

Article

Not peer-reviewed version

Learning-Driven Intelligent Passivity Control Using Nonlinear State Observers for Induction Motors

[Belkacem Bekhiti](#) , Kamel Hariche , [Mohamed Roudane](#) , [Aleksey Kabanov](#) ^{*} , [Vadim Kramar](#)

Posted Date: 16 July 2025

doi: 10.20944/preprints202507.1158.v1

Keywords: induction motor; Lyapunov stability; non-linear observer; passivity control; recurrent neural networks; adaptive neural-fuzzy control



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Learning-Driven Intelligent Passivity Control Using Nonlinear State Observers for Induction Motors

Belkacem Bekhiti ¹, Kamel Hariche ², Mohamed Roudane ³, Aleksey Kabanov ^{4,*}
and Vadim Kramar ⁴

¹ Institute of Aeronautics and Space Studies (IASS), Aeronautical Sciences Laboratory, Blida 09000, Algeria

² Institute of electrical and electronic engineering, Boumerdes 35000, Algeria

³ Department of Mechanical Engineering, Saad Dahlab, Blida 09000, Algeria

⁴ Robotics and Intelligent Control Systems Laboratory, Sevastopol State University, Sevastopol 299053, Russia

* Correspondence: kabanovaleksey@gmail.com

Abstract: This paper proposes a learning-driven passivity-based control (PBC) strategy for sensorless induction motors, combining a nonlinear adaptive observer with recurrent neural networks (RNNs) to improve robustness and estimation accuracy under dynamic conditions. The main novelty is the integration of neural learning into the passivity framework, enabling real-time compensation for unmodeled dynamics and parameter uncertainties with only one gain adjustment across a broad speed range. Lyapunov-based analysis guarantees the global stability of the closed-loop system. Experiments on a 1.1 kW induction motor confirm the approach's effectiveness over conventional observer-based and fuzzy-enhanced methods. Under torque reversal and flux variation, the proposed controller achieves a torque mean absolute error (MAE) of 0.18 Nm and flux MAE of 0.21 Wb, compared to 1.58 Nm and 0.85 Wb with classical PBC. The peak torque deviation drops from 42.52% to 30.85% of nominal, torque symmetric mean absolute percentage error (SMAPE) improves by 7.6%, and settling time is reduced to 985 ms versus 1120 ms. These results validate the controller's precision, adaptability, and robustness in real-world sensorless motor control.

Keywords: induction motor; Lyapunov stability; non-linear observer; passivity control; recurrent neural networks; adaptive neural-fuzzy control

1. Introduction

The control of induction motors (IMs) continues to attract strong research interest due to their nonlinear structure, sensitivity to parameter variations, and widespread use in industrial applications. Traditional control methods often rely on accurate measurements of rotor speed and flux, which may not always be available or reliable in sensorless configurations. In this context, passivity-based control has emerged as a promising framework due to its solid theoretical foundation and robustness to system uncertainties. Early works demonstrated the use of passivity theory to ensure stable torque and flux regulation in IMs. For example, Cecati [1] proposed a passivity-based position controller, while Duarte-Mermoud [2] introduced an adaptive passivity approach for robust speed control. Travieso-Torres [3] extended this framework with adaptive structures for enhanced robustness under parameter variations. Moreover, Mansouri [4] applied passivity theory to sensorless IM control by integrating it with nonlinear observers for improved state reconstruction. In parallel, adaptive control strategies have played a pivotal role in addressing variations in machine parameters, such as rotor resistance and load torque, which frequently degrade control performance. Marino [5] introduced an adaptive input-output linearizing controller for IMs, achieving asymptotic tracking despite parametric uncertainties. Similarly, Chenafa [6] and Espinosa-Pérez [7] developed globally stable adaptive schemes using nonlinear feedback structures. In the sensorless domain, Belkacem Bekhiti [8], [9] proposed adaptive Luenberger and extended Kalman filter observers to enhance rotor speed estimation without mechanical sensors. More recently, Bekhiti [10] extended the

adaptation ideas by validating a hyper-stability-based multivariable nonlinear adaptive control framework on induction motors, demonstrating strong experimental performance under non-ideal conditions. The integration of nonlinear state observers has further strengthened sensorless control by enabling full-state estimation based solely on measurable stator currents and voltages. Gauthier [11] and Busawon [12] contributed to the theoretical foundations of nonlinear observers, while Mansouri [13] tailored these concepts to IMs operating under field-oriented control. These observer designs significantly improve robustness and observability, especially at low speeds and under load disturbances.

More recently, artificial intelligence (AI) and neural networks have emerged as powerful tools in intelligent motor control. Neural models can approximate complex nonlinear mappings and adapt online, making them particularly suitable for estimation and control in dynamic environments. Beltran-Carbajal [14] and Moghadasian [15] proposed neural network-based controllers for IMs that adapt to system variations in real time. Sujatha and Vaisakh [16], along with Brandstetter and Kuchar [17], demonstrated the effectiveness of adaptive neuro-fuzzy inference systems (ANFIS) and radial basis function (RBF) networks in sensorless IM control. Acikgoz [18] implemented a real-time speed controller using a type-2 fuzzy neural network, while Hussain [19] introduced a type-II adaptive neuro-fuzzy controller with strong transient response capabilities.

Several studies have specifically integrated AI-based observers into MRAS or passivity-based frameworks. For instance, Medjadji [20] and Govindharaj [21] enhanced speed estimation using fuzzy and neuro-fuzzy observers. Parimalasundar [22] combined artificial neural networks with sliding mode control for power steering applications, and Mekrini [23] applied fuzzy logic in the control of asynchronous machines. Recent contributions by Sun [24] and Mienye [25] explored adaptive learning and recurrent neural network architectures to further strengthen robustness. Wu [26] addressed neural optimization by improving the convergence of online learning through gradient normalization—relevant for real-time motor drive applications.

Despite significant progress in adaptive, passivity-based, and learning-driven control of induction motors, a critical gap remains in unifying these approaches into a robust and practical sensorless control framework. Existing methods often treat passivity and learning separately or rely on fixed observers with limited adaptability. Neural and neuro-fuzzy controllers, while effective in handling nonlinearities, frequently lack formal stability guarantees or disrupt the energy-preserving structure of the system. These limitations become especially pronounced under low-speed operation, load transients, and parameter variations, which are common in industrial applications. Moreover, many nonlinear observers are structurally complex or computationally intensive, making real-time implementation difficult. This motivates the need for a cohesive control architecture that integrates the stability of passivity theory, the adaptability of neural learning, and the practical efficiency of nonlinear observer design in a fully sensorless setting.

This paper addresses the above challenges and presents a unified learning-driven passivity control strategy for induction motors using a nonlinear neural-adaptive observer. The main contributions are summarized as follows:

- A novel neural-based adaptive observer is proposed to estimate rotor flux and speed in real time, enhancing estimation accuracy under partial observability and eliminating the need for mechanical sensors.
- A nonlinear passivity-based control law is developed that integrates the neural learning loop while preserving passivity and ensuring global asymptotic stability.
- Experimental validation is conducted on a 1.1 kW induction motor under dynamic benchmarks (speed reversal, torque steps, and parameter variations), demonstrating up to 88.6% reduction in torque error and 75.3% improvement in flux estimation accuracy compared to classical methods.

This architecture bridges the gap between model-based passivity control and data-driven adaptation, offering a scalable and robust solution for high-performance industrial IM drives. The remainder of this paper is organized as follows: Section 2 presents the nonlinear dynamic model of the IM motor and the passivity framework. Section 3 details the neural adaptive observer design and

learning algorithm. Section 4 introduces the passivity-based control strategy and its integration with the observer. Section 5 describes the experimental platform and test scenarios. Section 6 presents and discusses the results. Finally, Section 7 concludes the paper and outlines future directions.

2. Mathematical Framework and Problem Statement

Before deriving the control laws, it is essential to establish a physically consistent mathematical model of the induction motor that captures the key electromechanical dynamics relevant for observer and controller design. The modeling approach adopted here is based on an energy-based formulation that reflects the internal coupling between electrical and mechanical subsystems. This framework serves as the foundation for the passivity-based and adaptive structures developed later in the paper.

2.1. Dynamic Model of an Asynchronous Motor

In the mathematical analyses, it is standard practice to substitute the squirrel-cage motor, (rotor is characterized by uniform distributed conductors), with a theoretical equivalent rotor. This theoretical model matches the stator in the number of phases and features conductors distributed in a sinusoidal pattern. This method operates under the assumption that only the fundamental harmonic of the rotor's magnetomotive force (MMF) is considered. Practical findings confirm that analyses and controller designs using this reduced model are applicable to the actual machine. However, special attention is required when modeling wound rotors using sinusoidally distributed windings [27,29]. The dynamic model is derived through the direct use of Euler-Lagrange formalism, as outlined in [10] which results to (differential Lagrangian system):

$$\bullet \text{ Electrical Part } \mathbf{D}_e(q_m)\ddot{\mathbf{q}}_e + \mathbf{W}_1(q_m)\dot{q}_m\dot{\mathbf{q}}_e + \mathbf{R}_e\dot{\mathbf{q}}_e = \mathbf{M}_e\mathbf{u} \quad (1)$$

$$\bullet \text{ Mechanical Part } D_m\ddot{q}_m + R_m\dot{q}_m = \mathcal{T} - \mathcal{T}_L = \frac{1}{2}\dot{\mathbf{q}}_e^T \mathbf{W}_1(q_m)\dot{\mathbf{q}}_e - \mathcal{T}_L \quad (2)$$

Or more compactly we write:

$$\begin{bmatrix} \mathbf{D}_e(q_m) & 0 \\ 0 & D_m \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_e \\ \ddot{q}_m \end{bmatrix} + \begin{bmatrix} \mathbf{W}_1(q_m)\dot{q}_m\dot{\mathbf{q}}_e \\ -\frac{1}{2}\dot{\mathbf{q}}_e^T \mathbf{W}_1(q_m)\dot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{R}_e & 0 \\ 0 & R_m \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_e \\ \dot{q}_m \end{bmatrix} = \begin{bmatrix} \mathbf{M}_e\mathbf{u} \\ -\mathcal{T}_L \end{bmatrix} \quad (3)$$

The conventional two-phase $\alpha\beta$ model, where the stator axes are stationary and the rotor axes rotate at the electrical angular velocity, is commonly used for a p pole-pair squirrel-cage induction motor with a uniform air-gap. This model defines the electrical parameters that are essential for analysis and control design.

$$\begin{aligned} \mathbf{D}_e(q_m) &= \begin{bmatrix} L_s \mathbf{I}_2 & L_m e^{j p q_m} \\ L_m e^{-j p q_m} & L_r \mathbf{I}_2 \end{bmatrix}; \quad \mathbf{R}_e = \begin{bmatrix} R_s \mathbf{I}_2 & 0 \\ 0 & R_r \mathbf{I}_2 \end{bmatrix}; \quad \mathbf{M}_e = \begin{bmatrix} \mathbf{I}_2 \\ 0 \end{bmatrix}; \quad \mathcal{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\ e^{j p q_m} &= \begin{bmatrix} \cos(p q_m) & -\sin(p q_m) \\ \sin(p q_m) & \cos(p q_m) \end{bmatrix}; \quad e^{-j p q_m} = (e^{j p q_m})^T; \quad \mathcal{J} = -\mathcal{J}^T \\ \mathbf{W}_1(q_m) &:= \frac{\partial \mathbf{D}_e(q_m)}{\partial q_m} = \begin{bmatrix} 0 & p L_m e^{j p q_m} \\ -p L_m e^{-j p q_m} & 0 \end{bmatrix}; \quad \mathcal{J} e^{-j p q_m} = e^{-j p q_m} \mathcal{J} \end{aligned}$$

The current vector is: $\dot{\mathbf{q}}_e := [\dot{\mathbf{q}}_s^T, \dot{\mathbf{q}}_r^T]^T = [\dot{q}_{sa}, \dot{q}_{s\beta}, \dot{q}_{ra}, \dot{q}_{r\beta}]^T$ with $L_s, L_r, L_m > 0$, $R_s, R_r > 0$, $D_m > 0$, $R_m \geq 0$. The flux vector $\boldsymbol{\lambda}$ is related to the current vector $\dot{\mathbf{q}}_e$ via $\boldsymbol{\lambda} := \mathbf{D}_e(q_m)\dot{\mathbf{q}}_e = [\boldsymbol{\lambda}_s^T, \boldsymbol{\lambda}_r^T]^T = [\lambda_{sa}, \lambda_{s\beta}, \lambda_{ra}, \lambda_{r\beta}]^T$. To carry out the controller design, a suitable linear factorization of these workless forces into a form

$$\begin{bmatrix} \mathbf{W}_1(q_m)\dot{q}_m\dot{\mathbf{q}}_e \\ -\frac{1}{2}\dot{\mathbf{q}}_e^T \mathbf{W}_1(q_m)\dot{\mathbf{q}}_e \end{bmatrix} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \quad (4)$$

In particular, the matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ must be chosen such that $\dot{\mathbf{D}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is skew-symmetric, and its 3rd-4th rows are independent of the quantity $\dot{\mathbf{q}}_e$. These conditions are

necessary for the subsequent stability analysis. By utilizing the transpose of \mathcal{T} and $\mathbf{W}_1(q_m)$, it becomes apparent that the desired objectives can be attained with the following selection

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & 0 & \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \\ -pL_m \mathcal{J} e^{-\mathcal{J} p q_m} & 0 & 0 \\ \mathbf{f}^\top(\mathbf{q}, \dot{\mathbf{q}}) & 0 & 0 \end{bmatrix}; \text{ and } \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) = pL_m \mathcal{J} e^{\mathcal{J} p q_m} \dot{\mathbf{q}}_r \quad (5)$$

This decomposition results in the next compact model [13]:

$$\mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{R} \dot{\mathbf{q}} = \mathbf{M} \mathbf{u} + \boldsymbol{\xi} \quad (6)$$

with $\mathbf{D}(\mathbf{q}) = \text{diag}(\mathbf{D}_e(q_m), D_m)$, $\mathbf{R} = \text{diag}(\mathbf{R}_e, R_m)$, $\dot{\mathbf{q}} = [\dot{q}_e^\top, \dot{q}_m]^\top$, $\mathbf{M} = [\mathbf{M}_e^\top, 0]^\top$, and $\boldsymbol{\xi} = [0, -\mathcal{J}_L]^\top$

2.2. Problem Formulation

Consider the induction motor model given by Eq(6) with outputs \mathcal{T} and $\|\boldsymbol{\lambda}_r\|$ to be controlled. Assume that:

A1 \mathcal{T}_d is an unknown constant (switched step).

A2 $\dot{q}_{s\alpha}$, $\dot{q}_{s\beta}$ are accessible.

A3 All parameters are well-known and $R_m > 0$.

Let \mathcal{T}_d be a differentiable function, with a known bounded, first-order derivative, and let $\beta(t) > 0$ be the desired norm of the rotor flux, which is a twice differentiable with known, bounded 1st and 2nd-order derivatives. Under these assumptions, the goal is to design a nonlinear control law that guarantees the stability (i.e. internally) and the asymptotic tracking of both $\mathcal{T}_d(t)$ and $\beta(t)$. This control law must guarantee that, for any initial conditions, all signals within the closed-loop system remain uniformly bounded, thereby achieving the desired system performance.

$$\lim_{t \rightarrow \infty} |\mathcal{T} - \mathcal{T}_d| = 0, \quad \lim_{t \rightarrow \infty} \|\boldsymbol{\lambda}_r\| - \beta = 0 \quad (7)$$

To address the need for accurate flux and torque estimation using only stator-side measurements, we proceed to design a reduced-order nonlinear observer formulated in the stationary $\alpha\beta$ reference frame. The observer is structured to be consistent with the motor's energy-based dynamics and is later augmented with a recurrent neural network to enhance estimation accuracy under nonlinearities and uncertainties.

3. Nonlinear Observer Design

In this section, we review several flux observer designs from the literature, followed by the development of a tailored observer formulated specifically for the nonlinear framework considered in this work.

3.1. Flux Observer in a Closed Loop

A foundational reference in this area is the observer introduced by Verghese and Sanders [32], which has since inspired several variants, including those proposed by De Luca and Ulivi [33], Garcia in [34], and by Mansouri [4]. The general structure of this classical observer is given by:

$$\frac{d}{dt} \begin{bmatrix} \hat{\mathbf{i}}_s \\ \hat{\boldsymbol{\lambda}}_r \end{bmatrix} = \begin{bmatrix} -\gamma \mathbf{I}_2 & (K/T_r) \mathbf{I}_2 - K p \Omega \mathcal{J} \\ (L_m/T_r) \mathbf{I}_2 & (-1/T_r) \mathbf{I}_2 + p \Omega \mathcal{J} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{i}}_s \\ \hat{\boldsymbol{\lambda}}_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_2 \end{bmatrix} \mathbf{u}_s + \begin{bmatrix} k_1 \mathbf{I}_2 + k_2 p \Omega \mathcal{J} \\ k_3 \mathbf{I}_2 + k_4 p \Omega \mathcal{J} \end{bmatrix} \mathbf{e}_s \quad (8)$$

where $\mathbf{i}_s = d\dot{\mathbf{q}}_s/d$, $\Omega = \dot{q}_m$, $\hat{\mathbf{i}}_s = [\hat{i}_{s\alpha}, \hat{i}_{s\beta}]^\top$, $\hat{\boldsymbol{\lambda}}_r = [\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}]^\top$, $\mathbf{u}_s = [u_{s\alpha}, u_{s\beta}]^\top$, the k_i 's being scalars and $\mathbf{e}_s = (\hat{\mathbf{i}}_s - \mathbf{i}_s)$. Note that the gains depend on the speed in (8). We show the diagram block of this observer in Figure. 1.

