

Article

Sensorless Control of PMSM Based on Backstepping-PSO-Type Controller and ESO-Type Observer Using Real-Time Hardware

Claudiu-Ionel Nicola ^{1,2,*} , Marcel Nicola ^{2,*}  and Dan Selișteanu ¹ 

¹ Department of Automatic Control and Electronics, University of Craiova, 200585 Craiova, Romania; dansel@automation.ucv.ro

² Research and Development Department, National Institute for Research, Development and Testing in Electrical Engineering-ICMET Craiova, 200746 Craiova, Romania

* Correspondence: claudiu@automation.ucv.ro (C.-I.N.); marcel_nicola@yahoo.com (M.N.)

Abstract: In the case of using a Permanent Magnet Synchronous Motor (PMSM) linear model of limited-range parametric variations and of relatively low dynamic of the load torque, the Field Oriented Control (FOC) type strategy ensures good performance of the PMSM control. Therefore, when using a non-linear model of wide-range parametric variations and of high dynamic of the load torque, a backstepping-type controller is proposed, whose tuning parameters are optimized by using a Particle Swarm Optimization (PSO) method. By designing an Extended State Observer (ESO), which provides a good estimate of the PMSM rotor position and speed under uncertainty conditions and with a response time shorter than that of the backstepping-type controller, this observer can be incorporated into the PMSM sensorless control system. The superior performance of the proposed sensorless control system based on the backstepping-PSO-type controller and an ESO-type observer is demonstrated through numerical simulations. Given that the real-time implementation of the control algorithms and observers in an embedded system is a difficult task, consisting of several steps, it is presented after the numerical simulations, which can be assimilated into the Software-in-the-Loop (SIL) step, the Processor-in-the-Loop (PIL) intermediate step, and the Hardware-in-the-Loop (HIL) final step. A comparison between the backstepping-PSO-type controller and the PI-PSO-type controller is presented by means of the real-time implementation of these controllers and demonstrates the superiority of the backstepping-PSO-type controller.

Keywords: permanent magnet synchronous motor; backstepping control; extended state observer; hardware-in-the-loop



Citation: Nicola, C.-I.; Nicola, M.; Selișteanu, D. Sensorless Control of PMSM Based on Backstepping-PSO-Type Controller and ESO-Type Observer Using Real-Time Hardware. *Electronics* **2021**, *10*, 2080. <https://doi.org/10.3390/electronics10172080>

Academic Editors: Enric Vidal Idiarte, Eric Monmasson and Miro Milanovic

Received: 28 June 2021

Accepted: 25 August 2021

Published: 27 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

PMSMs have increasing applicability due to their advantages such as superior performance, reliability, and the fact that they are compact and suitable for operating conditions. Thanks to its high-duty density and smaller dimensions, the PMSM has become the preferred solution for the control of speed and position in machine tools and robots. A PMSM is a motor that uses permanent magnets to generate the magnetic field in the air gap, instead of electromagnets. These motors have significant advantages, drawing the interest of researchers and industry for their use in many applications. PMSMs are widely used in low-power and medium-power applications such as computer peripherals, robotics, adjustable speed systems, electric vehicles, and heavy-duty systems such as aerospace technology [1–3].

Usually, systems with high-performance motors require fast and accurate response, rapid recovery from any disturbances and insensitiveness to parameter variations. The dynamic behavior of an AC motor can be significantly improved with the help of vector control theory—FOC, in which the motor variables are transformed into a set of orthogonal axes $d-q$ so that speed and torque can be controlled separately. This gives PMSM-type motors the desired capacity of the dynamic performance of the DC voltage machine

energized separately, while retaining the overall advantages of AC machines over DC motors [4,5].

The FOC-type control strategy of the PMSM includes all the advantages deriving from the simplicity of using PI controllers, but inherently, the control performance is limited due to the non-linear PMSM model, the need to perform the control in a wide range and with a high-dynamic of speed and load torque, but also due to the parametric uncertainties resulting, in particular, from the variation of the combined rotor-load moment of inertia [6,7].

We note the difficulty of such an approach in view of the fact that it uses mathematical models of electromechanical systems (the mathematical model of the PMSM) with the mathematical models of power electronics (the mathematical model of IGBT, controlled rectifiers, inverters, etc.). Another characteristic regarding the difficulty of harmonizing these models is that the mathematical models of electromechanical systems are continuous (which can be discretized with the usual methods) while the mathematical models of power electronic systems are hybrid (continuous-discrete). The rapid progress in computing technology and hardware in recent years has enabled the development of several directions for creating PMSM modelling and control strategies, based on predictive controllers [8,9], adaptive controllers [10,11], fuzzy controllers [12,13], or controllers based on artificial neural networks [14,15], but also controllers based on the fractional calculus [16,17] in order to improve the functionality of industrial installations. Moreover, to eliminate some of the transducers and mechanical parts which can decrease the reliability of the system, a series of observers has been developed for the estimation of the PMSM rotor speed and position [17–25].

This article is a continuation of the article in [26]. Starting from the fact that the mathematical model of the PMSM is non-linear, we developed a backstepping controller, which is built in several stages using Lyapunov functions, and thus has the possibility to achieve a control system that incorporates the PMSM nonlinearities as well as good parametric robustness in terms of maintaining the performance of the PMSM control system, even in the case of parameter variation. Among the parametric variations that stress the control system with a view to maintaining high performance, we recall the variation of the combined rotor-load moment of inertia, the variation of the stator resistance of the PMSM, but also the sudden and high-amplitude variations of the load torque. To eliminate the position and speed transducers, an ESO-type observer is synthesized with very good performance in estimating the speed and position of the PMSM rotor, even when there are uncertainties. The response time of the observer is very good, shorter than that of the control system, and because of this, the integration of the observer into the control system is an appropriate solution. An observer of the load torque and of the stator resistance of the PMSM was also designed. The numerical simulations of the PMSM control algorithms are performed in the MATLAB/Simulink development environment, which uses the Simscape Electrical toolbox [27]. After synthesizing the backstepping control law, the performances of the control system and its robustness in the case of large variations of the system parameters are demonstrated through numerical simulations. Naturally, the next step is the real-time implementation using embedded systems. Given that the real-time implementation of the control algorithms and observers in a digital signal processor (DSP), such as the TMS320F28379D microcontroller from Texas Instruments [28], is a difficult task consisting of several steps, it is presented after the numerical simulations, which can be assimilated to the SIL step, the PIL intermediate step, and the HIL final step.

The main contributions of this paper can be summarized as follows:

- Design of a backstepping-type controller, optimization of its tuning parameters using a PSO optimization method, and synthesis of an ESO-type observer to estimate the speed and position of the PMSM rotor and the load torque;
- Validation of the superior performance of the proposed sensorless PMSM control system through numerical simulations, based on the backstepping-PSO-type con-

troller and ESO-type observer under the conditions of parametric variations and high-dynamic of the load torque;

- Presentation of all the steps for the real-time implementation in the embedded system, covering the SIL, PIL, and HIL steps. Moreover, the real-time implementation demonstrates the superiority of the backstepping-PSO-type controller over the PI-PSO-type controller.

The rest of the paper is structured as follows. The description of the mathematical model of the PMSM and the FOC-type strategy are presented in Section 2 and the backstepping control of the PMSM is presented in Section 3. The design of the ESO-type observer is presented in Section 4, and Section 5 presents the numerical simulations of the sensorless PMSM control system. Section 6 presents the PIL and HIL steps for real-time implementation and experimental results. Some conclusions are presented in the final section.

2. Description of the Mathematical Model of the PMSM and the FOC-Type Strategy

In order to achieve PMSM control systems, it is necessary to develop and implement an appropriate mathematical model of this type of motor to allow the reasonably accurate description of its behavior and parameters. The mathematical model can be described in a real frame abc , or in a rotating frame $d-q$. By using the abc reference frame to describe the mathematical model of the motor, the equations obtained are nonlinear, which means that they are relatively complex, and the variables are time variants.

Thus, if the rotor reference frame $d-q$ is used, the equations obtained are in the form of relations (1), allowing for the mathematical calculations required by the numerical simulations to be carried out faster, and thus, the implementation of the control system is simplified [4].

$$\begin{cases} \frac{di_d}{dt} = -\frac{R_s}{L_d}i_d + \frac{L_q}{L_d}n_p\omega i_q + \frac{1}{L_d}u_d \\ \frac{di_q}{dt} = -\frac{R_s}{L_q}i_q - \frac{L_d}{L_q}n_p\omega i_d - \frac{\lambda_0}{L_q}n_p\omega + \frac{1}{L_q}u_q \\ \frac{d\omega}{dt} = \frac{3}{2} \frac{n_p}{J} (\lambda_0 i_q + (L_d - L_q)i_d i_q) - \frac{1}{J}T_L - \frac{B}{J}\omega \\ \frac{d\theta_e}{dt} = n_p\omega \end{cases} \quad (1)$$

where u_d and u_q represent stator voltages in the $d-q$ reference frame, and i_d and i_q represent the currents in the $d-q$ reference frame. The stator inductances in the $d-q$ reference frame will be denoted as L_d and L_q , and the stator resistances in the $d-q$ reference frame as R_d and R_q . The electrical angular velocity of the rotor is denoted as ω_e , and the linkage stream as λ_0 .

Based on the description equations in the three-phase system in the abc reference frame, we obtained the equations of the form (1) in the $d-q$ reference frame, using a transform as follows [4]:

$$\begin{bmatrix} d \\ q \\ 0 \end{bmatrix} = \begin{bmatrix} \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (2)$$

where a , b , and c represent the components of the three-phase system in the abc reference frame, d and q are used to denote the components for the two-axis system in the rotating reference frame, and 0 represents the zero component for the two-axis system in the stationary reference frame.

Using the differential operator denoted as ρ in the relations (1), we can obtain another form of interest of the description equations in the d - q reference frame, i.e., the form under which the control voltages u_d and u_q are highlighted.

$$\begin{bmatrix} u_q \\ u_d \end{bmatrix} = \begin{bmatrix} R_q + \rho L_q & \omega_e L_d \\ -\omega_e L_q & R_d + \rho L_d \end{bmatrix} \begin{bmatrix} i_q \\ i_d \end{bmatrix} + \begin{bmatrix} \omega_e \lambda_0 \\ \rho \lambda_0 \end{bmatrix} \quad (3)$$

The electromagnetic fluxes of the PMSM in the d - q reference frame are described by the following relations:

$$\begin{aligned} \lambda_q &= L_q i_q \\ \lambda_d &= L_d i_d + \lambda_0 \end{aligned} \quad (4)$$

The electromagnetic torque developed by the PMSM will be denoted as T_e , and the following relations can be written:

$$\begin{aligned} T_e &= \frac{3}{2} n_p (\lambda_d i_q - \lambda_q i_d) \\ T_e &= K_t i_q \\ T_e &= T_L + B\omega + J \frac{d\omega}{dt} \end{aligned} \quad (5)$$

In Equation (5), the following notations are used: $K_t = \frac{3}{2} n_p \lambda_0$ —the torque constant; B is the viscous friction coefficient; J is the rotor inertia; n_p is the number of pole pairs; T_L is the load torque.

Using commonly accepted simplifications, $L_d = L_q = L$, $R_d = R_q = R_s$, the following relations can be written:

$$\begin{bmatrix} \dot{i}_d \\ \dot{i}_q \\ \dot{\omega} \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L} & n_p \omega & 0 & 0 \\ -n_p \omega & -\frac{R_s}{L} & -\frac{n_p \lambda_0}{L} & 0 \\ 0 & \frac{K_t}{J} & -\frac{B}{J} & 0 \\ 0 & 0 & n_p & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ \omega \\ \theta_e \end{bmatrix} + \begin{bmatrix} \frac{u_d}{L} \\ \frac{u_q}{L} \\ -\frac{T_L}{J} \\ 0 \end{bmatrix} \quad (6)$$

where θ_e represents the electrical angular position of the PMSM rotor, θ represents the electrical angular position of the PMSM rotor ($\theta_e = n_p \cdot \theta$) and $\omega_e = n_p \cdot \omega$, and ω represents the angular velocity of the PMSM rotor.

For the control of the PMSM in the classical structure, the so-called FOC strategy control (Figure 1), the outer speed control loop provides references for the inner current control loops i_d and i_q . The reference for i_{dref} is selected as equal to 0 for the maximization of the electromagnetic torque T_e . The controllers of the three control loops are normally of PI-type.

Figure 2 presents the time evolution for the numerical simulation of the PMSM control system based on the PI speed controller. The nominal parameters of the PMSM are presented in Table 1. In this numerical simulation, the speed reference $\omega_{ref} = 1600$ rpm, the load torque $T_L = 1$ Nm, and the tuning parameters of the speed controller are $K_p = 1.1$ and $K_i = 11$. This controller was tuned using the Ziegler–Nichols method.

A PSO optimization method will be used to achieve superior performance of the PMSM control system. This method will yield the optimal values of parameters K_p and K_i in the sense of minimizing an error criterion of the following type:

$$f(t) = \int_0^t e^2(t) dt \quad (7)$$

where the error $e(t)$ is defined as follows:

$$e(t) = \omega_{ref}(t) - \omega(t) \quad (8)$$

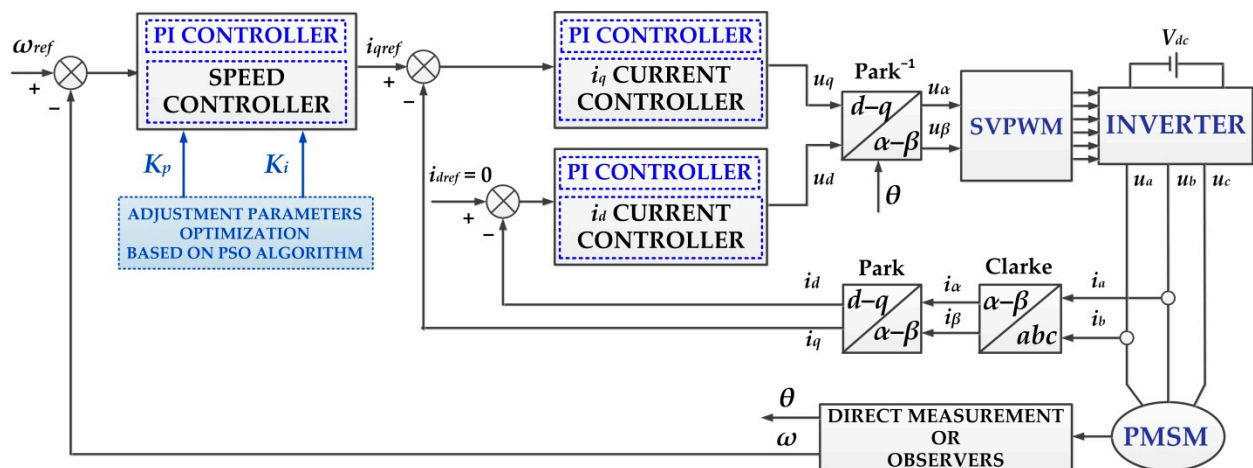


Figure 1. The control system of the PMSM based on the FOC strategy and the PI-PSO speed controller.

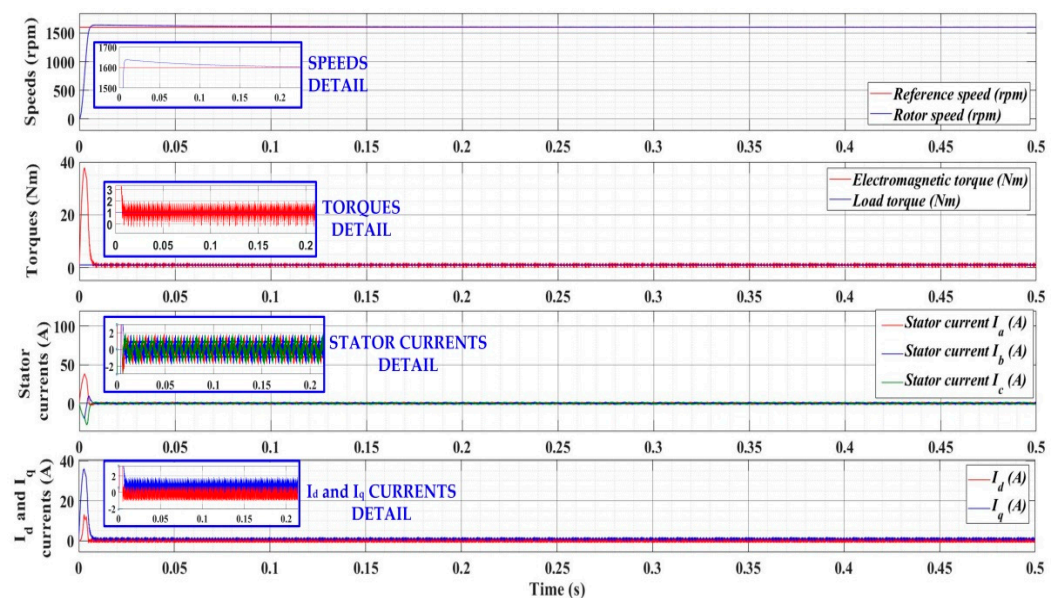


Figure 2. Time evolution for the numerical simulation of the PMSM control system based on the PI speed controller: $\omega_{ref} = 1600$ rpm, $T_L = 1$ Nm, $K_p = 1.1$, and $K_i = 11$.

Table 1. The nominal parameters of the PMSM.

PMSM Parameters	Value	Unit
The stator resistance— R_s	2.875	Ω
The d -axis and q -axis inductances— L_d and L_q	0.0085	H
The combined rotor inertia coefficient— J	0.8×10^{-3}	kg·m ²
The combined viscous friction coefficient— B	0.005	N·m·s/rad
The flux induced by the rotor in the stator phases— λ_0	0.175	Wb
The pole pairs number— n_p	4	-

The PSO method can be described in an analogous way to the trajectory of a group of particles in the search space with a view to obtaining an optimal trajectory in terms of minimizing an error criterion, i.e., approximating a reference trajectory [29]. The current

position of each particle is denoted as x_i and the current speed of each particle as v_i . Based on these, we can describe the personal optimum of a particle as follows:

$$y_i(t+1) = \begin{cases} x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \\ y_i(t) & \text{else} \end{cases} \quad (9)$$

Furthermore, by denoting the neighborhood best position as \hat{y}_i , the social optimum for the neighborhood of a particle can be described in the following form:

$$\hat{y}(t) = \operatorname{argmin}(f(y_1(t)), f(y_2(t)), \dots, f(y_n(t))) \quad (10)$$

Recurrent relations between position and speed can be described as follows:

$$\begin{cases} v_{i,j}(t) = wv_{i,j}(t) + c_1r_{1,j}(t)(y_{i,j}(t) - x_{i,j}(t)) + c_2r_{2,j}(t)(\hat{y}_{i,j}(t) - \hat{x}_{i,j}(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (11)$$

where w is the inertia weight (usual $w = 0.9$), $i = 1, \dots, N$ (N represents the number of particles), $j = 1, \dots, D$ (D represents the dimension of the problem), r_1 and r_2 are random numbers, and c_1 and c_2 are positive constants.

