mathematics

MDPI

*Article*

# Realization of Intelligent Observer for Sensorless PMSM Drive Control

**Dwi Sudarno Putra [1,2], Seng-Chi Chen [1,*], Hoai-Hung Khong [3] and Chin-Feng Chang [4]**

[1] Department of Electrical Engineering, Southern Taiwan University of Science and Technology, Tainan 710, Taiwan
[2] Department of Automotive Engineering, Universitas Negeri Padang, Padang 25132, Indonesia
[3] Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Transport, Ho Chi Minh City 70000, Vietnam
[4] Fukuta Electric and Machinery Co., Ltd., Taichung City 429, Taiwan
[*] Correspondence: amtfcsg123@stust.edu.tw; Tel.: +886-6-253-3131 (ext. 3324)

**Abstract:** An observer is a crucial part of the sensorless control of a permanent magnet synchronous motor (PMSM). An observer, based on mathematical equations, depends on information regarding several parameters of the controlled motor. If the motor is replaced, then we need to know the motor parameter values and reset the observer's parameters. This article discusses an intelligent observer that can be used for several motors with different parameters. The proposed intelligent observer was developed using machine learning methods. This observer's core algorithm is a modified Jordan neural network. It processes $I_\alpha$, $I_\beta$, $v_\alpha$, and $v_\beta$ to produce $Sin\ \theta$ and $Cos\ \theta$ values. It is combined with a phase-locked loop function to generate position and speed feedback information. The offline learning process is carried out using data acquired from the simulations of PMSM motors. This study used five PMSMs with different parameters, three as the learning reference sources and two as testing sources. The proposed intelligent observer was successfully used to control motors with different parameters in both simulation and experimental hardware. The average error in position estimated for the simulation was 0.0078 p.u and the error was 0.0100 p.u for the experimental realization.

**Keywords:** intelligent observer; PMSM drive control; machine learning realization; modified Jordan neural networks

**MSC:** 37N35

check for updates

**Citation:** Putra, D.S.; Chen, S.-C.; Khong, H.-H.; Chang, C.-F. Realization of Intelligent Observer for Sensorless PMSM Drive Control. *Mathematics* **2023**, *11*, 1254. https://doi.org/10.3390/math11051254

Academic Editor: Aydin Azizi

Received: 6 February 2023
Revised: 27 February 2023
Accepted: 2 March 2023
Published: 5 March 2023

## 1. Introduction

The permanent magnet synchronous motor (PMSM) is one of the most popular electric motors. This motor, also known as a brushless direct current motor (BLDCM) with sinusoidal back emf, has better efficiency than the trapezoidal back emf BLDCM [1–4]. PMSM control can be sensored [5–8] or sensorless [9–14]. Controlling PMSMs without sensors can reduce manufacturing costs and eliminate the presence of sensors in the motor.

There are several types of sensorless PMSM control techniques. Back EMF (electromotive force) estimation: this technique estimates the position of the rotor by measuring the voltage generated by the motor's windings as the rotor rotates [15–17]. High-frequency injection: this technique injects a high-frequency signal into the motor windings and measures the resulting changes in current to determine the position of the rotor [18–23]. Kalman filter: this technique uses a statistical filter to estimate the position of the rotor based on the measured currents and voltages [24–28]. Sliding mode observer: this technique uses a mathematical model of the motor to estimate the position of the rotor based on the measured currents and voltages [29–34]. The Luenberger observer: Provides a simple and computationally efficient method for estimating rotor position and speed. It can operate over a wide range of speeds and can compensate for disturbances and uncertainties in the system [35,36]. The motor's specification parameters are needed for observer calculation.

Some more research has revealed that observers can be developed using a machine learning (ML) method. Some research mentioned that ML-based observers could be established after some training processes in artificial neural networks (ANN) [37–40]. Training data used in the training process were current and voltage data from the simulation control process. The learning process did not involve the specific motor parameters.

Previously, the authors successfully implemented an ML-based observer in a sensorless PMSM field-oriented control motor (PMSM-FOC) in both simulation and hardware experiments. The ML-based observer was developed using modified Elman neural networks (MENN). Figure 1 shows the MENN consisted of seven neurons in the hidden layer and seven in the context layer. The hidden layer emitted seven data arrays (*[1 2 3 4 5 6 7]*). There were a total of 98 networks leading into the hidden layer. Learning data were taken from a PMSM control simulation by Simulink-MATLAB. DSP-Texas Instrument F28069M hardware could handle the procedure correctly at 10KHz sampling frequency. However, if the number of neurons increased, the hardware implementation process would not run correctly [39].

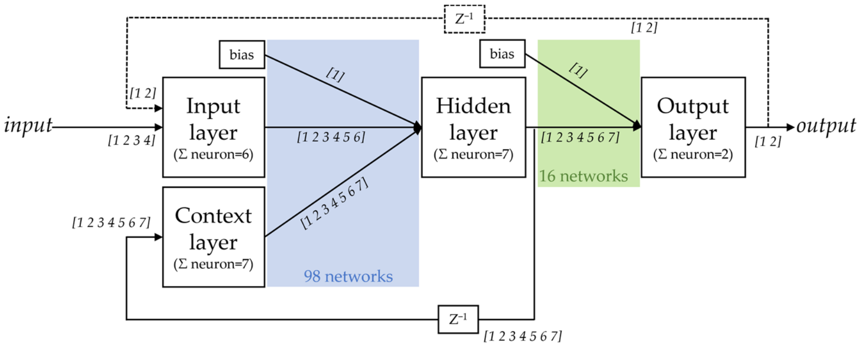


**Figure 1.** Modified Elman neural network.

In the configuration of neural networks, the more neurons, the more network connections, which means that the network's computational process also becomes longer. Although it becomes longer, this can increase the performance of neural networks. Therefore, it is necessary to consider the number of neurons [41–45]. In previous studies, the length of the computational process affected hardware performance [39].

This paper proposes an intelligent observer. The novelty of the proposed intelligent observer is its compatibility with several PMSMs with different parameters and a shorter calculation time process. A trained modified Jordan neural network (MJNN) is used as the observer's core. Figure 2 shows the MJNN configuration.

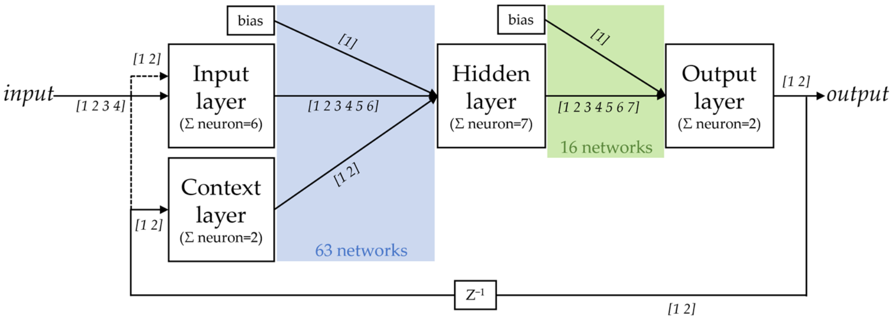


**Figure 2.** Modified Jordan neural network.

Figures 1 and 2 show the modified part in both Elman and Jordan marked with a dashed line. The output value was returned to the input layer as a part of the input parameter. Both Elman and Jordan networks have a context layer that acts as a memory. Jordan's context layer memorizes data from the output layer, while Elman's context memorizes data from the hidden layer [46]. In this study, the output layer had two neurons: *Sin* $\theta$ and *Cos* $\theta$ values. Therefore, the proposed MJNN algorithm had simpler networks; only two neurons were needed in the context layer. There were a total of 63 networks leading into the hidden layer.

Similar to earlier research, PMSM control simulation data were used as the source for the learning data in this work. The values of $I_\alpha$, $I_\beta$, $v_\alpha$, and $v_\beta$, along with *Sin* $\theta$ and *Cos* $\theta$, were recorded as the simulation data [38,39]. However, this work differs from other research that it incorporates some data from several simulations of PMSMs. This is intended to make the proposed observer smarter. Then, the initial algorithm from the successfully developed and trained ML-based observer model is merged with a phase-locked loop function (PLL) to form an intelligent observer [38,39]. The proposed intelligent observer of the sensorless PMSM drive control system was successfully evaluated in both simulation and TI F28335-based experimental hardware.

Figure 3 shows how to implement the intelligent observer in a sensorless PMSM-FOC control process. The proposed intelligent observer receives and processes $I_\alpha$, $I_\beta$, $v_\alpha$, and $v_\beta$ data from the control process. Then, the observer gives the information of rotor position $\theta_e$ and rotor speed $\omega_e$ to the process. At start-up, there are no values for $I_\alpha$, $I_\beta$, $v_\alpha$, and $v_\beta$. That is why there are two switch modes in the control process. Switch mode-1 is to activate the start-up process using an open-loop *I-f* start-up mechanism. Switch mode-2 is to start the close-loop sensorless control motor scheme [31,33,47–50].

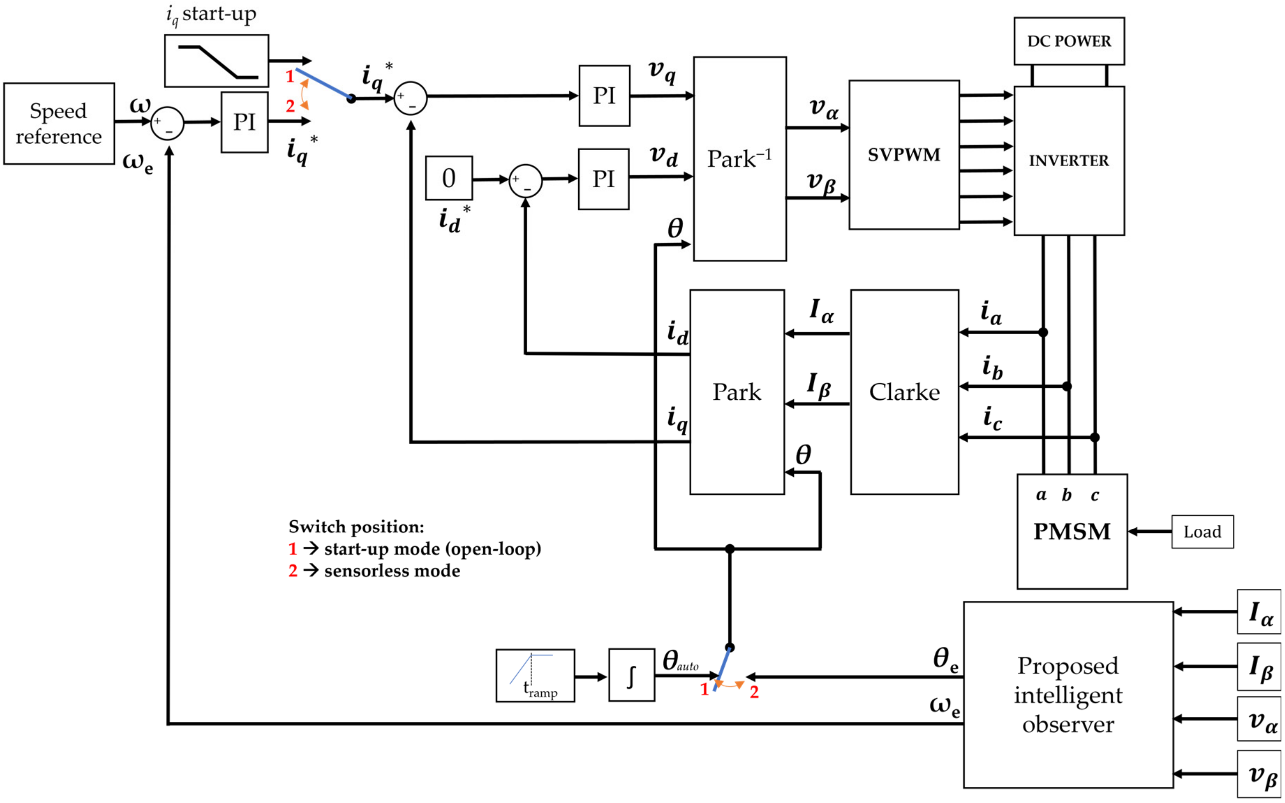


**Figure 3.** Sensorless PMSM-FOC control scheme process using a proposed intelligent observer.

As an overview, the following steps were taken to develop and realize the proposed intelligent observer:

- Developing the initial intelligent observer algorithm with the MJNN method;
- Collecting the learning data from several PMSM control simulations;
- Training and validating the ML-basing observer developed in the first step using the data obtained from the second step;

- Merging the trained ML-based observer algorithm with the PLL function to be an intelligent observer;
- Simulating the intelligent observer in the PMSM sensorless control simulation process by using Simulink-MATLAB;
- Implementing the intelligent observer into the experimental hardware platform.