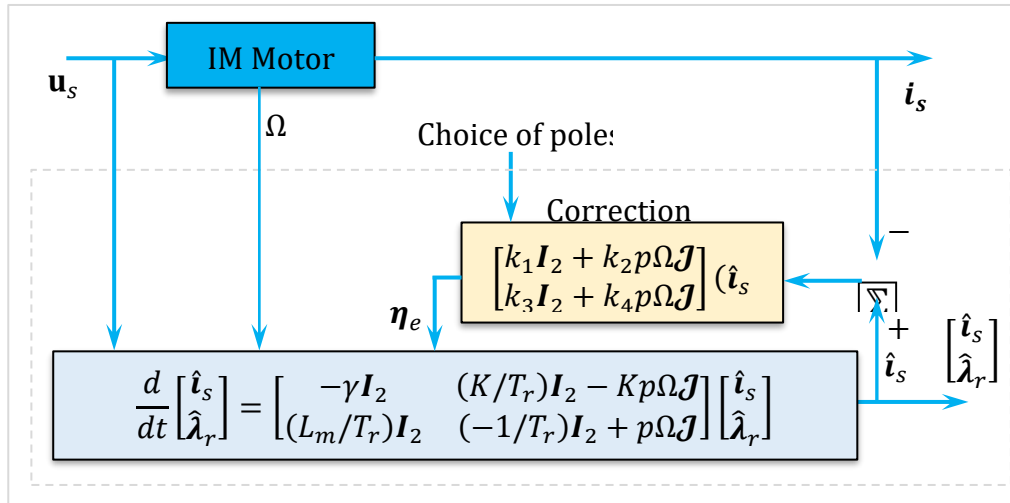


Figure 1. Closed-loop flux observer block diagram.

The resulting model for the observer error dynamics is then

$$\frac{de}{dt} = \begin{bmatrix} (k_1 - \gamma)I_2 + k_2 p \Omega J & (K/T_r)I_2 - K p \Omega J \\ k_3 + (L_m/T_r)I_2 + k_4 p \Omega J & (-1/T_r)I_2 + p \Omega J \end{bmatrix} e \text{ with } e = \begin{bmatrix} \hat{i}_s - i_s \\ \hat{\lambda}_r - \lambda_r \end{bmatrix} \quad (9)$$

Note that we can freely determine the scalar coefficients in equation (9). If k_1 and k_3 are selected such that $k_1 - \gamma = -K/T_r$, $k_3 + (L_m/T_r) = -k_4/T_r$ the error dynamics become

$$\frac{de}{dt} = A_e Q(\Omega) e \text{ with } A_e = \begin{bmatrix} k_2 I_2 & -K I_2 \\ k_4 I_2 & I_2 \end{bmatrix}; Q(\Omega) = \begin{bmatrix} p \Omega J - (I_2/T_r) & 0 \\ 0 & p \Omega J - (I_2/T_r) \end{bmatrix} \quad (10)$$

We select k_2 and k_4 to place the eigenvalues of A_e in arbitrary positions. Note that the characteristic polynomial of A_e is $[s^2 - (1 + k_2)s + k_2 + k_4 K]^2$. If the eigenvalues of A_e are s_1 (twice) and s_2 (twice), then the eigenvalues of time varying matrix $A_e Q(\Omega)$ are $[(-1/T_r) \pm p \Omega j]s_1$ and $[(-1/T_r) \pm p \Omega j]s_2$ using the slow dynamics theory.

3.2. The Proposed Nonlinear Intelligent Observer Design

Passivity-based control strategies are widely recognized for their reliance on rotor flux information, which is often challenging to measure directly. Consequently, considerable research efforts have been directed toward developing methods for estimating these values. Now, we introduce the design of a reduced-order flux observer for an asynchronous motor operating in the $\alpha\beta$ reference frame. The proposed observer requires inputting only the stator voltage, current, and rotor speed. Specifically, this new nonlinear-observer incorporates the speed measurements, ensuring that only the electrical quantities are required in the estimation process [13]. Consider the $\alpha\beta$ model for induction motor:

$$\left\{ \begin{array}{l} \frac{d}{dt} \begin{bmatrix} \dot{q}_{s\alpha} \\ \dot{q}_{s\beta} \\ \lambda_{r\alpha} \\ \lambda_{r\beta} \end{bmatrix} = \begin{bmatrix} -\gamma \dot{q}_{s\alpha} + \frac{K}{T_r} \lambda_{r\alpha} + K p \dot{q}_m \lambda_{r\beta} + \frac{1}{\sigma L_s} u_{s\alpha} \\ -\gamma \dot{q}_{s\beta} - K p \dot{q}_m \lambda_{r\alpha} + \frac{K}{T_r} \lambda_{r\beta} + \frac{1}{\sigma L_s} u_{s\beta} \\ \frac{L_m}{T_r} \dot{q}_{s\alpha} - \frac{1}{T_r} \lambda_{r\alpha} - p \dot{q}_m \lambda_{r\beta} \\ \frac{L_m}{T_r} \dot{q}_{s\beta} + p \dot{q}_m \lambda_{r\alpha} - \frac{1}{T_r} \lambda_{r\beta} \end{bmatrix} \\ [y_1 y_2]^T = [\dot{q}_{s\alpha} \dot{q}_{s\beta}]^T \end{array} \right. \quad (11)$$

where

$$T_r = \frac{L_r}{R_r}; \quad \sigma = 1 - \frac{L_m^2}{L_s L_r}; \quad K = \frac{L_m}{\sigma L_s L_r}; \quad \gamma = \frac{R_s}{\sigma L_s} + \frac{R_r L_m^2}{\sigma L_s L_r^2}$$

The system (11) is the form

$$\dot{\mathbf{z}} = \mathbf{F}(\Omega)\mathbf{z} + \mathbf{G}(\mathbf{u}, \Omega, \mathbf{z}) \text{ and } \mathbf{y} = \mathbf{C}\mathbf{z} \quad (12)$$

with

$$\left\{ \begin{array}{l} \mathbf{F}(\Omega) = \begin{bmatrix} 0 & \mathbf{F}_1(\Omega) \\ 0 & 0 \end{bmatrix}; \mathbf{G}(\mathbf{u}, \Omega, \mathbf{z}) = \begin{bmatrix} \mathbf{g}_1(\mathbf{u}, \Omega, \mathbf{z}) \\ \mathbf{g}_2(\mathbf{u}, \Omega, \mathbf{z}) \end{bmatrix}; \mathbf{u} = \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix}; \mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}; \mathbf{y} = \begin{bmatrix} \dot{q}_{s\alpha} \\ \dot{q}_{s\beta} \end{bmatrix}; \mathbf{z}_1 = \begin{bmatrix} \dot{q}_{s\alpha} \\ \dot{q}_{s\beta} \end{bmatrix}; \mathbf{z}_2 = \begin{bmatrix} \lambda_{r\alpha} \\ \lambda_{r\beta} \end{bmatrix}; \Omega = \dot{q}_m \\ \mathbf{F}_1(\Omega) = \begin{bmatrix} \frac{K}{T_r} & Kp\Omega \\ -Kp\Omega & \frac{K}{T_r} \end{bmatrix}; \mathbf{g}_1(\mathbf{u}, \Omega, \mathbf{z}) = \begin{bmatrix} -\gamma\dot{q}_{s\alpha} + \frac{1}{\sigma L_s} u_{s\alpha} \\ -\gamma\dot{q}_{s\beta} + \frac{1}{\sigma L_s} u_{s\beta} \end{bmatrix}; \mathbf{g}_2(\mathbf{u}, \Omega, \mathbf{z}) = \begin{bmatrix} \frac{L_m}{T_r} \dot{q}_{s\alpha} - \frac{1}{T_r} \lambda_{r\alpha} - p\Omega \lambda_{r\beta} \\ \frac{L_m}{T_r} \dot{q}_{s\beta} + p\Omega \lambda_{r\alpha} - \frac{1}{T_r} \lambda_{r\beta} \end{bmatrix}; \text{ and } \mathbf{C}^\top = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{array} \right\}$$

Here, we make the following assumptions as posed by K. Busawon (see [12])

(A4) There exists a class \mathcal{U} of bounded admissible controls, a compact set $\mathcal{K} \subset \mathbb{R}^{2 \times 2}$ and positive constants $\alpha, \beta > 0$ such that for every $\mathbf{u} \in \mathcal{U}$ and every output $\mathbf{y}(t)$ associated with \mathbf{u} and with an initial state $\mathbf{z}(0) \in \mathcal{K}$, we have $0 < \alpha \mathbf{I}_2 \leq \mathbf{F}_1^\top \mathbf{F}_1 \leq \beta \mathbf{I}_2$.

(A5) $\Omega(t)$ and its time derivative $d\Omega(t)/dt$ are bounded.

(A6) The matrix $\mathbf{F}_1(\Omega)$ is of class \mathcal{C}^r , $r \geq 1$ with respect to their arguments.

(A7) Functions $\mathbf{g}_1, \mathbf{g}_2$ are global Lipschitz with respect to \mathbf{z} uniformly in \mathbf{u} and Ω .

We characterize the observer design for the system (12) in the following theorem

THEOREM: 1 Assume that the system (12) satisfies Assumptions (A4) to (A7). Then there exists $\theta > 0$ such that the system

$$\dot{\hat{\mathbf{z}}} = \mathbf{F}(\Omega)\hat{\mathbf{z}} + \mathbf{G}(\mathbf{u}, \Omega, \hat{\mathbf{z}}) - \mathbf{L}_\theta(\Omega)(\hat{\mathbf{z}} - \mathbf{z}) \text{ with } \mathbf{L}_\theta(\Omega) = \Lambda^{-1}(\Omega)\mathbf{S}_\theta^{-1}\mathbf{C}^\top\mathbf{C} \quad (13)$$

is an exponential observer for the system (12), where \mathbf{S}_θ is the unique solution of the algebraic Lyapunov equation $\theta\mathbf{S}_\theta + \mathbf{A}^\top\mathbf{S}_\theta + \mathbf{S}_\theta\mathbf{A} - \mathbf{C}^\top\mathbf{C} = \mathbf{0}$, $\Lambda(\Omega) = \text{blkdiag}[\mathbf{I}_2, \mathbf{F}_1(\Omega)]$, $\mathbf{S}_\theta^{-1}\mathbf{C}^\top = [2\theta\mathbf{I}_2 \ \theta^2\mathbf{I}_2]^\top$, $\mathbf{A} = [[\mathbf{0}_2 \ \mathbf{I}_2]^\top, [\mathbf{0}_2 \ \mathbf{0}_2]^\top]^\top$ and the explicit formula of $\mathbf{L}_\theta(\Omega)$ is

$$\mathbf{L}_\theta(\Omega) = \Lambda^{-1}(\Omega)\mathbf{S}_\theta^{-1}\mathbf{C}^\top\mathbf{C} = [[2\theta\mathbf{I}_2 \ \mathbf{0}_2]^\top, [\theta^2\mathbf{F}_1^{-1}(\Omega) \ \mathbf{0}_2]^\top]^\top, \text{ and } \theta > 0 \quad (14)$$

Moreover, we can make the dynamics of this observer arbitrarily fast. ■

So, by multiplying the left and right-hand sides of (14) by $\Lambda(\Omega)$ and $\Lambda^\top(\Omega)$ respectively, the following algebraic equation holds:

$$\theta\hat{\mathbf{S}}_\theta + \mathbf{F}^\top(\Omega)\hat{\mathbf{S}}_\theta + \hat{\mathbf{S}}_\theta\mathbf{F}(\Omega) - \mathbf{C}^\top\mathbf{C} = \mathbf{0} \text{ with } \hat{\mathbf{S}}_\theta = \Lambda^\top(\Omega)\mathbf{S}_\theta\Lambda(\Omega) \quad (15)$$

The choice of θ permits the pole placement of the observer according to the speed.

In summary the system can be written as $\dot{\mathbf{x}}_{\text{IM}}(t) = \mathbf{A}_{\text{IM}}(\Omega)\mathbf{x}_{\text{IM}}(t) + \mathbf{B}_{\text{IM}}(\Omega)\mathbf{u}_s(t)$ or

$$\frac{d}{dt} \begin{bmatrix} \dot{q}_s \\ \lambda_r \end{bmatrix} = \begin{bmatrix} -\gamma\mathbf{I}_2 & (K/T_r)\mathbf{I}_2 - Kp\Omega\mathbf{J} \\ (L_m/T_r)\mathbf{I}_2(-1/T_r)\mathbf{I}_2 + p\Omega\mathbf{J} \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \lambda_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_2 \end{bmatrix} \mathbf{u}_s; \mathbf{y} = \dot{q}_s$$

The observer is $\dot{\mathbf{x}}_0(t) = \mathbf{A}_{\text{IM}}(\Omega)\mathbf{x}_0(t) + \mathbf{B}_{\text{IM}}(\Omega)\mathbf{u}_s(t) + \mathbf{L}_\theta(\Omega)\mathbf{e}_s(t)$ or

$$\frac{d}{dt} \begin{bmatrix} \hat{q}_s \\ \hat{\lambda}_r \end{bmatrix} = \begin{bmatrix} -\gamma\mathbf{I}_2 & (K/T_r)\mathbf{I}_2 - Kp\Omega\mathbf{J} \\ (L_m/T_r)\mathbf{I}_2(-1/T_r)\mathbf{I}_2 + p\Omega\mathbf{J} \end{bmatrix} \begin{bmatrix} \hat{q}_s \\ \hat{\lambda}_r \end{bmatrix} + \frac{1}{\sigma L_s} \begin{bmatrix} \mathbf{I}_2 \\ \mathbf{0}_2 \end{bmatrix} \mathbf{u}_s + \begin{bmatrix} 2\theta\mathbf{I}_2 & \mathbf{0} \\ \theta^2\mathbf{h}(\Omega) & \mathbf{0} \end{bmatrix} \mathbf{e}_s$$

So the error equation is $\dot{\mathbf{e}}_s(t) = \mathbf{A}_0(\Omega)\mathbf{e}_s(t)$ with $\mathbf{A}_0(\Omega) = \mathbf{A}_{\text{IM}}(\Omega) - \mathbf{L}_\theta(\Omega)$ or

$$\frac{d}{dt} \mathbf{e}_s = \begin{bmatrix} 2\theta\mathbf{I}_2 - \gamma\mathbf{I}_2 & (K/T_r)\mathbf{I}_2 - Kp\Omega\mathbf{J} \\ \theta^2\mathbf{h}(\Omega) + (L_m/T_r)\mathbf{I}_2(-1/T_r)\mathbf{I}_2 + p\Omega\mathbf{J} \end{bmatrix} \mathbf{e}_s \text{ and } \mathbf{h}(\Omega) = \frac{(K/T_r)\mathbf{I} + Kp\Omega\mathbf{J}}{(K/T_r)^2 + (Kp\Omega)^2}$$

To incorporate learning-based compensation for model uncertainties and to enhance the robustness of the nonlinear observer, we introduce a radial basis function neural networks (RBFNN) term into the observer dynamics. The network approximates unknown disturbances or residual nonlinearities using a regressor $\boldsymbol{\phi}(t) \in \mathbb{R}^{n_\phi}$ composed of observable and estimated signals. Let $\Delta(\boldsymbol{\phi}) \in \mathbb{R}^{n_z}$ represent an *unknown nonlinear function* (e.g., modeling error or disturbance). The *universal approximation theorem* states that, for any continuous function $\Delta(\boldsymbol{\phi})$, there exists a feedforward neural network (three-layer NN) of the form: $\Delta(\boldsymbol{\phi}) \approx \hat{\Delta}(\boldsymbol{\phi}) = \sum_{i=1}^{n_h} \mathbf{w}_i \psi_i(\boldsymbol{\xi})$ with $\boldsymbol{\xi} = \mathbf{V}\boldsymbol{\phi} + \mathbf{B}$ or we omit the intermediary variable $\boldsymbol{\xi}$ as $\Delta(\boldsymbol{\phi}) = \mathbf{W}^\top \boldsymbol{\Psi}(\boldsymbol{\phi})$ where

$$\bullet \quad \boldsymbol{\Psi}(\boldsymbol{\phi}) = [\psi_1(\boldsymbol{\phi}), \dots, \psi_{n_h}(\boldsymbol{\phi})]^\top \in \mathbb{R}^{n_h} \text{ is the activation vector,}$$

- $\mathbf{W} = \arg \min_{\mathbf{W} \in \mathbb{R}^{n_h \times n_z}} \left\{ \sup_{\phi \in \Omega_\phi} |\Delta(\mathbf{x}) - \mathbf{W}^\top \Psi(\phi)| \right\}$ is the *output weight matrix*,
- $\psi_i(\phi)$ are nonlinear sigmoids activation functions (e.g., tanh, ReLUs, etc.).
- $\mathbf{V} \in \mathbb{R}^{n_h \times n}$: stacked weight matrix (i^{th} row is \mathbf{v}_i^\top), $\mathbf{B} \in \mathbb{R}^{n_h}$: stacked biases,
- The exponential and division are *element-wise*.

