International Journal of Robotics and Control Systems
Vol. 5, No. 3, 2025, pp. 1828-1849
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Artificial Intelligence-Enhanced Sensorless Vector Control of Induction Motors Using Adaptive Neuro-Fuzzy Systems: Experimental Validation and Benchmark Analysis

Belkacem Bekhiti [a,1], George F. Fragulis [b,2], Kamel Hariche [a,3], Abdel-Nasser Sharkawy [c,d,4,*]

[a] Institute of Aeronautics and Space Studies, (IASS), University of Blida 1, BP 270 Blida (09000), Algeria
[b] Dept. of Electrical and Computer Engineering, ZEP Campus, University of Western Macedonia, Kozani, 50100, Greece
[c] Mechanical Engineering Department, Faculty of Engineering, South Valley University, Qena 83523, Egypt
[d] Mechanical Engineering Department, College of Engineering, Fahad Bin Sultan University, Tabuk 47721, Saudi Arabia
[1] belkacem1988@hotmail.co.uk; [2] gfragulis@uowm.gr; [3] khariche@yahoo.com; [4] abdelnassersharkawy@eng.svu.edu.eg
* Corresponding Author

ARTICLE INFO

Article history
Received May 22, 2025
Revised July 15, 2025
Accepted August 07, 2025

Keywords
Induction Motor;
Adaptive Neural-Fuzzy Control;
Sensorless MRAS Control;
Field Oriented Control;
dSPACE (DS1104)

ABSTRACT

This study addresses the limitations of traditional Model Reference Adaptive Systems (MRAS) in sensorless induction motor (IM) control, particularly the degraded performance at low speeds and under dynamic load conditions. The main objective is to enhance speed and torque estimation accuracy by replacing the classical proportional-integral (PI) adaptation mechanism with an adaptive neuro-fuzzy architecture. The research contribution lies in developing and experimentally validating two intelligent adaptation schemes: one based on fuzzy logic and another combining fuzzy inference with a recurrent neural network (RNN) within a sensorless field-oriented control (FOC) framework. The proposed system integrates a fuzzy logic-based estimator and an RNN-driven torque predictor to improve tracking precision and robustness. Real-time implementation was carried out on a 1.1 kilowatt, 1430 revolutions per minute induction motor using a dSPACE DS1104 platform. Comparative experiments were conducted under two challenging benchmark profiles that include load disturbances, parameter mismatches, and full-speed reversals. Results showed that the hybrid neuro-fuzzy controller reduced the steady-state speed error by 91 %, from 0.65 rad/s to 0.08 rad/s, and improved torque estimation accuracy by 42%, reducing SMAPE from 45.2 % to 26.3 %, compared to the PI-based MRAS. It also outperformed the standalone fuzzy and neural MRAS controllers in rise time, tracking error, overshoot suppression, and adaptation quality. These findings confirm that the proposed method provides improved estimation fidelity, enhanced control robustness, and reliable sensorless operation suitable for real-time industrial applications. The study concludes that the integration of neuro-fuzzy intelligence into MRAS-based control structures offers a technically effective and scalable solution for advanced IM drives.

1. Introduction

Induction motors (IMs) are widely used in industrial applications due to their reliability, simplicity, and cost-effectiveness, making them a preferred choice for variable-speed electric drives

in manufacturing and transportation systems [1], [2]. To achieve high-performance control, techniques such as Field-Oriented Control (FOC) have been developed, enabling decoupled torque and flux regulation similar to that of DC motors by aligning the reference frame with rotor flux [3]. This has allowed IMs to meet the demands of precision, dynamics, and stability required in advanced drive systems [4]. A key challenge in such systems lies in the accurate estimation of rotor speed, especially in sensorless configurations, where mechanical speed sensors are eliminated to reduce cost, complexity, and maintenance requirements [5]. In this context, sensorless estimation techniques have become crucial, among which the Model Reference Adaptive System (MRAS) has gained widespread popularity due to its simple structure and suitability for real-time implementation [6], [7]. The Model Reference Adaptive System (MRAS) estimates rotor speed by evaluating the difference between two model outputs: a reference model that is independent of the rotor speed [8], and an adaptive model that incorporates the estimated speed as a parameter [9]. This discrepancy, often referred to as the adaptation error, is used to adjust the speed estimate through a dedicated adaptation mechanism. In most implementations, this mechanism is realized using a proportional-integral (PI) controller due to its simplicity and ease of tuning [10]. Despite the widespread use of PI-based MRAS structures, they suffer from three critical limitations:

(i) Degraded performance at low speeds due to poor signal-to-noise ratios,

(ii) Sensitivity to parameter variations such as rotor resistance drift, and

(iii) Inadequate handling of nonlinearities and un-modeled disturbances [11]-[15].

These limitations motivate the integration of intelligent techniques for improved robustness and adaptability [16], [17]. In response to these limitations, fuzzy logic controllers (FLCs) have been introduced into MRAS schemes to improve adaptation accuracy without relying on precise mathematical models [18]. Fuzzy systems are especially useful when domain expertise can be encoded into rule-based structures and have shown improved performance under parameter uncertainty [19]. However, their static rule base and lack of learning mechanisms can hinder adaptability in dynamic environments [20].

In contrast, artificial neural networks (ANNs) offer strong learning capabilities and function approximation properties, allowing them to capture complex, nonlinear system dynamics [21], [22]. Neural networks, including feedforward and recurrent architectures, have been successfully employed in various control applications [23]-[26], and more recently, in sensorless induction motor drives [27], [28]. However, purely neural approaches may require extensive training data, and their black-box nature often reduces transparency and interpretability [29], [30].

To exploit the complementary strengths of both techniques, hybrid neuro-fuzzy systems have emerged as a powerful solution, combining the reasoning capability of fuzzy inference with the learning and adaptation capability of neural networks [31], [32]. In particular, recurrent neural networks (RNNs), known for their internal memory and temporal modeling abilities, have shown promise for dynamic control and torque estimation tasks [33]-[36]. Yet, the integration of RNNs within MRAS frameworks remains limited in the literature, especially in practical sensorless induction motor applications. Recent works on intelligent MRAS design (e.g., [37]-[40]) have reported improvements over classical methods, but often lack comprehensive experimental validation, robust benchmarking, or critical analysis against dynamic operating conditions such as speed reversal, load disturbances, and parameter mismatches. Moreover, the theoretical motivation for combining fuzzy and neural methods is often underexplored, leaving a gap in fully understanding the strategic benefits of hybridization.

This paper is motivated by three goals. First, to overcome the known limitations of PI-based MRAS systems by integrating adaptive neuro-intelligent techniques into the speed estimation loop. Second, to validate the proposed architectures under realistic industrial scenarios involving fast dynamics and parameter variation. Third, to investigate the synergistic effect of combining fuzzy logic and neural networks within the MRAS framework. To this end, we propose and evaluate three progressively refined sensorless MRAS schemes:

- A fuzzy-logic-based estimator,
- A neural-network-based torque predictor, and.
- A hybrid neuro-fuzzy MRAS architecture, which serves as the primary contribution

The design is implemented in real time on a 1.1 kW induction motor using a dSPACE DS1104 platform, and performance is analyzed using both control metrics (e.g., rise time, overshoot, steady-state error) and machine learning metrics (e.g., MAE, SMAPE, R^2). The research contribution is summarized as follows:

- A novel hybrid neuro-fuzzy MRAS estimator is introduced, combining fuzzy logic adaptation with an RNN-based torque prediction model.
- The individual impact of fuzzy/neural components is quantified through comparative testing.
- The method is evaluated against PI-MRAS, fuzzy MRAS, and neural MRAS baselines.
- Experimental results show a 91% reduction in speed error and a 42% gain in torque accuracy, confirming the hybrid controller's effectiveness under real-time sensorless conditions.

The remainder of this paper is structured as follows. Section 2 outlines the dynamic model of the induction motor and the field oriented vector control principle. Section 3 covers the fundamentals of the MRAS observer. Section 4 presents the first proposed adaptive fuzzy control scheme, while Section 5 introduces the second proposed framework based on neural networks and learning systems. Section 6 details the experimental setup and test conditions, followed by a comprehensive performance analysis comparing the proposed and conventional methods. Finally, Section 7 concludes the paper and suggests directions for future research.

2. Induction Motor Model and Vector Control

The mathematical model (1)-(4) of the IM in the d-q coordinates, formulated within a rotor flux-oriented reference frame (i.e. rotates at angular speed ω), can be expressed as [15]:

$$\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} d/dt & -\omega \\ \omega & d/dt \end{bmatrix} \begin{bmatrix} \psi_{sd} \\ \psi_{sq} \end{bmatrix}; \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_r & 0 \\ 0 & R_r \end{bmatrix} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \begin{bmatrix} d/dt & -\omega_{sl} \\ \omega_{sl} & d/dt \end{bmatrix} \begin{bmatrix} \psi_{rd} \\ \psi_{rq} \end{bmatrix} \tag{1}$$

where $\omega_{sl} = \omega - \omega_r$ and the expressions for stator and rotor flux linkages are defined by [16]:

$$\begin{bmatrix} \psi_{sd} \\ \psi_{sq} \end{bmatrix} = \begin{bmatrix} L_s & 0 \\ 0 & L_s \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} L_m & 0 \\ 0 & L_m \end{bmatrix} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix}; \quad \begin{bmatrix} \psi_{rd} \\ \psi_{rq} \end{bmatrix} = \begin{bmatrix} L_r & 0 \\ 0 & L_r \end{bmatrix} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \begin{bmatrix} L_m & 0 \\ 0 & L_m \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} \tag{2}$$

The state-space model of the induction motor, using stator current components (1) and rotor flux linkages (2) as state variables, is expressed as [15]:

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} -\left(\frac{R_s}{\sigma L_s} + \frac{L_m^2}{\sigma L_s L_r T_r}\right) \boldsymbol{I} - \omega \boldsymbol{J} & \frac{L_m}{\sigma L_s L_r T_r} \boldsymbol{I} - \frac{L_m}{\sigma L_s L_r} \omega_r \boldsymbol{J} \\ \frac{L_m}{T_r} \boldsymbol{I} & -\frac{1}{T_r} \boldsymbol{I} + (\omega_r - \omega) \boldsymbol{J} \end{bmatrix} \mathbf{x}(t) + \frac{1}{\sigma L_s} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u}(t) \tag{3}$$

where L_s, L_r, R_s, R_r, L_m are the motor parameters and $\omega_r = p\omega_m$, $\mathbf{u}(t) = \begin{bmatrix} v_{sd} & v_{sq} \end{bmatrix}^\top$ is the stator input voltage, $\mathbf{x}(t) = \begin{bmatrix} i_{sd} & i_{sq} & \psi_{rd} & \psi_{rq} \end{bmatrix}^\top$ is the state vector composed of currents-fluxes and

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 2}; \boldsymbol{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}; \sigma = 1 - \frac{L_m^2}{L_s L_r}; T_r = \frac{R_r}{L_r}$$

The electromagnetic torque and the rotor speed are given by [22]:

$$\mathcal{T}_{em}(t) = pL_m\left(\psi_{rd} i_{sq} - \psi_{rq} i_{sd}\right)/L_r; \dot{\omega}_m(t) = \left[\left(\mathcal{T}_{em}(t) - \mathcal{T}_L(t)\right) - R_m \omega_m(t)\right]/D_m \tag{4}$$

Field-oriented control enables independent control of torque and flux by aligning the rotor flux with the d-axis of the rotating dq reference frame, mimicking the behavior of a DC motor [15], [20]:

$$\psi_{rq} = 0; \psi_{rd} = L_m i_{sd} \tag{5}$$

The slip frequency $\omega_{sl} = \omega - \omega_r$ can be calculated from the reference values of the stator current components represented in the rotor flux oriented reference frame [11]. The relationship for electromagnetic torque can be formulated as:

$$\mathcal{T}_{em} = (pL_m/L_r)\psi_{rd} i_{sq} = K_t i_{sq} \tag{6}$$

where K_t is the torque constant given by:

$$K_t = (pL_m/L_r)\psi_{rd} \tag{7}$$

For the field oriented control, the PI gains of the speed adaptive scheme are tuned according to [25]: $K_p = 3$, $K_i = 1800$. For a detailed explanation of FOC principles, the reader is referred to [15], [16].

