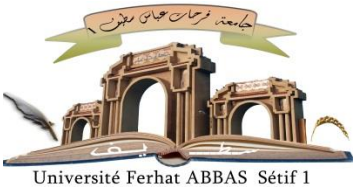
الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITÉ FERHAT ABBAS - SETIF1

FACULTÉ DE TECHNOLOGIE

THÈSE

Présentée au Département d'Electrotechnique

Pour l'obtention du diplôme de

DOCTORAT

Domaine : Sciences et Technologie

Filière : Electrotechnique **Option : Commande Electrique**

Par

NADJI Salah

THÈME

Commande intelligente robuste : Application à un entrainement électrique

Soutenue le ……./……./2022 devant le Jury:

BOUAFIA Abdelouahab	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Président**
BENAICHA Samira	**M.C.A.**	**Univ. Mustapha Ben Boulaid Batna 2**	**Directeur de thèse**
MOREAU Sandrine	**M.C.A.**	**Univ. Poitiers**	**Co-Directeur**
HEMSAS Kamel Eddine	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Examinateur**
CHAIBA Azzedine	**Professeur**	**Univ. Khenchela**	**Examinateur**
SARI Bilal	**M.C.A.**	**Univ. Ferhat Abbas Sétif 1**	**Examinateur**

Keywords

Artificial neural network, backstepping control, dynamic surface control, integral action, interior permanent magnet synchronous motor, leaner extended state observer, vector control.

Abstract

Despite many advantageous features of Interior Permanent Magnet Synchronous Motor (IPMSM), the precise speed control of IPMSM drive becomes a complex issue due to nonlinear coupling among its winding currents and the rotor speed as well as the nonlinearity present in the electromagnetic torque due to magnetic saturation of the rotor core particularly. Fast and accurate response, quick recovery of speed from any disturbances, and insensitivity to parameter variations are some of the important characteristics of high-performance drive systems used in robotics, rolling mills, traction, and spindle drives. The conventional controllers such as Proportional Integral (PI), and Proportional Integral Derivative (PID) are sensitive to parameter variations and load disturbances. To obtain high dynamic performance, recently researchers developed several nonlinear as well as intelligent controllers.

The backstepping control technique has gained more and more attention in its use for IPMSM drive systems, which greatly simplifies the overall system design process. However, the conventional backstepping control scheme suffers from some drawbacks such as steady-state errors, sensitivity to load torque disturbances, and the explosion of complexity. In order to overcome these problems and improve control performance, this thesis presents two backstepping control methods for IPMSM drives.

Firstly, a new control technique for IPMSM that has been designed using a modified integral backstepping controller (IBC) is proposed. This architecture consists of a nonlinear backstepping control scheme to achieve the purpose of speed tracking. Then the nonlinear controller based on the backstepping control scheme with the introduction of integral actions is designed for the IPMSM drive system. The suggested controller is not only designed to stabilize the IPMSM system, but also to ensure that the speed tracking error converges asymptotically to zero in the existence of system uncertainties and external disturbances. The advantage of the suggested approach is that the proposed controller guarantees good tracking performance, reduces steady-state errors, robustness against all parameter uncertainties, and load torque disturbances in the IPMSM system. The performance of the proposed controller is compared to that of the conventional PI controller.

In the second place, to further enhance the performance of the IPMSM drive, Dynamic Surface Control (DSC) method was presented, which overcomes the problem of the explosion of complexity associated with the traditional backstepping design procedure by introducing first-order low-pass filters. Compared to the conventional control method, the proposed DSC technique is very simple and easy to implement in practice. This controller is adapted via the online estimation of the unknown load torque and friction effects. A Feedforward Neural Network (FFNN) observer, whose architecture is based on a successful model-based approach, is devised for this purpose, thus ensuring high closed-loop performance of the motor trajectory tracking task.

At last, the complete IPMSM drive incorporating various control algorithms has been successfully implemented using digital signal processor (DSP) controller board-DSI104 for a laboratory 1.5 hp motor. The effectiveness of the proposed drive is verified both in simulation and experiment at different operating conditions. The results show the effectiveness of the proposed schemes and their potential for application in a real-time industrial drive application.

Table of Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

LIST OF SYMBOLS

a, b, c	stator stationary frame;
B_r	remanent flux density;
BH_{max}	maximum energy product;
b_o	neuron bias;
c_0, c_1	load torque LESO design coefficients;
d_{pj}	desired network outputs;
d, q, o	rotor synchronous frame;
$e_1, e_1', e_2, e_3, e_4, e_4'$	backstepping tracking errors;
e	vector of network errors;
e_τ	load torque estimation error;
E	objective function of w;
E_p	output error for each example p;
F	viscous friction coefficient;
F_ω	speed regulator transfer function;
F_{0L}	speed open-loop transfer function;
F_{CL}	speed closed-loop transfer function;
g	the gradient;
H_c	coercivity of the permanent magnet material;
H	Hessian matrix;
I_i	neuron inputs;
i_a, i_b, i_c	instantaneous phase currents;
i_d, i_q, i_o	d, q, and o axis currents;
i_{qref}	q-axis current reference;
i_{dref}	d-axis current reference;
J	rotor moment of inertia;
J_c	Jacobian matrix;
K_p	proportional gain;
K_i	integral gain;
$k_1, k_1', k_2, k_3, k_4, k_4'$	backstepping controller gains;
L_d, L_q, L_o	d, q, and o axis inductances;
L_{aa}, L_{bb}, L_{cc}	self-inductances of the three phases;
L_{ab}, L_{bc}, L_{ca}	mutual-inductances between one phase to the other ones;
$L_{\sigma s}$	leakage inductance;
L_{ms0}	constant component of the magnetization inductance;
L_m	amplitude of the anisotropy component of the magnetization inductance;
L_s	self-inductance;
M_s	average coefficient of mutual induction between the phases of the stator winding;
p	number of examples in the training set;
P_e	input electric power absorbed by the machine;

P_{cu}	joule losses in the stator windings;
P_{mf}	energy stored in the magnetic field;
P_m	output electromechanical power;
P	pole pair;
p_{d-obs}	second-degree characteristic polynomial;
R	phase resistance;
s	Laplace variable;
T_L	load torque;
T_c	Curie temperature;
$T_{w\text{-}max}$	PM maximum operating temperature;
T_{em}	electromechanical torque;
T_r	response time;
$U_r(t)$	output signal;
u_1, u_2	load torque LESO inputs;
v	activation function input;
v_a, v_b, v_c	instantaneous phase voltages;
v_d, v_q, v_o	*d*, *q*, and *o* axis voltages;
w	weight matrix;
w_i	neuron weights;
X_{k+1}	approximation to the Hessian matrix;
y_1	load torque LESO output;
y_{pj}	calculated network outputs;
θ_r	rotor mechanical position;
θ_e	rotor electrical position;
ω_r	rotor mechanical speed;
ω_{ref}	rotor reference speed;
ϕ_a, ϕ_b, ϕ_c	instantaneous magnetic phase flux linkage;
ϕ_f	magnet flux linkage;
$\phi_{ai}, \phi_{bi}, \phi_{ci}$	flux linkage produced by stator current excitation;
$\phi_{fa}, \phi_{fb}, \phi_{fc}$	flux linkage produced by permanent magnets;
ϕ_d, ϕ_q, ϕ_o	*d*, *q*, and *o* axis fluxes;
$\varepsilon(t)$	error signal;
τ_d	electrical time constant of the *d*-axis;
τ_q	electrical time constant of the *q*-axis;
τ_m	mechanical time constant of the machine;
τ_L	unknown time-varying load torque;
$\hat{\tau}_L$	estimated load torque;
ω_n	natural frequency;
ξ	damping factor;
ρ	damping coefficient;
α, β	fixed two-phase frame.

LIST OF ABBREVIATIONS

AC	Alternating Current;
Alnico	aluminum (Al) nickel (Ni) cobalt (Co);
ADRC	Active Disturbances Rejection Controller;
AI	Artificial Intelligence;
ANN	Artificial Neural Network;
Back-EMF	Back Electromotive Force;
CLTF	Closed Loop Transfer Function;
DNN	Deep Neural Network;
DC	Direct Current;
DSC	Dynamic Surface Control;
DSP	Digital Signal Processor;
EMF	Electromotive Force;
ESO	Extended State Observer;
FOC	Field-Oriented Control;
FFNN	Feedforward Neural Network;
H_∞	Horizon Infinity;
IPMSM	Interior Permanent Magnet Synchronous Motor;
IPMSG	Interior Permanent Magnet Synchronous Generator;
IBC	Integral Backstepping Controller;
LMA	Levenberg-Marquardt Algorithm;
LESO	Linear Extended State Observer;
MRAC	Model Reference Adaptive Control;
MMF	Magnetomotive Force;
NN	Neural Network;
NdFeB	neodymium (Nd) iron (Fe) boron (B);
OLTF	Open Loop Transfer Function;
PM	Permanent Magnet;
PI	Proportional Integral;
PID	Proportional Integral Derivative;
PWM	Pulse Width Modulation;
RMS	Root Mean Square;
ReLU	Rectifier Linear Unit.
SVPWM	Space Vector Pulse Width Modulation;
SPMSM	Surface-Mounted Permanent Magnet Synchronous Motor;
SynRM	Synchronous Reluctance Machine;
SMC	Sliding Mode Control;
SmCo	samarium (Sm) cobalt (Co);
TD	Tracking Differentiator;

List of Publications

PEER-REVIEWED PUBLICATION

- N. Salah, B. Samira, and S. Moreau, "Modified backstepping control of IPMSM: Experimental Tests," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 236, pp.1-13, 2022.

CONFERENCES ATTENDED

- S. Nadji and S. Benaicha 'Robust Dynamic Surface Control of Interior Permanent Magnet Synchronous Motors', *The International Conference on Electronics, Energy and Measurements (IC2EM-2018),* Algiers, Algeria, 2018, pp. 151-157.
- S. Nadji, S. Benaicha, N. Sabeur and F. M. Zaihidee 'Robust Backstepping Control with Integral Action of IPM Synchronous Motor', *The Third International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, Algiers, Algeria, 2018.
- S. Nadji and S. Benaicha 'Robust Integral Backstepping Control of PMSM for Electric Vehicle Application', *The 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, Boumerdes, Algeria, 2017.
- S. Nadji and S. Benaicha 'Sensorless Speed Control of Salient Pole PMSM According to the Backstepping Observer', *International Conference on Recent Advances in Electrical Systems*, Hammamet, Tunisia, 2016, pp. 226-231.

Acknowledgments

First of all, thanks to Almighty **ALLAH**, the Most Gracious, the Most Merciful, who guides me on every step I take and gave me the strength and knowledge to complete this thesis.

As I would like to mention, my thesis could not have been completed without the encouragement and support of my family. I would like to dedicate my thesis to my parents, sisters, and brothers who are a source of care, love, support, and strength during my graduate study.

Huge gratitude goes to my supervisors; Dr. Benaicha Samira of the department of electrical engineering at the University of Batna 2, and Dr. Moreau Sandrine of the University of Poitiers. Thank you for this opportunity and the guidance throughout the thesis.

Special thanks to Prof. Mekhilef Saad who hosted me in his laboratory for the first experimental validation of my control programs and was a source of inspiration for me.

Gratitude goes to all my other friends who have supported me along the way. Thanks for all your encouragement.

Abstract

Despite many advantageous features of Interior Permanent Magnet Synchronous Motor (IPMSM), the precise speed control of IPMSM drive becomes a complex issue due to the existence of nonlinearities and system uncertainties. The conventional linear controllers are sensitive to parameter variations and load disturbances. To obtain high dynamic performance, recently researchers developed several nonlinear as well as intelligent controllers. Among these new control methods, the backstepping control method has attracted much attention thanks to its robustness and recursive designs. However, the conventional backstepping control scheme suffers from some drawbacks such as steady-state errors, sensitivity to load torque disturbances, and the explosion of complexity. In order to overcome these problems, this thesis presents two backstepping control methods for IPMSM drives.

Firstly, a new control technique for IPMSM that has been designed using a modified integral backstepping controller (IBC) is proposed. This architecture consists of a nonlinear backstepping control scheme to achieve the purpose of speed tracking. Then the nonlinear controller based on the backstepping control scheme with the introduction of integral actions is designed for the IPMSM drive system. The advantage of the suggested approach is that the proposed controller guarantees good tracking performance, reduces steady-state errors, robustness against all parameter uncertainties, and load torque disturbances in the IPMSM system. The performance of the proposed controller is compared to that of the conventional PI controller.

In the second place, to further enhance the performance of the IPMSM drive, Dynamic Surface Control (DSC) method was presented, which overcomes the problem of the explosion of complexity associated with the traditional backstepping design procedure by introducing first-order low-pass filters. Compared to the conventional control method, the proposed DSC technique is very simple and easy to implement in practice. This controller is adapted via the online estimation of the unknown load torque and friction effects. A Feedforward Neural Network (FFNN) observer, whose architecture is based on a successful model-based approach, is devised for this purpose, thus ensuring high closed-loop performance of the motor trajectory tracking task. Finally, the validity of the proposed control schemes is confirmed through simulation and experimental studies on IPMSM testbed with a TMS320F240 DSP.

Keywords

Artificial neural network, backstepping control, dynamic surface control, integral action, interior permanent magnet synchronous motor, leaner extended state observer, vector control.

Résumé

Malgré les nombreux avantages du moteur synchrone à aimants permanents (MSAP), le contrôle précis de la vitesse du moteur devient un problème complexe en raison de l'existence de non-linéarités et d'incertitudes paramétriques. Les contrôleurs linéaires classiques sont sensibles aux variations paramétriques et aux perturbations de charge. Pour obtenir des performances dynamiques élevées, les chercheurs ont récemment développé plusieurs contrôleurs non linéaires et intelligents. Parmi ces nouvelles méthodes de contrôle, le contrôle par backstepping a attiré beaucoup d'attention grâce à sa robustesse et ses conceptions récursives. Cependant, le contrôle par backstepping classique souffre de certains inconvénients tels que les erreurs statiques, la sensibilité aux perturbations du couple de charge et l'explosion de la complexité. Afin de surmonter ces problèmes, cette thèse présente deux méthodes de contrôle par backstepping pour le MSAP.

Premièrement, une nouvelle technique de contrôle pour le MSAP qui a été conçue à l'aide d'un contrôleur par backstepping avec action intégral modifié est proposée. Cette architecture consiste en un schéma de commande non linéaire par backstepping pour atteindre l'objectif de suivi de la vitesse. Ensuite, le contrôleur non linéaire basé sur le contrôle par backstepping avec l'introduction d'actions intégrales est conçu pour le système d'entraînement du MSAP. L'avantage de l'approche suggérée est que le contrôleur proposé garantit de bonnes performances de suivi, réduit les erreurs statiques, la robustesse contre toutes les incertitudes paramétriques et les perturbations de couple de charge dans le système. Les performances du contrôleur proposé sont comparées à celles du contrôleur PI classique.

Deuxièmement, pour améliorer encore les performances du système d'entraînement, un contrôle par backstepping avec filtrage des commandes virtuels par Dynamic Surface Control (DSC) a été introduit, ce qui surmonte le problème d'explosion de complexité associée à la commande par backstepping classique en introduisant des filtres passe-bas du premier ordre. Par rapport à la méthode de contrôle classique, la technique DSC proposée est très simple et facile à mettre en œuvre en pratique. Ce contrôleur est adapté via l'estimation en ligne du couple de charge et du couple de frottement. Un observateur avec réseau de neurones à propagation avant, dont l'architecture est basée sur une approche similaire à un observateur d'état, est conçu à cet effet, assurant une grande précision de suivi de la vitesse de référence. Enfin, la validité des schémas de contrôle proposés est confirmée par des simulations et des études expérimentales sur un banc d'essai avec un DSP TMS320F240.

Mots clés

Réseau de neurones artificiels, contrôle par backstepping, contrôle par backstepping avec filtrage des commandes virtuels par Dynamic Surface Control, action intégrale, moteur synchrone à aimants permanents, observateur d'état, contrôle vectoriel.

الملخص

على الرغم من المزايا العديدة للمحرك المتزامن ذو المغناطيس الدائم (م م م د)، فإن التحكم الدقيق في سرعته يصبح مشكلة معقدة بسبب وجود اللاخطية وعدم اليقين في النظام. تعتبر وحدات التحكم الخطية التقليدية حساسة لتغيرات وسائط المحرك واضطرابات عزم دوران الحمل. للحصول على أداء ديناميكي عالي، طور الباحثون مؤخرًا العديد من خوارزمية التحكم الذكية وغير الخطية. من بين طرق التحكم الجديدة هذه، جذبت طريقة تحكم الخطوة الخلفية الكثير من الاهتمام بفضل خصائصها المتميزة. ومع ذلك، فإن مخطط تحكم الخطوة الخلفية التقليدي يعاني من بعض العيوب مثل أخطاء الحالة المستقرة، والحساسية لاضطرابات عزم دوران الحمل، وانفجار التعقيد. للتغلب على هذه المشاكل، تقدم هذه الأطروحة طريقتان للتحكم بالخطوة الخلفية لـ م م م د.

أولاً، تم اقتراح تقنية تحكم جديدة لـ م م م د التي تم تصميمها باستخدام متحكم الخطوة الخلفية متكامل معدل. تتكون هذه البنية من مخطط تحكم غير خطي بالخطوة الخلفية لتحقيق هدف تتبع السرعة. بعد ذلك، تم تصميم وحدة التحكم غير الخطية القائمة على التحكم بالخطوة الخلفية مع إدخال عناصر التكامل لنظام قيادة المحرك. تتمثل مزايا الطريقة المقترحة في أن وحدة التحكم تضمن أداء تتبع جيد، وتقلل أخطاء الحالة المستقرة، والمتانة ضد جميع حالات عدم اليقين واضطرابات عزم الدوران في النظام. تم مقارنة أداء المتحكم المقترح مع أداء متحكم كلاسيكي من نوع PI.

ثانيًا، لزيادة تحسين أداء نظام القيادة، تم تقديم طريقة التحكم الديناميكي في السطح (DSC)، والتي تم استخدامها للتغلب على مشكلة انفجار التعقيد المرتبط بتحكم الخطوة الخلفية التقليدي من خلال إدخال مرشحات تمرير منخفض من الدرجة الأولى. بالمقارنة مع طريقة التحكم التقليدية، فإن تقنية DSC المقترحة بسيطة للغاية وسهلة التنفيذ عمليا. بعدها تم تكييف وحدة التحكم هذه من خلال تخمين عزم دوران الحمل وعزم دوران الاحتكاك. تم تصميم مراقب بواسطة شبكة عصبية ذات تغذية أمامية لهذا الغرض، الذي تعتمد بنيته على نهج مماثل لمراقب الحالة، مما يضمن دقة تتبع عالية للسرعة المرجعية. أخيرًا، تم تأكيد صحة مخططات التحكم المقترحة من خلال المحاكاة والدراسات التجريبية على م م م د باستخدام DSP TMS320F240.

كلمات مفتاحية

شبكة عصبية اصطناعية، تحكم الخطوة الخلفية، التحكم الديناميكي في السطح، عنصر التكامل، المحرك المتزامن ذو المغناطيس الدائم، مراقب الحالة، التحكم الشعاعي.

Chapter 1: Introduction

1.1 MOTIVATIONS

Electrical drive controls are playing a principal role in a wide range of industrial control applications, often used in material handling, precision machining, transportation systems, and many automation processes. Growing user demand for dynamic response, precision, and robustness driven by advances in technology and energy conservation requires improved control [1]. Over the past decades, controlled electrical drives have performed in various configurations and have become a mature technology with an already substantial and continuously increasing worldwide market. This is due to developments in the field of semiconductor and microelectronics industry in the form of efficient power electronic devices and digital signal processors (DSPs) [2], [3]. AC motor drives mainly drive mechanical loads in the industry due to its free from the limitations of mechanically commutated DC drives [4]. The stator and rotor of an AC motor are the only contact-bearing components. The rotation of the rotor is caused by the magnetic field of the stator and requires more complex control technology to implement its motion systems [5], [6]. Through the evolution of semiconductor control devices and the course of the evolution of different types of AC motor control algorithms, the computational requirements for AC motor control can be satisfied easily. This advantage has renewed variable-speed AC motor drives and has contributed actively to the field of control engineering [7].

Interior Permanent Magnet Synchronous Motors (IPMSMs) are emerging in various applications replacing traditional DC and induction motors due to their small volume, lightweight, low losses, high efficiency, high power density, and fast dynamic performance [8]. They have been used in robotics, drivetrain, wind turbine, elevator, compressors, air conditioners, washing machines [9], etc. IPMSMs have additional reluctance torque and are compatible with sensorless techniques [10], [11].

Although IPMSM drives have been a quite popular choice of drives for research explorations in the last decades, still there are many avenues, especially from the control point of view, which have been wide open for research [7]. The IPMSM drives should achieve high control performance such as fast dynamic response, high trajectory tracking accuracy, and insensitivity to disturbances and parameter

perturbations [12]. Also, the designed control algorithms should have a low computational burden for implementation [13]. However, some challenging tasks are owing to various factors (e.g., nonlinear motor parameters, model parameter uncertainties, external disturbances, etc.) because these factors can severely degrade the trajectory tracking performance of the IPMSM drives [14], [15].

It is well known that linear control schemes, e.g., the Proportional Integral (PI) control scheme, are already widely applied in the IPMSM drives due to their relatively simple implementation [16]. However, the IPMSM system is a nonlinear system with unavoidable and unmeasured disturbances, as well as parameter variations [17]. This makes it very difficult for linear control algorithms to obtain a sufficiently high-performance for this kind of nonlinear system. It is reported that the PI control algorithm cannot assure a satisfying dynamic behavior in the entire operating range [18], [19]. Nonlinear control methods seek to provide more natural and effective solutions to the problems associated with substantially perturbed IPMSM systems.

Nonlinear control algorithms thus become a natural solution for controlling the IPMSM. Recently, with the rapid progress in power electronics, microprocessors, especially DSPs, and modern control theories, many researchers have aimed to develop nonlinear control methods for the IPMSM, and various algorithms have been proposed, e.g., input-output linearization control [18], adaptive control [20-22], robust control [23], sliding mode control (SMC) [24], [25], backstepping control [26], [27], and intelligent control [19], [25], [28]. Generally speaking, these nonlinear control techniques improve the performance, robustness, and stability of the IPMSM drive in closed-loop.

However, in real industrial applications, IPMSM drives are always confronted with different disturbances. These disturbances may come internally, e.g., friction force and unmodeled dynamics, or externally, e.g., load disturbances [2]. Conventional feedback-based control methods usually cannot react directly and fast to reject these disturbances, although these control methods can finally suppress them through feedback regulation in a relatively slow way. This results in a degradation of system performance when meeting severe disturbances [29].

Among the existing nonlinear control methods, the backstepping control is one of the most efficient methods to design controllers for nonlinear systems [14]. This method provides a powerful control designing tool for nonlinear systems [30].

Conventional backstepping is applied successfully to the control of IPMSMs recently [31]. The most appealing point of the backstepping scheme is to use the virtual control variable to make the original high order system to be simple enough. Thus, the final control outputs can be derived step by step through the Lyapunov functions [14], [32], [33]. However, this control approach suffers from speed static error when the IPMSM operates in a steady-state and is easily affected by the load torque disturbance [34], [35]. Consequently, there still exists a gap to achieve a remarkable improvement in speed tracking performance. To overcome these problems, this thesis proposes an improved backstepping control scheme that can effectively enhance the speed tracking performance in the transient and steady-state for the IPMSM drives. The proposed integral backstepping controller (IBC) introduces the integral of the tracking errors into the control law. Then by constructing the appropriate Lyapunov function, the control law with the integral terms is derived to eliminate the speed static error, weaken the influence of internal parameters perturbation and load disturbance on the motor drives.

There is still a problem inherent in the conventional backstepping control scheme, which is the explosion of complexity caused by the repeated differentiation of virtual input [36]. Theoretically, the calculation of virtual control derivation is simple, but it can be quite tedious and complicated in practical applications [15], [31]. To overcome this issue, a Dynamic Surface Control (DSC) method was used by introducing a first order filtering of the virtual input at each step of the conventional backstepping approach [33], [37], [38].

This thesis proposes a DSC method to solve the above problem of the conventional backstepping control for IPMSM drive systems, the controller is complemented with an online estimation scheme of the lumped effects related to unknown load torque disturbances, friction forces, and unmodeled dynamics. The approach uses a Feedforward Neural Network (FFNN) observer. Although methods used in previous works present satisfactory results [2], [12], [39], the objective is to present a simpler Artificial Neural Network (ANN) based observer, providing the used training data, simulation, and experimental results, without using online training. The main idea of the work presented here is to estimate the unknown load torque disturbances, friction forces, and unmodeled dynamics, then the information obtained from the FFNN observer adapted to the DSC method in a manner that the drive system

has a low computational cost and be of easy implementation, exploiting the advantages of DSC and ANNs mentioned in the literature. In this thesis, the simulations are performed using MATLAB/Simulink. Real-time implementations are carried out using a DSP board, DS1104 for an available laboratory 1.5 hp motor.

1.2 RESEARCH OBJECTIVES

The goal of this thesis is to design and implement high-performance control architecture for IPMSM drives. The main emphasis of this research is laid on:

- ***i.*** The design of robust nonlinear controllers for IPMSM drives to achieve high-performance in the existence of parametric variations and load torque disturbances and to solve the problems associated with conventional control methods;
- ***ii.*** Development of ANN-based load torque observer to estimate the load torque and unmodelled dynamics in order to make the system robust against disturbances and uncertainties;
- ***iii.*** The design of control algorithms with a low computational burden that are easy to implement in a low-cost microprocessor;
- ***iv.*** The last objective of this thesis is the simulation and the real-time implementation of the control algorithms developed earlier.

1.3 THESIS STRUCTURE

This thesis consists of five chapters:

Chapter 1 provides the motivation of the thesis. Next, the research objectives and the thesis structure are outlined.

Chapter 2 deals with the design of the IPMSM field-oriented control (FOC). This chapter was divided into six sections. It starts with an introduction and then the general theory of IPMSMs, with a focus on different rotor types and their properties is explained. Next, an overview of the existed control techniques for IPMSM drives is presented. Then, the mathematical model of the IPMSM is presented. After that, the chapter presents the PI controller design. Finally, the conclusion of this chapter is given.

In **Chapter 3**, a modified backstepping controller is presented. This chapter was divided into six sections. The first section is an introduction. The second section presents the general strategy for controlling the IPMSM. The IBC is introduced, built on the aforementioned mathematical model of the IPMSM, based on the Lyapunov criterion. The third section gives the controllers tuning. Simulation and experimental results are shown in sections 4 and 5 respectively, the static and dynamic performance of the proposed IBC and PI controller are examined through different operating conditions such as sudden command speed change, load disturbances, etc. A performance comparison is also provided. The last section of this chapter is a conclusion.

Chapter 4 addresses the design of the DSC method for the IPMSM system and the use of ANN to estimate load torque. This chapter has seven sections. In section 1, an introduction is presented. Next in section 2, a model-based observer approach is developed which would generate the training samples for the ANN-based observer. Section 3 gives an introduction to a FFNN Observer and its training to emulate the Load Torque Observer. In Section 4, the control design law based on DSC control theory for IPMSM systems is developed. In section 5, the tuning of observer and controllers is given. Section 6 presents the simulation results. In order to verify the advantages of the DSC technique over the conventional backstepping control technique, a comparison between the two methodologies was studied. Chapter 4 ends with a conclusion.

Finally, a summary of the thesis and suggestions for future work are highlighted in **chapter 5**. After that, appendices and all pertinent references are listed.

Chapter 2: Field-Oriented Control of Interior Permanent Magnet Synchronous Motor

2.1 INTRODUCTION

In this chapter, the FOC of the IPMSM is presented. In section 2.2 the permanent magnet (PM) materials and the classification of permanent magnet synchronous motors (PMSMs) are given. After that, a brief literature review of control techniques for IPMSM drives is discussed in section 2.3, which is related to the main studies in this thesis; these are control schemes for speed control systems for IPMSMs. In section 2.4, the mathematical model of the motor is presented with its reference frame first oriented in the stator (*a, b, c*) stationary frame then in the rotor (*d, q, o*) synchronous frame. PI controller for IPMSM speed control is detailed in section 2.5. The chapter ends with a conclusion.

2.2 PERMANENT MAGNET SYNCHRONOUS MOTOR

2.2.1 Permanent Magnet Materials

The permanent magnet motor develops with the discovery of new PM materials. One of the most important specifications to measure the quality of PM materials is its maximum achievable energy product. The larger this parameter is, the less PM material is needed for the same magnet force.

Alnico magnets, which are Aluminum, Nickel, and iron composite, were the first largely used PM material in history. It has high residual flux density, excellent temperature stability, and a strong corrosion resistance level. However, it can easily be demagnetized and its maximum energy product is not very high.

Ferrite magnets are another significant PM material discovery. Since it doesn't contain these noble metal elements, it is one of the cheapest PM materials ever found. It has relatively high intrinsic coercive force, which shows higher resistant demagnetization capability. Its lower mass density reduces the weight of its product. Its main drawback is the limited residual flux density and thus its lower maximum energy density.

Rare earth magnets are one of the most significant findings in the last century, which finally lead to the wide application of PMSMs. It has high coercive force as well as a large residual flux density. There are two types of rare earth magnets: one uses high-cost Samarium and Cobalt (SmCo) material. Another is NdFeB (Neodymium, Iron, and Boron), though with lower limited working temperature, but with even higher residual flux density and energy density. Table 2.1 summarizes all the specifications for different PM materials addressed above [40].

Table 2. 1. Typical permanent magnet material properties

Materials	B_r [T]	H_c [kA/m]	BH_{max} [kJ/m^3]	T_c [^{0}C]	T_{w-max} [^{0}C]
Alnico	1.2	10	6	500	500
Ferrite	0.43	10	5	300	300
SmCo	Up to 1.1	Up to 820	Up to 240	Up to 820	Up to 350
NdFeB	Up to 1.5	Up to 1033	Up to 422	Up to 380	Up to 200

where:

- B_r is the remanence, which measures the strength of the magnetic field;
- H_c is the coercivity, presenting the PM material resistance to becoming demagnetized;
- BH_{max} is the maximum energy product, indicating the density of magnetic energy, characterized by the maximum value of magnetic flux density (B) times magnetic field strength (H);
- T_c is the Curie temperature, above this temperature, the PM material irreversibly loses its magnetism;
- T_{w-max} is the maximum operating temperature, above this temperature, the PM material reversibly loses its magnetism.

2.2.2 Classification of Permanent Magnet Synchronous Motors

The physical characteristics of the PMSM are associated with its rotor and stator structures. The stator is composed of a three-phase wound such that the Electromotive Forces (EMFs) are generated by the rotation of the rotor field. Furthermore, the EMF can be sinusoidal or trapezoidal. This wound is represented by the three axes (a, b, c) phase-shifted, one from the other, by 120 electrical degrees.

The rotor incorporates PMs to produce a magnetic field. Regarding winding, the PMs have the advantage to eliminate the brushes, the rotor losses, and the need for a controlled DC source to provide the excitation current. However, the amplitude of the rotor flux is constant.

On the other hand, there exist several ways to place the magnets in the rotor [41]. Following the magnet position, the PMSM can be classified into four major types:

Surface-mounted magnets type

The magnets are placed on the surface of the rotor using high-strength glue. They present a homogeneous gap; the motor is a non-salient pole. The inductances do not depend on the rotor position (Figure 2.1(a)). The inductance of the *d*-axis is equal to that of the *q*-axis. This rotor configuration is simple to obtain. This type of rotor is the most usual. On the other hand, the magnets are exposed to a demagnetizing field. Moreover, they are subject to centrifuge forces which can cause the detachment of the rotor.

Inset magnets type

The inset magnets are placed on the surface of the rotor. However, the space between the magnets is filled with iron (see Figure 2.1(b)). The alternation between the iron and the magnets causes a salient effect. The inductance in the *d*-axis is slightly different from the inductance in the *q*-axis.

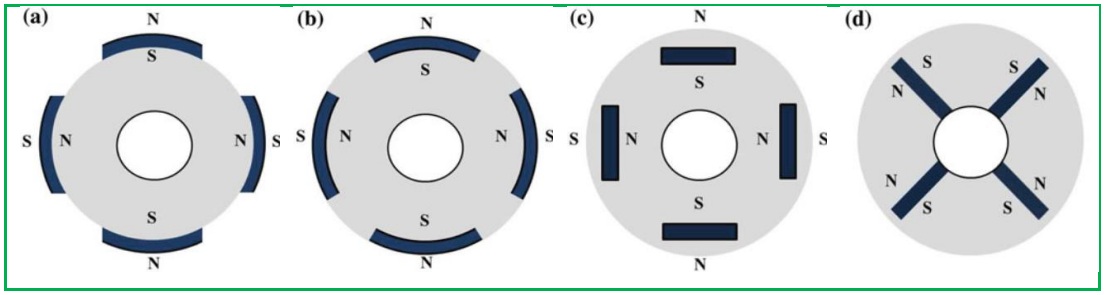


Figure 2. 1. PMSM rotor permanent magnets layout: (a) surface permanent magnets, (b) inset permanent magnets, (c) interior permanent magnets, (d) flux concentrating.

Interior magnets type

The magnets are integrated into the rotor's body (Figure 2.1(c)): the motor is a salient pole type. In this case, the rotor magnetism is anisotropic, the inductances depend on the rotor position. The magnets are placed in the rotor, providing more mechanical durability and robustness at high speeds. On the other hand, this motor is more expensive to manufacture and more complex to control.

Flux concentrating type

As shown in Figure 2.1(d), the magnets are deeply placed in the rotor's body. The magnets and their axes are radial. The flux on a polar arc of the rotor is a result of two separated magnets. The advantage of this configuration is the possibility to concentrate the flux generated by the PMs in the rotor and obtain a stronger induction in the gap. This type of machine has a salience effect.

However, surface permanent magnet synchronous motors (SPMSMs) and IPMSMs are the most used in the industry. Furthermore, the PMSMs can be classified according to the EMF profiles [42], [43]:

- Sinusoidal;
- Trapezoidal.

Particularly, the synchronous machines with sinusoidal EMF are classified into two subcategories in terms of magnets position:

Non-salient Poles

The magnets are located on the rotor surface (Figure 2.1(a)): Surface Permanent Magnet Synchronous Motor.

Salient Poles

The magnets are buried into the rotor (Figure 2.1(c)): Interior Permanent Magnet Synchronous Motor.

A further classification can be made based on the saliency ratio, its sorting in Figure 2.2 is obtained.

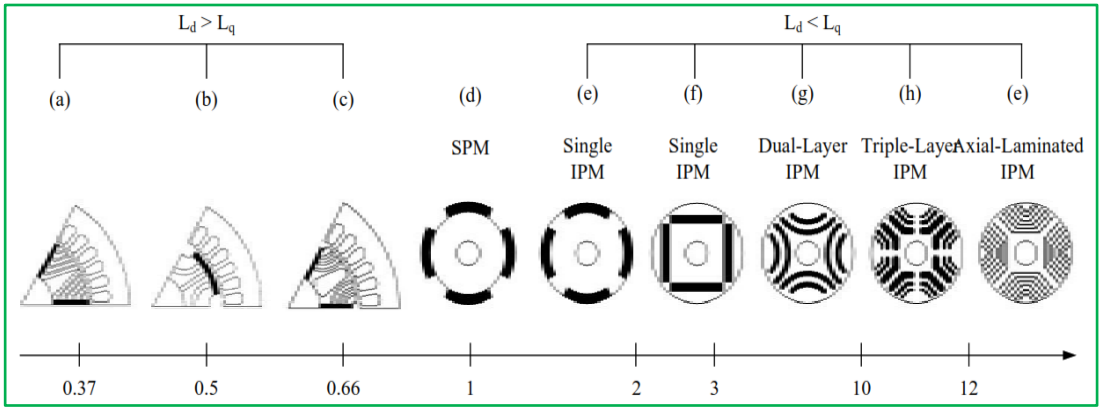


Figure 2. 2. PMSM with different saliency ratio.

The IPMSM, one category of PM machines based on the classification, is superior to other types of PMSMs because it combines both the characteristics of the SPMSM and the Synchronous Reluctance Machine (SynRM). Its performance is enhanced by both the PM torque at low speed and reluctance torque at the high speed field-weakening region [44]. The work in this thesis concentrates on the development of control algorithms based on this machine.

2.3 CONTROL TECHNIQUES OVERVIEW

Vector control or FOC is the most effective method for good performance control also more relevant to PMSM because the control is carried out through the variables of the stator side since the rotor side control is not possible [26], [45]. The main advantages of vector control of IPMSM using the conventional PI controller are simple structure, easy implementation, and reliability. The basic idea of vector control of IPMSM is decoupling the speed and currents, hence they can be controlled separately [46]. However, its control performance is affected by unpredictable parameter variations, external load disturbances, and nonlinear dynamics. So, it is hard to achieve a satisfactory performance by only using linear control algorithms, especially in complex working conditions. Therefore, various nonlinear control algorithm methods have been proposed to improve the control performance of IPMSM. In the following, we will review some of them [47].