The coefficients \mathbf{v}_i^\top and b_i are the hidden layer parameters ($\mathbf{v}_i^\top \in \mathbb{R}^n$ Input-to-hidden weights "nonlinear layer" and $b_i \in \mathbb{R}$ are the Bias of neuron i). Now, the resulting enhanced observer is given by:

$$\dot{\hat{\mathbf{z}}} = \mathbf{F}(\Omega)\hat{\mathbf{z}} + \mathbf{G}(\mathbf{u}, \Omega, \hat{\mathbf{z}}) - \mathbf{\Lambda}^{-1}(\Omega)\mathbf{S}_\theta^{-1}\mathbf{C}^\top(\mathbf{C}\hat{\mathbf{z}} - \mathbf{y}) + \mathbf{W}^\top \Psi(\phi(t)) \quad (16)$$

Here, $\mathbf{W} \in \mathbb{R}^{n_h \times n_z}$ is the adaptive weight matrix of the neural network and the vector $\Psi(\phi(t)) \in \mathbb{R}^{n_h}$ is a vector of radial basis activation functions, typically Gaussian ($\Psi_i = \exp[-\|\phi(t) - \mathbf{c}_i\|/\sigma_i^2]$) and $\phi(t)$ is the regressor vector, e.g., $\phi(t) = [\mathbf{z}^\top, \mathbf{u}^\top, \Omega]^\top$. The network parameters are updated online via adaptation rules derived from Lyapunov stability analysis to ensure convergence and passivity preservation. The RBF neural term $\mathbf{f}_{RBF}(\phi(t)) = \mathbf{W}^\top \Psi(\phi(t))$ is added as a correction to the standard observer dynamics, to compensate for unknown nonlinearities or disturbances not captured by \mathbf{F} and \mathbf{G} . This term acts like a learning system and is updated online (see Figure 2).

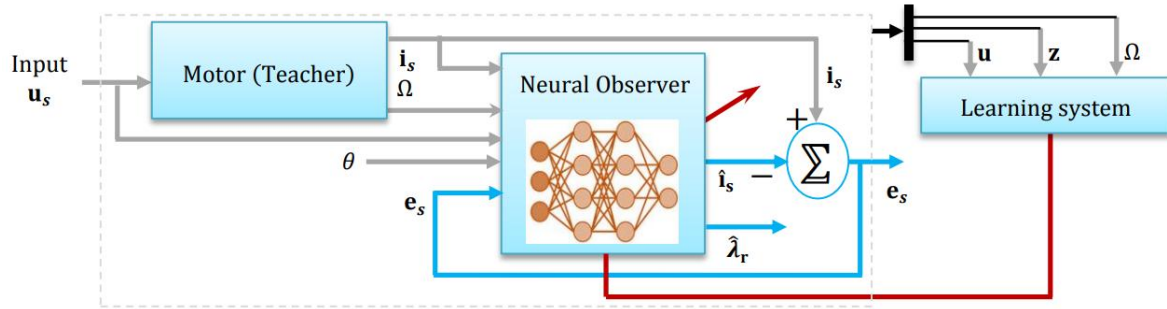


Figure 2. The RBFNN Neural Networks Based Observer: Toward Intelligent Observation.

So the RBFNN becomes: $\Delta(\phi) = \sum_{i=1}^p \mathbf{w}_i \psi_i(\phi) = \mathbf{W}^\top \Psi(\phi)$. In order to train the RBFNN we should to minimize the squared error objective function: $J = \|\mathbf{e}_n\|^2 = \|\hat{\mathbf{y}}_n - \mathbf{y}_n\|^2$. To adapt \mathbf{c}_j , perform $\partial J / \partial \mathbf{c}_j = -\mathbf{e}_n^\top \mathbf{w}_j \cdot \partial \psi_j(\phi_n) / \partial \mathbf{c}_j = -\mathbf{e}_n^\top \mathbf{w}_j \cdot \psi_j(\phi_n) \cdot (\phi_n - \mathbf{c}_j) / \sigma_j^2$. Again, for σ_j we have: $\partial J / \partial \sigma_j = -\mathbf{e}_n^\top \mathbf{w}_j \cdot \partial \psi_j(\phi_n) / \partial \sigma_j = -\mathbf{e}_n^\top \mathbf{w}_j \cdot \psi_j(\phi_n) \cdot \|\phi_n - \mathbf{c}_j\|^2 / \sigma_j^3$. Here is a formal description of this training algorithm

Algorithm: RBF Neural Network Training via Gradient Descent

Begin

Training data: $\phi = \{\phi_n\}_{n=1}^N$, $\mathbf{Y}_t = \{\mathbf{y}_n\}_{n=1}^N$

Learning rate: $\gamma_w, \gamma_c, \gamma_\sigma$. Number of RBF units: m

Initial parameters: output weights \mathbf{W} , RBF centers \mathbf{c}_j , and variances σ_j^2

Normalize input and target data.

Initialize \mathbf{W} , \mathbf{c}_j , σ_j^2 randomly (or using k-means for centers).

Begin loop until convergence

For each training sample (ϕ_n, \mathbf{y}_n) :

- Compute: $\psi_j(\phi_n) = \exp(-\|\phi_n - \mathbf{c}_j\|^2 / \sigma_j^2)$, $j = 1, \dots, m$,

- Stack all activations into a vector: $\Psi(\phi_n) = [\psi_1(\phi), \dots, \psi_m(\phi)]^\top \in \mathbb{R}^m$

- Compute: $\hat{\mathbf{y}}_n = \mathbf{W}^\top \Psi(\phi_n)$, with $\mathbf{e}_n = \hat{\mathbf{y}}_n - \mathbf{y}_n$

- Update output weights: $\mathbf{W} := \mathbf{W} - \gamma_w \Psi(\phi_n) \mathbf{e}_n^\top$; $\mathbf{W}^\top = [\mathbf{w}_1, \dots, \mathbf{w}_m]$

$\mathbf{c}_j := \mathbf{c}_j + \gamma_c \mathbf{e}_n^\top \mathbf{w}_j \cdot \psi_j(\phi_n) \cdot (\phi_n - \mathbf{c}_j) / \sigma_j^2$

$$\sigma_j := \sigma_j + \gamma_\sigma \mathbf{e}_n^T \mathbf{w}_j \cdot \psi_j(\boldsymbol{\varphi}_n) \cdot \|\boldsymbol{\varphi}_n - \mathbf{c}_j\|^2 / \sigma_j^3$$

End the loop

end

3.3. The Proposed Fuzzy-MRAS Estimator

Rotor speed $\hat{\Omega}$ can be estimated using a model reference adaptive system (MRAS) structure composed of two estimators: a *reference model* and an *adaptive model*. Both models independently compute the rotor flux linkage components $\lambda_{r\alpha}$ and $\lambda_{r\beta}$ in the stationary $\alpha\beta$ reference frame. By comparing the flux-linkage estimates, the adaptive model adjusts its internal speed estimate to minimize the discrepancy, thereby converging to the true rotor speed [8, 9]. The reference model, derived from the stator voltage equations, does not depend on rotor speed and is therefore suitable for speed-independent flux estimation. Its expression in the stationary frame is given by:

$$\frac{d\lambda_{r\alpha}^{\text{ref}}}{dt} = \frac{L_r}{L_m} \left[u_{s\alpha} - R_s i_{s\alpha} - \sigma L_s \frac{di_{s\alpha}}{dt} \right]; \frac{d\lambda_{r\beta}^{\text{ref}}}{dt} = \frac{L_r}{L_m} \left[u_{s\beta} - R_s i_{s\beta} - \sigma L_s \frac{di_{s\beta}}{dt} \right] \quad (17)$$

In contrast, the adaptive model, formulated from the rotor voltage equations, explicitly depends on the rotor flux components and the estimated speed $\hat{\Omega}$. It is expressed as:

$$\frac{d\hat{\lambda}_{r\alpha}}{dt} = \frac{L_m}{T_r} i_{s\alpha} - \frac{1}{T_r} \hat{\lambda}_{r\alpha} - p\hat{\Omega}\hat{\lambda}_{r\beta}; \frac{d\hat{\lambda}_{r\beta}}{dt} = \frac{L_m}{T_r} i_{s\beta} - \frac{1}{T_r} \hat{\lambda}_{r\beta} + p\hat{\Omega}\hat{\lambda}_{r\alpha} \quad (18)$$

The angular discrepancy between the estimated and reference flux vectors serves as the adaptation signal, which is processed by a linear PI controller to adjust the estimated speed. Applying the Popov hyperstability criterion [10,20] to guarantee global asymptotic stability of the closed-loop estimation system, the adaptive law is given by:

$$\frac{d\hat{\Omega}}{dt} = K_p \frac{d\varepsilon(t)}{dt} + K_i \varepsilon(t); \text{ with } \varepsilon(t) = [\hat{\lambda}_{r\alpha}\lambda_{r\beta}^{\text{ref}} - \hat{\lambda}_{r\beta}\lambda_{r\alpha}^{\text{ref}}] \quad (19)$$

Fuzzy logic is particularly advantageous for systems that are difficult to model analytically, due to limited data availability, incomplete information, or the inherent complexity of the underlying process [16]. These systems can be easily extended by adding new rules, allowing for performance enhancements and the inclusion of additional features. In conventional MRAS-based speed observers, a proportional-integral (PI) controller is typically employed within the adaptation mechanism to estimate rotor speed by minimizing the error between the reference and adaptive models [18]. In the proposed approach, the PI controller is replaced by a fuzzy logic controller, as depicted in Figure 3. The complete block diagram of the fuzzy controller is presented in Figure 4, and its internal structure is illustrated in Figure 5. The fuzzy logic system operates by first processing the input variables through a fuzzification stage, where crisp input values are mapped to fuzzy sets using predefined membership functions and scale transformations. This scale mapping adjusts the range of the input variables to match the domain expected by the controller. The rule base stores the fuzzy inference rules, while the inference mechanism applies these rules to derive fuzzy conclusions. Common inference strategies include the Mamdani method, which typically uses the logical AND operator for rule evaluation, and the Takagi-Sugeno Model (TSM), which generates functional outputs based on the input variables [23, 36]. Finally, the defuzzification stage converts the fuzzy output set into a crisp control signal suitable for direct application to the system.

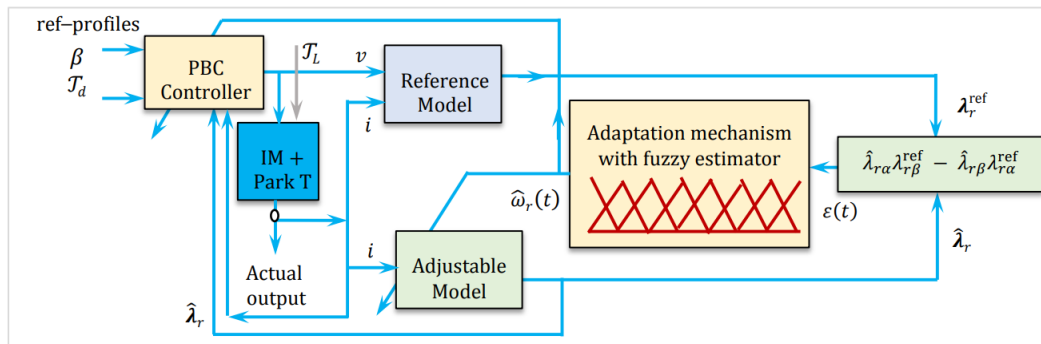


Figure 3. Adaptation mechanism using a fuzzy logic observer.

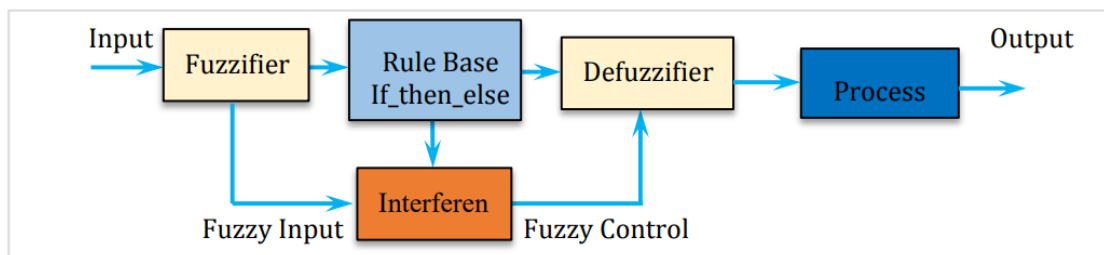


Figure 4. Configuration of the Fuzzy Rule-Based Control Framework.

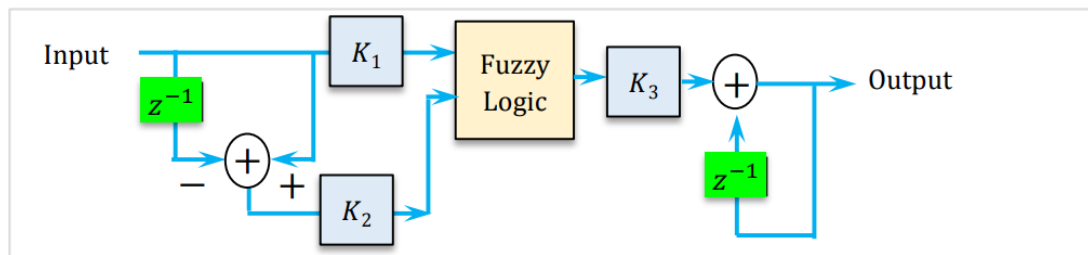


Figure 5. Schematic Representation of the Fuzzy Logic Control System.

Knowing that gains K_1 , K_2 and K_3 are scaling factors, they should be chosen carefully after doing trial and error technique to get the optimal performance of the controller. From the expert estimator, the following memberships of fuzzy sets are used. The rules supporting this system are expressed in the Table 1, where NB: Negative Big, PB: Positive Big, NS: Negative Small, PS: Positive Small and ZE: Zero Equal. The number of inputs to the fuzzy controller can be 2 or 3. In the proposed method to reduce the complexity the two inputs are considered. The linguistic values that can be considered for each input are 3, 5, 7 or 9 as the case may be. Since there are two inputs, the numbers of rules that are possible are $N \times N$ or N^2 which gives 9, 25, 49 or 81 rules for the above linguistic values respectively. The choice of linguistic value should be made keeping into consideration that the control should be good and computation taken by the fuzzy controller should not be high make the process slow. The linguistic value 3 which gives 9 rules is too small to get the better control. Figure. 6 shows the fuzzy membership functions and control surface of the proposed adaptive system.

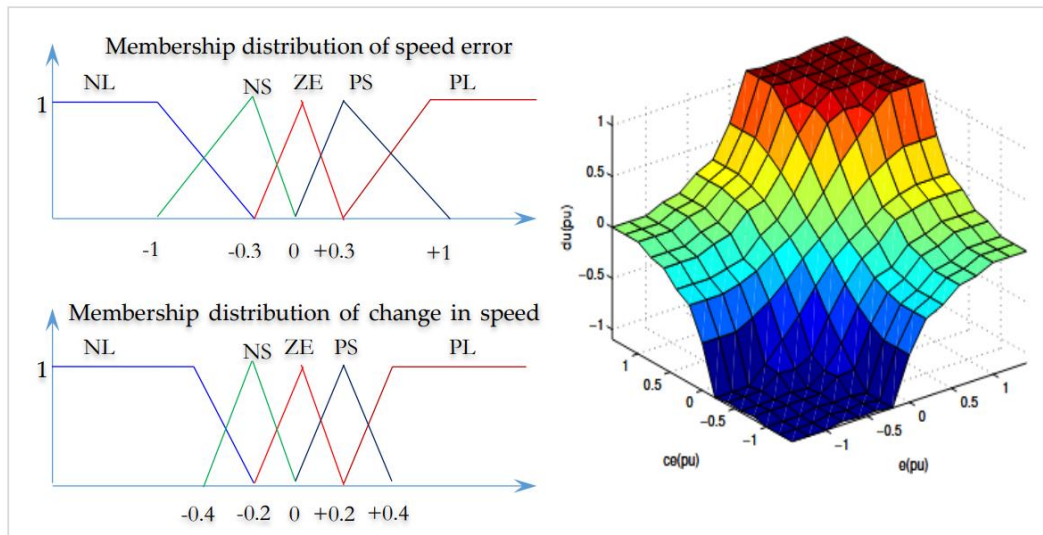


Figure 6. Membership distribution of $\Delta\epsilon$, change in ϵ and the control surface of the fuzzy logic estimator

The fuzzy inference rules used in the proposed system are summarized in Table 1.