In the initialization phase of the program, the number of N particles is chosen and the weights and constants in the relations (11) are selected. In this application, $N = 50$, $c_1 = 0.12$ and $c_2 = 1.2$. The stopping of the algorithm is performed after achieving the convergence or after completing a predefined number of iterations.

Following the use of such an algorithm, the optimal values can be obtained for the tuning parameters of the speed controller: $K_p = 1.69$ and $K_i = 1.56$. Based on these new values, the results of the numerical simulations of the PMSM control system are presented in Figure 3. We note an improvement in the transient and stationary regime in terms of reducing the overshoot and obtaining a response time of about 5 ms.

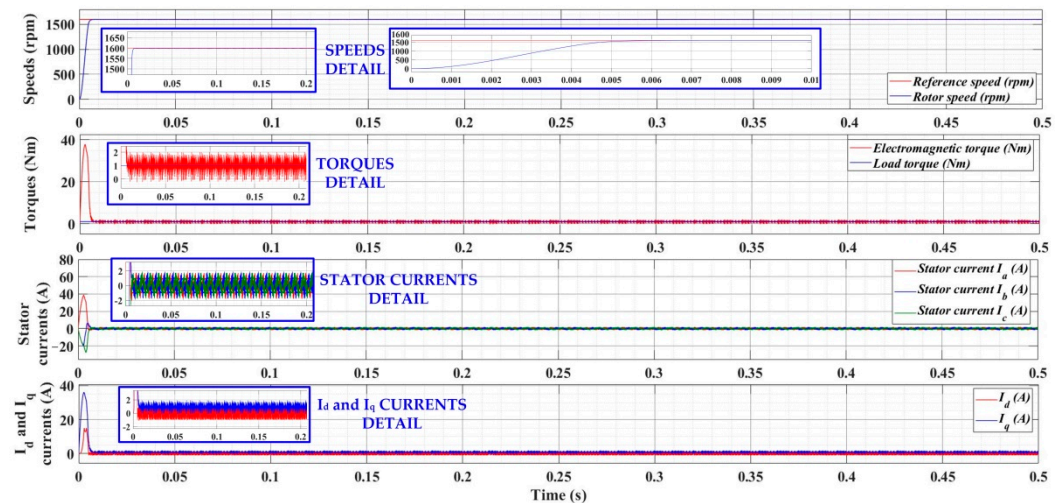


Figure 3. Time evolution for the numerical simulation of the PMSM control system based on the PI-PSO speed controller: $\omega_{ref} = 1600$, $T_L = 1$ Nm, $K_p = 1.69$, and $K_i = 1.56$.

If the variation of the load torque is between 1 Nm and 4 Nm, at time 0.25 s after the start of the simulation, a steady-state error of about 1% is noted (Figure 4).

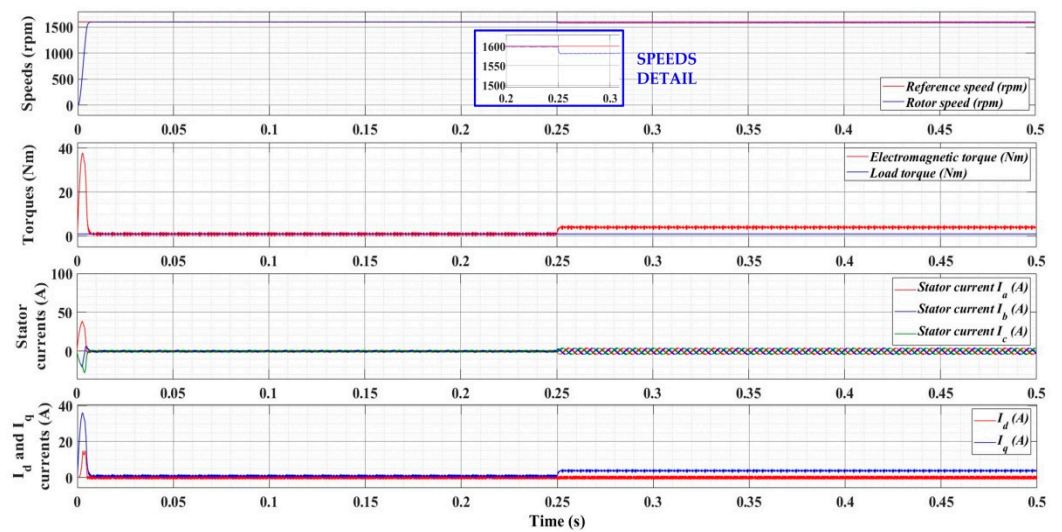


Figure 4. Time evolution for the numerical simulation of the PMSM control system based on the PI-PSO speed controller: $\omega_{ref} = 1600$ rpm, $T_L = 1 \text{ Nm} \rightarrow 4 \text{ Nm}$, $K_p = 1.69$, and $K_i = 1.56$.

This confirms that even an optimally tuned PI controller can achieve very good performance of the PMSM control system only in the case of a usual range of the reference quantity, but for relatively limited and small ranges of the other parameters. In this respect, to achieve superior performance of the PMSM control system as well as the parametric robustness of the control system, an advanced backstepping control system is presented in the next section.

3. Backstepping-Type Control Law

The mathematical model of the PMSM is nonlinear. The nonlinearity is due to the fact that products of the state vector components (terms such as $\omega \cdot i_q$ and $\omega \cdot i_d$) occur in the mathematical model. These nonlinearities can be considered as “soft” in the sense that the description functions concerned are of continuous type and even derivable due to the physical characteristics of the state vector components. It can be noted that the procedure for obtaining the controller presented starts from the FOC-type structure which exploits the cascade structure, i.e., the fact that the outer speed control loop provides the control quantity i_{qref} , which will supply the control quantities u_d and u_q through the current control loops. It can be noted that this structure can be used to follow the steps in designing a backstepping control law. The recursive control design for the backstepping law requires a system consisting of subsystems which radiate out from an inner system to the outer subsystems, and the stabilization is achieved in several stages, each based on a Lyapunov function. In addition, the functions that describe the nonlinearities of the system must be at least of Lipschitz type, a condition which is fulfilled, given that these functions are continuous due to the physical nature of the state vector components.

The backstepping control strategy is performed in a series of stages, as a cascade control system with several stages and applies especially to nonlinear systems. It starts from a stable subsystem and performs a cascade synthesis of the intermediate control signals (virtual controls) until each subsystem is stabilized. A Lyapunov function is used to design the control of each subsystem. Stability is thus ensured and, by choosing the appropriate parameters, we can obtain the intended control performance as well as the parametric robustness of the control system. The calculation of the control laws is recursive, in the sense that in each stage, the calculation is based on the states of the previous system (including the virtual control generated by it), thus ensuring the convergence of the entire control system [26].

In the case of the PMSM control system, $i_{dref} = 0$, and the control signals i_{qref} , u_d , and u_q are synthesized based on the backstepping control strategy. Figure 5 presents the block

diagram of the PMSM sensorless control system based on the backstepping-type controller and a series of observers for speed, position, and load torque (based on an ESO-type observer), as well as an observer of the load resistance. These four observers are detailed in the next section.

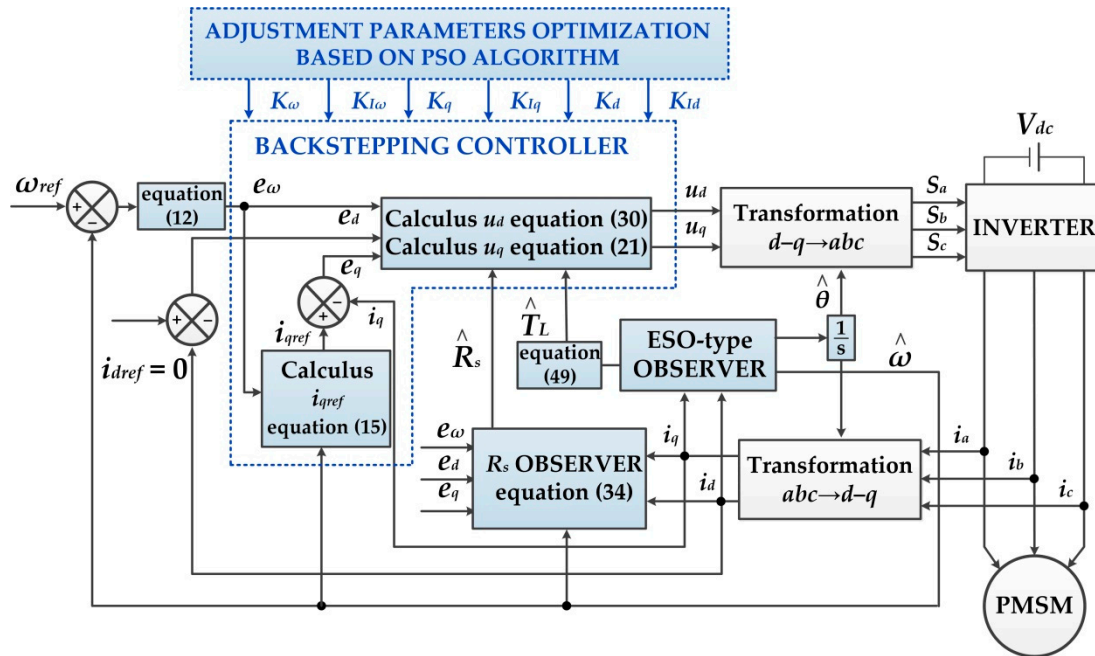


Figure 5. The proposed PMSM control system based on the backstepping-PSO controller.

Starting from the PMSM rotor speed control loop where the reference speed is ω_{ref} , we add the speed error signal of the following type:

$$e_{\omega} = \omega_{ref} - \omega + K_{I\omega} \int_0^t (\omega_{ref} - \omega) dt \quad (12)$$

In the equation of the speed error (12), we note the introduction of an integral term with the amplification factor $K_{I\omega}$, which is intended to ensure the convergence under the conditions of parametric uncertainties.

Next, the Lyapunov function $V_{\omega} = \frac{1}{2}e_{\omega}^2$ it is chosen and \dot{e}_{ω} it is calculated like in the following relation (13):

$$\dot{e}_{\omega} = \dot{\omega}_{ref} - \frac{n_p}{J} (L_d - L_q) i_d i_{qref} + \frac{B}{J} \omega - \frac{3}{2} \frac{n_p}{J} \lambda_0 i_{qref} + \frac{1}{J} T_L + K_{I\omega} (\omega_{ref} - \omega) \quad (13)$$

Using this relation, the derivative of the selected Lyapunov function it is obtained as indicated by the following expression:

$$\dot{V}_{\omega} = e_{\omega} \left[\dot{\omega}_{ref} - \frac{n_p}{J} (L_d - L_q) i_d i_{qref} + \frac{B}{J} \omega - \frac{3}{2} \frac{n_p}{J} \lambda_0 i_{qref} + \frac{1}{J} T_L + K_{I\omega} (\omega_{ref} - \omega) \right] \quad (14)$$

To ensure the stability of the subsystem considered, i_{qref} of the following form can be chosen from the relation (14):

$$i_{qref} = \frac{J}{\frac{3}{2} n_p \lambda_0 + n_p (L_d - L_q) i_d} \left[K_{\omega} e_{\omega} + \dot{\omega}_{ref} + \frac{B}{J} \omega + K_{I\omega} (\omega_{ref} - \omega) + \frac{1}{J} T_L \right] \quad (15)$$

By choosing i_{qref} given by the relation (15), we obtain $\dot{V}_\omega = -K_\omega e_\omega^2$, where $K_\omega > 0$. The block diagram of the synthesization of the control signal i_{qref} is presented in Figure 6.

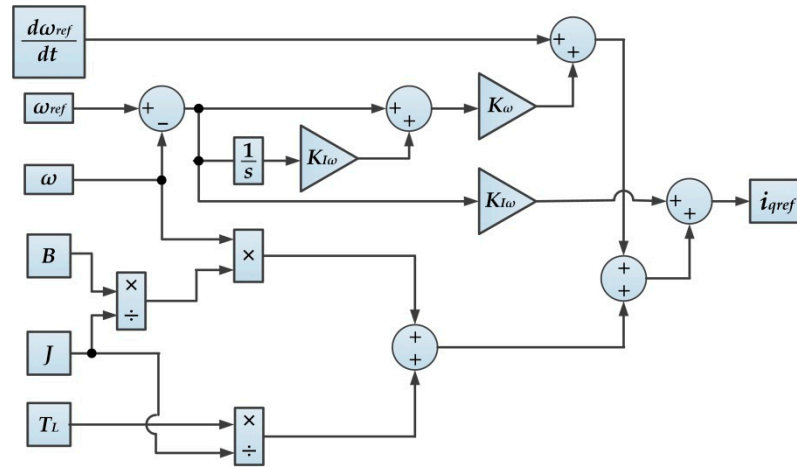


Figure 6. Block diagram of the synthesized control signal i_{qref} .

The backstepping algorithm continues with the stabilization of the current loop i_q . For this purpose, the following error is defined:

$$e_q = i_{qref} - i_q + e_{Iq} \quad (16)$$

where $e_{Iq} = K_{Iq} \int_0^t (i_{qref} - i_q) dt$ represents the integral action term.

Another Lyapunov function of the following form is considered:

$$V_q = V_\omega + \frac{1}{2} e_q^2 + \frac{1}{2} e_{Iq}^2 \quad (17)$$

By calculating the first derivative of V_q , the next relation can be written:

$$\dot{V}_q = -K_\omega e_\omega^2 + e_q \left[\frac{di_{qref}}{dt} - \frac{di_q}{dt} + K_{Iq}(i_{qref} - i_q) \right] + e_{Iq} K_{Iq}(i_{qref} - i_q) \quad (18)$$

In Equation (18), the following substitution can be made:

$$\frac{di_q}{dt} = -\frac{R_s}{L_q} i_q - \frac{L_d}{L_q} n_p \omega i_d - \frac{\lambda_0}{L_q} n_p \omega + \frac{1}{L_q} u_q \quad (19)$$

The 1st order derivative of V_q is computed as:

$$\begin{aligned} \dot{V}_q = & -K_\omega e_\omega^2 + e_q \left[\frac{di_{qref}}{dt} + \frac{R_s}{L_q} i_q + \frac{L_d}{L_q} n_p \omega i_d + \frac{\lambda_0}{L_q} n_p \omega - \frac{1}{L_q} u_q + K_{Iq}(i_{qref} - i_q) \right] \\ & + e_{Iq} K_{Iq}(i_{qref} - i_q) \end{aligned} \quad (20)$$

In the relation (20), the control signal u_q is synthesized as follows:

$$u_q = L_q \left[K_q e_q + \frac{L_d}{L_q} n_p \omega i_d + \frac{\lambda_0}{L_q} n_p \omega + \frac{R_s}{L_q} i_q + \frac{di_{qref}}{dt} \right] \quad (21)$$

The block diagram of the synthesization of the control signal u_q is presented in Figure 7.

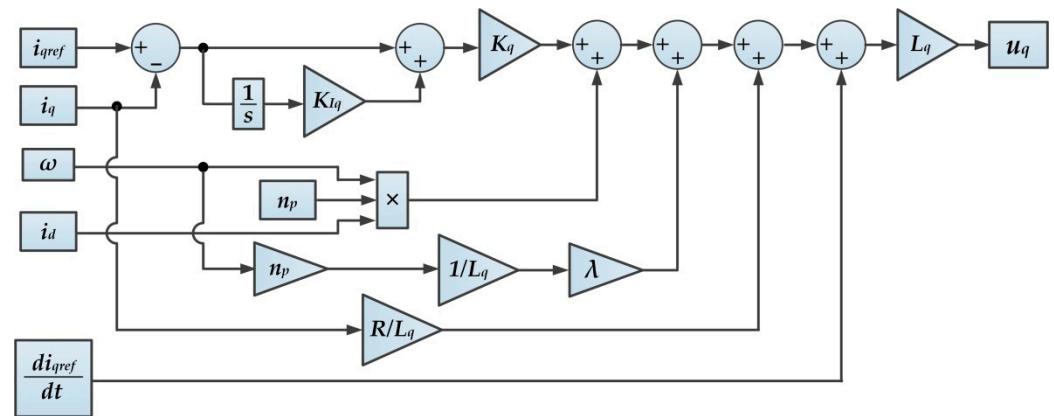


Figure 7. Block diagram of the synthesized control signal u_q .

Using the relation (21), the next relation is obtained:

$$\dot{V}_q = -K_\omega e_\omega^2 - K_q e_q^2 + (e_q + e_{Iq}) K_{Iq} (i_{qref} - i_q) \quad (22)$$

If in relation (22) the following substitution is made:

$$i_{qref} - i_q = e_q - e_{Iq} \quad (23)$$

we obtain the following relation:

$$\dot{V}_q = -K_\omega e_\omega^2 - K_q e_q^2 + (e_q + e_{Iq}) K_{Iq} (e_q - e_{Iq}) \quad (24)$$

where $\dot{V}_q \leq -\bar{K}_q V_q$, and $\bar{K}_q = \min\{K_\omega; (K_q + K_{Iq}); K_{Iq}\}$.

Based on this relation, we achieve the stability of the second subsystem.

The last loop that will be stabilized with the backstepping algorithm is the current loop i_d . $i_{dref} = 0$ is imposed and the control signal u_d will be synthesized. For this purpose, the following error is defined:

$$e_d = i_{dref} - i_d + e_{Id} \quad (25)$$

where $e_{Id} = K_{Id} \int_0^t (i_{dref} - i_d) dt$ represents the integral action term.

We choose the third Lyapunov function of the following form:

$$V_d = \frac{1}{2} e_d^2 + \frac{1}{2} e_{Id}^2 \quad (26)$$

Then, the derivative of function V_d is computed:

$$\dot{V}_d = e_d \left(-\frac{di_d}{dt} - K_{Id} i_d \right) + e_{Id} (-K_{Id} i_d) \quad (27)$$

If the next substitution is made in Equation (27):

$$i_d = e_{Id} - e_d \quad (28)$$

The first order derivative of V_d is obtained as follows:

$$\dot{V}_d = e_d \left(\frac{R_s}{L_d} i_d \frac{L_q}{L_d} n_p i_q - \frac{1}{L_d} u_d \right) + K_{Id} (e_d + e_{Id}) (e_d - e_{Id}) \quad (29)$$

In the relation (29), the control signal u_d is synthesized as follows:

$$u_d = L_d \left[K_d \left(i_{dref} - i_d + K_{Id} \int_0^t (i_{dref} - i_d) dt \right) + \frac{R_s}{L_d} i_d - \frac{L_q}{L_d} n_p i_q \omega \right] \quad (30)$$

The block diagram of the synthesization of the control signal u_d is presented in Figure 8.