3. Model Reference Adaptive Systems (MRAS)

Rotor speed $\hat{\omega}$ can be estimated using a model reference adaptive system structure composed of two estimators: a *reference model* and an *adaptive model*. Both models independently compute the rotor flux linkage components ψ_{rd} and ψ_{rq} in the stationary dq 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 [11], [12]. 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:

$$\frac{d\psi_{rd}^{\text{ref}}}{dt} = \frac{L_r}{L_m}\left[v_{sd} - R_s i_{sd} - \sigma L_s \frac{di_{sd}}{dt}\right]; \frac{d\psi_{rq}^{\text{ref}}}{dt} = \frac{L_r}{L_m}\left[v_{sq} - R_s i_{sq} - \sigma L_s \frac{di_{sq}}{dt}\right] \tag{8}$$

In contrast, the adaptive model, formulated from the rotor voltage equations, explicitly depends on the rotor flux components and the estimated speed $\hat{\omega}$ [10]. It is expressed as:

$$\frac{d\hat{\psi}_{rd}}{dt} = \left[\frac{L_m}{T_r} i_{sd} - \frac{1}{T_r}\hat{\psi}_{rd} + \hat{\omega}_{sl}\hat{\psi}_{rq}\right]; \frac{d\hat{\psi}_{rq}}{dt} = \left[\frac{L_m}{T_r} i_{sq} - \frac{1}{T_r}\hat{\psi}_{rq} - \hat{\omega}_{sl}\hat{\psi}_{rd}\right] \tag{9}$$

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 [6], [15] to guarantee global asymptotic stability of the closed-loop estimation system, the adaptive law is given by:

$$\hat{\omega}_r = K_p \varepsilon(t) + K_i \int \varepsilon(t) dt \tag{10}$$

with $\varepsilon(t) = \left[\hat{\psi}_{rd}(t)\psi_{rq}^{\text{ref}}(t) - \hat{\psi}_{rq}(t)\psi_{rd}^{\text{ref}}(t)\right]$ is the error between the real and estimated rotor fluxes in the (d,q) frame. The controller gains were tuned using the method described in [7], [11].

4. The First Proposed Fuzzy Logic Controller (FLC)

Fuzzy logic is useful for problems that are difficult to model mathematically due to limited data, incomplete information, or system complexity [13]. Such systems can be improved by adding new rules to enhance performance or introduce features. In conventional MRAS speed observers, a PI controller is typically used in the adaptation mechanism to estimate speed by minimizing the

error between the adaptive and reference models [17]. In the proposed structure, this PI controller is replaced by a fuzzy logic system, as shown in Fig. 1. The overall fuzzy controller is presented in Fig. 2, and its internal structure in Fig. 3. The process starts with the fuzzifier, which performs scale mapping and converts crisp inputs into fuzzy sets using selected membership functions [18]. Scale mapping adjusts the input range to a normalized domain for inference. The fuzzified inputs are then processed through a rule base containing expert-defined fuzzy rules [20]. The inference engine evaluates these rules using logical operations to determine the control action. Mamdani inference, with the AND operator as implication, is commonly used. Alternatively, the Takagi-Sugeno (TS) model can be applied when higher accuracy is needed [14]. Finally, the defuzzifier converts the fuzzy output into a crisp signal applied to the plant.

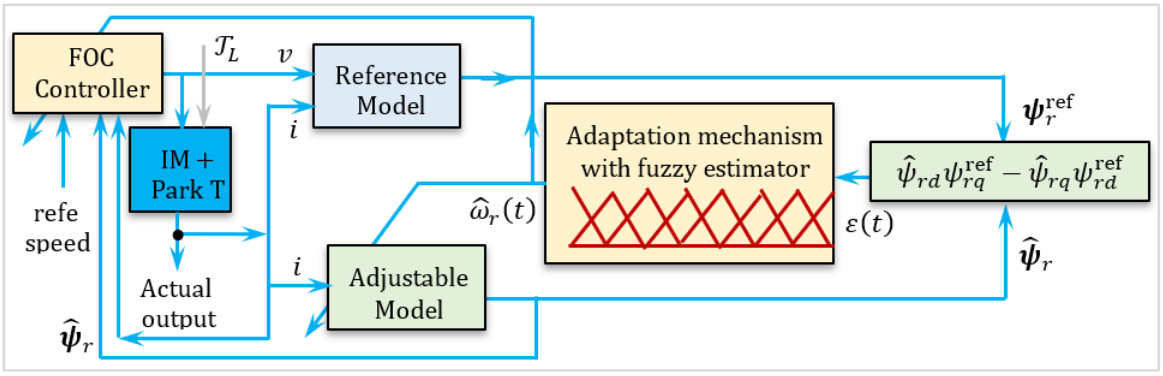

Fig. 1. Adaptation mechanism using a fuzzy logic controller

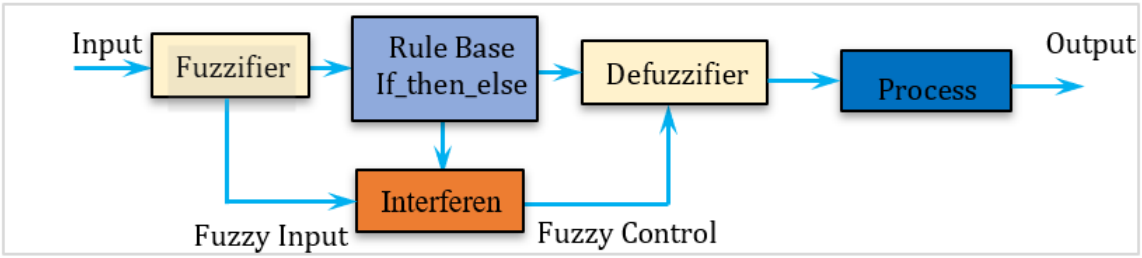

Fig. 2. Configuration of the fuzzy rule-based control framework

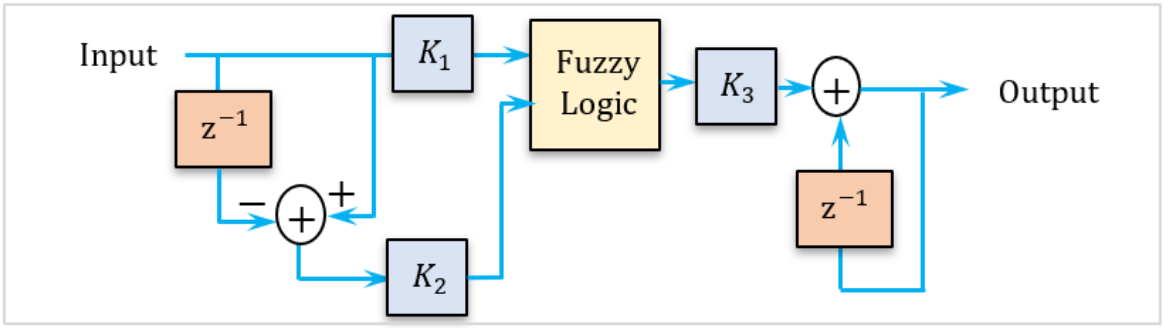

Fig. 3. Schematic representation of the fuzzy logic control system

The gains K_1, K_2 and K_3 act as scaling factors and must be selected carefully to achieve optimal controller performance. This is typically done through trial-and-error tuning. Based on expert knowledge, the fuzzy system employs predefined membership functions, and the associated rule base is presented in Table 1, using linguistic labels such as NB (Negative Big), NS (Negative Small), ZE (Zero), PS (Positive Small), and PB (Positive Big). The fuzzy controller may accept either two or three inputs; however, to reduce system complexity, the proposed design considers only two inputs. Each input can be characterized using 3, 5, 7, or 9 linguistic terms. Consequently,

the total number of fuzzy rules is $N \times N$ or N^2, yielding 9, 25, 49, or 81 rules, respectively. The choice of the number of terms must balance control performance and computational cost, as too few rules may degrade accuracy, while too many may slow down real-time processing. Fig. 4 illustrates the selected membership functions and the resulting control surface of the adaptive fuzzy system. In this work, five linguistic labels with trapezoidal and triangular membership functions were adopted as they offer a practical trade-off between interpretability, real-time efficiency, and control accuracy, as commonly practiced in real-time fuzzy control systems [10]-[12], [15], [22]. The fuzzy inference rules used in the proposed system are summarized in Table 1. The Simulink model developed for this purpose (used for implementation) is shown in Fig. 5.

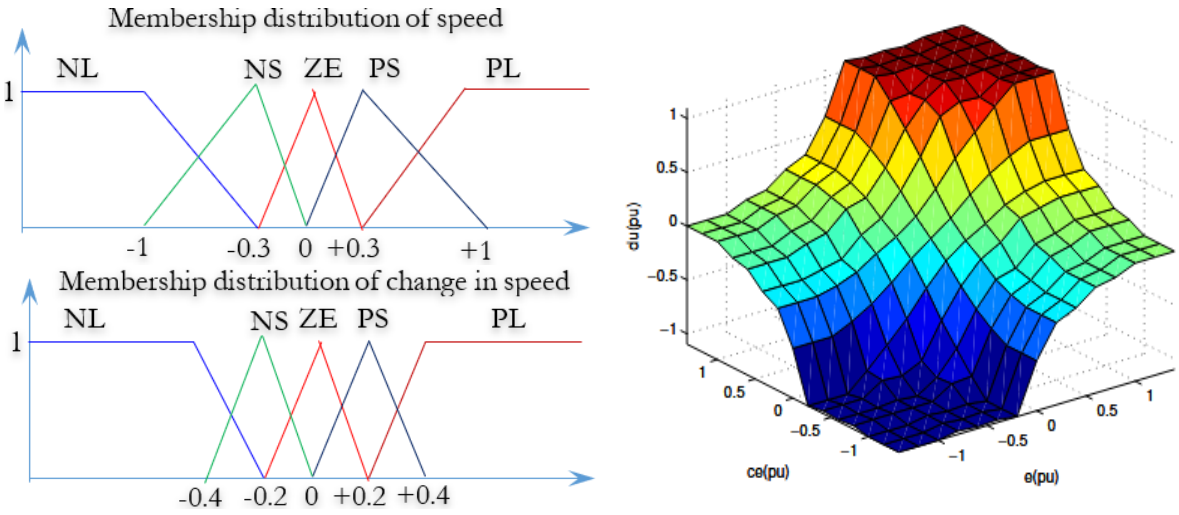

Fig. 4. Membership distribution of speed error, change in speed error and control surface of FLC

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

The fuzzy logic controller design uses five linguistic terms for each input and output variable. This choice represents a widely accepted trade-off between control resolution and rule base complexity, ensuring real-time feasibility. Triangular and trapezoidal membership functions were selected due to their computational efficiency and suitability for embedded implementation. The rule base was constructed using domain expertise and refined empirically through simulation to balance convergence speed and estimation robustness. Let the inputs to the fuzzy controller be defined as the MRAS error signal $e(t)$ and its derivative $\dot{e}(t)$, where:

$$e(t) = y_{1,\text{ref}}(t) - y_{1,\text{est}}(t), \dot{e}(t) = \frac{de(t)}{dt} \tag{11}$$

These two signals are mapped into the fuzzy domain through a scaling process:

$$x_1 = K_1 \cdot e(t), x_2 = K_2 \cdot \dot{e}(t) \tag{12}$$

where K_1 and K_2 are input scaling gains that normalize the crisp input variables to the universe of discourse, typically defined in the range $[-1, 1]$. The fuzzy system output is a speed adaptation signal $v_1(t)$, given by:

$$v_1(t) = K_3 \cdot u_f(t) \tag{13}$$

Here, $u_f(t)$ is the defuzzified output of the fuzzy inference engine, and K_3 is the output gain used to re-scale the fuzzy output to its actual value domain.