Table 1. Fuzzy Rules for the Proposed System with a Rule Confidence Weights for Enhanced Inference

| $e/\Delta e$ | NB | NS | ZE | PS | PB |
|--------------|----|----|----|----|----|
| NB | NB | NB | NB | NS | ZE |
| NS | NB | NB | NS | ZE | PS |
| ZE | NB | NS | ZE | PS | PB |
| PS | NS | ZE | PS | PB | PB |
| PB | ZE | PS | PB | PB | PB |

| $e/\Delta e$ | NB | NS | ZE | PS | PB |
|--------------|------|------|------|------|------|
| NB | 1.00 | 0.95 | 0.90 | 0.85 | 0.80 |
| NS | 0.95 | 0.92 | 0.88 | 0.84 | 0.80 |
| ZE | 0.90 | 0.88 | 0.85 | 0.88 | 0.90 |
| PS | 0.85 | 0.84 | 0.88 | 0.92 | 0.95 |
| PB | 0.80 | 0.85 | 0.90 | 0.95 | 1.00 |

4. Passivity Based Output-Feedback Controller

4.1. Non-Intelligent Sensorless Adaptive Passivity-Based Controller Design

The fundamental concept of passivity lies in shaping the total energy of the system and introducing an additional damping term to guide its dynamic behavior. Using the Euler–Lagrange modeling (ELM) framework, the total energy expression can be systematically formulated and then shaped toward a desired minimum. If the control input is designed to inject a supplementary dissipative term into the system, the convergence to the equilibrium state can be significantly improved beyond what is achieved through the system’s natural dissipation alone. In essence, passivity is an intrinsic property of certain physical systems and is closely linked to the notion of input–output stability [10,35].

DEFINITION: 1 Assume A system is said to be passive if there exists a non-negative function $V: \mathbb{R}^n \rightarrow \mathbb{R}$, called a storage function, with $V(0) = 0$, as for all $\mathbf{u} \in \mathcal{U}$, $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{X}$ and $t \geq 0$, it is satisfied

$$V(\mathbf{x}_t) - V(0) \leq \int_0^t \mathbf{y}^\top(\xi) \mathbf{u}(\xi) d\xi \quad (20)$$

where V is a continuous storage function with continuous derivatives of order r . Condition (20) can also be expressed as $\dot{V}(\mathbf{x}_t) \leq \mathbf{y}^\top(t) \mathbf{u}(t)$ [29]. The system is said to be lossless when $\dot{V}(\mathbf{x}_t) = \mathbf{y}^\top(t) \mathbf{u}(t)$. ■

In cases where the nature of the system does not involve any motion of coordinates or changes in reference frames (systems that are not explicitly time dependent), the Hamiltonian \mathcal{H} is simply the total energy of the system. The Hamiltonian of IM (i.e. conservative system with zero potential energy) equals the sum of the kinetic energy \mathcal{K} and the potential energy \mathcal{P}

$$\mathcal{H}(\dot{q}_e, \dot{q}_m, q_m) = \frac{1}{2} \dot{q}_e^T \mathbf{D}_e(q_m) \dot{q}_e + \frac{1}{2} D_m \dot{q}_m^2 \quad (21)$$

The time derivative of this Hamiltonian function along the trajectories of the system can be expressed as

$$\frac{d\mathcal{H}}{dt} = \frac{1}{2} \dot{q}_e^T \left\{ \frac{\partial \mathbf{D}_e}{\partial q_m} \dot{q}_m \right\} \dot{q}_e + \dot{q}_e^T \mathbf{D}_e(q_m) \ddot{q}_e + D_m \ddot{q}_m \dot{q}_m \quad (22)$$

Using the previous equations $\mathbf{D}_e \ddot{q}_e = \mathbf{u}_e - \mathbf{W}_1 \dot{q}_m \dot{q}_e - \mathbf{R}_e \dot{q}_e$; $D_m \ddot{q}_m = \mathcal{T} - \mathcal{T}_L - R_m \dot{q}_m$ and $\mathcal{T} = [\dot{q}_e^T \mathbf{W}_1 \dot{q}_e]/2$ we get

$$\left\{ \begin{aligned} \frac{d\mathcal{H}}{dt} &= -\mathcal{T}_L \dot{q}_m + \dot{q}_e^T \mathbf{u}_e - [\dot{q}_e^T \mathbf{R}_e \dot{q}_e + \dot{q}_m^T R_m \dot{q}_m] \\ &= -[\dot{q}_e^T \dot{q}_m]^T \begin{bmatrix} \mathbf{R}_e & 0 \\ 0 & R_m \end{bmatrix} \begin{bmatrix} \dot{q}_e \\ \dot{q}_m \end{bmatrix} + [\dot{q}_e^T \dot{q}_m] \begin{bmatrix} \mathbf{u}_e \\ -\mathcal{T}_L \end{bmatrix} \end{aligned} \right\} \quad (23)$$

So, the derivative of Hamiltonian function of IM is given by:

$$d\mathcal{H}/dt = -\dot{\mathbf{q}}^T \mathbf{R} \dot{\mathbf{q}} + \mathbf{y}^T \mathbf{u}; \text{ with } \mathbf{u} = \begin{bmatrix} \mathbf{u}_e \\ -\mathcal{T}_L \end{bmatrix}; \mathbf{y} = \begin{bmatrix} \dot{q}_e \\ \dot{q}_m \end{bmatrix} \quad (24)$$

Stored Energy = Supplied Energy – Dissipated Energy

$$\mathcal{H}(t) - \mathcal{H}(0) = \int_0^t \mathbf{y}^T(\tau) \mathbf{u}(\tau) d\tau - \int_0^t \dot{\mathbf{q}}^T(\tau) \mathbf{R} \dot{\mathbf{q}}(\tau) d\tau \quad (25)$$

Because $\mathcal{H}(t) \geq 0$ we deduce that

$$\mathcal{H}(t) = \mathcal{H}(0) + \int_0^t \mathbf{y}^T(\tau) \mathbf{u}(\tau) d\tau - \int_0^t \dot{\mathbf{q}}^T(\tau) \mathbf{R} \dot{\mathbf{q}}(\tau) d\tau \geq 0 \quad (26)$$

therefore $\int_0^t \mathbf{y}^T(\tau) \mathbf{u}(\tau) d\tau \geq \int_0^t \dot{\mathbf{q}}^T(\tau) \mathbf{R} \dot{\mathbf{q}}(\tau) d\tau - \mathcal{H}(0)$

$$\int_0^t \mathbf{y}^T(\tau) \mathbf{u}(\tau) d\tau \geq \lambda_{\min}\{\mathbf{R}\} \int_0^t \|\dot{\mathbf{q}}(\tau)\|^2 d\tau - \mathcal{H}(0) \quad (27)$$

This outcome proves that the system (IM) is passive, hence the passivity of the induction motor is proved. Also we note that the vector given by $\mathbf{W} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$ contains the forces which do not produce work. As the IM system is an interconnection of two subsystems (electrical and mechanical see Figure 7) then it can be shown that each of the two subsystems are passive; so we arrive at a new approach to the structure of passivity based on the fact that the interconnection of two passive subsystems is passive.

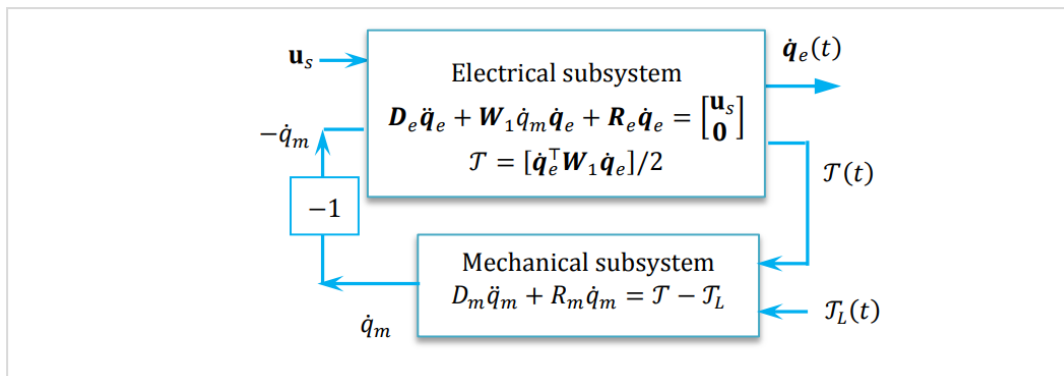


Figure 7. Interaction of subsystems in IM motor.

We consider the total energy \mathcal{H}_e of the electrical subsystem: $\mathcal{H}_e = [\dot{q}_e^T \mathbf{D}_e(q_m) \dot{q}_e]/2$, then the time derivative of this \mathcal{H}_e along the trajectories of the electrical system can be expressed as

$$\left\{ \begin{aligned} \frac{d\mathcal{H}_e}{dt} &= \frac{1}{2} \dot{\mathbf{q}}_e^\top \{\mathbf{W}_1 \dot{\mathbf{q}}_m\} \dot{\mathbf{q}}_e + \dot{\mathbf{q}}_e^\top \mathbf{D}_e(q_m) \dot{\mathbf{q}}_e \\ &= \frac{1}{2} \dot{\mathbf{q}}_e^\top \{\mathbf{W}_1 \dot{\mathbf{q}}_m\} \dot{\mathbf{q}}_e + \dot{\mathbf{q}}_e^\top \mathbf{u}_e - \dot{\mathbf{q}}_e^\top \mathbf{W}_1 \dot{\mathbf{q}}_m \dot{\mathbf{q}}_e - \dot{\mathbf{q}}_e^\top \mathbf{R}_e \dot{\mathbf{q}}_e \\ &= \dot{\mathbf{q}}_e^\top \mathbf{u}_e - \mathcal{T} \dot{\mathbf{q}}_m - \dot{\mathbf{q}}_e^\top \mathbf{R}_e \dot{\mathbf{q}}_e \\ &= [\dot{\mathbf{q}}_e^\top \mathcal{T}] \begin{bmatrix} \mathbf{u}_e \\ -\dot{\mathbf{q}}_m \end{bmatrix} - \dot{\mathbf{q}}_e^\top \mathbf{R}_e \dot{\mathbf{q}}_e \end{aligned} \right\} \quad (28)$$

So, the derivative of this Hamiltonian function is given by $d\mathcal{H}_e/dt = -\dot{\mathbf{q}}_e^\top \mathbf{R}_e \dot{\mathbf{q}}_e + \mathbf{y}_1^\top \mathbf{u}_1$ by integrating this, we obtain

$$\mathcal{H}_e(t) - \mathcal{H}_e(0) = \int_0^t \mathbf{y}_1^\top(\tau) \mathbf{u}_1(\tau) d\tau - \int_0^t \dot{\mathbf{q}}_e^\top(\tau) \mathbf{R}_e \dot{\mathbf{q}}_e(\tau) d\tau \quad (29)$$

Because $\mathcal{H}_e(t) \geq 0$ we deduce that

$$\mathcal{H}_e(0) + \int_0^t \mathbf{y}_1^\top(\tau) \mathbf{u}_1(\tau) d\tau - \int_0^t \dot{\mathbf{q}}_e^\top(\tau) \mathbf{R}_e \dot{\mathbf{q}}_e(\tau) d\tau \geq 0 \quad (30)$$

This inequality gives us:

$$\int_0^t \mathbf{y}_1^\top(\tau) \mathbf{u}_1(\tau) d\tau \geq \lambda_{\min}\{\mathbf{R}_e\} \int_0^t \|\dot{\mathbf{q}}_e(t)\|^2 dt - \mathcal{H}_e(0) \quad (31)$$

which proves the passivity of the electrical subsystem. On the other hand the mechanical subsystem is positive real for all D_m and R_m which is also passive.

PROPOSITION: 1 Consider the asynchronous motor given by equation(6) with outputs \mathcal{T} and $\|\lambda_r\| = \beta$ to be controlled, and assumptions A1 to A3. Let the control law be defined as,

$$\mathbf{u}(t) = L_s \ddot{\mathbf{q}}_s^{\text{ref}} + L_m e^{\mathcal{J}pqm} \ddot{\mathbf{q}}_r^{\text{ref}} + pL_m \mathcal{J} e^{\mathcal{J}pqm} \dot{\mathbf{q}}_r \dot{\mathbf{q}}_m^{\text{ref}} + R_s \dot{\mathbf{q}}_s^{\text{ref}} - K_1(\dot{\mathbf{q}}_m^{\text{ref}}) \dot{\mathbf{q}}_s \quad (32)$$

The superscript ref is the desired variable, with

$$\dot{\mathbf{q}}_r^{\text{ref}} = -e^{\mathcal{J}\rho_d} \begin{bmatrix} \dot{\beta}/R_r \\ \mathcal{T}_d/(p\beta) \end{bmatrix}; \quad \dot{\mathbf{q}}_s^{\text{ref}} = \frac{1}{L_m} e^{\mathcal{J}(pqm+\rho_d)} \begin{bmatrix} \beta + (L_r/R_r)\dot{\beta} \\ L_r \mathcal{T}_d/(p\beta) \end{bmatrix}; \quad \dot{\mathbf{q}}^{\text{ref}} = \begin{bmatrix} \dot{\mathbf{q}}_e^{\text{ref}} \\ \dot{\mathbf{q}}_m^{\text{ref}} \end{bmatrix}; \quad \dot{\mathbf{q}}_e^{\text{ref}} = \begin{bmatrix} \dot{\mathbf{q}}_s^{\text{ref}} \\ \dot{\mathbf{q}}_r^{\text{ref}} \end{bmatrix}$$

and with the following controller dynamics $\dot{\rho}_d(t) = (R_r/p\beta)\mathcal{T}_d(t)$, $\rho_d(0) = 0$ and $\ddot{\mathbf{q}}_m^{\text{ref}}(t) = [-pL_m \dot{\mathbf{q}}_r^\top \mathcal{J} e^{-\mathcal{J}pqm} \dot{\mathbf{q}}_s^{\text{ref}} - R_m \dot{\mathbf{q}}_m^{\text{ref}} - \hat{\mathcal{T}}_L + K_2(\dot{\mathbf{q}}^{\text{ref}}) \dot{\mathbf{q}}_m]/D_m$, $\dot{\mathbf{q}}_m^{\text{ref}}(0) = \dot{\mathbf{q}}_m(0)$.

The gains $K_1(\dot{\mathbf{q}}_m^{\text{ref}})$ and $K_2(\dot{\mathbf{q}}^{\text{ref}})$ are given as

$$K_1(\dot{\mathbf{q}}_m^{\text{ref}}) := \frac{p^2 L_m^2}{4\varepsilon_1} (\dot{\mathbf{q}}_m^{\text{ref}})^2 + k_1; \quad K_2(\dot{\mathbf{q}}^{\text{ref}}) := \frac{p^2 L_m^2}{4\varepsilon_1} [\|\dot{\mathbf{q}}_e^{\text{ref}}\|^2 + (\dot{\mathbf{q}}_m^{\text{ref}})^2] + k_2; \quad (33)$$

with $0 < \varepsilon_1 < R_r/2$, $k_1 \geq 0$, $k_2 > 0$. while the state estimator and the load adaptation law are

$$\mathbf{D}_e(q_m) \ddot{\mathbf{q}}_e + \mathbf{W}_1(q_m) \dot{\mathbf{q}}_m \dot{\mathbf{q}}_e + \mathbf{R}_e \dot{\mathbf{q}}_e = \mathbf{M}_e \mathbf{u} - \mathbf{L}(\dot{\mathbf{q}}_m) \dot{\mathbf{e}}_e \quad (34)$$

$$\dot{\hat{\mathcal{T}}}_L = -\gamma_L \dot{\mathcal{T}}_m, \quad \gamma_L > 0 \quad (35)$$

where $\dot{\mathbf{e}}_e = \dot{\mathbf{q}}_e - \dot{\mathbf{q}}_e^{\text{ref}}$, $\dot{\mathcal{T}}_m = \dot{\mathcal{T}}_m^{\text{ref}} - \dot{\mathcal{T}}_m$ and

$$\mathbf{L}(\dot{\mathbf{q}}_m) = \begin{bmatrix} 2\theta \mathbf{I}_2 & \mathbf{0}_2 \\ \theta^2 \mathbf{h}(\dot{\mathbf{q}}_m) & \mathbf{0}_2 \end{bmatrix}; \quad \text{and } \mathbf{h}(\dot{\mathbf{q}}_m) = \mathbf{F}_1^{-1}(\dot{\mathbf{q}}_m) = \frac{\{\frac{K}{T_r} \mathbf{I} + Kp\dot{\mathbf{q}}_m \mathcal{J}\}}{(K/T_r)^2 + (Kp\dot{\mathbf{q}}_m)^2} \in \mathbb{R}^{2 \times 2}$$

Under these conditions, the closed-loop system guarantees global tracking of torque and flux norm, while ensuring all signals remain uniformly bounded. ■