The next section of this paper explains in more detail the sensorless PMSM drive system, proposed intelligent observer, simulation results, experimental hardware results, and conclusions.

## 2. Sensorless PMSM Drive System

The sensorless PMSM drive system scheme used in this paper is shown in Figure 3 [33,39]. The PMSM mathematical model is often written as Equation (1). The voltages $v_d$ and $v_q$ are the voltages on the *d-q* axis, respectively, and the resistance $r_s$ is the resistance of each stator winding. In surface-mounted PMSM, $L_s = L_d = L_q$, where $L_d$ and $L_q$ are the inductance values of the *q-d* axis; $i_d$ and $i_q$ are currents on the *d-q* axis; the rotating speed of the magnetic flux is symbolized by $\omega_e$ and the permanent magnet flux linkage is denoted by $\lambda_f$.

$$\begin{cases} v_d = r_s i_d + L_s \frac{d}{di} i_d - \omega_e L_s i_q \\ v_q = r_s i_q + L_s \frac{d}{di} i_q - \omega_e L_s i_d + \omega_e \lambda_f \end{cases} \tag{1}$$

In this study, at start-up (mode-1), the $i_q$ start-up and $\theta$ values were generated by using an *I-f* start-up mechanism. The presence of $v_q$ and $\theta$ allows the inverse park (Equations (2) and (3)) to produce $v_\alpha$ and $v_\beta$. The values of $v_\alpha$ and $v_\beta$ become references for SVPWM to activate the inverter that rotates the motor.

$$v_\alpha = v_d \, Cos\,\theta + v_q \, Sin\,\theta \tag{2}$$

$$v_\beta = v_d \, Sin\,\theta + v_q \, Cos\,\theta \tag{3}$$

When the motor starts to rotate, a current flows in each phase. The Clarke transformation function (Equations (4) and (5)) transforms this current flow value into $I_\alpha$ and $I_\beta$.

$$I_\alpha = i_a \tag{4}$$

$$I_\beta = \frac{1}{\sqrt{3}} i_a + \frac{2}{\sqrt{3}} i_b \tag{5}$$

Furthermore, the Park transformation (Equations (6) and (7)) changes $I_\alpha$ and $I_\beta$ to $i_d$ and $i_q$. In control mode-1, $i_d$ and $i_q$ values are not used as a reference control. During mode-1, the angular position $\theta$ value still refers to the angle generated by the *I-f* start-up mechanism.

$$i_d = I_\alpha \, Cos\,\theta + I_\beta \, Sin\,\theta \tag{6}$$

$$i_q = I_\alpha \, Sin\,\theta + I_\beta \, Cos\,\theta \tag{7}$$

After the motor rotates, the $v_\alpha$, $v_\beta$, $I_\alpha$ and $I_\beta$ values can be obtained. The outer close-loop PI control drives the motor speed. The inner close-loop PI control drives the currents $i_d$ and $i_q$. $i_d$ is controlled to reach the constant zero value, while $i_q$ is controlled to follow the $i_q$* value. The $i_q$* reference obtained from the speed control results in the outer closed loop. The control mode switches to mode-2 if all the *I-f* start-up mechanism parameters are met. Mode-2 control means the motor runs under sensorless control; the rotor position $\theta$ information is taken from the observer.

## 3. Proposed Intelligent Observer

The proposed intelligent observer model was developed based on a machine learning development process [51]. Figure 4 illustrates the intelligent observer development process. Simulation data from several PMSMs with different parameters were acquired and used as training and validating data. The initial algorithm of the observer model did not have observer capabilities. Some simulation data extraction was used for the intelligent observer model's training process, and other extracted data were used for the validation process. An intelligent observer model was formed at the end of this learning stage.

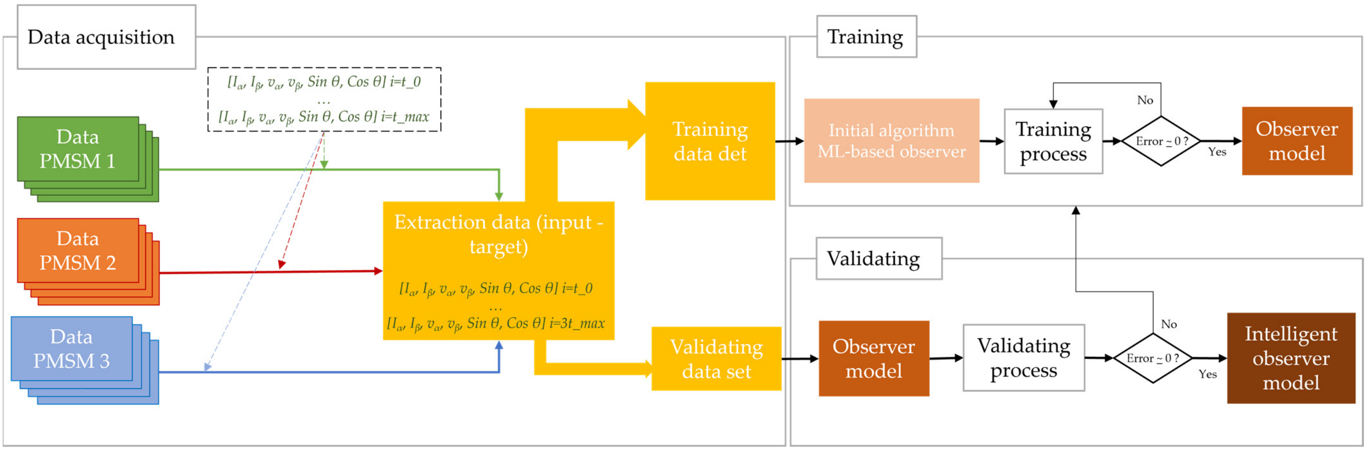

Data acquisition
[$I_\alpha$, $I_\beta$, $v_\alpha$, $v_\beta$, Sin θ, Cos θ] i=t_0
...
[$I_\alpha$, $I_\beta$, $v_\alpha$, $v_\beta$, Sin θ, Cos θ] i=t_max
Data PMSM 1
Data PMSM 2
Data PMSM 3
Extraction data (input - target)
[$I_\alpha$, $I_\beta$, $v_\alpha$, $v_\beta$, Sin θ, Cos θ] i=t_0
...
[$I_\alpha$, $I_\beta$, $v_\alpha$, $v_\beta$, Sin θ, Cos θ] i=3t_max
Training data det
Validating data set
Training
Initial algorithm ML-based observer
Training process
Error ≃ 0 ?
No
Yes
Observer model
Validating
Observer model
Validating process
Error ≃ 0 ?
No
Yes
Intelligent observer model

**Figure 4.** The development process of the intelligent observer model.

### 3.1. *Modified Jordan Neural Network*

Figure 2 is the configuration of the MJNN. If the input is denoted by $x$ and context is denoted by $c$, then the output $y$ in discrete time $k$ is a function of $y$ that can be formulated by:

$$y(k) = f(x(k), y(k-1), c(k)) \tag{8}$$

Figure 5 in this subsection illustrates the MJNN method established in this study. The input layer consists of the original input $x_i$ (where $i$ = 1 to $d$) and the feedback output $y'_j$; the output layer $y_j$; the context layer $c_j$ (where $j$ = 1 to $n$); and the hidden layer $z_p$ (where $p$ = 1 to $m$). If the layer context memory coefficients are $mch$ and $mcho$, then the value of $c_j$ is formulated by Equation (9):

$$c_j(k) = \left(y_j(k-1) \times mch\right) + \left(c_j(k-1) \times mcho\right) \tag{9}$$

Then the value of the hidden layer $z_p(k)$ is obtained by the following two formulas:

$$in\_z_p(k) = vb_p(k) + \sum_{j=1}^{n} y_j(k-1)vy'_{jp} + \sum_{i=1}^{d} x_i(k)vx_{ip} + \sum_{j=1}^{n} c_j(k)vc_{jp} \tag{10}$$

$$z_p(k) = f\left(in\_z_p(k)\right) = \frac{2}{1+e^{-(in\_z_p(k))}} - 1 \tag{11}$$

where $vb_p$ is the bias of the hidden neuron, $vy'_{jp}$ is the weight between $y'$ (feedback from the output) and hidden neuron $z$, $vx_{ip}$ is the weight between input $x$ and hidden neuron $z$, and $vc_{jp}$ is the weight between context layer $c$ and hidden neuron $z$. Equation (11) shows that the activation function used in $z_p(k)$ is the Tansig activation function [52]. Then the output of the network $y_j(k)$ is obtained using the following formula:

$$y_j(k) = wb_j(k) + \sum_{p=1}^{m} z_p(k)wz_{pj} \tag{12}$$

where $wb_j$ is the output bias of the neuron and $wz_{pj}$ is the weight between hidden neuron $z$ and output neuron $y$.

Furthermore, the training process was carried out using the same back-propagation method that applies in the back-propagation neural network, utilizing the error difference coefficient between the output and the target to update the weights in the network. The MATLAB editor was successfully used to construct and run the programs based on every stage of the MJNN algorithms.