In recent years, artificial intelligence techniques such as fuzzy logic and ANNs are gaining more attention in the field of PMSM control due to their many advantages over conventional methods. A fuzzy PI-type current controller is used in [48], which stabilizes the decoupled dynamics and ensures the stability of the PMSM. But this approach has some problems dealing with the time-varying uncertainties. In [28], a speed control integrated circuit is proposed under a system-on-a-programmable-chip environment for the PMSM drive. Here, the external load effect and dynamic uncertainty in the speed loop of the PMSM are considered for the operation of an adaptive fuzzy controller. A decentralized adaptive fuzzy controller for the PMSM drive is used in [49], based on the type-2 fuzzy logic systems and adaptive control theory. An ANN-based real-time adaptive controller is proposed in [50], which initially generates estimated coefficients using an offline training method. Then, to update the ANN, online training is executed using the dynamic backpropagation with the Levenberg-Marquardt algorithm. In [31], the unknown and nonlinear functions of

the PMSM are approximated using neural networks and an adaptive DSC is proposed. A speed control strategy for a six-phase PMSM is proposed in [51] using a robust adaptive backstepping sliding mode control with a recurrent wavelet fuzzy neural network considering the load disturbances and parameter variations. In [52], [53], a recurrent fuzzy neural cerebellar model articulation network is considered for a six-phase PMSM position servo drive system, which is also equipped with a fault-tolerant control scheme.

The Active Disturbances Rejection Controller (ADRC) was developed by the researcher Han Jingqing of the Chinese Academy of Sciences following in-depth research into modern control theory [54]. It combines Proportional Integral Derivative (PID) control technology based on error feedback, using this to eliminate the essence of error control, and thereby proposing a new, nonlinear, practical control method [55]. The ADRC suppresses overshoot using the Tracking Differentiator (TD) design transition process during operation and observes the external disturbance and parameter variation of the system through the Extended State Observer (ESO) [56]. The ESO accounts for the defects of the PID controller and results in the accelerated convergence of any error, as well as possessing desirable dynamic and static characteristics [57].

The SMC is one of the effective control methods since it is insensitive to parameter uncertainties and external disturbances, which can ensure excellent speed tracking performance [58]. However, the robustness of the SMC can only be guaranteed by the selection of large control gains, while large gains will cause the chattering phenomenon, which can provoke high-frequency dynamics [59]. In order to eliminate the chattering problem, different approaches have been proposed, such as high order SMC [60], integral sliding surface method [61], and reaching law method [62].

Model Reference Adaptive Control (MRAC), which is often used for estimating state variables of nonlinear control systems during periods when the measurements of the related state variables are not available for feedback [63], based nonlinear speed control of IPMSM was proposed in [64]. The controller was developed in the sense of the input to output feedback linearization scheme, the utilization of which may cause to cancel some useful nonlinearity [26]. The partial plant uncertainties; load torque disturbance and PM flux, are taken into account and this, in turn, results in that the

exact values of the other plant parameters are required to be known in the controller design. As another consequence of that, the control system gives unsatisfactory performance if these known parameters vary because of the different operating conditions such as variation of temperature, saturation and external disturbance [14], etc.

The H_∞ control is a powerful method for robust control of the IPMSM in the existence of parameter uncertainties and external disturbances. The main objective of the H_∞ control is to synthesize a controller that makes the closed-loop system satisfy a prescribed H_∞ norm constraint that represents the desired stability or tracking requirements [65]. However, to guarantee good performance robustness under parameter uncertainties and external disturbances, the H_∞ design gives usually a solution with high control gain, the use of this approach is not feasible in practice [66].

Feedback linearization control has been widely applied to IPMSM drives, whereby the original nonlinear motor model has been converted to a linear model through proper selection of coordinate transformation. But this method requires the exact mathematical model. However, the conventional feedback linearization method may fail to meet the high-performance requirements of systems because it is very sensitive to parameter uncertainties and unknown disturbances [67].

Yucelen et al. studied the controller based on the state-dependent Riccati equation, which is also called nonlinear quadratic optimal control [68]. The controller requires the solution of an algebraic Riccati equation [69], hence; has a relatively complex formulation and requires a significant computational effort.

Backstepping is a well-known recursive and systematic design methodology for feedback control of the uncertain nonlinear system with matched uncertainties [70]. The key point is to use the virtual control variable to simplify the original high order system; thus, the final controller can be derived by Lyapunov's stable theorem. In [71], an adaptive backstepping speed controller has been proposed for the speed control of PMSM. The controller is robust against stator resistance, viscous friction uncertainties, and load torque disturbances. However, this approach uses feedback linearization. Using a nonlinear and adaptive backstepping design method, a speed and current control scheme for PMSM was presented in [72], in which all parameters of PMSM and load dynamics were considered unknown. In [14], a new nonlinear and fully adaptive backstepping speed tracking control scheme was developed for uncertain

PMSM. Except for the number of pole pairs, all other parameters of PMSM and load dynamics were assumed to be unknown.

However, there are some drawbacks in the conventional backstepping control method, such as steady-state errors, sensitivity to parameter uncertainties, and load torque disturbances [2]. In order to improve the control performance, the integral of the tracking error is introduced in the backstepping control. The design of an integral backstepping controller (IBC) for the IPMSM control system has been reported in [73-75].

2.4 CONVENTIONAL MATHEMATICAL AND CIRCUITAL MODELLING THEORY OF IPMSM

2.4.1 IPMSM Mathematical Model in the Stator Reference Frame

In order to study the electric and magnetic behavior of the IPMSM in several working conditions, it is necessary to know the dynamic and mathematical machine model and its electrical and magnetic parameters. In literature, several dynamic and mathematical IPMSM models are described [76-78]. In this chapter, the conventional dynamic and mathematical machine models are presented. The simplified hypothesis at the base of the conventional dynamic and mathematical IPMSM model are:

- Sinusoidal spatial distribution of the Magnetomotive Force (MMF) in the air gap;
- Linear magnetic behavior of the machine;
- Absence of hysteresis and eddy currents in the iron core;
- Absence of cross-coupling effects;
- Absence of slot harmonics effects;
- Absence of the temperature effects on the machine.

Referring to a three-phase IPMSM, the general voltage equations are:

$$\begin{cases} v_a = Ri_a(t) + \dfrac{d\phi_a(t,\theta_e)}{dt} \\ v_b = Ri_b(t) + \dfrac{d\phi_b(t,\theta_e)}{dt} \\ v_c = Ri_c(t) + \dfrac{d\phi_c(t,\theta_e)}{dt} \end{cases} \tag{2.1}$$

where θ_e is the rotor electrical position (given by the product from the number of the pole pair P and the rotor mechanical position θ_r), and:

- v_a, v_b, v_c are the instantaneous phase voltages;
- i_a, i_b, i_c are the instantaneous phase currents;
- ϕ_a, ϕ_b, ϕ_c are the instantaneous magnetic phase flux linkage;
- R is the phase resistance, which is supposed to be equal for all three-phase.

Flux linkage to the stator phases are both a function of the currents circulating in each stator phase and of the magnetic flux emanating from the rotor PMs. It is assumed that the rotor and the stator of the machine exhibit constant magnetic permeability, or rather the linear magnetic behavior and that the losses in the machine iron are negligible. The flux linkage can be expressed with the following relationships:

$$\begin{cases} \phi_a(t) = \phi_{a,i}(t) + \phi_{fa}(t) \\ \phi_b(t) = \phi_{b,i}(t) + \phi_{fb}(t) \\ \phi_c(t) = \phi_{c,i}(t) + \phi_{fc}(t) \end{cases} \tag{2.2}$$

where $\phi_{n,i}$ is the flux linkage produced by stator current excitation, and ϕ_{fn} is the flux linkage produced by PMs.

Based on the model hypothesis, it is possible to assume sinusoidal the flux linkage ϕ_{fa}, ϕ_{fb}, and ϕ_{fc}:

$$\begin{cases} \phi_{fa} = \phi_f \cos P\theta_r \\ \phi_{fb} = \phi_f \cos\left(P\theta_r - \frac{2}{3}\pi\right) \\ \phi_{fc} = \phi_f \cos\left(P\theta_r - \frac{4}{3}\pi\right) \end{cases} \tag{2.3}$$

where ϕ_f represents the peak value of flux linkage of each phase due to the PMs.

The flux linkage produced by stator current excitation can be expressed with the following relationships:

$$\begin{cases} \phi_{a,i} = L_{aa}i_a + L_{ab}i_b + L_{ac}i_c \\ \phi_{b,i} = L_{ba}i_a + L_{bb}i_b + L_{bc}i_c \\ \phi_{c,i} = L_{ca}i_a + L_{cb}i_b + L_{cc}i_c \end{cases} \tag{2.4}$$

where L_{aa}, L_{bb}, and L_{cc} are the self-inductances of the three phases, L_{ab}, L_{bc}, and L_{ca} are the mutual-inductances between one phase to the other ones.

The auto and mutual-inductances can be considered dependent on the rotor position and then also on the time. The self-inductances can be expressed as follow:

$$\begin{cases} L_{aa} = L_{\sigma s} + L_{ms0} + L_m \cos 2P\theta_r \\ L_{bb} = L_{\sigma s} + L_{ms0} + L_m \cos\left(2P\theta_r - \frac{2}{3}\pi\right) \\ L_{cc} = L_{\sigma s} + L_{ms0} + L_m \cos\left(2P\theta_r - \frac{4}{3}\pi\right) \end{cases} \quad (2.5)$$

That is given by the sum of a sinusoidal term with a double electrical frequency (of amplitude L_m) and a constant term ($L_{\sigma s} + L_{ms0}$). The constant $L_{\sigma s}$ represents the leakage inductance and it is relative to the stator flux which encloses in the air without interesting the rotor. L_{ms0} and L_m represent, the constant component and the amplitude of the anisotropy component ($2P\theta_r$ angle function) of the magnetization inductance, respectively. The mutual-inductances take into account the flux linked by each winding due to the current flowing in the other windings. These also depend on the anisotropy of the machine and can be expressed through the following relationships:

$$\begin{cases} L_{ab} = L_{ba} = -M_s + L_m \cos\left(2P\theta_r - \frac{2}{3}\pi\right) \\ L_{bc} = L_{cb} = -M_s + L_m \cos\left(2P\theta_r\right) \\ L_{ca} = L_{ac} = -M_s + L_m \cos\left(2P\theta_r - \frac{4}{3}\pi\right) \end{cases} \quad (2.6)$$

where M_s is the average coefficient of mutual induction between the phases of the stator winding.

The relationships described above can also be expressed in matrix form as follows:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d\begin{bmatrix} \phi_a \\ \phi_b \\ \phi_c \end{bmatrix}}{dt} \quad (2.7)$$

$$\begin{bmatrix}\phi_a\\ \phi_b\\ \phi_c\end{bmatrix}=\begin{bmatrix}L_{aa} & L_{ab} & L_{ac}\\ L_{ba} & L_{bb} & L_{bc}\\ L_{ca} & L_{cb} & L_{cc}\end{bmatrix}\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix}+\begin{bmatrix}\phi_{fa}\\ \phi_{fb}\\ \phi_{fc}\end{bmatrix} \tag{2.8}$$

The expressions (2.7) and (2.8) can also be expressed in compact form:

$$[v]=[R][i]+\frac{d[\phi]}{dt} \tag{2.9}$$

$$[\phi]=[L][i]+\phi_f \tag{2.10}$$

Replacing the equation (2.8) in the equation (2.7) of the electric voltage, it is possible to obtain the following relationship:

$$\begin{aligned}[v]&=[R][i]+\frac{d}{dt}\left([L][i]+\phi_f\right)\\ &=[R][i]+\frac{d[L]}{dt}[i]+\frac{d[i]}{dt}[L]+\frac{d[\phi_f]}{dt}\\ &=[R][i]+\frac{d[\theta_r]}{dt}\frac{d[L]}{d\theta_r}[i]+\frac{d[i]}{dt}[L]+\frac{d[\theta_r]}{dt}\frac{d[\phi_f]}{d\theta_r}\\ &=[R][i]+\omega_r\frac{d[L]}{d\theta_r}[i]+[L]\frac{d[i]}{dt}+\omega_r\frac{d[\phi_f]}{d\theta_r}\end{aligned} \tag{2.11}$$

where:

- $[R][i]$ are the drop voltages component in each phase (a, b, c);
- $[L]\frac{d[i]}{dt}$ are the back-EMFs auto and mutually induced in the stator windings;
- ω_r is the rotor mechanical speed;
- $\omega_r\frac{d[L]}{d\theta_r}[i]$ are the rotational back-EMFs due to the anisotropy of the machine;
- $\omega_r\frac{d[\phi_f]}{d\theta_r}$ are the rotational back-EMFs induced by the flux of the PMs.

Multiplying both members of equation (2.11), for the transposed vector of the currents $[i]^T$, it is possible to obtain the expression of the input power of IPMSM:

$$[P_e]=[i]^T[v]$$

$$=[i]^T[R][i]+[i]^T[L]\frac{d[i]}{dt}+\omega_r[i]^T\frac{d[L]}{d\theta_r}[i]+\omega_r[i]^T\frac{d[\phi_f]}{d\theta_r} \tag{2.12}$$

where:

- $[P_e]=[i]^T[v]$ represents the input electric power absorbed by the machine;
- $[P_{cu}]=[i]^T[R][i]$ represents the joule losses in the stator windings;
- $[i]^T[L]\frac{d[i]}{dt}$ represents the power stored in the magnetic field;
- $[P_m]=\omega_r[i]^T\frac{d[L]}{d\theta_r}[i]+\omega_r[i]^T\frac{d[\phi_f]}{d\theta_r}$ represents the mechanical power of the motor shaft.

Therefore, it is also possible to obtain the expression of the electromechanical torque to the shaft with the following relation:

$$T_{em}=\frac{P_m}{\omega_r}$$

$$=[i]^T\frac{d[L]}{d\theta_r}[i]+[i]^T\frac{d[\phi_f]}{d\theta_r}$$

$$=[i_a i_b i_c]\frac{d}{d\theta_r}\begin{bmatrix}L_{aa} & L_{ab} & L_{ac}\\ L_{ba} & L_{bb} & L_{bc}\\ L_{ca} & L_{cb} & L_{cc}\end{bmatrix}\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix}+[i_a i_b i_c]\frac{d}{d\theta_r}\begin{bmatrix}\phi_{fa}\\ \phi_{fb}\\ \phi_{fc}\end{bmatrix} \tag{2.13}$$

where:

- $[i_a i_b i_c]\frac{d}{d\theta_r}\begin{bmatrix}L_{aa} & L_{ab} & L_{ac}\\ L_{ba} & L_{bb} & L_{bc}\\ L_{ca} & L_{cb} & L_{cc}\end{bmatrix}\begin{bmatrix}i_a\\ i_b\\ i_c\end{bmatrix}$ represents the reluctance torque component, or the torque developed by the machine in the absence of PMs;

- $[i_a i_b i_c]\frac{d}{d\theta_r}\begin{bmatrix}\phi_{fa}\\ \phi_{fb}\\ \phi_{fc}\end{bmatrix}$ represents the torque component developed by the machine through the interaction between the flux produced by the PMs and the currents circulating in the stator phases.

The mathematical model of the IPMSM thus found consists of a system of differential equations with non-constant coefficients, whose solutions are not easy to determine. This difficulty is overcome by resorting to a mathematical artifice that consists in transforming the coordinates from the (*a*, *b*, *c*) stator reference frame system to the (*d*, *q*, *o*) reference frame system rotating at synchronism.

2.4.2 IPMSM Mathematical Model in (*d*, *q*, *o*) Synchronous Reference Frame

The mathematical model in the stator reference frame system determined in the previous paragraph is formed by three differential equations with variable coefficients. The use of such a mathematical model to control an IPMSM is not simple to implement. To obtain differential equations with constant parameters, it is possible to use linear mathematical transformations. Specifically, the Park transformation allows the transition from a three-phase stator reference system to a two-phase reference frame system, that is rotating at the mechanical rotor speed ω_r and with the real axis coincident with the polar rotor axis. In particular, this transformation has the advantage of being reduced to a system of equations in the reference (*d*, *q*, *o*) where the coefficients of self and mutual inductances are independent of the rotor position and therefore constant over time.

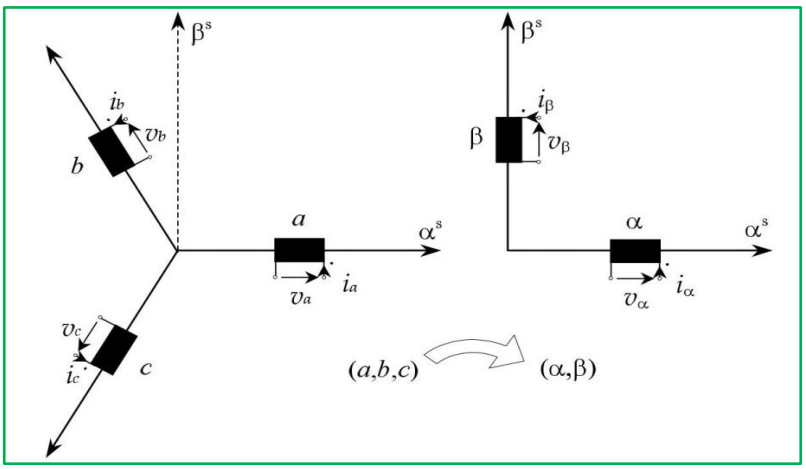


Figure 2. 3. Transformation of coordinates from the stator reference system (*a*, *b*, *c*) to the fixed two-phase reference system (*α*, *β*)

To obtain the Park model, a first coordinate transformation is carried out from the stator reference system (*a*, *b*, *c*) to the fixed two-phase reference system (*α*, *β*), as shown in Figure 2.3. With such a transformation is obtained a system that which there are only two windings arranged at 90° and therefore, magnetically decoupled. This model has the stator and rotor quantities evaluated regarding the corresponding physical reference frame, i.e., the inductive coefficients are still a function of the angular position of the rotor. Therefore, a further transformation is needed which allows expressing stator and rotor quantities under a single reference. A system of axes (*d*, *q*, *o*) rotating integrally with the rotor is considered, where the rotor quantities are referred to this system. These further coordinates transformation, where the stator quantities referred to the system (*α*, *β*) are reported to the system (*d*, *q*), as shown in Figure 2.4.

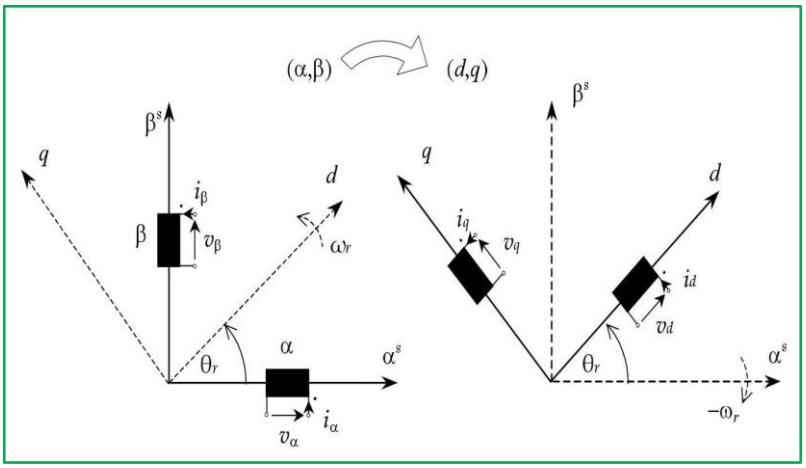


Figure 2. 4. Transformation of coordinates from the fixed two-phase reference system (*α*, *β*) to the rotating (*d*, *q*, *o*) system.

The transformation of the quantities from the three-phase reference system (*a*, *b*, *c*) to the reference system (*d*, *q*, *o*) is carried out through the following transformation matrix:

$$[T]=\frac{2}{3}\begin{bmatrix} \cos P\theta_r & \cos\left(P\theta_r-\frac{2}{3}\pi\right) & \cos\left(P\theta_r-\frac{4}{3}\pi\right) \\ -\sin P\theta_r & -\sin\left(P\theta_r-\frac{2}{3}\pi\right) & -\sin\left(P\theta_r-\frac{4}{3}\pi\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{2.14}$$

It is also possible to define the inverse matrix which allows the transformation of the quantities from the coordinate system (*d*, *q*, *o*) to the three-phase coordinate system (*a*, *b*, *c*):

$$[T]^{-1} = \frac{2}{3}\begin{bmatrix} \cos P\theta_r & -\sin P\theta_r & 1 \\ \cos\left(P\theta_r - \frac{2}{3}\pi\right) & -\sin\left(P\theta_r - \frac{2}{3}\pi\right) & 1 \\ \cos\left(P\theta_r - \frac{4}{3}\pi\right) & -\sin\left(P\theta_r - \frac{4}{3}\pi\right) & 1 \end{bmatrix} \tag{2.15}$$

The electrical quantities in the (*d*, *q*, *o*) reference system are defined with the following relationships:

$$\left[v_{d,q,o}\right] = [T]\left[v_{a,b,c}\right] = \frac{2}{3}\begin{bmatrix} \cos P\theta_r & \cos\left(P\theta_r - \frac{2}{3}\pi\right) & \cos\left(P\theta_r - \frac{4}{3}\pi\right) \\ -\sin P\theta_r & -\sin\left(P\theta_r - \frac{2}{3}\pi\right) & -\sin\left(P\theta_r - \frac{4}{3}\pi\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \tag{2.16}$$

$$\left[i_{d,q,o}\right] = [T]\left[i_{a,b,c}\right] = \frac{2}{3}\begin{bmatrix} \cos P\theta_r & \cos\left(P\theta_r - \frac{2}{3}\pi\right) & \cos\left(P\theta_r - \frac{4}{3}\pi\right) \\ -\sin P\theta_r & -\sin\left(P\theta_r - \frac{2}{3}\pi\right) & -\sin\left(P\theta_r - \frac{4}{3}\pi\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{2.17}$$

$$\left[\phi_{d,q,o}\right] = [T]\left[\phi_{a,b,c}\right] = \frac{2}{3}\begin{bmatrix} \cos P\theta_r & \cos\left(P\theta_r - \frac{2}{3}\pi\right) & \cos\left(P\theta_r - \frac{4}{3}\pi\right) \\ -\sin P\theta_r & -\sin\left(P\theta_r - \frac{2}{3}\pi\right) & -\sin\left(P\theta_r - \frac{4}{3}\pi\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} \phi_a \\ \phi_b \\ \phi_c \end{bmatrix} \tag{2.18}$$

Applying the aforementioned transformation of coordinates to the IPMSM mathematical model, the flux equation is:

$$\left[\phi_{d,q,o}\right]=[T]\left[\phi_{a,b,c}\right]$$

$$=[T][L]\left[i_{a,b,c}\right]+[T]\left[\phi_{fa,b,c}\right]$$

$$=[T][L][T]^{-1}\left[i_{d,q,o}\right]+[T][T]^{-1}\left[\phi_{fd,q,o}\right]$$

$$=\begin{bmatrix} L_d & 0 & 0 \\ 0 & L_q & 0 \\ 0 & 0 & L_o \end{bmatrix}\begin{bmatrix} i_d \\ i_q \\ i_o \end{bmatrix}+\begin{bmatrix} \phi_f \\ 0 \\ 0 \end{bmatrix} \tag{2.19}$$

where:

- v_d, v_q, v_o are the *d*, *q*, and *o* axis voltages;
- i_d, i_q, i_o are the *d*, *q*, and *o* axis currents;
- ϕ_d, ϕ_q, ϕ_o are the *d*, *q*, and *o* axis fluxes;
- L_d, L_q, L_o are the *d*, *q*, and *o* axis inductances.

where each inductance component is:

$$L_d = L_s + M_s + \frac{3}{2}L_m \tag{2.20}$$

$$L_q = L_s + M_s - \frac{3}{2}L_m \tag{2.21}$$

$$L_o = L_s - 2M_s \tag{2.22}$$

$$L_s = L_{\sigma s} - L_{mso} \tag{2.23}$$

where L_s is the self-inductance.

The voltage balance equations results:

$$\left[v_{d,q,o}\right]=[T]\left[v_{a,b,c}\right]$$

$$=[T][R]\left[i_{a,b,c}\right]+[T]\frac{d\left[\phi_{a,b,c}\right]}{dt}$$

$$=[T][R][T]^{-1}\left[i_{d,q,o}\right]+[T]\frac{d\left([T]^{-1}\left[\phi_{d,q,o}\right]\right)}{dt}$$

$$= \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_o \end{bmatrix} + [T][T]^{-1} \frac{d \begin{bmatrix} \phi_d \\ \phi_q \\ \phi_o \end{bmatrix}}{dt} + [T] \omega_r \frac{d\left([T]^{-1}\right)}{d\theta_r} \begin{bmatrix} \phi_d \\ \phi_q \\ \phi_o \end{bmatrix}$$

$$= \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_o \end{bmatrix} + \frac{d \begin{bmatrix} \phi_d \\ \phi_q \\ \phi_o \end{bmatrix}}{dt} + P\omega_r \begin{bmatrix} -\phi_q \\ \phi_d \\ 0 \end{bmatrix} \tag{2.24}$$

Replacing equation (2.19) in equation (2.24), the voltage equation became:

$$\begin{bmatrix} v_d \\ v_q \\ v_o \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_o \end{bmatrix} + \begin{bmatrix} L_d & 0 & 0 \\ 0 & L_q & 0 \\ 0 & 0 & L_o \end{bmatrix} \frac{d \begin{bmatrix} i_d \\ i_q \\ i_o \end{bmatrix}}{dt} + P\omega_r \begin{bmatrix} -L_q i_q \\ L_d i_d + \phi_f \\ 0 \end{bmatrix} \tag{2.25}$$

The balance of powers in the (*d*, *q*, *o*) coordinate system is given by the following expression:

$$[P_i] = \left[i_{a,b,c}\right]^T \left[v_{a,b,c}\right]$$

$$= \left[i_{d,q,o}\right] \left([T]^{-1}\right)^T [T]^{-1} \left[v_{d,q,o}\right]$$

$$= \frac{3}{2}\left(v_d i_d + v_q i_q + v_o i_o\right) \tag{2.26}$$

Developing this equation, it is possible to obtain:

$$[P_i] = \left[i_{a,b,c}\right]^T \left[v_{a,b,c}\right]$$

$$= \left[i_{d,q,o}\right] \left([T]^{-1}\right)^T [T]^{-1} \left[v_{d,q,o}\right]$$

$$= \left[i_{d,q,o}\right] \left([T]^{-1}\right)^T [T]^{-1} \left\{ [R]\left[i_{d,q,o}\right] + \left[L_{d,q,o}\right] \frac{d}{dt}\left[i_{d,q,o}\right] + P\omega_r \begin{bmatrix} -\phi_q \\ \phi_d \\ 0 \end{bmatrix} \right\}$$

$$= \left[i_{d,q,o}\right] \left([T]^{-1}\right)^T [T]^{-1} [R]\left[i_{d,q,o}\right] + \left[i_{d,q,o}\right] \left([T]^{-1}\right)^T [T]^{-1} \left[L_{d,q,o}\right] \frac{d}{dt}\left[i_{d,q,o}\right]$$

$$+\left[i_{d,q,o}\right]\left(\left[T\right]^{-1}\right)^{T}\left[T\right]^{-1}P\omega_r\begin{bmatrix}-\phi_q\\ \phi_d\\ 0\end{bmatrix} \tag{2.27}$$

where the first term represents the joule losses in the stator windings P_{cu}, the second term represents variation in the time unit of the energy stored in the magnetic field produced by the armature ampere-turns (P_{mf}) and the last term represents the output electromechanical power P_m. The electromechanical power P_m is equal to:

$$P_m=\left[i_{d,q,o}\right]\left(\left[T\right]^{-1}\right)^{T}\left[T\right]^{-1}P\omega_r\begin{bmatrix}-\phi_q\\ \phi_d\\ 0\end{bmatrix}$$

$$=\frac{3}{2}P\omega_r\left[\phi_d i_q-\phi_q i_d\right]$$

$$=\frac{3}{2}P\omega_r\left[\phi_f i_q+\left(L_d-L_q\right)i_d i_q\right] \tag{2.28}$$

From this equation, it is possible to obtain the expression of electromechanical torque:

$$T_{em}=\frac{3}{2}P\left[\phi_d i_q-\phi_q i_d\right]$$

$$=\frac{3}{2}P\left[\phi_f i_q+\left(L_d-L_q\right)i_d i_q\right] \tag{2.29}$$

The expression of electromechanical torque is composed of two components, where the first, named fundamental torque, is a function of the flux linkage produced by PMs and of *q*-axis current and the second term, named reluctance torque, is a function of both *dq*-axis currents and the saliency of the machine. Below, all the equations of the IPMSM conventional dynamic and mathematical model, including the mechanical balance equation, are reported:

$$\begin{cases} v_d = Ri_d + L_d \dfrac{di_d}{dt} - P\omega_r L_q i_q \\ v_q = Ri_q + L_q \dfrac{di_q}{dt} + P\omega_r L_d i_d + P\omega_r \phi_f \\ T_{em} = \dfrac{3}{2} P\left[\phi_f i_q + \left(L_d - L_q\right) i_d i_q\right] \\ T_{em} = T_L + F\omega_r + J\dfrac{d\omega_r}{dt} \\ \dfrac{d\theta_r}{dt} = \omega_r \end{cases} \tag{2.30}$$

where:

- T_L is the load torque;
- F is the viscous friction coefficient;
- J is the rotor moment of inertia.

The circuital model of the conventional IPMSM mathematical model is reported in Figure 2.5. In particular, Figure 2.5(a) represents the electromagnetic phenomena that occur in the *d*-axis circuit, while Figure 2.5(b) represents the electromagnetic phenomena that occur in the *q*-axis circuit.

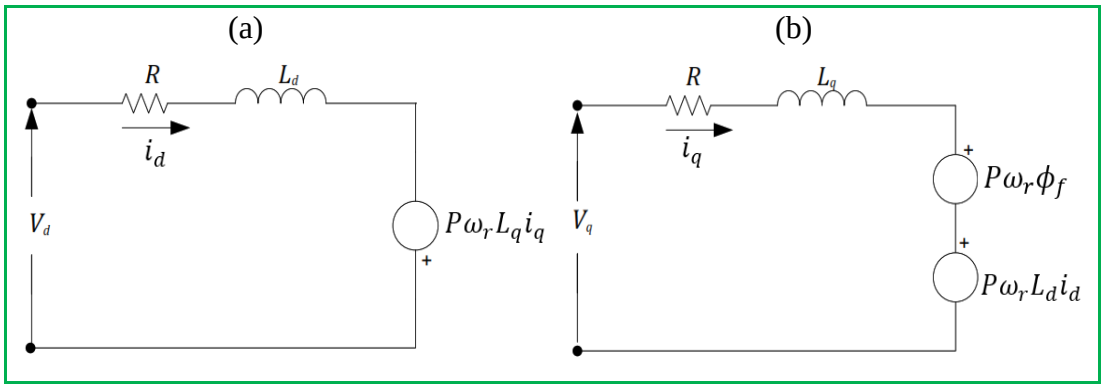


Figure 2. 5. Circuital model derived from conventional IPMSM mathematical model.

2.5 VECTOR CONTROL OF THE IPMSM

Chronologically FOC was the first vector control method developed for controlling induction motors. The principle of this method was proposed in the early 1970s by F. Blaschke of Siemens, who used physical analysis to show that two components of the stator current space vector projected along two rectangular axes, called the direct and quadrature axes, play the same roles as field and armature

windings in a DC motor. The direct axis was found to be oriented along the axis of the rotor magnetic field, which is why this approach has been called field orientation [79].

2.5.1 Field-Oriented Control Principles

In the DC machine, the field current in the stationary poles producing the magnetizing flux and the armature current directly controlling the torque is independently accessible. Moreover, for a fully compensated DC motor, the spatial angle between the flux and the armature MMF is held at 90^o with respect to each other, independent of the load, by the commutator and the brushes whereas in an AC motor (both induction and synchronous motor), the spatial angle between the rotating stator and rotor fields varies with the load and gives rise to oscillatory dynamic response. Control methods for AC motors that emulate the DC motor control by orienting the stator current so that the spatial angle between rotor and stator flux can be maintained at 90^o in this way, this technic attains independent and decoupled control of flux and torque are known as FOC [42]. It requires control of both the magnitude and phase of AC quantities and thus are referred to as vector control methods. Now, it has been established as a powerful technique in the field of AC motor drives and adopted worldwide [80], [81].

The FOC of the IPMSM is an important variation of vector control methods. The FOC method aims to control the magnetic field and the torque by controlling the *d* and *q* components of the stator currents or relatively fluxes. With the information of the stator currents and the rotor angle, the FOC technique makes it possible to control the motor torque and the flux very efficiently [82].

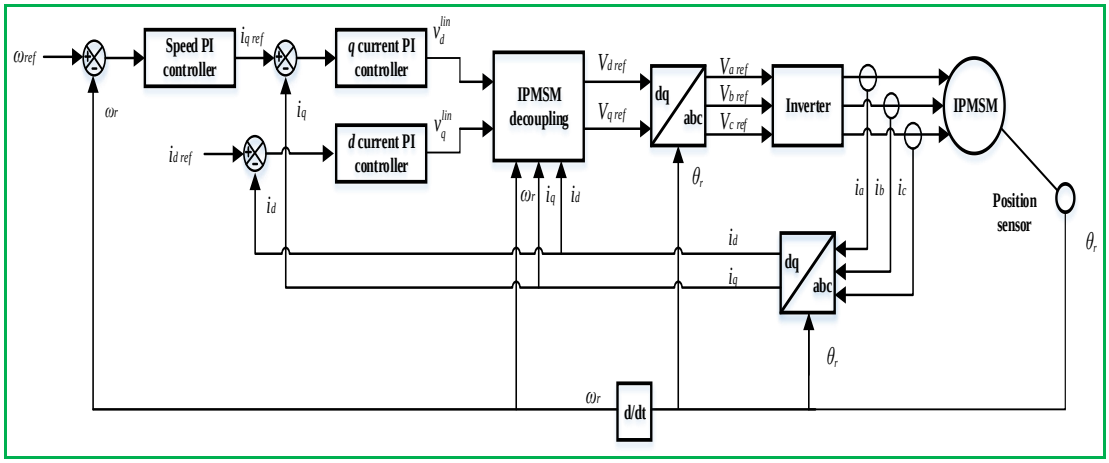


Figure 2. 6. Field-oriented control schematic.

The implementation of this technique will be carried out using two current regulators, one for the *d*-axis component and another for the *q*-axis component, and one-speed regulator. Figure 2.6 shows the block diagram of the FOC method. As shown in the figure, there are three PI regulators in the control system. One is for the mechanical system (speed) and two are for the electrical system (*d* and *q* currents).

At first, the reference speed ω_{ref} is compared with the measured speed ω_r, and the error signal is fed to the speed PI controller. This regulator compares the actual and reference speed and outputs a torque command. Once is obtained the torque command; the regulator can be turned into the *q*-axis current reference i_{qref}.

There is a PI controller to regulate the *d* component of the stator current. The reference value i_{dref} is zero in this part of the thesis since there is no loss minimization. The *d* component error of the current is used as an input for the PI regulator. Moreover, there is another PI controller to regulate the *q* component of the current. The reference value is compared with the measured and then fed to the PI regulator [83], [84].

2.5.2 Stator Voltage Decoupling

The *d*-axis stator current i_d (rotor field component) and the *q*-axis stator current i_q (torque-producing component) must be controlled independently. However, the equations of the stator voltage components are coupled. The *d*-axis component v_d also depends on i_q, and the *q*-axis component v_q also depends on i_d.

The stator voltage components v_d and v_q cannot be considered as decoupled control variables for the rotor flux and electromagnetic torque. The stator currents i_d and i_q can be independently controlled (decoupled control), only if the stator voltage equations are decoupled, so these stator current components are indirectly controlled by controlling the terminal voltages of the synchronous motor [84], [85].

The equations of the stator voltage components in the (*d*, *q*) reference frame can be reformulated and separated into two components: linear components v_d^{lin}, v_q^{lin}, and decoupling components $v_d^{decouple}$, $v_q^{decouple}$. The equations are decoupled as follows:

$$v_d = v_d^{lin} + v_d^{decouple} \tag{2.31}$$

$$v_q = v_q^{lin} + v_q^{decouple} \tag{2.32}$$

The linear components are defined:

$$v_d^{lin} = Ri_d + L_d \frac{di_d}{dt} \tag{2.33}$$

$$v_q^{lin} = Ri_q + L_q \frac{di_q}{dt} \tag{2.34}$$

The decoupling components are defined:

$$v_d^{decouple} = -P\omega_r L_q i_q \tag{2.35}$$

$$v_q^{decouple} = P\omega_r \left(L_d i_d + \phi_f \right) \tag{2.36}$$

The decoupling components $v_d^{decouple}$, $v_q^{decouple}$ are evaluated from the stator voltages equation 2.30. They eliminate cross-coupling for current control loops at a given motor operating point. Linear components v_d^{lin}, v_q^{lin} are set by the outputs of the current controllers and the decoupling components are feedforwarded to the *d* and *q* voltages according to the following equations:

$$v_d = v_d^{lin} - P\omega_r L_q i_q \tag{2.37}$$

$$v_q = v_q^{lin} + P\omega_r \left(L_d i_d + \phi_f \right) \tag{2.38}$$

The above equations 2.37 and 2.38 are evaluated in the decoupling block (see Figure 2.6).

2.5.3 Calculation of Regulators

To calculate the parameters of the regulators, continuous linear models are adopted. Classic automatic methods can be used. These methods have the advantage of being simple and easy to implement.

The fundamental elements for the realization of the regulators are the actions PID. Algorithms, even the best performing, are always a combination of these actions. We consider that the machine is vector-oriented and completely decoupled. This allows us to write the equations of the machine in a simple way and calculate the coefficients of the regulators [86].