Proof: Because the control law utilizes the estimated states rather than the actual ones, as opposed to the ideal scenario, the error dynamics in this case are:

$$\mathbf{D}(q) \ddot{\tilde{\mathbf{q}}}(t) + [\mathbf{C}(q, \dot{\mathbf{q}}) + \mathbf{R} + \mathbf{K}(\dot{\mathbf{q}}^{\text{ref}})] \dot{\tilde{\mathbf{q}}}(t) = \mathbf{S}(q_m, \dot{\mathbf{q}}^{\text{ref}}) \dot{\mathbf{e}}_e(t) + \tilde{\boldsymbol{\xi}}(t) \quad (36)$$

where $\mathbf{K}(\dot{\mathbf{q}}^{\text{ref}}) = \text{diag}\{K_1(\dot{\mathbf{q}}_m^{\text{ref}}) \mathbf{I}_2, 0, 0, K_2(\dot{\mathbf{q}}^{\text{ref}})\}$, $\tilde{\boldsymbol{\xi}} = [\mathbf{0}_{1 \times 4}, \hat{\mathcal{T}}_L]^\top = [0, 0, 0, 0, (\hat{\mathcal{T}}_L - \mathcal{T}_L)]^\top$ and

$$S(q_m, \dot{q}^{\text{ref}}) = \begin{bmatrix} \mathbf{0}_2 & pL_m \mathbf{J} e^{j p q_m} \dot{q}_m^{\text{ref}} \\ \mathbf{0}_{1 \times 2} & \mathbf{0}_2 \\ \mathbf{0}_2 & -pL_m (\dot{q}_s^{\text{ref}})^\top \mathbf{J} e^{j p q_m} \end{bmatrix} \in \mathbb{R}^{5 \times 4}; \quad \tilde{\mathbf{q}}(t) = \begin{bmatrix} \dot{q}_s(t) \\ \dot{q}_r(t) \\ \dot{q}_m(t) \end{bmatrix} - \begin{bmatrix} \dot{q}_s^{\text{ref}}(t) \\ \dot{q}_r^{\text{ref}}(t) \\ \dot{q}_m^{\text{ref}}(t) \end{bmatrix} \in \mathbb{R}^{5 \times 1}$$

Alternatively, based on equations (34) and (1), the observation error $\dot{\mathbf{e}}_e$ satisfies the following equation: $\mathbf{D}_e(q_m)\ddot{\mathbf{e}}_e + [\mathbf{W}_1(q_m)\dot{q}_m + \mathbf{L}(\dot{q}_m)]\dot{\mathbf{e}}_e + \mathbf{R}_e\dot{\mathbf{e}}_e = 0$. So, consider the following Lyapunov function candidate

$$V = \frac{1}{2} \tilde{\mathbf{q}}^\top \mathbf{D}(\mathbf{q}) \tilde{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{e}}_e^\top \mathbf{D}_e(q_m) \dot{\mathbf{e}}_e + \frac{1}{2\gamma_L} \tilde{\mathcal{J}}_L^2 \quad (37)$$

Knowing that $\dot{\mathbf{D}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is skew-symmetric and $\dot{\mathbf{D}}_e(q_m) - 2[\mathbf{W}_1(q_m)\dot{q}_m + \mathbf{L}(\dot{q}_m)]$ which yields the derivative

$$\dot{V} = -\tilde{\mathbf{q}}^\top [\mathbf{R} + \mathbf{K}(\dot{q}^{\text{ref}})] \tilde{\mathbf{q}} + \tilde{\mathbf{q}}^\top S(q_m, \dot{q}^{\text{ref}}) \dot{\mathbf{e}}_e - \dot{\mathbf{e}}_e^\top \mathbf{R}_e \dot{\mathbf{e}}_e + \frac{1}{\gamma_L} \tilde{\mathcal{J}}_L \dot{\tilde{\mathcal{J}}}_L + \tilde{\mathbf{q}}^\top \tilde{\xi} \quad (38)$$

Use of equation (35) in the equation above and let we define $\mathbf{z} = [\tilde{\mathbf{q}}^\top, \dot{\mathbf{e}}_e^\top]^\top$, which results in the next quadratic function $\dot{V} = -\mathbf{z}^\top \mathbf{M} \mathbf{z}$ with

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} + \mathbf{K}(\dot{q}^{\text{ref}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_e \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \mathbf{0} & S(q_m, \dot{q}^{\text{ref}}) \\ S^\top(q_m, \dot{q}^{\text{ref}}) & \mathbf{0} \end{bmatrix}$$

Verifying that $\mathbf{K}(\dot{q}^{\text{ref}})$ guarantee the strict positive definiteness of \mathbf{M} , i.e. $\mathbf{M} \geq \delta \mathbf{I}_9 > 0$ ■

To enable full sensorless operation prior to incorporating intelligent learning, the control framework is first augmented with a conventional PI-based speed estimator. This allows rotor speed reconstruction from electrical measurements alone, eliminating the need for mechanical sensors while preserving the structure and stability of the original passivity-based control loop.

The estimation flux is used to drive a PI-like update law for \hat{q}_m . A standard structure is:

$$\hat{q}_m = K_p \varepsilon(t) + K_i \int \varepsilon(t) dt \text{ with } \varepsilon(t) = \dot{q}_s^\top \mathbf{J} \lambda_r = [\hat{\lambda}_{r\alpha}(t) \dot{q}_{s\beta}(t) - \hat{\lambda}_{r\beta}(t) \dot{q}_{s\alpha}(t)] \quad (39)$$

$\varepsilon(t) = \dot{q}_s^\top \mathbf{J} \lambda_r$ is the error between the real and estimated rotor fluxes and currents.

The overall architecture of the non-intelligent sensorless adaptive passivity-based controller is illustrated in Figure 8.

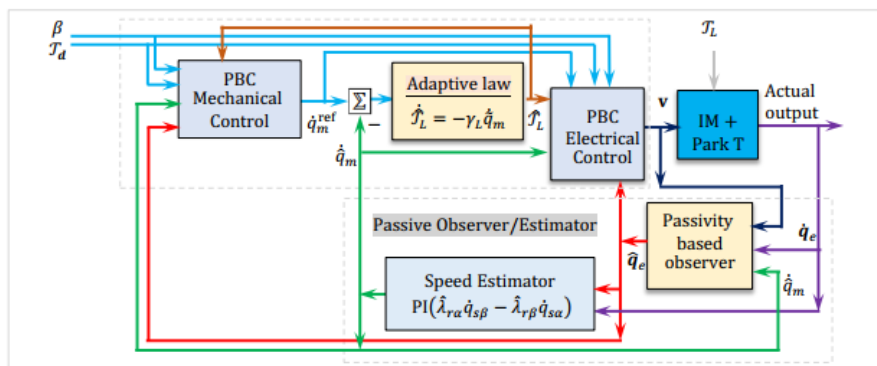


Figure 8. Diagram of the Adaptive Sensorless Passivity Control Scheme for Induction Motor Drive.

Although the previous controller ensures tracking and robustness via passivity control, its performance may degrade under unmodeled dynamics or uncertainties. To enhance adaptability, a recurrent neural network is integrated to learn residual dynamics in real time, yielding an intelligent adaptive control.

4.2. The Proposed Intelligent Adaptive Passivity Based Controller Design

The proposed intelligent adaptive passivity-based controller integrates neural learning for enhanced sensorless performance. As shown in Figure 9, a recurrent neural network estimates both the load torque \hat{T}_L and rotor speed \hat{q}_m , enabling full sensorless operation. These estimates drive a passivity-based control loop to generate stator voltage commands, ensuring robustness in both steady-state and dynamic conditions.

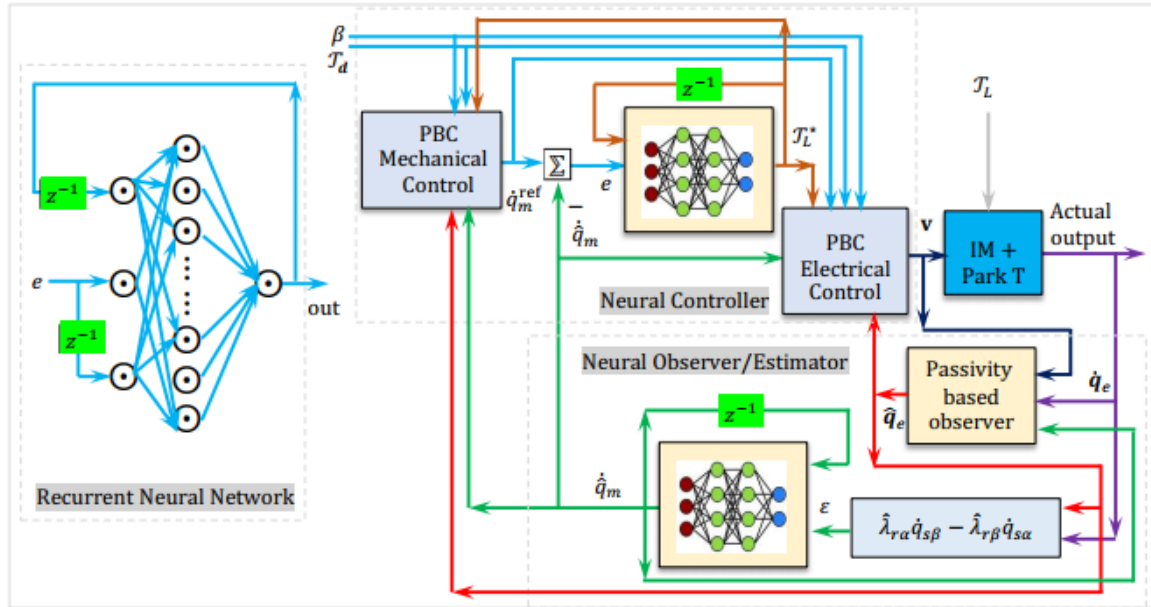


Figure 9. Intelligent Adaptive Sensorless Passivity Control Scheme for Induction Motor Drive.

The proposed scheme employs a recurrent neural network to estimate the load torque \hat{T}_L^* and the rotor speed \hat{q}_m . By capturing temporal dependencies, the RNN effectively models the motor's nonlinear dynamics, outperforming conventional feedforward networks. Let $e_\omega(k) = \dot{q}_m^{\text{ref}}(k) - \hat{q}_m(k)$ and $e_T(k) = \hat{\lambda}_{ra}(k)\dot{q}_{s\beta}(k) - \hat{\lambda}_{r\beta}(k)\dot{q}_{sa}(k)$ denote the speed and the electromagnetic torque errors, at discrete time step k . The internal structure of the RNN is governed by the following equations [25-26]:

1. Hidden State Update: $\mathbb{h}(k) = \Psi(\mathcal{W}_{\text{in}}e(k) + \mathcal{W}_{\text{rec}}\mathbb{h}(k-1) + \mathbf{b}_h)$ where:
 - $\mathbb{h}(k) \in \mathbb{R}^n$: hidden state vector, $\mathbf{b}_h \in \mathbb{R}^n$: bias vector,
 - $\mathcal{W}_{\text{in}} \in \mathbb{R}^{n \times 1}$: input weight matrix, $\mathcal{W}_{\text{rec}} \in \mathbb{R}^{n \times n}$: recurrent weight matrix,
 - $\Psi(\cdot)$: element-wise activation function (e.g., tanh or ReLU).
2. The Output Estimation: $y_{\text{net}}^*(k) = \mathcal{W}_{\text{out}}\mathbb{h}(k) + b_o$ where:
 - $\mathcal{W}_{\text{out}} \in \mathbb{R}^{1 \times n}$: output weight vector, $b_o \in \mathbb{R}$: output bias.

The complete mapping from input error to the needed estimate is compactly expressed as one function: $y_{\text{net}}^*(k) = \mathbf{f}_{\text{RNN}}(e(k), \mathbb{h}(k-1); \theta)$ where $\theta = \{\mathcal{W}_{\text{in}}, \mathcal{W}_{\text{rec}}, \mathcal{W}_{\text{out}}, \mathbf{b}_h, b_o\}$ is the set of all trainable parameters [30]. This RNN-based estimator enables the controller to adaptively generate commands that account for the system's nonlinearities, time delays, and dynamic uncertainties, thus enhancing overall control performance. The RNN parameters θ are optimized to minimize the mean squared error loss: $\mathcal{L} = [\sum_{k=1}^N \{y_{\text{net}}^*(k) - y_{\text{net}}(k)\}^2]/N$. First we compute hidden states and output estimates sequentially: $\mathbb{h}(k) = \Psi(\mathcal{W}_{\text{in}}e(k) + \mathcal{W}_{\text{rec}}\mathbb{h}(k-1) + \mathbf{b}_h)$ and $y_{\text{net}}^*(k) = \mathcal{W}_{\text{out}}\mathbb{h}(k) + b_o$ then compute gradients $\Delta = \partial \mathcal{L} / \partial \theta$ by unrolling the network through time and applying the chain rule. Finally, update parameters using gradient descent or an advanced optimizer (e.g., Adam): $\theta =$

$\theta + \eta\Delta$ where η is the learning rate. This iterative process continues until convergence, enabling the RNN to learn the dynamic mapping from error sequences to the output commands effectively [31]. If we define $\delta(k) = \partial\mathcal{L}/\partial y_{\text{net}}^* = 2[y_{\text{net}}^*(k) - y_{\text{net}}(k)]/N$ then the gradients for output layer are given by $\mathbf{g}_{\text{out}} = \partial\mathcal{L}/\partial\mathbf{w}_{\text{out}} = \sum_{k=1}^N \delta(k) \mathbf{h}^T(k)$ and $\mathbf{g}_{b_o} = \partial\mathcal{L}/\partial b_o = \sum_{k=1}^N \delta(k)$. Also, if we back-propagate through time (BPTT) into hidden states $\epsilon(k) = \partial\mathcal{L}/\partial \mathbf{h}(k) \in \mathbb{R}^n$ then we compute recursively (i.e. $k = N:1$) as $\epsilon(k) = \mathbf{w}_{\text{out}}^T \delta(k) + \mathbf{w}_{\text{rec}}^T [\epsilon(k+1) \odot \Psi'(\mathbf{a}(k+1))]$ where $\mathbf{a}(k) = \mathbf{w}_{\text{in}} e(k) + \mathbf{w}_{\text{rec}} \mathbf{h}(k-1) + \mathbf{b}_h$, $\Psi'(\mathbf{a})$ is the element-wise derivative of activation function and \odot is the element-wise multiplication. Now use $\epsilon(k)$ to compute the following gradients: $\mathbf{g}_{\text{rec}} = \partial\mathcal{L}/\partial\mathbf{w}_{\text{rec}} = \sum_{k=1}^N \mathbf{s}(k) \cdot \mathbf{h}^T(k-1) \in \mathbb{R}^{n \times n}$, $\mathbf{g}_{\text{in}} = \partial\mathcal{L}/\partial\mathbf{w}_{\text{in}} = \sum_{k=1}^N \mathbf{s}(k) \cdot e^T(k) \in \mathbb{R}^{n \times 1}$ and $\mathbf{g}_{b_h} = \partial\mathcal{L}/\partial \mathbf{b}_h = \sum_{k=1}^N \mathbf{s}(k) \in \mathbb{R}^n$ with $\mathbf{s}(k) = [\epsilon(k) \odot \Psi'(\mathbf{a}(k))]$. Finally, if we let η be the learning rate, then we can update: $\mathbf{w}_{\text{out}} \leftarrow \mathbf{w}_{\text{out}} - \eta \mathbf{g}_{\text{out}}$, $\mathbf{w}_{\text{rec}} \leftarrow \mathbf{w}_{\text{rec}} - \eta \mathbf{g}_{\text{rec}}$ etc. All parameters are updated this way. The structure of this learning procedure is given in the flowchart shown in Figure. 10

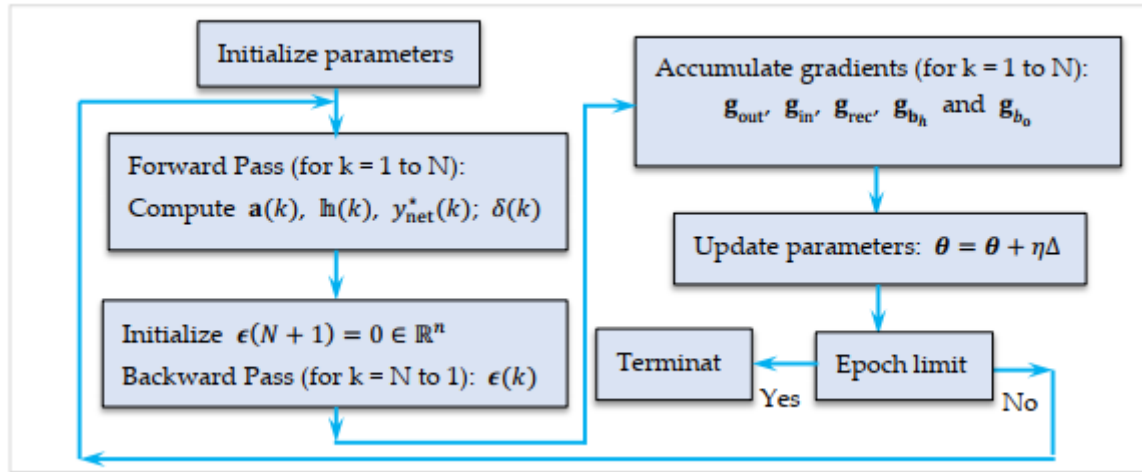


Figure 10. Training Procedure of the RNN-Based Load Torque Estimator Using BPTT.