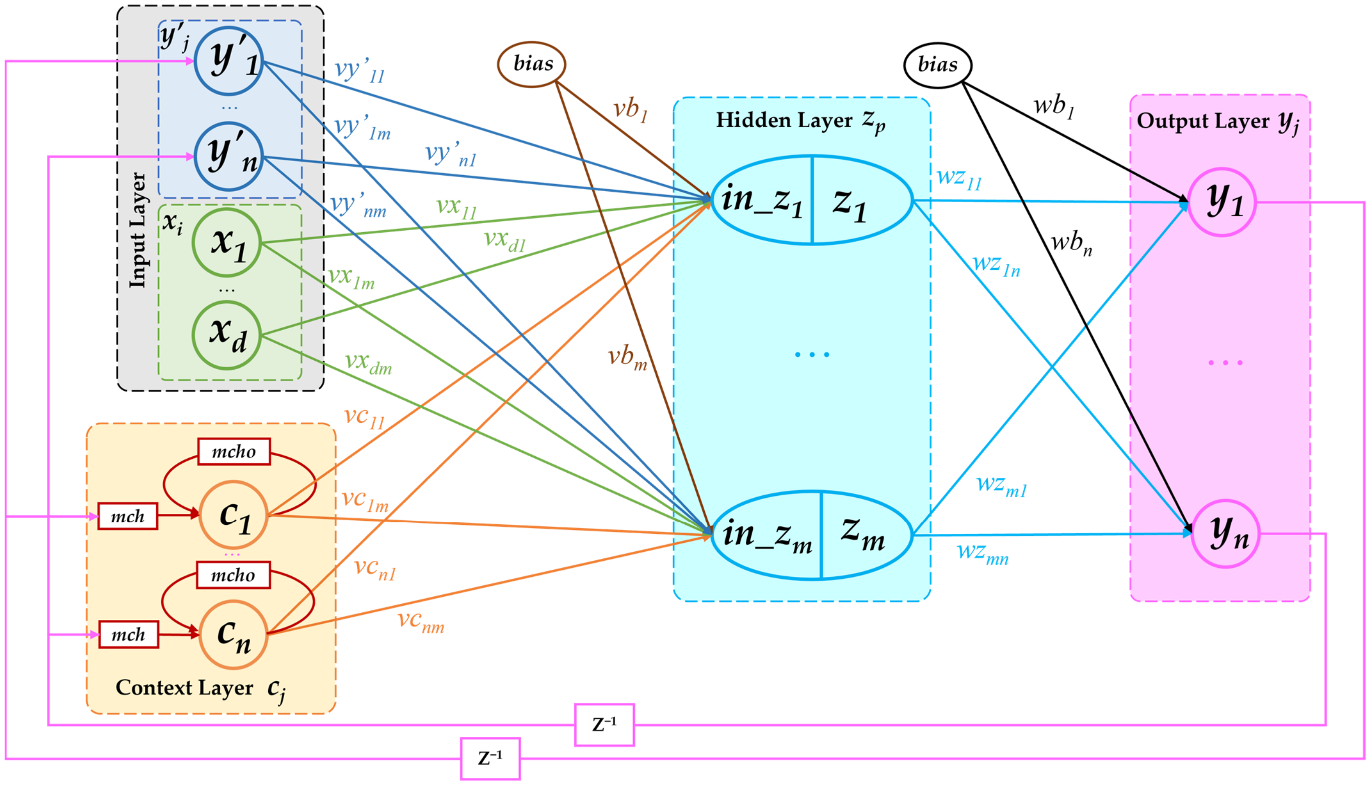


**Figure 5.** Detailed configuration of MJNN.

As previously mentioned, even though the Jordan and Elman networks were identical, the use of the Jordan network had a better processing time in the case of this research. Table 1 shows the processing time comparison between the two of them. It is the result of running ten tests to calculate the network computation time to execute 100 iterations of the training process in the same system environment. It is evidenced that the Jordan network is 22.12% faster than the Elman network.

**Table 1.** Processing time comparison.

| No | MENN Processing Time (Second) | MJNN Processing Time (Second) |
|---|---|---|
| 1 | 12.34 | 9.42 |
| 2 | 13.45 | 9.79 |
| 3 | 13.43 | 10.54 |
| 4 | 12.60 | 10.76 |
| 5 | 12.77 | 10.15 |
| 6 | 11.97 | 10.24 |
| 7 | 13.02 | 10.12 |
| 8 | 13.33 | 9.79 |
| 9 | 13.16 | 9.67 |
| 10 | 12.93 | 9.99 |
| Average | 12.90 | 10.05 |

### *3.2. Learning Process*

#### 3.2.1. Learning Data Acquisition

This research's learning data were taken from the PMSM motor control simulation instead of the experimental control process [37–40]. Based on the sensored control scheme shown in Figure 6, the simulation was developed using the motor control block library (mcb-library) in Simulink-MATLAB [53]. Data log acquisition was carried out using the Simulation Data Inspector, with the sampling occurring every 1 × $10^{-4}$ s. The values of $v_\alpha$, $v_\beta$, $I_\alpha$, and $I_\beta$, were recorded as input for the neural network training process, and the values of *Sin θ* and *Cos θ* were recorded for the targeted data training. Table 2 shows the five PMSMs used in this study. PMSM 1, PMSM 2 and PMSM 3 were used for learning and testing, and PMSM 4 and PMSM 5 were used only for testing.

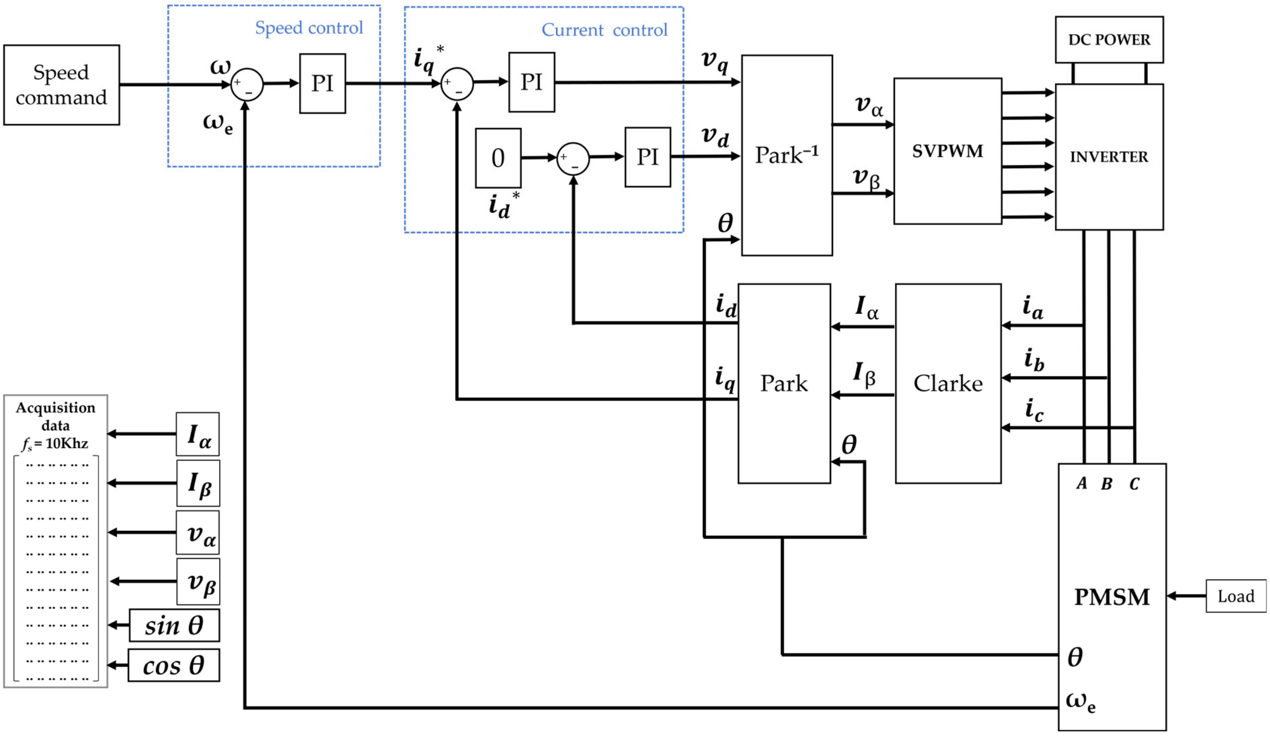


**Figure 6.** Sensored PMSM control algorithm and data acquisition.

**Table 2.** Types of PMSMs and their parameters.

| No | Motor | Power | Rate Voltage | Rate Current | Rate Speed | Used for |
|---|---|---|---|---|---|---|
| 1 | PMSM 1 | 40 W | 24 Vac | 1.80 A | 6000 rpm | Learning, testing |
| 2 | PMSM 2 | 40 W | 24 Vac | 3.50 A | 4000 rpm | Learning, testing |
| 3 | PMSM 3 | 400 W | 200 Vac | 2.80 A | 3000 rpm | Learning, testing |
| 4 | PMSM 4 | 40 W | 24 Vac | 7.10 A | 6000 rpm | Testing |
| 5 | PMSM 5 | 750 W | 220 Vac | 4.24 A | 2000 rpm | Testing |

Speed command and load variations were applied during the simulation process for each motor. For training data, the combination of speed–load set-point variations is shown in Figure 7. Meanwhile, for validating data, the combination of speed–load set-point variations is shown in Figure 8. In the current investigation, to make the computational process as efficient as possible, a per-unit system, also known as a p.u system, was utilized to scale the international value system of units or SI as p.u-values ranging from −1 to 1. For example, if the motor's maximum speed is 6000 rpm, 3000 rpm of the motor speed becomes 0.5 p.u speed, and −0.8 p.u speed means 4800 rpm in the reverse direction [54].

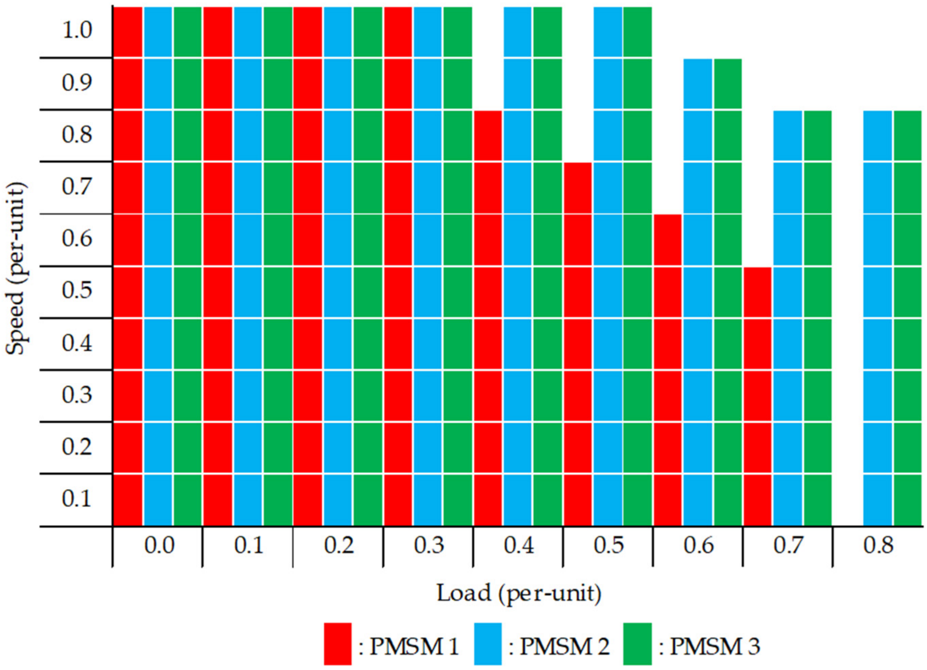


**Figure 7.** Speed–load set-point combination for training data acquisition.

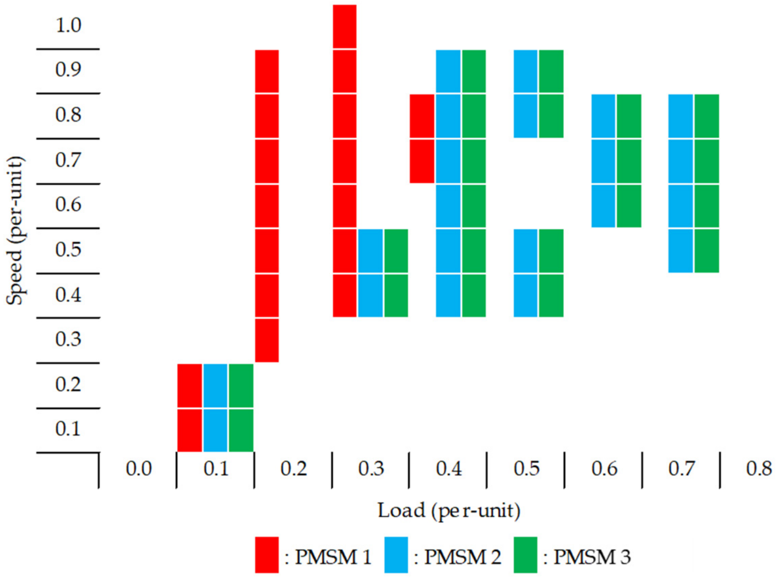


**Figure 8.** Speed–load set-point combination for validating data acquisition.

### 3.2.2. Training and Validating Process

All three PMSM simulation data acquired were combined and used to train the initial model of the MJNN. The total data training comprised 2,708,576 rows of data. The offline training process was performed with learning rate = 0.002, and after 27,347 epochs, the final MSE value reached $5 \times 10^{-6}$. At the end of the training process, the final values of weights and biases were stored as the model's core. In order to validate the model, another 600,000 rows of simulation data were used. The histogram errors of both the training and validating process are shown in Figures 9 and 10. From these two error histograms, it can be concluded that the MJNN model can respond well to the input given.