We are satisfied with classic PI-type regulators. In this case, we can distinguish two modes:

- Electric mode (fast mode: internal loop);
- Mechanical mode (slow mode: external loop).

For each of the current loops, we propose to control the machine with conventional PI regulators to compensate for the disturbance of the load torque and eliminate the speed static error.

The PI regulator is a combination of a P regulator and an I regulator, as shown in Figure 2.7. The relation between the output $U_r(t)$ and the error signal $\varepsilon(t)$ is given by:

$$U_r(t) = K_p \varepsilon(t) + K_i \int_0^t \varepsilon(t) d(t) \tag{2.39}$$

It means that $\frac{U_r(s)}{\varepsilon(s)} = K_p + \frac{K_i}{s}$ (2.40)

where:

- K_p is the proportional gain;
- K_i is the integral gain.

The PI regulator is shown in the following figure:

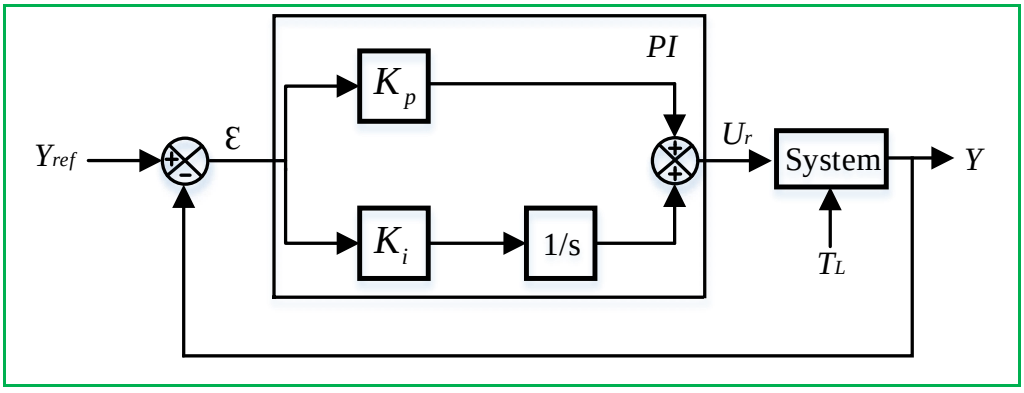


Figure 2. 7. PI regulator.

The transfer function will be:

$$U_r = K_p + \frac{K_i}{s} \tag{2.41}$$

We can also write the regulator in the following form:

$$PI \rightarrow \frac{U_r}{\varepsilon} = \frac{1+sT_1}{sT_2} \tag{2.42}$$

with

$$\begin{cases} K_p = \dfrac{T_1}{T_2} \\ K_i = \dfrac{1}{T_2} \end{cases} \tag{2.43}$$

As the machine is decoupled along two axes (*d*, *q*), the regulation on the *d*-axis is done by a single loop, while the regulation on the *q*-axis is done by two cascading loops, internal to regulate the current and external to regulate the speed.

Current Regulator i_q

The IPMSM is controlled by controlling the i_d and i_q currents, the system consists of a speed regulation loop, which imposes the current reference i_{qref}, the *d*-axis current is kept zero, and the control is carried out by the PI regulators to have the zero static error ($\varepsilon = 0$).

Knowing that: $i_q = \dfrac{v_q^{lin}}{R+sL_q}$

And that the general form of the regulator is: $\dfrac{1+sT_1}{sT_2}$ (equation (2.42)), gives the following diagram:

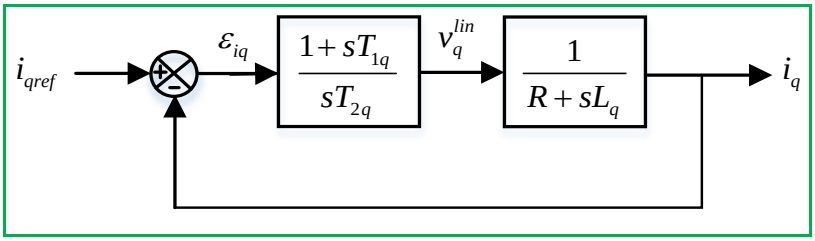


Figure 2. 8. i_q current regulation loop.

The Open Loop Transfer Function (OLTF) in Figure 2.8 is:

$$OLTF = \frac{1+sT_{1q}}{sT_{2q}\left(R+sL_q\right)}$$

$$= \frac{1+sT_{1q}}{sT_{2q}R\left(1+s\frac{L_q}{R}\right)} \tag{2.44}$$

By using the pole compensation method, we will have:

$1+sT_{1q} = 1+s\frac{L_q}{R}$ on condition that:

$$\frac{L_q}{R} = T_{1q} = \tau_q \tag{2.45}$$

where τ_q is the electrical time constant.

By replacing the constant T_{1q} by its value (2.45) in equation (2.44), we obtain:

$$OLTF = \frac{1}{sRT_{2q}} \tag{2.46}$$

For the Closed Loop Transfer Function (CLTF):

$$CLTF = \frac{OLTF}{1+OLTF}$$

$$= \frac{1}{1+sRT_{2q}} \tag{2.47}$$

By identifications to $\frac{1}{1+s\tau_q}$ we find:

$$\tau_q = RT_{2q} \Rightarrow T_{2q} = \frac{\tau_q}{R} \tag{2.48}$$

We impose the response time: $T_{rq} = 3\tau_q$ (criterion of ± 5%)

$$T_{2q} = \frac{T_{rq}}{3R} \tag{2.49}$$

where:

- T_{rq} is the imposed response time;
- $\tau_q = \frac{L_q}{R}$ is the electrical time constant of the *q*-axis of the machine.

By replacing (2.48) in (2.43), we obtain:

$$K_i = \frac{R}{\tau_q} \tag{2.50}$$

By replacing the response time T_{rq} in equation (2.50), we obtain:

$$\begin{cases} K_{pq} = \dfrac{3L_q}{T_{rq}} \\ K_{iq} = \dfrac{3R}{T_{rq}} \end{cases} \tag{2.51}$$

Current Regulator i_d

The regulation loop on the *d*-axis is shown in the following figure:

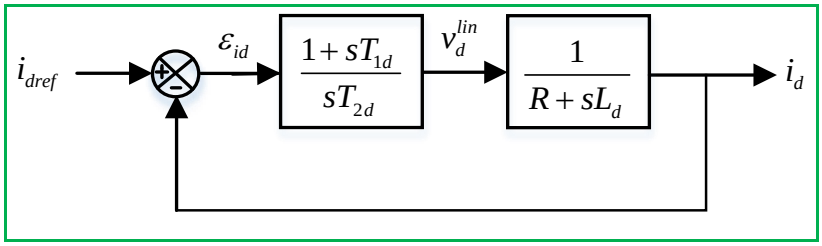


Figure 2. 9. i_d current regulation loop.

For the regulation of the *d*-axis current, the same procedure used for the regulation of the *q*-axis current is followed.

$$OLTF = \frac{1+sT_{1d}}{sT_{2d}R\left(1+s\dfrac{L_d}{R}\right)} \tag{2.52}$$

$$CLTF = \frac{1}{1+sRT_{2d}} \tag{2.53}$$

$$\frac{L_d}{R} = T_{1d} = \tau_d \tag{2.54}$$

Imposing that: $T_{rd} = 3\tau_d$

$$\tau_d = RT_{2d} \Rightarrow T_{2d} = \frac{\tau_d}{R} \tag{2.55}$$

where $\tau_d = \dfrac{L_d}{R}$ is the electrical time constant of the *d*-axis of the machine.

By replacing the response time T_{rd} in equation (2.54), we obtain:

$$T_{2d} = \frac{T_{rd}}{3R} \tag{2.56}$$

Finally, we can get

$$\begin{cases} K_{pd} = \dfrac{3L_d}{T_{rd}} \\ K_{id} = \dfrac{3R}{T_{rd}} \end{cases} \tag{2.57}$$

Speed Regulator

The process to be controlled can be broken down into two subsystems:

- i_q current control subsystem therefore also torque;
- Subsystem of the mechanical part.

The speed regulation loop is represented by the figure below:

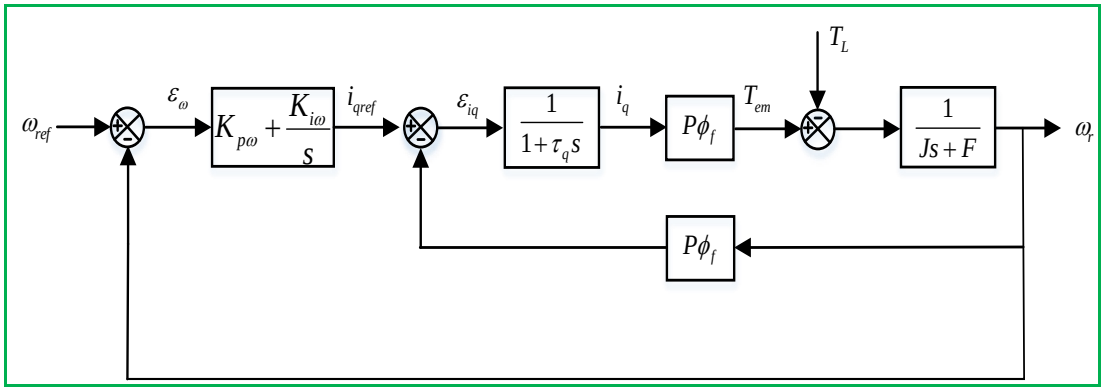


Figure 2. 10. Speed regulation loop.

The diagram in Figure 2.10 can be simplified by Figure 2.11:

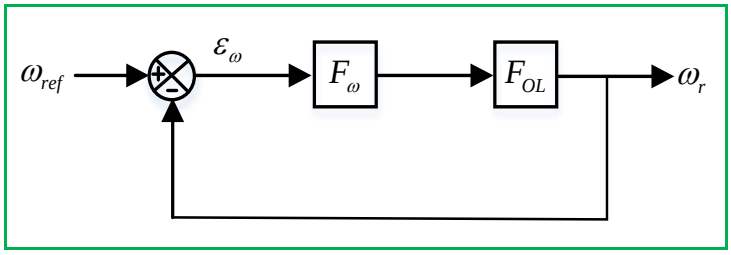


Figure 2. 11. Simplified speed regulation loop.

The speed regulator transfer function $F_\omega(s)$ is expressed as:

$$F_\omega(s) = K_{p\omega} + \frac{K_{i\omega}}{s}$$

$$= \frac{1+sT_{1\omega}}{sT_{2\omega}} \tag{2.58}$$

with

$$\begin{cases} K_{p\omega} = \dfrac{T_{1\omega}}{T_{2\omega}} \\ K_{i\omega} = \dfrac{1}{T_{2\omega}} \end{cases} \tag{2.59}$$

The open-loop transfer function $F_{0L}(s)$ is given by:

$$F_{OL}(s) = \frac{P\phi_f}{F(1+s\tau_q)+(1+s\tau_m)} \tag{2.60}$$

where $\tau_m = \dfrac{J}{F}$ is the mechanical time constant of the machine.

The CLTF is given by:

$$F_{CL}(s) = \frac{F_\omega(s)F_{OL}(s)}{1+F_\omega(s)F_{OL}(s)} \tag{2.61}$$

After calculations, we find:

$$F_{CL}(s) = \frac{P\phi_f\left(K_{p\omega}s+K_{i\omega}\right)}{J\tau_q s^2+\left(J+F\tau_q\right)s^2+\left(P\phi_f K_{p\omega}+F\right)s+P\phi_f K_{i\omega}} \tag{2.62}$$

By neglecting $J\tau_q$ and $F\tau_q$ in front of J, the polynomial of the function becomes:

$$p(s) = Js^2+\left(P\phi_f K_{p\omega}+F\right)s+P\phi_f K_{i\omega} \tag{2.63}$$

By imposing on the characteristic closed-loop polynomial two conjugate complex poles $s_{1,2} = \rho(1 \mp j)$, it becomes:

$$p(s) = s^2+2\rho s+2\rho^2 \tag{2.64}$$

By term-to-term identification between equations (2.63) and (2.64), we find:

$$\begin{cases} K_{p\omega} = \dfrac{2J\rho-F}{P\phi_f} \\ K_{i\omega} = \dfrac{2\rho^2 J}{P\phi_f} \end{cases} \tag{2.65}$$

where:

- ρ is the damping coefficient ($\rho = \xi\omega_n$);
- ξ is the damping factor;
- ω_n is the natural frequency.

2.6 CONCLUSION

This chapter aims to present the FOC of the IPMSM. An overview of the PM materials and the classification of PMSMs has been first given. Then a brief literature review of the existed control techniques for IPMSM drives has been presented. After that, the modeling of the IPMSM is reviewed based on the theory of reference frame transformation. Both the electrical and mechanical equations of the machine are illustrated, and finely the PI controller design has been introduced.

Chapter 3: Modified Backstepping Control of Interior Permanent Magnet Synchronous Motor

3.1 INTRODUCTION

The IPMSM is broadly used in various industrial fields due to its simple structure, high efficiency, and good dynamic response [36], [87]. However, it is a difficult problem to control the IPMSM to achieve satisfactory performance because its dynamics are usually coupled, highly nonlinear, multivariable; and sensitive to parameter perturbations and load disturbances [88]. Conventional linear controllers such as PI and PID are commonly used because of their simple implementation and applicability in most industrial control processes [89]. On the other hand, such kinds of controllers may fail to achieve the control performance requirements of industrial servo and speed tracking drive applications since they are sensitive to motor parameter variations and external torque disturbances [90]. To reach a higher performance of IPMSM, a lot of work has been dedicated to the development of nonlinear control methods, and several control techniques have been suggested that improve the motor control performance in various aspects.

Backstepping control was introduced in the early 1990s by Kokotovic et al. [70], which gives a feasible idea for the design of nonlinear controllers. Backstepping control is a recursive and systematic design methodology. Depending on the Lyapunov stability theory, the control law fulfilling the convergence condition of the Lyapunov function is obtained by constructing the feedback control law and the Lyapunov function at the same time [91], [92]. Backstepping control theory has been successfully applied to many engineering fields such as motor drive, as it simplifies the design process, and achieves the complete decoupling and global stability of the IPMSM system [93].

Traditional backstepping is successfully applied to the control of IPMSM drive systems, which greatly simplifies the design process of the general system. However, there are some drawbacks in the conventional backstepping control method, such as speed steady-state error, sensitivity to parameter uncertainties, and load torque

disturbances [2]. In order to improve the control performance, the integral of the tracking error is introduced in the backstepping control.

This chapter introduces an IBC that can precisely follow the speed reference trajectory of the motor in the existence of load fluctuation and parameter uncertainties. The influence of the perturbation of the load torque and the variation of motor parameters can be considerably decreased by using integral action at each step to guarantee high precision speed control [30], [94].

The contribution is the design of a nonlinear controller based on a modified backstepping technique to ensure high-performance speed regulation and guarantee high-accuracy control in the presence of load torque disturbances and parameter uncertainties, the contribution is to develop a controller with equations different from other integral backstepping controllers. The Lyapunov stability theorem is utilized to ensure the asymptotic stability of the resulting closed-loop system. Consequently, the suggested control approach is not only to stabilize the IPMSM system but also to force the motor speed to follow the reference speed asymptotically. To verify the advantage of the proposed controller, a comparison between the PI controller and the proposed one was studied. Simulation and experimental results illustrate the correctness and effectiveness of the proposed control method.

This chapter is structured as follows: Section 3.2 introduces the design of the backstepping control with integral action. In section 3.3, the controller's tunning is presented. In Sections 3.4 and 3.5, simulation and experimental results are shown to exhibit the effectiveness of the proposed control scheme. In Section 3.6, a conclusion is given.

From equation (2.30), we can see that the IPMSM is a nonlinear system with multiple variables and strong coupling effects. However, the machine parameters keep varying during motor operation. For example, stator resistance R is prone to change with the temperature variation, while the ϕ_f, L_d and L_q may vary with the changes in the operating current, this concerns the electrical parameters. As for the mechanical parameters, J and F may greatly increase when the external mechanical load is applied to the IPMSM drive [95], [96].

3.2 DESIGN OF BACKSTEPPING CONTROL WITH INTEGRAL ACTION

Backstepping control is a new type of recursive and systematic design methodology for the nonlinear system. By introducing the virtual control law, it decomposes the complex nonlinear system into low order systems. It can not only ensure the overall stability but also perfect tracking performance in real implementation. Therefore, it is particularly adapted for the speed control of the motor, and improved dynamic performance of the IPMSM can be experienced [97].

In this section, the proposed integral backstepping control law has been derived in two steps to control the rotor speed of the IPMSM. The integral of the tracking error is inserted into the control law, and by choosing a suitable Lyapunov function, the control law with integral action is obtained to reduce the effect of parameter variations and load torque perturbations on the control system. The block diagram that presents the backstepping control scheme is shown in Figure 3.1. For equation (2.30), the design of the IBC is carried out in two steps, where i_d and i_q are the outputs and v_d, v_q and ω_r are the inputs. At each design step, virtual control functions will be established by using the convenient Lyapunov functions. In the last step, the overall controller design is established to control the system.

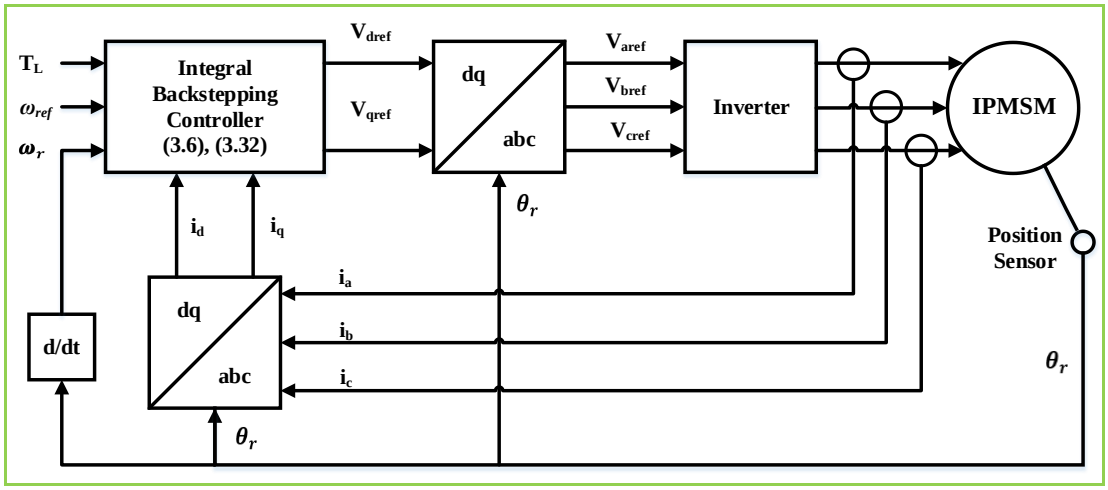


Figure 3. 1. The proposed control configuration.

Step 1

In order to realize complete decoupling and precise current tracking of the proposed controller, the *d*-axis current i_d is forced to be zero by choosing the control input of the first step $i_{dref} = 0$. For this, we choose the following Lyapunov function:

$$V_1(e_1) = \frac{1}{2}e_1^2 + \frac{1}{2}e_1'^2 \tag{3.1}$$

where e_1 represent the *d*-axis current tracking error.

$$e_1(t) = i_d(t) - i_{dref}(t) + e_1'(t) \tag{3.2}$$

where $e_1'(t) = k_1' \int_0^t \left(i_d(t) - i_{dref}(t) \right) dt$ is an integral action.

The first step virtual control input i_{dref} is fixed to zero, the time derivative of V_1 is given by

$$\dot{V}_1(e_1) = e_1 \dot{e}_1 + e_1' \dot{e}_1'$$

$$= e_1 \left[\frac{di_d}{dt} + k_1' i_d \right] + e_1' k_1' i_d \tag{3.3}$$

where

$$i_d(t) = e_1(t) - e_1'(t) \tag{3.4}$$

Equation (3.4) is substituted into (3.3). We have

$$\dot{V}_1(e_1) = e_1 \left[-\frac{R}{L_d} i_d + p \frac{L_q}{L_d} \omega_r i_q + \frac{1}{L_d} v_d + k_1' i_d \right] + e_1' k_1' i_d$$

$$= e_1 \left[-\frac{R}{L_d} i_d + p \frac{L_q}{L_d} \omega_r i_q + \frac{1}{L_d} v_d \right] + k_1' \left[e_1 + e_1' \right] \left[e_1 - e_1' \right] \tag{3.5}$$

Therefore, to make equation (3.5) negative and definite, we suggest the *d*-axis control input v_{dref} as

$$v_{dref} = L_d \left[\frac{R}{L_d} i_d - p \frac{L_q}{L_d} \omega_r i_q - k_1 e_1 \right] \tag{3.6}$$

where k_1 and k_1' are positive design control gains.

Substituting (3.6) in (3.5), we get

$$\dot{V}_1(e_1) = -\left[k_1 - k_1' \right] e_1^2 - k_1' e_1'^2 < 0 \tag{3.7}$$

Step 2

Persisting with the backstepping design method to solve the speed tracking problem, we suggest the following Lyapunov function:

$$V_2(e_1,e_2)=\frac{1}{2}e_2^2+V_1(e_1) \tag{3.8}$$

The position tracking error is defined as

$$e_2(t)=\theta_r(t)-\theta_{ref}(t) \tag{3.9}$$

where θ_{ref} is the rotor reference position.

The time derivative of V_2 is just

$$\dot{V}_2(e_1,e_2)=e_2\dot{e}_2-\left[k_1-k_1'\right]e_1^2-k_1'e_1'^2 \tag{3.10}$$

By taking the time derivative of the position tracking error e_2 as

$$\dot{e}_2(t)=\omega_r(t)-\omega_{ref}(t) \tag{3.11}$$

Substituting (3.11) in (3.10), we get

$$\dot{V}_2(e_1,e_2)=e_2\left[\omega_r(t)-\omega_{ref}(t)\right]-\left[k_1-k_1'\right]e_1^2-k_1'e_1'^2 \tag{3.12}$$

We define the tracking error e_3, which includes the first stabilizing function $\gamma_1(x)$, to obtain a time derivative for $V_2(e_1,e_2)$ which is negative definite

$$e_3(t)=\omega_r(t)-\gamma_1(x)\Leftrightarrow\omega_r(t)=e_3(t)+\gamma_1(x) \tag{3.13}$$

Substituting the rotor speed $\omega_r(t)$ from (3.13) into the time derivative of $V_2(e_1,e_2)$ to obtain

$$\dot{V}_2(e_1,e_2)=e_2\left[e_3(t)+\gamma_1(x)-\omega_{ref}(t)\right]-\left[k_1-k_1'\right]e_1^2-k_1'e_1'^2 \tag{3.14}$$

Defining the stabilizing function $\gamma_1(x)$ as

$$\gamma_1(x)=\omega_{ref}(t)-k_2e_2 \tag{3.15}$$

where k_2 is a positive design control gain.

The time derivative of $V_2(e_1,e_2)$ is computed as:

$$\dot{V}_2(e_1,e_2)=-\left[k_1-k_1'\right]e_1^2-k_1'e_1'^2-k_2e_2^2+e_2e_3 \tag{3.16}$$

The tracking error $e_3(t)$ should be zero to ensure that $V_2(e_1, e_2)$ derivative is negative definite.

$$\dot{V}_2(e_1, e_2) = -\left[k_1 - k_1'\right]e_1^2 - k_1' e_1'^2 - k_2 e_2^2 < 0 \tag{3.17}$$

For this, we suggest the following Lyapunov function:

$$V_3(e_1, e_2, e_3) = \frac{1}{2}e_3^2 + V_2(e_1, e_2) \tag{3.18}$$

Its time derivative is

$$\begin{aligned}\dot{V}_3(e_1, e_2, e_3) &= e_3 \dot{e}_3 + \dot{V}_2(e_1, e_2) \\ &= e_3 \dot{e}_3 + e_2 e_3 - \left[k_1 - k_1'\right]e_1^2 - k_1' e_1'^2 - k_2 e_2^2 \\ &= e_3\left[\dot{e}_3 + e_2\right] - \left[k_1 - k_1'\right]e_1^2 - k_1' e_1'^2 - k_2 e_2^2 \end{aligned} \tag{3.19}$$

From (3.13), the tracking error e_3 time derivative can be calculated as

$$\begin{aligned}\dot{e}_3(t) &= \dot{\omega}_r(t) - \dot{\gamma}_1(x) \\ &= y_4(t) - \frac{1}{J}\left(T_L + F\omega_r\right) + k_2 \dot{e}_2 - \dot{\omega}_{ref}(t) \end{aligned} \tag{3.20}$$

Continuing with the backstepping methodology, we define the second step virtual control input $y_4(t)$ as:

$$y_4(t) = \frac{p}{J}\left[\left(L_d - L_q\right)i_d + \phi_f\right]i_q \tag{3.21}$$

Then by substituting (3.20) in (3.19), the time derivative of $V_3(e_1, e_2, e_3)$, along with the solutions of (3.20), is given by

$$\dot{V}_3(e_1, e_2, e_3) = e_3\left[y_4(t) - \frac{1}{J}\left(T_L + F\omega_r\right) + k_2 \dot{e}_2 - \dot{\omega}_{ref}(t) + e_2\right] - \left[k_1 - k_1'\right]e_1^2 - k_1' e_1'^2 - k_2 e_2^2 \tag{3.22}$$

We suggest a tracking error e_4 that includes the second stabilizing function $\gamma_2(x)$ which makes the time derivative of $V_3(e_1, e_2, e_3)$ negative definite. Thus,

$$e_4(t) = y_4(t) - \gamma_2(x) + e_4'(t) \Leftrightarrow y_4(t) = e_4(t) + \gamma_2(x) - e_4'(t) \tag{3.23}$$

where $e'_4(t) = k'_4 \int_0^t \left(y_4(t) - \gamma_2(x)\right) dt$ is an integral action.

The time derivative of $V_3(e_1, e_2, e_3)$ resulting from the substitution of (3.23) in (3.22) is

$$\dot{V}_3\left(e_1,e_2,e_3\right)=e_3\left[e_4+\gamma_2\left(x\right)-e'_4\left(t\right)-\frac{1}{J}\left(T_L+F\omega_r\right)+k_2\dot{e_2}-\dot{\omega_{ref}}+e_2\right]-\left[k_1-k'_1\right]e_1^2-k'_1e'^2_1-k_2e_2^2$$

(3.24)

From (3.24), we suggest the second stabilizing function $\gamma_2(x)$ as

$$\gamma_2\left(x\right)=\dot{\omega_{ref}}-k_2\dot{e_2}-k_3e_3+\frac{1}{J}\left(T_L+F\omega_r\right)-e_2 \tag{3.25}$$

where k_3 is a positive design control gain.

Substituting (3.25) in (3.24), we can get the following function:

$$\dot{V}_3\left(e_1,e_2,e_3\right)=-\left[k_1-k'_1\right]e_1^2-k'_1e'^2_1-k_2e_2^2-k_3e_3^2+e_3e_4-e_3e'_4 \tag{3.26}$$

If we choose $e_4 = 0$, then the time derivative of $\dot{V}_3(e_1, e_2, e_3)$ is negative definite

$$\dot{V}_3\left(e_1,e_2,e_3\right)=-\left[k_1-k'_1\right]e_1^2-k'_1e'^2_1-k_2e_2^2-k_3e_3^2-e_3e'_4<0 \tag{3.27}$$

Persisting with the backstepping design method, we suggest the last Lyapunov function as

$$V_4\left(e_1,e_2,e_3,e_4\right)=\frac{1}{2}e_4^2+V_3\left(e_1,e_2,e_3\right) \tag{3.28}$$

Its time derivative is

$$\dot{V}_4\left(e_1,e_2,e_3,e_4\right)=-\left[k_1-k'_1\right]e_1^2-k'_1e'^2_1-k_2e_2^2-k_3e_3^2+e_4\left[\dot{e_4}+e_3\right]-e_3e'_4 \tag{3.29}$$

where e_4 time derivative is given by

$$\dot{e_4}=\dot{y_4}\left(t\right)-\dot{\gamma_2}\left(x\right)+\dot{e'_4}\left(t\right)$$

$$=\frac{p}{J}\left[\left(L_d-L_q\right)i_d+\phi_f\right]\left[-\frac{R}{L_q}i_q-p\frac{L_d}{L_q}\omega_r i_d-p\frac{1}{L_q}\phi_f\omega_r+\frac{1}{L_q}v_q\right]$$

$$+k_2\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)-\dot{\omega_{ref}}\right]+k_3\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)+k_2\dot{e_2}-\dot{\omega_{ref}}\right]$$

$$+k_4'\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)+k_2\dot{e}_2+k_3e_3+e_2-\dot{\omega}_{ref}\right]-\frac{1}{J}\frac{d}{dt}\left(T_L+F\omega_r\right)+\dot{e}_2-\ddot{\omega}_{ref} \quad (3.30)$$

Substituting the expressions (3.5) and (3.30) into (3.29), we obtain

$$\begin{aligned}\dot{V}_4\left(e_1,e_2,e_3,e_4\right)=&e_1\left[-\frac{R}{L_d}i_d+p\frac{L_q}{L_d}\omega_r i_q+\frac{1}{L_d}v_d\right]+k_1'\left[e_1+e_1'\right]\left[e_1-e_1'\right]-k_2e_2^2-k_3e_3^2\\&+e_4\left\{\frac{p}{J}\left[\left(L_d-L_q\right)i_d+\phi_f\right]\left[-\frac{R}{L_q}i_q-p\frac{L_d}{L_q}\omega_r i_d-p\frac{1}{L_q}\phi_f\omega_r+\frac{1}{L_q}v_q\right]\right.\\&+k_2\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)-\dot{\omega}_{ref}\right]+k_3\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)+k_2\dot{e}_2-\dot{\omega}_{ref}\right]\\&+k_4'\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)+k_2\dot{e}_2+k_3e_3+e_2-\dot{\omega}_{ref}\right]\\&\left.-\frac{1}{J}\frac{d}{dt}\left(T_L+F\omega_r\right)+\dot{e}_2+e_3-\ddot{\omega}_{ref}\right\}-e_3e_4'\end{aligned} \quad (3.31)$$

We propose the q-axis control input v_{qref} as

$$\begin{aligned}v_{qref}=L_q\left\{\left[\frac{R}{L_q}i_q+p\frac{L_d}{L_q}\omega_r i_d+p\frac{1}{L_q}\phi_f\omega_r\right]-\frac{J}{P\left[\left(L_d-L_q\right)i_d+\phi_f\right]}\right.\left(k_2\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)-\dot{\omega}_{ref}\right]\right.\\+k_3\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)+k_2\dot{e}_2-\dot{\omega}_{ref}\right]+k_4'\left[y_4-\frac{1}{J}\left(T_L+F\omega_r\right)+k_2\dot{e}_2+k_3e_3+e_2-\dot{\omega}_{ref}\right]\\\left.\left.-\frac{1}{J}\frac{d}{dt}\left(T_L+F\omega_r\right)+\dot{e}_2+e_3-\ddot{\omega}_{ref}+k_4e_4\right)\right\}\end{aligned} \quad (3.32)$$

Therefore, by substituting (3.6) and (3.32) into (3.31), we can obtain

$$\dot{V}_4\left(e_1,e_2,e_3,e_4\right)=-\left[k_1-k_1'\right]e_1^2-k_1'e_1'^2-k_2e_2^2-k_3e_3^2-k_4e_4^2-e_3e_4' \quad (3.33)$$

where k_4 and k_4' are positive design control gains.

Because $k_1>k_1'>0$, $k_2>0$, $k_3>0$, $k_4>0$, and $k_4'>0$, clearly $\dot{V}_4$ in (3.33) is negative definite, so it indicates that the asymptotic stability of the resulting closed-loop system is guaranteed according to Lyapunov's stability theorem and, in consequence, all the tracking errors e_1, e_2, e_3, and e_4 will converge to zero

asymptotically. Therefore, the rotor speed will track the reference speed accurately, and the control objective has been achieved.

3.3 CONTROLLERS TUNING

In the IBC, the parameters tuning of the controller is the result of repeated debugging to satisfy the Lyapunov stability condition and to consider the system convergence speed. The gains of the proposed controller are chosen as follows: $k_1 = 300$, $k_1' = 100$, $k_2 = 300$, $k_3 = 5$, $k_4 = 300$, and $k_4' = 5$. The calculation of the PI controller gains is done by using the pole compensation method considering both overshoot and response performance. Specifically, the PI controller gains of the current loop are selected as $K_{pd} = 19 \times 10^{-2}$, $K_{id} = 24$, $K_{pq} = 19 \times 10^{-2}$, $K_{iq} = 27$, and the PI controller gains of the speed loop are chosen as $K_{p\omega} = 793 \times 10^{-4}$, and $K_{i\omega} = 208 \times 10^{-3}$.

3.4 SIMULATION RESULTS

3.4.1 Comparative Tests Under the Nominal Conditions

In the first simulation, the motor runs at 104.72 rad/s, and a step load of 0.65 N.m is applied to the motor at t = 5 s.

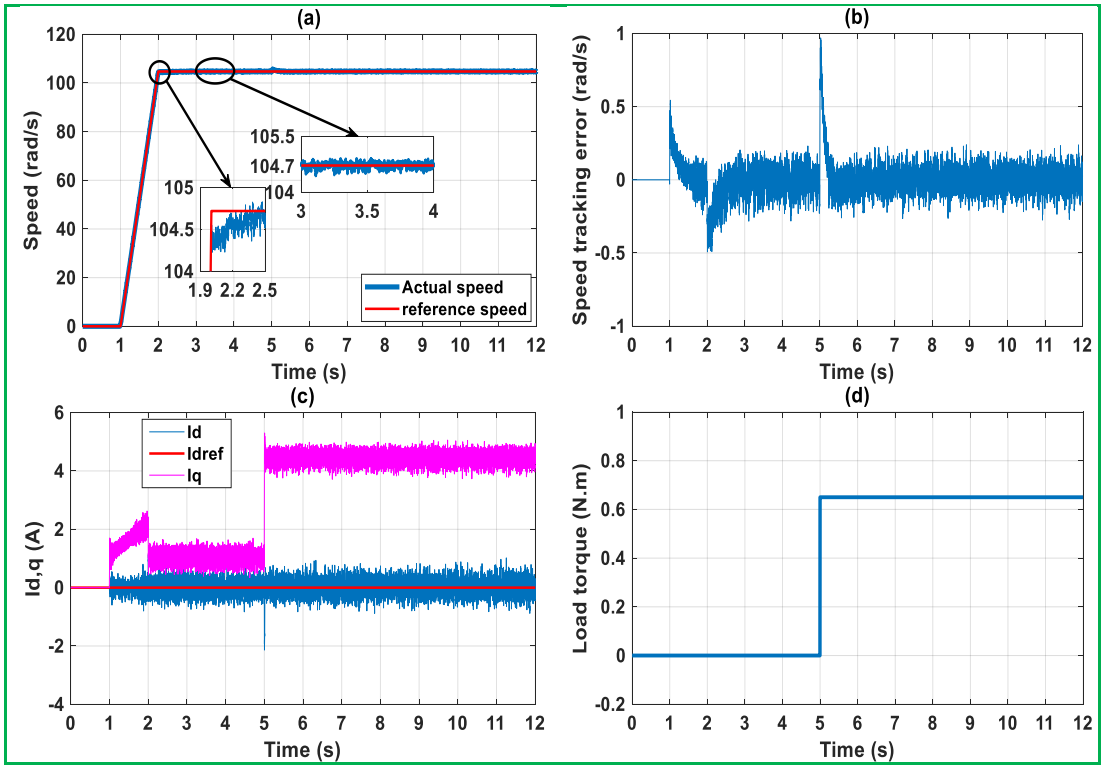


Figure 3. 2. IBC simulation results under nominal motor parameters: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

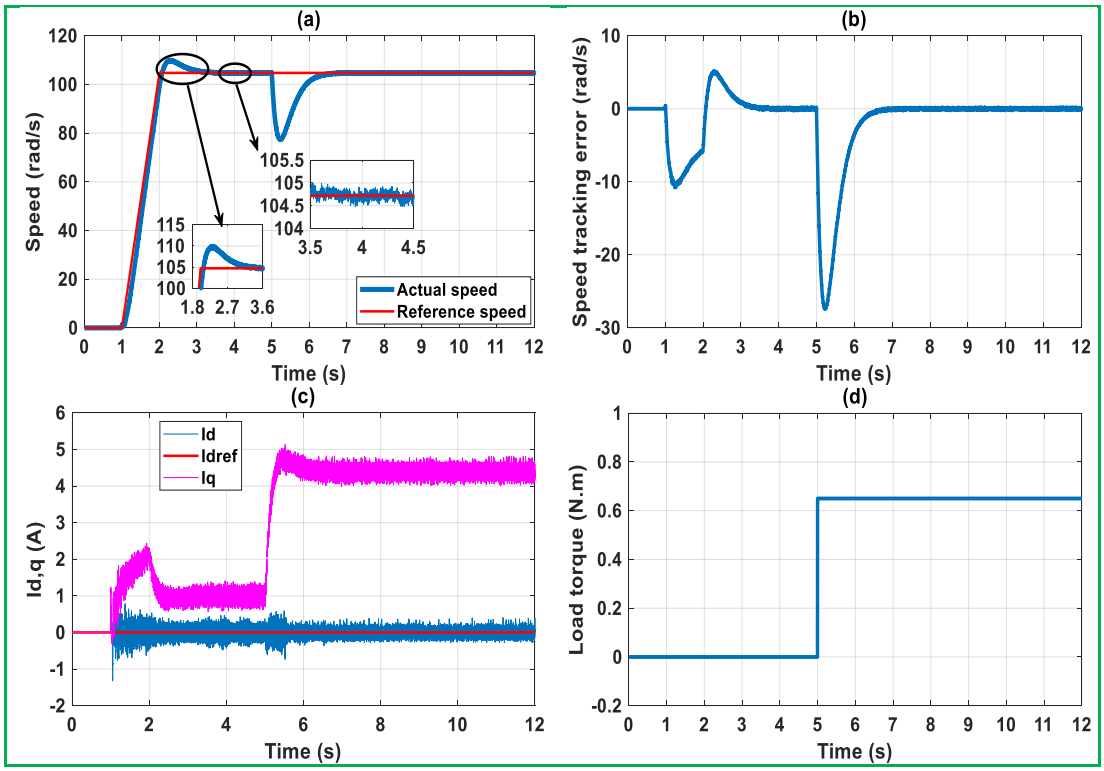


Figure 3. 3. PI controller simulation results under nominal motor parameters: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

Figures 3.2 and 3.3 show the response of each algorithm under nominal motor parameters. The parameters of the motor used in the simulation are shown in Table A.1 (Appendix A).