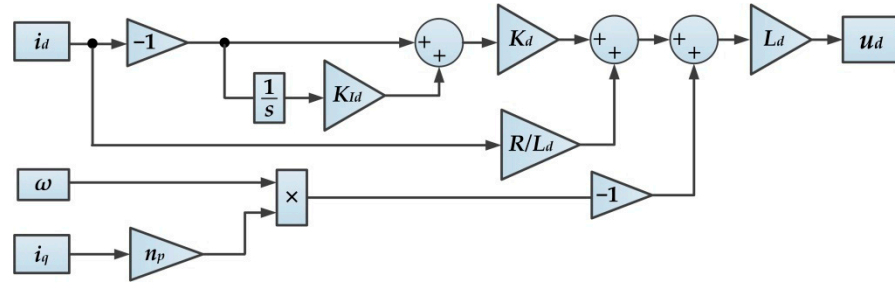


Figure 8. Block diagram of the synthesized control signal u_d .

If in relation (29), the control signal u_d given by the relation (30) is used, then the next relation is obtained:

$$\dot{V}_d = -(K_d + K_{Id})e_d^2 - K_{Id}e_{Id}^2 \quad (31)$$

For $\bar{K}_d = \min\{(K_d + K_{Id}); K_{Id}\}$, we obtain $\dot{V}_d \leq -\bar{K}_d V_d$. Hence, the stability of the third subsystem is achieved. Based on these relations, we achieve the global stabilization of the PMSM control system in which the requirements are fulfilled: $i_d \rightarrow i_{dref}$, $i_q \rightarrow i_{qref}$, and finally, $\omega \rightarrow \omega_{ref}$.

4. Resistance Estimation and Rotor Speed Estimation Based on the ESO-Type Observer

By using appropriate Lyapunov functions, an observer can be designed for the parameters of the PMSM control system, which has uncertainties. For example, the stator resistance of the PMSM significantly changes its value during operation. Thus, an observer of the resistance R_s will be designed.

Let us note $\tilde{R}_s = \hat{R}_s - R_s$, where \hat{R}_s represents the estimated value of R_s , and R_s represents the actual value (yet unknown) of the stator resistance of the PMSM.

Next, a Lyapunov function is chosen of the form [26]:

$$V_R = \frac{1}{2} \left(e_\omega^2 + e_q^2 + e_{Iq}^2 + e_d^2 + e_{Id}^2 + \frac{\tilde{R}_s^2}{\gamma} \right) \quad (32)$$

where $\gamma > 0$, and in the expressions $e_\omega^2, e_q^2, e_{Iq}^2, e_d^2, e_{Id}^2$ (presented in the previous section), the value of the stator resistance of the PMSM, R_s , is substituted by \hat{R}_s .

If $\tilde{R}_s = \hat{R}_s$, and we make the calculus for \dot{V}_R , the following form is obtained:

$$\dot{V}_R = \dot{V}_q + \dot{V}_d + \tilde{R}_s \left[\dot{\hat{R}}_s \frac{1}{\gamma} - \frac{1}{L_d} e_d i_d - \frac{1}{L_q} e_q i_q \right] \quad (33)$$

To obtain the negativity of \dot{V}_R , we choose:

$$\dot{\hat{R}}_s = -\gamma \left[\frac{1}{L_d} e_d i_d + \frac{1}{L_q} e_q i_q \right] \quad (34)$$

Thus, the dynamic expression of the observer for the stator resistance R_s is obtained. The block diagram for the estimation of R_s stator resistances is presented in Figure 9.

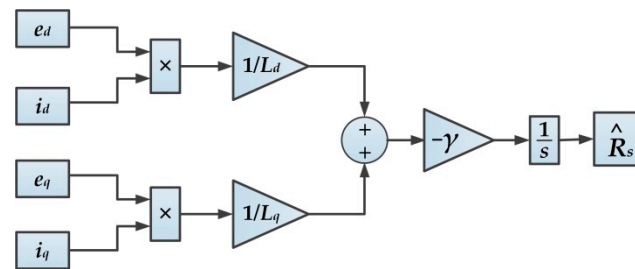


Figure 9. Block diagram for the estimation of R_s stator resistances.

To estimate the speed of the PMSM rotor and to exploit the nonlinear form of the equations used so far, an ESO-type observer is used. Its general description for a non-linear system of the form expressed by the relation (35) is presented below [24,25].

Let us consider the non-linear system with disturbances, with a single input and a single output, of the following form:

$$y^{(n)}(t) = f(y(t), \dot{y}(t), \dots, y^{(n-1)}(t), d(t), t) + bu(t) \quad (35)$$

where $y^{(l)}$ represents the l th derivative of the output y , u is the input, and d is the disturbance.

Let us note $x_1 = y$, $x_2 = \dot{y}$, \dots , $x_n = y^{(n-1)}$, and hence, the next system is obtained:

$$\begin{cases} \dot{x}_i = x_{i+1}, i = 1, \dots, n-1 \\ \dot{x}_n = f(x_1, x_2, \dots, x_n, d, t) + bu \end{cases} \quad (36)$$

An additional state of the system is chosen:

$$\begin{cases} x_{n+1} = f(x_1, x_2, \dots, x_n, d, t) \\ \dot{x}_{n+1} = h(t) \end{cases} \quad (37)$$

where:

$$h(t) = \dot{f}(x_1, x_2, \dots, x_n, d, t) \quad (38)$$

In the description above, the variable f represents lumped disturbances of the system.

For the non-linear system, together with the transformations described above given by the relations (35)–(38), we can design an ESO-type observer of the form:

$$\begin{cases} \dot{\hat{x}}_i = \hat{x}_{i+1} + \beta_i(y - \hat{x}_1), i = 1, \dots, n \\ \dot{\hat{x}}_{n+1} = \beta_{n+1}(y - \hat{x}_1) \end{cases} \quad (39)$$

Using minimum information, the ESO observer system described by the relation (39) will estimate the state variables of the initial system, as well as the term f -lumped disturbances.

By customizing Equations (35)–(39), to estimate the speed of the PMSM rotor, the design starts from Equation (40):

$$\frac{d\omega}{dt} = \frac{1}{J} \left(\frac{3}{2} n_p \lambda_0 i_q - T_L - B\omega \right) \quad (40)$$

where f is the generalized disturbance:

$$f = -\frac{B}{J}\omega - \frac{T_L}{J} \quad (41)$$

By replacing relation (41) in relation (40), the following equation is obtained:

$$\frac{d\omega}{dt} = f + bi_q \quad (42)$$

Let us note that $x_1 = \omega$, $x_2 = f$, and $\dot{f} = h$. Using these formulas, the expression of the angular speed system is obtained as a matrix:

$$\begin{cases} \dot{x} = Ax + Bu + Eh \\ y = Cx \end{cases} \quad (43)$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} b \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $E = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Based on these relations, the structure of the second order ESO-type observer can be written as follows [22,24–26]:

$$\begin{cases} \dot{x} = Ax + Bu + L(x_1 - z_1) \\ y = Cz \end{cases} \quad (44)$$

Let us consider the bandwidth of the observer ω_0 , and, by assigning the poles of the ESO system to $-\omega_0$, the observer gain L can be determined based on the equality of polynomials as follows:

$$\lambda(s) = s^2 + L_1s + L_2 = (s + \omega_0)^2 \quad (45)$$

$$\begin{cases} L_1 = 2\omega_0 \\ L_2 = \omega_0^2 \end{cases} \quad (46)$$

$$L = \begin{bmatrix} 2\omega_0 \\ \omega_0^2 \end{bmatrix} \quad (47)$$

When the ESO-type observer (ω_0 is chosen as equal to 900 for rapid convergence) is designed like this, as shown in the diagram in Figure 10, the state variable x_1 of the observer will estimate the PMSM rotor speed (Figure 11), and the state variable x_2 of the observer will estimate the term f , which represents the lumped disturbance.

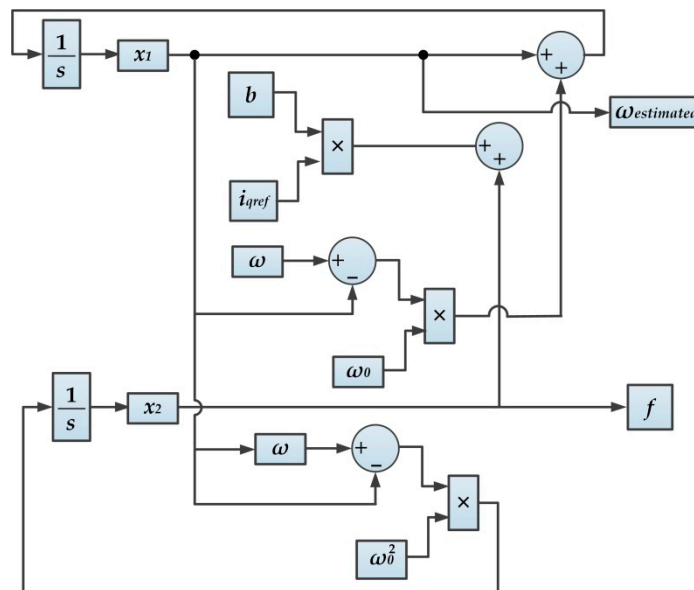


Figure 10. Block diagram for the ESO-type observer.

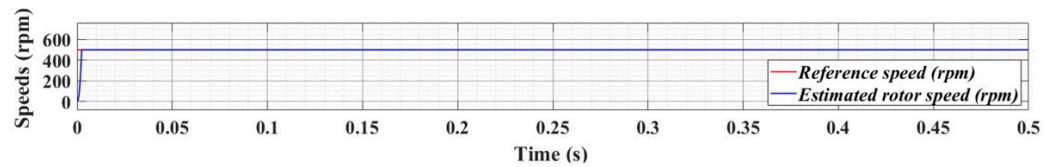


Figure 11. Time evolution of the PMSM rotor speed estimation.

Good results are noted with a negligible steady-state error and a response time of about 1 ms. The observer's behavior is also good in light of the fact that it has to respond faster than the global PMSM control system.

Using the rotor speed estimated by the ESO-type observer, the estimation of the angular position of the rotor is obtained as follows:

$$\hat{\theta}(t) = \theta(t_0) + \int_{t_0}^t \hat{\omega}(t) dt \quad (48)$$

where $\theta(t_0)$ is the initial position of the rotor in relation to a considered benchmark.

The implementation of Equation (48) is presented in Figure 12. Figure 13 shows the result of the numerical simulation of the PMSM rotor position estimation.

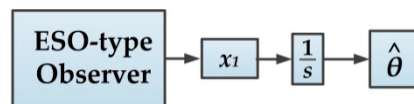


Figure 12. Block diagram of the PMSM rotor position estimation.

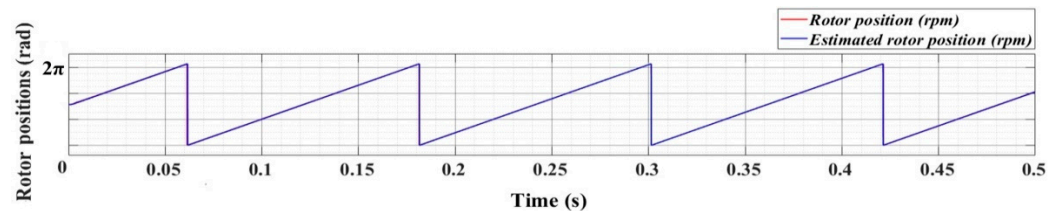


Figure 13. Time evolution of the rotor speed and of the estimated rotor speed of the PMSM.

Based on Equation (41), the estimation of the load torque is obtained as follows:

$$\hat{T}_L = J \left(-\frac{B}{J} x_1 - x_2 \right) \quad (49)$$

where x_1 and x_2 are the states of the ESO-type observer.

The implementation of Equation (49) is presented in Figure 14. The time evolution of the load torque and of its estimation is presented in Figure 15.

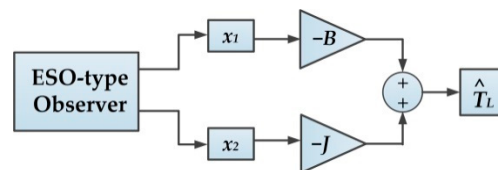


Figure 14. Block diagram of the estimation of the load torque T_L .

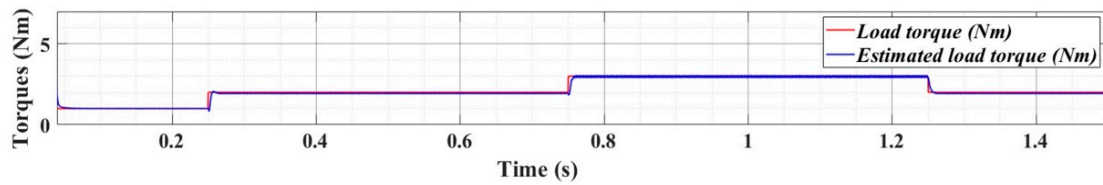


Figure 15. Time evolution of the load torque and of the estimated load torque T_L .

This section presented the numerical simulations for the observers of the stator resistance, speed, and position of the PMSM rotor and of the load torque. We note the stability of the observers, with good results in stationary and in dynamic regime. The short response time of the observers (approximately 1 ms) fulfills the condition for their use in the global PMSM control structure.

5. Numerical Simulations for the Sensorless Control of the PMSM

For a PMSM with the nominal parameters specified in Table 1, this section will present the results of the numerical simulations in which the controller is of the backstepping type presented in Section 3, and the position and speed observers of the PMSM rotor, the load torque, and the stator resistance are presented in Section 4.

Figure 16 presents the implementation in the MATLAB/Simulink development environment of the sensorless control of the PMSM based on the backstepping PSO controller and ESO-type observer. The main blocks implemented are the ones which synthesize the controls u_d , u_q , and i_{qref} , the observer for the estimation of the stator resistance, the ESO-type observer for the estimation of the PMSM rotor position and speed, and of the load torque. Furthermore, the blocks performing the Clarke–Park-type direct and reverse implementations, the reference speed and load torque generators, and the block of debugging, analysis and visualization signals are also implemented.

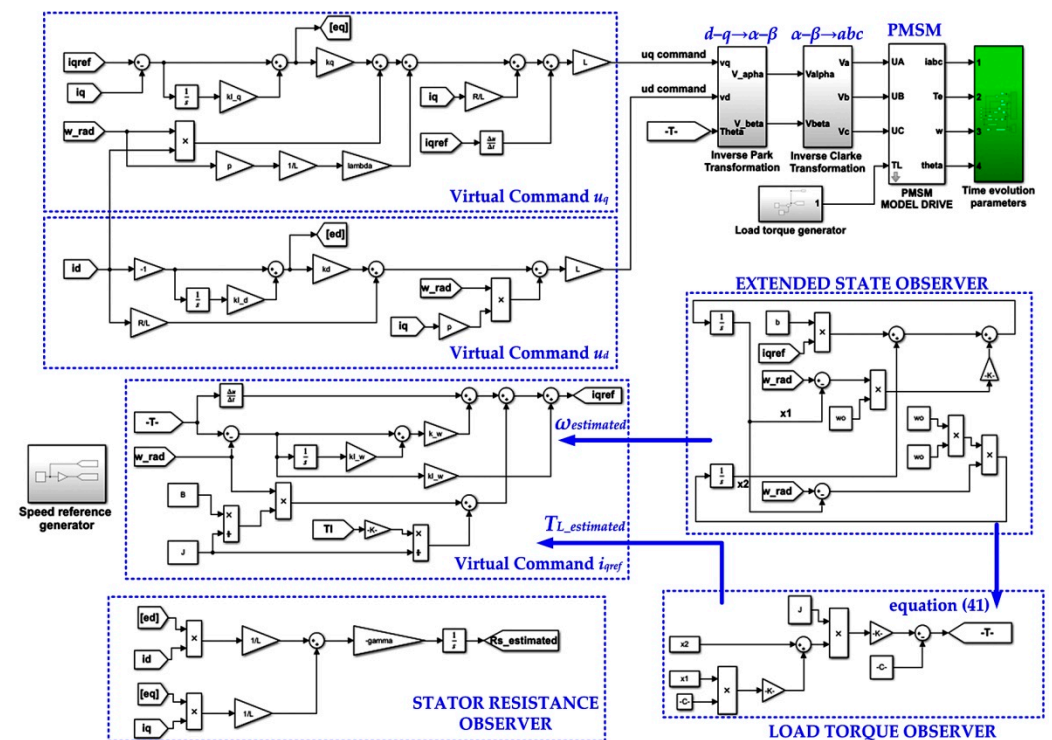


Figure 16. Block diagram for the implementation in MATLAB/Simulink of the sensorless control of the PMSM based on backstepping-PSO controller and ESO-type observer.

Based on the results obtained in Figure 4 at the end of Section 2, which show that, even when using a PI-type controller optimally tuned according to a PSO-type method, it can only provide good results for a relatively limited variation of the system parameters, and Figure 17 presents the results obtained when using a backstepping controller and an ESO-type observer, under the same conditions as in Figure 4, where $\omega_{ref} = 1600$ rpm, and there is a load torque change: $T_L = 1 \text{ Nm} \rightarrow 4 \text{ Nm}$ at time 0.25 s. We note a rise time of less than 2 ms and a steady-state error of less than 0.5%, which shows a qualitative improvement in the response of the PMSM control system. The parameters of the backstepping-type controller were tuned using a trial-and-error method. The values of the tuning parameters are: $K_\omega = 0.01$, $K_{I\omega} = 139$, $K_q = 2900$, $K_{Iq} = 150$, $\gamma = 0.01$, $K_d = 100$, and $K_{id} = 900$.