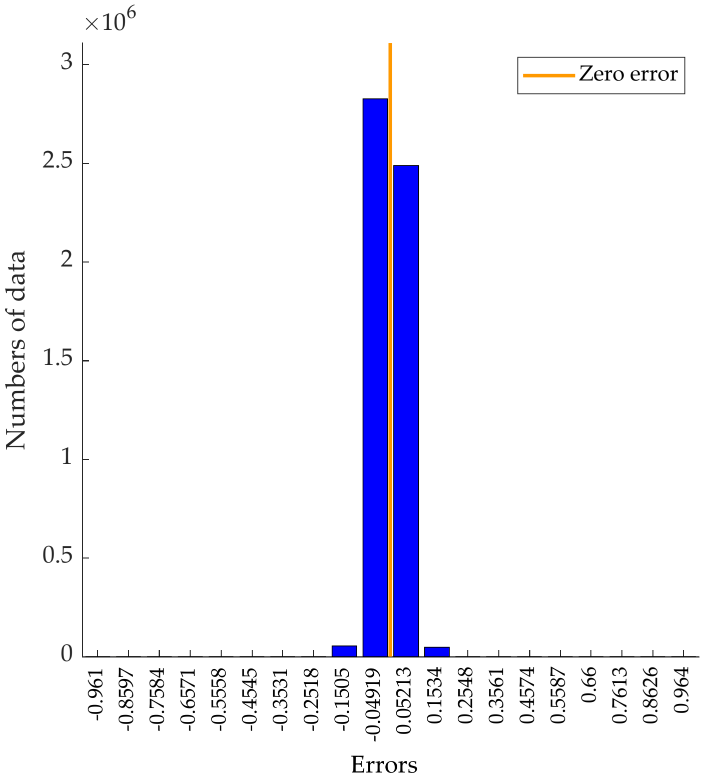


**Figure 9.** Error histogram of the training process.

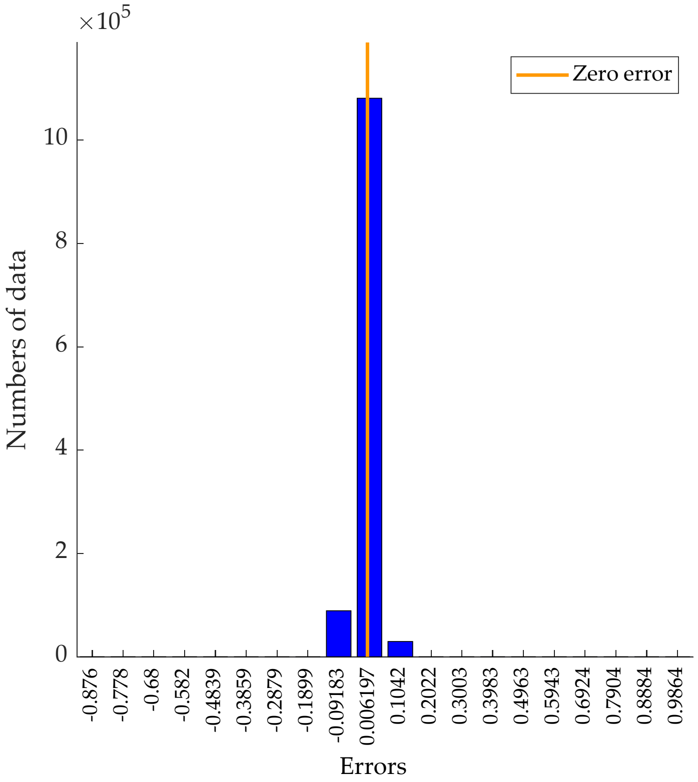


**Figure 10.** Error histogram of the validating process.

### *3.3. Intelligent Observer Function Block*

After training and validating, the intelligent observer could precisely estimate the values of *Sin* $\theta$ and *Cos* $\theta$ from the input given ($v_\alpha$, $v_\beta$, $I_\alpha$ and $I_\beta$). In addition, PLL was added to the model. The result of the combined structure, as shown in Figure 11, was an intelligent observer function block that generates the position $\theta_e$ and speed feedback $\omega_e$ information. This function block was then utilized as an observer in the sensorless PMSM control scheme shown in Figure 3.

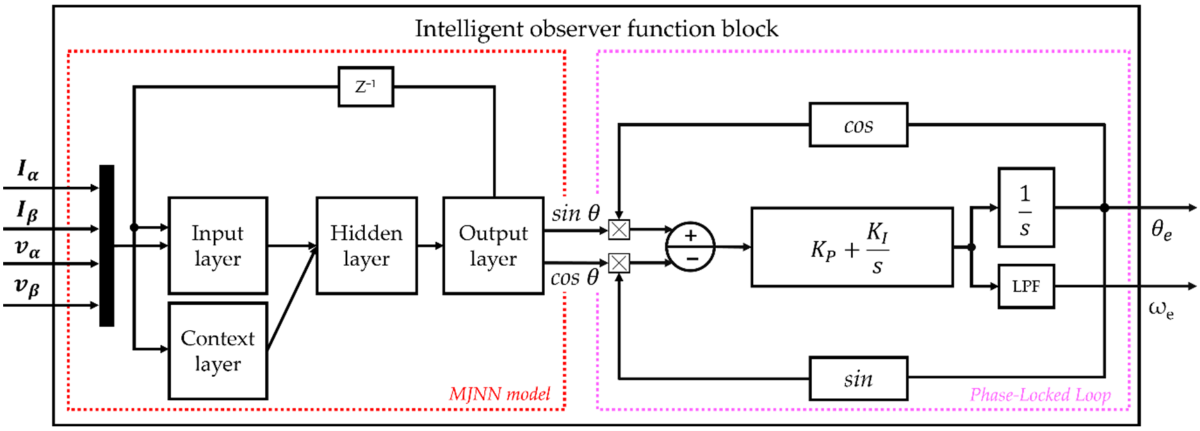


**Figure 11.** Intelligent observer function block.

## 4. Simulation Results and Discussions

In the next step, the proposed observer was implemented in the PMSM-FOC sensorless control process simulation to validate its observer function performance in the complete control process environment. All PMSMs in Table 2 were used in the simulation control process. The simulation was run with Simulink-MATLAB, and the results were logged with Simulation Data Inspector. As mentioned, the I-f start-up mechanism started the motor from zero to a certain constant speed using an open loop and then switched it to closed loop sensorless mode using the proposed observer.

### 4.1. Simulation Results of Rotor Position Information

Based on Figure 11, the proposed observer generated rotor position and speed feedback information. Figure 12 shows the simulation result samples for the rotor position generated by the proposed observer compared with the position reference. Figure 12a–c are the results for the learning source PMSMs. Figure 12d,e are for non-learning PMSMs. Table 3 shows the average position error for each PMSM. The rotor positions generated by the observer have small errors compared to the position reference, 0.0078 p.u on average. This fact confirms that the observer can generate rotor position information correctly.

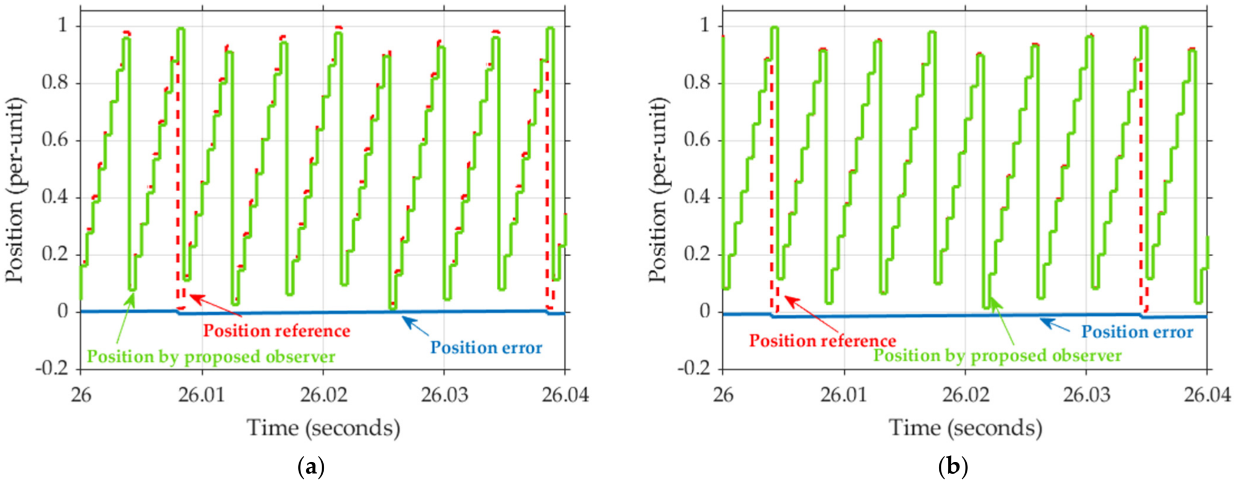
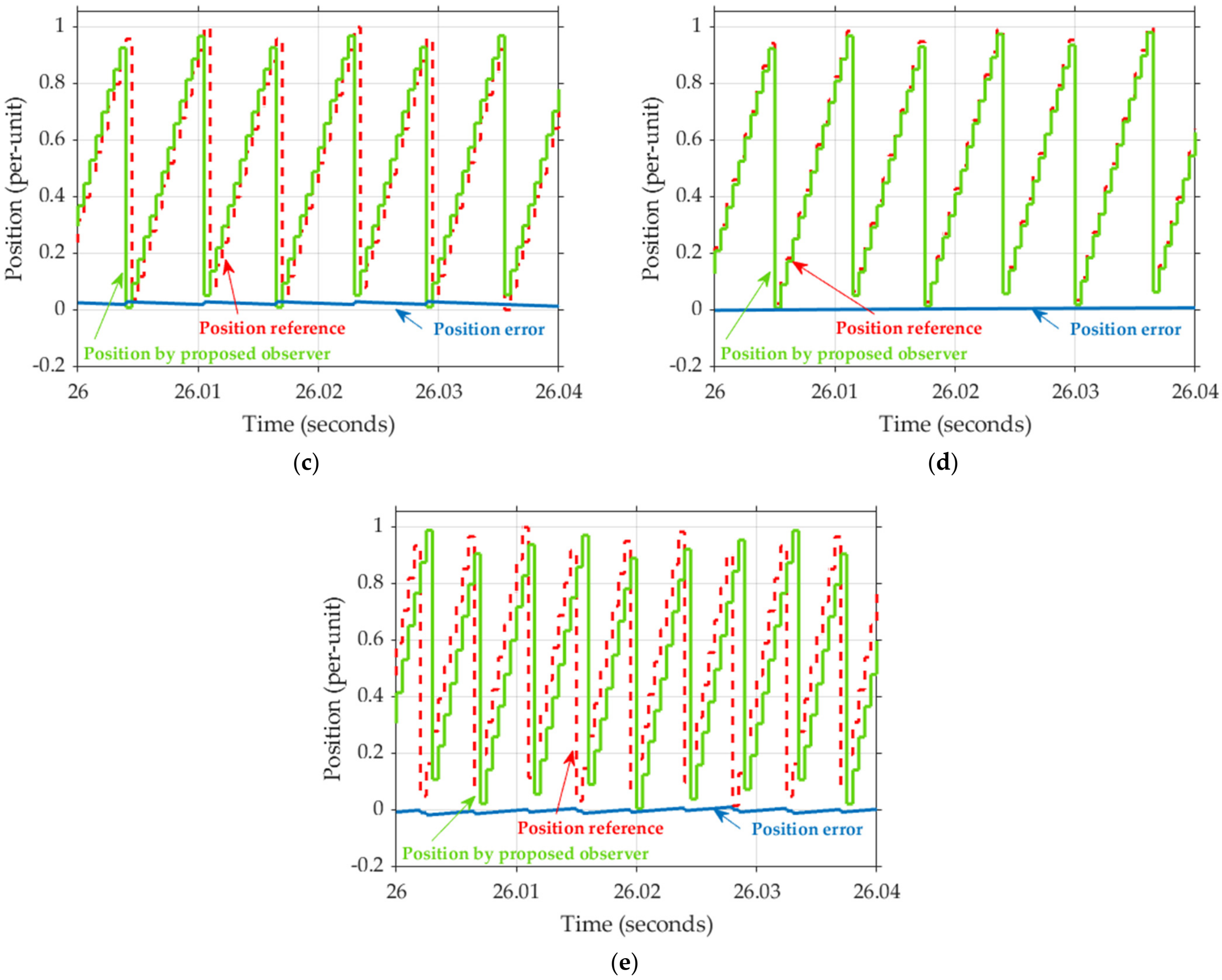


