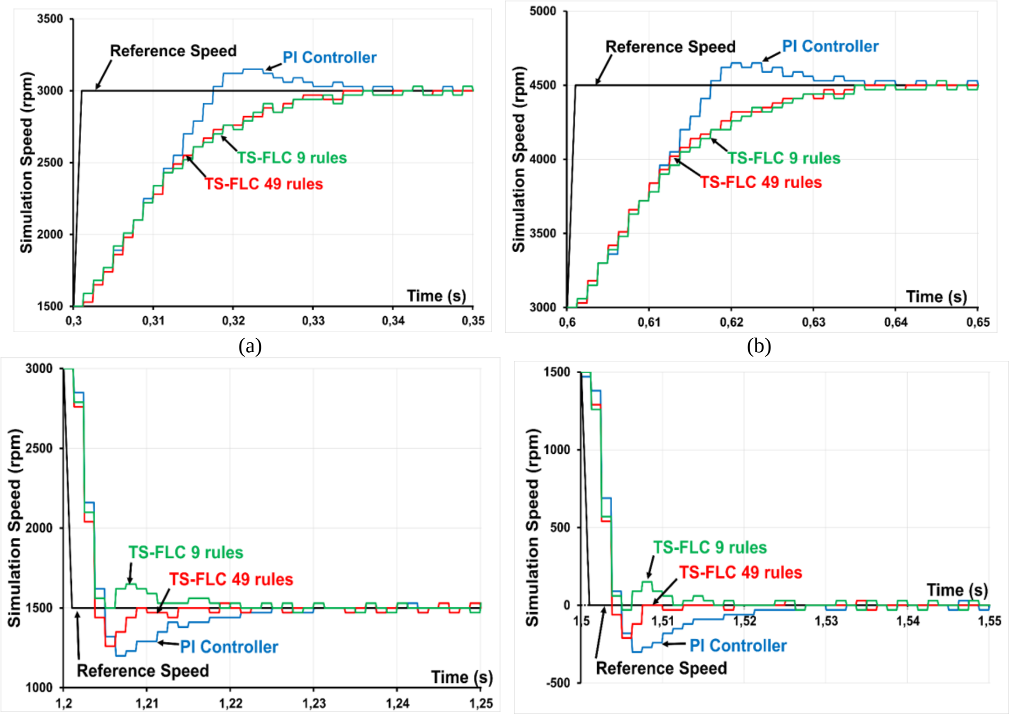
Fig. 9 and Fig. 11 demonstrate that the motor speed accurately tracks the reference speed in steady state for all the proposed controllers. However, in the transient regime, there are differences in the speed response between the PI controller and the TS-FLCs, as shown in Fig. 10 and Fig. 12. In these figures, the transient regime is illustrated by four cases labeled as (a), (b), (c), and (d), which correspond to the respective instants of 0.3 s, 0.6 s, 1.2 s, and 1.5 s. At these instants, the motor reference speed changes instantaneously as follows: (a) 1500 to 3000 rpm, (b) 3000 to 4500 rpm, (c) 3000 to 1500 rpm, and (d) 1500 to 0 rpm. Detailed comparisons of settling time and percentage overshoot for the proposed controllers under no-load and load conditions are provided in Table IV and Table V regarding cases (a) and (c). For instance, under no-load conditions, the speed response of the PI controller exhibits a 22% overshoot in case (a) and a 24% overshoot in case (b), achieving stability at 30 ms. In contrast, the speed response of the TS-FLCs shows no overshoot, with stability achieved at 23 ms for the standard TS-FLC and 25 ms for the simplified TS-FLC. It's noteworthy that the standard TS-FLC outperforms slightly the simplified TS-FLC, especially in terms of settling time, particularly under load conditions.