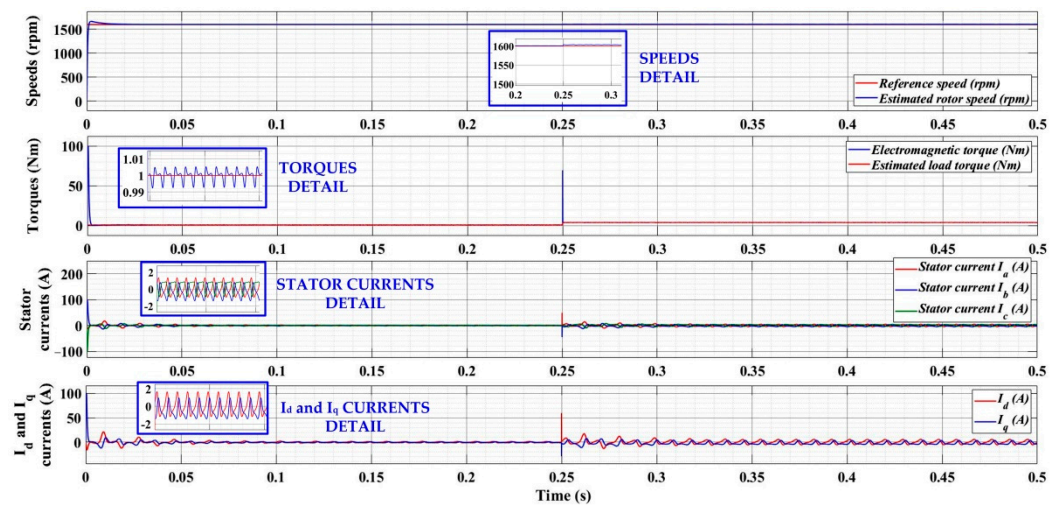


Figure 17. Time evolution for the numerical simulation of the PMSM control system based on the backstepping controller: $\omega_{ref} = 1600$ rpm, $T_L = 1 \text{ Nm} \rightarrow 4 \text{ Nm}$, $K_\omega = 0.01$, $K_{I\omega} = 139$, $K_q = 2900$, $K_{Iq} = 150$, $\gamma = 0.01$, $K_d = 100$, and $K_{id} = 900$.

To optimize the tuning parameters of the backstepping-type controller, a PSO optimization method is used, and the following optimal parameter values are obtained: $K_\omega = 0.01$, $K_{I\omega} = 140$, $K_q = 5000$, $K_{Iq} = 50$, $\gamma = 0.01$, $K_d = 300$, and $K_{id} = 300$.

In Figure 18, we note an improvement in the response, in the sense that the steady-state error is almost null, even under the conditions of sudden variation of the load torque from 1 Nm to 4 Nm at time 0.25 s.

To demonstrate the parametric robustness of the sensorless PMSM control system based on the backstepping-type controller and ESO-type observer, the following numerical simulations in which the backstepping-type control is optimally tuned using the PSO, show the response of the control system when ω_{ref} reaches 3200 rpm, the load torque varies up to 8 Nm to which a 0.2 Nm uniform random noise is added, and the load resistance and the moment of inertia J vary by up to 100% of the nominal value. Thus, in Figure 19, for $\omega_{ref} = 1600$ rpm and $T_L = 1 \text{ Nm}$, among the special performances of the control system, we note a response time of approximately 1.5 ms, a rise time of 1 ms, zero steady-state error and neglected overshoot (0.25%). We also note the same special performances in stationary and transitory regime in Figure 20, where there is a doubling of the reference speed: $\omega_{ref} = 3200$ rpm and $T_L = 0.1 \text{ Nm}$.

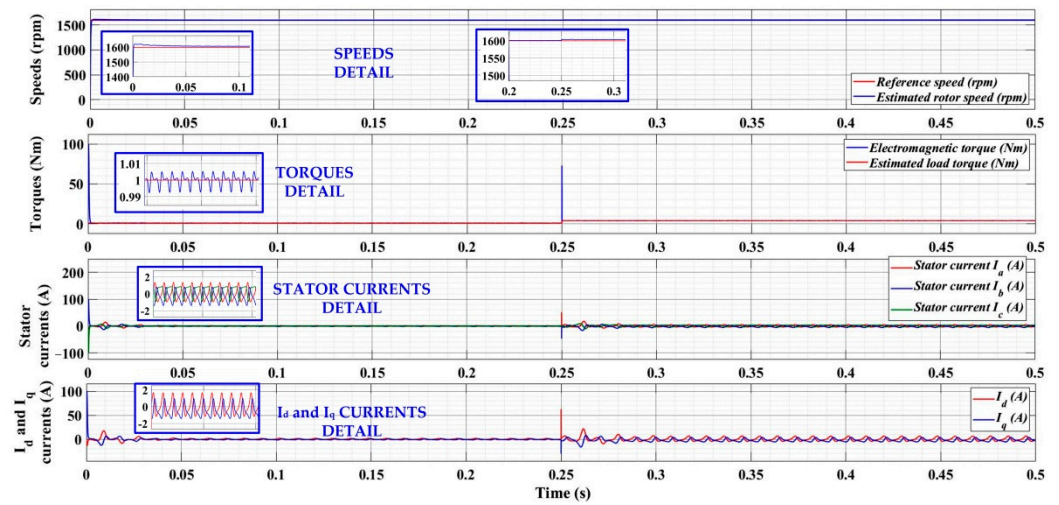


Figure 18. Time evolution for the numerical simulation of the PMSM control system based on the backstepping-PSO controller: $\omega_{ref} = 1600$ rpm, $T_L = 1 \text{ Nm} \rightarrow 4 \text{ Nm}$, $K_\omega = 0.01$, $K_{I\omega} = 140$, $K_q = 5000$, $K_{Iq} = 50$, $\gamma = 0.01$, $K_d = 300$, and $K_{id} = 300$.

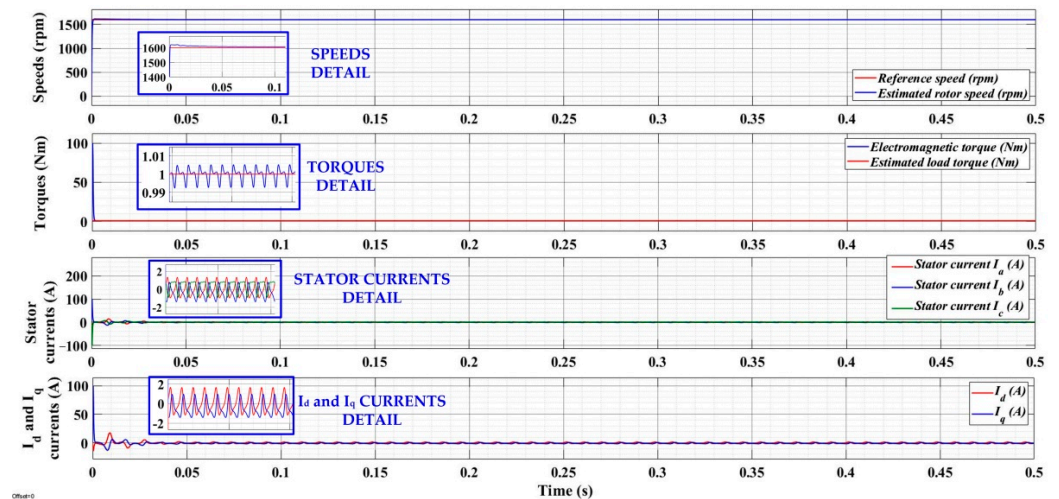


Figure 19. Time evolution for the numerical simulation of the PMSM control system based on the backstepping-PSO controller: $\omega_{ref} = 1600$ rpm, $T_L = 1 \text{ Nm}$, $K_\omega = 0.01$, $K_{I\omega} = 140$, $K_q = 5000$, $K_{Iq} = 50$, $\gamma = 0.01$, $K_d = 300$, and $K_{id} = 300$.

Figure 21 shows the response of the sensorless PMSM control system where $\omega_{ref} = 1600$ rpm and $T_L = 8 \text{ Nm}$. We note that even in this case, where the nominal load torque has a double value, good control performance is achieved. In Figures 22 and 23, under the same conditions as those in Figure 21, but to which a 0.2 Nm uniform random noise for the load torque is added, and in which the moment of inertia J has an increase of 100%, and in Figure 23, the stator resistance of the PMSM additionally has an increase of 100%, good performance is achieved in stationary and in dynamic regime, with a proportional increase in the starting current due to mechanical parametric variations.

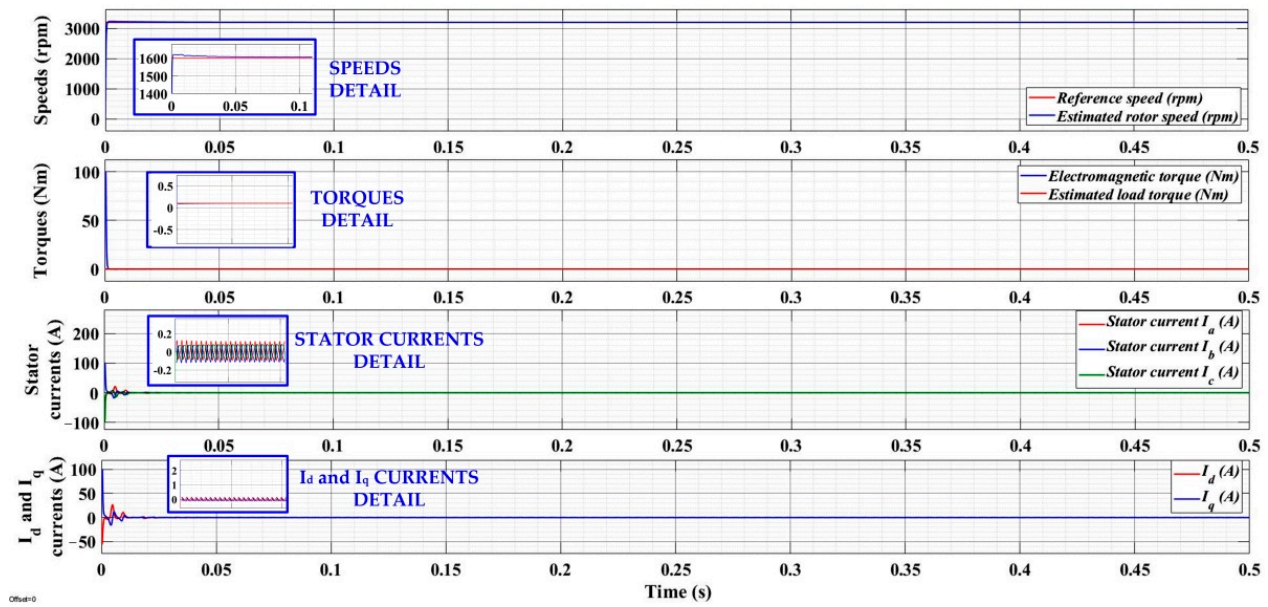


Figure 20. Time evolution for the numerical simulation of the PMSM control system based on the backstepping-PSO controller: $\omega_{ref} = 3200$ rpm, $T_L = 1$ Nm, $K_\omega = 0.01$, $K_{I\omega} = 140$, $K_q = 5000$, $K_{Iq} = 150$, $\gamma = 0.01$, $K_d = 300$, and $K_{id} = 300$.

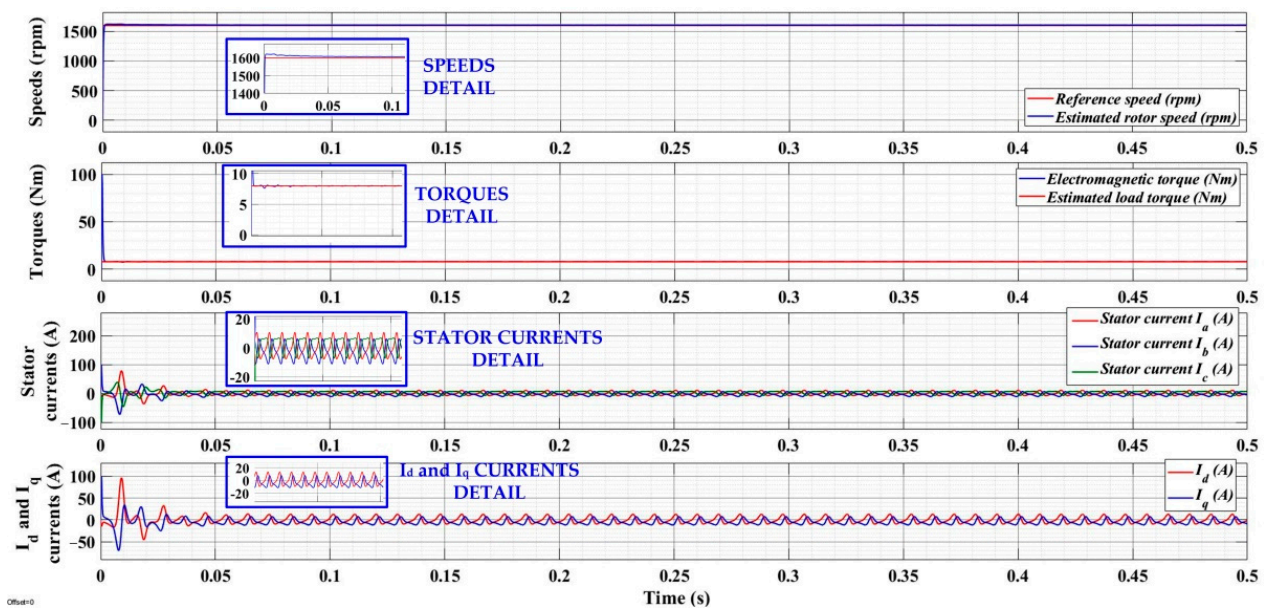


Figure 21. Time evolution for the numerical simulation of the PMSM control system based on the backstepping-PSO controller: $\omega_{ref} = 1600$ rpm, $T_L = 8$ Nm, $K_\omega = 0.01$, $K_{I\omega} = 140$, $K_q = 5000$, $K_{Iq} = 50$, $\gamma = 0.01$, $K_d = 300$, and $K_{id} = 300$.

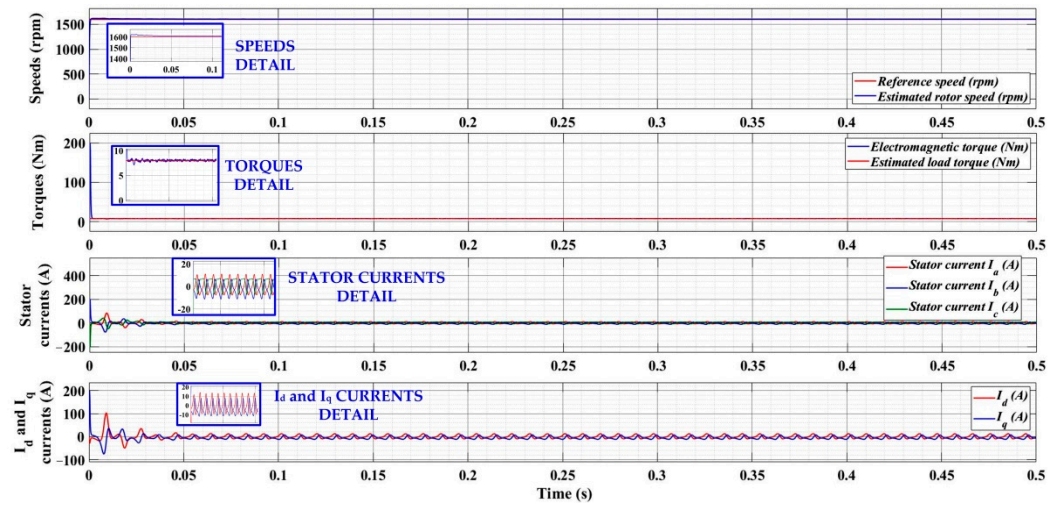


Figure 22. Time evolution for the numerical simulation of the PMSM control system based on the backstepping-PSO controller: $\omega_{ref} = 1600$ rpm, $T_L = 8$ Nm with a uniform random noise, $K_\omega = 0.01$, $K_{I\omega} = 140$, $K_q = 5000$, $K_{Iq} = 50$, $\gamma = 0.01$, $K_d = 300$, $K_{id} = 300$, and 100% increase in J parameter.

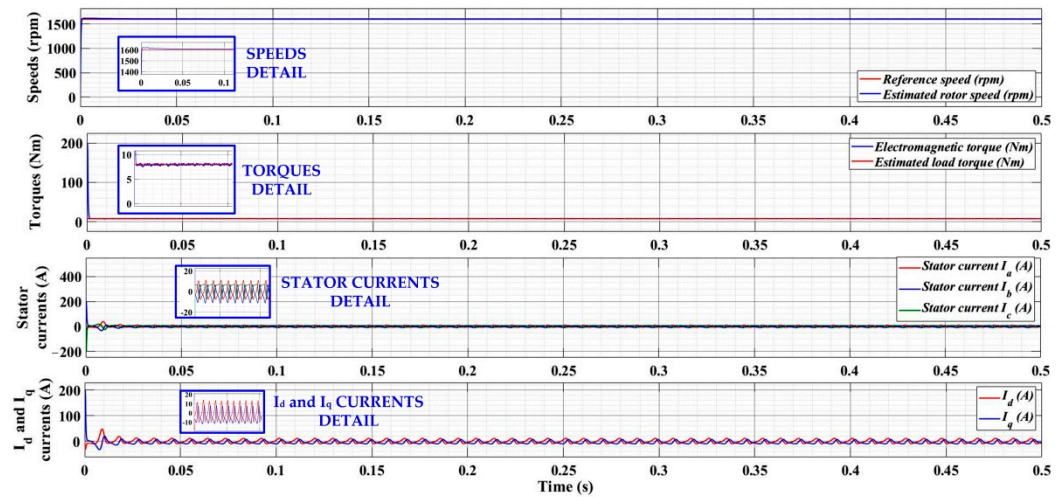


Figure 23. Time evolution for the numerical simulation of the PMSM control system based on the backstepping-PSO controller: $\omega_{ref} = 1600$ rpm, $T_L = 8$ Nm with a uniform random noise, $K_\omega = 0.01$, $K_{I\omega} = 140$, $K_q = 5000$, $K_{Iq} = 50$, $\gamma = 0.01$, $K_d = 300$, $K_{id} = 300$, and 100% increase in J parameter and stator resistance R_s .

6. PIL and HIL Stages for Real-Time Implementation and Experimental Results

In the stages of development of a real-time control application, we note the conceptualization and formalization of the problem by phenomenological description using the differential equations, the continuation of the study by following a particular formalism of the control theory, and the synthesis of the controller in the form of equations, followed by several series of numerical simulations performed using specialized simulation environments to test the entire system without resorting to hardware elements (except the PC which runs the numeric simulations). These steps fall within the SIL paradigm and are summarized in Sections 2–5.

The MATLAB/Simulink development environment is used for the real-time implementation of the PMSM sensorless control application using the Motor Control Blockset (MCB) and the Embedded Coder Support Package for TI C2000 Processors toolboxes. The MCB toolbox offers the possibility to implement the FOC control structure by means of specialized blocks for space-vector generator, the Clarke and Park transformations. The code of the PMSM control algorithm compatible with TI C2000 Processors, in particular,

with the TMS320F28379D microcontroller, is generated and executed by means of the Embedded Coder Support Package. This package contains a series of block libraries for on-chip and on-board peripherals, digital I/O blocks, ePWM blocks, SPI blocks, blocks for the connection with the PC host, DMC and IQMath libraries to perform code optimization, as well as PIL or HIL block for code generation, validation, testing, or execution of the algorithms developed using a real PMSM [30].

