

Research Article

Microcontroller-Based Direct Torque Control Servodrive

Rafael Rodríguez-Ponce  and Francisco Gustavo Mota-Muñoz

Departamento de Ingeniería Robótica, Universidad Politécnica de Guanajuato, Av. Universidad Sur 1001, Cortazar, Guanajuato 38496, Mexico

Correspondence should be addressed to Rafael Rodríguez-Ponce; rrodriguez@upgto.edu.mx

Received 31 October 2019; Accepted 11 January 2020; Published 8 February 2020

Academic Editor: L. Fortuna

Copyright © 2020 Rafael Rodríguez-Ponce and Francisco Gustavo Mota-Muñoz. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Robot technology has become an integral part of the automotive industry in several tasks such as material handling, welding, painting, and part assembly. Therefore, the knowledge and skills to control the electric motors in these manipulators are essential for undergraduate electrical engineering students. Currently, the digital signal processor (DSP) is the core chip in industrial motor-control drives; however, the implementation of DSP control algorithms can be quite challenging for an experienced programmer, even more so for the novice. Considerable research has been done on this topic, although authors usually focus on DSP-based motor drives using popular control techniques such as field-oriented control (FOC). Although highly efficient, this approach is usually reserved for postgraduate education due to its complex structure and functionality. In this paper, the authors present a modular servodrive design on a low-cost, general-purpose microcontroller using the direct torque control (DTC) method, an alternative known for greater simplicity and torque response, compared with FOC. The system design was based on Micropython language allowing the software structure to be more manageable and the code to be more understandable. This design will be useful to undergraduates and researchers with interests in motor control design.

1. Introduction

It is well known that robotic machines are becoming an extensive and vital part of our everyday life. Especially in manufacturing industries such as the automotive, robotic manipulators are used for a wide variety of applications such as material handling, arc and spot welding, painting, and part assembly, among others. Almost any task, which may be repetitive, difficult, or hazardous for a human being, will usually involve some kind of robotic machinery [1]. For this reason, it is paramount for electrical engineering undergraduate students to learn the fundamentals of industrial servomotor control.

To achieve fast and accurate movements in a robot, servomotors are controlled employing high-performance techniques, such as field-oriented control [2, 3]. This method allows control of the motor's speed while maintaining constant torque. Although FOC is commonly used in industrial motor controls due to high efficiency, its physical implementation can be quite challenging since complicated

mathematics is involved. Also, it requires a fast and robust processor and great design efforts [3]. For these reasons, FOC-based servodrive design is a topic usually reserved for graduate or postgraduate engineering education.

Direct torque control (DTC) is another high-performance motor control method which may not be as popular or widely spread as FOC; however, it has many merits such as a more simplistic structure, low computational complexity, quick-response time, and a good dynamic performance [4]. This allows the implementation to require a less powerful processor [5], and a less experienced programmer for the software design. DTC has minor disadvantages compared with FOC, such as difficulty to control the motor at low speeds, variable switching frequency, and high-frequency ripple present on flux and torque signals [3]. Several techniques have been proposed to overcome these issues in classical DTC drives; however, the torque dynamics and the simplicity of the algorithm structure are mostly lost [2, 3].

In this paper, the authors present a prototype servodrive controller based on the classical DTC technique. The

electronic device used is the Espressif ESP32, which is a 32 bit, low-cost embedded processor. For the complete coding of DTC, Micropython programming language was used. Experimental results were carried out to validate the effectiveness of the design. This allowed for the implementation of an open-source, user-friendly controller for permanent magnet synchronous motors (PMSMs). Although the aforementioned disadvantages of the DTC technique are present, the aim of this project is the design of a modular DTC test-bed for educational purposes. This will allow a fundamental understanding of DTC, as well as a simplified implementation and experimental verification of future improvements.

2. Motor Control Strategies

Industrial AC motors are used in a wide variety of applications that run at a constant speed regardless of the load, for example, fans or pumps. But there are also instances where it may be necessary to control the speed or torque, and in this case, a motor drive is required. Currently, there are several techniques used by manufacturers for motion control, two of the most dominating are scalar controls and vector controls [6].

Scalar control is based on mathematical expressions that describe the machine in the steady-state. In this type of control, there is no motor feedback; therefore, torque control is inefficient [7]. Speed control is carried out by varying the voltage and frequency ratio (V/Hz) so that this quotient is always constant. Although these motor drives are simple and economical, speed regulation is only around 3 percent of the motor's base frequency, and more than that, torque is greatly diminished [8]. For this reason, this method is not adequate for high-precision applications, such as industrial robot manipulators or metal-machining centers.

Alternatively, vector control techniques such as FOC are more powerful and efficient, although the control algorithm is significantly more complicated than V/Hz [6]. This methodology, developed by Blaschke [9] in the early 1970s for induction machines, is a closed-loop system; motor output signals such as voltage, current, and shaft position are constantly monitored to estimate the current and magnetic flux spatial vectors. The basic idea is to reorient these vectors perpendicular to each other in order to achieve maximum torque per amp, regardless of the motor's speed [3].

Unlike scalar control, FOC is based on mathematical expressions that describe the steady and transient state model of the machine. This technique not only manipulates the magnitude and angular velocity but also controls the instantaneous position of the spatial vectors of voltage, current, and magnetic flux, allowing a more stable and precise speed and torque control [2].

Although FOC is a high-performance and well-established technology widely applied in power electronics, it is well known that certain features make it difficult to implement in a motor drive. For instance, it has a cascaded architecture that involves coordinate-transformation blocks, proportional-integral (PI) controllers which contain many parameters to be tuned, current regulators, and pulse width modulation (PWM) signal

generators, all of which increases implementation complexity and execution time [10]. Therefore, FOC usually requires a specialized electronic device optimized for processing digital signals in real time. Such a device is usually a DSP since its architecture is enhanced for this kind of task. In other cases, a field-programmable gate array (FPGA) may also be a good alternative, considering that it is possible to design in it, customized parallel operations to lower execution times [11–13]. The main requirement for using these devices is certain expertise in programming languages, such as C++ for the DSP; for the FPGA, it usually will be either VHDL (Very-High-Speed-IC Hardware Descriptive Language) or Verilog [14, 15].