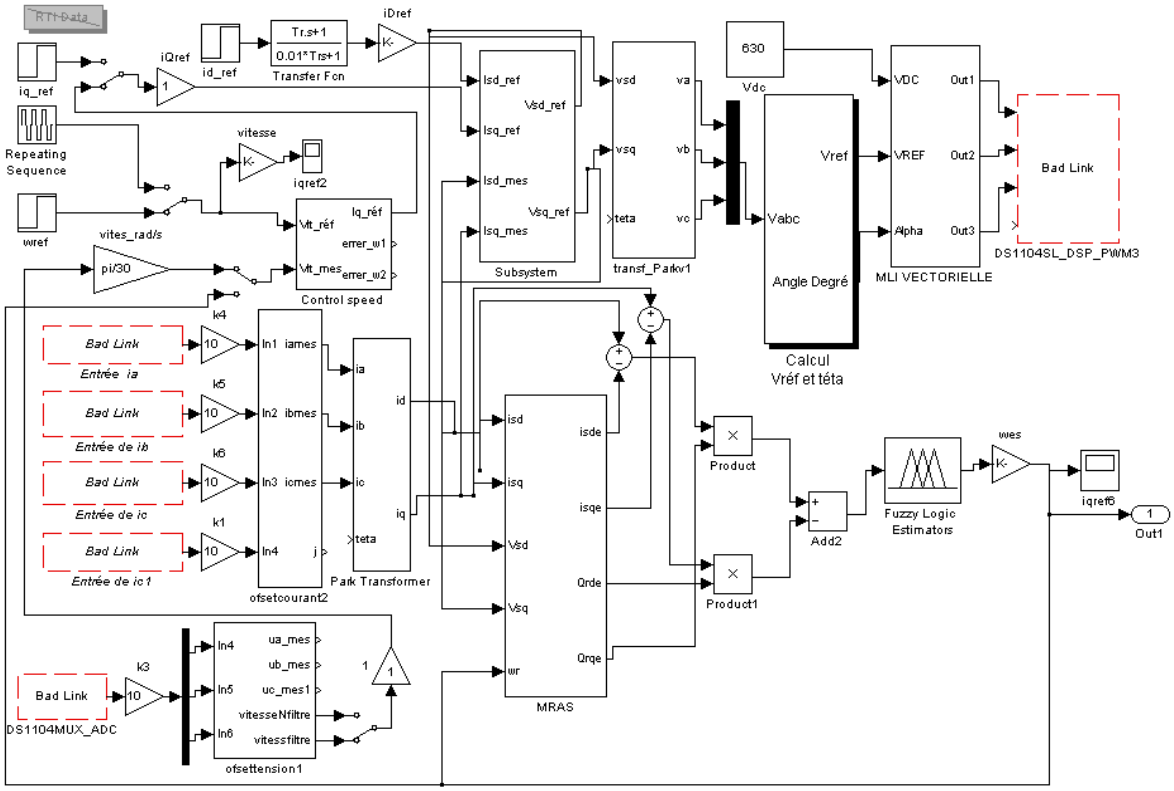

Fig. 5. Real time system block diagram

Each fuzzy input variable x_i (for $i = 1,2$) is described by five linguistic labels: NB, NS, ZE, PS, and PB. Each label is associated with a membership function $\mu_{Aj}(x_i)$, where $j = 1,2,\ldots,5$. These functions are defined using symmetric triangular or trapezoidal shapes, such as:

$$\mu_{Aj}(x) = \left\{0 \quad \text{if } x \le a \text{ or } x \ge c; \frac{x-a}{b-a} \text{ if } a < x \le b; \frac{c-x}{c-b} \text{if } b < x < c\right\} \tag{14}$$

where a, b, and c define the support and peak of the triangle or trapezoid. The fuzzy rule base R consists of $N \times N = 25$ rules, derived from all combinations of input labels. Each rule R_{ij} takes the general Mamdani form:

$$R_{ij}: \quad \text{IF } x_1 \text{ is } A_i \text{ AND } x_2 \text{ is } B_j \quad \text{THEN } v_1 \text{ is } C_{ij} \tag{15}$$

where A_i, B_j are input fuzzy sets and C_{ij} is the output fuzzy set associated with rule R_{ij}. The fuzzy inference engine computes the firing strength w_{ij} for each rule using the minimum operator:

$$w_{ij} = \min\left[\mu A_i(x_1), \mu B_j(x_2)\right] \tag{16}$$

In the aggregation phase, the weighted outputs of all rules are combined using the max–min or centroid method. For the centroid defuzzification, the final crisp output $u_f(t)$ is computed as:

$$u_f(t) = \sum_{ij} w_{ij} z_{ij} / \sum_{ij} w_{ij} \tag{17}$$

where z_{ij} is the centroid (center) of the output membership function C_{ij}. Thus, the complete control law for the fuzzy speed adaptation signal becomes:

$$v_1(t) = K_3 \cdot \left\{ \sum_{ij} \min\left[\mu A_i(x_1), \mu B_j(x_2)\right] z_{ij} / \sum_{ij} \min\left[\mu A_i(x_1), \mu B_j(x_2)\right] \right\} \quad (18)$$

This fuzzy-based adaptation signal is then fed back into the MRAS loop to update the rotor speed estimate, replacing the classical PI-based correction mechanism. The design ensures smoother adaptation, improved noise immunity, and enhanced robustness to parameter variation—critical for real-time sensorless motor drives.

5. The Proposed Practical Neural-Fuzzy Logic Controller

Fig. 6 presents the proposed neural adaptive model reference control system for sensorless induction motor (IM) drives. A neural controller generates control signals based on the reference speed, estimated rotor flux $\hat{\boldsymbol{\psi}}_r$, and current feedback. The system employs both reference and adjustable models, driven by identical inputs, to assess motor performance. The rotor flux error $\varepsilon(t) = \hat{\psi}_{rd}\psi_{rq}^{\text{ref}} - \hat{\psi}_{rq}\psi_{rd}^{\text{ref}}$ is processed by an adaptive fuzzy estimator to estimate the rotor speed $\hat{\omega}_r$. This sensorless approach enhances robustness and eliminates the need for mechanical speed sensors [21]-[25].

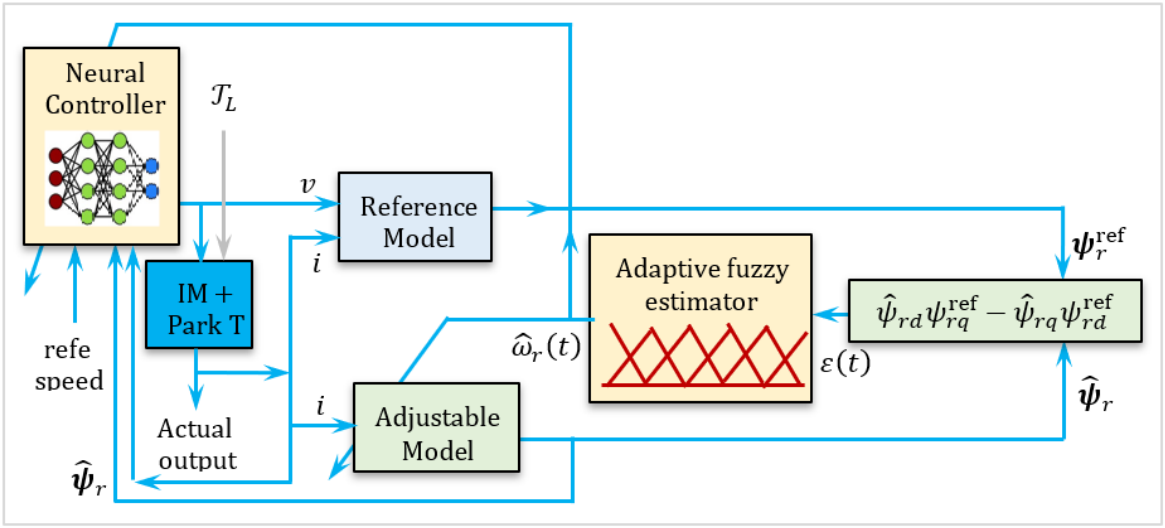

Fig. 6. The NN-Fuzzy MRAS adaptive sensorless control system for induction motor drives

Fig. 7 illustrates the proposed hybrid neural-fuzzy control architecture for sensorless induction motor drives. A recurrent neural network (RNN) generates the reference torque $\mathcal{T}_{\text{emf}}^{\star}$ based on speed error and historical data. This torque is applied through a field-oriented control block to produce stator voltage commands. The rotor speed $\hat{\omega}_r$ is estimated via a fuzzy logic-based estimator, removing the need for mechanical sensors. By combining neural learning and fuzzy adaptation, the system achieves robust performance under both steady-state and dynamic conditions.

The proposed scheme employs a recurrent neural network to estimate the electromagnetic torque $\mathcal{T}_{\text{emf}}^{\star}$ using the rotor speed error and past values. By capturing temporal dependencies, the RNN effectively models the motor's nonlinear dynamics, outperforming conventional feedforward networks. Let $e(k) = \omega_{\text{ref}}(k) - \hat{\omega}_r(k)$ denote the speed tracking error at discrete time step k. The internal structure of the RNN is governed by the following equations [26]-[34]:

1. Hidden State Update: $\mathbf{h}(k) = \boldsymbol{\phi}(\mathbf{W}_{\text{in}} e(k) + \mathbf{W}_{\text{rec}}\mathbf{h}(k-1) + \mathbf{b}_h)$ where:
   - $\mathbf{h}(k) \in \mathbb{R}^n$: hidden state vector, $\mathbf{b}_h \in \mathbb{R}^n$: bias vector,

- $\mathbf{W}_{\text{in}} \in \mathbb{R}^{n\times 1}$: input weight matrix, $\mathbf{W}_{\text{rec}} \in \mathbb{R}^{n\times n}$: recurrent weight matrix,
- $\boldsymbol{\phi}(\cdot)$: element-wise activation function (e.g., tanh or ReLU).

2. Torque Output Estimation: $\mathcal{T}^{\star}_{\text{emf}}(k) = \mathbf{W}_{\text{out}}\mathbf{h}(k) + b_{\text{o}}$ where:

- $\mathbf{W}_{\text{out}} \in \mathbb{R}^{1\times n}$: output weight vector, $b_{\text{o}} \in \mathbb{R}$: output bias.