Figures 3.2(a)-3.2(b), and 3.3(a)-3.3(b) present the speed and the speed tracking error for both controllers, two sections of the speed figures are magnified to show the transient response and the steady-state response. In regard to the IBC compared to the PI controller, the undershoot/overshoot, the settling time and steady-state error are (-0.477 %, 0 s, 0.258 %) and (4.927 %, 2.74 s, 0.229 %), respectively. It can be seen that the IBC has a faster transient response, and almost the same steady-state error compared to the PI controller. When the load torque is applied, the PI controller has a larger dipping in speed tracking about 27.4 rad/s and a long recovery time of about 2.3 s. But compared to IBC, the tracking error can rapidly converge to zero with a smaller dip of about 0.97 rad/s after 1.34 s. Figures 3.2(c) and 3.3(c) show the response of the *dq*-axis currents based on the IBC and PI controller, it can be seen that they can immediately converge to the stable values and the *d*-axis current component is well decoupled and is regulated quite well to be zero, we can also see that the *dq*-axis currents of the IBC exhibits obvious oscillation compared to the *dq*-axis currents of the PI controller. Figures 3.2(d) and 3.3(d) show the load torque. Based on the obtained

results, we can notice that the proposed controller has a significantly better performance compared to the PI controller in the case of nominal motor parameters. A detailed comparative analysis of the performance of the two controllers is shown in Table 3.1.

Table 3. 1. Performance comparison of two control methods under nominal motor parameters.

	IBC	PI Controller
Undershoot/Overshoot (%)	0	4.927
Settling Time (s)	0	2.74
Steady-State Error (%)	0.258	0.229
Dipping in Speed (rad/s)	0.97	27.4
Recovery Time (s)	1.34	2.3

In the next simulation, the speed dynamic behavior is analyzed after a change in the reference speed under no-load condition. The reference speed is given by

$$\begin{cases} \omega_{ref} = 104.72 rad/s & \text{if } 2s \le t \le 8s \\ \omega_{ref} = -104.72 rad/s & \text{if } 10s \le t \le 16s \end{cases} \quad (3.34)$$

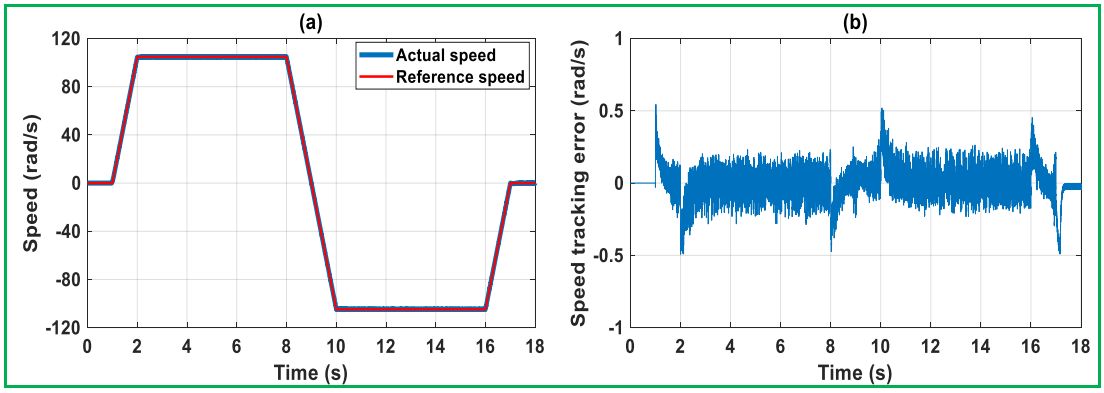


Figure 3. 4. IBC simulation results under nominal motor parameters and change of speed reference: (a) motor speed (b) speed tracking error.

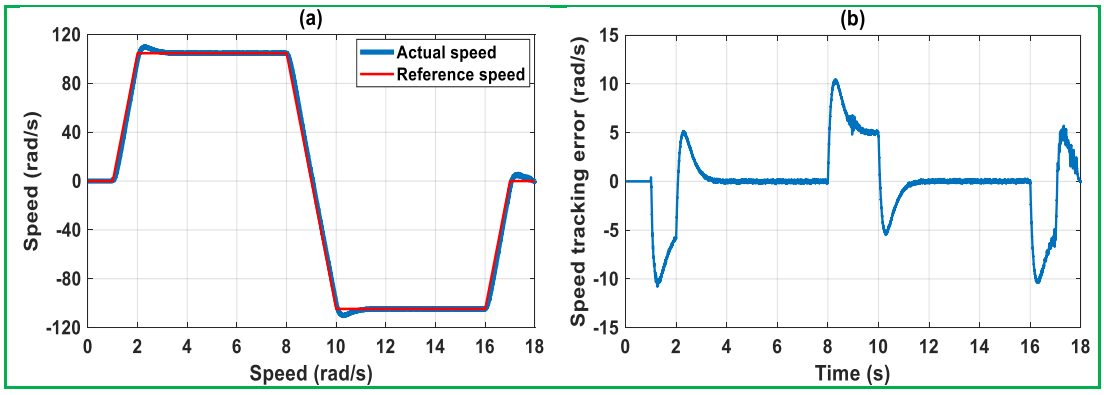


Figure 3. 5. PI controller simulation results under nominal motor parameters and change of speed reference: (a) motor speed (b) speed tracking error.

As seen in Figures 3.4 and 3.5, the comparisons of actual speed and the speed tracking error of the motor are made, including the IBC and PI controller. The parameters of the motor used in the controllers are all nominal values. The simulation results show that the PI controller is not able to follow fast enough precisely the desired speed in both directions, in particular when changing the direction of rotation. Also, the motor speed under this controller shows a large overshoot. It is observed that the IBC has a fast transient response and a smoother steady-state response in tracking the desired speed with a smaller overshoot, which proves that the proposed controller has better dynamic control performance than the classical controller.

3.4.2 Comparative Tests with Electrical Parameters Variations

To illustrate the robustness of the proposed controller, the influence of the variations of the electrical parameters is investigated. The stator resistance R of the IPMSM varies as a function of the temperature, while the ϕ_f, L_d, and L_q vary as a function of the operating current. The R increases as the temperature rises, and the ϕ_f and L_q decrease as the i_q increases. However, the L_d slightly increases as the i_d is negative.

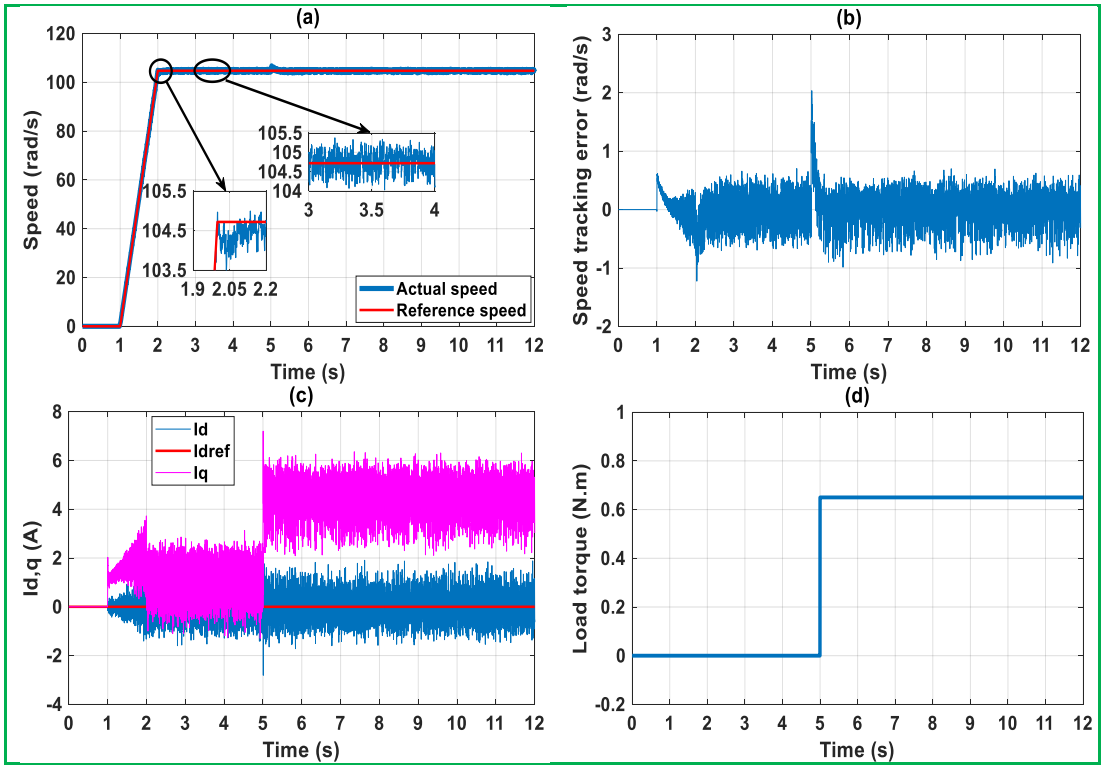


Figure 3. 6. IBC simulation results under electrical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

Variations of parameters are intentionally introduced at the same time into the controller's schemes: $\Delta R = +0.5R$, $\Delta L_d = +0.1L_d$, $\Delta L_q = -0.3L_q$, and $\Delta\phi_f = -0.2\phi_f$. The dynamic responses of the proposed IBC involving the reference and actual rotor speeds and their zooms, the speed tracking error, the *dq*-axis currents, and the load torque signals are depicted in Figure 3.6. Furthermore, the dynamic responses of the PI controller including the same signals are shown in Figure 3.7.

Figures 3.6(a) and 3.7(a) show the speed response of each algorithm when the motor is running at 104.72 rad/s and a load of 0.65 N.m is applied at t = 5 s. We observe that the overshoots of the two methods are 0.238 % and 5.639 %, the settling times are 0 s and 1.87 s, and the steady-state errors are 0.649 % and 0.401 %, respectively. According to the simulation results, the proposed controller has a high fast response with a small overshoot compared with the conventional controller. Also, when load torque is applied, the results show that the IBC gives less dipping in speed tracking (2.04 rad/s) than the PI controller (30.887 rad/s) and has a shorter recovery time (0.82 s) than the PI controller (1.92 s) has. It can be seen that the variations in the electrical parameters affect the overshoot, steady-state error, and dipping in speed tracking of the two controllers.

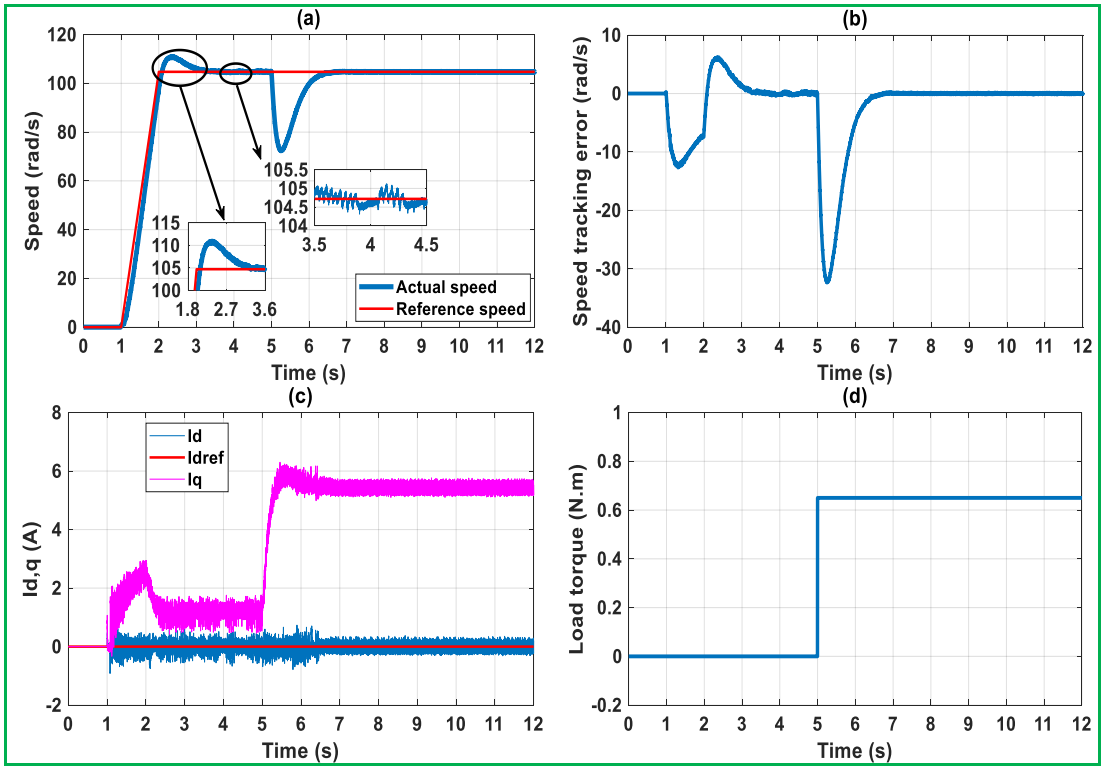


Figure 3. 7. PI controller simulation results under electrical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

Figures 3.6(c) and 3.7(c) show the current waveforms of the dq-axis. We can observe a good decoupling of the d-axis current component introduced by the two controllers, the magnitude of the q-axis current is proportional to the load torque shown in Figures 3.6(d) and 3.7(d). We can observe also that the currents of the proposed controller have more ripples content than the case of the nominal motor parameters. Therefore, it was proved that the developed IBC is quite robust under the variations of the electrical parameters and can achieve better control performance than the PI controller in this case too. The comparative analysis of the control performance is illustrated in Table 3.2.

Table 3. 2. Performance comparison of two control methods under electrical parameters variations.

	IBC	PI Controller
Overshoot (%)	0.238	5.639
Settling Time (s)	0	1.87
Steady-State Error (%)	0.649	0.401
Dipping in Speed (rad/s)	2.04	30.887
Recovery Time (s)	0.82	1.92

The speed dynamic behavior of the two algorithms after a change in the desired speed under electrical parameters variations and without loading the motor is compared. The desired speed is as equation (3.34). According to Figures 3.8 and 3.9, which include the motor speed and the speed tracking error, it can be seen that the PI controller has obvious overshoot, and the speed control performance is poor compared with the IBC. The proposed controller can maintain good speed control performance even if the electrical parameters of the controller deviate significantly from the nominal values. The overall performance is still good.

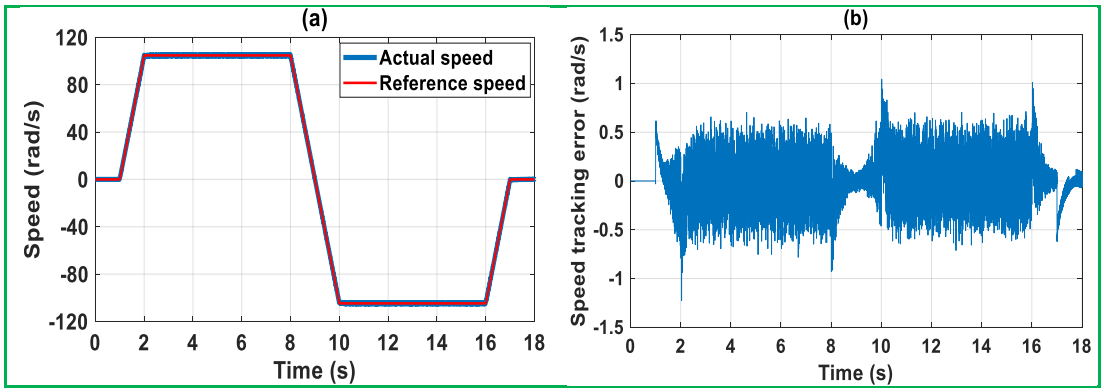


Figure 3. 8. IBC simulation results under electrical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

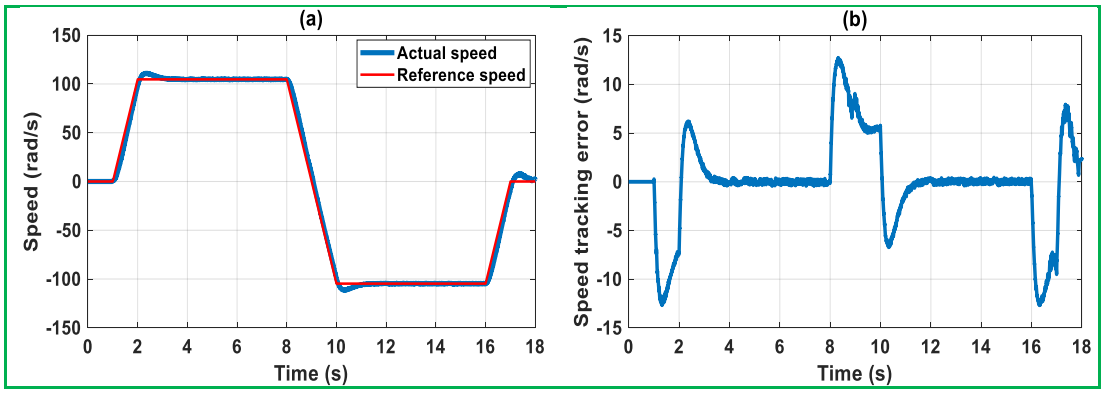


Figure 3. 9. PI controller simulation results under electrical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

3.4.3 Comparative Tests with Mechanical Parameters Variations

To further investigate the effectiveness of the developed control scheme, the simulations under mechanical parameters variations are presented. Variations of parameters are intentionally introduced at the same time into the controller's schemes: $\Delta J = +0.5J$ and $\Delta F = +0.5F$. The dynamic responses of the proposed IBC involving the reference and actual rotor speeds and their zooms, the speed tracking error, the *dq*-axis currents, and the load torque signals are depicted in Figure 3.10. Furthermore, the dynamic responses of the PI controller including the same signals are shown in Figure 3.11.

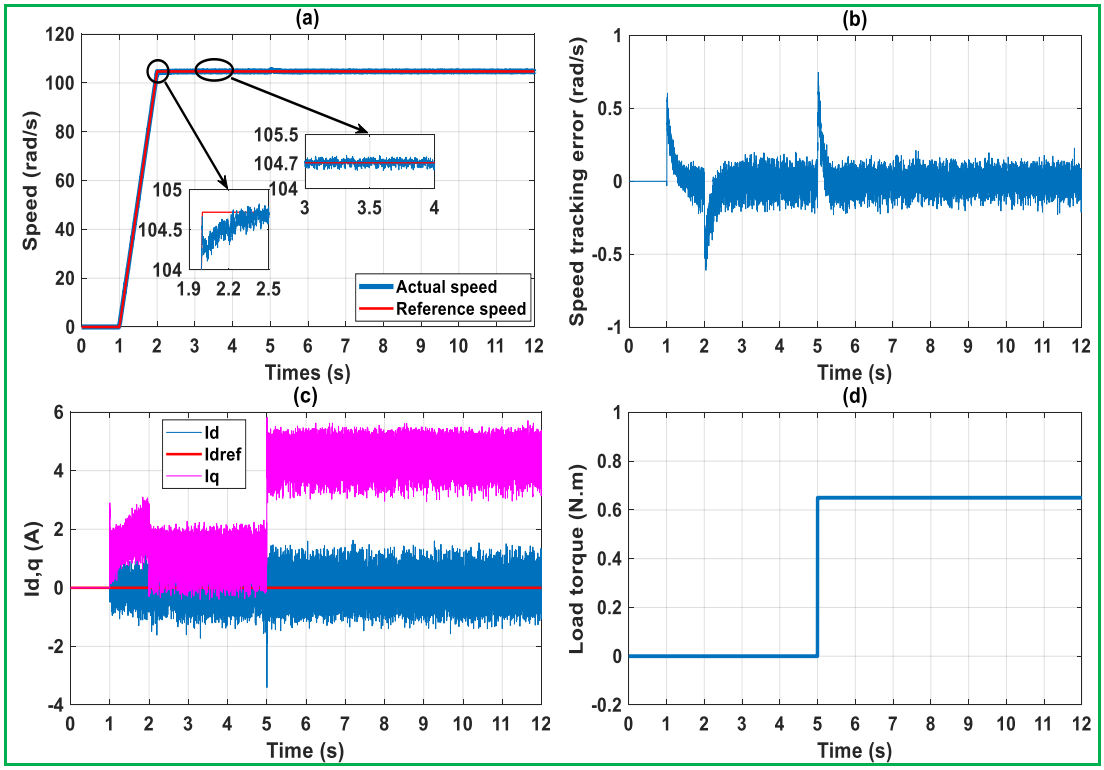


Figure 3. 10. IBC simulation results under mechanical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

Figures 3.10(a) and 3.11(a) show the speed response of each algorithm when the motor is running at 104.72 rad/s and a load of 0.65 N.m is applied at t = 5 s, we observe that the undershoot/overshoot of the two methods are -0.582 % and 6.655 %, the settling times are 0 s and 1.99 s, and the steady-state errors are 0.21 % and 0.191 %, respectively. According to the simulation results, the proposed controller is featured by a faster response than the PI controller has. Also, when the motor is loaded, the IBC provides a fast and precise response with a speed dipping of 0.75 rad/s and a recovery time of 0.8 s, while the PI controller provides more dipping in speed about 24.5 rad/s, and a long recovery time of about 3.37 s. We can remarque that the variations of the mechanical parameters affect the recovering time of the PI controller and the overshoot for both controllers. Figures 3.10(c) and 3.11(c) show the current waveforms of the *dq*-axis. It can be seen that the *d*-axis current is well decoupled and quite regulated to be zero, the magnitude of the *q*-axis current is proportional to the load torque shown in Figures 3.10(d) and 3.11(d). We can also observe that the *dq*-axis currents of the IBC have a larger ripple content than the nominal motor parameters case.

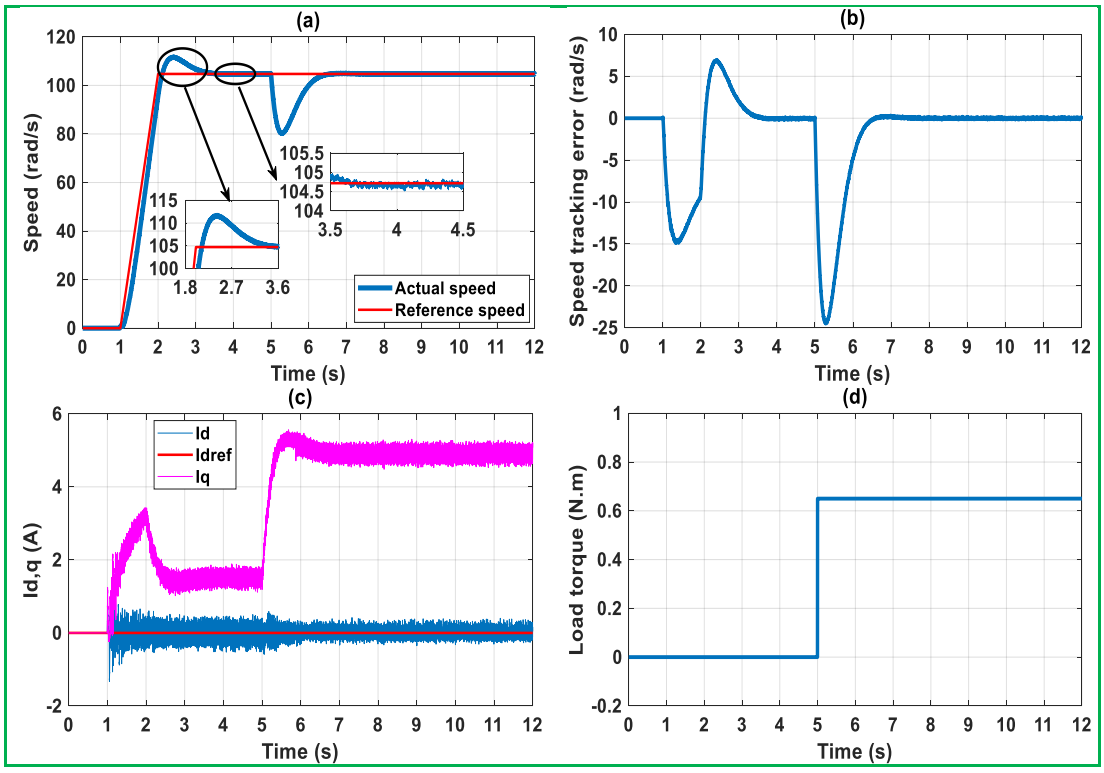


Figure 3. 11. PI controller simulation results under mechanical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

From the simulation results, we can see that the proposed controller can guarantee strong robustness against the variations of mechanical parameters and can

achieve better control performance than the conventional controller in this case as well. A detailed comparison is shown in Table 3.3.

Table 3. 3. Performance comparison of two control methods under mechanical parameters variations.

	IBC	PI Controller
Undershoot/Overshoot (%)	0	6.655
Settling Time (s)	0	1.99
Steady-State Error (%)	0.21	0.191
Dipping in Speed (rad/s)	0.75	24.5
Recovery Time (s)	0.8	3.37

In the following test, the dynamic speed behavior after a change in the reference speed under mechanical parameters variations and without applying any load to the motor shaft is analyzed. The reference speed is given by equation (3.34). In this case, it can be seen that the IBC can rapidly track the reference speed with small stability error, fast response, and small overshoot. We can note that the designed controller shows strong robustness against motor parameters changes compared to the PI controller, as shown in Figures 3.12 and 3.13.

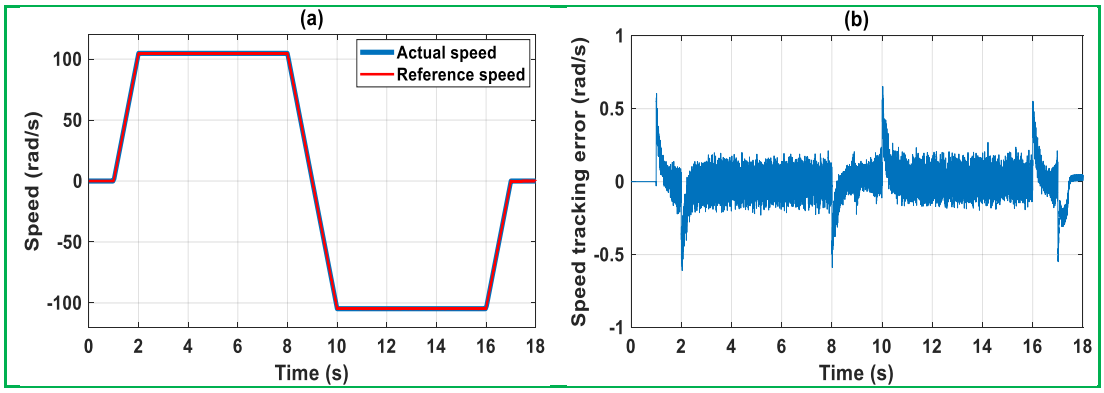


Figure 3. 12. IBC simulation results under mechanical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

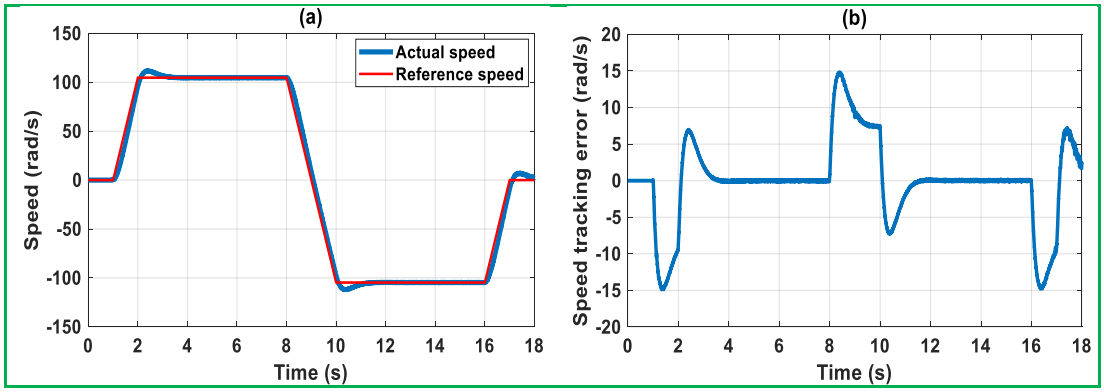


Figure 3. 13. PI controller simulation results under mechanical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

3.5 EXPERIMENTAL RESULTS

In this section, the proposed IBC is implemented on a hardware testbed and its performance is compared to the conventional PI controller experimentally. Motor parameters are presented in Table A.1 (Appendix A), and the experimental tests were carried out on the testbed as shown in Figure A.1 (Appendix A). The experiments in this section are done under both nominal motor parameters and parameter uncertainties. It is known that the IPMSM parameters can be modified simply in the simulations, but the direct modification of the IPMSM parameters in the experiments is difficult. In general, there can be another solution by indirectly changing the IPMSM parameters in a real motor drive by simply changing the drive parameters in the control scheme [98]. Therefore, to carry out experiments on the IBC by changing the electrical and mechanical motor parameters (R, L_d, L_q, ϕ_f, J, and F), the motor parameters in the controller are indirectly changed rather than those in the real IPMSM drive.

3.5.1 Comparative Experiment Under the Nominal Conditions

In the first experiment, the motor runs at 104.72 rad/s, and a step load of 0.65 N.m is applied to the motor at t = 5 s. Figures 3.14 and 3.15 show the response of each algorithm under nominal motor parameters.

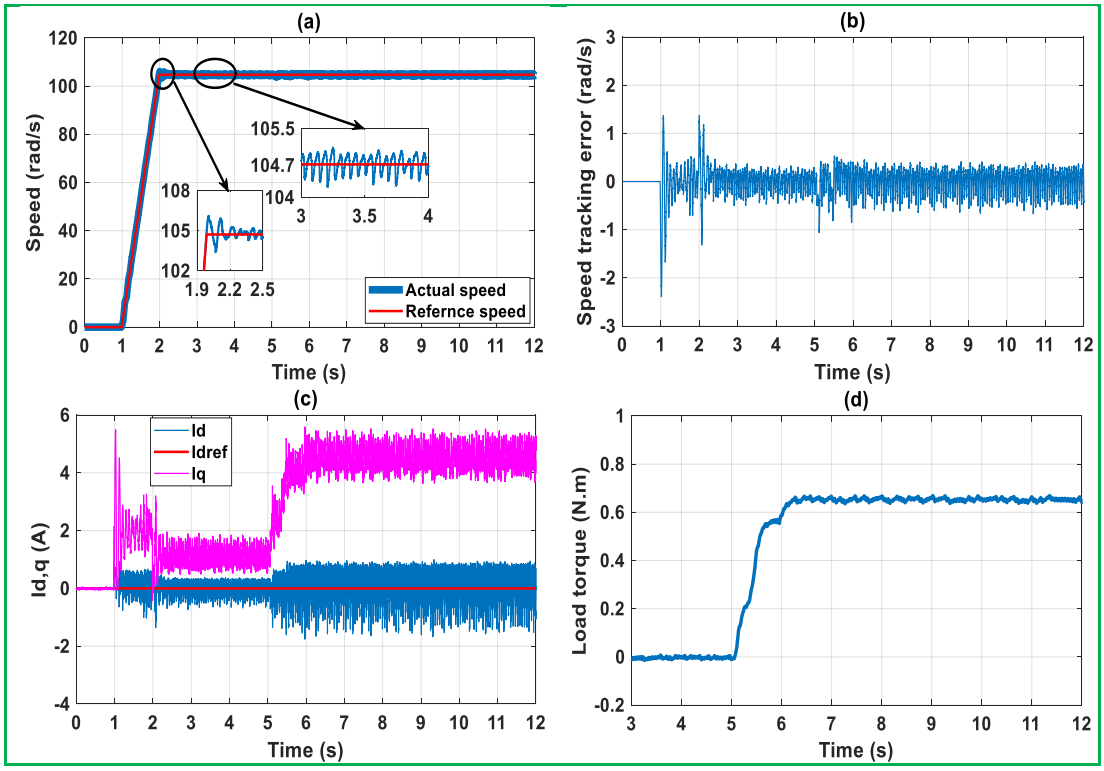


Figure 3. 14. IBC experimental results under nominal motor parameters: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

Figures 3.14(a)-3.14(b), and 3.15(a)-3.15(b) present the speed and the speed tracking error for both controllers, two sections of the speed figures are magnified to show the transient response and the steady-state response. In regard to the IBC compared to the PI controller, the overshoot, the settling time and steady-state error are (1.337 %, 0.026 s, 0.477 %) and (8.976 %, 1.565 s, 0.525 %), respectively. It can be seen that the IBC has a smaller overshoot, faster transient response, and lower steady-state error compared to the PI controller. When the load torque is applied, the PI controller has a larger dipping in speed tracking about 17.8 rad/s and a long recovery time of about 2.7 s. But compared to IBC, the tracking error can rapidly converge to zero with a smaller dip of about 1.05 rad/s after 1.26 s. Figures 3.14(c) and 3.15(c) show the response of the *dq*-axis currents based on the IBC and PI controller, it can be seen that they can immediately converge to the stable values and the *d*-axis current component is well decoupled and is regulated quite well to be zero, we can also see that the *dq*-axis currents of the IBC exhibits obvious oscillation compared to the *dq*-axis currents of the PI controller.

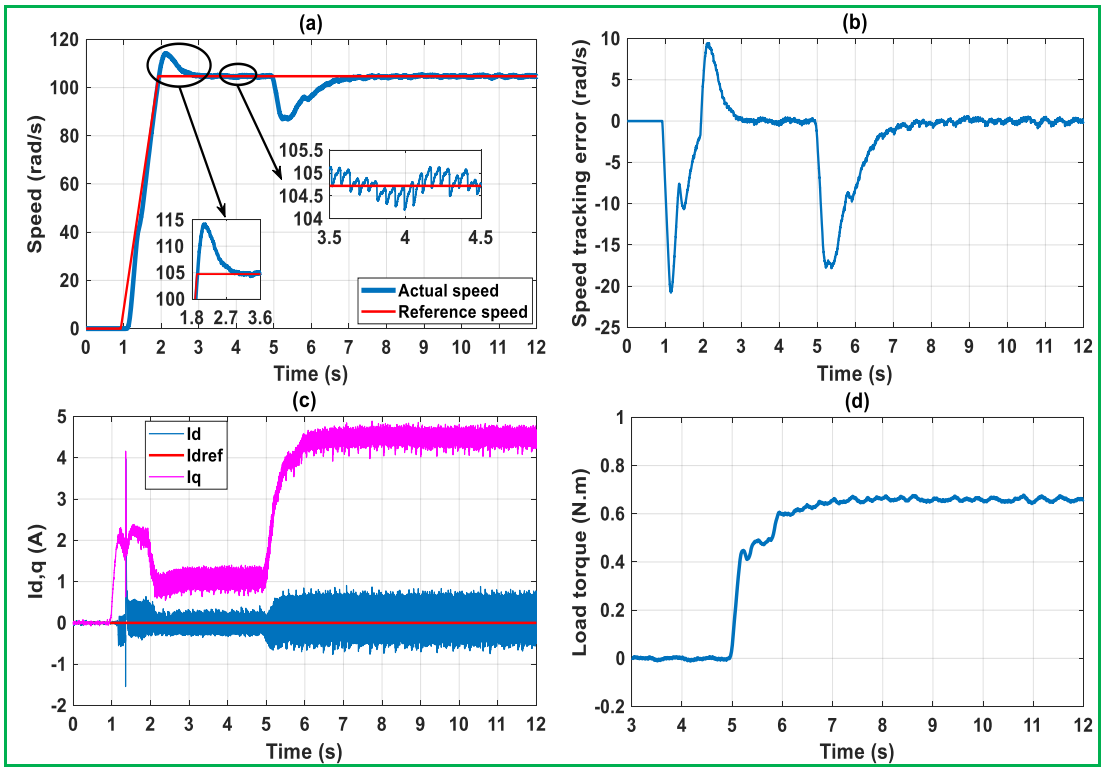


Figure 3. 15. PI controller experimental results under nominal motor parameters: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

Figures 3.14(d) and 3.15(d) show the load torque. Based on the obtained results, we can notice that the proposed controller has a significantly better performance

compared to the PI controller in the case of nominal motor parameters. A detailed comparative analysis of the performance of the two controllers is shown in Table 3.4.

Table 3. 4. Performance comparison of two control methods under nominal motor parameters.