3. Classical DTC Principle

DTC is a motor control strategy that works differently than FOC. It was proposed in the mid-1980s by I. Takahashi for controlling induction motors [16]. The main idea in DTC is to estimate the stator flux and electromagnetic-torque spatial vectors, and control their magnitude separately and independently, with the use of hysteresis controllers [17, 18]. The outputs of the hysteresis controllers are applied directly to a predefined table to select the optimum voltage vector for the voltage-source inverter (VSI). With it, torque and flux vectors are controlled directly and maintained within two hysteresis bands [19].

In DTC there is no need for PI or current controllers. Also, since DTC is in the stator reference frame, no coordinate transformation is applied. Additionally, due to the direct control feature of torque and flux, a PWM modulator is not necessary [20–22]. For these reasons, execution time is lower, and dynamic torque response is higher, compared with FOC [23]. A block diagram of the classical DTC strategy is shown in Figure 1.

3.1. Torque and Flux Estimation. The DTC algorithm is quite straightforward. The process starts by sensing two lines of the three-phase motor current (i_a , i_b). The values are converted from the stator three-phase reference frame (a , b , c) to a two-phase reference frame (α , β) as

$$\begin{aligned} I_\alpha &= I_a, \\ I_\beta &= \frac{\sqrt{3}}{3} (I_a + 2I_b). \end{aligned} \quad (1)$$

The DC link voltage (V_{dc}) is also measured and converted to a two-phase reference frame; together with the last vector applied to the VSI (S_a , S_b , S_c), the two-phase voltage reference components are obtained as

$$\begin{aligned} V_\alpha &= \frac{V_{cd}}{3} (2S_a - S_b - S_c), \\ V_\beta &= \frac{\sqrt{3}}{3} V_{dc} (S_b - S_c). \end{aligned} \quad (2)$$

To estimate the stator magnetic flux vector (ψ_s), the following expression is applied using stator resistance, voltage, and current (R_s , V_s , I_s) as

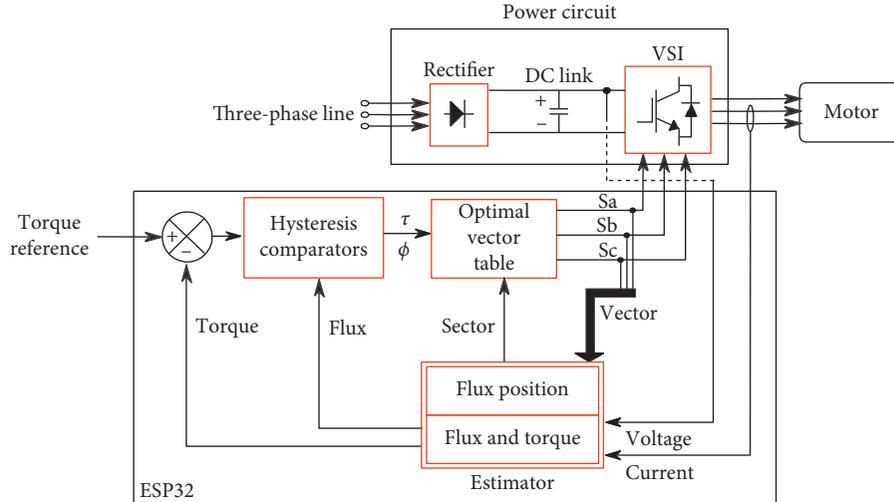


FIGURE 1: Block diagram for the classical DTC strategy.

$$\psi_S = \int (V_S - R_S I_S) dt. \quad (3)$$

However, since the DTC strategy must be implemented digitally, this expression must be formulated as a discrete operation. Therefore, by changing the integration to an accumulated sum and by using the stator voltage and current two-phase reference components, it can be rewritten as follows:

$$\psi_\alpha(n+1) = \psi_\alpha(n) + (V_\alpha(n) - R_S i_\alpha(n)) T_s, \quad (4)$$

$$\psi_\beta(n+1) = \psi_\beta(n) + (V_\beta(n) - R_S i_\beta(n)) T_s, \quad (5)$$

where n is the notation to indicate the current discrete sample and $n+1$ is the next discrete sample. As for T_s , it represents the sampling time, where its value is based on factors such as microcontroller speed and algorithm execution time.

Now, by using the flux components calculated in (4) and (5), the total magnitude of the stator flux vector will be

$$|\psi_s| = \sqrt{\psi_\alpha^2 + \psi_\beta^2}. \quad (6)$$

Also, the angle of the flux vector can be calculated with the following expression:

$$\theta_s = \tan^{-1} \frac{\psi_\beta}{\psi_\alpha}. \quad (7)$$

This angle is of major importance since it will help determine in which sector the flux vector is located.

To estimate the electromagnetic torque, the two-phase current and flux frame reference components and the motor pole pairs (p) are used as

$$T_e(n+1) = \frac{3}{4} p (\psi_\alpha(n+1) i_\beta(n+1) - \psi_\beta(n+1) i_\alpha(n+1)). \quad (8)$$

Once the torque and flux have been estimated, they are compared with their respective reference values to obtain the torque error (e_T) and the flux error (e_ψ). These error values will be the input to the hysteresis comparators.

3.2. Hysteresis Comparators. In DTC, hysteresis comparators are used as a fast and simple equivalent to the PI controllers used in FOC. Their function is to restrict the magnitude of the torque and flux errors to values between two limits. Narrow hysteresis bands lead to smooth sinusoidal current and torque waveforms, although they will also cause high switching frequencies and greater switching losses [10]. Therefore, the hysteresis comparators must have the appropriate limits for a good DTC performance.

The flux-controller output (ϕ) has two possible values, 0 or 1, depending if the flux error (e_ψ) is under the lower limit or over the upper limit, respectively. On the contrary, the torque-controller output (τ) has three possible values, -1, 1, or 0, depending on if the torque error (e_T) is either under the lower limit, over the upper limit, or between both, respectively. Both hysteresis controllers are shown in Figure 2.

The output values of both controllers, together with the spatial placement of the flux vector, are used to select the optimal voltage vector for the VSI from the vector table.

3.3. Optimal Vector Selection. The flux vector rotation space is divided into six equally spaced sectors (1–6) containing six active vectors (V_1 to V_6) and two null vectors (V_0 and V_7). Figure 3 shows how the voltage vectors are distributed in each sector.

To select the optimum vector to apply to the VSI, the hysteresis outputs (ϕ , τ) are used. The flux vector angle θ_s is also required but only to determine in which sector the flux vector is located.