The RNN parameter update steps using BPTT are described in the following algorithm

Algorithm: Backpropagation for the RNN Torque Estimator

Inputs:

- Training sequence: $\{e(k), y_{\text{net}}(k)\}_{k=1}^N$,
- Initial RNN parameters: $\theta = \{\mathbf{w}_{\text{in}}, \mathbf{w}_{\text{rec}}, \mathbf{w}_{\text{out}}, \mathbf{b}_h, b_o\}$
- Activation function: $\Psi(\cdot)$ and its derivative $\Psi'(\cdot)$, Learning rate: η

Outputs: Updated parameters θ

1. Forward Pass: Compute Activations and Outputs

For $k = 1$ to N :

$$\begin{aligned} \mathbf{a}(k) &= \mathbf{w}_{\text{in}} e(k) + \mathbf{w}_{\text{rec}} \mathbf{h}(k-1) + \mathbf{b}_h, \\ \mathbf{h}(k) &= \Psi(\mathbf{a}(k)), \\ y_{\text{net}}^*(k) &= \mathbf{w}_{\text{out}} \mathbf{h}(k) + b_o \end{aligned}$$

2. Compute Loss Gradient at Output Layer

For $k = 1$ to N :

$$\delta(k) = \partial\mathcal{L}/\partial y_{\text{net}}^* = 2[y_{\text{net}}^*(k) - y_{\text{net}}(k)]/N$$

Initialize parameters

Loop over epochs

1. Forward pass: $\mathbf{a}(k), \mathbf{h}(k), y_{\text{net}}^*(k)$
2. Compute loss and $\delta(k)$
3. Backward pass: compute $\epsilon(k)$
4. Accumulate gradients
5. Update parameters
6. Check convergence

Loop or Terminate

3. Backward Pass: Compute Gradients via BPTT

Initialize $\epsilon(N+1) = 0 \in \mathbb{R}^n$

For $k = N$ down to 1:

$$\epsilon(k) = \mathbf{w}_{\text{out}}^\top \delta(k) + \mathbf{w}_{\text{rec}}^\top [\epsilon(k+1) \odot \Psi'(\mathbf{a}(k+1))]$$

4. Accumulate Gradients

Initialize all gradients to zero.

For $k = 1$ to N : $\mathbf{s}(k) = [\epsilon(k) \odot \Psi'(\mathbf{a}(k))]$

$$\mathbf{g}_{\text{out}}^+ = \delta(k) \cdot \mathbf{h}^\top(k) \in \mathbb{R}^{1 \times n}, \mathbf{g}_{\text{rec}}^+ = \mathbf{s}(k) \cdot \mathbf{h}^\top(k-1) \in \mathbb{R}^{n \times n},$$

$$\mathbf{g}_{\text{in}}^+ = \mathbf{s}(k) \cdot \mathbf{e}^\top(k) \in \mathbb{R}^{n \times 1}, \mathbf{g}_{b_o}^+ = \delta(k) \in \mathbb{R}^{1 \times n}, \mathbf{g}_{b_h}^+ = \mathbf{s}(k) \in \mathbb{R}^n$$

5. Parameter Update

$$\mathbf{w}_{\text{out}} \leftarrow \mathbf{w}_{\text{out}} - \eta \mathbf{g}_{\text{out}}, \mathbf{w}_{\text{rec}} \leftarrow \mathbf{w}_{\text{rec}} - \eta \mathbf{g}_{\text{rec}},$$

$$\mathbf{w}_{\text{in}} \leftarrow \mathbf{w}_{\text{in}} - \eta \mathbf{g}_{\text{in}}, \mathbf{b}_h \leftarrow \mathbf{b}_h - \eta \mathbf{g}_{b_h}, b_o \leftarrow b_o - \eta \mathbf{g}_{b_o}^+$$

6. Return Updated Parameters θ

5. Experimental Results and Discussion

The neural passive controller has been tested on a test setup (Fig. 11) which consists of:

- Three-phase induction motor with rated values:
 - Rated voltage: 380 V,
 - Rated current: 2.2 A,
 - Number of phases: 3,
 - Rated power: 1.1 kW,
 - Frequency: 50 Hz,
 - Rated speed: 1430 rpm/min,
- Synchronous machine with powder brake for loading the IM.
- Electronic power converter (without any control system):
- 3 ϕ (three-phase) diode rectifier and
- VSI composed of three IGBT modules
 - Electronic card for monitoring the stator phase with
- Instantaneous voltage sensors (LEM LV 25-P) and
- Instantaneous current sensors (LEM LA 55-P)
 - Voltage sensor for monitoring the instantaneous value of
- The dc-link voltage. (model LEM CV3-1000)
 - Incremental encoder only for comparison measurements.
- Model RS 256-499, 2500 pulses per round, (for validation-comparison, not for control)
- dSPACE card (DS1104) with a PowerPC 604e at 400 MHz and floating-point digital signal processor DSP-TMS320F240. During the real-time operation of the control, the supervision/capturing of the important data can be done by the CONTROLDESK software provided with DSP board.



Figure 11. Experimental setup for the proposed neural-PBC adaptive control of the IM.

Simulations in MATLAB verified the proposed control scheme using a rated flux reference and 500 V DC-link voltage. The feedback signal reconstruction was confirmed in simulation, and experimental results further validated the effectiveness-robustness of the sensorless control strategy. The electrical and mechanical parameters of the induction motor are listed in Table 2.

Table 2. Induction Motor Parameters.

| Parameters name | | Rating values |
|----------------------------|-------|--------------------------|
| • Stator resistance | R_s | 11.8 Ω |
| • Rotor resistance | R_r | 11.3085 Ω |
| • Mutual cyclic inductance | L_m | 0.5400 H |
| • Stator cyclic inductance | L_s | 0.5578 H |
| • Rotor cyclic inductance | L_r | 0.6152 H |
| • Moment of inertia | D_m | 0.0020 Kg.m ² |
| • Friction coefficient | R_m | 3.1165e-004 N.m/rad/s |
| • Number of pair poles | p | 1 |

The “Control/Command” part is based on the DS1104 R&D Controller Board developed by the German company dSPACE and housed in a computer. This control card is made up of two processors. The master processor manages the application while the slave processor, a DSP (“Digital Signal Processor”) from TEXAS INSTRUMENT (type TMS320F240), generates the PWM (Pulse Width Modulation) control signals in 0/5 V TTL logic. This constitutes the “hardware” part of dSPACE.. The “software” part consists of two software programs. The first, MATLAB/Simulink, allows easy programming of the real-time application under Simulink by using specific blocks (Belonging to the “Real Time Interface (RTI) toolbox”) allowing the configuration of the inputs/outputs of the DS1104 card. The second software, ControlDesk, allows the program code to be loaded onto the card (written in graphical form in Simulink, compiled and transformed into C code), to create a complete experimental environment and in particular a graphical interface for controlling the real-time process, to process the data and save it in a format compatible with MATLAB (for later processing) or even to monitor in real time the evolution of the measured or calculated data using graphical or digital displays. The exchange of information between the two parts described is carried out via an

external connection box (Connector Panel CP1104 from dSPACE) connected to the card via a shielded cable and receiving the analog signals via BNC connectors, an interface for conditioning the PWM control signals and any error signals returned by the Semikron converter and a measurement environment consisting of various sensors. The signal conditioning interface converts the signals from TTL 0/5 V logic to CMOS 0/15 V logic and vice versa. This modification is essential because the DS1104 control card works with TTL 0/5 V logic signals, while these must be in CMOS 0/15 V logic for the voltage inverter. The measurement environment consists of LEM type LA25TP sensors (closed-loop current sensors using the Hall effect) for current measurements, LEM type LV100-500 sensors (closed-loop voltage sensors using the Hall effect) for voltage measurements and an incremental encoder to measure the motor rotation speed. Finally, the measured analog currents are intended for digital processing and must therefore be sampled. Therefore, in order to avoid any spectral aliasing phenomenon, it is necessary to insert a guard filter (with an estimated cut-off frequency of 500 Hz, one order of magnitude above the fundamental frequency of 50 Hz) between each sensor and the analog-to-digital converter.

Analyzing the operation of an IM supplied by a voltage inverter is complex due to its complicated nonlinear nature. Moreover, neural (or neuro-fuzzy) adaptive control of an IM requires solid theoretical foundations in areas such as electrical machines, power electronics, and control systems. Using the dSPACE-instrumented platform, the proposed control strategy is first implemented in Simulink, then deployed in real-time via dSPACE. The PWM-driven inverter operates at a switching frequency of 9.5 kHz. A detailed view of the experimental setup is provided in Figure 12.

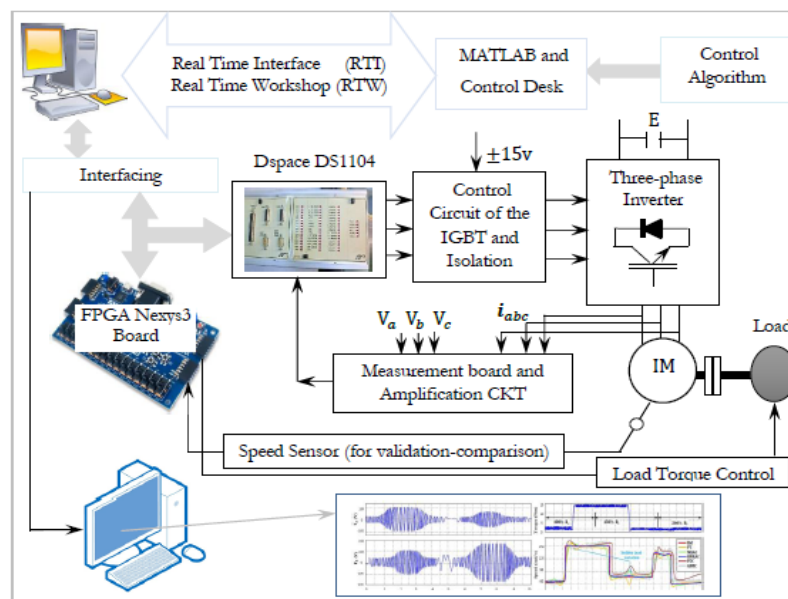


Figure 12. Architecture of the Neural PBC Adaptive Control Scheme for Sensorless IM Drive.

5.1. Study of the Proposed Nonlinear Intelligent Observer

The performance of the proposed nonlinear observer is first evaluated in open-loop and then integrated into a closed-loop configuration with the nonlinear controller of the induction motor. To align the observer dynamics with those of the plant, a tuning parameter $5 \leq \theta \leq 30$ is used. The trajectories of the motor and observer poles, are experimentally analyzed under this setting. Figure 13 illustrates the rotor flux observation error obtained at various operating speeds (0.0125, 3.4, and 150 rad/s), with $\theta = 30$, and $\theta = 30$, demonstrating the accuracy and robustness of the observer. The given architecture is tested under different perturbation scenarios, including parameter variations and unmodeled dynamics, to validate the strength and reliability of the proposed intelligent strategy.

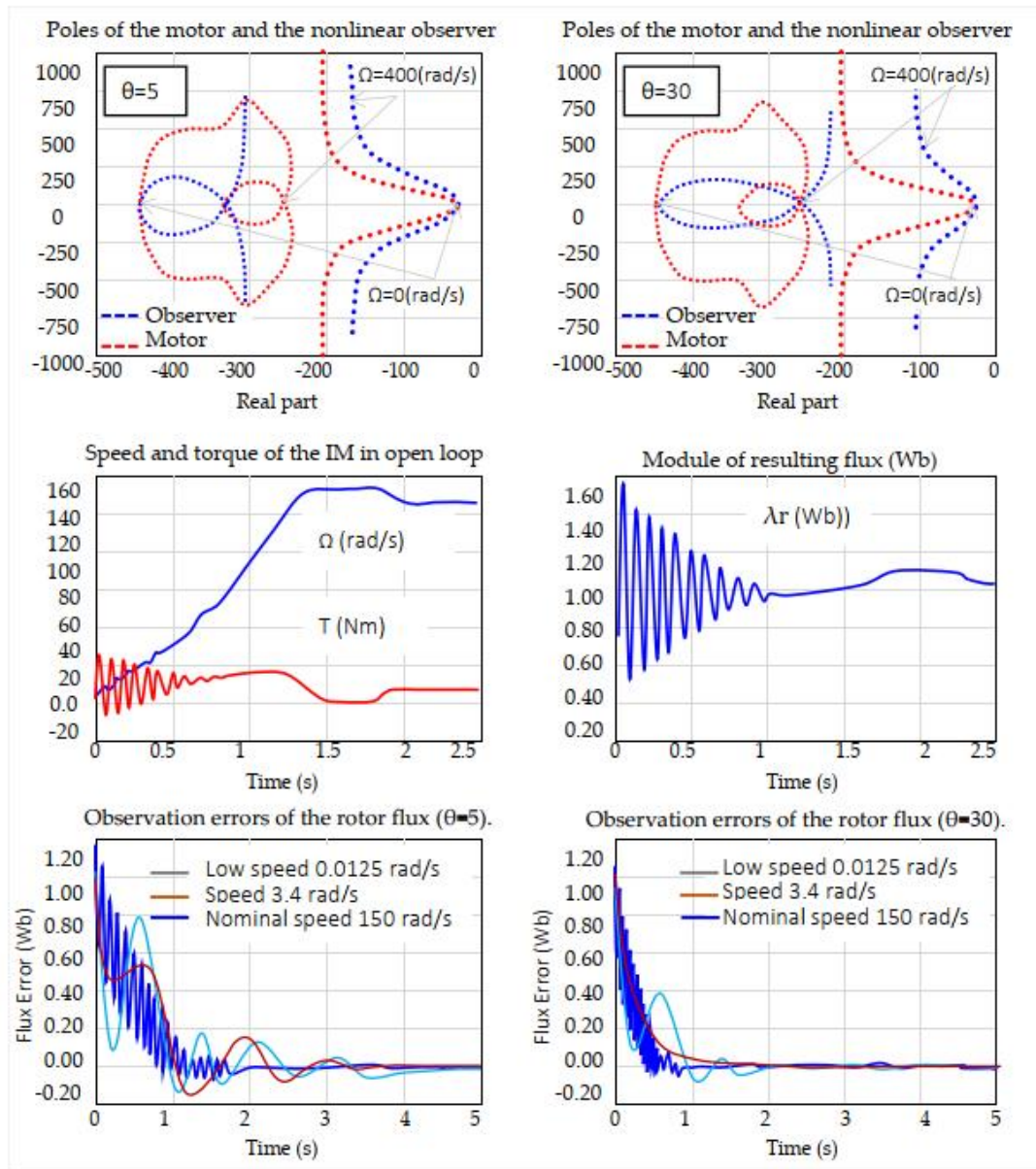


Figure 13. Rotor Flux Observation Errors Via the Proposed Neural Nonlinear Observer.

The convergence characteristics of the proposed nonlinear observer are significantly affected by the tuning parameter θ , which shapes the placement of its eigenvalues relative to those of the motor. As shown in the upper subplots of Figure 14, increasing θ shifts the observer poles (blue) to follow the motor poles (red) more closely across the operating range $\Omega = 0$ to 400 rad/s. For $\theta = 5$, the observer exhibits a moderately damped response, while $\theta = 30$ yields faster convergence with reduced transients. The lower subplots of Figure 14 illustrate the corresponding rotor flux observation errors under three representative speeds (0.0125, 3.4, and 150 rad/s). In both cases, the observer demonstrates strong convergence properties, but higher θ improves transient rejection and tracking precision, especially at low and nominal speeds.

To evaluate the sensitivity of the proposed nonlinear observer to rotor resistance variations, Figure 14 presents experimental results under three resistance values: 3.047 Ohm (nominal), 6.4571 Ohm (+50%), and 8.094 Ohm (+100%). The observer is initialized with rotor flux components set to $\lambda_{ra0} = \lambda_{rb0} = 0.5$ Wb.

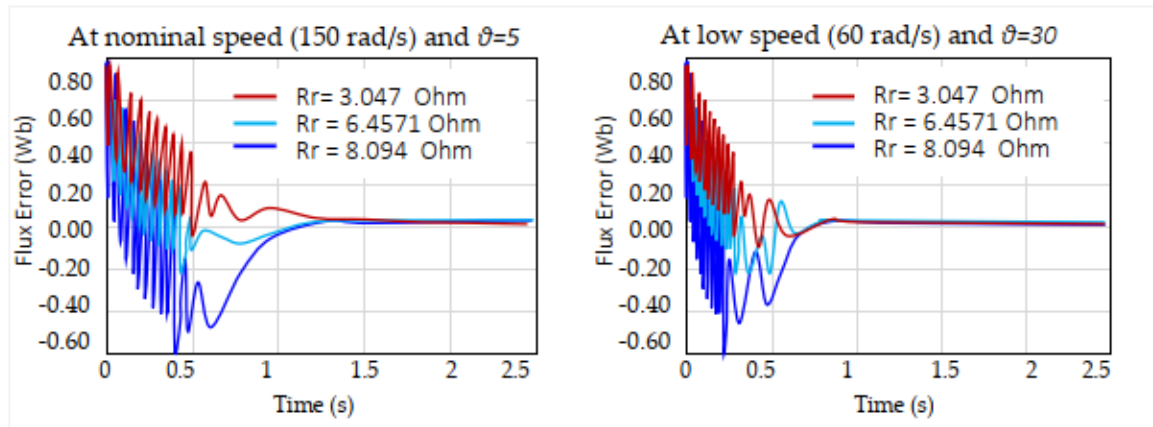


Figure 14. Observer Convergence and Sensitivity Analysis Under Rotor Resistance Variations.

Figure 14 demonstrates that, even with R_r increased from 3.047 Ohm (nominal) to 6.457 Ohm (+50%) and 8.094 Ohm (+100%), the nonlinear observer ensures rapid flux convergence. At both nominal (150 rad/s) and low speed (60 rad/s), transients settle within 0.6 s, showing strong disturbance rejection. A small steady-state offset remains (about 0.15 Wb at +50% and 0.30 Wb at +100%) indicating sensitivity to parameter mismatch. To address this, an RBFNN-based observer is introduced (Figure 14), incorporating an adaptive term that continuously compensates for rotor resistance errors and effectively eliminates steady-state flux bias under large uncertainties.