**Figure 12.** *Cont.*



(**c**)

(**d**)

(**e**)

**Figure 12.** Simulation results for the rotor positions (**a**) PMSM 1; (**b**) PMSM 2; (**c**) PMSM 3; (**d**) PMSM 4; and (**e**) PMSM 5.

**Table 3.** Position error.

| No | Motor | Average Error Position (p.u) |
|---|---|---|
| 1 | PMSM 1 | 0.0069 |
| 2 | PMSM 2 | 0.0062 |
| 3 | PMSM 3 | 0.0089 |
| 4 | PMSM 4 | 0.0081 |
| 5 | PMSM 5 | 0.0088 |
|  | Average | 0.0078 |

### *4.2. Simulation Results of Rotor Speed Feedback Information*

Speed feedback information generated by the proposed observer is discussed in this sub-section. The simulation used per-unit speed values, and just two PMSM simulation results were used as a representation. This was to make the performance comparisons clear. In the control simulation, by using the I-f start-up mechanism, the motor started in open-loop mode and then switched to the closed-loop sensorless mode. The proposed observer was used during this sensorless mode. Figure 13 shows the comparison of speed reference and speed feedback information generated by the observer for PMSM 1 (as a learning source PMSM) and PMSM 4 (as a non-learning source PMSM). The speed feedback from the two PMSMs control processes could smoothly follow the speed reference, as shown in Figure 13. In more detail, for low, middle, and high-speed ranges, the proposed observer had a better speed feedback response for PMSM 1 than for PMSM 4. However, because the gap was very small, there was no significant difference in the overall PMSM control process. It can be emphasized that the observer performs well in generating speed feedback information.

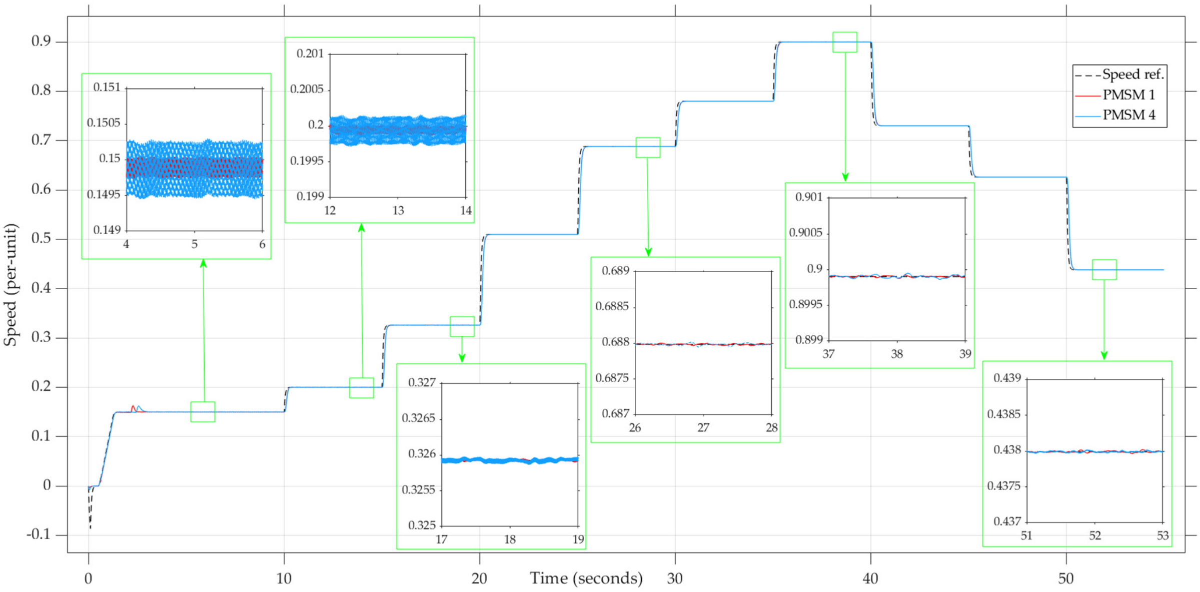


**Figure 13.** Simulation result for speed feedback generated by proposed observers PMSM 1 and PMSM 4.

### 4.3. Simulation Results of Rotor Speed Control

The proposed observer's good performance in generating rotor position and speed feedback information guarantees the success of the motor control process. Figure 14 shows the simulation results for all PMSM rotor speed responses. All the rotor speeds could smoothly track the reference speed in sensorless mode. The results show that the rotor speed response could follow speed reference in constant speed, speed-up transition and speed-down transition. This demonstrates that, in general, the proposed observer can work well in the PMSM-FOC sensorless control process for learning source PMSMs and non-learning source PMSMs.

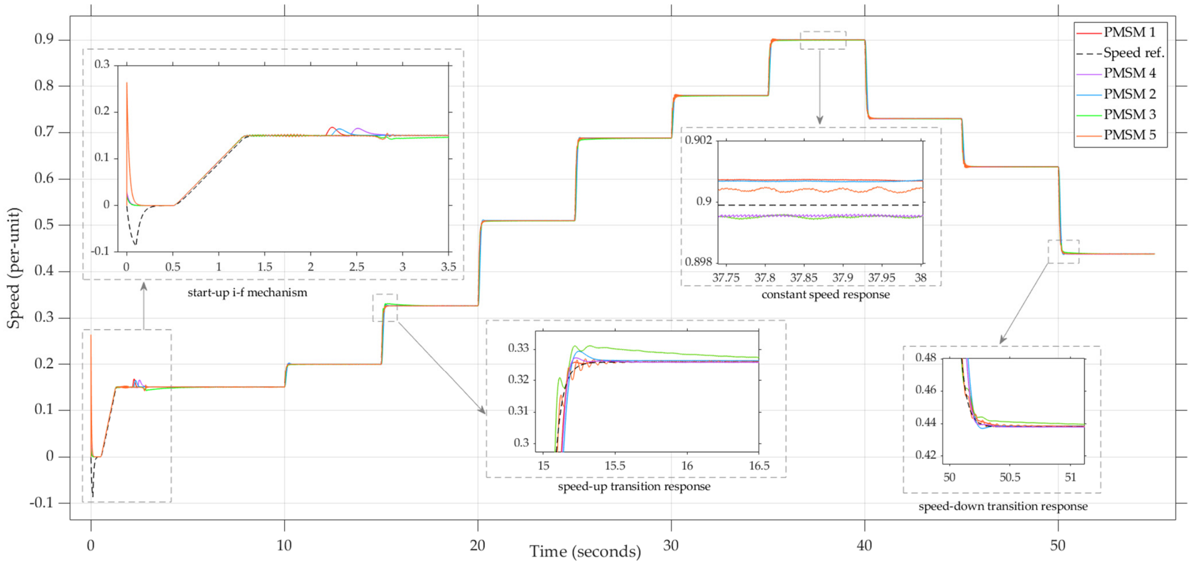
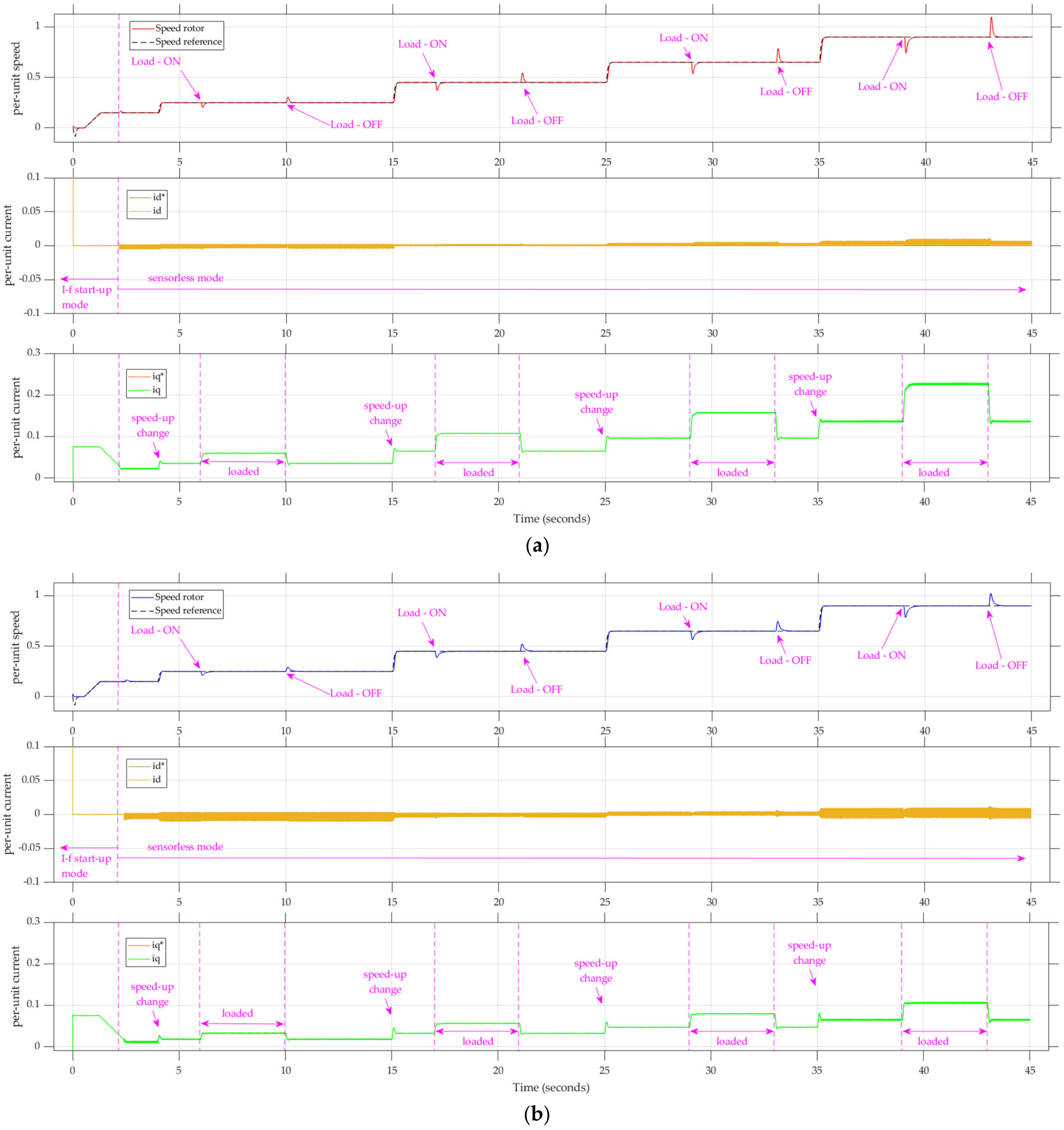


**Figure 14.** Rotor speed simulation results for all PMSM.

### *4.4. Simulation Results of the Current Response and External Load Robustness*

The proposed observer was trained in the learning stage using various speeds and load data training (Figure 7). Previous simulation results show that the proposed observer performs well in variable speed control. The simulation results in this subsection aim to know the performance of the proposed observer in response to load change. For this reason, the *d*-*q* current response was observed during the load variation. The motor was run in simulation in step-up speeds from 0.15 to 0.25; 0.45; 0.65; and 0.90 p.u speed. Load given was to the motor at 0.25 p.u speed ($t$ = 6 s to 10 s); 0.45 p.u speed (at 17 s to 21 s); 0.65 p.u speed (29 s to 33 s); and 0.90 p.u speed (39 s to 43 s). Figure 15 shows the speed and *d*-*q* current value in response to the load given to the PMSM control system.