Both torque and flux can be controlled directly by selecting the appropriate active voltage vector and regulate their magnitudes within their respective hysteresis limits. When a null vector is applied to the VSI, it serves as a soft transition from one switching state to the next. Also, they prevent a short circuit in the DC bus due to a delay in the turn-off time of the power switches [24, 25].

To understand the effect of the voltage vector on the flux or torque vector, an example is presented. In Figure 3, the flux vector ψ_s is located in sector 1 and is currently rotating

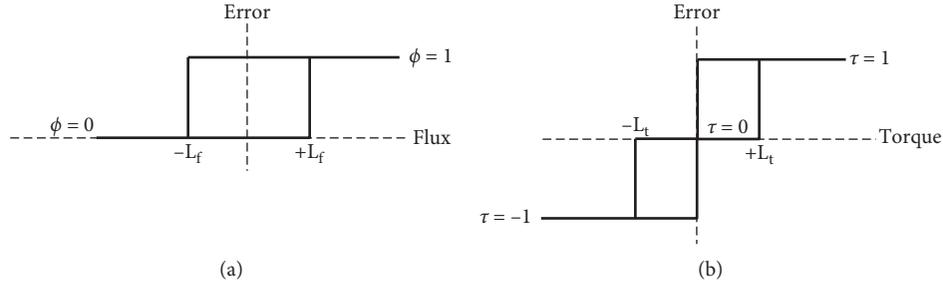


FIGURE 2: Hysteresis controllers for (a) magnetic flux and (b) electromagnetic torque.

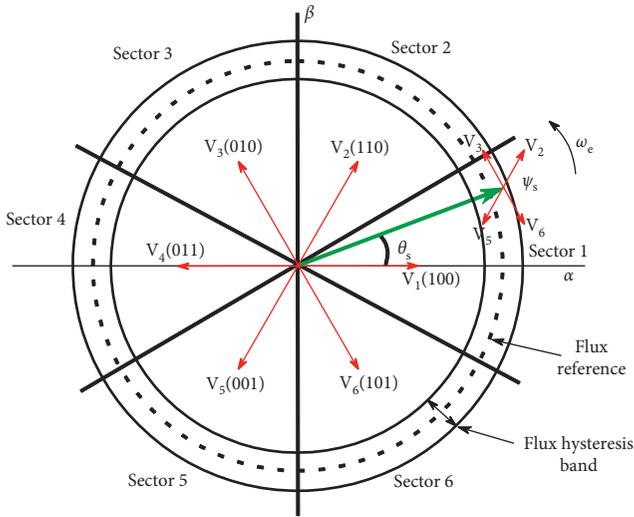


FIGURE 3: Active voltage vectors and sectors of the flux vector space.

counterclockwise. From the optimal vector table (Table 1), the vectors that would have a greater impact on both torque and flux are V_2 , V_3 , V_5 , and V_6 . Applying V_2 would increment both the torque and the flux, whereas V_5 would decrease both. Instead, if V_3 is applied, it would increase the torque but would decrease the flux. Similarly, if V_6 is applied, it would increase the flux and decrease the torque. Applying V_1 or V_4 would have an impact mostly on the flux vector, either decreasing or increasing it, respectively. Lastly, if either null vector is applied, it would slowly decrease both vectors. Therefore, for this example, if sector = 1, $\phi = 1$, and $\tau = 1$, it can be seen in Table 1 that the vector to be applied would be V_2 .

Once the optimal vector is selected, it is applied directly to the VSI for torque and flux correction.

4. DTC Servodrive Description

The DTC servodrive was designed as a customized and modular test-bed for industrial servocontrol. The main idea is that each one of the modules may be replaced by another with different types of hardware. For example, the current sensor module in this design uses inductive sensors but can be replaced easily by a Hall-effect sensor module or a shunt resistor module. Also, the processing module which currently uses a 32 bit microcontroller may be replaced in the

TABLE 1: List of all optimal vectors for the VSI based on the sector number (1–6), the torque error (τ), and the flux error (ϕ).

ϕ	τ	Sector					
		1	2	3	4	5	6
$\phi = 1$	$\tau = 1$	V_2	V_3	V_4	V_5	V_6	V_1
	$\tau = 0$	V_7	V_0	V_7	V_0	V_7	V_0
	$\tau = -1$	V_6	V_1	V_2	V_3	V_4	V_5
$\phi = 0$	$\tau = 1$	V_3	V_4	V_5	V_6	V_1	V_2
	$\tau = 0$	V_0	V_7	V_0	V_7	V_0	V_7
	$\tau = -1$	V_5	V_6	V_1	V_2	V_3	V_4

future by a different type of embedded system. Even the DTC algorithm has an open-source architecture so that the main blocks can be modified or enhanced, or even new functionalities may be easily added. The primary reason is to be able to experimentally test and visualize the effects these changes have on the motor performance. The servodrive system is shown in Figure 4.

A list of the electronic components used in this prototype is presented in Table 2.

4.1. Current-Voltage Sensing and Signal Conditioning Circuitry. One of the most important stages of the DTC process starts in current sensing. Since the flux and torque estimations are based on the current values, it is necessary to have a precise current sensor. Firstly, the current signals of two of the motor lines are measured and converted to a DC signal by adding an offset voltage. Secondly, the signal is amplified and filtered so it can be converted to a digital signal using the analog to digital converters (ADC). Antialiasing filters are employed to effectively cut off frequencies higher than about half the maximum frequency of interest.

For this project, a MURATA 56300C inductive current sensor was used. As shown in Figure 5, the current is sensed, positive offset, and amplified using operational amplifiers.

To calculate the two-phase reference components for the voltage, the DC power bus must be monitored in case of voltage drops. For the DC voltage sensing (Figure 6), a resistive divider circuit was used and also amplified using JFET-Input operational amplifiers. Since this voltage is always positive, no offset is required. The signal is scaled, amplified, and filtered so it can be converted to a digital signal.