Figure 15 highlights the impact of the RBFNN augmentation on the observer's transient performance under large rotor resistance variations. In the baseline observer, the flux error exhibits pronounced oscillations and peak deviations nearing 0.8 Wb for a +100 % resistance change, with settling oscillations persisting beyond 0.6s. With the RBFNN-enhanced observer, the peak error is reduced by over 50 %, and oscillations are significantly damped—settling in under 0.4 s across all resistance cases. This clear reduction in transient amplitude and faster damping confirms that the online neural adaptation effectively compensates for parameter mismatches, yielding smoother and more responsive flux estimation throughout the speed range.

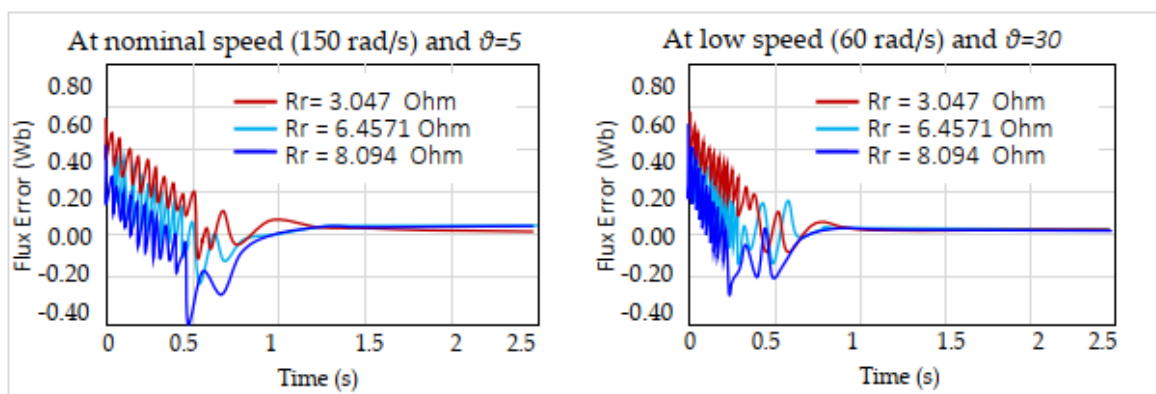


Figure 15. Impact of RBFNN Compensation on Flux Estimation Robustness with R_r Mismatch.

5.2. Comprehensive Evaluation of the Proposed Nonlinear Intelligent Passivity-Based Control

This section presents an overall assessment of the proposed nonlinear intelligent passivity-based control, which integrates a neural PBC controller, an RNN-based adaptive observer, and a sensorless speed estimator. The complete system is tested in closed-loop with the induction motor under various conditions, including load disturbances, speed reversals, and parameter variations. The results highlight the dynamic performance of key signals such as torque, rotor flux, speed, and control

voltages, confirming the robustness, adaptability, and precision of the proposed scheme. Figure 16 presents these responses under all test scenarios.

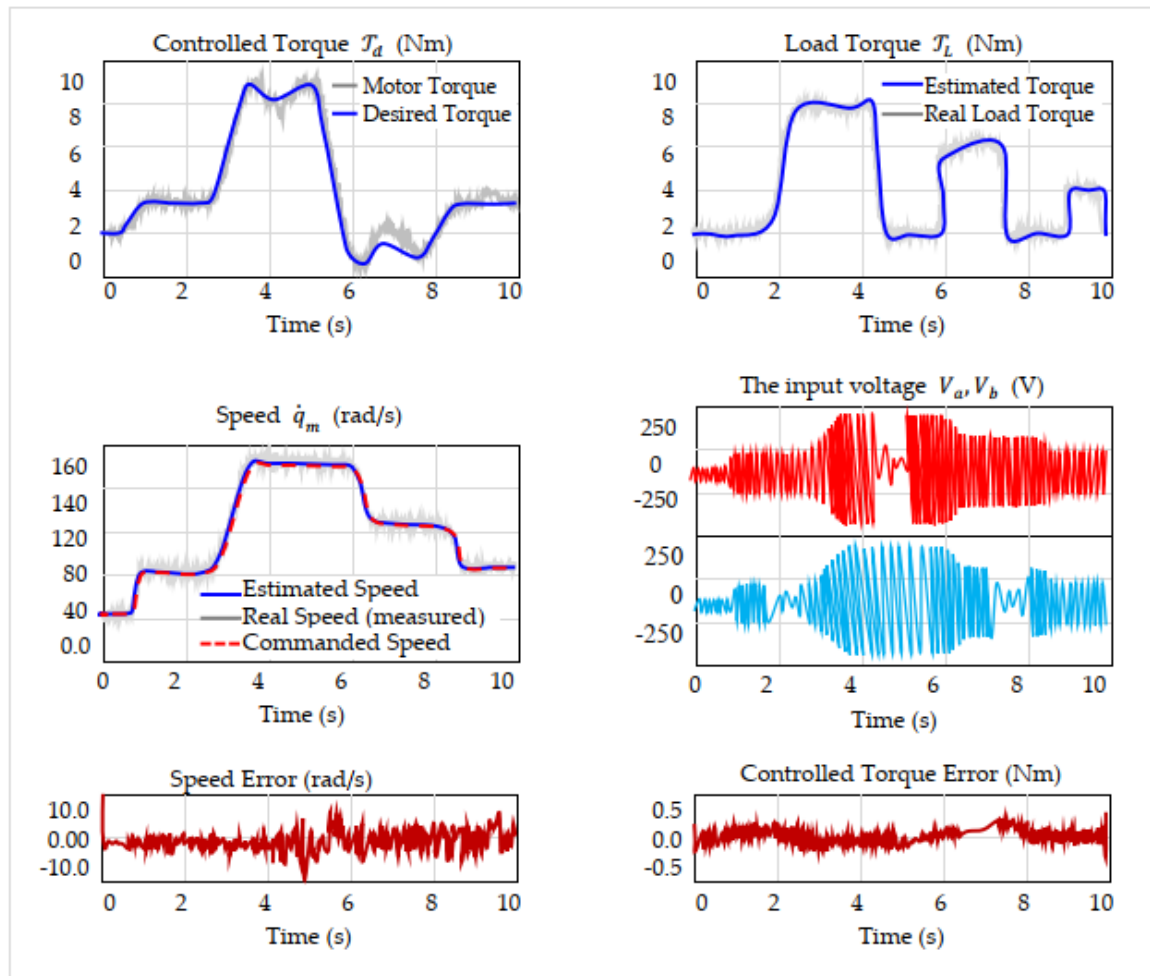


Figure 16. Performance Study of the Proposed Control Scheme under Torque/Flux Variation.

The performance of the proposed nonlinear intelligent PBC strategy is illustrated in Figure 16, which presents both control and estimation results under time-varying speed and load torque conditions. The top-left subplot shows that the controlled electromagnetic torque $T_d(t)$ tracks the reference smoothly with minimal steady-state error. The top-right plot confirms accurate load torque estimation $T_L(t)$, closely matching the timing and magnitude of applied disturbances. The middle-left subplot shows that both estimated and measured speeds follow the commanded reference $\dot{q}_m^{\text{ref}}(t)$ throughout acceleration, cruising, deceleration, and re-acceleration phases. The speed error, shown bottom-left, remains within ± 8 rad/s, demonstrating the precision of the RNN-based speed estimator. The control voltages V_a and V_b , in the middle-right plot, reach ± 250 V with consistent modulation behavior. The bottom-right plot shows torque error bounded within ± 0.5 Nm, confirming high tracking accuracy and effective disturbance rejection. These results validate the proposed architecture's capability for precise tracking, reliable estimation, and smooth control under sensorless operation.

Figure 17 was performed on a sensorless induction motor using the proposed PBC strategy for torque and flux control. The system faced load variations, noise, and rotor resistance mismatch, with all estimations handled observer-based. Control ran in real time with PWM injection, and key signals were precisely recorded for evaluation. The applied torque includes direction reversals to assess robustness.

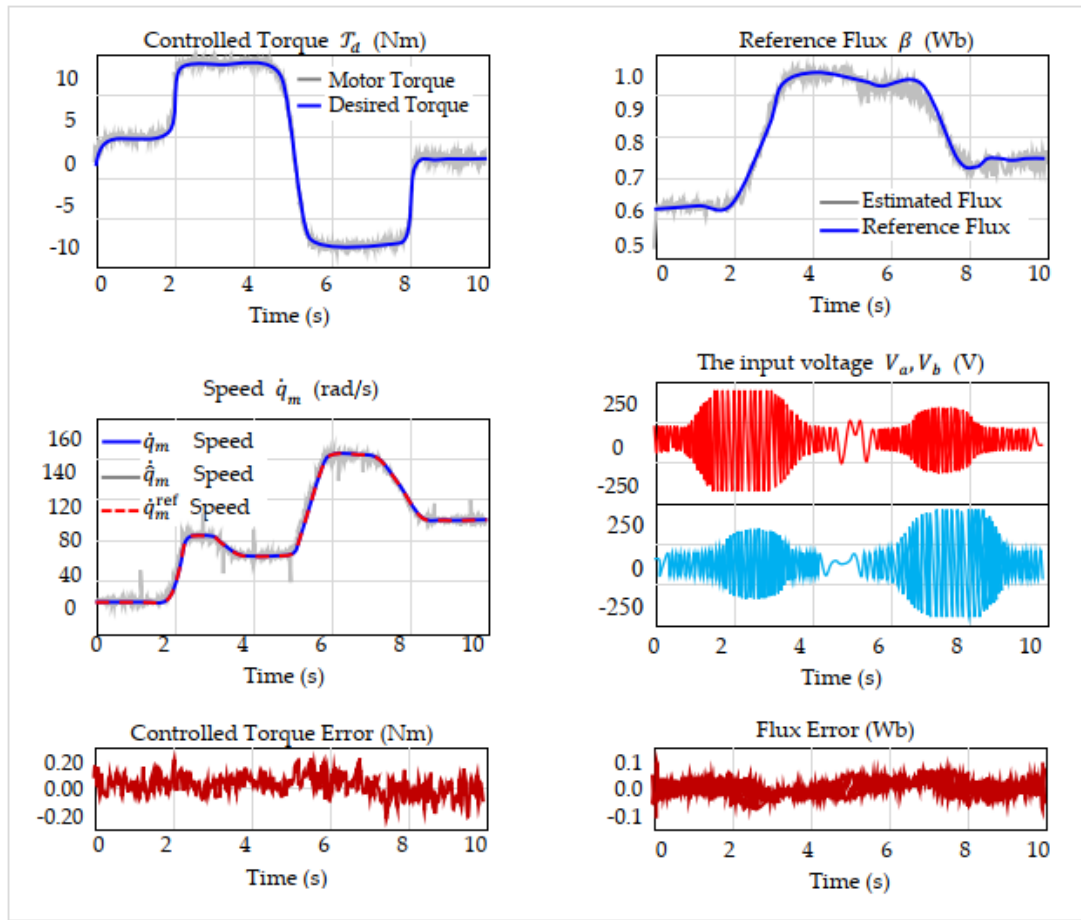


Figure 17. Performance Study of the Proposed Control Scheme under Torque Reversal.

The experimental results clearly confirm the effectiveness of the proposed nonlinear intelligent passivity-based control strategy. The motor tracks the torque reference with high accuracy, including during rapid transients and full torque reversal, maintaining a peak error within ± 0.2 Nm. The estimated load torque closely matches the real applied load, capturing both magnitude and timing under dynamic conditions. The speed response aligns well with the reference, and the speed estimation remains precise even during acceleration and sharp changes. The flux estimation error is well confined within ± 0.1 Wb, indicating strong observer performance. Input voltages exhibit clean modulation without saturation or chattering, demonstrating the quality of PWM-based voltage control. Despite load disturbances, parameter uncertainties, and full sensorless operation, the controller preserves stability, accurate tracking, and robustness, confirming its suitability for advanced industrial applications.

5.3. Comparative study and Performance Evaluation

To evaluate the performance of the proposed control, we use standard electrical engineering (EE) control metrics based on the response to torque amplitude or flux amplitude (i.e. $|\text{final} - \text{initial}|$):

- $t_{2\%}$ is the time at which the response signal has covered 2% of the ramp amplitude.
- $t_{95\%}$ is the time after which the response remains at less than 5% of the target value.
- $D_{\%}$ Peak overshoot relative to target (% of ramp amplitude).
- E_{ss} (steady-state error settling) the error once the $t_{95\%}$ has been reached.
- E_{foi} Error when reference reaches 50% of ramp.
- ΔT_{\max} Max torque deviation during speed ramp.

And in order to analyze the performance at global scope, we employ standard machine learning (ML) metrics: mean absolute error (MAE), symmetric mean absolute percentage error (SMAPE), and coefficient of determination (R^2). These metrics are defined as: $MAE = \frac{\sum_{t=1}^T |y_t - \hat{y}_t|}{T}$; $SMAPE = 100 \cdot \frac{\sum_{t=1}^T [|y_t - \hat{y}_t| / (|y_t| + |\hat{y}_t|)]}{T}$ and $R^2 = 1 - \frac{\sum_{t=1}^T [\bar{y}_t - \hat{y}_t]^2}{\sum_{t=1}^T [\bar{y}_t - y_t]^2}$ where y_t is the ground truth, \hat{y}_t is the predicted output of the model at time t , and T is the total experiment duration. \bar{y}_t denotes the mean of y_t . These metrics jointly characterize accuracy, relative error, and explained variance, enabling a comprehensive assessment of model fidelity.

To comprehensively assess the effectiveness of the proposed adaptive neuro-PBC sensorless control scheme under realistic and demanding operating conditions, two benchmark trajectories were designed:

1. **Pseudo-Static Benchmark:** The torque reference T_d ramps from 0 to 8 Nm in the first second and remains constant for the rest of the experiment. During the period from 1 s to 3 s and 5 s to 8.5 s, a flux reference β step of 8 Wb is applied with superimposed ripples to emulate disturbances. This test evaluates steady-state tracking and disturbance rejection.
2. **Dynamic Benchmark:** Here, the torque T_d follows a complex trajectory: it ramps from 0 to 8 Nm in the first second, holds steady until 5 s, then reverses sharply to approximately -8 Nm, followed by a final ramp back to 6 Nm by 10 s. Simultaneously, the flux reference β features a sequence of four step changes (switch on to 8 Wb in some time intervals), testing the controller's performance under rapid transients and flux variations.

Throughout the entire profile, small random perturbations are superimposed to simulate external disturbances, ensuring a challenging scenario for evaluating control robustness, responsiveness, and adaptation under dynamic and load-varying conditions. Figure 18 shows the profiles used during the experimental tests.

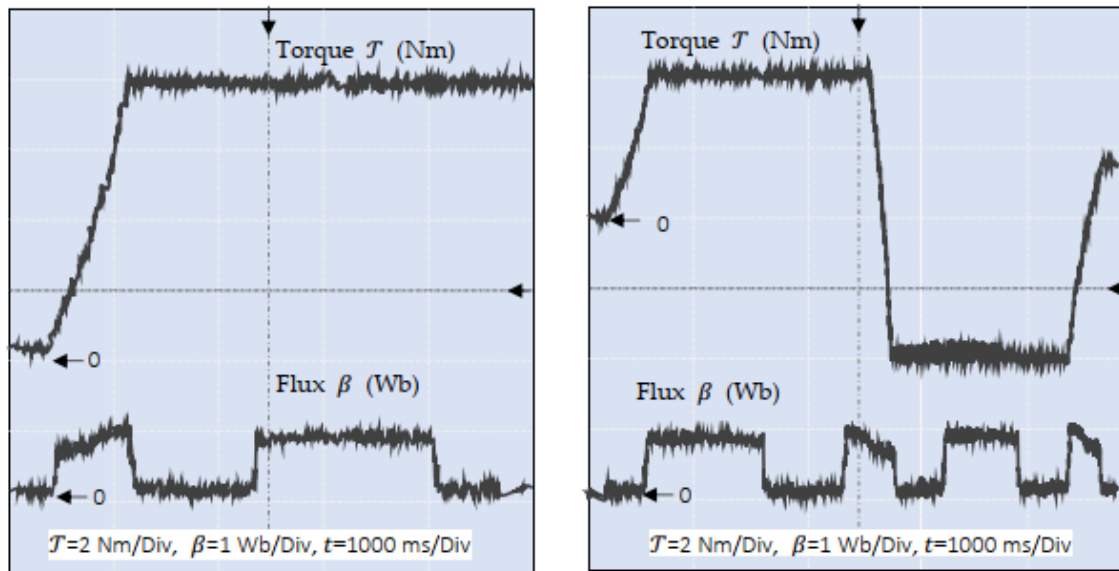


Figure 18. Realistic Operating Conditions Benchmark Profiles for Comparative Study.

Now we present a comparative evaluation of the proposed neural adaptive PBC controller against four advanced control strategies. These include:

- **CNO-PBC:** classical PBC control with a nonlinear observer,
- **CFE-PBC:** classical PBC control with a fuzzy enhanced MRAS estimator
- **CRBF-PBC:** classical PBC control with a RBF network enhanced observer, and a
- **DRNN-PBC:** deep learning adaptive PBC strategy with RNN network observer.