TABLE IV: PERFORMANCE COMPARISON OF SIMULATION SPEED CONTROL FOR PI CONTROLLER AND TS-FLCS AT 0.3 S

Reference Speed 1500 to 3000 rpm		**Overshoot (%)**	**Settling time (ms)**
No load conditions	PI Controller	22.00	21.25
	Simplified TS-FLC	0.00	22.50
	Standard TS-FLC	0.00	20.00
Load conditions	PI Controller	10.00	28.75
	Simplified TS-FLC	0.00	33.75
	Standard TS-FLC	0.00	27.50

TABLE V: PERFORMANCE COMPARISON OF SIMULATION SPEED CONTROL FOR PI CONTROLLER AND TS-FLCS AT 1.2 S

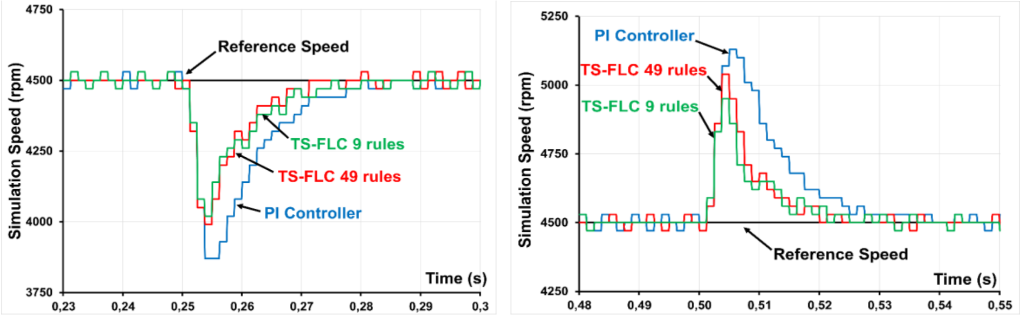
Reference Speed 3000 to 1500 rpm		**Overshoot (%)**	**Settling time (ms)**
No load conditions	PI Controller	24.00	27.50
	Simplified TS-FLC	0.00	33.75
	Standard TS-FLC	0.00	28.75
Load conditions	PI Controller	20.00	21.25
	Simplified TS-FLC	0.00	28.75
	Standard TS-FLC	16.00	13.75



(a) (b)

(c) (d)

Fig. 12. Simulation rotor speed response (rpm) at (a) 0.3 s (b) 0.6 s (c) 1.2 s (d) 1.5 s under load conditions.



(a) (b)

Fig. 13. Simulation rotor speed response (rpm) at (a) 0.25 s (b) 0.5 s during load rejection.

In addition, the load rejection capabilities of the proposed controllers were investigated. For the simulation, the motor was started with no load. Then, at time t=0.25 s, the load torque was instantaneously varied to the nominal value of 0.6 N·m for 0.25 s and then returned to the initial value of 0 N·m. The speed reference was set to a constant speed of 4500 rpm. This was done to observe the load rejection properties of the drives, at time instants 0.25 s and 0.5 s, which correspond to the cases where the load torque changes instantaneously from 0 to 0.6 N·m and from 0.6 to 0 N·m, respectively, as shown in Fig. 13.

The results indicate that at time t=0.25s, the motor speed decreases by approximately 14%, 11.33%, and 10.67% respectively for the PI controller, standard TS-FLC, and simplified TS-FLC. Then, the speed response quickly returns to the reference speed with recovery times of 30 ms, 21 ms, and 25 ms respectively for the PI controller, standard TS-FLC, and simplified TS-FLC. The performance of the TS-FLCs is practically similar, stabilizing almost at the same time and slightly outperforming the PI controller. Both TS-FLCs can maintain the speed at the reference value under varying loads.

V. DSP Real Implementation

A. *Experimental Setup*

To validate the practical performance of the proposed controllers, an application was developed using a professional development kit (MCK28335) obtained from Technosoft Company (as depicted in Fig. 14).

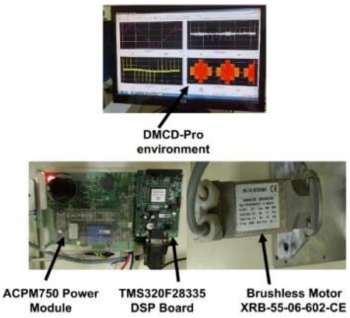


Fig. 14. Experimental setup for hardware implementation.