The complete mapping from input error to torque estimate is compactly expressed as one function: $\mathcal{T}^{\star}_{\text{emf}}(k) = \mathbf{f}_{\text{RNN}}(e(k), \mathbf{h}(k-1); \boldsymbol{\theta})$ where $\boldsymbol{\theta} = \{\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{rec}}, \mathbf{W}_{\text{out}}, \mathbf{b}_h, b_{\text{o}}\}$ is the set of all trainable parameters [35]. This RNN-based estimator enables the controller to adaptively generate torque commands that account for the system's nonlinearities, time delays, and dynamic uncertainties, thus enhancing overall control performance. The RNN parameters $\boldsymbol{\theta}$ are optimized to minimize the mean squared error loss over N time steps: $\mathcal{L} = \left[\sum_{k=1}^{N}\{\mathcal{T}^{\star}_{\text{emf}}(k) - \mathcal{T}_{\text{emf}}(k)\}^2\right]/N$. First we compute hidden states and output torque estimates sequentially: $\mathbf{h}(k) = \boldsymbol{\phi}(\mathbf{W}_{\text{in}}e(k) + \mathbf{W}_{\text{rec}}\mathbf{h}(k-1) + \mathbf{b}_h)$ and $\mathcal{T}^{\star}_{\text{emf}}(k) = \mathbf{W}_{\text{out}}\mathbf{h}(k) + b_{\text{o}}$ then compute gradients $\Delta = \partial\mathcal{L}/\partial\boldsymbol{\theta}$ by unrolling the network through time and applying the chain rule. Finally, update parameters using gradient descent or an advanced optimizer (e.g., Adam): $\boldsymbol{\theta} = \boldsymbol{\theta} + \eta\Delta$ where η is the learning rate. This iterative process continues until convergence, enabling the RNN to learn the dynamic mapping from speed error sequences to torque commands effectively [36]-[40].

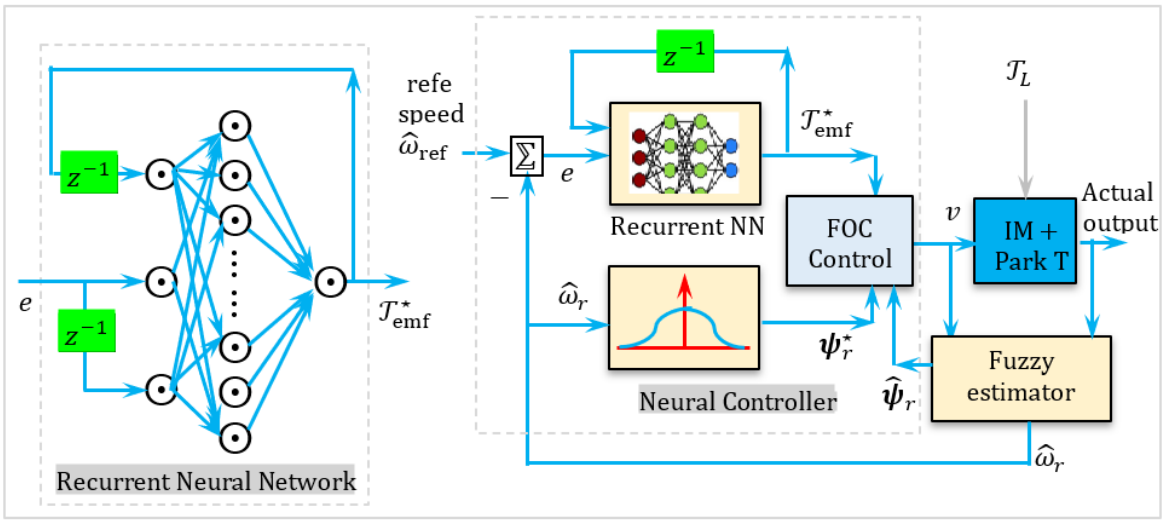

Fig. 7. Detailed diagram of the hybrid neural-fuzzy control scheme for sensorless induction motor drives

If we define $\delta(k) = \partial\mathcal{L}/\partial\mathcal{T}^{\star}_{\text{emf}} = 2[\mathcal{T}^{\star}_{\text{emf}}(k) - \mathcal{T}_{\text{emf}}(k)]/N$ then the gradients for output layer are given by the relations $\mathbf{g}_{\text{out}} = \partial\mathcal{L}/\partial\mathbf{W}_{\text{out}} = \sum_{k=1}^{N}\delta(k)\mathbf{h}^{\top}(k)$ and $\mathbf{g}_{b_{\text{o}}} = \partial\mathcal{L}/\partial b_{\text{o}} = \sum_{k=1}^{N}\delta(k)$. Also, if we back-propagate through time (BPTT) into hidden states $\boldsymbol{\epsilon}(k) = \partial\mathcal{L}/\partial\mathbf{h}(k) \in \mathbb{R}^n$ then we can compute recursively (with $k = N:1$) as $\boldsymbol{\epsilon}(k) = \mathbf{W}_{\text{out}}^{\top}\delta(k) + \mathbf{W}_{\text{rec}}^{\top}\left[\boldsymbol{\epsilon}(k+1)\odot\boldsymbol{\phi}'\left(\boldsymbol{a}(k+1)\right)\right]$ where $\boldsymbol{a}(k) = \mathbf{W}_{\text{in}}e(k) + \mathbf{W}_{\text{rec}}\mathbf{h}(k-1) + \mathbf{b}_h$, $\boldsymbol{\phi}'(\boldsymbol{a})$ is the element-wise derivative of activation and $\odot$ is the element-wise multiplication. Now use $\boldsymbol{\epsilon}(k)$ to compute the following gradients: $\mathbf{g}_{\text{rec}} = \partial\mathcal{L}/\partial\mathbf{W}_{\text{rec}} = \sum_{k=1}^{N}\mathbf{s}(k).\mathbf{h}^{\top}(k-1) \in \mathbb{R}^{n\times n}$, $\mathbf{g}_{\text{in}} = \partial\mathcal{L}/\partial\mathbf{W}_{\text{in}} = \sum_{k=1}^{N}\mathbf{s}(k).e^{\top}(k) \in \mathbb{R}^{n\times 1}$ and $\mathbf{g}_{\mathbf{b}_h} = \partial\mathcal{L}/\partial\mathbf{b}_h = \sum_{k=1}^{N}\mathbf{s}(k). \in \mathbb{R}^n$ with $\mathbf{s}(k) = \left[\boldsymbol{\epsilon}(k)\odot\boldsymbol{\phi}'\left(\boldsymbol{a}(k)\right)\right]$. Finally, if we let η be the learning rate, then: $\mathbf{W}_{\text{out}} \leftarrow \mathbf{W}_{\text{out}} - \eta\mathbf{g}_{\text{out}}$, $\mathbf{W}_{\text{rec}} \leftarrow \mathbf{W}_{\text{rec}} - \eta\mathbf{g}_{\text{rec}}$ etc. All parameters are updated this way. The structure of this learning procedure is illustrated in the flowchart shown in Fig. 8. The RNN parameter update steps using backpropagation are described in the following algorithm.

The RNN architecture employed consists of one hidden layer with 10 neurons using the tanh activation function. The network was trained using the Adam optimizer with a learning rate of 0.001

over 100 epochs. The dataset consisted of 10,000 samples (10 seconds at 1 kHz), divided into 80% training and 20% validation sets. Regularization and early stopping were used to mitigate overfitting. This setup ensures generalization while maintaining a low computational footprint suitable for real-time deployment.

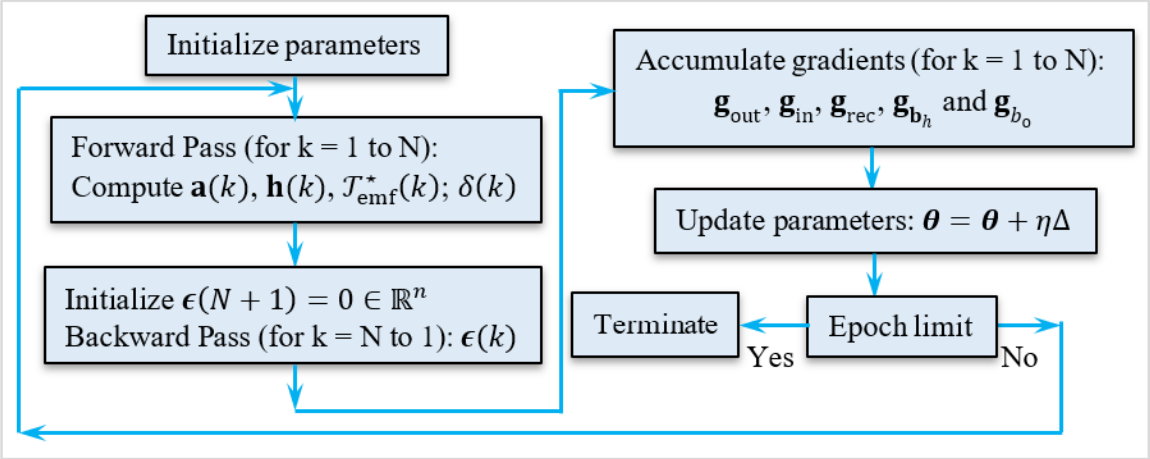

Fig. 8. Training procedure of the RNN-based electromagnetic torque estimator using BPTT

Algorithm: Backpropagation for the RNN Torque Estimator

Inputs:
- Training sequence: $\{e(k), \mathcal{T}_{\text{emf}}(k)\}_{k=1}^{N}$,
- Initial RNN parameters: $\boldsymbol{\theta} = \{\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{rec}}, \mathbf{W}_{\text{out}}, \mathbf{b}_h, b_{\text{o}}\}$
- Activation function: $\boldsymbol{\phi}(.)$ and its derivative $\boldsymbol{\phi}'(.)$, Learning rate: η

Outputs: Updated parameters $\boldsymbol{\theta}$

1. Forward Pass: Compute Activations and Outputs
   For $k = 1$ to N:
   $\boldsymbol{a}(k) = \mathbf{W}_{\text{in}} e(k) + \mathbf{W}_{\text{rec}} \mathbf{h}(k-1) + \mathbf{b}_h$,
   $\mathbf{h}(k) = \boldsymbol{\phi}(\boldsymbol{a}(k))$,
   $\mathcal{T}^{\star}_{\text{emf}}(k) = \mathbf{W}_{\text{out}} \mathbf{h}(k) + b_{\text{o}}$
2. Compute Loss Gradient at Output Layer
   For $k = 1$ to N:
   $\delta(k) = \partial \mathcal{L} / \partial \mathcal{T}^{\star}_{\text{emf}} = 2[\mathcal{T}^{\star}_{\text{emf}}(k) - \mathcal{T}_{\text{emf}}(k)]/N$
3. Backward Pass: Compute Gradients via BPTT
   Initialize $\boldsymbol{\epsilon}(N+1) = 0 \in \mathbb{R}^n$
   For $k = N$ down to 1:
   $\boldsymbol{\epsilon}(k) = \mathbf{W}_{\text{out}}^{\top} \delta(k) + \mathbf{W}_{\text{rec}}^{\top} [\boldsymbol{\epsilon}(k+1) \odot \boldsymbol{\phi}'(\boldsymbol{a}(k+1))]$
4. Accumulate Gradients
   Initialize all gradients to zero.
   For $k = 1$ to N: $\mathbf{s}(k) = [\boldsymbol{\epsilon}(k) \odot \boldsymbol{\phi}'(\boldsymbol{a}(k))]$
   $\mathbf{g}^{+}_{\text{out}} = \delta(k) \cdot \mathbf{h}^{\top}(k) \in \mathbb{R}^{1 \times n}$, $\mathbf{g}^{+}_{\text{rec}} = \mathbf{s}(k).\mathbf{h}^{\top}(k-1) \in \mathbb{R}^{n \times n}$,
   $\mathbf{g}^{+}_{\text{in}} = \mathbf{s}(k).e^{\top}(k) \in \mathbb{R}^{n \times 1}$, $\mathbf{g}^{+}_{b_{\text{o}}} = \delta(k) \in \mathbb{R}^{1 \times n}$, $\mathbf{g}^{+}_{\mathbf{b}_h} = \mathbf{s}(k) \in \mathbb{R}^{n}$
5. Parameter Update
   $\mathbf{W}_{\text{out}} \leftarrow \mathbf{W}_{\text{out}} - \eta \mathbf{g}_{\text{out}}$, $\mathbf{W}_{\text{rec}} \leftarrow \mathbf{W}_{\text{rec}} - \eta \mathbf{g}_{\text{rec}}$,
   $\mathbf{W}_{\text{in}} \leftarrow \mathbf{W}_{\text{in}} - \eta \mathbf{g}_{\text{in}}$, $\mathbf{b}_h \leftarrow \mathbf{b}_h - \eta \mathbf{g}_{\mathbf{b}_h}$, $b_{\text{o}} \leftarrow b_{\text{o}} - \eta \mathbf{g}^{+}_{b_{\text{o}}}$
6. Return Updated Parameters $\boldsymbol{\theta}$