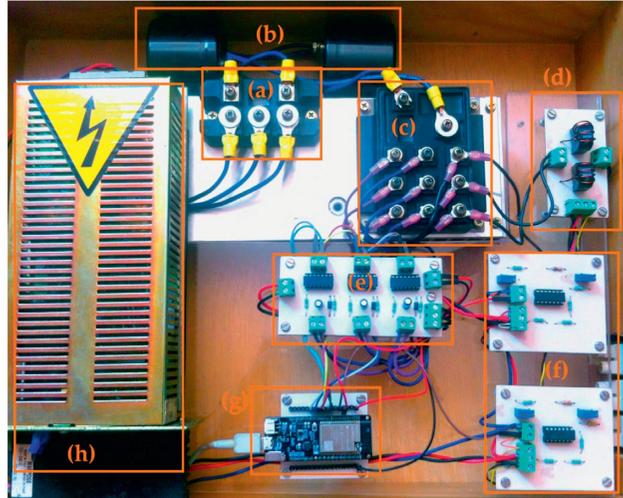


FIGURE 4: DTC servodrive prototype including (a) the three-phase rectifier bridge, (b) capacitor filters, (c) the IGBT power module, (d) the inductive current sensors, (e) the IGBT gate drivers, (f) the signal conditioning modules, (g) the ESP32 microcontroller, and (h) the DC power supply for low-power integrated circuits.

TABLE 2: Main electronic components used in the DTC servodrive.

Components	Description
A50I-2001-0232	Diode rectifier bridge, FUJI ELECTRIC 800 V, 30 A
A50I-0001-0230	IGBT power module, FUJI ELECTRIC, 600 V, 50 A
HCGF4A	HITACHI electrolytic capacitors, 1000 μ F, 400 V _{dc} , surge 450 V _{dc}
56300C	MURATA current sensor, 300 turns, 10 A, 180–335 mH @1V 10 KHz
IR2110	IGBT driver, INTERNATIONAL RECTIFIER, 500 V, 2 A
TL084	JFET-input OPAMPS TEXAS INSTRUMENTS
LTC1564	LINEAR antialias filter IC
ESP32	ESPRESSIF 32 bit microcontroller

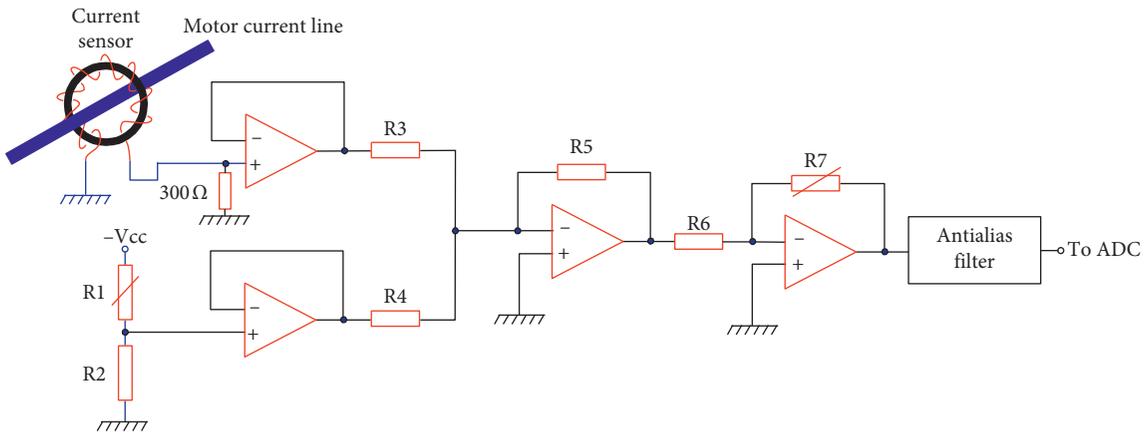


FIGURE 5: Circuitry for current sensing and signal conditioning.

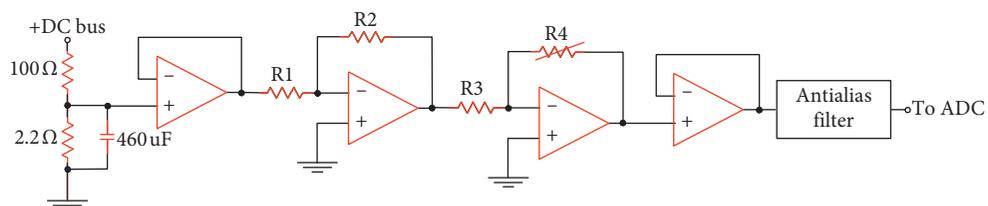


FIGURE 6: Circuitry for V_{dc} sensing and signal conditioning.

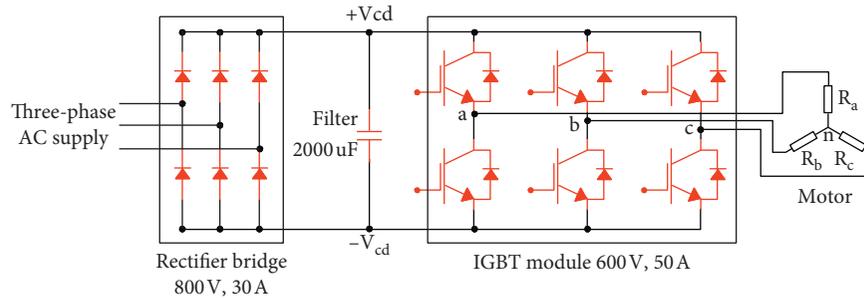


FIGURE 7: Power circuitry for the servodrive system.

4.2. Power Electronics Circuitry. In this stage, the three-phase voltage is converted to a DC signal so that it can be converted back again to a controlled AC signal for the motor. For this stage, industrial-grade power modules were used for the rectifier bridge and the VSI (Figure 7). Two 1000 μF electrolytic capacitors are connected in parallel for the DC filtering stage.

4.3. DTC Processing Module. The processor used for the implementation of the DTC technique is the ESPRESSIF ESP32-WROVER-B 32 bit microcontroller, which runs at 240 MHz. ESP32 is a great development board for IoT applications as it is a low-cost device and has many powerful features (Table 3). ESP32 can be programmed in various environments and programming languages such as Arduino IDE, Espressif-IDF, Lua, Micropython, and C/C++.

In this project, Micropython was used for the design and implementation of the DTC technique. Micropython is a reimplement of Python 3 programming language, targeted to microcontrollers and embedded systems. Both programming languages are very similar. Apart from a few exceptions, all the language features of Python are available in Micropython.

The main reason for using Micropython in the DTC prototype is that today Python is one of the most widely used, simple and easy to learn programming languages currently available, and well suited for computational analysis and design [26, 27]. With the emergence of Micropython, it becomes easy to program microcontrollers and embedded devices. This makes ESP32 and Micropython, a great experimental platform for students, teachers, and researchers [28].