	IBC	PI Controller
Overshoot (%)	1.337	8.976
Settling Time (s)	0.026	1.565
Steady-State Error (%)	0.477	0.525
Dipping in Speed (rad/s)	1.05	17.8
Recovery Time (s)	1.26	2.7

In the next experiment, the speed dynamic behavior is analyzed after a change in the reference speed under the no-load condition as equation (3.34).

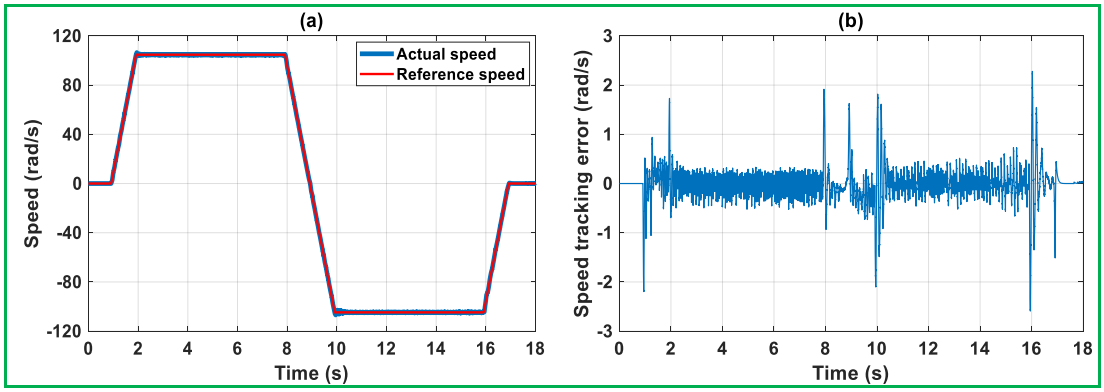


Figure 3. 16. IBC experimental results under nominal motor parameters and change of speed reference: (a) motor speed (b) speed tracking error.

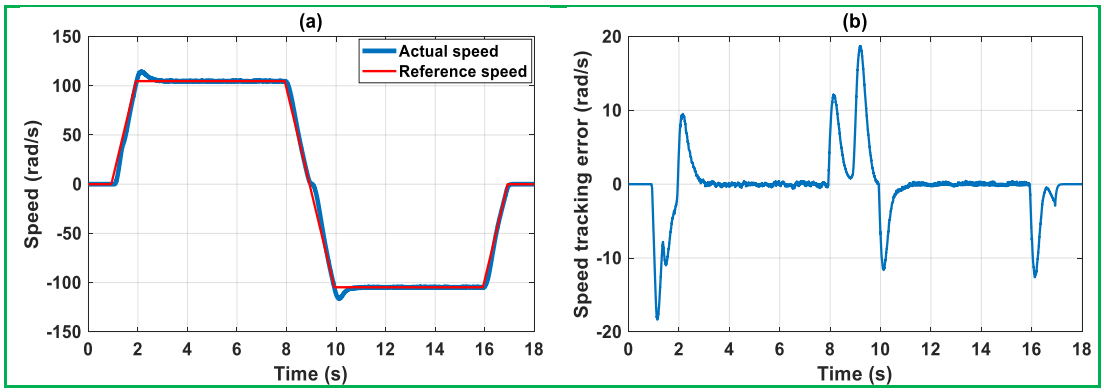


Figure 3. 17. PI controller experimental results under nominal motor parameters and change of speed reference: (a) motor speed (b) speed tracking error.

As seen in Figures 3.16 and 3.17, the comparisons of actual speed and the speed tracking error of the motor are made, including the IBC and PI controller. The parameters of the motor used in the controllers are all nominal values. The experimental results show that the PI controller is not able to follow fast enough precisely the desired speed in both directions, in particular when changing the direction of rotation. Also, the motor speed under this controller shows a large overshoot. It is

observed that the IBC has a fast transient response and a smoother steady-state response in tracking the desired speed with a smaller overshoot, which proves that the proposed controller has better dynamic control performance than the classical controller.

3.5.2 Comparative Experiment with Electrical Parameters Variations

To illustrate the robustness of the proposed controller, the influence of the variations of the electrical parameters is investigated. Variations of parameters are intentionally introduced at the same time into the controller's schemes: $\Delta R = +0.5R$, $\Delta L_d = +0.1L_d$, $\Delta L_q = -0.3L_q$, and $\Delta\phi_f = -0.2\phi_f$. The dynamic responses of the proposed IBC involving the reference and actual rotor speeds and their zooms, the speed tracking error, the *dq*-axis currents, and the load torque signals are depicted in Figure 3.18. Furthermore, the dynamic responses of the PI controller including the same signals are shown in Figure 3.19. Figures 3.18(a) and 3.19(a) show the speed response of each algorithm when the motor is running at 104.72 rad/s and a load of 0.65 N.m is applied at t = 5 s. We observe that the overshoots of the two methods are 1.241 % and 8.212 %, the settling times are 0 s and 1.53 s, and the steady-state errors are 0.525 % and 0.496 %, respectively.

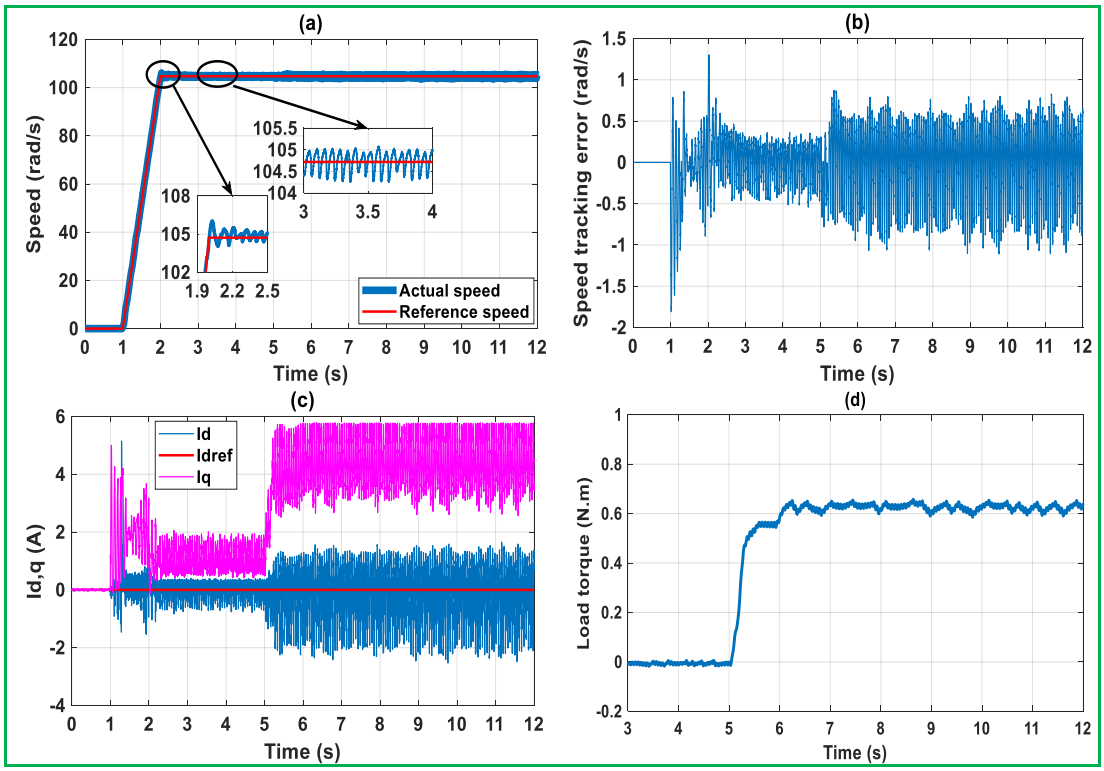


Figure 3. 18. IBC experimental results under electrical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

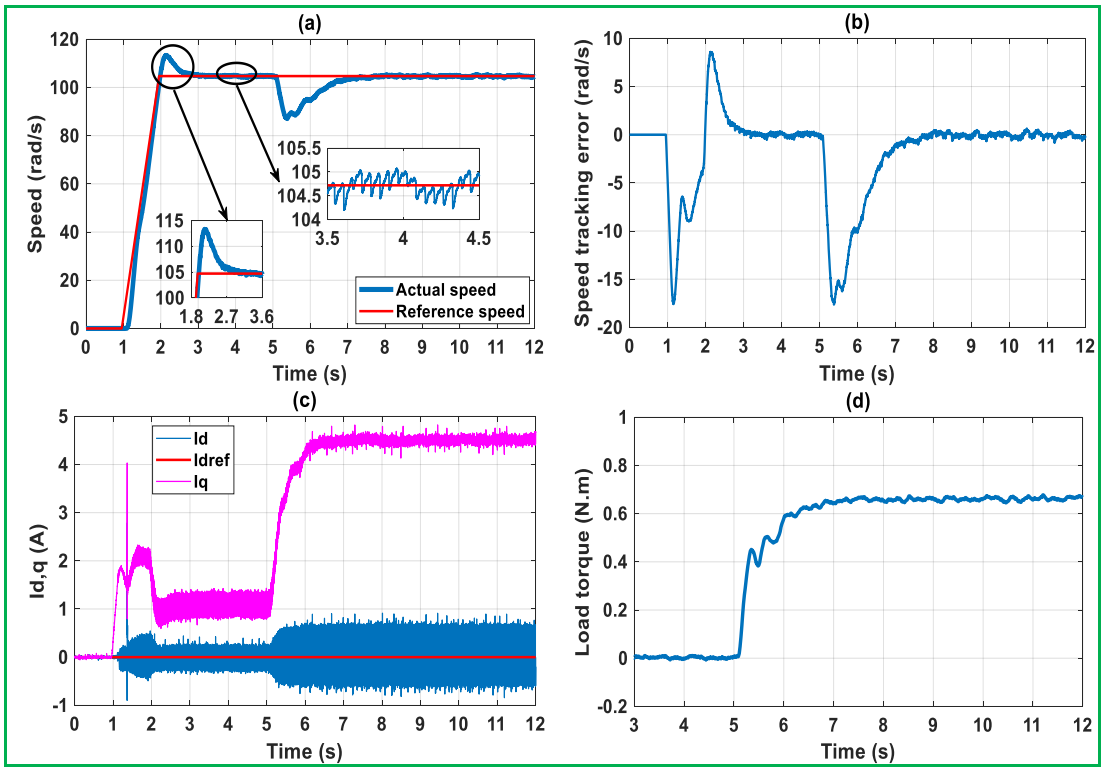


Figure 3. 19. PI controller experimental results under electrical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

According to the experimental results, the proposed controller has a high fast response with a small overshoot compared with the conventional controller. Also, when load torque is applied, the results show that the IBC gives less dipping in speed tracking (0.8 rad/s) than the PI controller (17.6 rad/s) and has a shorter recovery time (1.29 s) than the PI controller (3.25 s) has. It can be seen that the variations in the electrical parameters slightly affect the recovery time of the two controllers.

Figures 3.18(c) and 3.19(c) show the current waveforms of the *dq*-axis. We can observe a good decoupling of the *d*-axis current component introduced by the two controllers, the magnitude of the *q*-axis current is proportional to the load torque shown in Figures 3.18(d) and 3.19(d). We can observe also that the currents of the proposed controller have more ripples content than the case of the nominal motor parameters. Therefore, it was proved that the developed IBC is quite robust under the variations of the electrical parameters and can achieve better control performance than the PI controller in this case too. The comparative analysis of the control performance is illustrated in Table 3.5.

Table 3. 5. Performance comparison of two control methods under electrical parameters variations.

	IBC	PI Controller
Overshoot (%)	1.241	8.212
Settling Time (s)	0	1.53
Steady-State Error (%)	0.525	0.496
Dipping in Speed (rad/s)	0.8	17.6
Recovery Time (s)	1.29	3.25

The speed dynamic behavior of the two algorithms after a change in the desired speed under electrical parameters variations and without loading the motor is experimentally compared. The desired speed is as equation (3.34). According to Figures 3.20 and 3.21, which include the motor speed and the speed tracking error, it can be seen that the PI controller has obvious overshoot, and the speed control performance is poor compared with the IBC. The proposed controller can maintain good speed control performance even if the electrical parameters of the controller deviate significantly from the nominal values. The overall performance is still good.

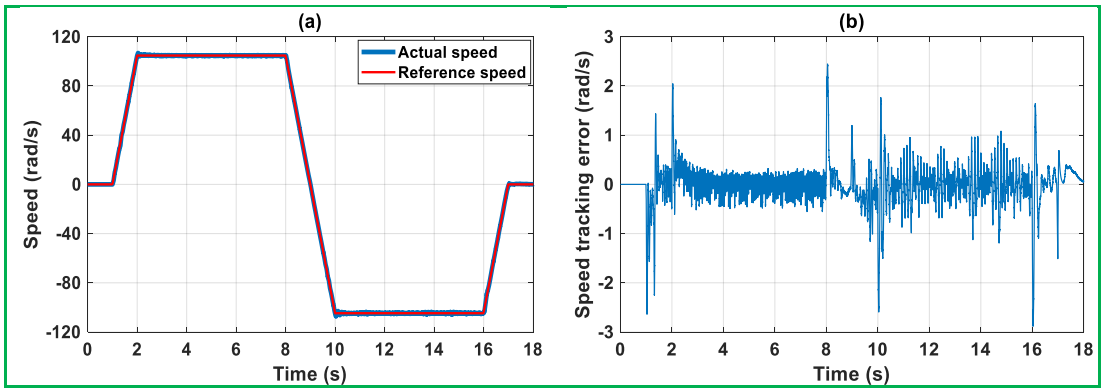


Figure 3. 20. IBC experimental results under electrical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

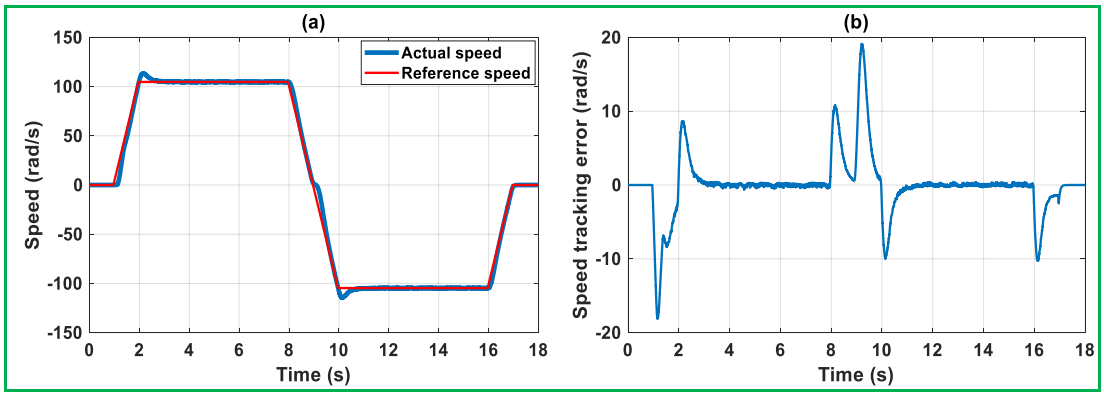


Figure 3. 21. PI controller experimental results under electrical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

3.5.3 Comparative Experiment with Mechanical Parameters Variations

To further investigate the effectiveness of the developed control scheme, the experimental tests under mechanical parameters variations are presented.

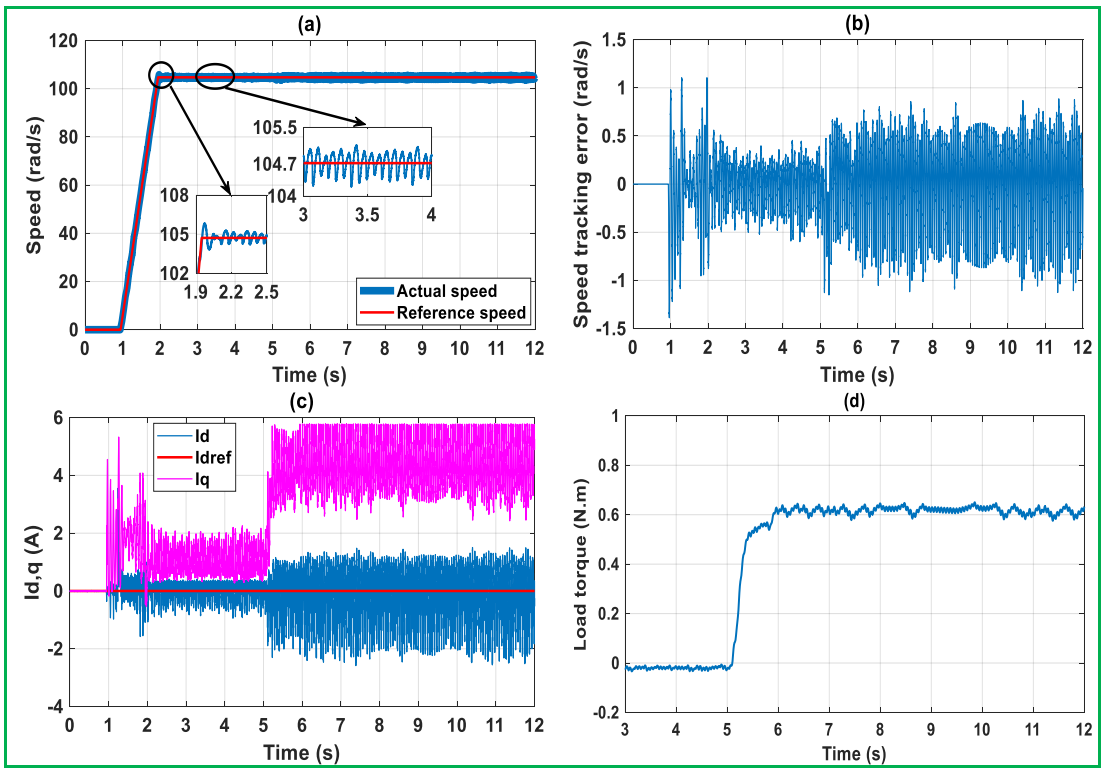


Figure 3. 22. IBC experimental results under mechanical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

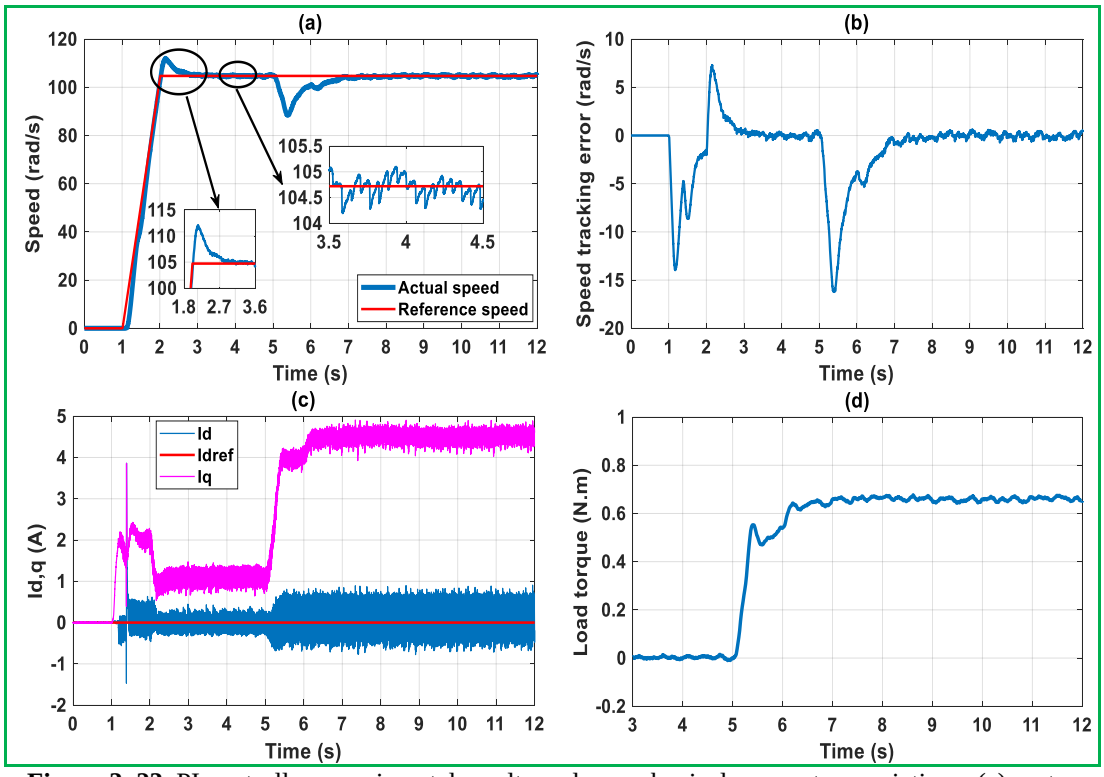


Figure 3. 23. PI controller experimental results under mechanical parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis currents (d) load torque.

Variations of parameters are intentionally introduced at the same time into the controller's schemes: $\Delta J = +0.5J$ and $\Delta F = +0.5F$. The dynamic responses of the proposed IBC involving the reference and actual rotor speeds and their zooms, the speed tracking error, the *dq*-axis currents, and the load torque signals are depicted in Figure 3.22. Furthermore, the dynamic responses of the PI controller including the same signals are shown in Figure 3.23.

Figures 3.22(a) and 3.23(a) show the speed response of each algorithm when the motor is running at 104.72 rad/s and a load of 0.65 N.m is applied at t = 5 s, we observe that the overshoots of the two methods are 1.05 % and 6.971 %, the settling times are 0 s and 1.45 s, and the steady-state errors are the same 0.573 %, respectively. According to the experimental results, the proposed controller is featured by fast response and has a smaller overshoot than the PI controller has. Also, when the motor is loaded, the IBC provides a fast and precise response with a speed dipping of 1.13 rad/s and a recovery time of 1.12 s, while the PI controller provides more dipping in the speed of about 16.2 rad/s and a long recovery time of about 2.82 s. We can remarque that the variations of the mechanical parameters slightly affect the recovering time of the PI controller and the dipping in speed for the IBC. Figures 3.22(c) and 3.23(c) show the current waveforms of the *dq*-axis. It can be seen that the *d*-axis current is well decoupled and quite regulated to be zero, the magnitude of the *q*-axis current is proportional to the load torque shown in Figures 3.22(d) and 3.23(d). We can also observe that the *dq*-axis currents of the IBC have a larger ripple content than the nominal motor parameters case. From the experimental results, we can see that the proposed controller can guarantee strong robustness against the variations of mechanical parameters and can achieve better control performance than the conventional controller in this case as well. A detailed comparison is shown in Table 3.6.

Table 3. 6. Performance comparison of two control methods under mechanical parameters variations.

	IBC	PI Controller
Overshoot (%)	1.05	6.971
Settling Time (s)	0	1.45
Steady-State Error (%)	0.573	0.573
Dipping in Speed (rad/s)	1.13	16.2
Recovery Time (s)	1.12	2.82

In the following experiment, the dynamic speed behavior after a change in the reference speed under mechanical parameters variations and without applying any load to the motor shaft is analyzed. The reference speed is given by equation (3.34). In this case, it can be seen that the IBC can rapidly track the reference speed with small stability error, fast response, and small overshoot. We can note that the designed controller shows strong robustness against motor parameter changes compared to the PI controller, as shown in Figures 3.24 and 3.25.

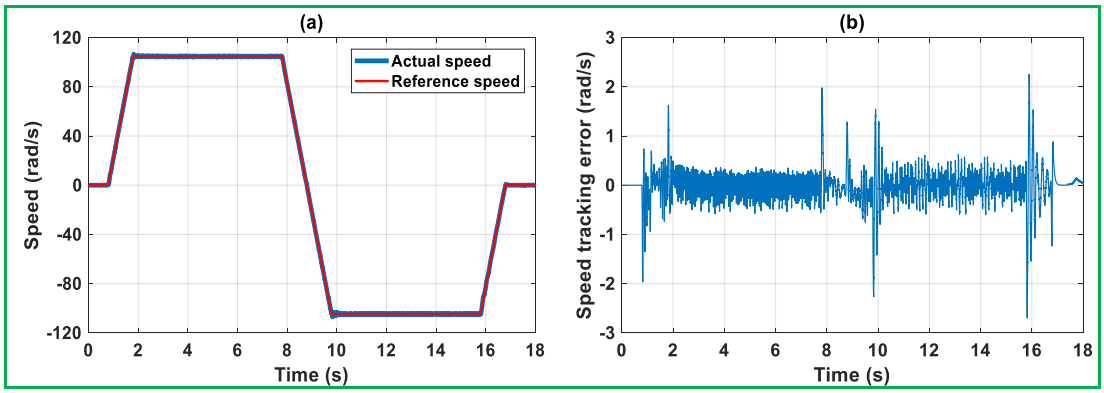


Figure 3. 24. IBC experimental results under mechanical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

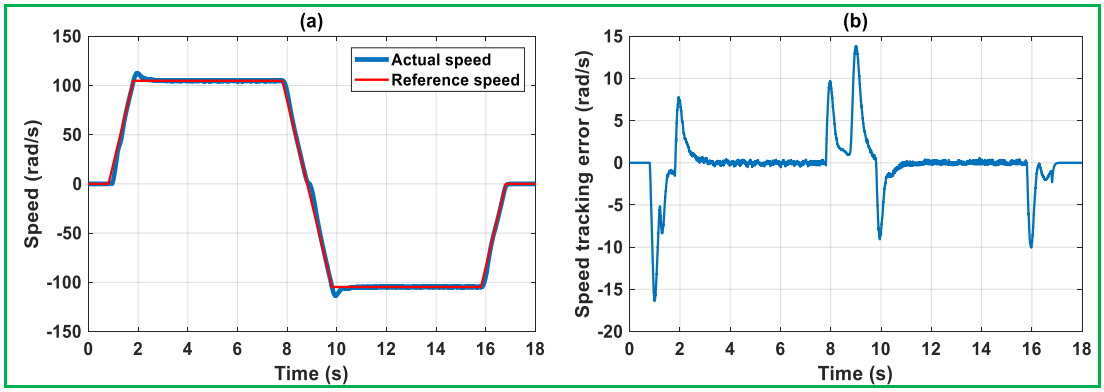


Figure 3. 25. PI controller experimental results under mechanical parameters variations and change of speed reference: (a) motor speed (b) speed tracking error.

The simulation and experimental results show that high precision speed tracking and robust characteristics are established using the proposed controller despite the existence of load torque disturbances, electrical and mechanical parameters uncertainties. Compared to the PI controller, the proposed IBC can achieve better control performances for the IPMSM drive system in case of parameter variations. The reason is that the error caused by the parameter uncertainties will be compensated by the integral actions, because the system inserts the integral of the tracking error into the control law, hence the dependence of the IBC on the precision of the system model is considerably decreased.

3.6 CONCLUSION

In this chapter, an integral backstepping controller-based IPMSM drive system was implemented, and its stability, feasibility, and efficiency were demonstrated through simulations and experiments. Lyapunov's method is used to prove the stability of the control method, which shows that the IBC could guarantee accurate tracking performance in the existence of different uncertainties. The integral action of the tracking error is introduced into the control law, and an IBC is designed. The influence of parameter uncertainties on speed control is decreased, and the dependence of the controller on the exactness of the system model is significantly reduced. The results proved that high precision speed tracking performance and robust characteristics are established by using the proposed control method despite the existence of parameter perturbations and load torque disturbances. A sufficient comparative test was performed with the PI controller, and the results show that the presented strategy ensures the stability of the system, obtains better dynamic and steady-state control performance, and has better load torque disturbances rejection in all conditions.

Chapter 4: Dynamic Surface Control of Interior Permanent Magnet Synchronous Motor

4.1 INTRODUCTION

Backstepping control is a favored subject for research in nonlinear control methods. By introducing the virtual control law, it decomposes the complex nonlinear system into low order systems. Combined with Lyapunov's function, the algorithm derives the final system control law and ensures the stability of the system [91]. But there are some drawbacks in the backstepping approach. One problem is the explosion of complexity caused by the repeated differentiation of virtual input [99], [100]. For instance, the backstepping controller in Appendix B is overly cumbersome, which is representative of the explosion of complexity problem.

The increasing role of the ANN in a wide variety of engineering applications has spurred interest in its application to power electronics and motor drive systems. The ANN has several attractive characteristics that justify this interest, including the capability of wide range approximation of nonlinear functions to any desired degree of accuracy, a fast optimization process, and strong learning ability. These characteristics suggest the enormous potential of the ANN in motor drive systems, including load torque estimation. Several Neural Network (NN) approaches have been proposed in the literature. However, the FFNN scheme is among the simplest and most popular schemes [101]. In real industrial applications, IPMSM systems undergo unknown disturbances, e.g., friction forces, effects of unmodeled dynamics, and exogenous time-varying load disturbances [2].

In this chapter, a FFNN-based load torque observer adapted to the DSC technique is proposed to solve the above problems of the conventional backstepping method for the IPMSM drive system. The FFNN is used to estimate the unknown load torque and its friction effects to solve the problem of the unknown disturbances, e.g., friction forces, effects of unmodeled dynamics, and exogenous time-varying load disturbances, and a DSC technique is proposed to solve the problem of the explosion of complexity by first-order filtering technique at each step of the conventional

backstepping design [31], [102], [103]. The proposed control scheme not only guarantees the boundedness of all signals in the closed-loop system but also reduces the complexity of the system, which alleviates the computational burden. Compared with the existing results on adaptive neural control for IPMSM, the main contributions are that the developed approximation-based neural controller has a simpler structure and both the problems of unknown disturbances and explosion of complexity are considered. As a result, the computational burden of the scheme is alleviated, which will render the designed scheme more suitable for practical applications. Finally, the effectiveness of the design method is verified by simulations to demonstrate the efficiency and robustness against the parameter uncertainties and load disturbances.

This chapter is organized as follows. Section 4.2 presents the model-based observer approach. The FFNN is presented in Section 4.3. In Section 4.4, the control design law based on DSC theory for the IPMSM system is developed. Section 4.5 gives the tuning of the observer and controllers. In section 4.6, the simulation results were presented to compare the static and dynamic performance of the proposed controller versus the conventional backstepping controller under different operating conditions to verify the robustness and stability of the proposed control scheme, also comparing the FFNN-based observer with the load torque Linear Extended State Observer (LESO). Finally, Section 4.7 presents the conclusion of this chapter.

4.2 MODEL-BASED OBSERVER APPROACH

Due to the high cost of direct measurement of real torque under the working state of the motor, and since direct measurement of torque is greatly affected by the precision of the instrument and the response speed of the instrument, the observation of the motor torque is usually realized by the algorithms [104]. For that in this section, a LESO is proposed to estimate the load torque signal and associated friction terms. From equation (2.30), we can get:

$$J\frac{d\omega_r}{dt} = T_{em} - \underbrace{\left[T_L + F\omega_r\right]}_{\tau_L} \tag{4.1}$$

where the function τ_L is the unknown time-varying load torque, which includes a viscous friction torque component: $F\omega_r$.

For this design, the measured direct and quadrature currents $u_1 = i_d$ and $u_2 = i_q$ are considered as inputs, while the angular velocity is considered as the measured output $y_1 = \omega_r$. The nominal value of the rotor moment of inertia J is assumed to be known. The estimated value of the load torque parameter is considered to be time-varying and unknown, but uniformly absolutely bounded so that existence of solutions for the estimation error dynamics are guaranteed to exist [105]. For the design of the load torque estimator, we consider a natural ultralocal piecewise linear internal model corresponding to a second-order dynamic model of the load torque. The corresponding LESO observer is obtained within a reduced-order observer philosophy via the artificial injection of exact differentials [106]. Therefore, we consider $\tau_1 = \hat{\tau}_L$ and $\tau_2 = \dfrac{d}{dt}\hat{\tau}_L$, as the observer state variables. Consider then the following LESO load torque estimator:

$$\frac{d\tau_1}{dt} = \frac{d\hat{\tau}_L}{dt} = \tau_2 + c_1\left(\tau_L - \tau_1\right) \tag{4.2}$$

$$\frac{d\tau_2}{dt} = \frac{d^2\hat{\tau}_L}{dt^2} = c_0\left(\tau_L - \tau_1\right) \tag{4.3}$$

Substituting the indirect measurement of the load torque τ_L given by (4.1) into (4.2) and (4.3), we have

$$\dot{\tau}_1 = \tau_2 + c_1\left(\frac{p}{J}\left[\left(L_d - L_q\right)u_1 + \phi_f\right]u_2 - J\dot{y}_1 - \tau_1\right)$$

$$\dot{\tau}_2 = c_0\left(\frac{p}{J}\left[\left(L_d - L_q\right)u_1 + \phi_f\right]u_2 - J\dot{y}_1 - \tau_1\right)$$

Setting $\Psi_1 = \tau_1 + c_1 J y_1$ and $\Psi_2 = \tau_2 + c_0 J y_1$, we have

$$\tau_1 = \hat{\tau}_L = \psi_1 - c_1 J y_1$$

$$\frac{d\psi_1}{dt} = -c_1\psi_1 + \psi_2 + \left(c_1^2 J - c_0 J\right)y_1 + c_1\frac{p}{J}\left[\left(L_d - L_q\right)u_1 + \phi_f\right]u_2$$

$$\tau_2 = \frac{d}{dt}\hat{\tau}_L = \psi_2 - c_0 J y_1$$

$$\frac{d\psi_2}{dt} = -c_0\psi_1 + c_0 c_1 J y_1 + c_0\frac{p}{J}\left[\left(L_d - L_q\right)u_1 + \phi_f\right]u_2 \tag{4.4}$$

The estimated error evolves according to

$$\frac{d^2}{dt^2}e_\tau + c_1\frac{d}{dt}e_\tau + c_0 e_\tau = \frac{d^2}{dt^2}\tau_L \tag{4.5}$$

where $e_\tau = \tau_L - \hat{\tau}_L$. Since $\ddot{\tau}_L(t)$ is assumed to be uniformly absolutely bounded, then there exist design coefficients c_0 and c_1 for the observer, such that the estimation errors are uniformly asymptotically convergent to a small neighborhood of the origin of the phase space of the estimation error dynamics. The faster the estimation error dynamics, the smaller the convergence neighborhood of the origin. The value of the gain parameters is customarily chosen via a term-by-term comparison of (4.5) with those of a desired, nominal, dominating, second-degree characteristic polynomial $p_{d-obs}(s) = (s^2 + 2\xi\omega_n s + \omega_n^2)$. Therefore, $c_1 = 2\xi\omega_n$, $c_0 = \omega_n^2$.

The model-based approaches like the one presented in this section require knowledge of the mechanical and electrical parameters of the IPMSM, which in some situations are not readily available or difficult to obtain. Because of this potential dilemma, a new approach should be considered-the neural observer. Due to the many advantages of the previously described approach, the FFNN-based neural observer is in principle based on the LESO observer structure developed.

4.3 FEEDFORWARD NEURAL NETWORK

ANNs are characterized by their nonlinear behavior, parallel processing, and their automatic optimization and learning capabilities. These advantages have been behind the increasing popularity of ANNs for numerical modeling, estimation, and control, especially for systems on which little is known about their dynamics and operating environments. The NN universal approximation theorem [107] shows that ANNs can approximate any arbitrary nonlinear function to any degree of accuracy with a single hidden layer; it is therefore a powerful tool for systems with unknown or uncertain dynamics. The network's learning mechanism is often carried out to minimize the network's output error based on a user-defined feedback signal [108].

ANN is a computational model, which is based on biological NN. To build ANN, artificial neurons, also called nodes, are interconnected. The architecture of the NN is very important for performing a particular computation. A NN combines several processing layers, using simple elements operating in parallel. The network consists of an input layer, one or more hidden layers, and an output layer. In each layer, there

are several nodes or neurons, and the nodes in each layer use the outputs of all nodes in the previous layer as inputs, such that all neurons interconnect with each other through the different layers. Each neuron typically is assigned a weight that is adjusted during the learning process and decreases or increases in the weight change the strength of that neuron's signal [109].

An artificial neuron is an abstraction of biological neurons and the basic unit in an ANN. The artificial neuron receives one or more inputs and sums them to produce an output. Usually, the sums of each node are weighted, and the sum is passed through a function known as an activation or transfer function. Figure 4.1 shows the artificial neuron model [110].

Activation functions are bio-inspired mathematical equations to represent the firing action potential in a neuron or node. They are a transformation and nonlinear mapping of inputs from one layer into the next layer. Moreover, the set of interconnected neurons in an intricate manner forms diverse kinds of NNs. Remarkably, decisions are made at the nonlinearity of those functions, which exist at zero most of the time. This property of activation functions enabled NNs to learn from complex and higher-order data to provide precise predictions and classifications. Moreover, their nonlinearity affects the convergence of NNs and plays a vital role in specifying the convergence speed and computational efficiency. Popular activation functions include rectifier linear unit (ReLU) and sigmoid [111].

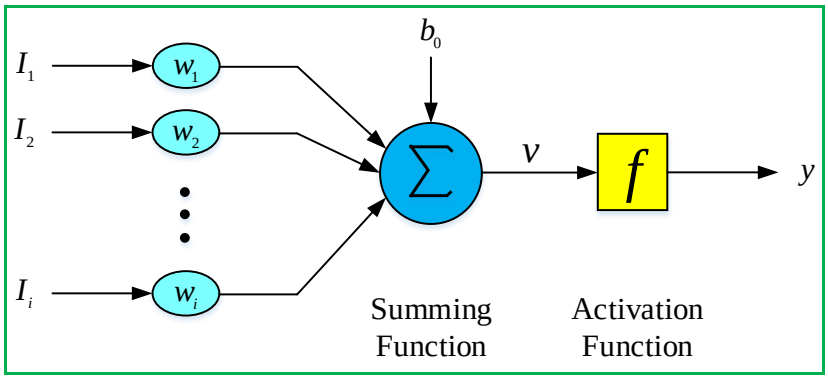


Figure 4. 1. Artificial neuron model.