The kit incorporates the following components:

- An ACPM750, which is a three-phase AC power module. It comprises a 750 W six IGBT inverter equipped with two current feedback. These currents are measured using isolated transducers connected in series with the inverter outputs. The module also includes DC-bus voltage feedback and protective hardware.
- The TMS320LF28335 DSP-based development board and a DMCD-Pro (Digital Motion Control Development software).
- A brushless motor, specifically the XRB-55-06-602-CE model. This motor is a 4-pole, three-phase PMSM that includes a 500-line quadrature incremental encoder and hall sensors. The specific parameters of this machine are provided in Table VI.

TABLE VI: MOTOR CHARACTERISTICS

Motor parameter	Value
Rated voltage (V) (Y-connexion)	230
Nominal torque (N.m)	0.6
Nominal speed (rpm)	6000
Nominal power (W)	377
Peak current (A)	6.6
Stator inductance (mH)	5
Stator resistance (Ω)	3.1
Back-EMF (electro-motive force) voltage at 1000 rpm (V/Krpm)	20
Torque constant (Nm/A)	0.42
Rotor inertia (kg.cm^2)	0.251

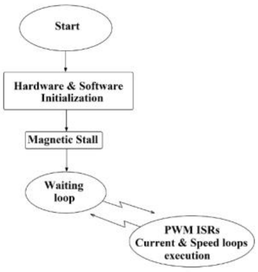

Start
Hardware & Software Initialization
Magnetic Stall
Waiting loop
PWM ISRs Current & Speed loops execution

Fig. 15. General software flowchart.

B. *Software Organization*

The code of the program is developed predominantly in C/C++ language and consists of two modules: the initialization and magnetic stall block, and the interrupt module (Fig. 15). On the other hand, the code for the speed PI controller and TS-FLCs is developed using Real-Time Workshop in the MATLAB environment. This tool enables the generation of complete C/C++ code that corresponds to the blocks in the Simulink model of these controllers.

The interrupt routines implement the motor control algorithm. They are invoked periodically and synchronized with the PWM (Pulse Width Modulation) period. The PWM frequency is determined according to the motor’s electrical constant.

Two levels of interruption are programmed [27]:

- A high-priority interrupt routine for current control;
- A low-priority interrupt routine used for speed control.

The time base of ePWM1 (Enhanced Pulse Width Modulation) module within the TMS320LF28335 DSP is configured to generate a 20-kHz PWM signal. This same timer serves a dual purpose by enabling real-time interrupt generation, which is crucial for activating both fast and slow interrupts. The initiation of an interrupt is triggered by a timer underflow event. With a 20 kHz PWM frequency, this results in an interrupt occurring every 50 μs. To manage current control, the system employs a software counter, executing the current control routine every two PWM periods, which translates to 100 μs. Furthermore, another software counter is utilized to execute the speed control routine every ten PWM periods, corresponding to a 1 ms interval.

The program execution proceeds as follows. After the program initialization phase, the rotor alignment on phase A is performed. In this process, a constant current is injected into phase A, which generates a constant and fixed-direction magnetic flux aligned with the axis of phase A. The rotor naturally aligns itself with this direction due to the presence of magnets. Thereafter, an infinite loop starts and is periodically interrupted by routines for communication with the graphical interface and/or for controller calculations. The recording of various global variables can be used for application debugging purposes.

VI. Real Time Results

Once the system has been simulated and its expected behavior has been verified, the following step involves the generation of C/C++ code for the TS-FLC control blocks, to ensure its implementation and test on the TMS320F28335 DSP controller. Experimental tests were conducted to verify the robustness and efficiency of the proposed controllers under speed reversal. These tests covered a wide range of motor speeds, like those in the simulation, as depicted in Fig. 16. This figure shows the real-time motor speed under no-load conditions using the PI controller, simplified TS-FLC, and standard TS-FLC.