Initialize parameters
Loop over epochs
1. Forward pass: $\mathbf{a}(k)$, $\mathbf{h}(k)$, $\mathcal{T}^{\star}_{\text{emf}}(k)$
2. Compute loss and $\delta(k)$
3. Backward pass: compute ε(k)
4. Accumulate gradients

6. Experimental Results and Discussion

The performance of the proposed neuro-fuzzy controller was assessed using the experimental platform illustrated in Fig. 9. The setup features a three-phase asynchronous induction motor with the following rated specifications: 380 V line voltage, 2.2 A nominal current, 1.1 kW rated power,

1430 rpm nominal speed, and a supply frequency of 50 Hz. The motor is loaded using a synchronous machine coupled with a powder brake to enable controlled loading under various conditions. The power conversion system includes an uncontrolled three-phase diode rectifier followed by a voltage source inverter (VSI) composed of three IGBT modules. For electrical measurement and monitoring, the platform employs LEM LV 25-P sensors for capturing instantaneous stator voltage and LEM LA 55-P sensors for stator current measurement. The instantaneous DC-link voltage is monitored using a LEM CV3-1000 voltage sensor. Although the proposed control scheme is fully sensorless, an incremental encoder (model RS 256-499, 2500 pulses per revolution) is used strictly for offline validation and performance comparison, and does not participate in the feedback loop. The entire control algorithm is executed on a dSPACE DS1104 board, which features a PowerPC 604e processor running at 400 MHz, supported by a TMS320F240 floating-point digital signal processor for real-time signal processing. During operation, key signals and parameters are monitored and recorded using CONTROLDESK software, which provides a real-time interface to the DSP platform.

Fig. 9. Experimental setup for the proposed neural-fuzzy MRAS 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	J	0.0020 Kg.m^2
• Friction coefficient	β	3.1165e-004 N. m/rad/s
• Number of pair poles	p	1

The control algorithm relies on the DS1104 R&D Controller Board created by the German company dSPACE and integrated within a computer. This board contains two processors: a master

processor that handles the application management, and a slave processor, which is a Texas Instruments digital signal processor model TMS320F240, responsible for generating pulse width modulation signals in 0 to 5 volts TTL logic. This hardware forms the physical core of the dSPACE system. On the software side, two programs are involved. The first is MATLAB/Simulink, which enables easy real-time application programming through special blocks from the Real Time Interface (RTI) toolbox that configure the inputs and outputs of the DS1104 card. The second program is ControlDesk, which loads the code (initially designed graphically in Simulink and then compiled into C language) onto the controller, creating a full experimental environment. ControlDesk also provides a graphical user interface to control the real-time process, handle data acquisition, save information in MATLAB-compatible formats for later use, and monitor measured or calculated data using graphical and digital displays. Communication between hardware and software is achieved via an external connection box known as the CP1104 Connector Panel from dSPACE, linked to the board by a shielded cable. This panel receives analog signals through BNC connectors, conditions PWM control signals, monitors any error feedback from the Semikron converter, and interfaces with various sensors. The signal conditioning converts signals between TTL logic levels of 0 to 5 volts and CMOS logic levels of 0 to 15 volts, which is necessary because the DS1104 board operates at TTL levels while the voltage inverter requires CMOS levels. The measurement system uses LEM LA25TP closed-loop Hall effect current sensors for current measurement, LEM type LV100-500 Hall effect voltage sensors for voltage measurement, and an incremental encoder to detect motor rotation speed. Since the analog current signals must be digitized, sampling is performed, and to avoid spectral aliasing, a protective low-pass filter with a cutoff frequency near 500 Hz, which is roughly ten times the fundamental frequency of 50 Hz, is inserted 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 [41]-[45]. Moreover, neural-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 Fig. 10.

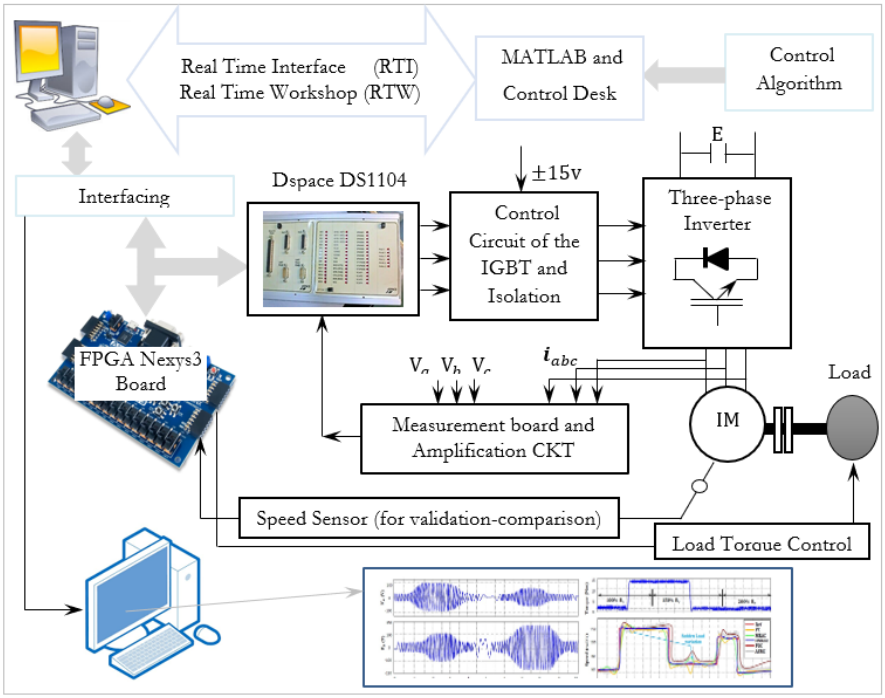

Fig. 10. Architecture of the neuro-fuzzy adaptive control scheme for sensorless induction motor drive

The performance of the proposed NN-fuzzy adaptive control strategy for the IM motor is illustrated in Fig. 11, which presents both control and estimation results under varying speed references.

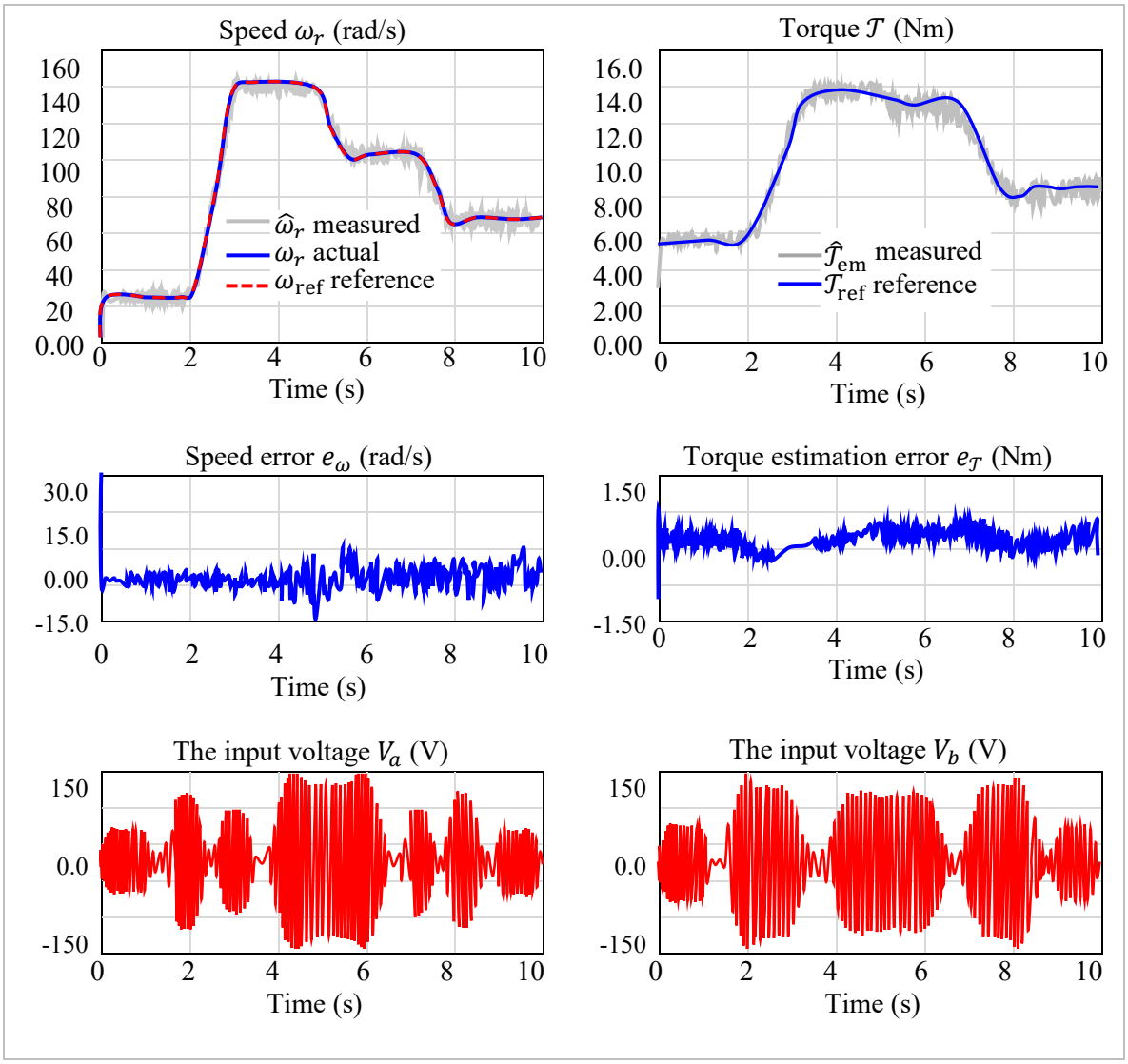

Fig. 11. Performance and estimation accuracy of the proposed control scheme under speed variation

As shown in the upper-left plot (Fig. 11), the rotor speed $\omega_r(t)$ accurately tracks the reference profile $\omega_{\text{ref}}(t)$ over the entire interval, with a rise time of approximately 1.5 s and negligible overshoot. The speed tracking error remains tightly bounded within ±0.52 rad/s, highlighting the controller's precision and dynamic responsiveness. Simultaneously, the motor torque $\mathcal{T}(t)$ and its estimated value $\mathcal{T}_{\text{em}}(t)$ are depicted in the upper-right plot. The torque follows the reference values effectively, with a peak transient around 14.8 Nm during acceleration, and exhibits minimal steady-state error. The estimation error in torque, plotted in the middle-right subplot, remains within ±1.50 Nm, indicating the high accuracy of the neuro-fuzzy observer under sensorless conditions. The control voltages $V_a(t)$ and $V_b(t)$, shown in the lower plots, exhibit typical PWM switching characteristics with modulation amplitudes reaching ±150 V. The dynamic variation of the voltage signals aligns with the torque and speed demands, confirming effective control signal generation via the adaptive neuro-fuzzy controller. Overall, the results validate the proposed control architecture's robustness, adaptability, and estimation fidelity. The controller ensures smooth transitions, high tracking accuracy, and minimal estimation error under varying operating conditions, all while preserving real-time feasibility and avoiding reliance on physical sensors.