The DTC scheme presented previously in Figure 1 was implemented in Micropython and the ESP32 with the use of peripherals such as ADCs, DACs (digital to analog converters), and timers. A summarized workflow of the DTC algorithm is shown in Figure 8.

The DTC algorithm is programmed in a Micropython interrupt service routine (Figure 9) which is triggered every 250 μs by the timer. Once initiated, the current and voltage analog signals are converted to digital values. These values are used to estimate flux components, flux, and torque values and then compared with reference values to select the optimal vector for the VSI. Any of these signals can be visualized in an external oscilloscope utilizing the internal DACs.

5. Experimental Results

The experimental testing was carried out using a standard 1 kW PMSM servo, and the main parameters of which are given as follows: $R_s = 15.57 \Omega$; pole pairs, $p = 8$; rated voltage, $V = 130 \text{ V}$; rated current, $I = 2.5 \text{ A}$; rated torque, $T = 2.7 \text{ Nm}$, and rated speed, $\omega_e = 5000 \text{ RPM}$.

To validate the DTC servodrive performance, several tests were executed with the main focus being on the flux, torque, and current signals. If the output signals are similar to the simulated signals, then it means that the algorithm has been properly implemented. Also, the torque output was measured to verify that it is as close as possible to the torque reference value.

Although it is important to clarify that the performance of the DTC algorithm and the quality of the output signals are directly related to the hysteresis controllers [29, 30]. For this reason, most of the tests were made by observing the effect of the hysteresis bands on the output signals.

5.1. Effects of the Hysteresis Flux Band. The hysteresis flux band has a considerable effect on the output signals. For instance, if the band is narrow, the switching frequency will be higher, the flux locus will tend to a circular path, and the current signal is very close to a sine wave; however, current ripple will increase since a narrow band will make it easier for the hysteresis limits to be exceeded. On the contrary, a wide band will lower the ripple content, but the switching frequency will decrease, the flux locus will tend to a hexagonal path, and the current signal is distorted [30]. This effect can be observed in simulations (Figure 10) and experimental tests (Figure 11).

The effect of the flux hysteresis band can also be observed in the current signals. When the band is narrow, the current signal will have greater ripple content, but the shape will be close to a sine wave. If instead, the band is wider, the ripple will be lower, but the signal will be heavily distorted. This behavior can be seen in simulations (Figure 12) and the experimental motor current signal (Figure 13).

5.2. Effects of the Hysteresis Torque Band. The width of the hysteresis torque band also has a considerable effect on DTC performance, mainly on the inverter switching frequency. Although it also influences the total harmonic distortion

TABLE 3: ESP32-WROVER-B specifications and peripherals.

Components	Description
CPU	Xtensa dual-core 32 bit LX6 240 MHz processor. Includes ultralow power coprocessor.
Embedded memory	448 KB ROM, 4 MB SPI flash, 8 MB PSRAM.
Peripherals	SD card, LED PWM, motor PWM, IR, pulse counter, GPIO, capacitive touch sensor, 2 x ADC, 2 x DAC, and Hall-effect sensor.
Communication	Wi-fi (2.4 GHz), Bluetooth 4.2, CAN bus 2.0, SPI, UART, I ² C, and I ² S.

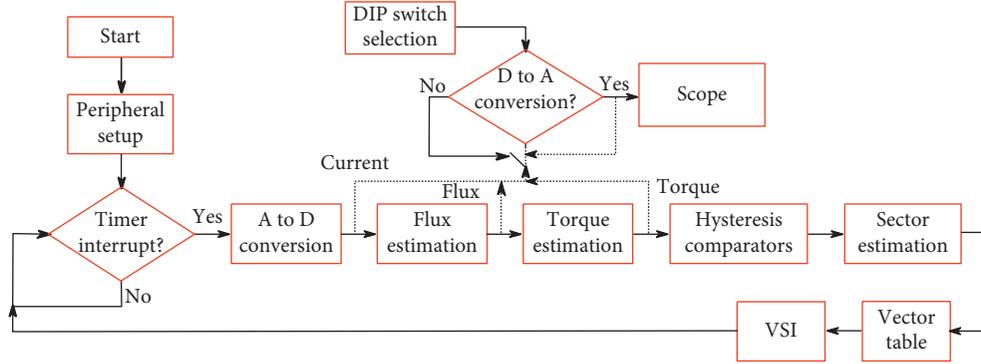


FIGURE 8: DTC algorithm workflow.

```

def isr_DTC(pin):
    global Ia,Ib,Te,flux_sqrd,Rs,Ts,P
    global f_band,t_band,Sa,Sb,Sc
    # current a2d conversion
    Ia = adc1.read()
    Ib = adc2.read()
    # magnitud adjustment and offset
    Ia = (Ia * fac_a)-offset
    Ib = (Ib * fac_b)-offset
    # current and voltage biphas components
    Ialfa = Ia
    Ibeta = 0.577*(Ia + 2*Ib)
    Valfa = 33.3*(2*Sa - Sb - Sc)
    Vbeta = 57.7*(Sb - Sc)
    #flux components
    palfa = flux(palfa_0,Valfa,Rs,Ialfa,Ts)
    pbeta = flux(pbeta_0,Vbeta,Rs,Ibeta,Ts)
    palfa_0 = palfa
    pbeta_0 = pbeta
    #flux and torque estimation
    flux_sqrd = (palfa**2) + (pbeta**2)
    Te = 0.75*P*(palfa*Ibeta - pbeta*Ialfa)
    #error calculation
    error_flux = (flux_ref**2) - flux_sqrd
    error_t = Te_ref.read() - Te
    #hysteresis Bands
    phi = hist_flux(error_flux,f_band)
    tau = hist_torque(error_t,t_band)
    #flux sector
    sector = calc_sector(palfa,pbeta)
    # optimal vector table
    vect = vector_table(phi,tau,sector)
    #vector to VSI
    Sa = vect[0]
    Sb = vect[1]
    Sc = vect[2]
    to_VSI(Sa,Sb,Sc)

```

FIGURE 9: Micropython code example for the DTC interrupt service routine.

(THD) in the current signals, this effect is due to both hysteresis bands.

Firstly, the switching frequency is defined as

$$F_{sw} = \frac{N_s}{T_f} \quad (9)$$

where N_s is the number of commutations per period and T_f is the period of the fundamental signal.

To observe the effect on the switching frequency, both hysteresis bands were varied. In Figure 14, it can be seen that even though both hysteresis bands affect the switching frequency, the torque band has a greater influence.