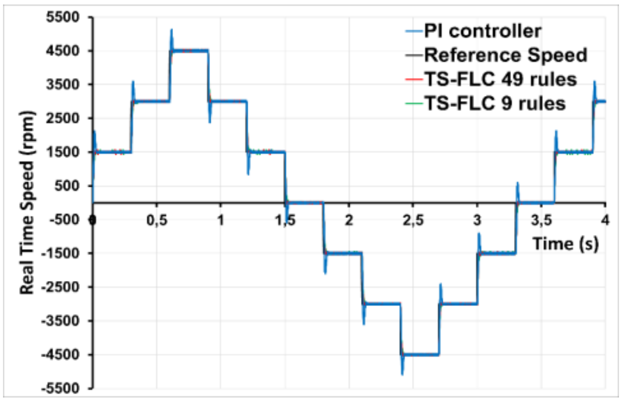


Fig. 16. Real time rotor speed (rpm) under no load conditions.

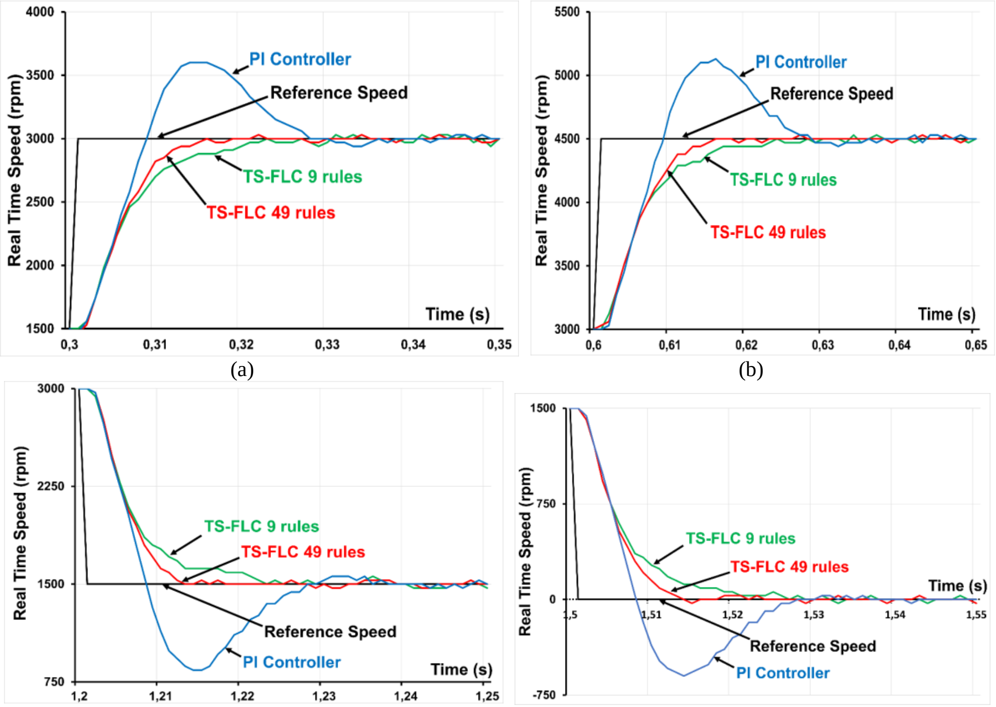
Fig. 17 specifically illustrates the real-time speed during startup at t=0.3 s, t=0.6 s, reverse operation at t=1.2 s, and t=1.5 s for the proposed controllers. these instants correspond respectively to cases (a), (b), (c), and (d) defined in the simulation section. Fig. 16 demonstrates that the motor speed accurately tracks the reference speed during steady-state operation. The stability of the drives achieved at different speeds demonstrates the reliability of the code and the controllers.

The performance of the controllers in transient state is compared in Table VII. As a result, it is clear to conclude that the TS-FLCs have a better speed response than the PI controller. Fig. 17 and Table VII demonstrate that the standard TS-FLC exhibits slightly better performance compared to the simplified TS-FLC in terms of settling time. It is observed that the real-time speed response from the PI controller shows a relatively high overshoot due to the parameters being calculated for optimal motor operation under load conditions, while the real-time test is conducted under no-load conditions.