The synapses or connecting links: that provide weights, w_i, to the input values, I_i for i = 1, 2, ...; An adder: that sums the weighted input values to compute the input to the activation function

$$v = b_0 + \sum_{i=1} w_i I_i \tag{4.6}$$

where, b_o is called the bias, is a numerical value associated with the neuron. It is convenient to think of the bias as he weights for an input I_o whose value is always equal to one, so that;

$$v = \sum_{i=0} w_i I_i \tag{4.7}$$

ANN presumes that the true underlying function that governs the relationship between inputs and outputs is not known a priori. It determines a mathematical function that can properly approach the representation of inputs and outputs.

One of the major aspects of ANN is the training process, which can be either supervised or unsupervised. Supervised learning, i.e., guided learning by ‘‘teacher’’; requires a training set that consists of input vectors and a target vector associated with each input vector. The advantage of supervised training is that the output can be interpreted based on the training values. The disadvantage is that a large number of inputs and outputs are required to guarantee adequate training [112].

Some examples where ANNs have been used to solve various tasks such as pattern recognition [113], parameter estimation [114], prediction of values [115], and of course in the control of different types of systems [116], [117].

4.3.1 Feedforward Neural Network Structure

FFNN is one of the most popular ANN models for engineering applications due to its simple structure, strong operability, good learning ability, and easy implementation in real-time systems. The FFNN represented in Figure 4.2 comprises three layers; the input layer receives the information, and the output layer has a single neuron and gives the internal calculation result. Between these two layers, there is another layer not visible from the outside called the hidden layer, which is responsible for performing intermediate computations [118].

Determination of the number of hidden layers, hidden neurons, and type of transfer function plays an important role in FFNN model constructions [119]. The number of hidden layers required depends on the complexity of the relationship between the input and the target parameters. It has an impact on the quality of the learning, FFNNs comprising more hidden layers are very rare, given that each new layer increases the quantity of calculations. In the majority of problems, only one hidden layer is sufficient. Hornik et al. [120] proved that FFNN with one hidden layer

is enough to approximate any continuous function. Therefore, one hidden layer was employed in the current research. Besides, transfer functions for the hidden nodes are needed to introduce nonlinearity into the network. In this study, the sigmoid was selected as the activation function of the hidden neurons while a linear activation function was used in the output neuron.

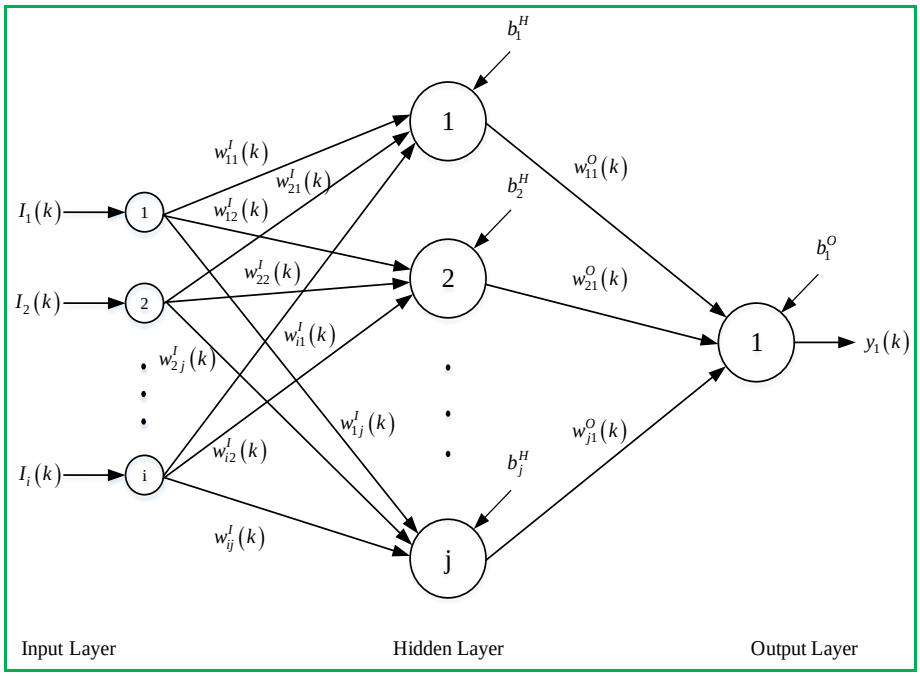


Figure 4. 2. FFNN structure.

Next, the choice of the optimal number of hidden layer neurons is an essential decision in the modeling phase. If an insufficient number of neurons are used, the network will be unable to model complicated data, and the resulting fit will be poor. Many hidden neurons will ensure correct training, and the network will be able to appropriately predict the data it has been trained on, but its performance on new data and its ability to generalize will be compromised. Whereas, with very few hidden neurons, the network may be inept to learn the associations between the input and output variables. In this sense, the error will fail to fall below an adequate level [121]. Thus, a compromise has to be reached between too many and too few neurons in the hidden layer. In this study, the optimal number in the hidden layer was selected by trial and error.

The structure of the FFNN is shown in Figure 4.2 for a system with *i* inputs, *j* neurons in the hidden layer, and one output neuron [118]. The generated output of the output layer neuron and j^{th} hidden layer neuron are given in (4.8) and (4.9), respectively, and the sum of inputs to hidden neurons is given in (4.10).

$$y_1(k) = \sum_j w_{j1}^O x_j(k) \tag{4.8}$$

$$x_j(k) = f\left(S_j(k)\right) j = 1, 2, ..., j \tag{4.9}$$

$$S_j(k) = \sum_i w_{ij}^I I_i(k)\, j = 1, 2, ..., j \tag{4.10}$$

In (4.10), $I_i(k)$ represents the i^{th} input to the NN at discrete time k. The weight matrices w_{ij}^I, and w_{j1}^O represent the input layer weight (connecting input i and hidden layer neuron j), and the output layer weight from hidden layer neuron j to the output neuron, respectively. The superscripts I, and O associated with the weights are used to indicate the input, and output layers, respectively. The function $f(.)$ in (4.9) is the commonly used hyperbolic tangent sigmoid transfer function (tansig), where:

$$f(x) = \frac{2}{1 + e^{(-2x)}} - 1 \tag{4.11}$$

The bias terms shown in Figure 4.2 for the hidden and output layer neurons are also included in (4.8) and (4.10) by considering the bias as a weighted connection with unity input. That is, $w_{o1}^O = b_1^O$ and $x_o(k) = 1$ in (4.8) while $w_{oj}^I = b_j^H$ and $I_o(k) = 1$ in (4.10).

4.3.2 Feedforward Neural Network Training

The objective of training the FFNN is to find optimal connection weights w^* in such a manner that the value of calculated outputs for each example matches the value of desired outputs [16], [122]. This is typically a nonlinear optimization problem, where w^* is given by equation (4.12):

$$w^* = \arg\min E(w) \tag{4.12}$$

where w is the weight matrix and $E(w)$ is an objective function on w, which is to be minimized.

The $E(w)$ is evaluated at any point of w given by equation (4.13):

$$E(w) = \sum_p E_p(w) \tag{4.13}$$

p is the number of examples in the training set and $E_p(w)$ is the output error for each example p. $E_p(w)$ is expressed by equation (4.14):

$$E_p(w) = \frac{1}{2}\sum_j \left(d_{pj} - y_{pj}(w)\right)^2 \tag{4.14}$$

where $y_{pj}(w)$ and d_{pj} are calculated and desired network outputs of the j^{th} output neuron for p^{th} example, respectively. The objective function to be minimized is represented by equation (4.15):

$$E(w) = \frac{1}{2}\sum_p \sum_j \left(d_{pj} - y_{pj}(w)\right)^2 \tag{4.15}$$

For each learning (training) process, the network calculated output value is compared to the desired output value. If there is a difference between the calculated and desired output network, the synaptic weights which contribute to generating a significant error will be changed more significantly than the weight that led to a marginal error. The adaptation of the weights begins at the output neurons and then continues toward the input data. There are many algorithms available to perform this weight selection and adjustment [104]. One of the most popular is the gradient descent, which suffers from slow convergence times and can easily get trapped in local minima within the vector space of w during the learning process; this leads the model to evolve in an accurate direction. Therefore, the Levenberg-Marquardt Algorithm (LMA) was chosen to train the NN. LMA is considered one of the most efficient training algorithms; the study by Hagan and Menhaj [123] proved that LMA is faster and has more stable convergence as compared to gradient descent algorithm.

4.3.3 Levenberg-Marquardt Algorithm

Like the quasi-Newton methods, the LMA was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training Feedforward networks), then the Hessian matrix can be approximated as

$$H = J_c^T J_c \tag{4.16}$$

and the gradient can be computed as

$$g = J_c^T e \tag{4.17}$$

where J_c is the Jacobian matrix that contains the first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The

Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix [123].

The LMA uses this approximation to the Hessian matrix in the following Newton-like update:

$$X_{k+1} = X_k - \left[J_c^T J_c + \mu I \right]^{-1} J_c^T e \tag{4.18}$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function is always reduced at each iteration of the algorithm.

The original description of the LMA is given in [124]. The application of Levenberg-Marquardt to NN training is described in [123] and [125]. This algorithm appears to be the fastest method for training moderate-sized FFNNs (up to several hundred weights). It also has an efficient implementation in MATLAB software, because the solution of the matrix equation is a built-in function, so its attributes become even more pronounced in a MATLAB environment.

4.3.4 FFNN-Based Load Torque Estimation

To estimate the load torque, the mathematical model is used successfully in many techniques [121-124], although they are dependent on machine dynamic parameters. Therefore, control performance is affected by nonlinear dynamics that are not specified in the used model, external load distortion, and systems parameters changes, which when considered by the motor model can result in quite complex analysis.

Given this situation, a new approach should be considered: the NN technique. This technique facilitates the performance of highly nonlinear systems or in cases where the mathematical model is difficult to obtain. Therefore, in load torque estimation, this method provides many advantages, such as robustness, nonlinear adaption, and learning ability.

Several NNs methods for identification and control of nonlinear dynamic systems have been proposed and verified in the literature [126-128]. These papers used NNs to estimate the load torque. Although the methods used in previous works show satisfactory results, the objective of this section is to present a simpler ANN-based observer, providing used training data, experimental results, without using online training. The main idea of the work presented in this section is to estimate the load torque and its friction effects so that the estimation is independent of the parameters, non-necessity for the mathematical model, and easy to implement, exploiting the advantages of ANNs mentioned in the literature to obtain a simple, fast and straightforward estimator. For this, we used a FFNN trained with data gathered from a load torque LESO. The data used in this section is the real data that was collected from the IPMSM testbed (Appendix A). We selected the FFNN for load torque estimation because it can be used to approximate arbitrary continuous functions. Details of the system topology and structure of the component are described in this section.

FFNN Topology for Load Torque Observer

In this work, a multilayer FFNN is chosen to estimate the load torque and its friction effects, and the information obtained from the neural observer is adapted to the controller. The FFNN is composed of one input layer, one or more hidden layers, and one output layer. Since a NN with one hidden layer has the capability to handle most of the complex functions, in this work the FFNN with one hidden layer is constructed. The network consists of three layers, i.e., input, hidden, and output layers as shown in Figure 4.3. The proposed network was used with the sigmoid function (tansig) in the hidden layer and the linear function (purlin) in the output layer.

Based on the preceding section, the FFNN-based neural observer is now developed. Like the model-based observer approach previously outlined, the system inputs to the proposed FFNN-based load torque observer structure are the *dq*-axis currents and the speed while the load torque and its friction effects have been used as a target. Notably, the model was trained with a different number of neurons, out of which, twenty neurons in the hidden layer were found to produce good results for the neural observer. This number was selected to produce fast convergence and robust dynamic capturing capabilities. The output layer contains one neuron since there is only a single output in the ANN-based observer. The number of neurons in the hidden

layer is chosen by trial and error, keeping in mind that the smaller the number is, the better it is in terms of both memory and time required to implement the ANN in the motor control.

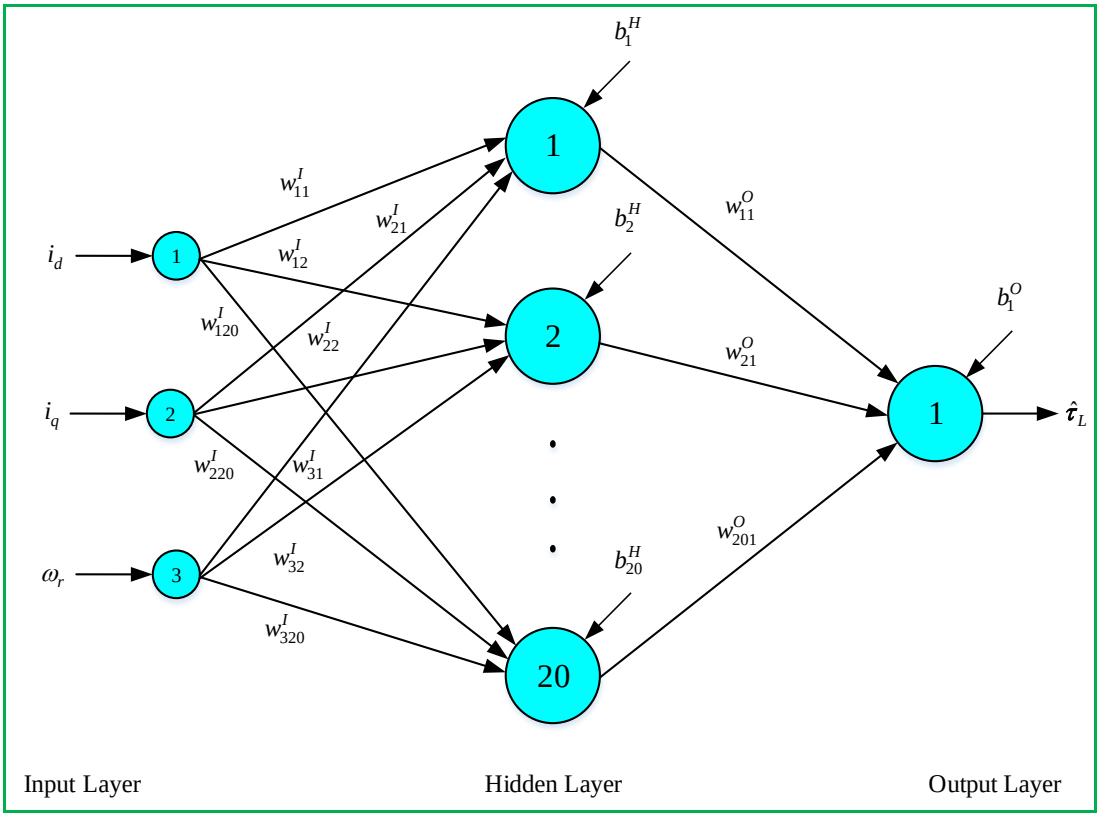


Figure 4. 3. Structure of the Feedforward neural network.

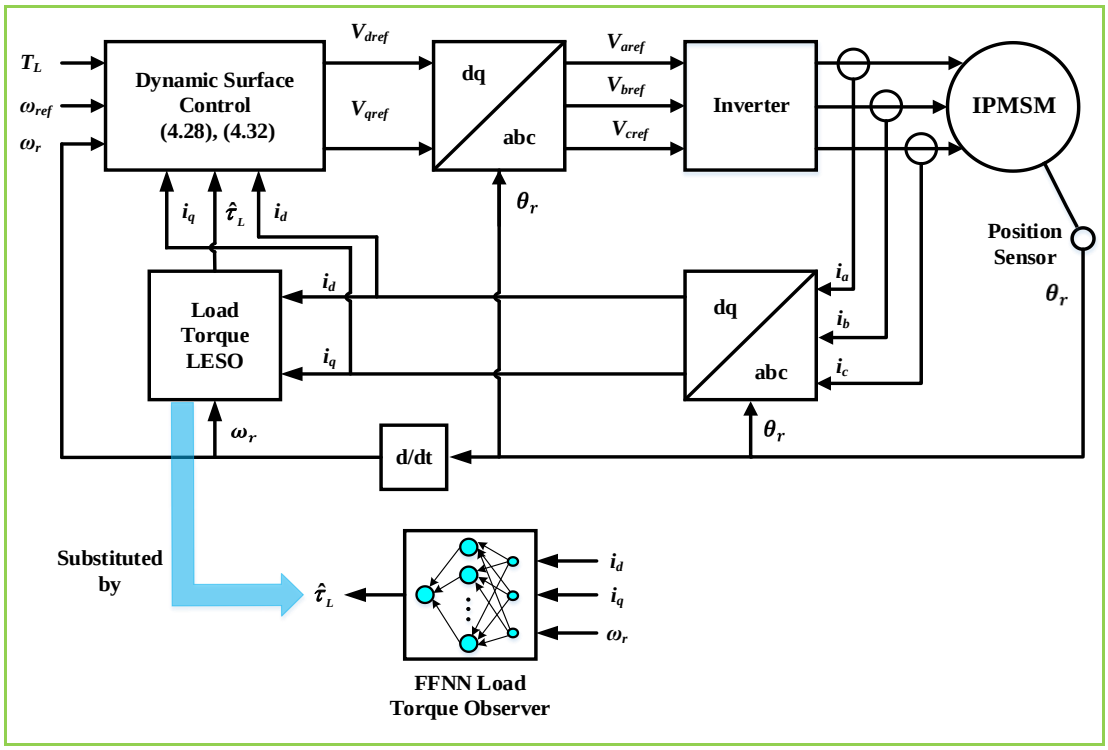


Figure 4. 4. FFNN-based DSC for IPMSM control.

Figure 4.4 shows the block diagram of the control structure with the NN observer, which is trained to replace the load torque LESO. The NN observer which is a static type has replaced the load torque LESO, thus simplifying the control implementation. The information gathered from the observer is adapted to the DSC technique, which controls the speed to give the desired output. Transformations and the currents controller derive the necessary pulses for the inverter.

FFNN Load Torque Observer Training

One of the most used NNs is called FFNN, and the best-known training method for this type of network is backpropagation. One of the advantages that backpropagation presents is a speed of convergence and robustness compared to other types of training. For this reason, in this work, the FFNN has been implemented using backpropagation as a training algorithm. This algorithm is used more for learning this kind of multilayer network. The backpropagation algorithm trains the NN from sample vectors of the system that are of interest for their modeling, such as the texture of an object or, in our case, the load torque of the motor [110].

To summarize, the backpropagation training system consists of the following steps:

- Initialize the weights of the network randomly;
- Enter input data from among those to be used for training;
- Let the network generate an output data vector (forward propagation);
- Compare the network output with the desired output;
- The difference between the generated and the desired output (called error) is used to adjust the weights in the output layer;
- The error spreads backward (hence the name of backpropagation) towards the previous neuron layer and is used to adjust the weights of that layer;
- Continue propagating the error backward and adjusting the weights until the input layer is reached.

The network is trained offline using the backpropagation algorithm with experimental data. The target is the estimated load torque by the load torque LESO. To generate this training data, the motor was running for 16 seconds at a speed of 104.72 rad/sec, and resistive torque (22% of the rated torque) was applied at t = 5 s

and then removed at t = 10 s as shown in Figure 4.5(a), thus generating the needed electric currents, actual speed, and load torque data. The training data set consists of 70589 samples; the number of parameters to be calculated (weights and biases) during the training stage are 101 for the proposed NN. In this case, the parameters of the NN observer and its training are presented in Table 4.1.

Table 4. 1. Neural network parameters.

Parameters	Value
Number of layers	3
Hidden layer neurons	20
Activation function	Sigmoid
Output layer neurons	1
Activation function	Linear
ANN architecture	Feedforward Neural Network
Training algorithm	Backpropagation
Variant	Levenberg-Marquardt
Performance	Mean Squared Error
Epoch	1000

Some observations regarding the values are shown in Table 4.1. First, it will be mentioned that, although the NN training indeed used the backpropagation algorithm, this algorithm has four variants. These variants are listed below:

- Momentum;
- Variable learning ratio;
- Conjugate gradient;
- Levenberg-Marquardt.

Being the LMA, the backpropagation variant with which, generally, better performance is obtained in the tasks assigned to the NN. For this reason, in this implementation, it was decided to use this variant. Although this variant indeed has the highest computational cost, it is compensated by the excellent performance it offers.

Another observation regards the number of epochs that were specified as the maximum value in this implementation. In general, for a NN to converge towards the optimal values of the weights, 1000 to 10,000 epochs are required. However, in the different experiments made, it was generally observed that the NN had convergence in a range of 94 to 413 epochs. The last-mentioned means that training does not take a long time of 413 epochs to obtain a result. Then, this parameter was established in

1000 as a maximum value that training needed to reach the optimal values of the weights.

The above is due in part to three main things. One of them is the use of the LMA for training, which has a faster convergence towards the optimal values of the network weights even if it requires more processing to get convergence of the network. The other is the number of network layers, which, in our case, only have three layers.

Finally, the number of neurons is mentioned; in our implementation, only twenty-one neurons are necessary. Twenty of these neurons are in the hidden layer, while the output layer only has one neuron.

It is important to emphasize that our implementation occupies few neurons, as previously mentioned. This is a feature that will allow us to implement this observer quickly and efficiently in a low-cost processor. The above would give portability to this observer, thus obtaining a lower-cost hardware implementation.

Once the implementation of the IPMSM load torque observer through the NN is presented, the obtained results are presented. Therefore, in section 4.6, the results obtained from this implementation will be presented.

4.4 DYNAMIC SURFACE CONTROL OF IPMSM

Research on the DSC method has progressed considerably since the late 1990s [129], [130]. DSC is an enhanced backstepping control method, whose design process is executed in a step-by-step manner. At each step of design, a feedback controller is designed to guarantee input to-state practical stability of the corresponding subsystem. The primary advantage of DSC is that it can circumvent the problem of the explosion of complexity inherent in the backstepping design procedure, by introducing a first-order low-pass filter of the virtual input at each step of the conventional backstepping approach. On the other hand, adaptive control of various nonlinear systems containing constant but indeterministic parameters has attracted many researchers' attention recently [131]. People usually utilize adaptive control techniques to eliminate the influence of uncertainties. To control the IPMSM with unknown load torque disturbances, friction forces, and unmodeled dynamics, a DSC law combined with a FFNN load torque observer is designed in this section. And then, to verify the advantages of the DSC technique over the conventional backstepping control technique, a comparison between the two methodologies was studied. Moreover, based

on Lyapunov asymptotical stability theory, we prove that the presented control law can render the closed-loop controlled system asymptotically stable. The simulation results indicate that the proposed control law is very effective and robust against uncertainties in the system parameters. Since the feasibility has been taken into account through the design process, the designed controller is more applicable to practical use [38].

In this section, we will present a DSC for IPMSM based on backstepping. The control technique is a nonlinear backstepping control having properties of strength. The pursuit of speed takes place with a high yield by the control voltage v_q as long as the current i_d is kept equal to zero. The calculation of the Lyapunov function is performed recursively, it is based on the previous system state. A new Lyapunov function is constructed by the increase of the Lyapunov function of the previous step, this procedure calculates allows us to ensure overall system stability. The corresponding block diagram of the proposed scheme of speed tracking design is shown in Figure 4.4.

Step 1

For the reference signal θ_{ref}, we define the tracking error variable as $e_1 = \theta_r - \theta_{ref}$. The error dynamic system is computed by $\dot{e}_1 = \omega_r - \omega_{ref}$.

Choose the Lyapunov function as $V_1 = \frac{1}{2}e_1^2$, then its time derivative is given by

$$\dot{V}_1 = e_1\dot{e}_1 = e_1\left(\omega_r - \omega_{ref}\right) \tag{4.19}$$

Construct the virtual control law α_1 as

$$\alpha_1 = -k_1 e_1 + \omega_{ref} \tag{4.20}$$

with $k_1 > 0$ being a positive design control gain. Next, introduce a new state variable α_{1d}. Let α_1 pass through a first-order filter with time constant ε_1 and $\alpha_{1d}(0) = \alpha_1(0)$. The purpose of this filter is to generate α_{1d} and its derivative $\dot{\alpha}_{1d}$ such that $|\alpha_{1d} - \alpha_1|$ is smaller than a given level.

$$\varepsilon_1\dot{\alpha}_{1d} + \alpha_{1d} = \alpha_1,\ \alpha_{1d}(0) = \alpha_1(0) \tag{4.21}$$

Define $e_2 = \omega_r - \alpha_{1d}$.

Step 2

Differentiating e_2 obtains

$$\dot{e}_2 = \frac{P}{J}\left(L_d - L_q\right) i_d i_q + \frac{P}{J}\phi_f i_q - \frac{1}{J}\tau_L - \dot{\alpha}_{1d} \tag{4.22}$$

Now, choose the Lyapunov function candidate as $V_2 = V_1 + \frac{1}{2}e_2^2$. The time derivative of V_2 can be expressed as

$$\dot{V}_2 = -k_1 e_1^2 + e_2\left[\frac{P\left[\left(L_d - L_q\right) i_d + \phi_f\right]}{J} i_q - \frac{1}{J}\tau_L - \dot{\alpha}_{1d}\right] \tag{4.23}$$

Then the virtual control α_2 is constructed as

$$\alpha_2 = \frac{J}{P\left[\left(L_d - L_q\right) i_d + \phi_f\right]}\left[\frac{1}{J}\tau_L + \dot{\alpha}_{1d} - k_2 e_2\right] \tag{4.24}$$

with $k_2 > 0$ being a positive design control gain. Then define a new state variable α_{2d}. Let α_2 pass through a first-order filter with time constant ε_2 to obtain α_{2d} as

$$\varepsilon_2 \dot{\alpha}_{2d} + \alpha_{2d} = \alpha_2,\ \alpha_{2d}(0) = \alpha_2(0) \tag{4.25}$$

and define $e_3 = i_q - \alpha_{2d}$

Step 3

Differentiating e_3 results in the following equation:

$$\begin{aligned}\dot{e}_3 &= \dot{i}_q - \dot{\alpha}_{2d} \\ &= -\frac{R}{L_q} i_q - p\frac{L_d}{L_q}\omega_r i_d - p\frac{1}{L_q}\phi_f \omega_r + \frac{1}{L_q} v_q - \dot{\alpha}_{2d}\end{aligned} \tag{4.26}$$

Choose the following Lyapunov function candidate as $V_3 = V_2 + \frac{1}{2}e_3^2$. Furthermore, differentiating V_3 yields

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 + e_3\left[-\frac{R}{L_q} i_q - p\frac{L_d}{L_q}\omega_r i_d - p\frac{1}{L_q}\phi_f \omega_r + \frac{1}{L_q} v_q - \dot{\alpha}_{2d}\right] \tag{4.27}$$

Now, the control input v_{qref} is designed as:

$$v_{qref} = L_q \left[\frac{R}{L_q} i_q + p \frac{L_d}{L_q} \omega_r i_d + p \frac{1}{L_q} \phi_f \omega_r + \dot{\alpha}_{2d} - k_3 e_3 \right] \tag{4.28}$$

with $k_3 > 0$ being a design control gain, it is obtained that:

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 - k_3 e_3^2 \tag{4.29}$$

Step 4

At this step, we will construct the control law v_d. To obtain a similar operation to that of a DC machine separately excited, the *d*-axis current reference is fixed to zero, $i_{dref} = 0$. Define $e_4 = i_d$, differentiating e_4 obtains

$$\dot{e}_4 = -\frac{R}{L_d} i_d + p \frac{L_q}{L_d} \omega_r i_q + \frac{1}{L_d} v_d \tag{4.30}$$

Defining the following candidate Lyapunov function as $V_4 = V_3 + \frac{1}{2} e_4^2$. The time derivative is computed as:

$$\dot{V}_4 = -k_1 e_1^2 - k_2 e_2^2 - k_3 e_3^2 + e_4 \left[-\frac{R}{L_d} i_d + p \frac{L_q}{L_d} \omega_r i_q + \frac{1}{L_d} v_d \right] \tag{4.31}$$

We design control input v_{dref} as

$$v_{dref} = L_d \left[\frac{R}{L_d} i_d - p \frac{L_q}{L_d} \omega_r i_q - k_4 e_4 \right] \tag{4.32}$$

where $k_4 > 0$ is a positive design control gain, it is obtained that:

$$\dot{V}_4 = -k_1 e_1^2 - k_2 e_2^2 - k_3 e_3^2 - k_4 e_4^2 \tag{4.33}$$

Thus, we have $\dot{V}_4 < 0$ which concludes that the closed-loop controlled system is asymptotically stabilized according to Lyapunov's stability theorem. On the other hand, since the adaptation algorithm is employed to estimate the unknown load torque and friction effects, the control properties can reduce the influence of load torque variation.

By combining FFNN estimation and DSC technique, the designed controller has a simple structure, and the problems of estimating unknown load torque as well as friction terms and explosion of complexity are overcome. This will alleviate the

computational burden and render the designed scheme more efficient and suitable in practical applications.

To demonstrate the advantages of the DSC technique over the conventional backstepping control technique summarized in Appendix B, we compare in section 4.6 the dynamics of the subsystems described in equations (4.28) and (4.32) with those described in the corresponding equations (B.8) and (B.12), respectively. It can be seen that by the conventional backstepping control technique, the direct derivatives of α_1 and α_2 should appear in the control input v_q. As a result, the expression of the backstepping controller (5.8) would be much more complicated than that of the controller (4.28). The number of terms in the backstepping controller expression is much larger. This drawback is called the explosion of complexity mentioned above [31].

4.5 TUNING OF OBSERVER AND CONTROLLERS

Tuning of observer and controllers gains was done separately. Thus, the load torque LESO gains have been set at $c_0 = 900$ and $c_1 = 120$. On the other hand, the gains of the DSC and the conventional backstepping controller were defined as $k_1 = 4$, $k_2 = 400$, $k_3 = 400$, and $k_4 = 500$. The first-order filter time constants for the DSC technique are $\varepsilon_1 = \varepsilon_2 = 0.001\ s$. These gain values provided good reference speed tracking in both controllers. The gains of the controllers and the estimator are moderately high. As it is well-known that a high gain minimizes the effects of endogenous and exogenous disturbances.

4.6 SIMULATION RESULTS

In this section, a comparison between the proposed controller (4.28) and (4.32) and the classical backstepping controller (B.8) and (B.12) (Appendix B) is given for the IPMSM drive system with parameters mentioned in Table 5.1 (Appendix A).

4.6.1 Comparative Test Under the Nominal Conditions

In the first simulation, the motor runs at 104.72 rad/s, and a step load of 0.65 N.m is applied to the motor at t = 5 s and then removed at t = 10 s. Figures 4.5 and 4.6 show the response of each algorithm under nominal motor parameters.

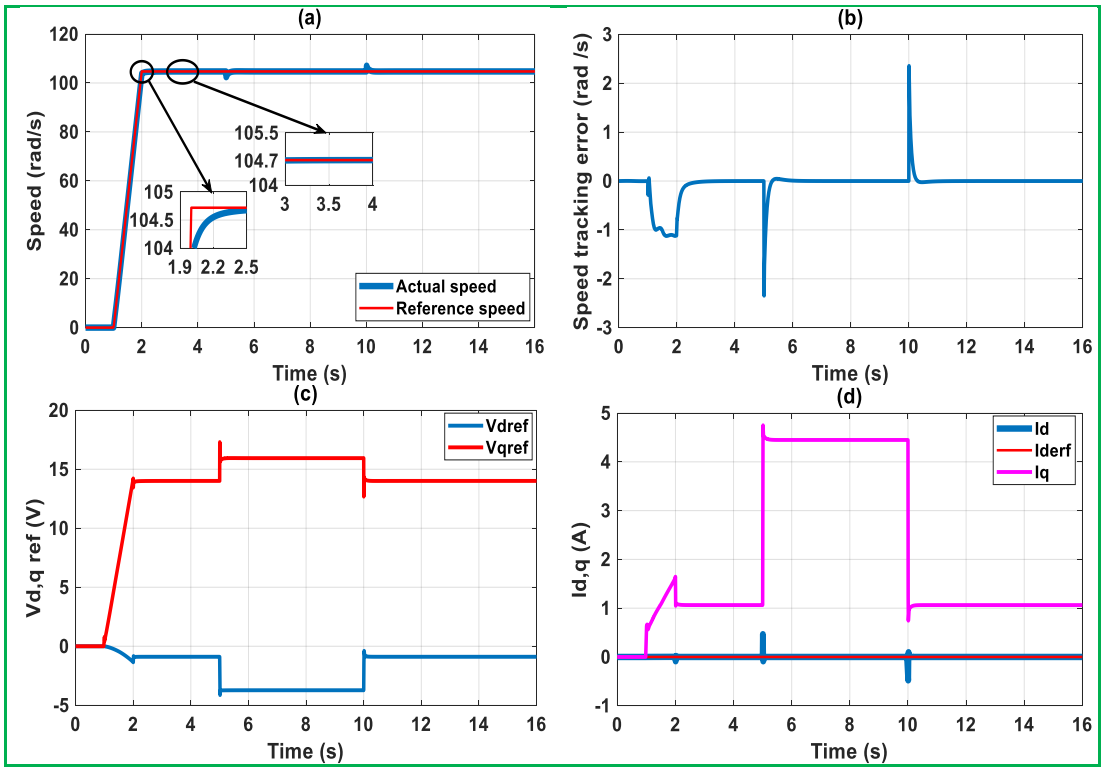


Figure 4. 5. DSC simulation results under nominal motor parameters: (a) motor speed (b) speed tracking error (c) *dq*-axis voltages (d) *dq*-axis currents.

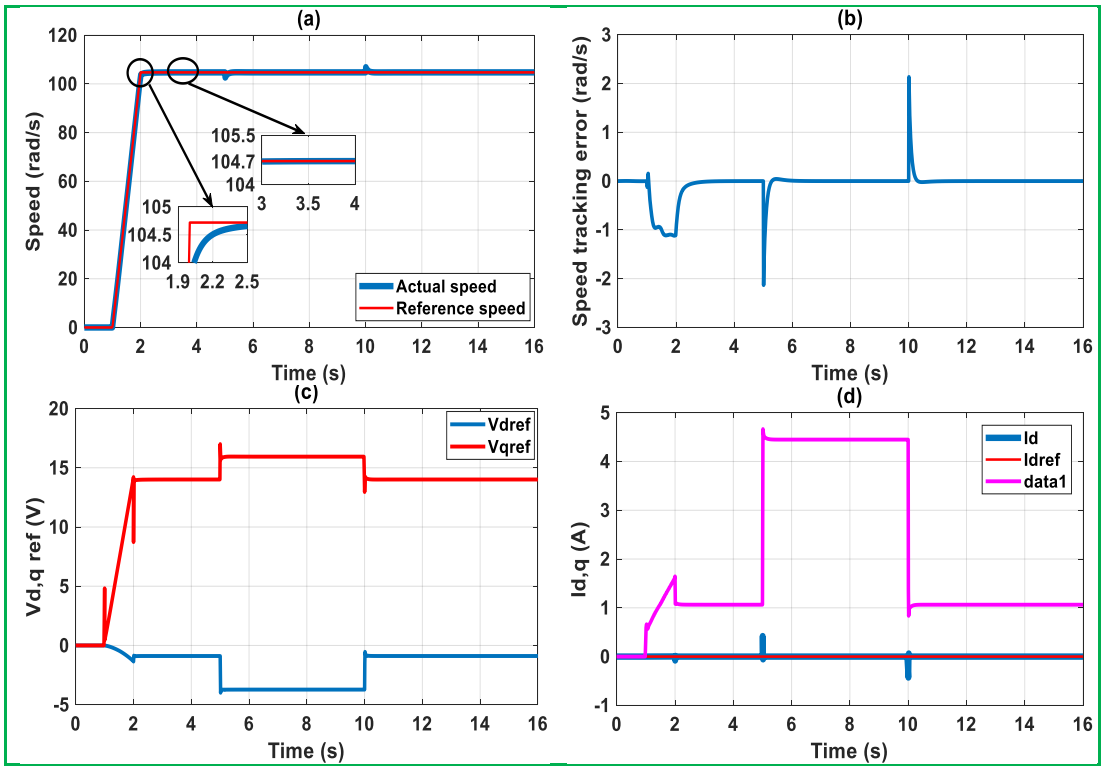


Figure 4. 6. Conventional backstepping controller simulation results under nominal motor parameters: (a) motor speed (b) speed tracking error (c) *dq*-axis voltages (d) *dq*-axis currents.

Figures 4.5(a)-4.5(b), and 4.6(a)-4.6(b) present the speed and the speed tracking error of both the proposed controller with the first-order filter and the traditional

backstepping controller, two sections of the speed figures are magnified to show the transient response and the steady-state response. Regarding the comparison of DSC with the backstepping controller, it can be seen that both controllers have almost similar tracking performance. The final control inputs v_{dref} and v_{qref} of the two controllers are given in Figures 4.5(c) and 4.6(c). We can note that the *d*-axis voltage v_{dref} stays near zero while the *q*-axis voltage v_{qref} responds to changes in the load torque. Figures 4.5(d) and 4.6(d) show the response of the *dq*-axis currents based on the DSC and the traditional backstepping control. It can be seen that they can immediately converge to the stable values and the *d*-axis current component is well decoupled and is regulated quite well to be zero, we can also see the existence of some sparks in the current curves when the load torque changes. From the simulations, it is clearly shown that the proposed DSC can trace the reference signals quite well in the case of nominal motor parameters, even though the controller has a much simpler structure than the classical one, which is more convenient to implement. A detailed comparative analysis of the performance of the two controllers is shown in Table 4.2.