The real-time implementation of the proposed control structure is based on the high performance of the dual-core architecture of the TMS320F28379D microcontroller. Among the technical characteristics of this dual-core microcontroller, we can mention: General-Purpose Input/Output (GPIO); analog subsystems (Analog-to-Digital Converter (ADC) with 12-bit/16-bit resolution, Subsystem Comparator (CMPSS), Buffered Digital-to-Analog Converter (DAC)); control peripherals (16-bit resolution Enhanced Pulse Width Modulator (ePWM) modules, Enhanced Capture (eCAP) modules, Enhanced Quadrature Encoder Pulse (eQEP) modules, etc.); memory unit with four different block modules (dedicated RAM blocks, shared local RAM blocks, shared global RAM blocks, message RAM blocks, 1 MB Flash memory, one-time programmable (OTP) memory); communication peripherals (Serial Peripheral Interface (SPI) module, four Serial Communication, Interface (SCI) module, Universal Serial Bus (USB) module, etc.); Inter-Processor Communications (IPC) which performs the communication between the two processors; and Control Law Accelerator (CLA) processor with a 32-bit floating-point architecture performing advanced mathematical calculations in parallel with the main processor on 32-bit fixed-point architecture with a frequency of 200 MHz.

6.1. PIL Stage

In the stages of development of a real-time control application, we note the conceptualization and formalization of the problem by phenomenological description using the differential equations, the continuation of the study by following a particular formalism of the control theory, the design of the controller in the form of equations, followed by several series of numerical simulations using specialized simulation environments to test the entire system without resorting to hardware elements (except the PC, which performs the numeric simulations). These stages were designed in the previous sections and fall within the SIL paradigm. The next testing step is to fit the system testing into the PIL paradigm. For this purpose, the application software will be downloaded into a development board, but, in order to avoid any problems caused by potential errors that could lead to possible damage to the driven object (in our case the PMSM) by failing to properly set some limitations of the software/hardware protections, the motor is software simulated on the PC host. Figure 24 shows a general diagram of the PIL paradigm into which the application presented is embedded.

The main tasks performed in the PIL test stage consist of:

- Checking the execution code by comparing the algorithms of the SIL stage with those used for the download to the hardware target module;
- Determining the execution times of the main routines in the algorithm for the implementation in the hardware target model, for the purpose of using the application.
- The main tasks described above are performed by several steps as described below:
- Selecting the hardware settings panel;
- Selecting the development board (TI Delphino F28379D LaunchPad, Texas Instruments, Dallas, USA) and choosing the build, load and run options for the tested software application subsystem of the global control of the PMSM (Figure 25);
- Configuring the main hardware elements: SCI and External mode (serial communication);
- Downloading the PMSM control algorithm subsystem of the main software application in C/C++ format compatible with Code Composer Studio (CCS) (Figures 26 and 28) to the target hardware model-TMS320F28379D;
- The control algorithm testing is performed in the MATLAB, (MathWorks, Natick, Massachusetts, USA) SIL/PIL manager application (Figure 27).

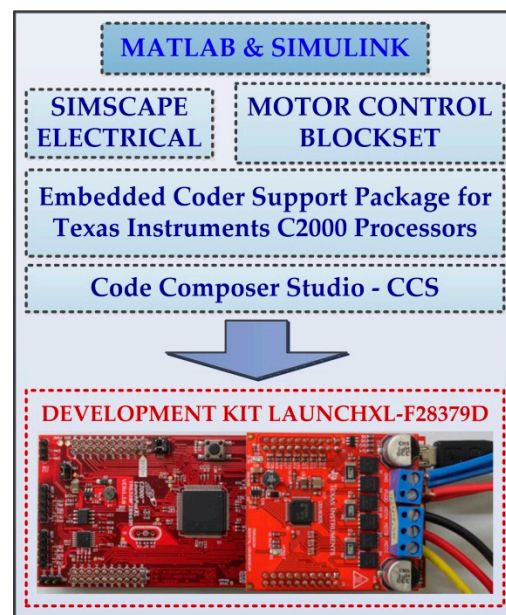


Figure 24. Development and testing architecture for the control of the PMSM application—PIL stage.

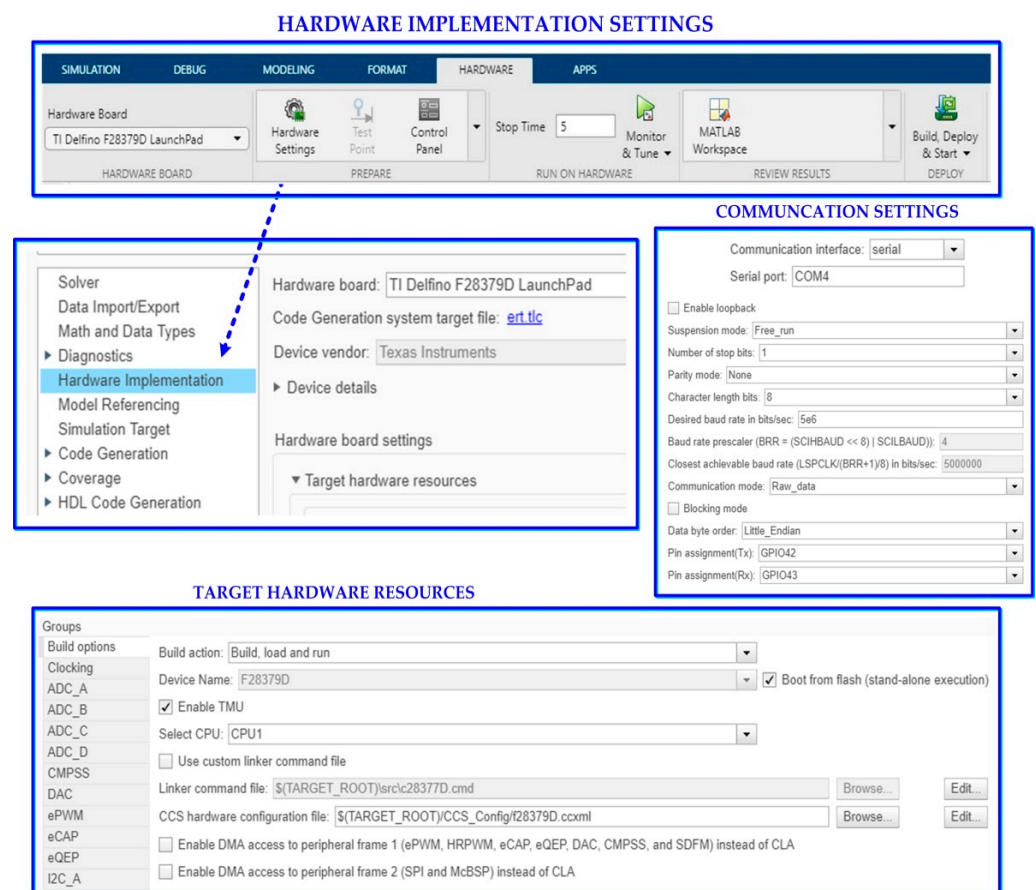


Figure 25. Hardware implementation settings for the PIL stage of the PMSM control system.

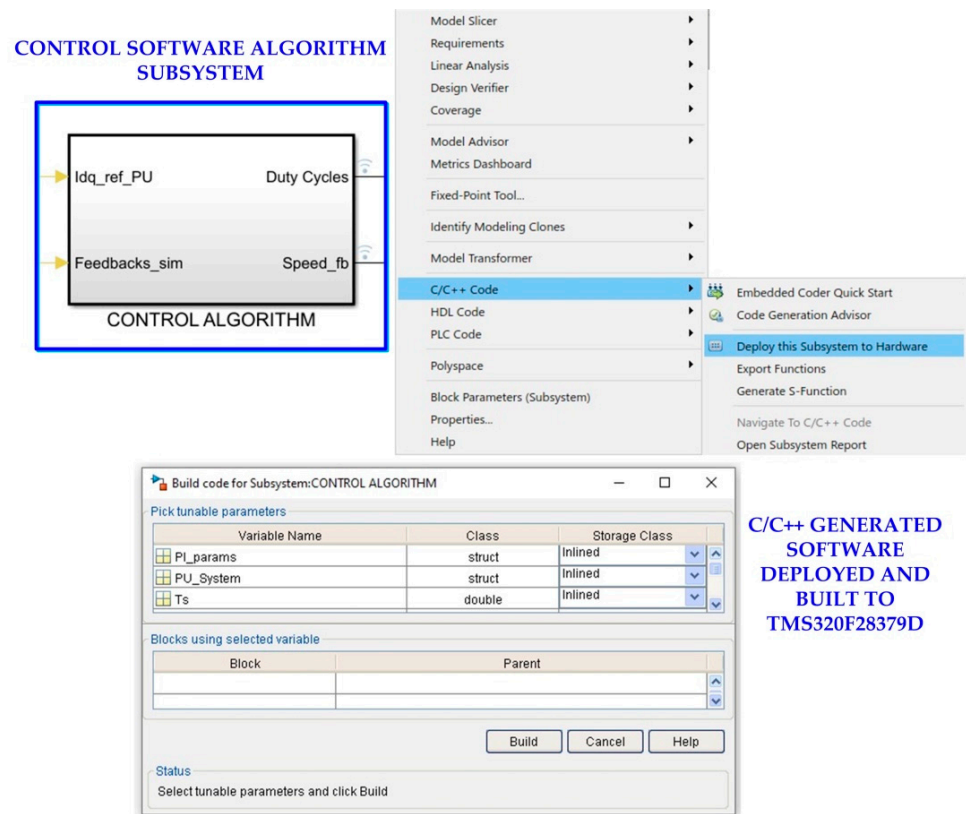


Figure 26. Generate and build code of the control software algorithm subsystem.

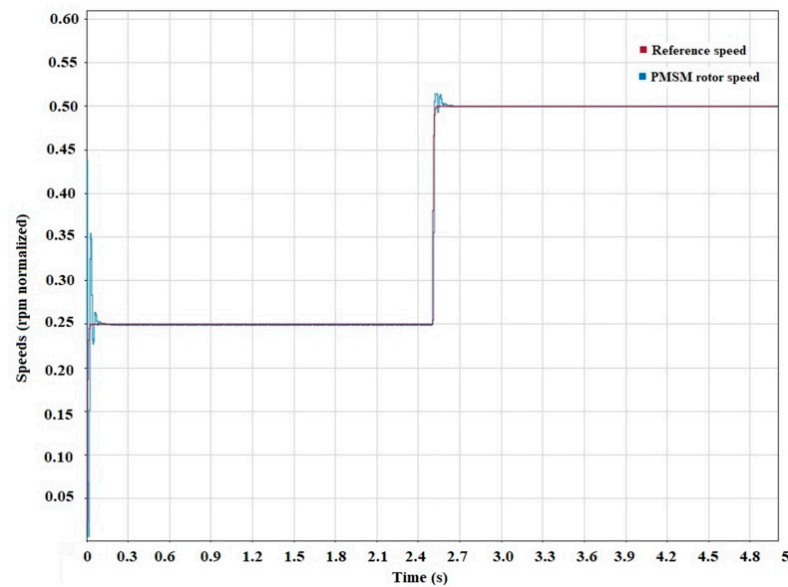


Figure 27. Time evolution of the PMSM rotor speed—PIL stage.



Figure 28. SIL/PIL manager settings for the control algorithm testing.

6.2. HIL Stage

The HIL paradigm is achieved by including a real PMSM in the presented assembly. The complete elements for the hardware/software testing are presented in Figure 29.

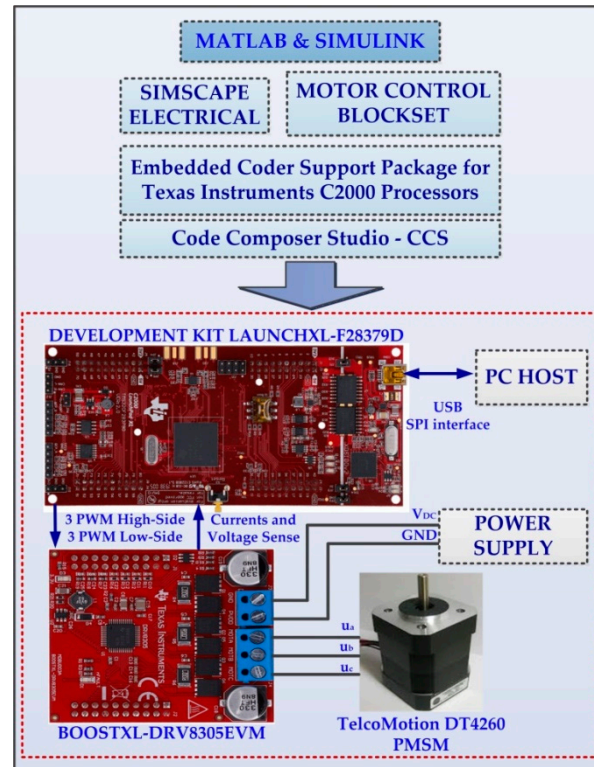


Figure 29. Development and testing architecture for the control of the PMSM application—HIL stage.

We specify that the software program of the PMSM control system used for the numerical simulations is rewritten with blocks from the MCB and Embedded Coder Support Package for TI C2000 Processors toolboxes, and the continuous integral block $\frac{1}{s}$ is replaced with a discrete integral block $\frac{kT_s}{z-1}$, where T_s is the sampling time, k is the current step, and s and z are complex, continuous, and discrete variables.

The general application program implemented in MATLAB/Simulink using software blocks from MCB toolbox is presented in Figure 30. There is an analogy with the implementation in MATLAB/Simulink in the SIL stage; the main difference is the addition of the Open-Loop-Control block, which implements the starting features of a PMSM.

In Figure 31, we note that the PMSM will be open-loop controlled until an imposed fraction of the rated speed is reached. During this time, the Position Generator block is used, and it generates a ramp-type position signal with a frequency equal to the frequency of the reference voltage. Further, the value of current i_q is maintained at a constant preset value (2 A for these applications).

Figure 32 shows the implementation in MATLAB/Simulink of the conversion of the signals received from the inverter control into PWM signals that will form the power-supply voltages of the PMSM.

The build stage translates the application program built with Simulink blocks, MCB and Embedded Coder Support Package for TI C2000 Processors toolboxes into executable code compatible with CCS code (C/C++). The deploy stage achieves the transfer of the executable code to TMS320F28379D MCU, and at this moment, it is also possible to start the PMSM control software application (Figure 33).

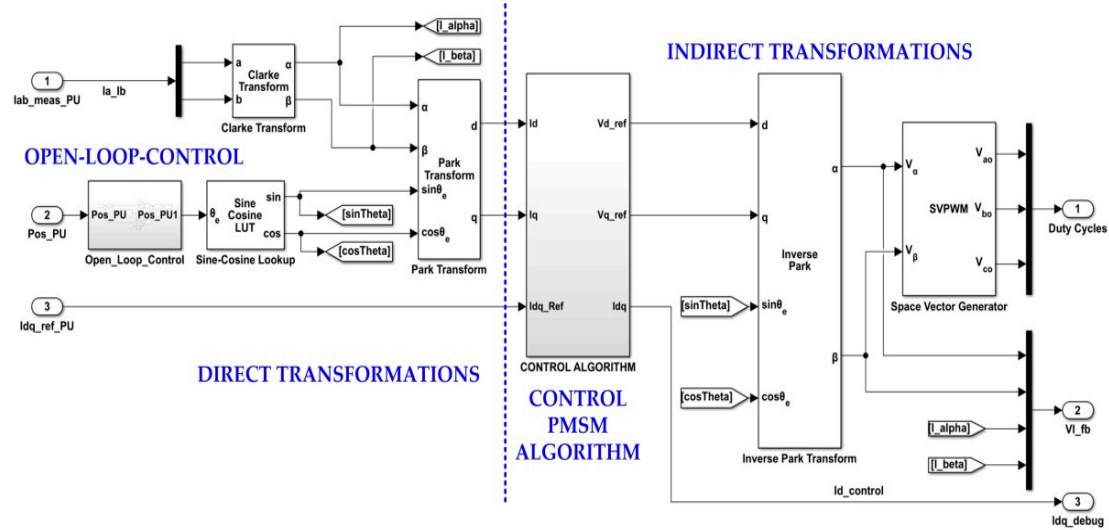


Figure 30. Global real-time software application implementation in MATLAB/Simulink for PMSM control—HIL stage.

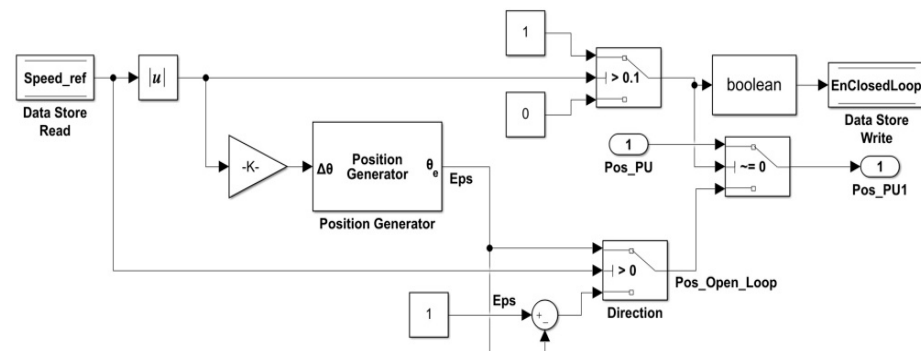


Figure 31. MATLAB/Simulink block diagram subsystem for open loop for real-time implementation—HIL stage.

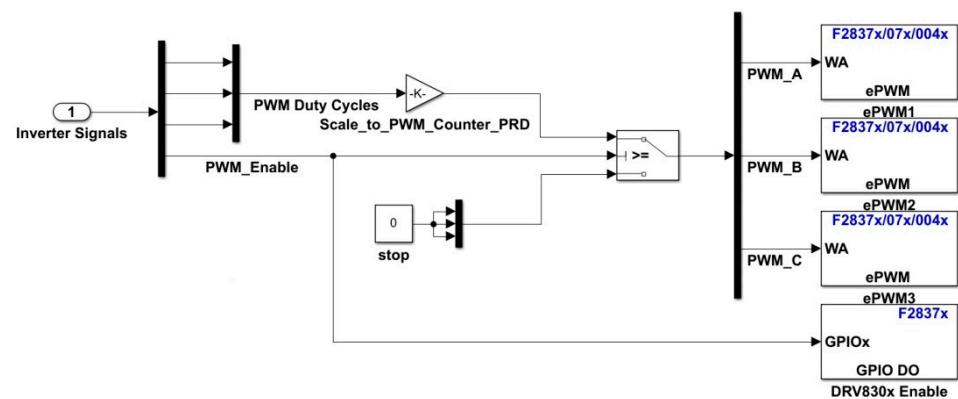


Figure 32. MATLAB/Simulink block diagram subsystem for PWM signals generation—HIL stage.