On the other hand, both TS-FLC controllers are effective in preventing overshoot during startup, setpoint changes and reverse operation. Fig. 18 illustrates that the electromagnetic torque indicates the effectiveness of the FOC in the case of the proposed controllers. The presence of positive and negative torque peaks enables instant motor acceleration or deceleration, ensuring accurate tracking of speed steps. This further demonstrates the efficacy of the PI current controllers. During the phases where the speed stabilizes, the torque is non-zero due to viscous friction.

TABLE VII: PERFORMANCES COMPARISON OF REAL TIME SPEED CONTROL FOR PI CONTROLLER AND FLCS AT 0.3 S, 0.6 S, 1.2 S AND 1.5 S

Speed controllers	PI controller		Simplified TS-FLC		Standard TS-FLC	
Working condition	Overshoot (%)	Settling time (ms)	Overshoot (%)	Settling time (ms)	Overshoot (%)	Settling time (ms)
At 0.3 s (1500 to 3000 rpm)	40,00	24	0,00	14	0,00	11
At 0.6 s (3000 to 4500 rpm)	42,00	23	0,00	12	0,00	10
At 1.2 s (3000 to 1500 rpm)	44,00	27	0,00	21	0,00	12
At 1.5 s (1500 to 0 rpm)	40,00	26	0,00	19	0,00	12



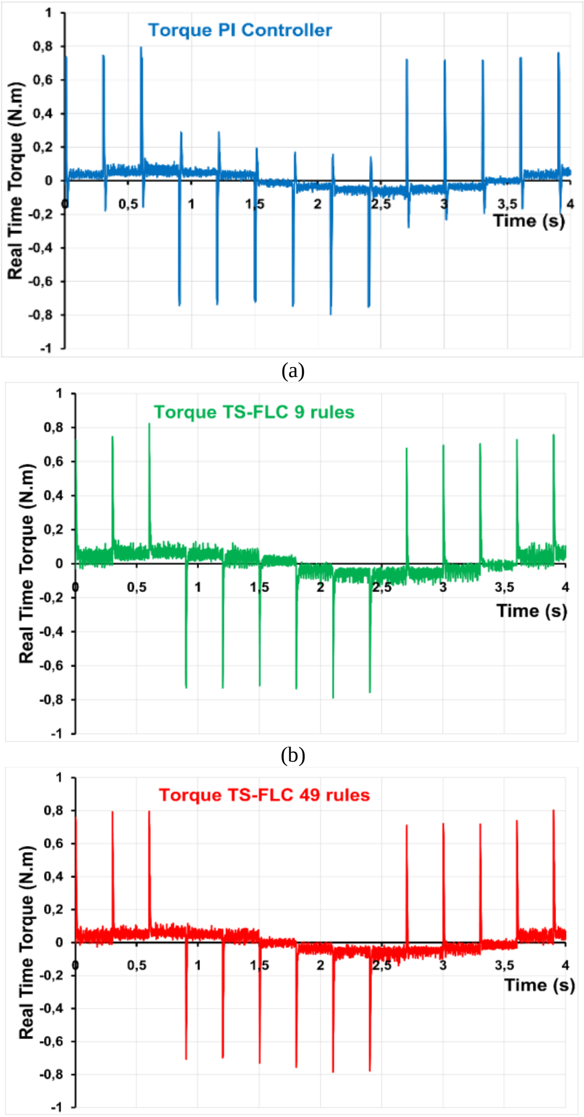
(a)

(b)

(c)

(d)

Fig. 17. Real time rotor speed (rpm) at (a) 0.3 s (b) 0.6 s (c) 1.2 s (d) 1.5 s.



(a)

(b)

(c)

Fig. 18. Real time torque response (N·m) (a) PI controller (b) Simplified TS-FLC (c) Standard TS-FLC.