Table 4. 2. Performance comparison of two control methods under nominal motor parameters.

	DSC	Backstepping Controller
Undershoot/Overshoot (%)	0	0
Settling Time (s)	0	0
Steady-State Error (%)	0	0
Dipping in Speed (rad/s)	2.33	2.14
Recovery Time (s)	1.5	1.5

Figure 4.7(a) illustrates the result of the torque estimation, which is estimated by the LESO and the FFNN observer. We found that the estimated torque of the IPMSM by the NN observer is very close to the estimated torque using the LESO, which has good dynamic tracking performance. Figure 4.7(b) shows the load torque estimation error for the performed simulation, calculated with the difference between the load torque estimated by the LESO and the FFNN observer. With the results exposed in this figure, it is possible to conclude that the proposed technique is suitable for torque estimation since the estimation error is contained in a relatively small range. In summary, the NN observer output value can well track the actual torque change of the motor and has good overall stability.

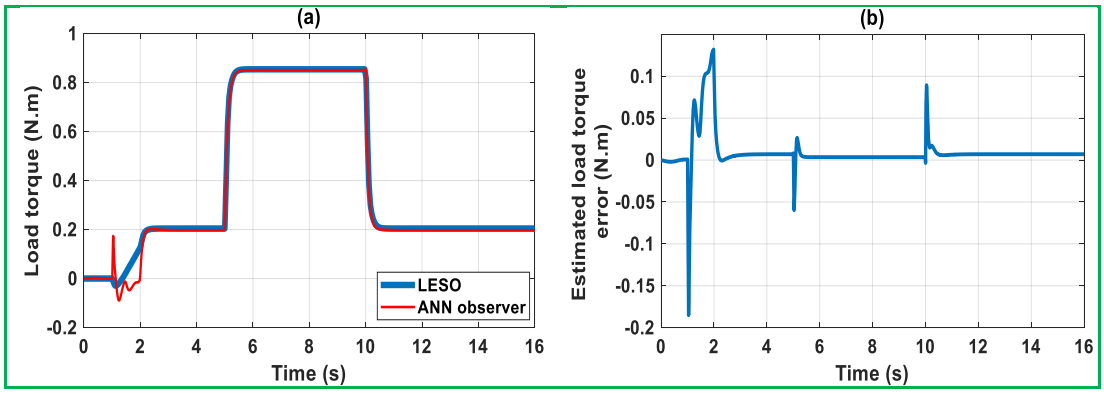


Figure 4. 7. Load torque estimation simulation results under nominal motor parameters: (a) load torque estimated by the LESO and the FFNN observer (b) load torque estimation error.

4.6.2 Comparative Test Under Parameters Variations

To illustrate the robustness of the proposed controller, the influence of the variations of all machine parameters is investigated. Variations of parameters are intentionally introduced at the same time into the controller's schemes: $\Delta R = +1R$, $\Delta L_d = +0.1L_d$, $\Delta L_q = -0.3L_q$, $\Delta \phi_f = -0.2\phi_f$, $\Delta J = +0.5J$, and $\Delta F = +0.5F$. The dynamic responses of the proposed controller involving the reference and actual rotor speeds and their zooms, the speed tracking error, the *dq*-axis voltages, and the *dq*-axis currents signals are depicted in Figure 4.8. Furthermore, the dynamic responses of the classical backstepping controller including the same signals are shown in Figure 4.9.

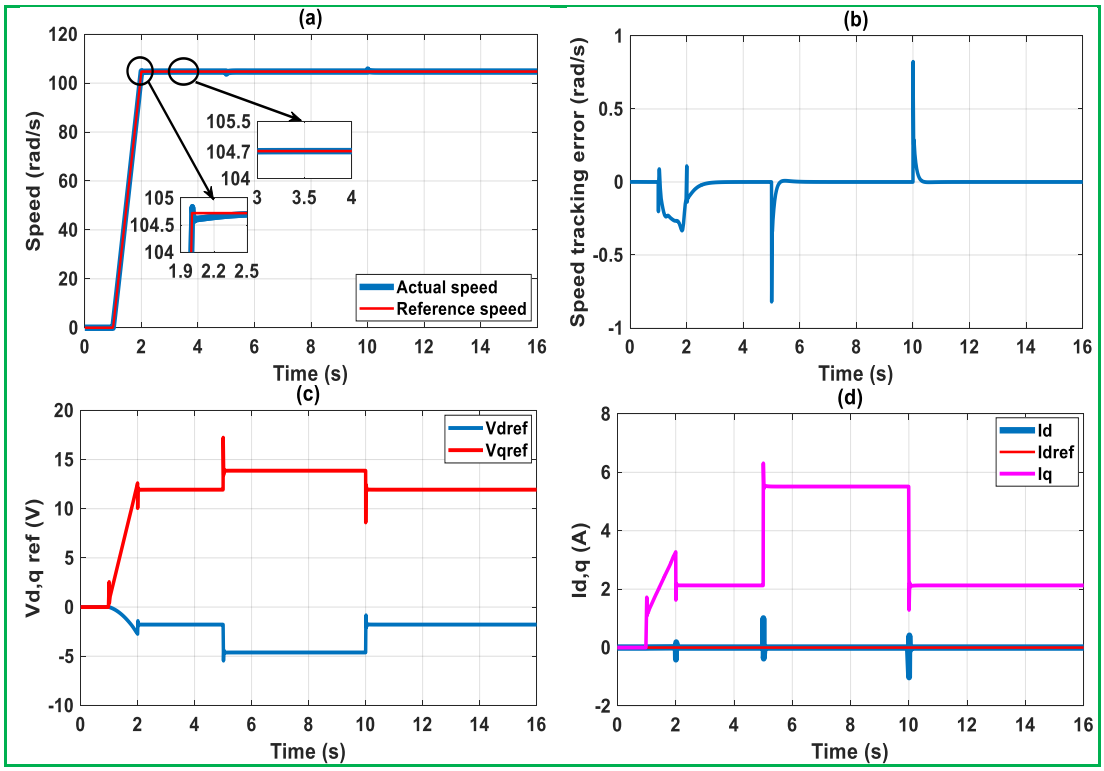


Figure 4. 8. DSC Simulation results under parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis voltages (d) *dq*-axis currents.

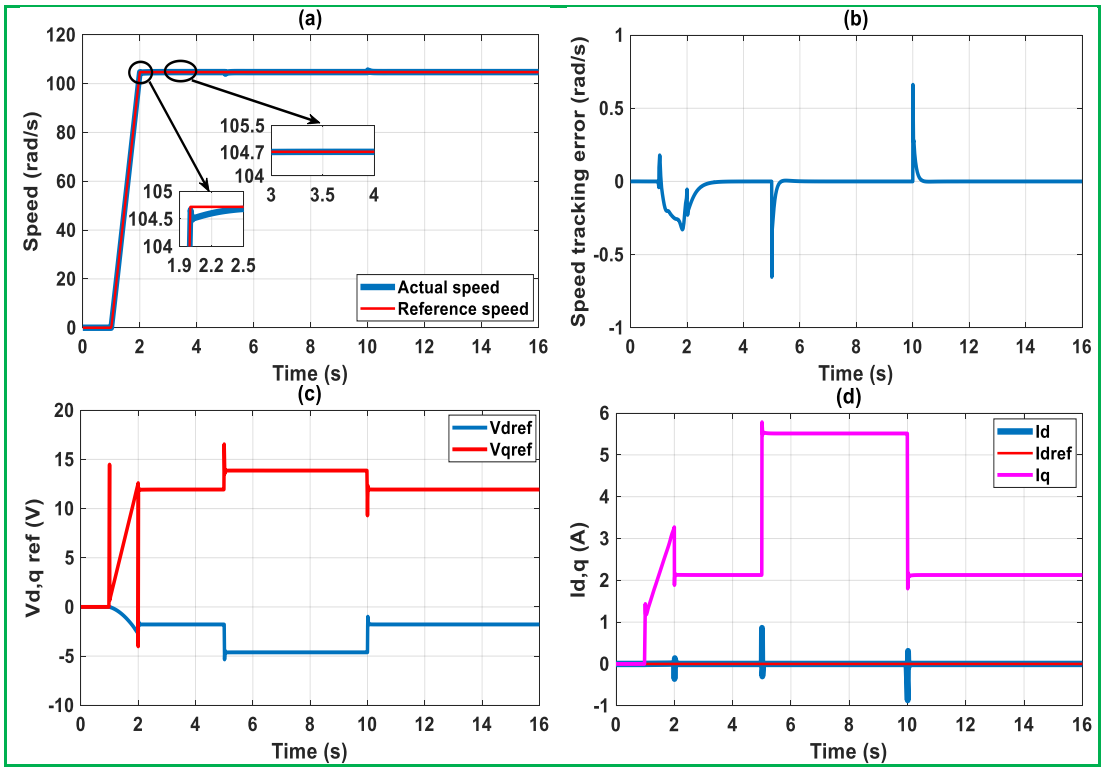


Figure 4. 9. Conventional backstepping controller simulation results under parameters variations: (a) motor speed (b) speed tracking error (c) *dq*-axis voltages (d) *dq*-axis currents.

Figures 4.8(a) and 4.9(a) show the speed response of each algorithm when the motor is running at 104.72 rad/s and a load of 0.65 N.m is applied at t = 5 s and then removed at t = 10 s. We observe that the overshoots of the two methods are 0.105 % and 0 %, the settling times, and the steady-state errors are 0. According to the simulation results, the proposed controller has an overshoot compared with the conventional controller. Also, when load torque is applied, the results show that the classical controller gives less dipping in speed tracking (0.65 rad/s) than the proposed controller (0.82 rad/s) and has a shorter recovery time (1.5 s) than the DSC technique (1.75 s) has, but the difference between them is quite small. It can be seen that the variations in the machine parameters affect the overshoot, and recovery time of the proposed controller. The final control inputs v_{dref} and v_{qref} of the two controllers are given in Figures 4.8(c) and 4.9(c). We can note that the *d*-axis voltage v_{dref} stays near zero while the *q*-axis voltage v_{qref} responds to changes in the load torque. Figures 4.8(d) and 4.9(d) shows the current waveforms of the *dq*-axis. We can observe a good decoupling of the *d*-axis current component introduced by the two controllers, we can also observe the existence of some sparks in the voltage and current curves when the load torque changes. Therefore, it was proved that the developed DSC is quite robust under the variations of all machine parameters and can achieve good control

performance with a simple structure than the backstepping controller in this case too. The comparative analysis of the control performance is illustrated in Table 4.3.

Table 4. 3. Performance comparison of two control methods under parameters variations.

	DSC	Backstepping Controller
Undershoot/Overshoot (%)	0.105	0
Settling Time (s)	0	0
Steady-State Error (%)	0	0
Dipping in Speed (rad/s)	0.82	0.65
Recovery Time (s)	1.75	1.5

In Figure 4.10, the robustness of the FFNN load torque observer to parameters variations is examined. Figure 4.10(a) illustrates the result of the torque estimation, which is estimated by the LESO and the FFNN observer. We found that the estimated torque of the IPMSM by the NN observer is close to the estimated torque using the LESO, which is somewhat affected by the variation of the machine parameters.

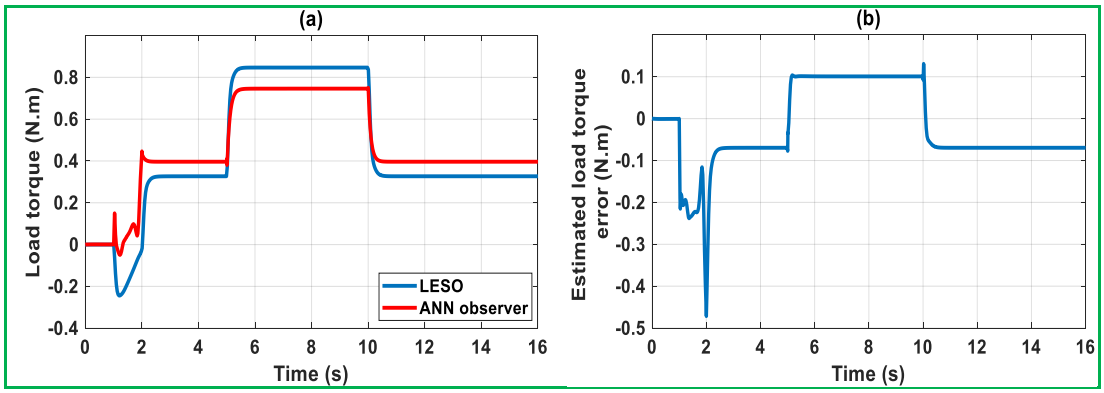


Figure 4. 10. Load torque estimation simulation results under parameters variations: (a) load torque estimated by the LESO and the FFNN observer (b) load torque estimation error.

Figure 4.10(b) shows the load torque estimation error for the performed simulation, calculated with the difference between the load torque estimated by the two observers. The figure shows that, despite the exaggerated error in the machine parameters, the load torque is still estimated with acceptable accuracy. The results of this experiment demonstrate that the proposed observer also works well under such conditions.

From the simulation results, it can be seen that high-accuracy speed tracking and robust characteristics are established using the proposed control scheme in all cases. Compared with the conventional backstepping controller, the DSC technique can achieve almost the same control performance for the IPMSM drive system with a simple structure that can be implemented using low-cost components. In addition, the NN observer offers the advantage of parameter independence and the non-necessity of

the mathematical model of the system. As such, it alleviates computational issues related to different observers like model-based approaches.

4.7 CONCLUSION

In this chapter, we have discussed the design and application of NN observer-based adaptive DSC for IPMSM, which together significantly enhance the tracking performance, and disturbance rejection performance of the system. The controller design is intuitively absorbing and uses a first-order low-pass filter to avoid model differentiation, therefore overcoming the problem of the explosion of complexity associated with the traditional backstepping approach. And then, based on Lyapunov asymptotical stability theory, we have proved that the presented control law can render the closed-loop controlled system asymptotically stable. This control law is robust under time-varying load torque, thanks to the overall uncertainty information gathered by a NN observer. The time-varying load torque is accurately estimated, thus allowing for an online adaptation to the DSC scheme. The high-level architecture of the proposed ANN for load torque estimation is illustrated at first. Then, the standard backpropagation algorithm used for the training of FFNN is introduced. Experimental data of the IPMSM drive is collected, and feature vectors for FFNN training collection are selected. With the motor data obtained, the developed NN observer is trained. The simulation results show that the developed observer, which is trained by backpropagation, can accurately estimate the load torque and its friction effects, and its performance is advantageous over other model-based approaches because it has nothing to do with the motor mathematical model and machine parameters. Since feasibility has been taken into account throughout the design process, the proposed control scheme is more applicable to practical use, as its implementation is simple and computationally less expensive, allowing the implementation of this control in low-cost hardware. The net result is an effective implementation of a robust nonlinear observer-based adaptive trajectory tracking scheme in the presence of unknown mechanical load including unmodeled linear viscous.

Chapter 5: Conclusions and Future Work

5.1 CONCLUSIONS

This thesis presented the design, simulation, and implementation of control algorithms for IPMSM drives which can be applied to a number of industrial applications. In chapter 1, the thesis motivation, the research objectives, and the thesis structure are given. In chapter 2, the PMSM classification is demonstrated, then some linear and nonlinear control techniques have been explained. Control issues and related technical challenges in terms of their performances and ease of implementation have also been explained and discussed. Mathematical machine model representation in stationary, as well as the synchronous reference frame, has been presented. During the derivation of the mathematical model of the IPMSM, Park's transformation was adopted to transform three-phase coordinates into a two-phase (d, q) reference frame. Vector control was used throughout the control design. Current regulators, as well as speed controller, were designed. The proper calculation of PI controller parameters ensured correct operation and high precision of the controller.

In chapter 3, a modified backstepping controller was presented for the IPMSM drive. The IBC is derived from the mathematical model in the synchronous reference frame (d, q) and is based on a backstepping control approach. The Lyapunov method is used to derive the backstepping controller. In addition, the stability characteristics of the IPMSM were investigated by Lyapunov's stability theory. The speed steady-state error and the sensitivity to load torque disturbance of the conventional backstepping control have been reduced by adding the integral of the tracking errors to the control low.

Since the PI controller is one of the most popular and widely used controllers, the performance of the IBC has been compared with that of the conventional PI controller to showcase their performance and robustness under different operating conditions. Extensive comparative simulation and experimental studies were presented to evaluate the performance of the IPMSM drive. First, a conventional PI controller was evaluated with the IBC in the case of nominal parameters. Comparative aspects

were speed response, speed tracking error, stator currents, and electromagnetic torque. Another comparative evaluation was with electrical and mechanical parameters uncertainties. We found that the conventional PI controller suffered from some issues such as slow response, larger overshoot, sensitivity to parameter uncertainties, and load torque disturbances. In contrast, the IBC exhibited super performance in simulated and experimental tests under all conditions.

In chapter 4, a precise robust adaptive dynamic surface controller is presented by combining DSC and NN load torque observer for IPMSM in the presence of uncertainties and unmodelled dynamics of the motor. Compared with the conventional backstepping approach, DSC is designed with a first-order filter to remove the explosion problem, and a NN load torque observer is designed to estimate the unknown time-varying load torque and its friction effects, which together significantly enhance the tracking performance and disturbance rejection performance of system output. The stability of the closed-loop control system is proved by the Lyapunov method, which verifies that the proposed controller of IPMSM has both a good transient and steady-state performance. Comparative simulation results demonstrate that the proposed controller can achieve a precise tracking performance and guarantee the disturbance rejection and robustness against uncertainties and load torque perturbation with a simpler structure.

A neural-based observer, whose architecture is patterned from the results of a model-based approach, was designed and applied to the load torque and its friction effects estimation task for the IPMSM. The FFNN was selected for the neural topology due to its simple structure, strong operability, good learning ability, fast convergence, and easy implementation in real-time systems. In addition, the NN approach has shown the ability to learn the IPMSM dynamics and robustness to the unavoidable drift and uncertainty of IPMSM parameters. The FFNN topology for load torque estimation is given at first. Then, the standard backpropagation algorithm used for training the NN is introduced. Real training data of the IPMSM drive is collected, and feature vectors for FFNN training collection are selected. With the motor data obtained, the developed multilayer network is trained. The ANN observer offers the upside of parameters independency and non-necessity of the system mathematical model. In that capacity, it lessens the computational issues connected with different observers like model-based approaches, such as load torque LESO. From the simulation results, it was

observed that the proposed observer is efficient, and satisfactory results similar to those presented by model-based approaches are obtained. Finally, it is noted that the implementation of the control scheme is computationally less expensive since it has a simple structure. This will allow the implementation of this control in low-cost hardware.

5.2 FUTURE WORK

5.2.1 Deep Neural Network Observer

The unknown load torque and its friction effects were estimated using two methods: the model-based approach and the ANN approach. Model-based approaches like the one presented in this thesis require knowledge of motor parameters, which in some situations are not readily available or difficult to obtain. ANNs, with their capacity for a wide range approximation of nonlinear functions, their fast optimization process, and their strong learning ability, offer the advantages of parameters independence and the non-necessity of the mathematical model. The proposed FFNN observer can estimate the load torque with an accepted accuracy and capture dynamic. Therefore, an improvement over the proposed estimation methods would be to use Deep Neural Network (DNN) observer.

DNN is an ANN with multiple layers between input and output layers. Deep networks process data in complex ways using sophisticated mathematical modeling that can model complex nonlinear relationships. While the universal approximation property holds for both DNNs and shallow networks, deep networks can approximate a class of functions as well as shallow networks, but with exponentially lower training parameters and sample complexity. Moreover, the number of neurons needed by a shallow network to approximate a function is exponentially larger than the corresponding number of neurons needed by a deep network for a given degree of function approximation. Furthermore, DNNs can approximate functions more efficiently than shallow neural networks [132], [133].

5.2.2 Field-Weakening

The suggested speed controllers can very precisely control the speed of the IPMSM below or at the rated speed. However, many applications require speeds higher than the rated speed; such as electric vehicles drive, compressor, fan, and servo. Therefore, an improvement over the proposed control methods would be extending the

design of the speed controllers to include speeds beyond the rated speed. Realization of such speeds requires the use of Field-Weakening.

As the speed goes high, the back-EMF increases. However, the line-to-line voltage is fixed at the DC-link voltage. In order to control the motor to run at high speed, more *d*-axis current needs to be injected to contradict the permanent magnet flux linkage. The flux linkage amplitude decreases and a higher speed can be reached [9]. Therefore, optimization techniques are used to create reference current signals for i_d and i_q to keep the air gap flux linkage constant above the rated speed.

5.2.3 Sensorless Control

The proposed control methods require position measurements. Installing mechanical position sensors increases the cost and decreases the reliability of IPMSM drive systems. Additionally, external sensors may be prohibited in some applications. Therefore, much attention has been paid to estimating the rotor position to achieve sensorless control, and many sensorless IPMSM methods have been proposed [134]. Sensorless control methods can be divided into two categories: signal injection-based methods, and back-EMF-based methods. The back-EMF-based control technique uses voltage and current measurements to estimate the back-EMF and then determines the position of the rotor. A drawback of such a technique is that it is unsuitable for low and zero speeds and is sensitive to parameter variations [30]. The high-frequency signal injection-based sensorless control technique uses high-frequency phase voltages to inject current into the machine. The current is then measured, and the position is estimated based on a high-frequency model of the IPMSM. This control method estimates position at a wide range of speeds, including standstill, which is an advantage over the back-EMF-based control method. Meanwhile, it suffers from acoustic noise generation, loss, secondary saliency, and cross-saturation problems [135]. Improvements over these control methods have taken place, such as using different state observers and combining the two techniques to estimate the rotor position. The topic of sensorless control is still under development. It is a promising control technique worth pursuing as an area of research in the future.

Appendices

Appendix A

Testbed Representation

The test testbed is composed of an IPMSM coupled to an interior permanent magnet synchronous generator (IPMSG) used as a load to apply an external step torque to the motor shaft and a three-phase inverter. The control algorithms are implemented with dSPACE 1104 card from Texas instruments, with a TMS320F240 DSP, and using the MATLAB-Simulink package. The control system is equipped with current and voltage sensors (LEMLA 55-P) and a resolver. Figure A.1 shows the testbed.

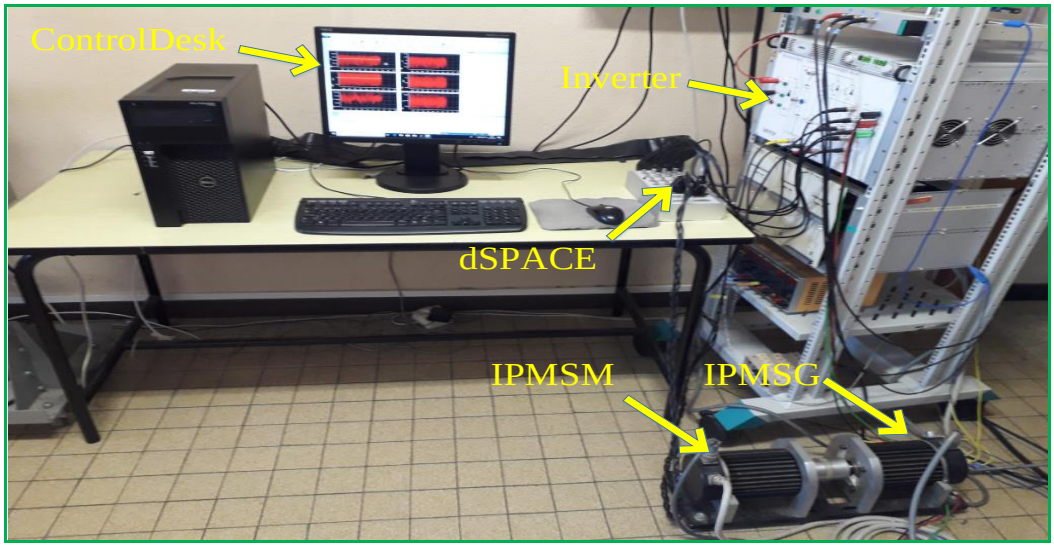


Figure A. 1. IPMSM control experimental testbed.

The nominal motor parameters are presented in Table 5.1.

Table A. 1. Parameters of the IPMSM.

Parameters	Value
Nominal power (kW)	1.1
Rated torque (N.m)	3
Rated speed, (rad/s)	314
Stator resistance, R (Ω)	0.57
d-axis inductance, L_d (H)	0.0045
q-axis inductance, L_q (H)	0.004
Magnet flux linkage, ϕ_f (Wb)	0.064
Pole pair, P	2
Rotor moment of inertia, J $(kg.m^2)$	0.00208
Viscous friction coefficient, F $(N.m/rd/s)$	0.0039

Appendix B

Conventional Backstepping Controller

This appendix is devoted to providing the designed controller by the classical backstepping approach.

Step 1

For the reference signal θ_{ref}, we define the tracking error variable as $e_1 = \theta_r - \theta_{ref}$. The error dynamic system is computed by $\dot{e}_1 = \omega_r - \omega_{ref}$.

Choose the Lyapunov function as $V_1 = \frac{1}{2}e_1^2$, then its time derivative is given by

$$\dot{V}_1 = e_1\dot{e}_1 = e_1\left(\omega_r - \omega_{ref}\right) \tag{B.1}$$

Construct the virtual control law α_ω as

$$\alpha_\omega = -k_1 e_1 + \omega_{ref} \tag{B.2}$$

with $k_1 > 0$ being a positive design control gain.

Define $e_2 = \omega_r - \alpha_\omega$

Step 2

Differentiating e_2 obtains

$$\dot{e}_2 = \frac{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}{J} i_q - \frac{1}{J}\tau_L - \dot{\alpha}_\omega$$

$$= \frac{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}{J} i_q - \frac{1}{J}\tau_L + k_1\dot{e}_1 \tag{B.3}$$

Now, choose the Lyapunov function candidate as $V_2 = V_1 + \frac{1}{2}e_2^2$. The time derivative of V_2 can be expressed as

$$\dot{V}_2 = -k_1 e_1^2 + e_2\left[\frac{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}{J} i_q - \frac{1}{J}\tau_L + k_1\dot{e}_1\right] \tag{B.4}$$

Then the virtual control i_{qref} is constructed as

$$i_{qref} = \frac{J}{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}\left[\frac{1}{J}\tau_L - k_1\dot{e}_1 - k_2 e_2\right] \tag{B.5}$$

with $k_2 > 0$ being a positive design control gain.

and define $e_3 = i_q - i_{qref}$

Step 3

Differentiating e_3 results in the following equation:

$$\dot{e}_3 = \dot{i}_q - \dot{i}_{qref}$$

$$= -\frac{R}{L_q}i_q - p\frac{L_d}{L_q}\omega_r i_d - p\frac{1}{L_q}\phi_f\omega_r + \frac{1}{L_q}v_q - \dot{i}_{qref}$$

$$= -\frac{R}{L_q}i_q - p\frac{L_d}{L_q}\omega_r i_d - p\frac{1}{L_q}\phi_f\omega_r + \frac{1}{L_q}v_q$$

$$-\frac{J}{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}\left[\frac{1}{J}\dot{\tau}_L - k_1\ddot{e}_1 - k_2\dot{e}_2\right]$$

$$= -\frac{R}{L_q}i_q - p\frac{L_d}{L_q}\omega_r i_d - p\frac{1}{L_q}\phi_f\omega_r + \frac{1}{L_q}v_q$$

$$-\frac{J}{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}\left[\frac{1}{J}\dot{\tau}_L - k_1\left[\frac{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}{J}i_q - \frac{1}{J}\tau_L\right]\right.$$

$$\left. -k_2\left[\frac{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}{J}i_q - \frac{1}{J}\tau_L + k_1\left(\omega_r - \omega_{ref}\right)\right]\right] \tag{B.6}$$

Choose the following Lyapunov function candidate as $V_3 = V_2 + \frac{1}{2}e_3^2$. Furthermore, differentiating V_3 yields

$$\dot{V}_3 = -k_1 e_1^2 - k_2 e_2^2 + e_3\left[-\frac{R}{L_q}i_q - p\frac{L_d}{L_q}\omega_r i_d - p\frac{1}{L_q}\phi_f\omega_r + \frac{1}{L_q}v_q\right.$$

$$-\frac{J}{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}\left[\frac{1}{J}\dot{\tau}_L - k_1\left[\frac{P\left[\left(L_d - L_q\right)i_d + \phi_f\right]}{J}i_q - \frac{1}{J}\tau_L\right]\right.$$

$$-k_2\left[\frac{P\left[\left(L_d-L_q\right)i_d+\phi_f\right]}{J}i_q-\frac{1}{J}\tau_L+k_1\left(\omega_r-\omega_{ref}\right)\right]\Bigg]\Bigg] \tag{B.7}$$

Now, the control input v_{qref} is designed as:

$$v_{qref}=L_q\Bigg[\frac{R}{L_q}i_q+p\frac{L_d}{L_q}\omega_r i_d+p\frac{1}{L_q}\phi_f\omega_r$$

$$+\frac{J}{P\left[\left(L_d-L_q\right)i_d+\phi_f\right]}\Bigg[\frac{1}{J}\dot{\tau}_L-k_1\Bigg[\frac{P\left[\left(L_d-L_q\right)i_d+\phi_f\right]}{J}i_q-\frac{1}{J}\tau_L\Bigg]$$

$$-k_2\Bigg[\frac{P\left[\left(L_d-L_q\right)i_d+\phi_f\right]}{J}i_q-\frac{1}{J}\tau_L+k_1\left(\omega_r-\omega_{ref}\right)\Bigg]\Bigg]-k_3e_3\Bigg] \tag{B.8}$$

with $k_3>0$ being a design control gain, it is obtained that:

$$\dot{V}_3=-k_1e_1^2-k_2e_2^2-k_3e_3^2 \tag{B.9}$$

Step 4

At this step, we will construct the control law v_d. To obtain a similar operation to that of a DC machine separately excited, the *d*-axis current reference is fixed to zero, $i_{dref}=0$. Define $e_4=i_d$, differentiating e_4 obtains

$$\dot{e}_4=-\frac{R}{L_d}i_d+p\frac{L_q}{L_d}\omega_r i_q+\frac{1}{L_d}v_d \tag{B.10}$$

Defining the following candidate Lyapunov function as $V_4=V_3+\frac{1}{2}e_4^2$. The time derivative is computed as:

$$\dot{V}_4=-k_1e_1^2-k_2e_2^2-k_3e_3^2+e_4\Bigg[-\frac{R}{L_d}i_d+p\frac{L_q}{L_d}\omega_r i_q+\frac{1}{L_d}v_d\Bigg] \tag{B.11}$$

We design control input v_{dref} as

$$v_{dref}=L_d\Bigg[\frac{R}{L_d}i_d-p\frac{L_q}{L_d}\omega_r i_q-k_4e_4\Bigg] \tag{B.12}$$

where $k_4>0$ is a positive design control gain, it is obtained that:

$$\dot{V}_4 = -k_1 e_1^2 - k_2 e_2^2 - k_3 e_3^2 - k_4 e_4^2 \tag{B.13}$$

Thus, we have $\dot{V}_4 < 0$ which concludes that the closed-loop controlled system is asymptotically stabilized according to Lyapunov's stability theorem. As a consequence, all the tracking errors e_1, e_2, e_3, and e_4 will converge to zero asymptotically.

Appendix C

Space Vector for a Two-Level Inverter

This appendix is devoted to the theory of Space Vector Pulse Width Modulation (SVPWM). There is no single PWM method that is the best suited for all applications, and with advances in solid-state power electronic devices and microprocessors. Various pulse-width modulation (PWM) techniques have been developed for industrial applications. The most widely used PWM schemes for three-phase voltage source inverters are carrier based sinusoidal PWM and space vector PWM (SVPWM) [136]. The output voltage per phase for a sinusoidal PWM based three phase converter is limited to 0.5Vdc (peak value) and the line-to-line RMS voltage is 0.612Vdc. SVM is another direct digital PWM technique proposed in 1982. It has become a basic power processing technique in three-phase converters. SVM based converter can have a higher output voltage output at 0.707Vdc (line-to-line, RMS). The classic SVM strategy, first proposed by Holtz and Van der Broeck. For vector SVM, the vector represents the three sinusoidal output voltages that one desires. This vector is best approximated during each modulation interval by acting on the control of the three complementary switch sets. This vector PWM is not based on separate calculations for each arm of the inverter but on the determination of an approximated global control vector over a modulation period T_e. The circuit diagram of the considered model of the three-phase inverter is shown in Figure C.1. The power stage consists of six switches (that is to say S_1, S_2, …, S_6) and a three-phase load correspondingly associated with a three-phase voltage $\{V_{an}, V_{bn}, V_{cn}\}$. V_α and V_β, which are used to control the three-phase voltage of the inverter [137, 138].

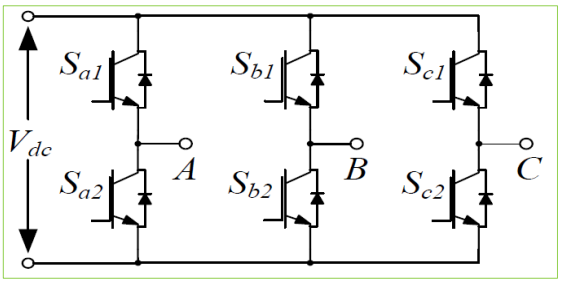


Figure C.1. Circuit diagram of the SVPWM three-phase inverter.

C.1. Principle of vector modeling SVM

The principle of vector modeling (SVM) consists in reconstructing the voltage vector V_a a from eight voltage vectors. Each of these vectors corresponds to a combination of the state of the switches of a three-phase voltage inverter, it does not rely on separate calculations modulations by each arm of the inverter [139]. This technique follows the following principles:

- V_a a reference vector is calculated globally and approximated over a modulation period T_s.
- All half-bridge switches have a state identical to the centers and ends of the period.

A combinatorial analysis of all the possible states of the switches makes it possible to calculate the voltage vector (V_α, V_β). So, we draw a Table C.1, different states of the inverter.

Table C.1. Possible voltage at the output of the inverter.

VECTOR	S_1	S_2	S_3	V_{an}	V_{bn}	V_{cn}	V_α	V_β
$\overrightarrow{V_0}$	0	0	0	0	0	0	0	0
$\overrightarrow{V_1}$	1	0	0	$2V_{dc}/3$	$-V_{dc}/3$	$-V_{dc}/3$	$2V_{dc}/3$	0
$\overrightarrow{V_3}$	0	1	0	$-V_{dc}/3$	$2V_{dc}/3$	$-V_{dc}/3$	$-V_{dc}/3$	$V_{dc}/\sqrt{3}$
$\overrightarrow{V_2}$	1	1	0	$V_{dc}/3$	$V_{dc}/3$	$-2V_{dc}/3$	$V_{dc}/3$	$V_{dc}/\sqrt{3}$
$\overrightarrow{V_5}$	0	0	1	$-V_{dc}/3$	$-V_{dc}/3$	$2V_{dc}/3$	$-V_{dc}/3$	$-V_{dc}/\sqrt{3}$
$\overrightarrow{V_6}$	1	0	1	$V_{dc}/3$	$-2V_{dc}/3$	$V_{dc}/3$	$V_{dc}/3$	$-V_{dc}/\sqrt{3}$
$\overrightarrow{V_4}$	0	1	1	$-2V_{dc}/3$	$V_{dc}/3$	$V_{dc}/3$	$-2V_{dc}/3$	0
$\overrightarrow{V_7}$	1	1	1	0	0	0	0	0

The vector V_α is approximated over the modulation period, by the generation of an average vector developed by the application of the available vectors. It consists in considering globally the three-phase system, and in applying to it a Concordia transform to be brought back into the plane (V_α, V_β). The three-phase system of voltages to be generated for the current sampling time can then be represented as a single vector in this plane. In this model, if the ideal three-phase voltage we have:

$$\begin{cases} V_{aref} = V_m \sin(2\pi ft) \\ V_{bref} = V_m \sin\left(2\pi ft - \frac{2\pi}{3}\right) \\ V_{cref} = V_m \sin\left(2\pi ft + \frac{2\pi}{3}\right) \end{cases} \quad \text{(C.1)}$$

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad \text{(C.2)}$$

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = C \begin{bmatrix} V_{aref} \\ V_{bref} \\ V_{cref} \end{bmatrix} \quad \text{(C.3)}$$

with C the transformation matrix developed by Concordia is given by the expression:

$$C = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad \text{(C.4)}$$

This control technique divides the α β resistance reference into six regions, as shown in Figure C.2. The voltage hexagon of the Figure 5.3 is the vector representation of the different combinations of the 3 magnitudes (S_a, S_b, S_c): non-zero vectors V_1 to V_6 (V_0 and V_7 being the null vectors) V_0 (S_a, S_b, S_c) = (0, 0, 0), V_7 (S_a, S_b, S_c) = (1, 1, 1).