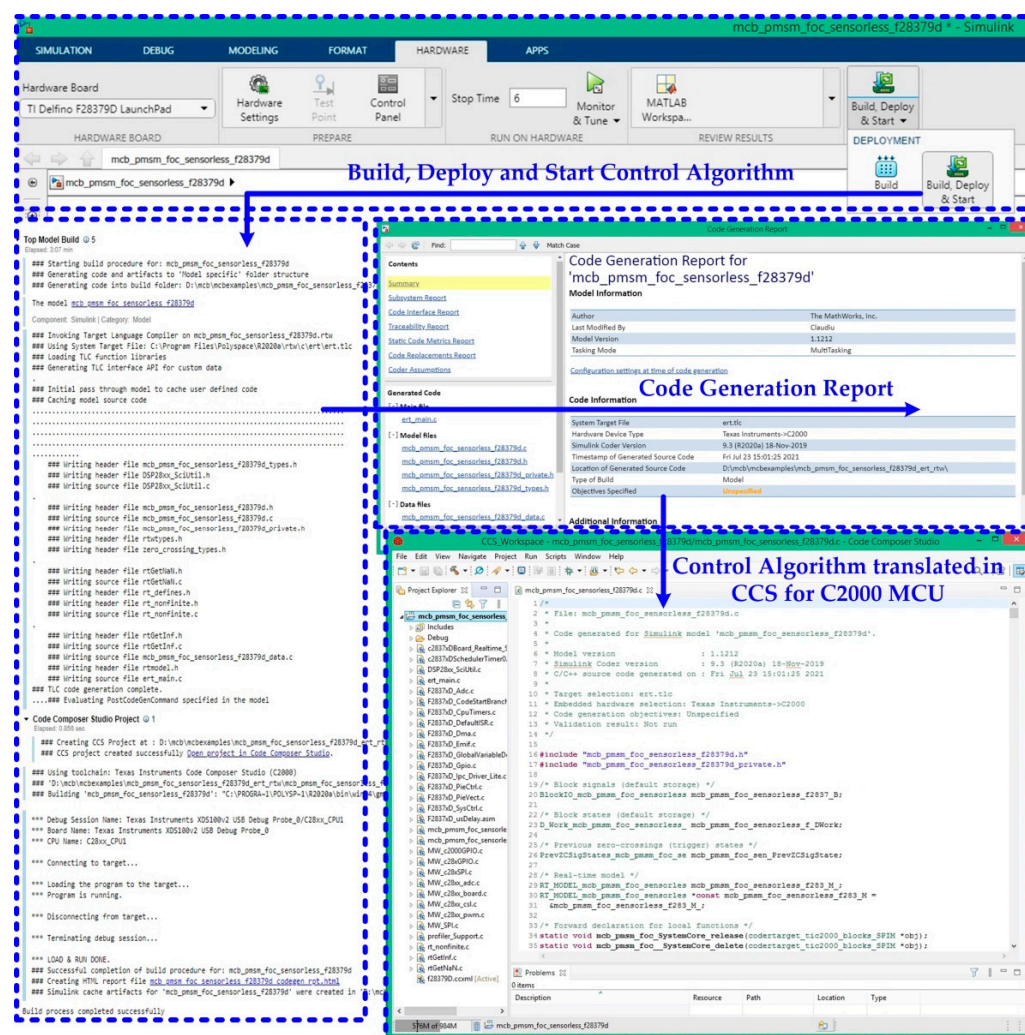


Figure 33. Build, deploy, and start of the PMSM control algorithm in TMS320F28379D MCU.

Figure 34 shows the oscilloscope image of a PWM signal.

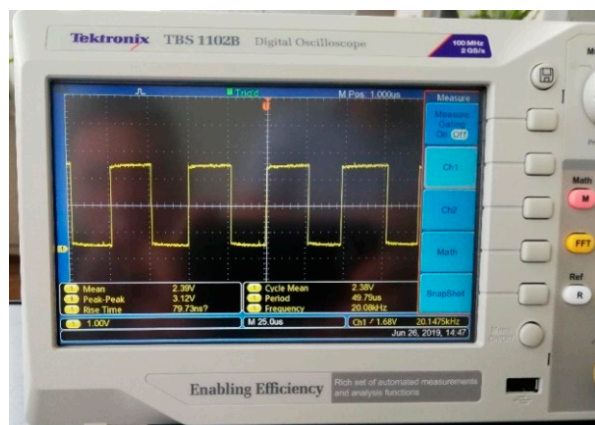


Figure 34. PWM signal waveform.

The software interface developed in MATLAB/Simulink for the monitoring and control of the PMSM sensorless control system in view of performing the analysis and time evolution of the quantities of interest is presented in Figure 35.

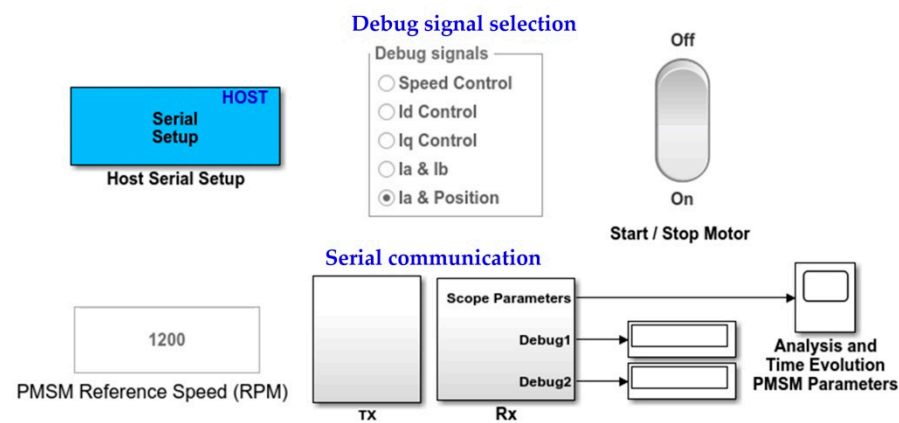


Figure 35. MATLAB/Simulink software interface for the monitoring and control application on the Host PC.

6.3. Experimental Results

This subsection will present the experimental results, in the sense of real-time implementation of the algorithms presented in the previous sections. Thus, in the first part, we present the implementation of the backstepping algorithm according to the stages presented above and we will demonstrate the similarity of the evolution of the PMSM control system in the SIL and HIL implementations. In the second part of this subsection, we present a comparison of the implementation of HIL, of the backstepping-type, and PI-type algorithms optimized by using a PSO-type method. The superiority of the backstepping algorithm for the PMSM control will be demonstrated by comparison in terms of the response time, the override, the steady-state error, the speed ripple and the improvement of the THD of the stator currents. The nominal parameters of the PMSM used in the SIL and HIL implementations are those presented in Table 1.

After completing the testing steps in HIL stage, Figures 36–40 present real-time evolutions of the estimations of the PMSM rotor speed and position, but also of currents i_a and i_b , current i_q , and of the electromagnetic torque T_e . The response of the PMSM control system is presented for a variation of the speed reference ω_{ref} defined by the sequence: [0 10 20 30 40] s → [1200 1600 3200 1600 1200] rpm and a load torque with a variation around 1 Nm.

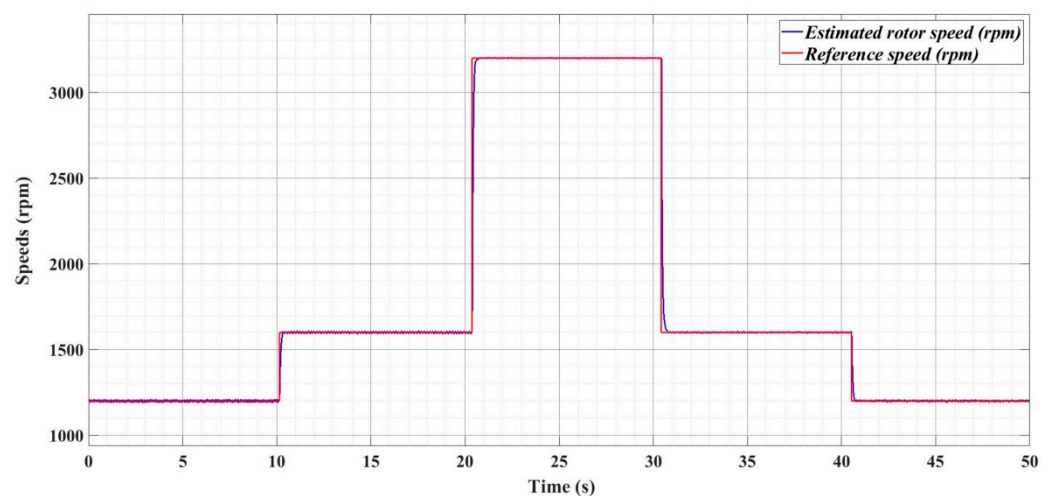


Figure 36. Real-time evolution for the estimated PMSM rotor speed—HIL stage; $T_L = 1$ Nm.

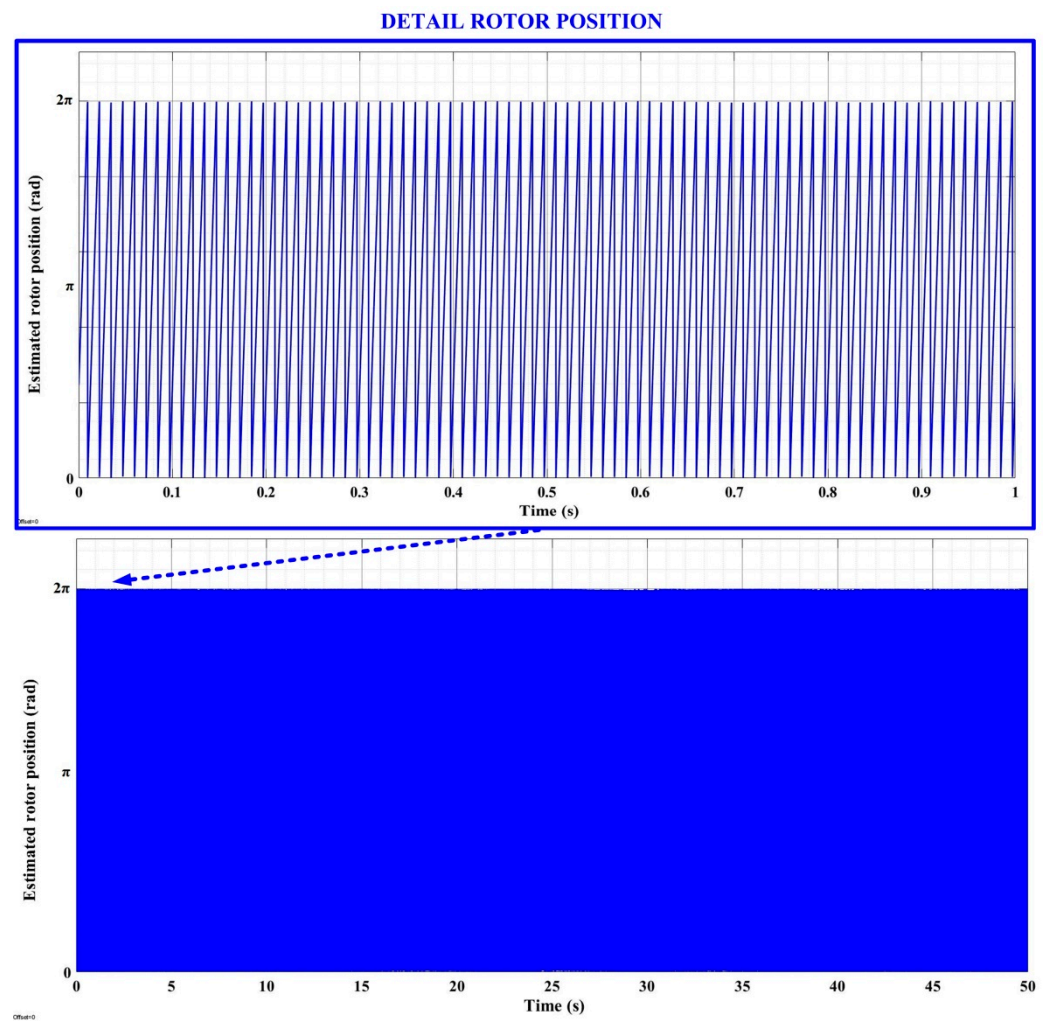


Figure 37. Real-time evolution for the estimated PMSM rotor position—HIL stage; $T_L = 1$ Nm.

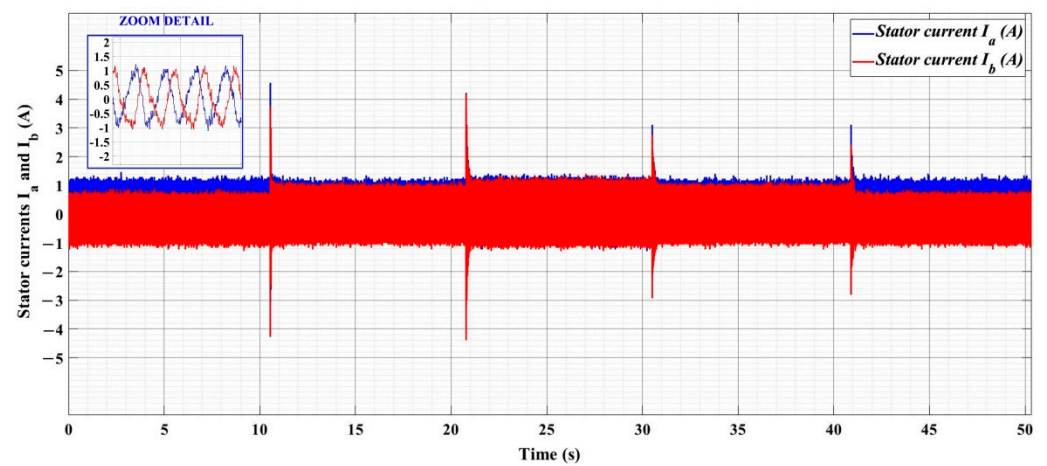


Figure 38. Real-time evolution for i_a and i_b stator currents—HIL stage; $T_L = 1$ Nm.

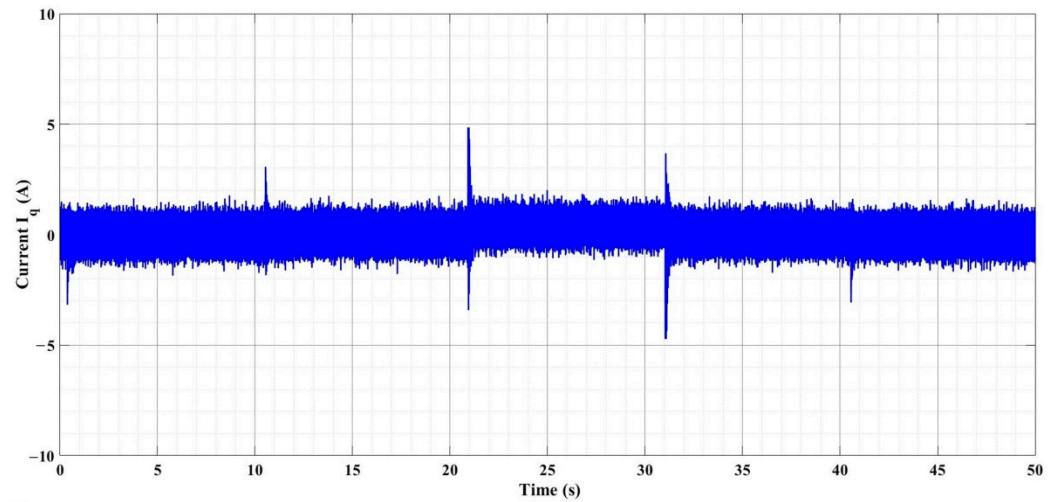


Figure 39. Real-time evolution for i_q current—HIL stage; $T_L = 1$ Nm.

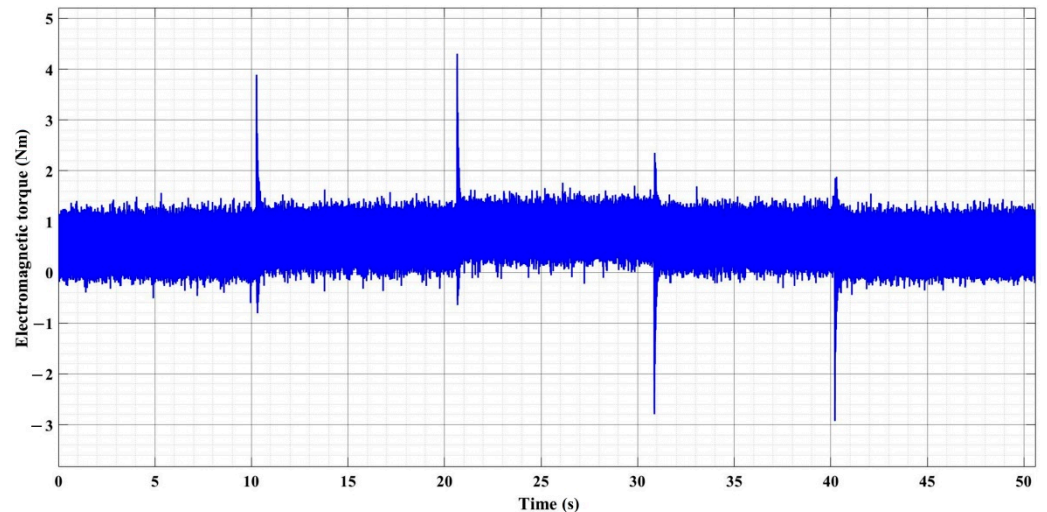


Figure 40. Real-time evolution for the electromagnetic torque of the PMSM—HIL stage; $T_L = 1$ Nm.

Figure 37 presents the real-time evolution for the estimated PMSM rotor position in HIL stage with a detail of the rotor speed positions at 1200 rpm of the rotor speed.

Figure 41 shows, in the HIL stage, the evolution of the electromagnetic torque, and of the estimated load torque for a reference of the PMSM rotor speed $\omega_{ref} = 1600$ rpm.

We specify that the sampling period in the numerical simulations in the previous sections is of 0.001 ms. The inner current loop execution speed in the hardware implementation is of 0.05 ms, and the speed loop execution speed is of 0.5 ms. The speed of communication between the PC host and the hardware development platform ensures the real-time viewing of items by a sampling period of 1 ms. Therefore, Figure 42 shows the variation of current i_q over an interval of 50 ms (under the same SIL and HIL test conditions, $\omega_{ref} = 1600$ rpm and $T_L = 1$ Nm) achieved by numerical simulation (SIL) and retrieved from the software interface for the monitoring and control application (HIL). It can be noted that there is a similarity between the evolutions of current i_q in SIL and HIL implementations, noting that 1000 times more samples are presented in the representation obtained by numerical simulation per time unit.