(**a**)

(**b**)

**Figure 15.** Speed and *d*-*q* current response of (**a**) PMSM 1 and (**b**) PMSM 4.

From the speed graphs, both PMSM 1 and PMSM 4 control systems can handle the load well. They have a small speed spike down for a moment when the load is ON. Contrary, a speed spike up occurs when the load is OFF. This demonstrates that the applied PI parameters can effectively handle the load and the proposed observer can adapt to this controlling behavior. The current $i_d$ value was maintained well, close to the $i_d$* reference (zero). The value of the current $i_q$ increased following the increasing value of the current reference $i_q$*, which was proportional to the increase in speed and load. There was a small current spike up in response to the speed up of the rotor. When the load was activated, the currents of $i_q$ and $i_q$* increased. When the load was released, the current values of $i_q$ and $i_q$* were reduced. All these current responses indicate that the control process ran as expected. This verifies that the proposed observer works as expected when the workload fluctuates.

## 5. Experimental Results and Discussions

Previously, the proposed observer implementation simulation process has shown good results for controlling learning-source PMSMs and non-learning-source PMSMs. This section discusses the implementation of the proposed observer in real hardware realization. A Texas Instrument TMS320F28335 was used for digital signal processing. This microcontroller (MCU) had a 150 MHz frequency and 512 KB memory. This MCU was integrated into a control card attached to a High Voltage Motor Control Developer's Kit. A non-learning source PMSM was deployed as a tested PMSM; it was PMSM 5 from Table 2. This PMSM 5 was coupled to a generator connected to the controllable electrical load. The experimental control process was carried out and monitored using a Simulink-MATLAB Host File and Code Composer Studio (CCS). The MCU, Simulink, and CSS data communication allowed for real-time monitoring and data logging. The experimental hardware realization used SI units instead of per-unit systems to monitor the real values of speeds and currents. Figure 16 shows the experimental hardware realization setup. Table 4 shows the experimental hardware specifications.

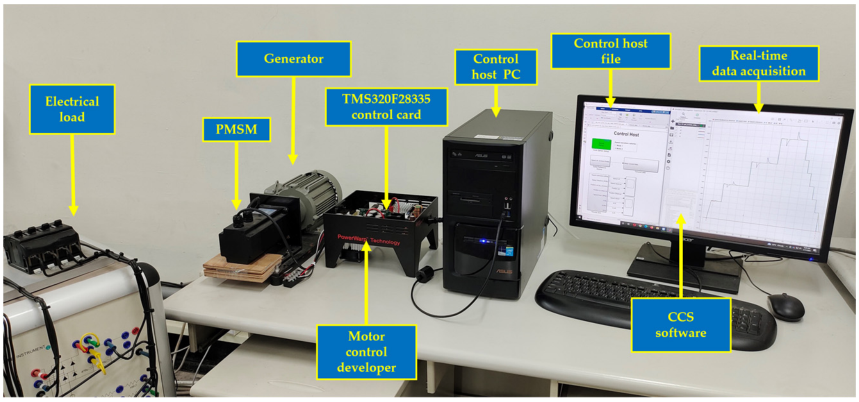


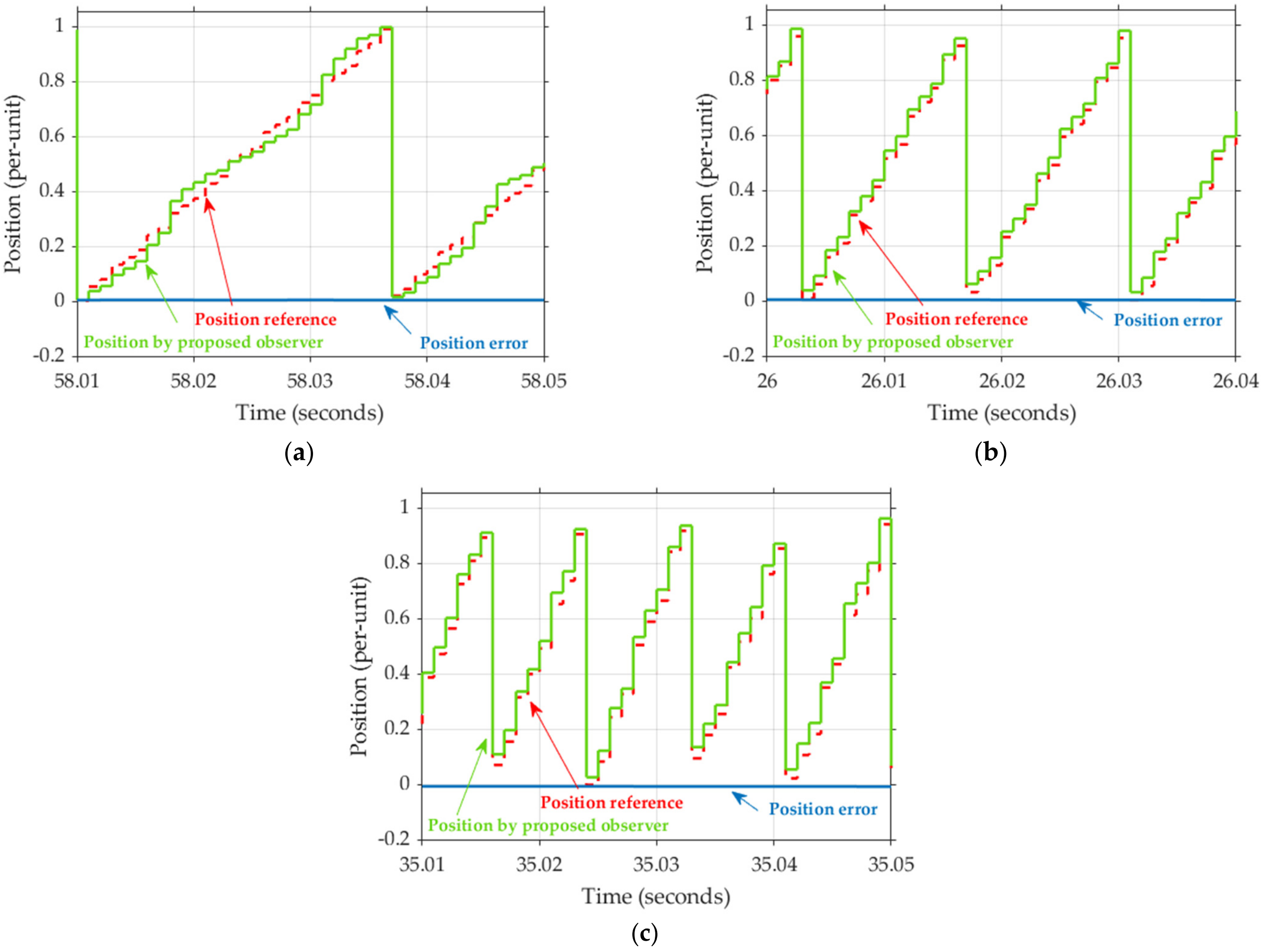
**Figure 16.** Experimental hardware realization.

**Table 4.** Experimental hardware tools and devices.

| No | Hardware | Specification |
|---|---|---|
| 1 | PMSM | 750 W; 4.24 A; 220 Vac; 2000 rpm |
| 2 | Motor control developer | Texas Instrument TMDSHVMTRPFCKIT |
| 3 | Control card | TMS320F28335 |
| 4 | Generator | 750 W |
| 5 | Electrical load | Configurable capacitors and resistors |
| 6 | MATLAB | Version 2021b with academic license |
| 7 | Code Composer Studio | Version 9.1.0 |

### 5.1. Experimental Results for Speed Variation

In this experiment, we aimed to know the rotor position (Figure 17) and speed feedback (Figure 18) information generated by the proposed observer in the real hardware implementation. Figure 17 shows the PMSM 5 was controlled from zero speed to 300 rpm using an open loop, and using the *I-f* start-up mechanism, the control switched to the closed-loop sensorless mode. The experimental results for the position and speed feedback information were taken in the sensorless mode when the proposed observer was used in the control process. Figure 16 shows the experimental results for the rotor position information generated by the proposed observer. Compared with the rotor position reference, the rotor position information generated by the proposed observer had small errors in all speed stages (400 rpm, 760 rpm, and 1265 rpm). The average error position was 0.0100 p.u position. This indicates that, in the experimental hardware, the observer worked well in generating rotor position information.



(**a**) (**b**) (**c**)

**Figure 17.** Experimental results for the rotor position at (**a**) 400 rpm, (**b**) 760 rpm and (**c**) 1265 rpm.

Figure 18 shows in the sensorless mode, the proposed observer produced speed feedback information close to the speed reference. This occurred in all control conditions: the constant speed, the speed-up transition, and the speed-down transition. This fact means the rotor speed can follow the speed reference as well. It can be emphasized that, in the experimental hardware, the proposed observer worked well in terms of speed variation control.