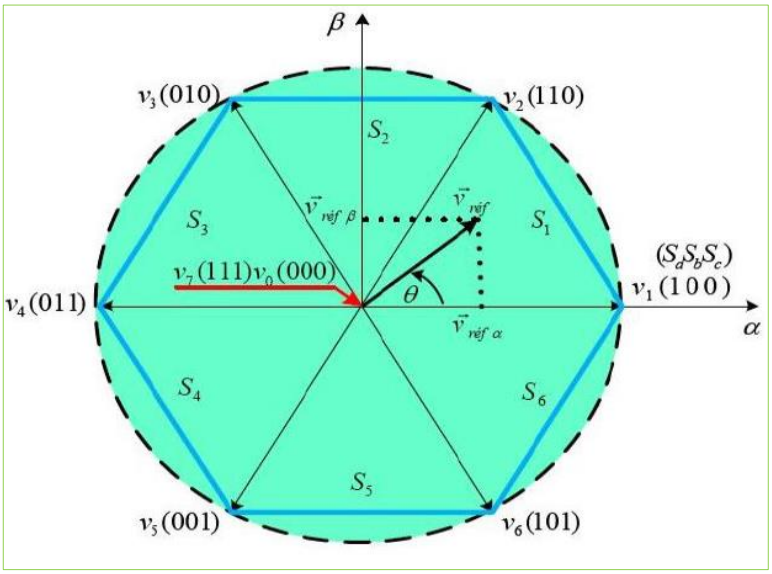


Figure C.2. Hexagon of tension defined in the plane α-β.

C.2. Detection of sectors by SVM algorithm

The SVM algorithm makes it possible to locate a vector represented in the plane α-β based on these two data: Algebraic sign of the components α and β of the vector; Amplitude of the β component with respect to the amplitude of the component see Figure C.3 [140].

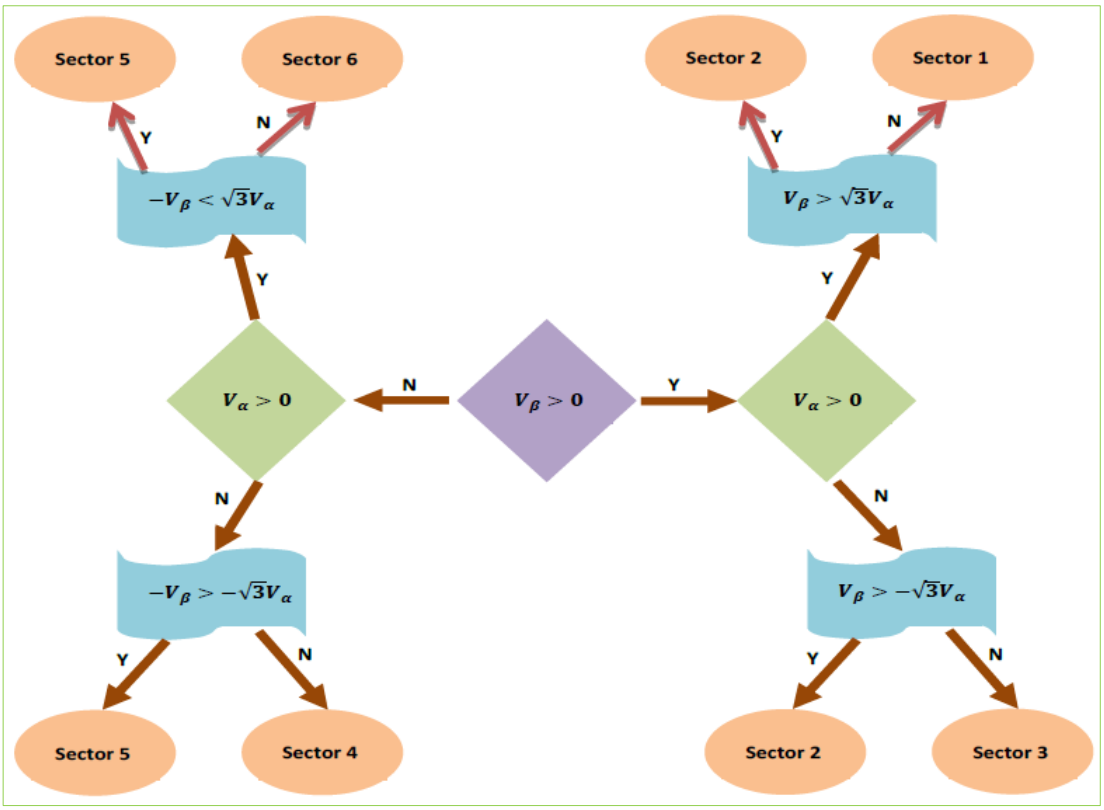


Figure C.3. Detection of sectors by the SVM algorithm.

C.3. Generation of PWM

A reference vector V_{ref} is considered rotating in the concentric circle of the voltage hexagon, as shown in Figure C.4(a) [141].

with:

$$V_s = \begin{cases} V_{sk} + V_{sk+1} \\ \frac{T_k}{T_s} V_k + \frac{T_{k+1}}{T_s} V_{k+1} \end{cases} \tag{C.5}$$

$$T_0 = \frac{(T_s - T_1 - T_2)}{4} \tag{C.6}$$

Case of zone 1: This situation is shown in Figure C.4(b).

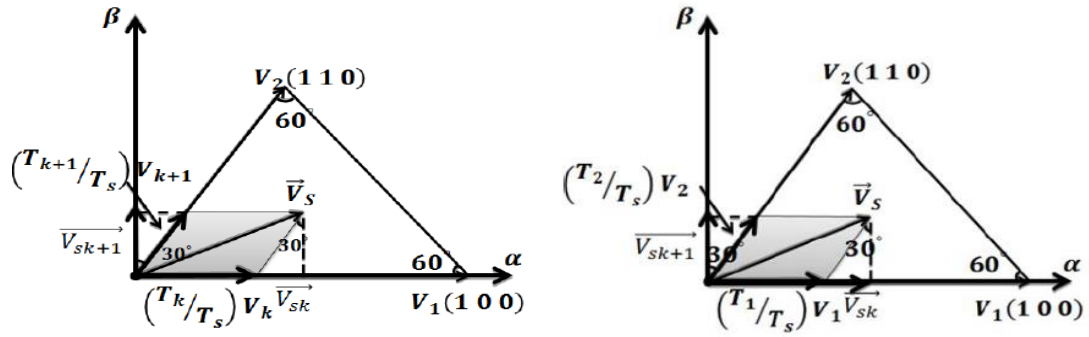


Figure C.4. (a) Rotation of the reference vector in the hexagon, (b) case of zone 1.

In this case:

$$V_s = \begin{cases} V_{sk} + V_{sk+1} \\ \dfrac{T_1}{T_s} V_1 + \dfrac{T_2}{T_s} V_2 \end{cases} \tag{C.7}$$

$$\begin{cases} V_{sk} = \dfrac{T_1}{T_s} V_1 + \dfrac{T_2}{T_s} V_2 \sin\left(30^0\right) \\ V_{sk+1} = \dfrac{T_2}{T_s} V_2 \cos\left(30^0\right) \end{cases} \tag{C.8}$$

$$V_2 \text{ and } V_1 = \sqrt{\frac{2}{3} V_{dc}} \tag{C.9}$$

Gets:

$$\begin{cases} T_1 = T_s \dfrac{\sqrt{6} V_{sk} - \sqrt{2} V_{sk+1}}{2 V_{dc}} \\ T_1 = T_s \sqrt{2} \dfrac{V_{sk+1}}{V_{dc}} \end{cases} \tag{C.10}$$

The pulse during a sampling period T_s is presented in the timing diagram of Figure C.5 (case of the three pulses of the three upper switches).

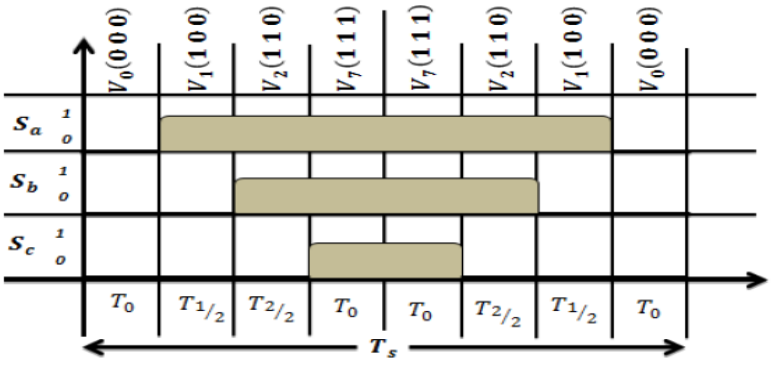


Figure C.5. Impulse of zone 1.

Bibliography

[1] Q. N. Khanh, "Development of FPGA Based Control Architecture for PMSM Drives," Citeseer, 2015.

[2] J. Linares-Flores, C. García-Rodríguez, H. Sira-Ramírez, and O. D. Ramírez-Cárdenas, "Robust backstepping tracking controller for low-speed PMSM positioning system: design, analysis, and implementation," *IEEE Transactions on industrial informatics,* vol. 11, pp. 1130-1141, 2015.

[3] D. Świerczyński, "Direct torque control with space vector modulation (DTC-SVM) of inverter-fed permanent magnet synchronous motor drive," The Institute of Control and Industrial Electronics, 2005.

[4] H. Yang, "Performance improvement of direct torque controlled permanent magnet synchronous motor drives."

[5] F. Wang, "Model predictive torque control for electrical drive systems with and without an encoder," Technische Universität München, 2014.

[6] N. Okafor, "Analysis and control of nonlinear phenomena in electrical drives," Newcastle University, 2013.

[7] P. Kumar, "Permanent magnet synchronous motor control for efficient motor drives," -, 2020.

[8] J. Wai and T. M. Jahns, "A new control technique for achieving wide constant power speed operation with an interior PM alternator machine," in *Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No. 01CH37248)*, 2001, pp. 807-814.

[9] Y. MIAO, "High-accuracy torque control and estimation for interior permanent magnet synchronous machine drives with loss minimization," 2018.

[10] S. Morimoto, K. Kawamoto, M. Sanada, and Y. Takeda, "Sensorless control strategy for salient-pole PMSM based on extended EMF in rotating reference frame," *IEEE transactions on industry applications,* vol. 38, pp. 1054-1061, 2002.

[11] Y. Sun, M. Preindl, S. Sirouspour, and A. Emadi, "Unified wide-speed sensorless scheme using nonlinear optimization for IPMSM drives," *IEEE Transactions on Power Electronics,* vol. 32, pp. 6308-6322, 2016.

[12] C. B. Butt, M. A. Hoque, and M. A. Rahman, "Simplified fuzzy-logic-based MTPA speed control of IPMSM drive," *IEEE Transactions on industry applications,* vol. 40, pp. 1529-1535, 2004.

[13] C. Mademlis and V. G. Agelidis, "On considering magnetic saturation with maximum torque to current control in interior permanent magnet synchronous motor drives," *IEEE transactions on energy conversion,* vol. 16, pp. 246-252, 2001.

[14] M. Karabacak and H. I. Eskikurt, "Design, modelling and simulation of a new nonlinear and full adaptive backstepping speed tracking controller for uncertain PMSM," *Applied Mathematical Modelling,* vol. 36, pp. 5199-5213, 2012.

[15] J. Yu, Y. Ma, B. Chen, and H. Yu, "Adaptive fuzzy backstepping position tracking control for a permanent magnet synchronous motor," *International Journal of Innovative Computing, Information and Control,* vol. 7, pp. 1589-1601, 2011.

[16] Y. X. Su, C. H. Zheng, and B. Y. Duan, "Automatic disturbances rejection controller for precise motion control of permanent-magnet synchronous motors," *IEEE Transactions on Industrial Electronics,* vol. 52, pp. 814-823, 2005.
[17] K.-H. Kim and M.-J. Youn, "A nonlinear speed control for a PM synchronous motor using a simple disturbance estimation technique," *IEEE Transactions on Industrial Electronics,* vol. 49, pp. 524-535, 2002.
[18] B. GrCar, P. Cafuta, M. Znidaric, and F. Gausch, "Nonlinear control of synchronous servo drive," *IEEE transactions on control systems technology,* vol. 4, pp. 177-184, 1996.
[19] G.-J. Wang, C.-T. Fong, and K. J. Chang, "Neural-network-based self-tuning PI controller for precise motion control of PMAC motors," *IEEE Transactions on Industrial Electronics,* vol. 48, pp. 408-415, 2001.
[20] Y. A.-R. I. Mohamed, "Design and implementation of a robust current-control scheme for a PMSM vector drive with a simple adaptive disturbance observer," *IEEE transactions on industrial electronics,* vol. 54, pp. 1981-1988, 2007.
[21] H. Jin and J. Lee, "An RMRAC current regulator for permanent-magnet synchronous motor based on statistical model interpretation," *IEEE Transactions on Industrial Electronics,* vol. 56, pp. 169-177, 2008.
[22] Y. A.-R. I. Mohamed and E. F. El-Saadany, "A current control scheme with an adaptive internal model for torque ripple minimization and robust current regulation in PMSM drive systems," *IEEE Transactions on Energy Conversion,* vol. 23, pp. 92-100, 2008.
[23] T.-L. Hsien, Y.-Y. Sun, and M.-C. Tsai, "H∞ control for a sensorless permanent-magnet synchronous drive," *IEE Proceedings-Electric Power Applications,* vol. 144, pp. 173-181, 1997.
[24] I.-C. Baik, K.-H. Kim, and M.-J. Youn, "Robust nonlinear speed control of PM synchronous motor using boundary layer integral sliding mode control technique," *IEEE Transactions on Control Systems Technology,* vol. 8, pp. 47-54, 2000.
[25] R.-J. Wai, "Total sliding-mode controller for PM synchronous servo motor drive using recurrent fuzzy neural network," *IEEE Transactions on Industrial Electronics,* vol. 48, pp. 926-944, 2001.
[26] M. A. Rahman, D. M. Vilathgamuwa, M. N. Uddin, and K.-J. Tseng, "Nonlinear control of interior permanent-magnet synchronous motor," *IEEE Transactions on Industry Applications,* vol. 39, pp. 408-416, 2003.
[27] J. Zhou and Y. Wang, "Adaptive backstepping speed controller design for a permanent magnet synchronous motor," *IEE Proceedings-Electric Power Applications,* vol. 149, pp. 165-172, 2002.
[28] Y.-S. Kung and M.-H. Tsai, "FPGA-based speed control IC for PMSM drive with adaptive fuzzy control," *IEEE transactions on power electronics,* vol. 22, pp. 2476-2486, 2007.
[29] S. Li and Z. Liu, "Adaptive speed control for permanent-magnet synchronous motor system with variations of load inertia," *IEEE transactions on industrial electronics,* vol. 56, pp. 3050-3059, 2009.
[30] M. A. Hamida, J. De Leon, and A. Glumineau, "High-order sliding mode observers and integral backstepping sensorless control of IPMS motor," *International Journal of Control,* vol. 87, pp. 2176-2193, 2014.
[31] J. Yu, P. Shi, W. Dong, B. Chen, and C. Lin, "Neural network-based adaptive dynamic surface control for permanent magnet synchronous motors," *IEEE*

transactions on neural networks and learning systems, vol. 26, pp. 640-645, 2014.

[32] R. George and A. S. Mathew, "Speed Control of PMSM using Backstepping Method," *International Journal of Engineering Research & Technology (IJERT),* vol. 4, pp. 609-612, 2015.

[33] X. Wang, Y. Chen, Y. Lu, X. Li, and W. He, "Dynamic surface method–based adaptive backstepping control for the permanent magnet synchronous motor on parameter identification," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 233, pp. 1172-1181, 2019.

[34] F. Mehazzem, A. Nemmour, A. Reama, and H. Benalla, "Nonlinear integral backstepping control for induction motors," in *International Aegean Conference on Electrical Machines and Power Electronics and Electromotion, Joint Conference*, 2011, pp. 331-336.

[35] H. Li, X. Li, Z. Chen, J. Mao, and J. Huang, "Model-Free Adaptive Integral Backstepping Control for PMSM Drive Systems," *Journal of Power Electronics*, vol. 19, pp. 1193-1202, 2019.

[36] J. Zhang, S. Wang, P. Zhou, L. Zhao, and S. Li, "Novel prescribed performance-tangent barrier Lyapunov function for neural adaptive control of the chaotic PMSM system by backstepping," *International Journal of Electrical Power & Energy Systems,* vol. 121, p. 105991, 2020.

[37] G. Li, W. Xu, J. Zhao, S. Wang, and B. Li, "Precise robust adaptive dynamic surface control of permanent magnet synchronous motor based on extended state observer," *IET Science, Measurement & Technology,* vol. 11, pp. 590-599, 2017.

[38] X. S. Luo, B. H. Wang, and J. Q. Fang, "Robust adaptive dynamic surface control of chaos in permanent magnet synchronous motor," *Physics Letters A,* vol. 363, pp. 71-77, 2007.

[39] F.-J. Lin, Y.-C. Hung, J.-M. Chen, and C.-M. Yeh, "Sensorless IPMSM drive system using saliency back-EMF-based intelligent torque observer with MTPA control," *IEEE Transactions on Industrial Informatics,* vol. 10, pp. 1226-1241, 2014.

[40] Z. Zhang, *Permanent Magnet Synchronous Machine based traction drive design for hybrid scooter considering control nonlinearities and compenstions*: The Ohio State University, 2013.

[41] A. Glumineau and J. de León Morales, "Sensorless AC electric motor control," *Cham: Springer International Publishing,* 2015.

[42] B. K. Bose, *Modern power electronics and AC drives* vol. 123: Prentice hall Upper Saddle River, NJ, 2002.

[43] S. J. Underwood, "On-line parameter estimation and adaptive control of permanent magnet synchronous machines," Ph.D. thesis, University of Akron, 2006.

[44] T. M. Jahns, "Motion control with permanent-magnet AC machines," *Proceedings of the IEEE,* vol. 82, pp. 1241-1252, 1994.

[45] R. Krishnan, *Permanent magnet synchronous and brushless DC motor drives*: CRC press, 2017.

[46] K. Prashanth and H. G. Navada, "Parameter estimation of PMSM using adaptive backstepping technique," in *2014 International Conference on Advances in Energy Conversion Technologies (ICAECT)*, 2014, pp. 1-6.

[47] Q. Song and C. Jia, "Robust speed controller design for permanent magnet synchronous motor drives based on sliding mode control," *Energy Procedia,* vol. 88, pp. 867-873, 2016.
[48] J.-W. Jung, Y.-S. Choi, V. Leu, and H. Choi, "Fuzzy PI-type current controllers for permanent magnet synchronous motors," *IET electric power applications,* vol. 5, pp. 143-152, 2011.
[49] S. Barkat, A. Tlemçani, and H. Nouri, "Noninteracting adaptive control of PMSM using interval type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems,* vol. 19, pp. 925-936, 2011.
[50] Y. Yi, D. M. Vilathgamuwa, and M. A. Rahman, "Implementation of an artificial-neural-network-based real-time adaptive controller for an interior permanent-magnet motor drive," *IEEE transactions on industry applications,* vol. 39, pp. 96-104, 2003.
[51] L. Sheng, G. Xiaojie, and Z. Lanyong, "Robust adaptive backstepping sliding mode control for six-phase permanent magnet synchronous motor using recurrent wavelet fuzzy neural network," *IEEE Access,* vol. 5, pp. 14502-14515, 2017.
[52] F. J. Lin, K. J. Yang, I. F. Sun, and J. K. Chang, "Intelligent position control of permanent magnet synchronous motor using recurrent fuzzy neural cerebellar model articulation network," *IET Electric Power Applications,* vol. 9, pp. 248-264, 2015.
[53] F.-J. Lin, I.-F. Sun, K.-J. Yang, and J.-K. Chang, "Recurrent fuzzy neural cerebellar model articulation network fault-tolerant control of six-phase permanent magnet synchronous motor position servo drive," *IEEE Transactions on Fuzzy Systems,* vol. 24, pp. 153-167, 2015.
[54] D. Su, C. Zhang, and Y. Dong, "An improved continuous-time model predictive control of permanent magnetic synchronous motors for a wide-speed range," *Energies,* vol. 10, p. 2051, 2017.
[55] M. Mohseni, S. M. Islam, and M. A. Masoum, "Impacts of symmetrical and asymmetrical voltage sags on DFIG-based wind turbines considering phase-angle jump, voltage recovery, and sag parameters," *IEEE Transactions on power electronics,* vol. 26, pp. 1587-1598, 2010.
[56] Y. Shen, M. Cui, Q. Wang, F. Shen, B. Zhang, and L. Liang, "Comprehensive reactive power support of DFIG adapted to different depth of voltage sags," *Energies,* vol. 10, p. 808, 2017.
[57] R. Chakib, M. Cherkaoui, and A. Essadki, "Stator flux control by active disturbance rejection control for dfig wind turbine during voltage dip," *Int. J. Circuits Syst. Signal Process.(USA),* vol. 9, pp. 281-288, 2015.
[58] Z. Yin, L. Gong, C. Du, J. Liu, and Y. Zhong, "Integrated position and speed loops under sliding-mode control optimized by differential evolution algorithm for PMSM drives," *IEEE Transactions on Power Electronics,* vol. 34, pp. 8994-9005, 2019.
[59] V. Utkin and H. Lee, "Chattering problem in sliding mode control systems," in *International Workshop on Variable Structure Systems, 2006. VSS'06.*, 2006, pp. 346-350.
[60] J. Zheng, Y. Feng, and Q. Lu, "High-order terminal sliding-mode control for permanent magnet synchronous motor," *Control Theory & Applications,* vol. 26, pp. 697-700, 2009.

[61] P. Mani, R. Rajan, L. Shanmugam, and Y. H. Joo, "Adaptive fractional fuzzy integral sliding mode control for PMSM model," *IEEE Transactions on Fuzzy Systems*, vol. 27, pp. 1674-1686, 2018.
[62] Y. Wang, Y. Feng, X. Zhang, and J. Liang, "A new reaching law for antidisturbance sliding-mode control of PMSM speed regulation system," *IEEE Transactions on Power Electronics*, vol. 35, pp. 4117-4126, 2019.
[63] S. Maiti, C. Chakraborty, and S. Sengupta, "Simulation studies on model reference adaptive controller based speed estimation technique for the vector controlled permanent magnet synchronous motor drive," *Simulation Modelling Practice and Theory*, vol. 17, pp. 585-596, 2009.
[64] K.-H. Kim, "An MRAC-based nonlinear speed control of an interior PM synchronous motor with improved maximum torque operation," *International journal of electronics*, vol. 90, pp. 481-494, 2003.
[65] J. Khedri, M. Chaabane, and M. Souissi, "Multivariable H [infinity] Robust Controller for a Permanent Magnet Synchronous Machine (PMSM)," *International Journal on Computer Science and Engineering*, vol. 7, p. 26, 2015.
[66] B. Liu, "Research on h infinity robust tracking controller for permanent magnet synchronous motor servo system," in *2009 International Conference on Information Engineering and Computer Science*, 2009, pp. 1-5.
[67] S. Xiao-Jing, "Design and simulation of PMSM feedback linearization control system," *TELKOMNIKA: Indonesian Journal of Electrical Engineering*, vol. 11, pp. 1245-1250, 2013.
[68] T. Yucelen, P. V. Medagam, and F. Pourboghrat, "Nonlinear quadratic optimal control for cascaded multilevel static compensators," in *2007 39th North American Power Symposium*, 2007, pp. 523-527.
[69] H. Banks, B. Lewis, and H. T. Tran, "Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach," *Computational Optimization and Applications*, vol. 37, pp. 177-218, 2007.
[70] P. V. Kokotovic, "The joy of feedback: nonlinear and adaptive," *IEEE Control Systems Magazine*, vol. 12, pp. 7-17, 1992.
[71] J. Zhou and Y. Wang, "Real-time nonlinear adaptive backstepping speed control for a PM synchronous motor," *Control Engineering Practice*, vol. 13, pp. 1259-1269, 2005.
[72] M. Karabacak and H. I. Eskikurt, "Speed and current regulation of a permanent magnet synchronous motor via nonlinear and adaptive backstepping control," *Mathematical and Computer Modelling*, vol. 53, pp. 2015-2030, 2011.
[73] W. Wang, F. Tan, H. Ge, J. Wu, and Y. Zhang, "Adaptive integral backstepping control of PMSM with differential terms based on parameters fuzzy self-tuning," *Int. J. Innov. Comput. Inf. Control*, vol. 15, pp. 2165-2181, 2019.
[74] C.-H. Lin, M.-K. Lin, R.-C. Wu, and S.-Y. Huang, "Integral backstepping control for a PMSM drive using adaptive FNN uncertainty observer," in *2012 IEEE International Symposium on Industrial Electronics*, 2012, pp. 668-673.
[75] S. Nadji, N. Sabeur, S. Benaicha, and F. Zaihidee, "Robust Backstepping Control With Integral Action of IPM Synchronous Motor," in *2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, 2018, pp. 1-6.
[76] C. Nevoloso, "Enhanced mathematical modelling of interior permanent magnet synchronous machines for loss minimization control," 2020.

[77] S. L. Kellner and B. Piepenbreier, "General PMSM d, q-model using optimized interpolated absolute and differential inductance surfaces," in *2011 IEEE International Electric Machines & Drives Conference (IEMDC)*, 2011, pp. 212-217.

[78] J. Liu and W. Chen, "Generalized DQ model of the permanent magnet synchronous motor based on extended park transformation," in *2013 1st International Future Energy Electronics Conference (IFEEC)*, 2013, pp. 885-890.

[79] F. Amin, E. Sulaiman, and H. Soomro, "Field Oriented Control Principles for Synchronous Motor," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, pp. 284-288, 2019.

[80] F. Blaschke, "A new method for the structural decoupling of AC induction machines," in *Conf. Rec. IFAC*, 1971, pp. 1-15.

[81] S. Sathiakumar, S. Biswas, and J. Vithayathil, "Microprocessor-based field-oriented control of a CSI-fed induction motor drive," *IEEE Transactions on Industrial Electronics*, pp. 39-43, 1986.

[82] F. Henríquez and E. José, "Quasi-Time-Optimal Controllers for Electrical Drives," Technische Universität München, 2015.

[83] R. K. Lakhe, H. Chaoui, M. Alzayed, and S. Liu, "Universal Control of Permanent Magnet Synchronous Motors with Uncertain Dynamics," in *Actuators*, 2021, p. 49.

[84] T. Jerčić, Š. Ileš, D. Žarko, and J. Matuško, "Constrained field-oriented control of permanent magnet synchronous machine with field-weakening utilizing a reference governor," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije,* vol. 58, pp. 439-449, 2017.

[85] H. Shen, X. Luo, G. Liang, and A. Shen, "A robust dynamic decoupling control scheme for PMSM current loops based on improved sliding mode observer," *Journal of Power Electronics*, vol. 18, pp. 1708-1719, 2018.

[86] S. Farhani, M. Amari, and F. Bacha, "Modeling and Simulation of Energy Conversion System Used in Electric Vehicle," *International Journal of Circuit Theory and Applications*, vol. 9, 2016.

[87] Z. Rongyun, G. Changfu, S. Peicheng, Z. Linfeng, and Z. Changsheng, "Research on chaos control of permanent magnet synchronous motor based on the synthetical sliding mode control of inverse system decoupling," *Journal of Vibration and Control*, p. 1077546320936499, 2020.

[88] X. Wang and C. S. Suh, "A nonlinear time–frequency control based FOC for permanent magnet synchronous motors," *International Journal of Dynamics and Control*, pp. 1-11, 2020.

[89] Y. Wei, Y. Wei, Y. Sun, H. Qi, and X.-Q. Guo, "Prediction Horizons Optimized Nonlinear Predictive Control for PMSM Position System," *IEEE Transactions on Industrial Electronics*, 2019.

[90] J. Jiang, X. Zhou, W. Zhao, and W. Li, "A model reference adaptive sliding mode control for the position control of permanent magnet synchronous motor," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, p. 0959651820941954, 2020.

[91] W. Wang, F. Tan, J. Wu, H. Ge, H. Wei, and Y. Zhang, "Adaptive Integral Backstepping Controller for PMSM with AWPSO Parameters Optimization," *Energies*, vol. 12, p. 2596, 2019.

[92] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Transactions on automatic control*, vol. 39, pp. 1910-1914, 1994.

[93] X. Sun, H. Yu, J. Yu, and X. Liu, "Design and implementation of a novel adaptive backstepping control scheme for a PMSM with unknown load torque," *IET Electric Power Applications,* vol. 13, pp. 445-455, 2019.
[94] A. Larbaoui, B. Belabbes, A. Meroufel, A. Tahour, and D. Bouguenna, "Backstepping control with integral action of PMSM integrated according to the MRAS Observer," *IOSR Journal of Electrical and Electronics Engineering,* vol. 9, pp. 59-68, 2014.
[95] Y.-H. Lan, "Backstepping control with disturbance observer for permanent magnet synchronous motor," *Journal of Control Science and Engineering,* vol. 2018, 2018.
[96] D. Q. Dang, N. T.-T. Vu, H. H. Choi, and J.-W. Jung, "Neuro-fuzzy control of interior permanent magnet synchronous motors: Stability analysis and implementation," *Journal of Electrical Engineering & Technology,* vol. 8, pp. 1439-1450, 2013.
[97] X.-D. Liu, K. Li, and C.-H. Zhang, "Improved backstepping control with nonlinear disturbance observer for the speed control of permanent magnet synchronous motor," *Journal of Electrical Engineering & Technology*, vol. 14, pp. 275-285, 2019.
[98] F. Fernandez-Bernal, A. Garcia-Cerrada, and R. Faure, "Determination of parameters in interior permanent-magnet synchronous motors with iron losses without torque measurement," *IEEE Transactions on Industry Applications,* vol. 37, pp. 1265-1272, 2001.
[99] C. Kwan and F. L. Lewis, "Robust backstepping control of induction motors using neural networks," *IEEE Transactions on Neural Networks,* vol. 11, pp. 1178-1187, 2000.
[100] S.-C. Tong, X.-L. He, and H.-G. Zhang, "A combined backstepping and small-gain approach to robust adaptive fuzzy output feedback control," *IEEE Transactions on Fuzzy systems,* vol. 17, pp. 1059-1069, 2009.
[101] T. D. Batzel and K. Y. Lee, "An approach to sensorless operation of the permanent-magnet synchronous motor using diagonally recurrent neural networks," *IEEE Transactions on Energy Conversion,* vol. 18, pp. 100-106, 2003.
[102] T.-S. Li, D. Wang, G. Feng, and S.-C. Tong, "A DSC approach to robust adaptive NN tracking control for strict-feedback nonlinear systems," *IEEE transactions on systems, man, and cybernetics, part b (cybernetics),* vol. 40, pp. 915-927, 2009.
[103] T.-S. Li, S.-C. Tong, and G. Feng, "A novel robust adaptive-fuzzy-tracking control for a class of nonlinearmulti-input/multi-output systems," *IEEE Transactions on Fuzzy Systems,* vol. 18, pp. 150-160, 2009.
[104] C. M. Bishop, *Neural networks for pattern recognition*: Oxford university press, 1995.
[105] J. Linares-Flores, J. L. Barahona-Avalos, H. Sira-Ramirez, and M. A. Contreras-Ordaz, "Robust passivity-based control of a buck–boost-converter/DC-motor system: An active disturbance rejection approach," *IEEE Transactions on Industry Applications,* vol. 48, pp. 2362-2371, 2012.
[106] N. Mohan, *Advanced electric drives: analysis, control, and modeling using MATLAB/Simulink*: John wiley & sons, 2014.
[107] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks,* vol. 4, pp. 251-257, 1991.

[108] M. Kubat, "Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7," *The Knowledge Engineering Review,* vol. 13, pp. 409-412, 1999.
[109] A. Essalmi, H. Mahmoudi, A. Abbou, A. Bennassar, and Y. Zahraoui, "DTC of PMSM based on artificial neural networks with regulation speed using the fuzzy logic controller," in *2014 International Renewable and Sustainable Energy Conference (IRSEC)*, 2014, pp. 879-883.
[110] O.-D. Ramírez-Cárdenas and F. Trujillo-Romero, "Sensorless speed tracking of a brushless DC motor using a neural network," *Mathematical and Computational Applications,* vol. 25, p. 57, 2020.
[111] R. Alkhatib, "Artificial Neural Network Activation Functions in Exact Analytical Form (Heaviside, ReLU, PReLU, ELU, SELU, ELiSH)," 2021.
[112] B. C. Love, "Comparing supervised and unsupervised category learning," *Psychonomic bulletin & review,* vol. 9, pp. 829-835, 2002.
[113] C. T. Nautiyal, S. Singh, and U. Rana, "Recognition of noisy numbers using neural network," in *Soft Computing: Theories and Applications*, ed: Springer, 2018, pp. 123-132.
[114] F. Ramirez-Leyva, F. Trujillo-Romero, S. Caballero-Morales, and E. Peralta-Sanchez, "Direct Torque Control of a Permanent-Magnet Synchronous Motor with Neural Networks," in *2014 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2014, pp. 71-76.
[115] F. T. Romero, J. d. C. J. Hernandez, and W. G. Lopez, "Predicting electricity consumption using neural networks," *IEEE Latin America Transactions,* vol. 9, pp. 1066-1072, 2011.
[116] S. Wu and J. Liu, "Simulation Analysis of Dynamic Characteristics of AC Motor Based on BP Neural Network Algorithm," in *The International Conference on Cyber Security Intelligence and Analytics*, 2019, pp. 277-286.
[117] S. A. Sadrossadat and O. Rahmani, "ANN-based method for parametric modelling and optimising efficiency, output power and material cost of BLDC motor," *IET Electric Power Applications,* vol. 14, pp. 951-960, 2020.
[118] A. A. Konate, H. Pan, N. Khan, and J. H. Yang, "Generalized regression and feed-forward back propagation neural networks in modelling porosity from geophysical well logs," *Journal of Petroleum Exploration and Production Technology,* vol. 5, pp. 157-166, 2015.
[119] H. White, *Artificial neural networks*: Blackwell Oxford, 1992.
[120] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks,* vol. 2, pp. 359-366, 1989.
[121] A. Abraham, "Artificial neural networks," *Handbook of measuring system design,* 2005.
[122] R. Kumar, R. Gupta, and A. K. Bansal, "Identification and control of PMSM using artificial neural network," in *2007 IEEE International Symposium on Industrial Electronics*, 2007, pp. 30-35.
[123] T. Hagan Martin and B. Menhaj Mohammad, "Training feed forward networks with the Marquardt algorithm," *IEEE Trans Neural Netw,* vol. 5, pp. 989-993, 1994.
[124] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics,* vol. 11, pp. 431-441, 1963.
[125] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design*: PWS Publishing Co., 1997.

[126] J.-S. Ko and B.-M. Han, "Precision position control of PMSM using neural network disturbance observer on forced nominal plant," in *2006 IEEE International Conference on Mechatronics*, 2006, pp. 316-320.
[127] Y.-B. Yan, J.-N. Liang, T.-F. Sun, J.-P. Geng, and D.-J. Pan, "Torque estimation and control of PMSM based on deep learning," in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, 2019, pp. 1-6.
[128] W. Nawae and K. Thongpull, "PMSM Torque Estimation Based on Machine Learning Techniques," in *2020 International Conference on Power, Energy and Innovations (ICPEI)*, 2020, pp. 137-140.
[129] D. Swaroop, J. C. Gerdes, P. P. Yip, and J. K. Hedrick, "Dynamic surface control of nonlinear systems," in *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, 1997, pp. 3028-3034.
[130] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, "Dynamic surface control for a class of nonlinear systems," *IEEE transactions on automatic control*, vol. 45, pp. 1893-1899, 2000.
[131] J.-H. Kim, C.-W. Park, E. Kim, and M. Park, "Fuzzy adaptive synchronization of uncertain chaotic systems," *Physics letters A*, vol. 334, pp. 295-305, 2005.
[132] H. Mhaskar, Q. Liao, and T. Poggio, "When and why are deep networks better than shallow ones?," in *Proceedings of the AAAI conference on artificial intelligence*, 2017.
[133] S. Liang and R. Srikant, "Why deep neural networks for function approximation?," *arXiv preprint arXiv:1610.04161*, 2016.
[134] M. Shao, H. Yu, J. Yu, and B. Shan, "Position control of permanent magnet synchronous motor speed sensorless servo system via backstepping," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, 2015, pp. 4089-4094.
[135] C.-X. Chen, Y.-X. Xie, and Y.-H. Lan, "Backstepping control of speed sensorless permanent magnet synchronous motor based on slide model observer," *International Journal of Automation and Computing*, vol. 12, pp. 149-155, 2015.
[136] I. Bouyakoub, R. Taleb, H. Mellah, and A. Zerglaine, "Implementation of space vector modulation for two level Three-phase inverter using dSPACE DS1104," *Available at SSRN 3918716*, 2020.
[137] Z. Miletic, W. Tremmel, R. Bründlinger, J. Stöckl, and B. Bletterie, "Optimal control of three-phase PV inverter under Grid voltage unbalance," in *2019 21st European Conference on Power Electronics and Applications (EPE'19 ECCE Europe)*, 2019, pp. P. 1-P. 11.
[138] D. Pham, "Modeling and simulation of two level three-phase voltage source inverter with voltage drop," in *2017 Seventh International Conference on Information Science and Technology (ICIST)*, 2017, pp. 317-322.
[139] J. B. Kummari and S. Keerthipati, "Operation of nine-phase induction machine under single-phase open-winding fault condition using dodecagonal SVPWM and hexagonal SVPWM," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, 2019, pp. 3231-3236.
[140] H. Chen, Y. He, J. Liu, X. Chen, H. Xiao, W. Zhi, and R. Cheng, "A novel hybrid SVPWM modulation algorithm for five level active neutral-point-clamped converter," in *2019 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2019, pp. 2494-2498.

[141] G. Vivek, J. Biswas, A. K. Muthavarapu, M. D. Nair, and M. Barai, "Optimised Design and Implementation of SVPWM Switching sequences for Five level VSI," in *2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, 2018, pp. 1-6.