To evaluate steady-state accuracy and low-dynamic performance, all four control strategies were tested under the pseudo-static benchmark, where the torque reference \mathcal{T}_d ramps from 0 to 8 Nm within the first second and is then held constant. During the intervals (1s \rightarrow 3s) and (5s \rightarrow 8.5s), a flux reference step of 8 Wb is applied, with superimposed ripple signals to emulate realistic disturbance conditions. This scenario is designed to assess tracking precision, overshoot suppression, robustness to steady-state perturbations, and the quality of torque/flux estimation in a low-dynamic environment. A summary of comparative performance under this benchmark is presented in Table 3.

Table 3. Comparative Evaluation of Control Laws under Pseudo-Static Conditions.

| | Torque (\mathcal{T}) | | | | | | | Flux (β) | |
|----------|--------------------------|-------------------|--------------------|-------------------|-----------------|------------------|--|-------------------|--------------|
| | MAESMAPE (--)(%) | $t_{2\%}$ (ms) | $t_{95\%}$ (ms) | E_{fol} (Hz) | $D_{\%}$ (%) | E_{ss} (Hz) | $\Delta\mathcal{T}_{max}$ (% _{nom}) | MAE (--) | SMAPE (%) |
| CNO-PBC | 0.68 21.40 | 16 882 | 0.49 1.92 | 0.73 36.83 | | | | 0.47 44.33 | |
| CFE-PBC | 0.15 17.35 | 18 861 | 0.21 1.08 | 0.26 30.40 | | | | 0.22 35.95 | |
| CRBF-PBC | 0.09 15.13 | 15 845 | 0.13 0.71 | 0.14 27.25 | | | | 0.13 27.12 | |
| DRNN-PBC | 0.06 14.23 | 13 825 | 0.08 0.43 | 0.05 24.81 | | | | 0.10 21.92 | |

R^2 is \approx 0.998 for all the used controller.

The results in Table 3 confirm a progressive improvement in performance from classical observer-based control to advanced neural network-enhanced strategies. The baseline CNO-PBC controller shows the highest errors in both torque and flux estimation, with a torque MAE of 0.68 Nm and SMAPE of 21.40%, along with the highest overshoot (1.92%) and torque deviation (36.83%). Introducing fuzzy enhancement in CFE-PBC improves tracking accuracy significantly, reducing torque MAE to 0.15 Nm and improving flux SMAPE by nearly 8.4 percentage points. Further enhancement with radial basis function networks (CRBF-PBC) yields even lower errors (torque MAE = 0.09 Nm, SMAPE = 15.13%), shorter rise and settling times, and better rejection of steady-state disturbances. The best overall performance is achieved by the proposed DRNN-PBC controller, which achieves the lowest torque and flux estimation errors (MAE = 0.06 Nm and 0.10 Wb), fastest response ($t_{2\%}$ = 13 ms), and minimal steady-state error (0.05 Hz). It also maintains the smallest overshoot (0.43%) and torque deviation (24.81% of nominal), confirming its robustness, adaptability, and estimation fidelity under pseudo-static conditions. All controllers maintain high model fit with $R^2\approx$ 0.998.

To evaluate the controllers under dynamic and nonlinear conditions, a challenging dynamic benchmark was executed. The torque reference \mathcal{T}_d ramps from 0 to 8 Nm within the first second, holds steady until 5 s, then reverses sharply to approximately -8 Nm before ramping back to 6 Nm by 10 s. Simultaneously, the flux reference β undergoes multiple step changes up to 8 Wb, simulating rapid flux variations. Throughout the test, small random perturbations emulate external disturbances, creating a demanding scenario to assess the controllers’ robustness, responsiveness, and adaptation under rapid transients and load variations. This benchmark emphasizes each control strategy’s ability to maintain estimation accuracy and transient response quality under simultaneous nonlinear torque and flux variations. The results of this dynamic benchmark, including the speed reversal and load variation, are presented in Table 4.

Table 4. Comparative Evaluation of Control Laws under Pseudo-Static Conditions.

| | Torque (\mathcal{T}) | | | | | | | Flux (β) | |
|--|--------------------------|-------------------|--------------------|-------------------|-----------------|------------------|--|------------------|--------------|
| | MAESMAPE (--)(%) | $t_{2\%}$ (ms) | $t_{95\%}$ (ms) | E_{fol} (Hz) | $D_{\%}$ (%) | E_{ss} (Hz) | $\Delta\mathcal{T}_{max}$ (% _{nom}) | MAE (--) | SMAPE (%) |

| | | | |
|----------|------------|------------------------------|------------|
| CNO-PBC | 1.58 26.12 | 18 1120 1.23 4.02 1.10 42.52 | 0.85 53.51 |
| CFE-PBC | 0.35 21.05 | 21 1005 0.65 2.31 0.41 36.51 | 0.41 39.54 |
| CRBF-PBC | 0.23 19.31 | 20 995 0.33 1.63 0.28 33.12 | 0.15 29.95 |
| DRNN-PBC | 0.18 18.52 | 19 985 0.27 1.13 0.21 30.85 | 0.21 28.63 |

R^2 is ≈ 0.998 for all the used controller.

Under dynamic torque and flux variation conditions involving rapid torque ramps, reversals, and flux step changes, the performance disparities between control strategies become more pronounced. The classical CNO-PBC controller exhibits relatively high tracking errors and slower transient response, with a torque disturbance index ($D_{\%}$) of 4.02% and steady-state frequency error (E_{ss}) of 1.10 Hz. Conversely, the adaptive neuro-based controllers significantly improve tracking precision and responsiveness. Notably, the DRNN-PBC achieves the best overall performance, attaining the lowest torque MAE (0.18 Nm) and SMAPE (18.5%), alongside minimized maximum transient torque deviation ($\Delta T_{\max} = 30.85\%$ nominal). Intermediate controllers, such as CRBF-PBC and CFE-PBC, also provide substantial enhancements in flux and torque estimation accuracy. These findings confirm the superior adaptability and robustness of proposed neural adaptive PBC schemes in handling rapid and complex torque and flux dynamics, ensuring stable and precise control during challenging real-world transient conditions.

6. Conclusions

This work introduced a learning-driven intelligent passivity control strategy for induction motors, integrating a nonlinear state observer with an adaptive recurrent neural framework. The proposed observer significantly enhances the estimation accuracy of key motor variables such as torque, speed, and rotor flux, while simplifying the tuning process by requiring only a single gain adjustment over a wide speed range. Unlike traditional control schemes, this approach ensures rapid convergence and robust performance even under low-torque (i.e. low-speed) and highly dynamic operating conditions. Extensive simulation studies validated the global stability of the closed-loop system using Lyapunov-based analysis and demonstrated the observer’s robustness against various torque disturbances and flux variations. The integration of neural adaptive mechanisms into the passivity-based control framework yielded superior transient response, disturbance rejection, and steady-state tracking performance. Future work will focus on real-time experimental validation and extending the methodology to other induction motor topologies, including doubly-fed and coaxial configurations, to establish the broad applicability and scalability of this intelligent control approach in advanced industrial drive systems.

Author Contributions: B.B: Conceptualization, data curation, formal analysis, investigation, methodology, project administration, resources, software, visualization, writing - original draft. K.H: Conceptualization, funding acquisition, investigation, supervision, resources, writing-original draft. A.K and V.K: Project administration, validation, writing-review & editing. K.H and M.R: Data curation, methodology, validation, writing - review & editing

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

| | | | |
|-------|--|------|-----------------------------|
| IMs | Induction Motors | PI | Proportional-Integral |
| PBC | Passivity-Based Control | TSM | Takagi-Sugeno Model |
| AI | Artificial Intelligence | ELM | Euler–Lagrange Modeling |
| ANFIS | Adaptive Neuro-Fuzzy Inference Systems | BPTT | Back-Propagate Through Time |
| RBF | Radial Basis Function | PWM | Pulse Width Modulation |
| MMF | Magneto-Motive Force | EE | Electrical Engineering |
| RNN | Recurrent Neural Network | ZMP | Zero Moment Point |
| RBFNN | Radial Basis Function Neural Networks | ML | Machine Learning |
| MRAS | Model Reference Adaptive System | MAE | Mean Absolute Error |
| RTI | Real Time Interface | DSP | Digital Signal Processor |
| RTW | Real Time Workshop | | |

References

1. Carlo Cecati, Position Control of the Induction Motor Using a Passivity-Based Controller, *IEEE Transactions On Industry Applications*, Vol. 36, No. 5, **2000** <https://doi.org/10.1109/IAS.1998.729765>.

2. Manuel A. Duarte-Mermoud, Induction motor Control Based On Adaptive Passivity, *Asian Journal of Control*, Vol. 14, No. 1, pp. 67–84, January **2012** <https://doi.org/10.1002/asjc.260>.

3. Juan Carlos Travieso-Torres, Robust Combined Adaptive Passivity-Based Control for Induction Motors, *Machines* **2024**, 12, 272. <https://doi.org/10.3390/machines12040272>.

4. A. Mansouri, “Passivity Based Control with Robust Observer for Induction Motor” IEEE International Symposium on Industrial Electronics, May 4-7, **2004**, <https://doi.org/10.1109/ISIE.2004.1572013>.

5. R., S. Marino, Adaptive input-output linearizing control of induction motors. *IEEE Transactions on Automatic Control* 38 (2), pp208-221 (**1993**). <https://doi.org/10.1109/9.250510>.

6. M. Chenafa, Global Stability of the Linearizing Control with a New Robust Non linear Observer of Induction Motor, *Int. J. Appl. Math. Comput. Sci.*, **2005**, Vol. 15, No. 2, 235–243.

7. G. Espinosa-Pérez, An output feedback globally stable controller for induction motors. *IEEE Transactions on Automatic Control* 40 (1), pp 138-143. **1995**. <https://doi.org/10.1109/9.362883>

8. Belkacem Bekhiti, “Sensorless Control of a Feedback Linearized Induction Motor via the Adaptive Luenberger Observer”, 1st International Conference on Applied Automation And Industrial Diagnosis (ICAAID 2015), Djelfa 29-30 March **2015**. Algeria.

9. Belkacem Bekhiti, “Adaptive Sensorless Control of a Feedback Linearized Induction Motor via the Extended Kalman Filter”, 2nd International Conference on Power Electronics and Their Applications (ICPEA 2015), Djelfa 29-30 March **2015**. Algeria.

10. Belkacem Bekhiti, “On Hyper-Stability Theory Based Multivariable Nonlinear Adaptive Control: Experimental Validation on Induction Motors,” *IET Electric Power Applications* Volume 19, Issue 1 (**2025**), <https://doi.org/10.1049/elp2.70035>.

11. J.P. Gauthier, A simple observer for nonlinear systems Application to bioreactors. *IEEE Transactions on Automatic Control* Vol. 37, Iss. 6, **1992**. <https://doi.org/10.1109/9.256352>.

12. K.K. Busawon, An observer for a class of disturbance driven nonlinear systems. *Applied Mathematics Letters* Volume 11, Issue 6, November **1998**, Pages 109-113

13. A. Mansouri, Powerful nonlinear observer associated with the field-oriented control of the induction motor. *Int. J. Appl. Math. Comput. Sci.*, Vol. 14, No. 2, pp. 209–220 (**2004**).

14. Francisco Beltran-Carbajal, Adaptive neuronal induction motor control with an 84-pulse voltage source converter, *Asian J Control*. **2020**;1–14. <https://doi.org/10.1002/asjc.2322>

15. Mahmood Moghadasian, Sensorless Speed Control of Induction Motors using Adaptive Neural-Fuzzy Inference System, **2011** IEEE/ASME Conference on Advanced Intelligent Mechatronics (AIM2011) <https://doi.org/10.1109/aim.2011.6027023>
16. K. Naga Sujatha, K. Vaisakh, "Implementation of Adaptive Neuro Fuzzy Inference System in Speed Control of Induction Motor Drives," Journal of Intelligent Learning Systems and Applications Vol.2 No.2, June 1, **2010** <https://doi.org/10.4236/jilsa.2010.22014>.
17. Pavel Brandstetter, Martin Kuchar, Sensorless control of variable speed induction motor drive using RBF neural network, Journal of Applied Logic 24 (**2017**) 97–108 <http://dx.doi.org/10.1016/j.jal.2016.11.017>
18. Hakan Acikgoz, Real-time adaptive speed control of vector-controlled induction motor drive based on online-trained Type-2 Fuzzy Neural Network Controller, Int Trans Electr Energ Syst. **2020**;e12678. <https://doi.org/10.1002/2050-7038.12678>
19. Shoeb Hussain, Adaptive Neural Type II Fuzzy Logic-Based Speed Control of Induction Motor Drive, Advances in Intelligent Systems and Computing, January **2018** https://doi.org/10.1007/978-981-10-7386-1_7
20. Nassira Medjadji, Quality Improvement of Speed Estimation in Induction Motor Drive by Using Fuzzy-MRAS Observer with Experimental Investigation," IREACO. Vol 11, No 1 (**2018**) <https://doi.org/10.15866/ireaco.v11i1.14425>.
21. Govindharaj I, Sensorless vector-controlled induction motor drives: Boosting performance with Adaptive Neuro-Fuzzy Inference System integrated augmented Model Reference Adaptive System, MethodsX 13 (**2024**) 102992. <https://doi.org/10.1016/j.mex.2024.102992>
22. E. Parimalasundar, Artificial Neural Network-Based Experimental Investigations for Sliding Mode Control of an Induction Motor in Power Steering Applications, International Journal of Intelligent Systems Volume **2023**, Article ID 9381915, 14 pages <https://doi.org/10.1155/2023/9381915>
23. Zineb Mekrini, "Fuzzy Logic Application for Intelligent Control of An Asynchronous Machine," Indonesian Journal of Electrical Engineering and Computer Science Vol. 7, No. 1, **2017**, pp. 61-70 <https://doi.org/10.11591/ijeecs.v7.i1>.
24. Qiming Sun, "Adaptive Robust Control of an Industrial Motor-Driven Stage with Disturbance Rejection Ability Based on Multidimensional Taylor Network," Appl. Sci. **2023**, 13(22), 12231; <https://doi.org/10.3390/app132212231>.
25. Ibomoieye Domor Mienye, Theo G. Swart, Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications, Information **2024**, 15(9), 517; <https://doi.org/10.3390/info15090517>
26. Xinyi Wu, Optimizing Recurrent Neural Networks: A Study on Gradient Normalization of Weights for Enhanced Training Efficiency, Appl. Sci. **2024**, 14(15), 6578; <https://doi.org/10.3390/app14156578>
27. Werner Leonhard, "Control of electrical drives", Springer Verlag, 2001.
28. G. Espinosa Perez, "Torque and flux tracking of induction motors, Int. Jour of Robust and Nonlinear Control, vol. 7 N. 1, pp 1-9 **1997**. [https://doi.org/10.1002/\(SICI\)1099-1239\(199701\)7:1<1::AID-RNC194>3.0.CO;2-2](https://doi.org/10.1002/(SICI)1099-1239(199701)7:1<1::AID-RNC194>3.0.CO;2-2)
29. Romeo Ortega, On Speed Control of Induction Motors, Automatica volume 32, issue 3 (**1996**), [https://doi.org/10.1016/0005-1098\(95\)00171-9](https://doi.org/10.1016/0005-1098(95)00171-9)
30. Mohammad Hosein Sabzalian, A Neural Controller for Induction Motors: Fractional-Order Stability Analysis and Online Learning Algorithm, Mathematics **2022**, 10(6), 1003; <https://doi.org/10.3390/math10061003>.
31. Narongrit Pimkumwong, Online Speed Estimation Using Artificial Neural Network for Speed Sensorless Direct Torque Control of Induction Motor based on Constant V/F Control Technique, Energies **2018**, 11(8), 2176; <https://doi.org/10.3390/en11082176>
32. George Verghese, Observers for Flux Estimation in Induction Machines, IEEE Transactions on Industrial Electronics, Vol 35, N 1, **1988**; <https://doi.org/10.1109/41.3067>
33. A. De Luca, G. Ulivi, The Design of Linearizing Outputs for Induction Motors, IFAC Proceedings Volumes Volume 22, Issue 3, June **1989**, Pages 363-367; [https://doi.org/10.1016/S1474-6670\(17\)53662-9](https://doi.org/10.1016/S1474-6670(17)53662-9)
34. G.O. Garcia, An efficient controller for an adjustable speed induction motor drive, IEEE Transactions on Industrial Electronics (Volume: 41, Issue: 5, October 1994); <https://doi.org/10.1109/41.315272>

35. Belkacem Bekhiti, "Advanced nonlinear control and state observation in robotics," *Scholars' Press* **2017**, ISBN: 978-3-330-65073-2
36. T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, **1985**, 15: 116–132. <https://doi.org/10.1109/TSMC.1985.6313399>.
37. Elhassen Benfriha, Nonlinear adaptive observer for sensorless passive control of permanent magnet synchronous motor, *Journal of King Saud University - Engineering Sciences* Vol. 32, N 8, **2020**, 510-517 <https://doi.org/10.1016/j.jksues.2019.06.003>.
38. Youcef Belkhier, Experimental analysis of passivity-based control theory for permanent magnet synchronous motor drive fed by grid power, *IET Control Theory Appl.* **2023** 18, Issue 4 pp. 495-510. <https://doi.org/10.1049/cth2.12574>

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.