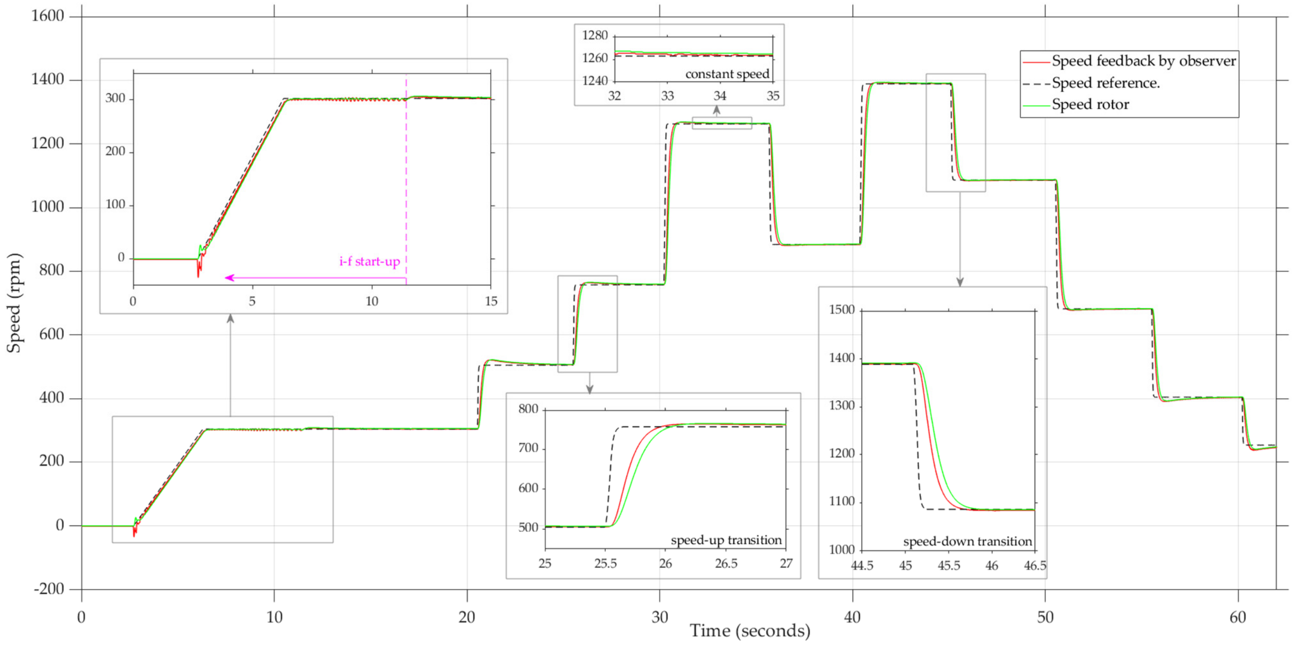


**Figure 18.** Experimental results for speed feedback generated by the proposed observer.

### 5.2. Experimental Results for Load Variation

In this sub-section, we aim to uncover the speed and *d*-*q* current characteristics in response to the load addition in the real hardware implementation. The load was given at low, medium, and high speeds. Figure 19 shows the results. For the speed characteristic, there were speed spikes down when the load was ON and up when the load was OFF. This shows that, despite the PI control speed settings, the observer properly generated speed feedback information in the load-ON and load-OFF conditions.

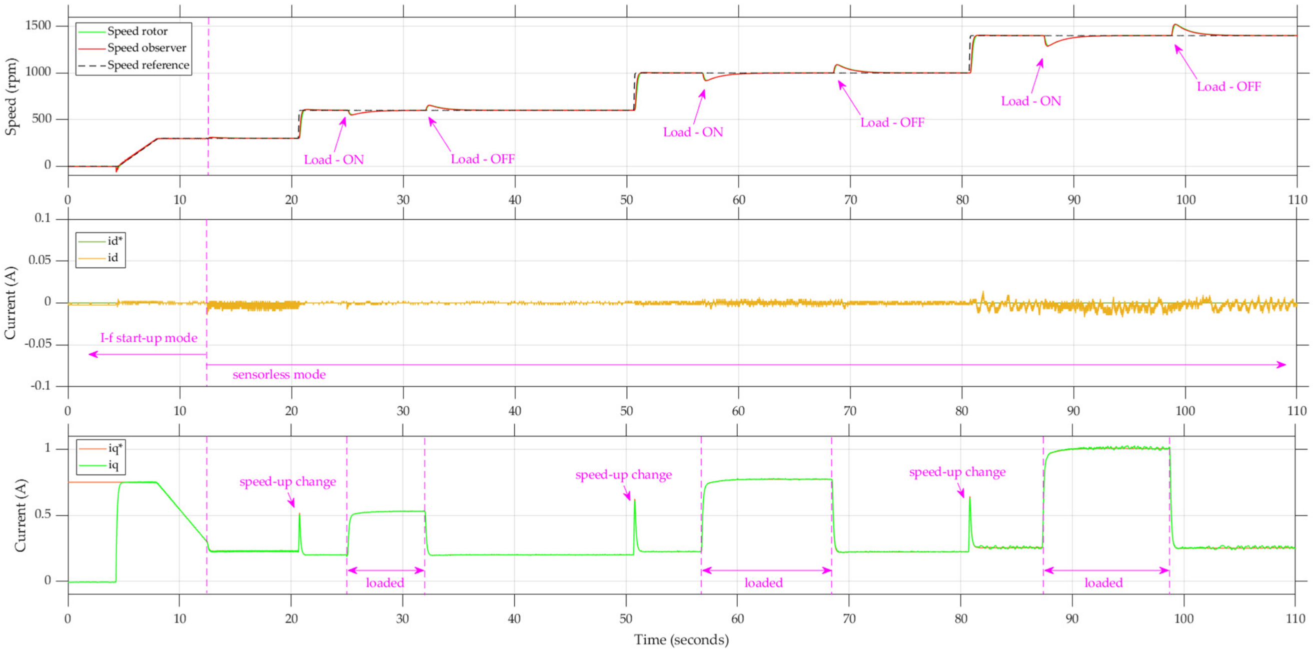


**Figure 19.** Experimental results for speed and current response to load variation.

Figure 19 also shows spike currents in $i_q$* and $i_q$ in every speed-up change. These current values also increased at higher speeds and when the load was applied. The other fact is the $i_d$ current was well controlled close to the $i_d$* reference value. Our experimental results show that the proposed observer can be used properly to control a PMSM, even if it is a non-learning source PMSM.

Based on simulations and experimental results, the proposed observer in this study is consistent with other studies that used a sliding mode observer [31,33,33] and machine-learning-based observers [38,39]. The proposed observer can accurately estimate position. Table 5 shows a comparison of the proposed observer's performance with those in previous studies. As can be seen, the proposed observer performs better than those of previous studies since its position error is lower than in previous studies.

**Table 5.** Position error comparison.

| No | Result | Position Error (p.u-Position) | |
|---|---|---|---|
| | | Previous Study [39] | Proposed Observer |
| 1 | Simulation | 0.0127 | 0.0078 |
| 2 | Experimental | 0.0607 | 0.0100 |

The proposed control process also provides a good response when the speed and load are changed. What sets it apart from previous research is that in a sliding mode observer-based study, the SMO parameter set must be changed according to the PMSM used. Similarly, there is only one PMSM that can use machine-learning-based observers. In this study, PMSMs with different parameters could be controlled using the same proposed observer.

## 6. Conclusions

An intelligent observer was successfully designed and implemented into the PMSM control process both in simulation and experimental hardware. The "brain" of this intelligent observer is a MJNN. This algorithm was found to be 22% faster than the previous algorithm. The learning process of machine learning with some PMSM control simulation data makes the observer capable of being used to control several PMSMs, even PMSMs that are not a source of learning.

Variations in speed and load were utilized to evaluate the observer's effectiveness in the PMSM control process. All testing, including simulation and experimental hardware, produced accurate outcomes. Simulation revealed that the observer's position information had variation position errors from 0.0061 p.u to 0.0089 p.u position. As for the experimental hardware realization, the inaccuracy was 0.0100 p.u position.

**Author Contributions:** Conceptualization, D.S.P. and S.-C.C.; formal analysis, D.S.P. and S.-C.C.; funding acquisition, C.-F.C.; investigation, D.S.P. and H.-H.K.; methodology, D.S.P. and S.-C.C.; software, D.S.P. and H.-H.K.; writing—original draft, D.S.P. Resources, S.-C.C. and C.-F.C.; supervision, S.-C.C., validation, S.-C.C., H.-H.K. and C.-F.C. Writing—review and editing, S.-C.C., H.-H.K. and C.-F.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Fukuta Electric and Machinery Co., Ltd., Taiwan, under contract No. 14001110328.

**Data Availability Statement:** Not applicable.

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

## Nomenclature

| | |
|---|---|
| $I_\alpha$ and $I_\beta$ | The currents generated by Clarke transform in the FOC process |
| $v_\alpha$ and $v_\beta$ | The voltages generated by Park inverse transform in the FOC process |
| $\theta$ | The rotor position angle |
| $\omega$ | The rotor speed |
| $i_d$ and $i_q$ | The currents on the d-q axis |
| $v_d$ and $v_q$ | The voltages on the d-q axis |

| | |
|---|---|
| $k$ | Discrete time |
| $x$ | Input layer's neurons |
| $y$ | Output layer's neurons |
| $y'$ | Feedback from the output layer's neurons |
| $z$ | Hidden layer's neurons |
| $c$ | Context layer's neurons |
| $mch$ and $mcho$ | The context layer's memory coefficients |
| $vb$ | The bias of the hidden neuron |
| $vx$ | The weights between input layer's neurons and hidden layer's neurons |
| $vy'$ | The weights between feedback neurons and hidden layer's neurons |
| $vc$ | The weights between context layer's neurons and hidden layer's neurons |
| $wb$ | The bias of the output neurons |
| $wz$ | The weights between hidden layer's neurons and output layer's neurons |
| $Z^{-1}$ | Delay |

## References

1. Rauth, S.S.; Samanta, B. Comparative Analysis of IM/BLDC/PMSM Drives for Electric Vehicle Traction Applications Using ANN-Based FOC. In Proceedings of the 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 10–13 December 2020; pp. 1–8.
2. Sakunthala, S.; Kiranmayi, R.; Mandadi, P.N. A Study on Industrial Motor Drives: Comparison and Applications of PMSM and BLDC Motor Drives. In Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 1–2 August 2017; pp. 537–540.
3. Derammelaere, S.; Haemers, M.; De Viaene, J.; Verbelen, F.; Stockman, K. A Quantitative Comparison between BLDC, PMSM, Brushed DC and Stepping Motor Technologies. In Proceedings of the 2016 19th International Conference on Electrical Machines and Systems (ICEMS), Chiba, Japan, 13–15 November 2016; pp. 1–5.
4. Park, S.-Z.; Kim, Y.-K.; Song, C.-H.; Lee, J.-W.; Mok, H.-S. Operation Method of Electric Bicycle Using Change of BLDC Operation Mode and PMSM Operation Mode. In Proceedings of the 8th International Conference on Power Electronics–ECCE Asia, Jeju, Republic of Korea, 30 May–3 June 2011; pp. 2529–2536.
5. Badini, S.S.; Kumar, C.D.; Verma, V. Single Sensor Based Vector Controlled PMSM Drive. In Proceedings of the 2020 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), Jaipur, India, 16–19 December 2020; pp. 1–5.
6. Kumar, C.D.; Shiva, B.S.; Verma, V. Vector Control of PMSM Drive with Single Current Sensor. In Proceedings of the 2020 IEEE Students Conference on Engineering & Systems (SCES), Prayagraj, India, 10–12 July 2020; pp. 1–6.
7. Bi, Q.; Shao, D.; Li, J. Signal Correction of Linear Hall for PMSM Control System. In Proceedings of the 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 11–13 October 2019; pp. 341–345.
8. Nauduri, B.S.; Shaga, G. A Novel Approach of Using a Planar Inductive Position Sensor for the Permanent Magnet Synchronous Motor Control Application. In Proceedings of the 2018 IEEE Sensors Applications Symposium (SAS), Seoul, Republic of Korea, 12–14 March 2018; pp. 1–5.
9. Xiong, Y.; Wang, A.; Zhang, T. Sensor-Less Complex System Control of PMSM Based on Improved SMO. In Proceedings of the 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 15–17 July 2021; pp. 228–232.
10. Nicola, M.; Nicola, C.-I. Sensorless Control of PMSM Using SMC and Sensor Fault Detection Observer. In Proceedings of the 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), Monastir, Tunisia, 22–25 March 2021; pp. 518–525.
11. Yan, H.; Wang, W.; Xu, Y.; Zou, J. Position Sensorless Control for PMSM Drives With Single Current Sensor. *IEEE Trans. Ind. Electron.* **2023**, *70*, 178–188. [CrossRef]
12. Wei, X.; Yan, L.; Jin, Z.; Zhou, Y.; Xiang, P. SPMSM Deadbeat Predictive Control with HF Injection Sensorless Rotor Position Estimate. In Proceedings of the 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 16–19 December 2022; pp. 920–925.
13. Wu, L.; Lyu, Z.; Chen, Z.; Liu, J.; Lu, Y. An Enhanced Sensorless Control Scheme for PMSM Drives Considering Self-Inductance Asymmetry. *CES Trans. Electr. Mach. Syst.* **2022**, *6*, 384–392. [CrossRef]
14. Yao, G.; Xing, L.; Huang, Y. Sensorless Control Simulation of Permanent Magnet Synchronous Motor Based on Sliding Mode Observer. In Proceedings of the 2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, 18–20 November 2022; pp. 688–693.
15. Nahid-Mobarakeh, B.; Meibody-Tabar, F.; Sargos, F.-M. Back EMF Estimation-Based Sensorless Control of PMSM: Robustness With Respect to Measurement Errors and Inverter Irregularities. *IEEE Trans. Ind. Appl.* **2007**, *43*, 485–494. [CrossRef]
16. Genduso, F.; Miceli, R.; Rando, C.; Galluzzo, G.R. Back EMF Sensorless-Control Algorithm for High-Dynamic Performance PMSM. *IEEE Trans. Ind. Electron.* **2010**, *57*, 2092–2100. [CrossRef]
17. Wang, Z.; Lu, K.; Blaabjerg, F. A Simple Startup Strategy Based on Current Regulation for Back-EMF-Based Sensorless Control of PMSM. *IEEE Trans. Power Electron.* **2012**, *27*, 3817–3825. [CrossRef]