The experimental validation (Fig. 12) was conducted on a sensorless vector-controlled induction motor drive system operating under a hybrid neuro-fuzzy control scheme. The test environment was configured to simulate realistic industrial conditions by including various dynamic challenges. The motor was controlled to follow a complex speed reference profile involving acceleration, deceleration, and full speed reversal within a 10-second interval. The system was intentionally exposed to external disturbances, such as load torque variations and noise injection on the measurement channels, in order to test its resilience. Parameter mismatches were also introduced by modifying rotor resistance and stator inductance to assess the adaptability of the observer-controller loop. All control computations were executed in real-time with PWM-based voltage injection, and no mechanical sensors were used, enforcing a fully sensorless operation. Input voltages, speed, and torque signals were sampled with high resolution to ensure accurate monitoring and analysis.

The experimental results clearly demonstrate the strength and reliability of the proposed hybrid neuro-fuzzy control algorithm. The motor accurately tracks the reference speed and torque commands, including during fast transitions and speed reversals, with minimal overshoot and nearly zero steady-state error. The estimated torque closely follows the reference with a peak transient around 12.3 Nm and an estimation error confined within ±1.25 Nm throughout the operation. The speed error remains bounded under ±10 rad/s even during reversal, showing excellent stability. Furthermore, the system sustains stable voltage waveforms across all phases, with no sign of chattering or saturation, validating the effectiveness of the PWM switching strategy. Despite disturbances, parameter variations, and sensorless operation, the proposed controller maintained high performance and strong adaptation capabilities. These results confirm that the hybrid algorithm offers robust and precise control, making it well-suited for real-world applications demanding accuracy, disturbance rejection, and reliable estimation. Comparative study: To evaluate the performance of the proposed control, we use standard electrical engineering (EE) control metrics based on the response to speed or torque ramps (ramp amplitude = |final − initial|):

- $t_{2\%}$ is the time value 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_{fol} Error when reference reaches 50% of ramp.
- $\Delta\mathcal{T}_{\max}$ Max torque deviation during speed ramp.

And in order 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: $\mathrm{MAE} = [\sum_{t=1}^{T}|y_t - \hat{y}_t|]/T$; $\mathrm{SMAPE} = 100.\sum_{t=1}^{T}[|y_t - \hat{y}_t|/(|y_t| + |\hat{y}_t|)]/T$ and $R^2 = 1 - \sum_{t=1}^{T}[\bar{y}_t - \hat{y}_t]^2/\sum_{t=1}^{T}[\bar{y}_t - y_t]^2$ 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 [46]-[51]. To comprehensively assess the effectiveness of the proposed adaptive neuro-fuzzy sensorless control scheme under realistic and demanding operating conditions, two benchmark trajectories were designed:

1. Pseudo-Static Benchmark: In this scenario, the reference speed rapidly ramps from 0 to $157\ \mathrm{rad/s}$ within 1 second and maintains this steady state for 8 seconds. During the interval $[1\mathrm{s}, 2.5\mathrm{s}]$ and $[4\mathrm{s}, 6\mathrm{s}]$, a torque of approximately 7.5 Nm is switched on, with superimposed oscillations post-step to simulate external disturbances or system ripples. This setup evaluates the controller's ability to handle sudden load changes and steady-state performance amidst disturbances.

Dynamic Benchmark: This test case simulates aggressive maneuvering conditions. The reference *speed trajectory* begins with a linear ramp from rest to 157 rad/s during the first second, followed by a constant speed hold at 157 rad/s between 2 and 5 seconds. A rapid reversal then occurs, driving the speed from +157 rad/s to –157 rad/s within the short interval from 5 to 5.5 seconds. This reversed speed is maintained steady between 5.5 and 9 seconds, after which the signal ramps upward from –157 rad/s to 77 rad/s in the final half-second, completing the profile. The *torque profile* starts with zero torque for the first second. A step increase to 8 Nm is applied over a short transition from 1.0 to 1.1 seconds and held constant until 3.5 seconds, followed by a sharp drop back to 0 Nm. A second 8 Nm step is introduced at 5.0 seconds and is then linearly decreased to 4 Nm by 5.9 seconds, followed by another quick return to 0 Nm. At 6.5 seconds, a third 8 Nm step is applied and maintained until 8.0 seconds, after which the torque begins a gradual descent toward 0 Nm, passing through 4 Nm at 9.0 seconds and reaching zero at 9.6 seconds.

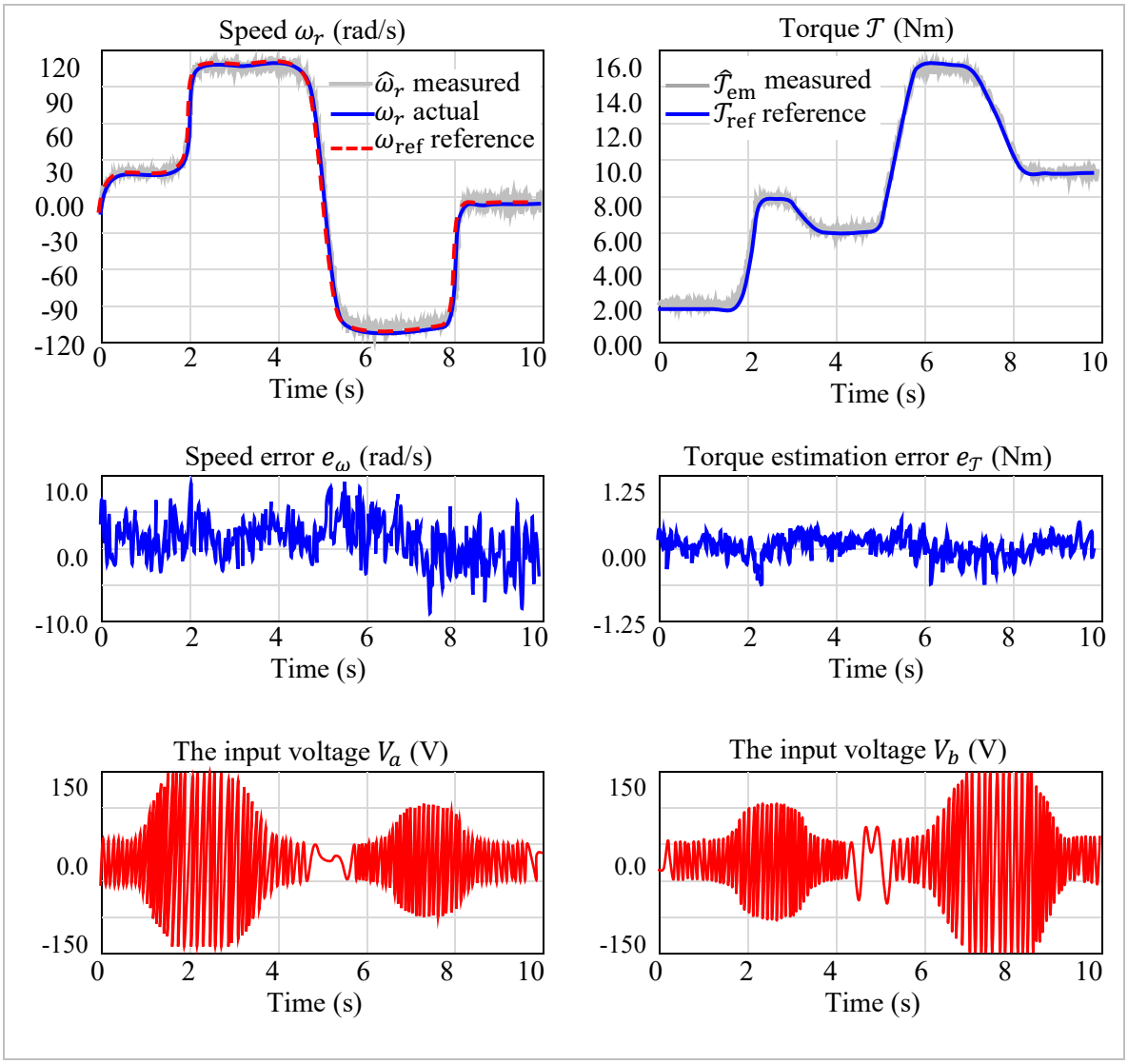

Fig. 12. Performance and estimation accuracy of the proposed control scheme under speed reversal

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. Fig. 13 shows the profiles used during the experimental tests. To assess steady-state and low-dynamic behavior, the four control strategies were subjected to a pseudo-static benchmark consisting of a speed ramp from 0 to 157 rad/s over 1 second, followed by a steady-state hold. Simultaneously, a step torque

disturbance of 7.5 Nm was introduced and held constant, with superimposed high-frequency perturbations to emulate load noise. The benchmark is designed to evaluate tracking accuracy, overshoot suppression, steady-state precision, and torque estimation quality in stable operating conditions. A quantitative comparison of control performance under pseudo-static conditions is summarized in Table 3.

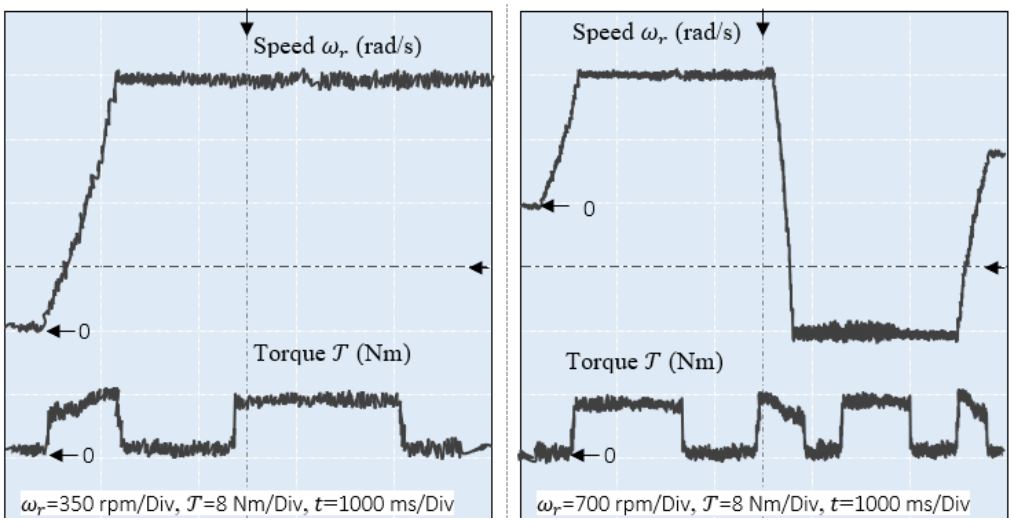

Fig. 13. Realistic operating conditions benchmark profiles for comparative performance evaluation

Table 3. Comparative evaluation of control laws under pseudo-static speed and torque conditions

	Speed (ω_r)								Torque $\mathcal{T}_{em}$	
	MAE (--)	**SMAPE** (%)	$t_{2\%}$ (ms)	$t_{95\%}$ (ms)	E_{fol} (Hz)	$D_{\%}$ (%)	E_{ss} (Hz)	$\Delta\mathcal{T}_{max}$ ($\%_{nom}$)	**MAE** (--)	**SMAPE** (%)
PI-MRAS [11]	0.72	21.30	12	858	0.44	1.60	0.65	36.54	0.48	45.23
Fuzzy-MRAS [7]	0.13	17.91	16	841	0.18	0.90	0.21	30.41	0.19	36.52
NN-MRAS [29]	0.09	16.83	15	855	0.11	0.74	0.13	28.72	0.13	29.81
NN-Fuzzy	**0.07**	**16.54**	**14**	**848**	**0.10**	**0.59**	**0.08**	**27.23**	**0.11**	**26.37**
R^2 is ≈ 0.99 for all the used controller.										