As for the THD on the current signal, it is defined as

$$THD = \sqrt{\frac{(I^2 - I_1^2)}{I_1^2}}, \quad (10)$$

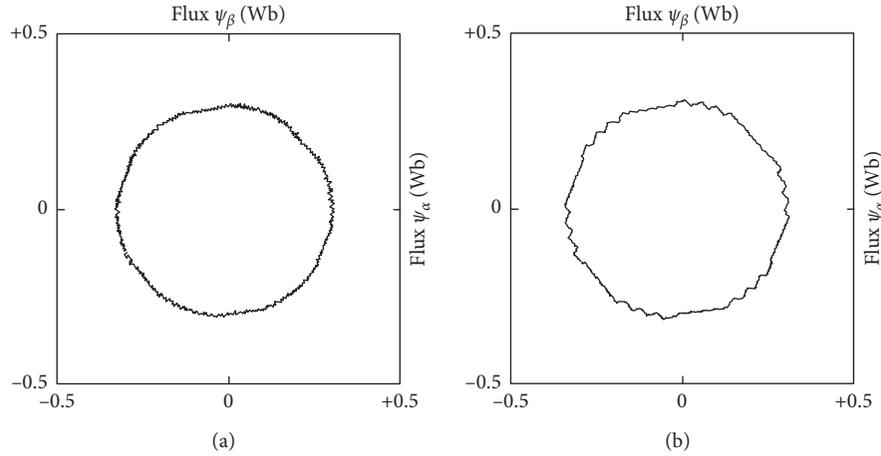


FIGURE 10: Flux locus simulation to observe the effect of the flux hysteresis band with (a) 2% and (b) 5% of the flux reference value.

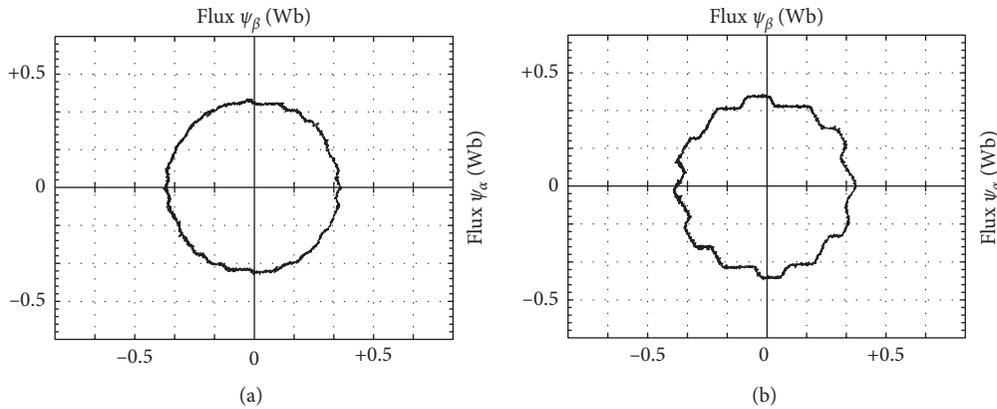


FIGURE 11: Flux locus on the experimental output to observe the effect of the flux hysteresis band with (a) 2% and (b) 5% of the flux reference value.

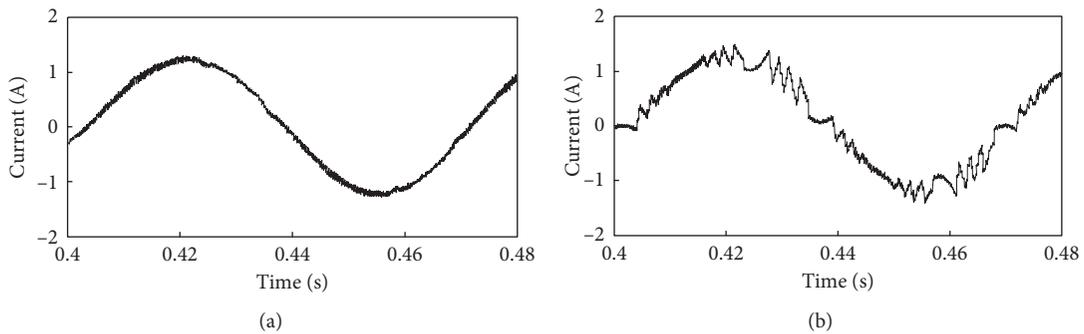


FIGURE 12: Current output simulation to observe the effect of the flux hysteresis band with (a) 2% and (b) 5% of the flux reference value.

where I is the RMS value of the current and I_1 is the RMS value of the fundamental harmonic component.

Both hysteresis bands were varied, but in this case, the flux band had a greater influence on the current signal distortion as shown in Figure 15.

5.3. Output Torque Measurements. To validate the torque performance of the servocontroller, measurements were

made using a 10 Nm rotary torque sensor. In these tests, it was important to verify that the torque output corresponds to the torque reference and that the ripple content is relatively low.

Once again, the hysteresis bands were varied to observe the effect on the torque signal. Since the original DTC scheme with no enhancements was implemented, it was expected to have relatively high ripple content. The best torque performance with the lowest ripple was obtained