VII. Conclusion

This paper presents the simulation and DSP implementation of PMSM speed control using both classic PI controller and Takagi-Sugeno fuzzy logic controller. Two fuzzy logic speed controllers, consisting of a standard TS-FLC with 49 rules and a simplified TS-FLC with 9 rules, are implemented. The performance of these controllers is compared under varying loads and for a wide range of speeds. The drive system employs a vector control method. The simulation results highlight the superior performance of TS-FLCs compared to the conventional PI controller in terms of speed tracking and load rejection capabilities. Indeed, the TS-FLCs exhibit no overshoot, while the PI controller can experience an overshoot of approximately 24% during startup operations. Furthermore, they provide faster load rejection capabilities compared to the PI controller. The real-time results also indicate that the TS-FLCs outperform the PI controller in terms of overshoot and settling time during transient regime. Specifically, the TS-FLCs demonstrate zero overshoot during both startup and reverse operations. Furthermore, the standard TS-FLC exhibits a shorter settling time in comparison to the simplified TS-FLC, with a variance ranging from 2 to 9 ms. Consequently, the simplified TS-FLC can maintain drive performance and exhibit behavior slightly inferior to that of the standard TS-FLC, but still within very acceptable limits. It is important to note that to ensure stability in the speed closed loop and achieve satisfactory dynamic performance, appropriate control design parameters such as PI controller gains and scaling factors for TS-FLCs must be carefully applied. On another note, the TMS320LF28335 digital signal processor offers high performance, enabling most of the required functionality for building a PMSM drive to be written in the C/C++ language. However, the controller blocks proposed in this work are constructed in the Simulink environment and then integrated into the main program using the Real-Time Workshop tool in the MATLAB environment. This approach significantly reduces development time and allows for the implementation of more complex controller structures. This can be achieved by first testing them in the MATLAB Simulink environment and then validating the solutions in the real control environment of the DSP. Furthermore, this software framework facilitates future developments, such as adaptive FLC or fuzzy PI parameter self-tuning, as the necessary groundwork has already been established.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

R. Askour conducted the simulation, experimentation of the proposed controllers, analyzed the results, and wrote the paper. H. Jbari contributed to the manuscript drafting. B. Bououlid Idrissi provided scientific guidance. All authors approved the final version.

References

[1] K. Ullah, J. Guzinski, and A. F. Mirza, “Critical review on robust speed control techniques for permanent magnet synchronous motor (PMSM) speed regulation,” *Energies*, vol. 15, no. 3, 2022, doi: 10.3390/en15031235.

[2] M. Nicola and C.-I. Nicola, “Improvement of linear and nonlinear control for PMSM using computational intelligence and reinforcement learning,” *Mathematics*, vol. 10, no. 24, 2022, doi: 10.3390/math10244667.

[3] A. Mishra, G. Dubey, D. Joshi, P. Agarwal, and S. P. Sriavstava, “A complete fuzzy logic based real-time simulation of vector controlled PMSM drive,” in *Proc. of 2018 2nd IEEE Int. Conf. on Power Electronics, Intelligent Control and Energy Systems*, Delhi, India: 2018, pp. 809-814, doi: 10.1109/ICPEICES.2018.8897373.

[4] P. T. M. Sahridayan and R. Gopal, “Modeling and analysis of field-oriented control based permanent magnet synchronous motor drive system using fuzzy logic controller with speed response improvement,” *IJECE*, vol. 12, no. 6, 2022, doi: 10.11591/ijece.v12i6.pp6010-6021.

[5] U. R. S. Yalavarthy and V. S. K. R. Gadi, “Modelling, simulation and analysis of indirect space vector control of electric vehicle driven by permanent magnet synchronous motor with fuzzy controller,” *MMEP*, vol. 9, no. 2, pp. 523-532, 2022, , doi: 10.18280/mmep.090231.

[6] J. Peng and M. Yao, “Overview of predictive control technology

for permanent magnet synchronous motor systems," *Applied Sciences*, vol. 13, no. 10, 2023, doi: 10.3390/app13106255.

[7] M. S. Reddy, V. S. R. Subrahmanyam, C. S. Prakash, and G. V. S. U. Shankar, "Performance analysis of fuzzy and neural controller implementation in permanent magnet synchronous motor," in *Proc. of 2021 5th Int. Conf. on Intelligent Computing and Control Systems*, Madurai, India, 2021, pp. 566-574, doi: 10.1109/ICICCS51141.2021.9432232.