The results demonstrate a clear performance improvement when transitioning from conventional PI-based MRAS to advanced AI-based methods. The hybrid NN-Fuzzy controller achieved the lowest speed tracking error (MAE = 0.07 rad/s) and minimal steady-state error (E_{ss} = 0.08 rad/s), outperforming all other control laws in terms of stability and precision. In torque estimation, the NN-Fuzzy method again delivered the best result with a 26.3% SMAPE, highlighting its ability to handle low-frequency disturbances with high accuracy. The overshoot ($D_{\%}$) and torque deviation ($\Delta\mathcal{T}_{max}$) are also significantly reduced compared to the baseline controller, indicating strong robustness under near-static load conditions.

To evaluate the controllers under dynamic and nonlinear conditions, a comprehensive benchmark was conducted, incorporating fast acceleration, full speed reversal, and variable torque loading. The speed reference varied between ±157 rad/s, while torque underwent rapid steps, ramps, and sinusoidal perturbations—highlighting the controllers' ability to adapt to simultaneous nonlinear speed and torque variations while preserving estimation fidelity and transient performance. Additionally, the robustness of the proposed controller was assessed against distinct disturbance categories, including step changes in load torque, rotor resistance mismatches (±10% to emulate parametric uncertainty), and high-frequency Gaussian noise (variance = 0.002) injected into voltage and current measurements to simulate sensor disturbances. Each disturbance was tested independently to isolate its effects and assess adaptability across scenarios. The results of these evaluations, including speed reversal and load variation, are summarized in Table 4. Future work

will investigate systematic variations in disturbance magnitude and frequency to establish performance bounds and quantify robustness more formally.

Table 4. Comparative performance under dynamic speed reversal and load transients

	Speed (ω_r)								Torque $\mathcal{T}_{em}$	
	MAE (--)	SMAPE (%)	$t_{2\%}$ (ms)	$t_{95\%}$ (ms)	E_{fol} (Hz)	$D_{\%}$ (%)	E_{ss} (Hz)	$\Delta\mathcal{T}_{max}$ ($\%_{nom}$)	MAE (--)	SMAPE (%)
PI-MRAS [11]	1.48	25.6	18	1025	1.22	4.10	1.10	42.33	0.83	53.2
Fuzzy-MRAS [7]	0.34	21.3	22	1010	0.57	2.37	0.39	36.75	0.41	39.64
NN-MRAS [29]	0.22	19.25	21	996	0.32	1.63	0.27	33.54	0.13	29.81
NN-Fuzzy	**0.17**	**18.44**	**20**	**990**	**0.26**	**1.12**	**0.19**	**30.83**	**0.20**	**28.55**

R^2 is ≈ 0.98 for all the used controller.

Under dynamic speed reversal and variable torque conditions, the performance gap between the control strategies becomes more pronounced. While the classical PI-MRAS struggles with overshoot and tracking delays ($D_{\%}$ = 4.1%, E_{ss} = 1.10 rad/s), the AI-based approaches maintain significantly better precision. The NN-Fuzzy controller exhibits the best performance overall, with the lowest speed MAE (0.17 rad/s) and torque SMAPE (28.5%), while also minimizing transient torque deviation ($\Delta\mathcal{T}_{max}$ = 30.8%). These results confirm the superior adaptability and dynamic tracking capabilities of the hybrid neuro-fuzzy architecture, especially in complex, real-world operation scenarios where both speed and torque vary rapidly. Table 5 presents the speed error values recorded during a challenging test scenario involving full speed reversal combined with a sudden 2 Nm load step, under both nominal and perturbed conditions. Specifically, it illustrates the performance degradation of each control scheme in the presence of ±10% rotor resistance variation, highlighting their sensitivity to parametric uncertainty and their ability to maintain accurate speed estimation under abrupt dynamic changes.

Table 5. Performance degradation under ±10% rotor resistance variation

	Speed Error (rad/s)		
	Nominal R_r	**+10% R_r**	**+10% R_r**
PI-MRAS [11]	0.82	1.45	1.33
Fuzzy-MRAS [7]	0.48	0.91	0.84
NN-MRAS [29]	0.39	0.65	0.62
NN-Fuzzy	**0.07**	**0.13**	**0.11**

Compared to previous studies such as [11], [7] and [29], which employed standalone MRAS, fuzzy-MRAS, or neural-based MRAS designs, the proposed hybrid controller exhibits superior robustness and performance under speed reversals, noise, and parametric disturbances. This improvement stems from its ability to integrate nonlinear inference with dynamic memory, enhancing adaptability across varying operating conditions.

These improvements suggest potential benefits in real-world industrial applications. Enhanced estimation accuracy reduces startup delays, minimizes energy waste during transients, and may extend motor lifespan by reducing thermal and mechanical stress. While the current implementation is tailored to a specific drive configuration, future research will explore broader applicability, hardware abstraction, and cost-aware optimization.

Explanation of Findings along with Strengths and Limitations: The fuzzy logic component provides nonlinear inference from expert-defined rules, handling signal noise and uncertainties effectively. Meanwhile, the RNN learns temporal relationships from past behavior, enabling dynamic adaptation during transient conditions such as reversals. The architecture combines generalization and online adaptability while maintaining real-time feasibility. However, training the

RNN requires significant data and preprocessing, and tuning the fuzzy inference mechanism may require expert input, which limits plug-and-play deployment.

Benchmarking Scope and Justification: The proposed neuro-fuzzy MRAS estimator was benchmarked against three reference schemes: a classical PI-based MRAS, a standalone fuzzy-MRAS, and a neural-MRAS observer. These schemes were chosen because they represent the most widely used and practically implementable classes of sensorless control strategies—conventional linear, rule-based, and data-driven approaches, respectively. This comparison provides a fair and balanced evaluation of the proposed method across both classical and intelligent baselines. Other advanced methods such as sliding mode observers or robust adaptive control were not included, as they follow fundamentally different design philosophies or require significant structural modifications beyond the MRAS framework, which fall outside the scope of this study.

The reported results include MAE, SMAPE, and R^2 to quantify performance. While these metrics confirm strong accuracy, the current study does not include confidence intervals or hypothesis testing due to the dataset's size and structure. These will be addressed in future work through statistical validation and broader test campaigns.

7. Conclusion

This work presented a sensorless control strategy for induction motors based on a hybrid neuro-fuzzy Model Reference Adaptive System (MRAS). The main contribution lies in replacing the traditional proportional-integral (PI) adaptation mechanism with a novel architecture that integrates a fuzzy logic estimator and a recurrent neural network (RNN)-based torque predictor. This combination enables robust, real-time estimation of rotor speed and electromagnetic torque, even under low-speed, high-disturbance, or parameter-variant conditions. The approach was experimentally validated using a 1.1 kW induction motor under realistic industrial operating conditions. Two benchmark profiles, including speed reversals, torque steps, and load disturbances, were used to assess dynamic adaptability and steady-state performance. The proposed hybrid controller achieved a 91% reduction in steady-state speed error (from 0.65 rad/s to 0.08 rad/s) and a 42% improvement in torque estimation accuracy, reducing SMAPE from 45.2% to 26.3%. It consistently outperformed the standalone PI-MRAS, fuzzy-MRAS, and neural-MRAS controllers across all tested metrics, including rise time, overshoot, and mean absolute error.

From a theoretical standpoint, this work introduces a neuro-fuzzy structure tailored for MRAS-based observers, bridging symbolic reasoning and data-driven adaptation within a unified framework. The architecture contributes new insights into the synergistic benefits of combining fuzzy logic with temporal learning mechanisms like RNNs for motor control applications. However, the proposed scheme has some limitations. The RNN component requires retraining if applied to motors with significantly different dynamics, and the fuzzy rule base may need adaptation under drastically varying load profiles. These factors may affect generalization in cross-domain deployment without recalibration. Despite these constraints, the system retains strong adaptability and is well-suited for real-time sensorless drives.

Looking ahead, future work will focus on improving scalability and extending the method to more complex systems such as doubly-fed induction machines and coaxial rotor configurations. Another promising direction involves replacing the RNN block with advanced memory-based models like Long Short-Term Memory (LSTM) networks to better capture long-range temporal dependencies in torque dynamics. Additionally, efforts will be directed toward automating the fuzzy rule optimization process using evolutionary algorithms or reinforcement learning to enhance controller portability across different drive types. In summary, this study offers a validated, AI-driven control solution that improves estimation accuracy, increases robustness to uncertainties, and advances the state-of-the-art in sensorless induction motor control.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

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

References

[1] A. Bennassar, A. Abbou and M. Akherraz, "Combining fuzzy Luenberger observer and Kalman filter for speed sensorless integral backstepping controlled induction motor drive," *International Journal of Automation and Control*, vol. 11, no. 3, pp. 298-313, 2017, https://doi.org/10.1504/IJAAC.2017.084871.

[2] A. Rihar *et al*., "Emerging Technologies for Advanced Power Electronics and Machine Design in Electric Drives," *Applied Sciences*, vol. 14, no. 24, p. 11559, 2024, https://doi.org/10.3390/app142411559.

[3] F. Shiravani, P. Alkorta, J. A. Cortajarena, and O. Barambones, "An Enhanced Sliding Mode Speed Control for Induction Motor Drives," *Actuators*, vol. 11, no. 1, p. 18, 2022, https://doi.org/10.3390/act11010018.

[4] P. M. Menghal and A. J. Laxmi, "Neural network based dynamic simulation of induction motor drive," *2013 International Conference on Power, Energy and Control (ICPEC)*, pp. 566-571, 2013, https://doi.org/10.1109/ICPEC.2013.6527722.

[5] U. Sengamalai *et al*., "Three Phase Induction Motor Drive: A Systematic Review on Dynamic Modeling, Parameter Estimation, and Control Schemes," *Energies*, vol. 15, no. 21, p. 8260, 2022, https://doi.org/10.3390/en15218260.

[6] H. Kubota, K. Matsuse and T. Nakano, "DSP-based speed adaptive flux observer of induction motor," *IEEE Transactions on Industry Applications*, vol. 29, no. 2, pp. 344-348, 1993, https://doi.org/10.1109/28.216542.

[7] N. Medjadji, A. Chaker, D. M. Medjahed, "Quality Improvement of Speed Estimation in Induction Motor Drive by Using Fuzzy-MRAS Observer with Experimental Investigation," *International Review of Automatic* Control, vol. 11, no. 1, 2018, https://doi.org/10.15866/ireaco.v11i1.14425.

[8] Q. Sun *et al*., "Adaptive Robust Control of an Industrial Motor-Driven Stage with Disturbance Rejection Ability Based on Multidimensional Taylor Network," *Applied Sciences*, vol. 13, no. 22, p. 12231, 2023, https://doi.org/10.3390/app132212231.

[9] X. Liu, S. Zhen, F. Wang, M. Li, "Adaptive robust control of the PMSM servo system with servo and performance constraints," *Journal of Vibration and Control*, 2024, https://doi.org/10.1177/10775463241278003.

[10] Z. Mekrini, S. Bri, "Fuzzy Logic Application for Intelligent Control of An Asynchronous Machine," *Indonesian Journal of Electrical Engineering and Computer* Science, vol. 7, no. 1, pp. 61-70, 2017, http://doi.org/10.11591/ijeecs.v7.i1.pp61-70.

[11] W. Hamdi, M. Y. Hammoudi, and A. Betka, "Sensorless Speed Control of Induction Motor Using Model Reference Adaptive System and Deadbeat Regulator," *Engineering Proceedings*, vol. 56, no. 1, p. 16, 2023, https://doi.org/10.3390/ASEC2023-15240.

[12] K. N. 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, pp. 110-118, 2010, http://dx.doi.org/10.4236/jilsa.2010.22014.