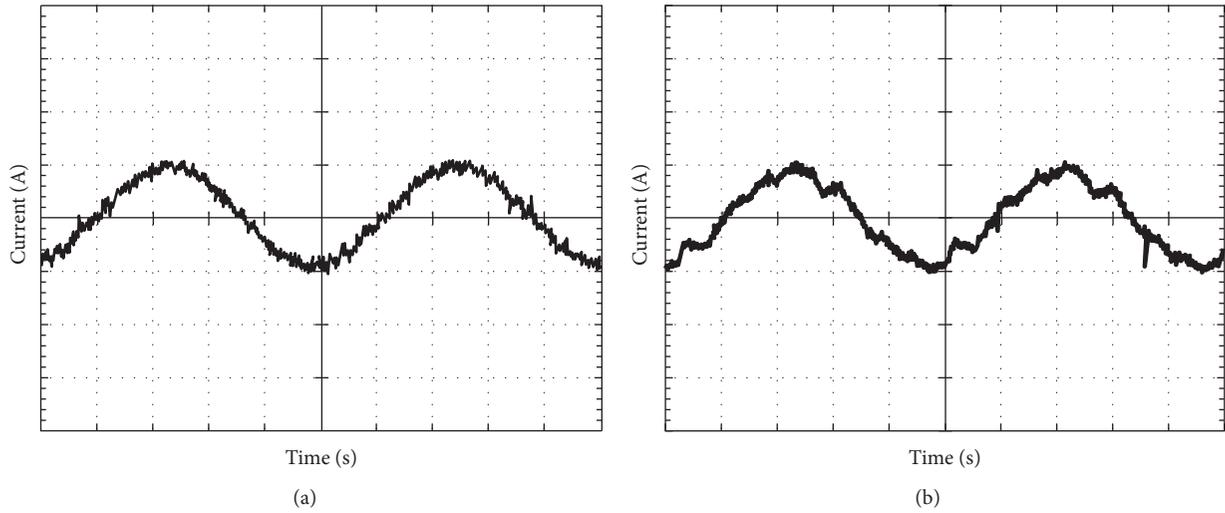


FIGURE 13: Experimental motor current output to observe the effect of the flux hysteresis band with (a) 2% and (b) 5% of the flux reference value. The current amplitude is 2 A/div, and the period is 5 mS/div.

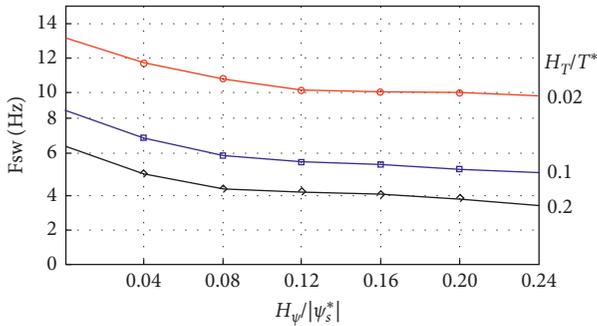


FIGURE 14: Average inverter switching frequency as function of the flux (H_ψ) and torque (H_T) hysteresis bands.

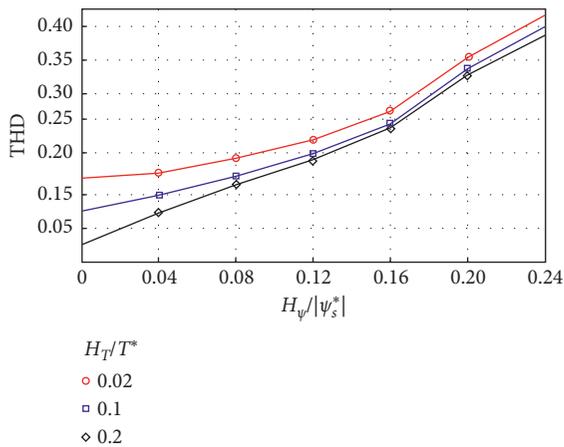


FIGURE 15: Total harmonic distortion as a function of the flux (H_ψ) and torque (H_T) hysteresis bands.

when the flux band was set to 2% of the reference value (0.8 Wb), and the torque band was set to 1% of the reference value. The output torque signal is shown in Figure 16.

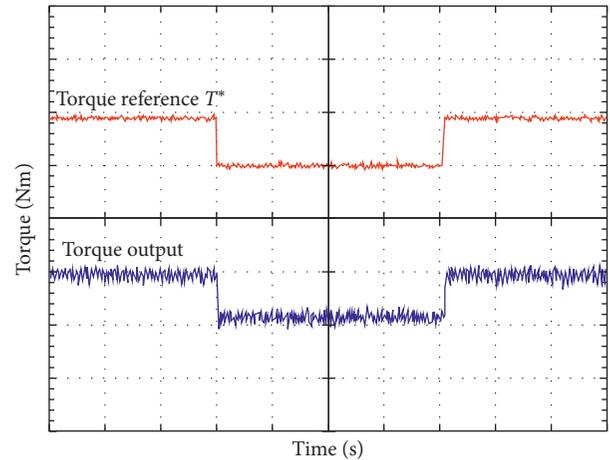


FIGURE 16: Experimental torque response with a torque reference $T^* = \pm 1$ Nm, torque band $H_T = 1\%$, and flux band $H_\psi = 2\%$. The torque amplitude is 2 Nm/div, and the period is 1 S/div.

Needless to say, the verification of the torque response in the servodrive is very important since this ensures that changes in torque and direction will be carried out by the drive. However, the use of a physical torque sensor is usually expensive and requires additional installation and software. Additionally, torque sensors have a specific range; if a wider range is required, the cost becomes prohibitive. In future projects, it would be interesting to make use of virtual instruments [31] for the monitoring of torque and magnetic flux signals, and in this way avoid the limitations of physical sensors, especially in electromechanical projects aimed to education [32] where funding is usually limited.

6. Conclusion

The proposal of a PMSM drive based on the DTC technique was presented. This motor control algorithm was coded in Micropython for implementation on the ESP32

microcontroller. The aim of this prototype is to be used in didactic applications for undergraduate engineering students. Experimental and simulation results validate the effectiveness of the design and show that good torque performance can be achieved with DTC on a low-cost, general-purpose processor. The main benefit of this prototype is its simplicity since it did not require any modification to the original DTC structure, and even so current and torque ripple content was acceptable. The code written on Micropython made it easy to understand and to adjust for different testing parameters such as modifying hysteresis bands to observe the quality of current, flux, and torque output signals.