[8] L. Mo, Y. Liu, and Y. Zhang, "Sliding mode variable structure control for surface permanent magnet synchronous motors based on a fuzzy exponential reaching law," *Mathematical Problems in Engineering*, vol. 2019, pp. 1-14, May 2019, doi: 10.1155/2019/8340956.

[9] J. Ding, T. Zhang, R. Tao, and A. Wang, "Research on PMSM vector control system based on fuzzy PI parameter self-tuning," in *Proc. of 2020 10th Int. Conf. on Power and Energy Systems (ICPES)*, Chengdu, China, 2020, pp. 305-310, doi: 10.1109/ICPES51309.2020.9349642.

[10] K. Suleimenov and T. D. Do, "Design and analysis of a generalized high-order disturbance observer for PMSMs with a fuzy-PI speed controller," *IEEE Access*, vol. 10, pp. 42252-42260, 2022, doi: 10.1109/ACCESS.2022.3167429.

[11] H. Jigang, W. Jie, and F. Hui, "An anti-windup self-tuning fuzzy PID controller for speed control of brushless DC motor," *Automatika*, vol. 58, no. 3, pp. 321-335, 2017, doi: 10.1080/00051144.2018.1423724.

[12] B. Song, Y. Xiao, and L. Xu, "Design of fuzzy PI controller for brushless DC motor based on PSO–GSA algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 67-77, 2020, doi: 10.1080/21642583.2020.1723144.

[13] N. Vu Quynh, "The fuzzy PI controller for PMSM's speed to track the standard model," *Mathematical Problems in Engineering*, vol. 2020, 2020, doi: 10.1155/2020/1698213.

[14] C. Wang and Z. Q. Zhu, "Fuzzy logic speed control of permanent magnet synchronous machine and feedback voltage ripple reduction in flux-weakening operation region," *IEEE Trans. on Ind. Applicat.*, vol. 56, no. 2, pp. 1505-1517, 2020, doi: 10.1109/TIA.2020.2967673.

[15] P. Sahridayan and R. Gopal, "Fuzzy logic controller based performance of SPMSM fed with improved direct torque control," *IJRER*, 2019, doi: 10.20508/ijrer.v9i3.9625.g7714.

[16] Q. Chen, S. Kang, L. Zeng, Q. Xiao, C. Zhou, and M. Wu, "PMSM control for electric vehicle based on fuzzy PI," *IJEHV*, vol. 12, no. 1, 2020, doi: 10.1504/IJEHV.2020.104251.

[17] S. Ünsal and I. Aliskan, "Investigation of performance of fuzzy logic controllers optimized with the hybrid genetic-gravitational search algorithm for PMSM speed control," *Automatika*, vol. 63, no. 2, pp. 313-327, 2022, doi: 10.1080/00051144.2022.2036936.

[18] F. V. A. Raj and V. K. Kannan, "Adaptive ELM neural computing framework with fuzzy PI controller for speed regulation in permanent magnet synchronous motors," *Soft Comput*, vol. 24, no. 14, pp. 10963-10980, 2020, doi: 10.1007/s00500-020-04994-6.

[19] C.-Q. Zhong, L. Wang, and C.-F. Xu, "Path tracking of permanent magnet synchronous motor using fractional order fuzzy PID controller," *Symmetry*, vol. 13, no. 7, 2021, doi: 10.3390/sym13071118.

[20] C. Sain, A. Banerjee, P. K. Biswas, T. S. Babu, and T. Dragicevic, "Updated PSO optimised fuzzy-PI controlled buck type multi-phase inverter-based PMSM drive with an over-current protection scheme," *IET Electric Power Applications*, vol. 14, no. 12, pp. 2331-2339, 2020, doi: 10.1049/iet-epa.2020.0165.

[21] H. Ghadiri, H. Khodadadi, H. Eijei, and M. Ahmadi, "PSO based Takagi-Sugeno fuzzy PID controller design for speed control of permanent magnet synchronous motor," *Facta Univ Electron Energ*, vol. 34, no. 2, pp. 203-217, 2021, doi: 10.2298/FUEE2102203G.