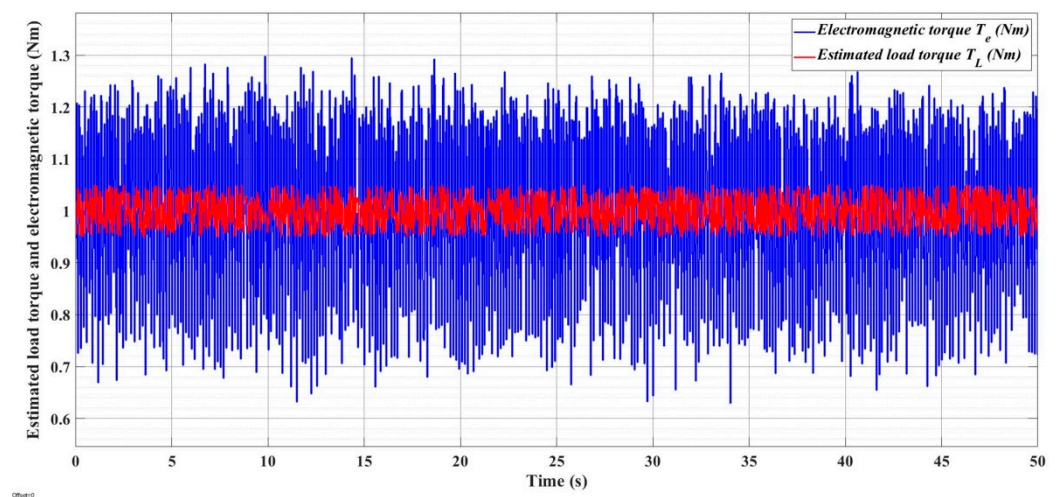


Figure 41. Real-time evolution for the estimate load torque and electromagnetic torque of PMSM—HIL stage; $\omega_{ref} = 1600$ rpm and $T_L = 1$ Nm.

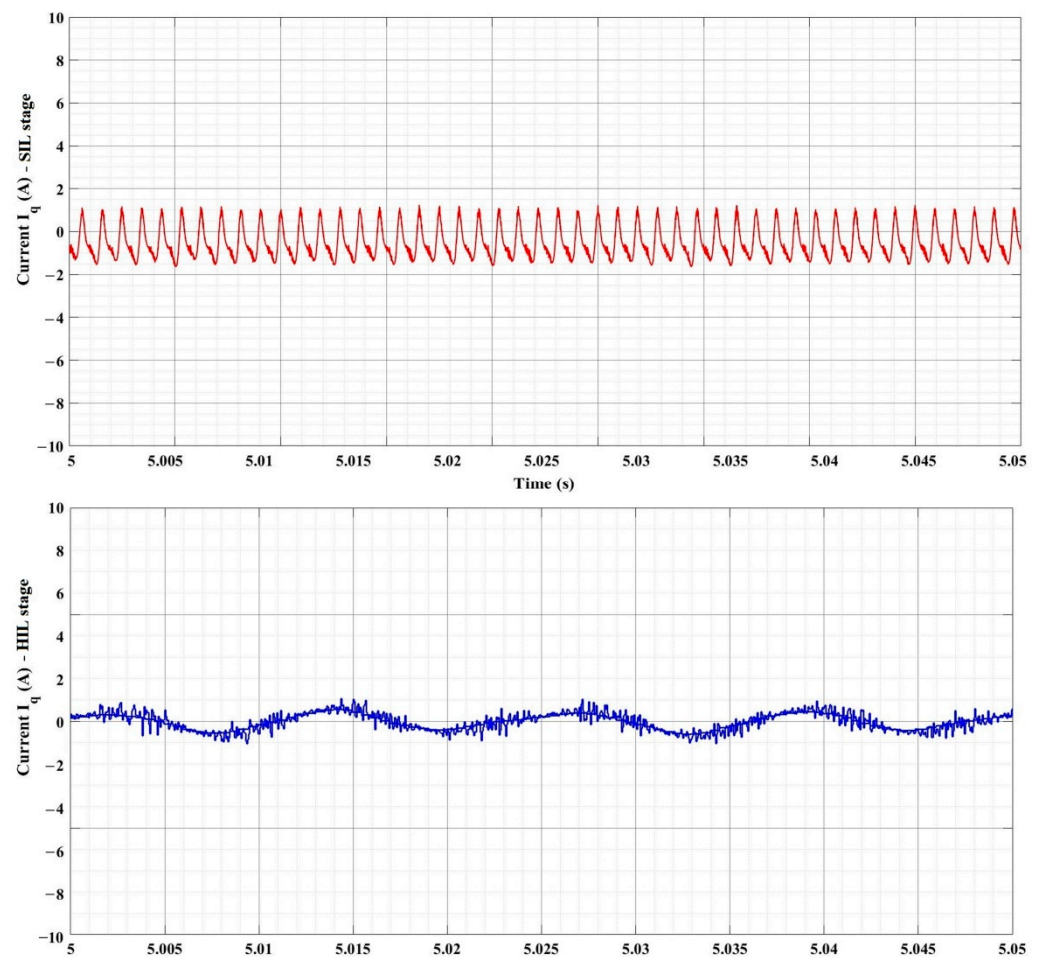


Figure 42. Comparison of i_q current evolution in the numerical simulation (SIL stage) and in real-time (HIL stage).

Next, a comparison is presented for the implementation of HIL, of the backstepping-type and PI-type algorithms optimized by using a PSO-type method for PMSM control. The reference speed profile used for this comparison is given by the sequence: [0 10 20 30 40 50 60 70 80] s \rightarrow [1000 1500 2000 2500 3000 2500 2000 1500 1000] rpm and a load torque with a variation around 1 Nm.

The experimental results obtained in order to make the proposed comparison are presented below. Thus, Figures 43–46 present real-time evolutions of the comparative PMSM rotor speeds, comparative i_a and i_b currents, comparative i_q currents, and comparative electromagnetic torques. There is an improved response time in the case of the backstepping controller compared to the PI-PSO controller, both for an upping step and downing step of the imposed reference speed (55 ms for backstepping controller and 85 ms for PI-PSO controller). Although for both controllers the override is zero, in the case of the backstepping controller, the steady-state error is less than 0.15%, and in the case of the PI-PSO controller, it is less than 0.35%.

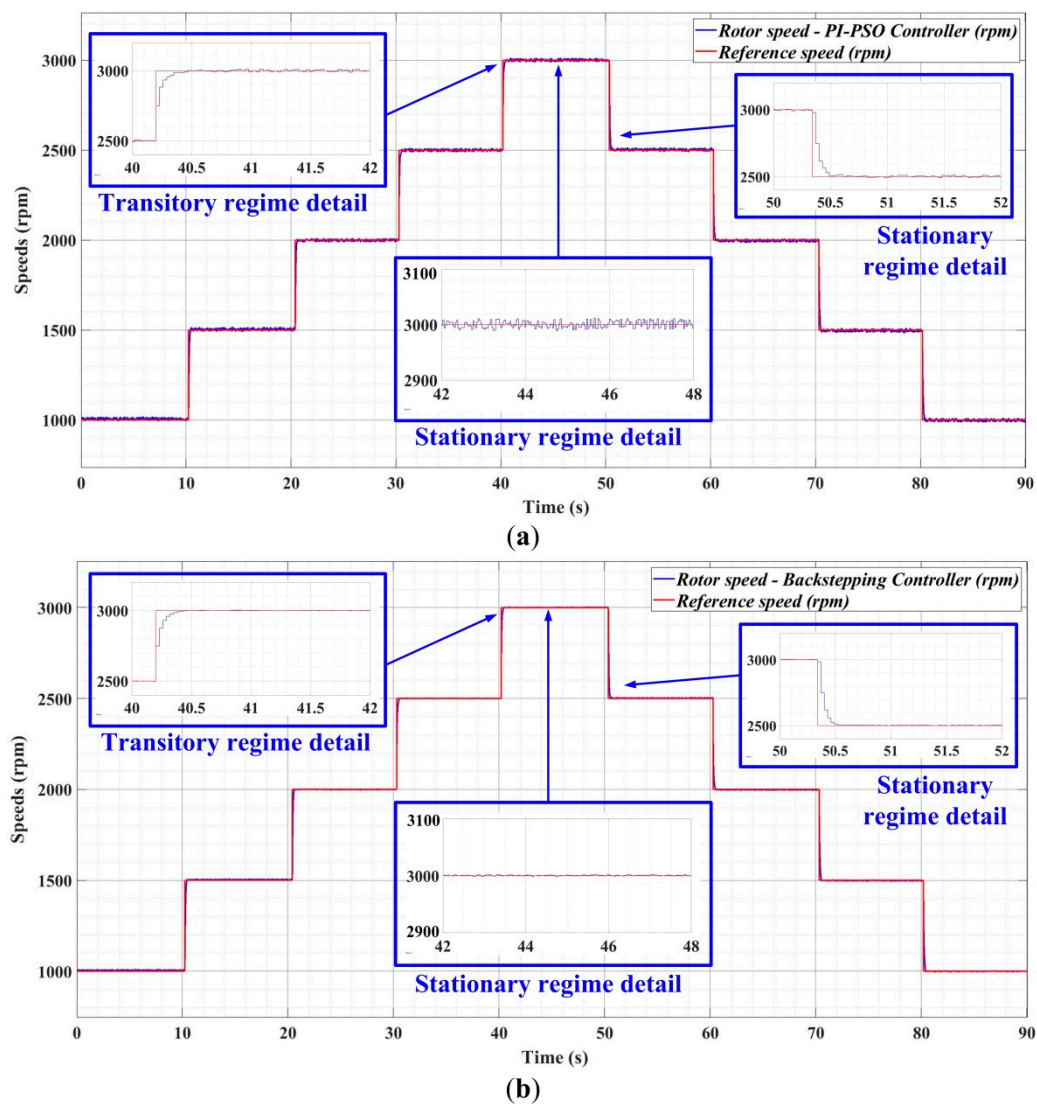


Figure 43. Comparative real-time evolution of the PMSM rotor speeds: (a) with PI-PSO controller; (b) with backstepping controller.

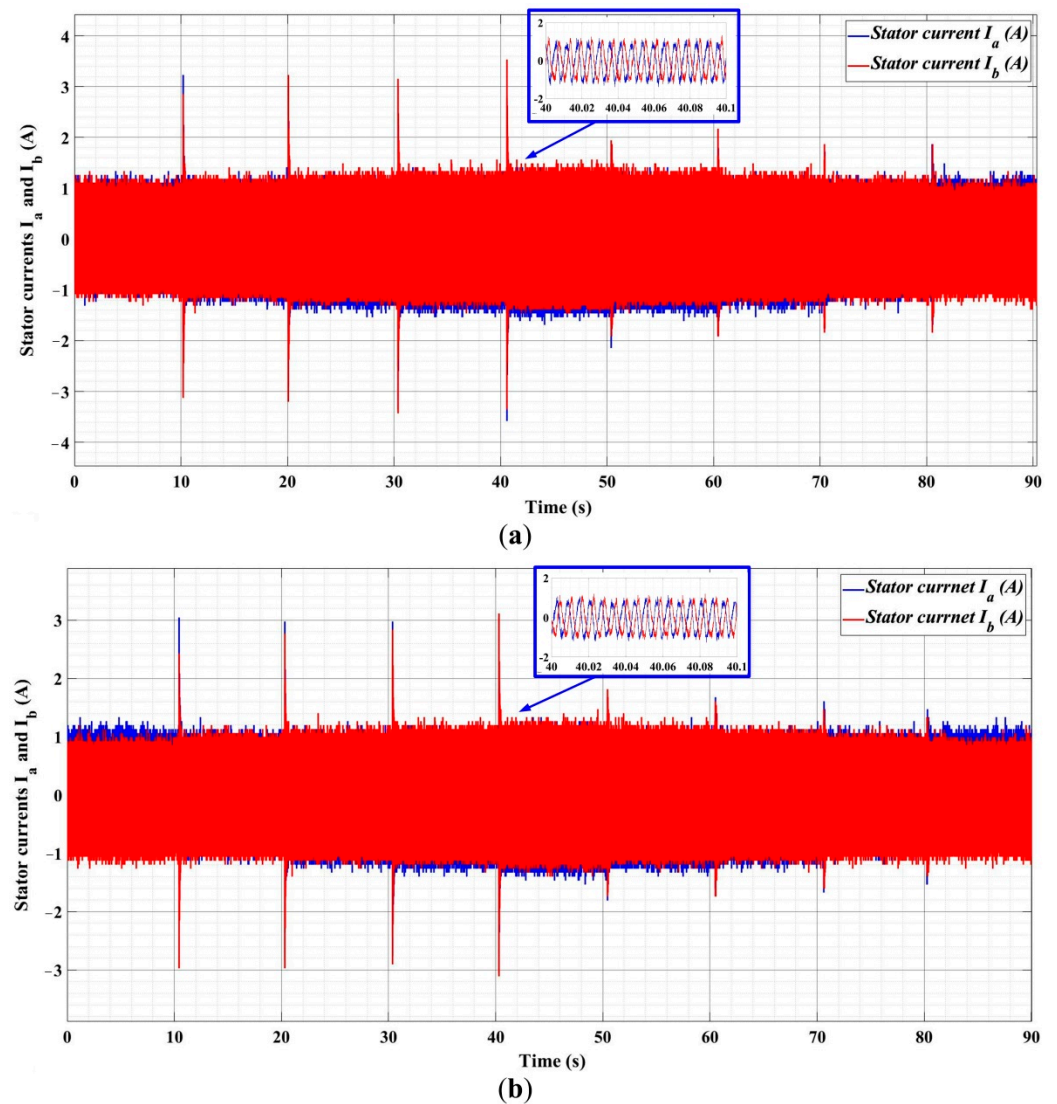


Figure 44. Comparative real-time evolution of the i_a and i_b stator currents: (a) with PI-PSO controller; (b) with backstepping controller.

A synthetic indicator for comparing the performance of backstepping and PI-PSO controllers is the implementation in the discrete domain of a root mean square integral, given by the speed ripple:

$$\omega_{rip} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\omega(i) - \omega_{ref}(i))^2} \quad (50)$$

where N is the number of samples and ω and ω_{ref} are the PMSM rotor speed and PMSM reference speed, respectively.

Following the calculations based on the recorded speed response signals of the two controllers, by means of the MATLAB/Simulink software interface for the monitoring and control application on the Host PC described in Section 6.2, a ripple of 25.17 rpm was obtained in the case of the backstepping controller and of 32.49 rpm in the case of the PI-PSO controller.

The superiority of the backstepping-type controller over the PI-PSO-type controller is also demonstrated by means of the FFT analysis and THD calculation of stator currents, as shown in Figure 47. There is a smoothing of current harmonics and a reduction in THD from 40.32% to 24.21%.

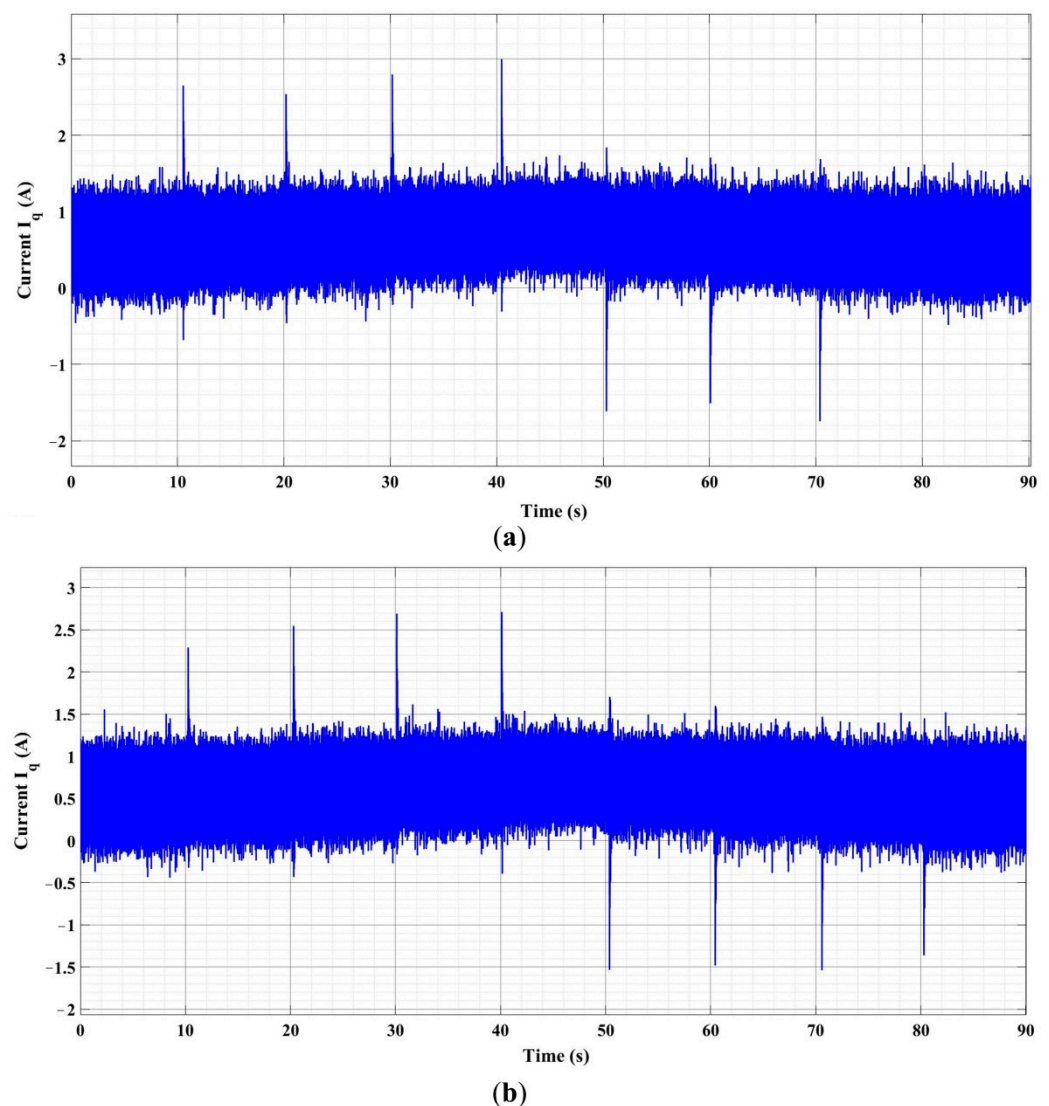


Figure 45. Comparative real-time evolution of the i_q currents: (a) with PI-PSO controller; (b) with backstepping controller.