This will allow students and researchers interested in motion control to experiment with a simple and practical tool and in future projects to improve even the performance and structure of this prototype.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] X. Sun, Y. Zhang, and J. Chen, "RTPO: a domain knowledge base for robot task planning," *Electronics*, vol. 8, no. 10, p. 1105, 2019.
- [2] F. Wang, Z. Zhang, X. Mei, J. Rodríguez, and R. Kennel, "Advanced control strategies of induction machine: field oriented control, direct torque control and model predictive control," *Energies*, vol. 11, no. 1, p. 120, 2018.
- [3] J. Zhu, X. Wang, B. Xu, H. Feng, and X. Xu, "A comparative study on field oriented control and direct torque control for permanent magnet linear synchronous motor," *Journal of Computers*, vol. 8, pp. 256–263, 2013.
- [4] Q. Song, Y. Li, and C. Jia, "A novel direct torque control method based on asymmetric boundary layer sliding mode control for PMSM," *Energies*, vol. 11, no. 3, p. 657, 2018.
- [5] B. Metidji, N. Taib, L. Baghli, T. Rekioua, and S. Bacha, "Low-cost direct torque control algorithm for induction motor without AC phase current sensors," *IEEE Transactions on Power Electronics*, vol. 27, no. 9, pp. 4132–4139, 2012.
- [6] B. K. Bose, *Modern Power Electronics and AC Drives*, Prentice-Hall, Upper Saddle River, NJ, USA, 2002.
- [7] A. Munoz-Garcia, T. A. Lipo, and D. W. Novotny, "A new induction motor V/f control method capable of high-performance regulation at low speeds," *IEEE Transactions on Industry Applications*, vol. 34, no. 4, pp. 813–821, 1998.
- [8] M. S. Aspolli, S. Kalshetti, and P. V. Hunagund, "Speed control of 3 Φ induction motor using volts hertz control method," *The International Journal of Electrical Engineering*, vol. 3, pp. 231–236, 2011.
- [9] F. Blaschke, "The principle of field-orientation as applied to the new transvector closed-loop system for rotating-field machines," *Siemens Review*, vol. 34, no. 3, pp. 217–220, 1972.
- [10] D. Casadei, F. Profumo, G. Serra, and A. Tani, "FOC and DTC: two viable schemes for induction motors torque control," *IEEE Transactions on Power Electronics*, vol. 17, no. 5, pp. 779–787, 2002.
- [11] M. Hamouda, H. F. Blanchette, K. Al-Haddad, and F. Fnaiech, "An efficient DSP-FPGA-based real-time implementation method of SVM algorithms for an indirect matrix converter," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 11, pp. 5024–5031, 2011.
- [12] C. Koulamas and M. Lazarescu, "Real-time embedded systems: present and future," *Electronics*, vol. 7, no. 9, p. 205, 2018.
- [13] M. Yushkova, A. Sanchez, A. de Castro, and M. A. Martínez-García, "A comparison of filtering approaches using low-speed DACs for hardware-in-the-loop implemented in FPGAs," *Electronics*, vol. 8, no. 10, p. 1116, 2019.
- [14] E. Monmasson and M. N. Cirstea, "rFPGA Design Methodology for Industrial Control Systems-A Review," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1824–1842, 2007.
- [15] C.-K. Lai, H.-Y. Lai, Y.-X. Tang, and E.-S. Chang, "Field programmable gate array-based linear shaft motor drive system design in terms of the trapezoidal velocity profile consideration," *Machines*, vol. 7, no. 3, p. 59, 2019.
- [16] I. Takahashi and T. Noguchi, "A new quick-response and high-efficiency control strategy of an induction motor," *IEEE Transactions on Industry Applications*, vol. IA-22, no. 5, pp. 820–827, 1986.
- [17] H. Hamidreza, A. Rassolkin, T. Vaimann et al., "A novel vector control strategy for a six-phase induction motor with low torque ripples and harmonic currents," *Energies*, vol. 12, no. 6, p. 1102, 2019.
- [18] G. Zhang, C. Chen, X. Gu, Z. Wang, and X. Li, "An improved model predictive torque control for a two-level inverter fed interior permanent magnet synchronous motor," *Electronics*, vol. 8, no. 7, p. 769, 2019.
- [19] I. M. Alsofyani and K.-B. Lee, "Evaluation of direct torque control with a constant-frequency torque regulator under various discrete interleaving carriers," *Electronics*, vol. 8, no. 7, p. 820, 2019.
- [20] N. Pimkumwong and M.-S. Wang, "annsOnline 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.
- [21] G. M. Sung, W. Y. Wang, W. S. Lin, and C. P. Yu, "Predictive direct torque control application-specific integrated circuit of an induction motor drive with a fuzzy controller," *Journal of Low Power Electronics and Applications*, vol. 7, no. 2, p. 15, 2017.
- [22] J. Park and X. Wang, "Sensorless direct torque control of surface-mounted permanent magnet synchronous motors with nonlinear kalman filtering," *Energies*, vol. 11, no. 4, p. 969, 2018.
- [23] J. Rodriguez, R. M. Kennel, J. R. Espinoza, M. Trincado, C. A. Silva, and C. A. Rojas, "High-performance control strategies for electrical drives: an experimental assessment," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 812–820, 2012.
- [24] L. Yaohua, M. A. Jian, L. Jingyu et al., "Study on the use of zero voltage vectors in the PMSM DTC system," in *Proceedings of the 30th Chinese Control Conference*, pp. 3559–3564, Yantai,

- China, July 2011.
- [25] E.-S. Jun, S.-Y. Park, and S. Kwak, "A comprehensive double-vector approach to alleviate common-mode voltage in three-phase voltage-source inverters with a predictive control algorithm," *Electronics*, vol. 8, no. 8, p. 872, 2019.
 - [26] J. Vega and J. M. Cañas, "An innovative python-based educational framework using real and simulated Arduino robots," *Electronics*, vol. 8, no. 8, p. 889, 2019.
 - [27] M. Badenhorst, C. J. Barry, C. J. Swanepoel, C. T. Van Staden, J. Wissing, and J. M. Rohwer, "Workflow for data analysis in experimental and computational systems Biology: using python as "Glue"," *Processes*, vol. 7, no. 7, p. 460, 2019.
 - [28] L. O. Aghenta and M. T. Iqbal, "Low-cost, open source IoT-based SCADA system design using thinger.IO and ESP32 thing," *Electronics*, vol. 8, no. 8, p. 822, 2019.
 - [29] I. M. Alsofyani, N. R. N. Idris, and K.-B. Lee, "Dynamic hysteresis torque band for improving the performance of lookup-table-based DTC of induction machines," *IEEE Transactions on Power Electronics*, vol. 33, no. 9, pp. 7959–7970, 2018.
 - [30] D. Casadei, G. Grandi, G. Serra, and A. Tani, "Effects of flux and torque hysteresis band Amplitude in direct torque control of induction machines," in *Proceedings of the 20th Annual Conference of IEEE Industrial Electronics*, pp. 299–304, Bologna, Italy, September 1994.
 - [31] L. Fortuna, P. Giannone, S. Graziani, and M. G. Xibilia, "Virtual instruments based on stacked neural networks to improve product quality monitoring in a refinery," in *Proceedings of the IEEE Instrumentation and Technology Conference*, pp. 1889–1893, Ottawa, Canada, May 2005.
 - [32] M. Bucolo, A. Buscarino, C. Famoso, L. Fortuna, and M. Frasca, "Control of imperfect dynamical systems," *Nonlinear Dynamics*, vol. 98, no. 4, pp. 2989–2999, 2019.