[22] Y.-C. Chang, C.-H. Chen, Z.-C. Zhu, and Y.-W. Huang, "Speed control of the surface-mounted permanent-magnet synchronous motor based on Takagi–Sugeno fuzzy models," *IEEE Trans. Power Electron.*, vol. 31, no. 9, pp. 6504-6510, 2016, doi: 10.1109/TPEL.2015.2504392.

[23] Q. A. Tarbosh, Ö. Aydoğdu, N. Farah *at al.*, "Review and investigation of simplified rules fuzzy logic speed controller of high-performance induction motor drives," *IEEE Access*, vol. 8, pp. 49377-49394, 2020, doi: 10.1109/ACCESS.2020.2977115.

[24] B. Sarsembayev and K. Suleimenov, and T. D. Do, "High order disturbance observer-based PI-PI control system with tracking anti-windup technique for improvement of transient performance of PMSM," *IEEE Access*, vol. 9, pp. 66323-66334, 2021, doi: 10.1109/ACCESS.2021.3074661.

[25] B. Gao, G. Wang, Z. Li, W. Song, and H. Lin, "A novel exponential-type anti-windup method applied to the power converter in marine electrical propeller system," *IET Power Electronics*, vol. 15, no. 7, pp. 571-581, 2022, doi: 10.1049/pel2.12250.

[26] M. Z. B. Ismail, Md. H. N. Talib, Z. Ibrahim, J. B. M. Lazi, M. S. A. M. Yusoff, and B. B. Ismail, "Experimental investigations into the behavior of scaling factors in a fuzzy logic speed control induction motor with model reference adaptive control," *ECTI-EEC*, vol. 20, no. 2, pp. 174-185, 2022, doi: 10.37936/ecti-eec.2022202.246896.

[27] R. Askour and B. Bououlid Idrissi, "DSP-based sensorless speed control of a permanent magnet synchronous motor using sliding mode current observer," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 4, no. 3, pp. 281-289, 2014, doi: 10.11591/ijpeds.v4i3.5704.

Rachid Askour, originally from Casablanca, Morocco, obtained his Ph.D. degree from National Higher School of Arts and Crafts (ENSAM-Meknès), Meknès, Morocco, in 2015. He also holds a Master's degree in electrical engineering from University of Science and Technology of Lille, France, completed in 2002, and an engineer's degree in Electromechanical Engineering from National Higher School of Mines of Rabat, Morocco, achieved in 1996. His primary areas of interest include Power Electronics, Electric Drives, Motion Control, DSP based control systems, and Electrical Vehicles. Since 1997, he has been working at ENSAM-Meknès, Moulay Ismaïl University, Meknès, Morocco, where he currently serves as an assistant professor in the Department of Electromechanical Engineering.

Hatim Jbari was born in Benslimane-Morocco, in 1986. He obtained a degree in electromechanical engineering in 2012 from the National High School of Arts and Crafts of Meknes (ENSAM-Meknes) in Morocco, and a Master's degree in electric vehicle engineering from the Paris Institute of Technology in Paris-France. He is currently a Ph.D. student in electrical engineering at ENSAM-Meknes. He has published six research papers in reputed journals. His research focuses on electric vehicle, modeling and control of multi-physics systems and energy management systems.

Badr Bououlid Idrissi, originally from Marrakech, Morocco, earned his Ph.D. degree from Polytechnic Faculty of Mons, Belgium, in 1997. He also holds an engineer's degree from National Higher School of Mines of Rabat, Morocco, which he obtained in 1992. Since 1999, he has been affiliated with Higher School of Arts and Crafts (ENSAM-Meknès), Moulay Ismaïl University, Meknès, Morocco, where he currently serves as a Professor in the Department of Electromechanical Engineering, specializing in power electronics and electrical machines. His research interests primarily revolve around electric drives, industrial control systems, DSP/FPGA based control systems, and Electrical Vehicles.