The superiority of the results in terms of the PMSM control in the case of using a backstepping-type controller versus a PI-PSO-type controller is summarized in Table 2.

An image of the experimental stand for testing the PMSM sensorless control system in HIL stage is presented in Figure 48.

Table 2. The comparative performances of the backstepping-type controller versus PI-PSO-type controller.

Performance Indices	PI-PSO Controller	Backstepping Controller
Response time (ms)	85	55
Overshoot (%)	0	0
Steady-state error (%)	0.35	0.15
Speed ripple (rpm)	32.49	25.17
Currents THD (%)	40.32	24.21

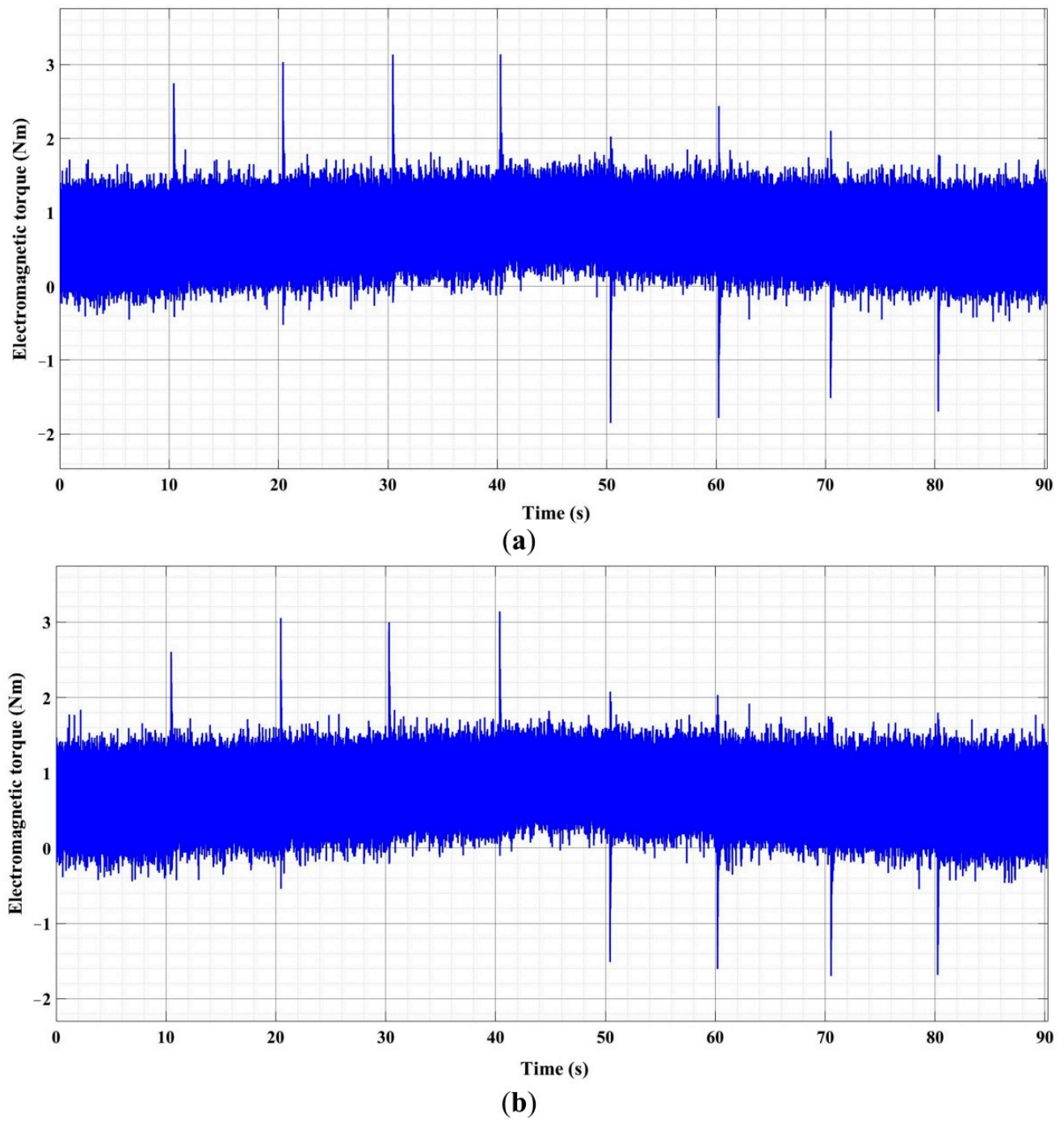
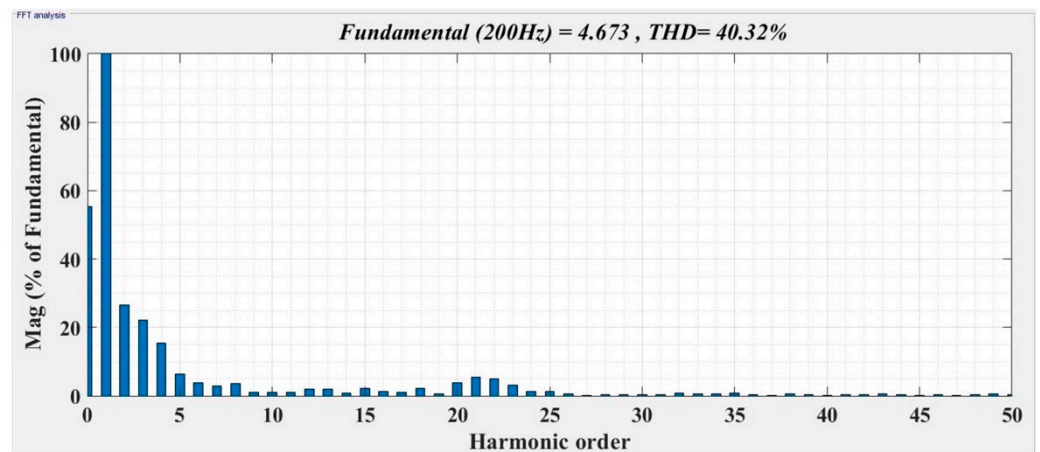
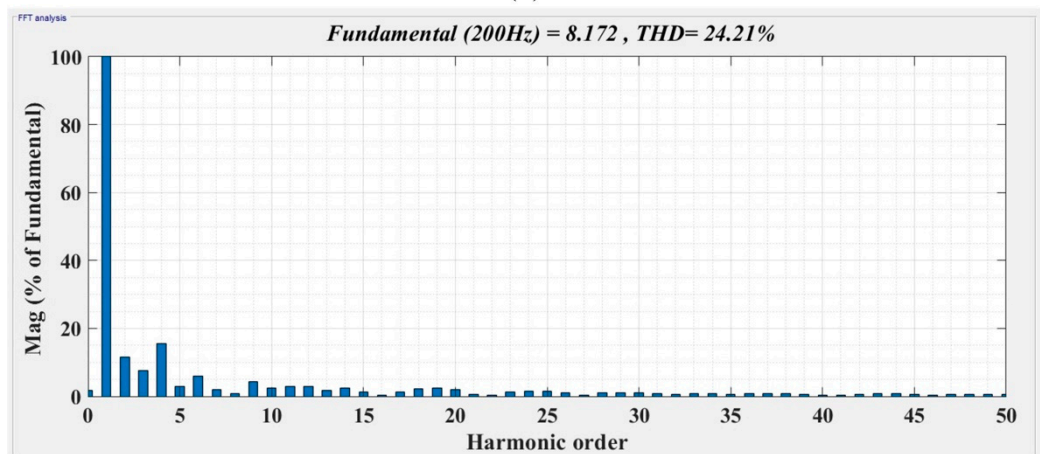


Figure 46. Comparative real-time evolution of the electromagnetic torques: (a) with PI-PSO controller; (b) with backstepping controller.



(a)



(b)

Figure 47. Comparative FFT and THD analysis: (a) with PI-PSO controller; (b) with backstepping controller.

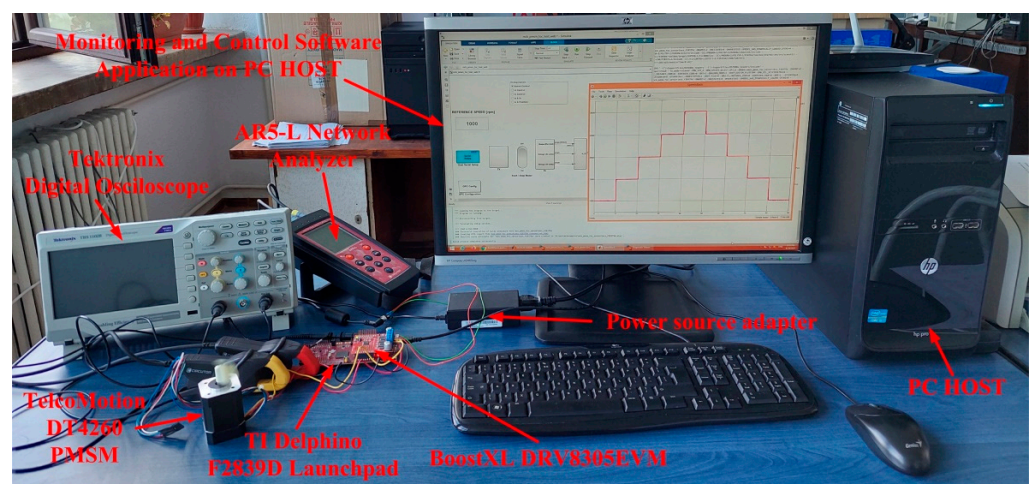


Figure 48. Experimental image for testing the PMSM sensorless control system.

7. Conclusions

Based on the FOC-type classic control structure, under the conditions where even the optimization of the tuning parameters of the classic PI controller is performed by a PSO-type method, we note that the PMSM control system can ensure good performance only in the case of limited variations of the system parameters, and not in the case of high dynamics and sudden variations of the load torque. For this purpose, to embed the non-linearities of the control system, a backstepping-type control law is synthesized, whose tuning parameters are optimized by a PSO method. In terms of estimating the PMSM rotor position speed in order to embed the disturbances of the system, an ESO-type observer is designed with a response time higher than that of the control system. The superior performance of the proposed PMSM sensorless control system based on a backstepping-PSO-type controller and an ESO-type observer is verified by numerical simulations, both within the usual range of the control parameters and in the case of rapid and high-amplitude variations of the load torque accompanied by a uniform random noise and in the case of variation of up to 100% of the system parameters such as the moment of inertia J and the static resistance R_s . At the end of the paper, we summarize the transition and the stages of paradigms SIL, PIL, and HIL related to the real-time implementation of a PMSM control system. In addition, we present a series of experiments of the PMSM control system in the HIL stage, which confirm the robustness and superiority of the proposed backstepping control law. The experimental results prove the superiority of the backstepping-type controller over the PI-PSO-type controller and are backed by comparing the following performance indicators: response time, override, steady-state error, speed ripple, and THD currents.

Author Contributions: Conceptualization, C.-I.N., M.N. and D.S.; Data curation, C.-I.N., M.N. and D.S.; Formal analysis, C.-I.N., M.N. and D.S.; Funding acquisition, D.S.; Investigation, C.-I.N., M.N. and D.S.; Methodology, C.-I.N., M.N. and D.S.; Resources, C.-I.N., M.N. and D.S.; Software, C.-I.N. and M.N.; Supervision, C.-I.N., M.N. and D.S.; Validation, C.-I.N., M.N. and D.S.; Visualization, C.-I.N. and M.N.; Writing—original draft, C.-I.N. and M.N.; Writing—review and editing, C.-I.N. and M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by European Regional Development Fund Competitiveness Operational Program, project TISIPRO, ID: P_40_416/105736, 2016–2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Prabhakaran, K.K.; Karthikeyan, A.; Varsha, S.; Perumal, B.V.; Mishra, S. Standalone single stage PV fed reduced switch inverter based PMSM for water pumping application. *IEEE Trans. Ind. Appl.* **2020**, *56*, 6526–6535. [\[CrossRef\]](#)
2. Zhang, Z.; Ma, R.; Wang, L.; Zhang, J. Novel PMSM control for anti-lock braking considering transmission properties of the electric vehicle. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10378–10386. [\[CrossRef\]](#)
3. Choo, K.M.; Won, C.Y. Design and analysis of electrical braking torque limit trajectory for regenerative braking in electric vehicles with PMSM drive systems. *IEEE Trans. Power Electron.* **2020**, *35*, 13308–13321. [\[CrossRef\]](#)
4. Bose, B.K. *Modern Power Electronics and AC Drives*; Prentice Hall: Upper Saddle River, NJ, USA, 2002; pp. 439–534.
5. Wang, H.; Leng, J. Summary on development of permanent magnet synchronous motor. In Proceedings of the Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 689–693.
6. Aqil, M.; Im, J.-H.; Hur, J. Application of Perovskite Layer to Rotor for Enhanced Stator-Rotor Capacitance for PMSM Shaft Voltage Reduction. *Energies* **2020**, *13*, 5762. [\[CrossRef\]](#)
7. Liu, S.-M.; Tu, C.-H.; Lin, C.-L.; Liu, V.-T. Field-Oriented Driving/Braking Control for Electric Vehicles. *Electronics* **2020**, *9*, 1484. [\[CrossRef\]](#)
8. Liu, X.; Zhang, Q. Robust Current Predictive Control-Based Equivalent Input Disturbance Approach for PMSM Drive. *Electronics* **2019**, *8*, 1034. [\[CrossRef\]](#)
9. Zhang, G.; Chen, C.; Gu, X.; Wang, Z.; Li, X. An Improved Model Predictive Torque Control for a Two-Level Inverter Fed Interior Permanent Magnet Synchronous Motor. *Electronics* **2019**, *8*, 769. [\[CrossRef\]](#)
10. Wang, Q.; Yu, H.; Wang, M.; Qi, X. A Novel Adaptive Neuro-Control Approach for Permanent Magnet Synchronous Motor Speed Control. *Energies* **2018**, *11*, 2355. [\[CrossRef\]](#)
11. Lu, S.; Tang, X.; Song, B. Adaptive PIF Control for Permanent Magnet Synchronous Motors Based on GPC. *Sensors* **2013**, *13*, 175–192. [\[CrossRef\]](#)

12. Wang, M.-S.; Hsieh, M.-F.; Lin, H.-Y. Operational Improvement of Interior Permanent Magnet Synchronous Motor Using Fuzzy Field-Weakening Control. *Electronics* **2018**, *7*, 452. [\[CrossRef\]](#)
13. Hicham, F.; Yousfi, D.; Youness, A.D.; Larbi, E.M.; Rahim, N.A. Sliding-Mode Speed Control of PMSM with Fuzzy-Logic Chattering Minimization—Design and Implementation. *Information* **2015**, *6*, 432–442. [\[CrossRef\]](#)
14. Hoai, H.-K.; Chen, S.-C.; Chang, C.-F. Realization of the Neural Fuzzy Controller for the Sensorless PMSM Drive Control System. *Electronics* **2020**, *9*, 1371. [\[CrossRef\]](#)
15. Hoai, H.-K.; Chen, S.-C.; Than, H. Realization of the Sensorless Permanent Magnet Synchronous Motor Drive Control System with an Intelligent Controller. *Electronics* **2020**, *9*, 365. [\[CrossRef\]](#)
16. Nicola, M.; Nicola, C.-I. Sensorless Fractional Order Control of PMSM Based on Synergetic and Sliding Mode Controllers. *Electronics* **2020**, *9*, 1494. [\[CrossRef\]](#)
17. Nicola, M.; Nicola, C.-I. Fractional-Order Control of Grid-Connected Photovoltaic System Based on Synergetic and Sliding Mode Controllers. *Energies* **2021**, *14*, 510. [\[CrossRef\]](#)
18. Zhu, Y.; Tao, B.; Xiao, M.; Yang, G.; Zhang, X.; Lu, K. Luenberger Position Observer Based on Deadbeat-Current Predictive Control for Sensorless PMSM. *Electronics* **2020**, *9*, 1325. [\[CrossRef\]](#)
19. Comanescu, M.; Xu, L. Sliding-mode MRAS speed estimators for sensorless vector control of induction Machine. *IEEE Trans. Ind. Electron.* **2006**, *53*, 146–153. [\[CrossRef\]](#)
20. Gao, W.; Zhang, G.; Hang, M.; Cheng, S.; Li, P. Sensorless Control Strategy of a Permanent Magnet Synchronous Motor Based on an Improved Sliding Mode Observer. *World Electr. Veh. J.* **2021**, *12*, 74. [\[CrossRef\]](#)
21. Kamel, T.; Abdelkader, D.; Said, B.; Padmanaban, S.; Iqbal, A. Extended Kalman Filter Based Sliding Mode Control of Parallel-Connected Two Five-Phase PMSM Drive System. *Electronics* **2018**, *7*, 14. [\[CrossRef\]](#)
22. Humaidi, A.J.; Badr, H.-M.; Ajil, A.R. Design of Active Disturbance Rejection Control for Single-Link Flexible Joint Robot Manipulator. In Proceedings of the 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 10–12 October 2018; pp. 452–457.
23. Zhao, Y. Applications of Sliding Mode Controller and Linear Active Disturbance Rejection Controller to a PMSM Speed System. Ph.D. Thesis, Master of Science in Electrical Engineering, Cleveland State University, Cleveland, OH, USA, 2013.
24. Chen, W.-H.; Yang, J.; Guo, L.; Li, S. Disturbance-Observer-Based Control and Related Methods—An Overview. *IEEE Trans. Ind. Electron.* **2016**, *63*, 1083–1096. [\[CrossRef\]](#)
25. Yang, J.; Chen, W.; Li, S.; Guo, L.; Yan, Y. Disturbance/Uncertainty Estimation and Attenuation Techniques in PMSM Drives—A Survey. *IEEE Trans. Ind. Electron.* **2017**, *64*, 3273–3285. [\[CrossRef\]](#)
26. Nicola, M.; Nicola, C.-I. Sensorless Control of PMSM using Backstepping Control and ESO-type Observer. In Proceedings of the 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 25–27 June 2020; pp. 1–6.
27. Simscape Electrical. Available online: <https://www.mathworks.com/getting-started-with-simscape-electrical.html> (accessed on 4 May 2020).
28. LAUNCHXL-F28379D C2000 Delfino MCU F28379D LaunchPad™ Development Kit. Available online: <https://www.ti.com/tool/LAUNCHXL-F28379D> (accessed on 10 August 2020).
29. Engelbrecht, A.-P. *Computational Swarm Intelligence in Computational Intelligence: An Introduction*, 2nd ed.; John Wiley & Sons Ltd.: Chichester, UK, 2007; pp. 289–357.
30. Motor Control Blockset. Available online: <https://www.mathworks.com/help/mcb/index.html> (accessed on 1 October 2020).