18. Bi, G.; Zhang, G.; Wang, Q.; Ding, D.; Li, B.; Wang, G.; Xu, D. High-Frequency Injection Angle Self-Adjustment Based Online Position Error Suppression Method for Sensorless PMSM Drives. *IEEE Trans. Power Electron.* **2023**, *38*, 1412–1417. [CrossRef]
19. Choi, C.-H.; Seok, J.-K. Compensation of Zero-Current Clamping Effects in High-Frequency-Signal-Injection-Based Sensorless PM Motor Drives. *IEEE Trans. Ind. Appl.* **2007**, *43*, 1258–1265. [CrossRef]
20. Li, H.; Zhang, X.; Yang, S.; Liu, S. Unified Graphical Model of High-Frequency Signal Injection Methods for PMSM Sensorless Control. *IEEE Trans. Ind. Electron.* **2020**, *67*, 4411–4421. [CrossRef]
21. Liu, J.M.; Zhu, Z.Q. Novel Sensorless Control Strategy With Injection of High-Frequency Pulsating Carrier Signal Into Stationary Reference Frame. *IEEE Trans. Ind. Appl.* **2014**, *50*, 2574–2583. [CrossRef]
22. Wang, S.; Yang, K.; Chen, K. An Improved Position-Sensorless Control Method at Low Speed for PMSM Based on High-Frequency Signal Injection into a Rotating Reference Frame. *IEEE Access* **2019**, *7*, 86510–86521. [CrossRef]
23. Trancho, E.; Ibarra, E.; Arias, A.; Kortabarria, I.; Prieto, P.; Martínez de Alegría, I.; Andreu, J.; López, I. Sensorless Control Strategy for Light-Duty EVs and Efficiency Loss Evaluation of High Frequency Injection under Standardized Urban Driving Cycles. *Appl. Energy* **2018**, *224*, 647–658. [CrossRef]
24. Bolognani, S.; Oboe, R.; Zigliotto, M. Sensorless Full-Digital PMSM Drive with EKF Estimation of Speed and Rotor Position. *IEEE Trans. Ind. Electron.* **1999**, *46*, 184–191. [CrossRef]
25. Termizi, M.S.; Lazi, J.M.; Ibrahim, Z.; Talib, M.H.N.; Aziz, M.J.A.; Ayob, S.M. Sensorless PMSM Drives Using Extended Kalman Filter (EKF). In Proceedings of the 2017 IEEE Conference on Energy Conversion (CENCON), Kuala Lumpur, Malaysia, 30–31 October 2017; pp. 145–150.
26. Nak, H.; Gülbahce, M.O.; Gokasan, M.; Ergene, A.F. Performance Investigation of Extended Kalman Filter Based Observer for PMSM Using in Washing Machine Applications. In Proceedings of the 2015 9th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, 26–28 November 2015; pp. 618–623.
27. Quang, N.K.; Hieu, N.T.; Ha, Q.P. FPGA-Based Sensorless PMSM Speed Control Using Reduced-Order Extended Kalman Filters. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6574–6582. [CrossRef]
28. Bolognani, S.; Tubiana, L.; Zigliotto, M. Extended Kalman Filter Tuning in Sensorless PMSM Drives. *IEEE Trans. Ind. Appl.* **2003**, *39*, 1741–1747. [CrossRef]
29. Lu, Q.; Quan, L.; Zhu, X.; Zuo, Y.; Wu, W. Improved Sliding Mode Observer for Position Sensorless Open-Winding Permanent Magnet Brushless Motor Drives. *Prog. Electromagn. Res. M* **2019**, *77*, 147–156. [CrossRef]
30. An, Q.; An, Q.; Liu, X.; Zhang, J.; Bi, K. Improved Sliding Mode Observer for Position Sensorless Control of Permanent Magnet Synchronous Motor. In Proceedings of the 2018 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific), Piscataway, NJ, USA, 6–9 June 2018; pp. 1–7.
31. Kung, Y.-S.; Risfendra, R.; Lin, Y.-D.; Huang, L.-C. FPGA-Realization of a Sensorless Speed Controller for PMSM Drives Using Novel Sliding Mode Observer. *Microsyst. Technol.* **2018**, *24*, 79–93. [CrossRef]
32. Kung, Y.-S.; Risfendra; Lin, Y.-D.; Huang, L.-C. FPGA-Based Sensorless Controller for PMSM Drives Using Sliding Mode Observer and Phase Locked Loop. In Proceedings of the 2016 International Conference on Applied System Innovation (ICASI), Okinawa, Japan, 26–30 May 2016; pp. 1–4.
33. Hoai, H.-K.; Chen, S.-C.; Than, H. Realization of the Sensorless Permanent Magnet Synchronous Motor Drive Control System with an Intelligent Controller. *Electronics* **2020**, *9*, 365. [CrossRef]
34. Hoai, H.-K.; Chen, S.-C.; Chang, C.-F. Realization of the Neural Fuzzy Controller for the Sensorless PMSM Drive Control System. *Electronics* **2020**, *9*, 1371. [CrossRef]
35. Zhang, Z.; Kong, L.; Wu, Y.; Zhang, S.; Liu, Z. PMSM Sensorless Control Based on Adaptive Luenberger Observer. In Proceedings of the 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Beijing, China, 19–20 November 2022; pp. 398–403.
36. Lagrioui, A.; Mahmoudi, H. Speed and Current Control for the PMSM Using a Luenberger Observer. In Proceedings of the 2011 International Conference on Multimedia Computing and Systems, Ouarzazate, Morocco, 7–9 April 2011; pp. 1–6.
37. Makni, Z.; Zine, W. Rotor Position Estimator Based on Machine Learning. In Proceedings of the IECON 2016 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 6687–6692.
38. Zine, W.; Makni, Z.; Monmasson, E.; Idkhajine, L.; Condamin, B. Interests and Limits of Machine Learning-Based Neural Networks for Rotor Position Estimation in EV Traction Drives. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1942–1951. [CrossRef]
39. Putra, D.S.; Chen, S.-C.; Khong, H.-H.; Cheng, F. Design and Implementation of a Machine-Learning Observer for Sensorless PMSM Drive Control. *Appl. Sci.* **2022**, *12*, 2963. [CrossRef]
40. Matsuura, K.; Akatsu, K. A Motor Control Method by Using Machine Learning. In Proceedings of the 2020 23rd International Conference on Electrical Machines and Systems (ICEMS), Hamamatsu, Japan, 24–27 November 2020; pp. 652–655.
41. Beniaguev, D.; Segev, I.; London, M. Single Cortical Neurons as Deep Artificial Neural Networks. *Neuron* **2021**, *109*, 2727–2739. [CrossRef]
42. Pang, Z.; Niu, F.; O'Neill, Z. Solar Radiation Prediction Using Recurrent Neural Network and Artificial Neural Network: A Case Study with Comparisons. *Renew. Energy* **2020**, *156*, 279–289. [CrossRef]
43. Grossi, E.; Buscema, M. Introduction to Artificial Neural Networks. *Eur. J. Gastroenterol. Hepatol.* **2007**, *19*, 1046. [CrossRef]
44. Bengio, E.; Bacon, P.-L.; Pineau, J.; Precup, D. Conditional Computation in Neural Networks for Faster Models. *arXiv* **2015**, arXiv:1511.06297.

45. Siegelmann, H.T.; Sontag, E.D. On the Computational Power of Neural Nets. In Proceedings of the 5th Annual Workshop on Computational Learning Theory, Pittsburgh, PE, USA, 27–29 July 1992; Association for Computing Machinery: New York, NY, USA, 1 July, 1992; pp. 440–449.
46. Pham, D.T.; Karaboga, D. Training Elman and Jordan Networks for System Identification Using Genetic Algorithms. *Artif. Intell. Eng.* **1999**, *13*, 107–117. [CrossRef]
47. Xing, J.; Qin, Z.; Lin, C.; Jiang, X. Research on Startup Process for Sensorless Control of PMSMs Based on I-F Method Combined With an Adaptive Compensator. *IEEE Access* **2020**, *8*, 70812–70821. [CrossRef]
48. Song, Z.; Yao, W.; Lee, K.; Li, W. An Efficient and Robust I-f Control of Sensorless IPMSM With Large Startup Torque Based on Current Vector Angle Controller. *IEEE Trans. Power Electron.* **2022**, *37*, 15308–15321. [CrossRef]
49. Tang, Q.; Chen, D.; He, X. Integration of Improved Flux Linkage Observer and I–f Starting Method for Wide-Speed-Range Sensorless SPMSM Drives. *IEEE Trans. Power Electron.* **2020**, *35*, 8374–8383. [CrossRef]
50. Xu, Y.; Lin, C.; Xing, J.; Zeng, Q.; Sun, J. I-f Starting Rapid and Smooth Transition Method of Full-Speed Sensorless Control for Low Current Harmonic Ultra-High-Speed PMSM. In Proceedings of the 2022 IEEE Applied Power Electronics Conference and Exposition (APEC), Houston, TX, USA, 20–24 March 2022; pp. 1820–1826.
51. El Naqa, I.; Murphy, M.J. What Is Machine Learning? In *Machine Learning in Radiation Oncology: Theory and Applications*; El Naqa, I., Li, R., Murphy, M.J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 3–11. ISBN 978-3-319-18305-3.
52. Putra, D.S.; Azmi, M.; Muslikhin; Purwanto, W. ANN Activation Function Comparative Study for Sinusoidal Data. *J. Phys. Conf. Ser.* **2022**, *2406*, 012029. [CrossRef]
53. Motor Control Blockset Documentation. Available online: https://www.mathworks.com/help/mcb/ (accessed on 27 February 2022).
54. Per-Unit System—MATLAB & Simulink. Available online: https://www.mathworks.com/help/mcb/gs/per-unit-system.html (accessed on 17 April 2022).

**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.