[13] P. Stewart, D. A. Stone, P. J. Fleming, "Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop," *Engineering Applications of Artificial* Intelligence, vol. 17, no. 3, pp. 275-284, 2004, https://doi.org/10.1016/j.engappai.2004.03.003.

[14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116-132, 1985, https://doi.org/10.1109/TSMC.1985.6313399.

[15] B. Bekhiti, B. Nail, I. E. Tibermacine, and R. Salim, "On Hyper-Stability Theory Based Multivariable Nonlinear Adaptive Control: Experimental Validation on Induction Motors," *IET Electric Power Applications*, vol. 19, no. 1, p. e70035, 2025, https://doi.org/10.1049/elp2.70035.

[16] S. Rigon, R. Antonello, P. Mercorelli and M. Zigliotto, "Sensorless Control of AC Motor Drives with Adaptive Extended Kalman Filter," *2024 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 5991-5998, 2024, https://doi.org/10.1109/ECCE55643.2024.10860865.

[17] L. F. C. Ccari *et al*., "Robust Finite-Time Adaptive Nonlinear Control System for an EOD Robotic Manipulator: Design, Implementation, and Experimental Validation," *IEEE Access*, vol. 12, pp. 93859-93875, 2024, https://doi.org/10.1109/ACCESS.2024.3424463.

[18] A. A. Mostfa, N. A. Zakar, R. R. Al-Mola, and A. N. Sharkawy, "Enhancing Wind Turbine Power Output Estimation Using Causal Inference and Adaptive Neuro-Fuzzy Inference System ANFIS," *Kufa Journal of Engineering*, vol. 16, no. 2, pp. 400-422, 2025, https://doi.org/10.30572/2018/KJE/160224.

[19] I. A. Kheioon, R. Al-Sabur, and A.-N. Sharkawy, "Design and Modeling of an Intelligent Robotic Gripper Using a Cam Mechanism with Position and Force Control Using an Adaptive Neuro-Fuzzy Computing Technique," *Automation*, vol. 6, no. 1, p. 4, 2025, https://doi.org/10.3390/automation6010004.

[20] D. D. Saputra *et al*., "Performance evaluation of sliding mode control (SMC) for dc motor speed control," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 2, pp. 502-510, 2023, https://doi.org/10.26555/jiteki.v9i2.26291.

[21] M. H. Sabzalian *et al*., "A Neural Controller for Induction Motors: Fractional-Order Stability Analysis and Online Learning Algorithm," *Mathematics*, vol. 10, no. 6, p. 1003, 2022, https://doi.org/10.3390/math10061003.

[22] M. A. Denaï and S. A. Attia, "Intelligent Control of an Induction Motor," *Electric Power Components and Systems*, vol. 30, no. 4, pp. 409-427, 2002, https://doi.org/10.1080/15325000252888010.

[23] N. Pimkumwong and M.-S. Wang, "Online Speed Estimation Using Artificial Neural Network for Speed Sensorless Direct Torque Control of Induction Motor based on Constant V/F Control Technique," *Energies*, vol. 11, no. 8, p. 2176, 2018, https://doi.org/10.3390/en11082176.

[24] A. A. Bohari, W. M. Utomo, Z. A. Haron, N. M. Zin, S. Y. Sim, and R. M. Ariff, "Speed tracking of indirect field oriented control induction motor using neural network," *Procedia Technology*, vol. 11, pp. 141-146, 2013, https://doi.org/10.1016/j.protcy.2013.12.173.

[25] M. Demirtas, E. Ilten, and H. Calgan, "Pareto-based multi-objective optimization for fractional order PIλ speed control of induction motor by using Elman Neural Network," *Arabian Journal for Science and Engineering*, vol. 44, no. 3, pp. 2165-2175, 2019, https://doi.org/10.1007/s13369-018-3364-2.

[26] P. Brandstetter and M. Kuchar, "Sensorless control of variable speed induction motor drive using RBF neural network," *Journal of Applied Logic*, vol. 24, pp. 97-108, 2017, https://doi.org/10.1016/j.jal.2016.11.017.

[27] H. Saleeb, R. Kassem, and K. Sayed, "Artificial neural networks applied on induction motor drive for an electric vehicle propulsion system," *Electrical Engineering*, vol. 104, no. 3, pp. 1769-1780, 2022, https://doi.org/10.1007/s00202-021-01418-y.

[28] H. Acikgoz, "Real-time adaptive speed control of vector-controlled induction motor drive based on online-trained Type-2 Fuzzy Neural Network Controller," *International Transactions on Electrical Energy Systems*, vol. 30, no. 12, p. e12678, 2020, https://doi.org/10.1002/2050-7038.12678.

[29] A. B. Sharma, S. Tiwari and B. Singh, "Intelligent Speed Estimation in Induction Motor Drive Control using Feed - Forward Neural Network Assisted Model Reference Adaptive System," *2020 IEEE Students Conference on Engineering & Systems (SCES)*, pp. 1-6, 2020, https://doi.org/10.1109/SCES50439.2020.9236736.

[30] S. Hussain and M. A. Bazaz, "Adaptive neural Type II fuzzy logic-based speed control of induction motor drive," *Ambient Communications and Computer Systems: RACCCS 2017*, pp. 81-92, 2018, https://doi.org/10.1007/978-981-10-7386-1_7.

[31] M. H. Sabzalian, A. Mohammadzadeh, S. Lin, and W. Zhang, "New approach to control the induction motors based on immersion and invariance technique," *IET Control Theory & Applications*, vol. 13, no. 10, pp. 1466-1472, 2019, https://doi.org/10.1049/iet-cta.2018.5026.

[32] R. Kumar, S. Das, P. Syam, and A. K. Chattopadhyay, "Review on model reference adaptive system for sensorless vector control of induction motor drives," *IET Electric Power Applications*, vol. 9, no. 7, pp. 496-511, 2015, https://doi.org/10.1049/iet-epa.2014.0220.

[33] Y. Ren, R. Wang, S. J. Rind, P. Zeng, and L. Jiang, "Speed sensorless nonlinear adaptive control of induction motor using combined speed and perturbation observer," *Control Engineering Practice*, vol. 123, p. 105166, 2022, https://doi.org/10.1016/j.conengprac.2022.105166.

[34] J. C. Travieso-Torres and M. A. Duarte-Mermoud, "Normalized Model Reference Adaptive Control Applied to High Starting Torque Scalar Control Scheme for Induction Motors," *Energies*, vol. 15, no. 10, p. 3606, 2022, https://doi.org/10.3390/en15103606.

[35] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," *Information*, vol. 15, no. 9, p. 517, 2024, https://doi.org/10.3390/info15090517.

[36] V. S. Lalapura, V. R. Bhimavarapu, J. Amudha, and H. S. Satheesh, "A Systematic Evaluation of Recurrent Neural Network Models for Edge Intelligence and Human Activity Recognition Applications," *Algorithms*, vol. 17, no. 3, p. 104, 2024, https://doi.org/10.3390/a17030104.

[37] X. Wu, B. Xiang, H. Lu, C. Li, X. Huang, and W. Huang, "Optimizing Recurrent Neural Networks: A Study on Gradient Normalization of Weights for Enhanced Training Efficiency," *Applied Sciences*, vol. 14, no. 15, p. 6578, 2024, https://doi.org/10.3390/app14156578.

[38] A. G. Zinyagin, A. V. Muntin, V. S. Tynchenko, P. I. Zhikharev, N. R. Borisenko, and I. Malashin, "Recurrent Neural Network (RNN)-Based Approach to Predict Mean Flow Stress in Industrial Rolling," *Metals*, vol. 14, no. 12, p. 1329, 2024, https://doi.org/10.3390/met14121329.

[39] C.-H. Yu *et al*., "Recurrent Neural Network Methods for Extracting Dynamic Balance Variables during Gait from a Single Inertial Measurement Unit," *Sensors*, vol. 23, no. 22, p. 9040, 2023, https://doi.org/10.3390/s23229040.

[40] D. Utebayeva, L. Ilipbayeva, and E. T. Matson, "Practical Study of Recurrent Neural Networks for Efficient Real-Time Drone Sound Detection: A Review," *Drones*, vol. 7, no. 1, p. 26, 2023, https://doi.org/10.3390/drones7010026.

[41] R. Pascual, M. Esteban, J. M. Guerrero, and C. A. Platero, "Recurrent Neuronal Networks for the Prediction of the Temperature of a Synchronous Machine During Its Operation," *Machines*, vol. 13, no. 5, p. 387, 2025, https://doi.org/10.3390/machines13050387.

[42] A. J. Abougarair, M. K. Aburakhis, and M. M. Edardar, "Adaptive neural networks based robust output feedback controllers for nonlinear systems," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 37-56, 2022, https://doi.org/10.31763/ijrcs.v2i1.523.

[43] S. E. Mezaache and E. Zaidi, "Performance Enhancement of Dual-Star Induction Machines Using Neuro-Fuzzy Control and Multi-Level Inverters: A Comparative Study with PI Controllers," *International Journal of Robotics and Control Systems*, vol. 5, no. 1, pp. 197-221, 2025, https://doi.org/10.31763/ijrcs.v5i1.1670.

[44] N. T. Pham and P. D. Nguyen, "A Novel Hybrid Backstepping and Fuzzy Control for Three Phase Induction Motor Drivers," *International Journal of Robotics and Control Systems*, vol. 5, no. 1, pp. 599-610, 2025, https://doi.org/10.31763/ijrcs.v5i1.1707.

[45] N. T. Pham, "Design of Novel STASOSM Controller for FOC Control of Dual Star Induction Motor Drives," *International Journal of Robotics and Control Systems*, vol. 4, no. 3, pp. 1059-1074, 2024, https://doi.org/10.31763/ijrcs.v4i3.1443.

[46] N. Basil and H. M. Marhoon, "Selection and evaluation of FOPID criteria for the X-15 adaptive flight control system (AFCS) via Lyapunov candidates: Optimizing trade-offs and critical values using optimization algorithms," *e-Prime-advances in electrical engineering, electronics and energy*, vol. 6, p. 100305, 2023, https://doi.org/10.1016/j.prime.2023.100305.

[47] B. Bekhiti, J. Iqbal, K. Hariche, and G. F. Fragulis, "Neural Adaptive Nonlinear MIMO Control for Bipedal Walking Robot Locomotion in Hazardous and Complex Task Applications," *Robotics*, vol. 14, no. 6, p. 84, 2025, https://doi.org/10.3390/robotics14060084.

[48] K. H. Mahmoud, A. N. Sharkawy, and G. T. Abdel-Jaber, "Development of safety method for a 3-DOF industrial robot based on recurrent neural network," *Journal of Engineering and Applied Science*, vol. 70, no. 1, p. 44, 2023, https://doi.org/10.1186/s44147-023-00214-8.

[49] B. Bekhiti, G. F. Fragulis, M. Rahmouni, and K. Hariche, "A Novel Three-Dimensional Sliding Pursuit Guidance and Control of Surface-to-Air Missiles," *Technologies*, vol. 13, no. 5, p. 171, 2025, https://doi.org/10.3390/technologies13050171.

[50] A. N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "A recurrent neural network for variable admittance control in human–robot cooperation: simultaneously and online adjustment of the virtual damping and Inertia parameters," *International Journal of Intelligent Robotics and Applications*, vol. 4, no. 4, pp. 441-464, 2020, https://doi.org/10.1007/s41315-020-00154-z.

[51] B. Bekhiti, G. F. Fragulis, G. S. Maraslidis, K. Hariche, and K. Cherifi, "A Novel Recursive Algorithm for Inverting Matrix Polynomials via a Generalized Leverrier–Faddeev Scheme: Application to FEM Modeling of Wing Vibrations in a 4th-Generation Fighter Aircraft," *Mathematics*, vol. 13, no. 13, p. 2101, 2025, https://doi.org/10.3390/math13132101.