

Original Article

# FOC-Based Sensorless Control for Torque Ripple Reduction in PMSM Drives Using STM32 Platform

Pandry Narendra Rao<sup>1</sup>, Abhilash Krishna<sup>2</sup>, Jhansi Priya R<sup>3</sup>

<sup>1,2,3</sup>Department of Electrical and Electronics, RV College of Engineering, Karnataka, India

<sup>1</sup>Corresponding Author : [pnrao@rvce.edu.in](mailto:pnrao@rvce.edu.in)

Received: 20 November 2025

Revised: 24 December 2025

Accepted: 22 January 2026

Published: 17 February 2026

**Abstract** - This paper presents the design and implementation of a Field Orientation Control (FOC) based speed controller for a three-phase Permanent Magnet Synchronous Motor (PMSM), aiming to reduce rotor torque ripple effectively. The system is developed using an STM32F401RE microcontroller along with a Simonk 30A Electronic Speed Controller (ESC), ACS712 current sensor, and TCRT IR sensor for effective motor control. The use of the FOC algorithm results in smoother operation and improves efficiency. The implementation is done on the STM32 Motor Control Workbench and validated through hardware testing. The initial torque is observed to be  $0.15 \text{ N} \cdot \text{m}$  at 2500rpm, and the peak torque is  $0.45 \text{ N} \cdot \text{m}$  at 9000rpm, respectively. The torque becomes constant at  $0.05 \text{ N} \cdot \text{m}$  beyond 16000rpm. Whereas for a similar setup using a six-step computation strategy, the torque generated initially at 2500rpm is  $0.6 \text{ N} \cdot \text{m}$ , and the peak torque is  $1.3 \text{ N} \cdot \text{m}$  at 9000rpm. The obtained results are validated according to the IEEE standards (IEEE 181-2011) for safe operation of the motor within the prescribed operating region of torque ripples. The torque ripples using the FOC algorithm are observed to be 6.6% and six-step commutation is 20%. Therefore, comparative analysis with traditional six-step commutation reveals that the proposed FOC strategy significantly reduces torque ripple, improves speed stability, and enhances the dynamic response of the motor. This makes it a suitable solution for high-performance applications such as robotics, EVs, and industrial automation.

**Keywords** - ESC, FOC, PMSM, STM32, Torque Ripple Reduction.

## 1. Introduction

PMSMs are now a key part of modern electric drive systems because they are efficient, small, and offer great torque for their weight. They are used in many areas like electric vehicles, robots, aircraft, and factories, where exact control of speed and torque is important. However, one major issue with these motors is torque ripples, which cause issues such as vibration, noise, and reduced efficiency. Traditional control techniques, such as trapezoidal or six-step commutation, are simple and cost-effective but fail when high accuracy and smooth torque delivery are needed. These methods generate abnormal current transitions and result in significant torque pulsations, especially at low speeds. This makes them unsuitable for precision-driven applications such as CNC machines, medical robotics, and electric propulsion systems. To solve these issues, better control methods like FOC, known as Vector Control (VC), have been developed. FOC transforms the stator's three-phase current system into a rotating reference frame, enabling independent control of the torque and flux-generating components of the stator current. Aligning the stator magnetic field with the rotor's magnetic field allows for optimal torque generation with minimal ripple [1, 2]. Moreover, the integration of Proportional-Integral (PI) controllers in the current and speed control loops ensures fast

dynamic response and improved system stability. When implemented on advanced microcontroller platforms like STM32, FOC offers a cost-effective and scalable solution for practical motor control [3]. This helps create a smooth sine-like current, leading to less torque ripple and better performance than older methods like the six-step or trapezoidal control. This paper describes the model and setup of a sensorless FOC system for controlling PMSM speed using an STM32F401RE microcontroller. The system uses a Simonk 30A ESC, an ACS712 current sensor, a TCRT IR sensor, and a potentiometer to set the desired speed. The FOC algorithm is built using the STM32 Motor Control Workbench and programmed on Arduino IDE platforms. The implementation was tested on real hardware for variable speed conditions. The corresponding torque and its variations were noted through the serial monitor and serial plotter. The system's performance is checked by comparing torque ripples and speed response to the six-step method. These results were validated against the IEEE standards for safe operation of the motor within the prescribed torque ripple percentages [4].

## 2. Methodology

The methodology used in this project involves design, configuration, hardware setup, and practical implementation



of FOC for three-phase PMSM. The main objective is to minimize torque ripple and improve speed regulation using a sensorless closed-loop control strategy implemented on the STM32 microcontroller platform. The block diagram, as shown in Figure 1, depicts the control architecture for implementing FOC on a 3-phase PMSM. The system begins with a reference input, basically from a potentiometer that sets the desired motor speed. This input is processed by a PI controller, which compares it with feedback signals to generate an appropriate control response. The main motor current is decomposed into two orthogonal components: Direct-Axis Current ( $I_d$ ) for controlling Magnetic Flux and Quadrature-Axis Current ( $I_q$ ) for controlling torque. PI

controllers regulate these two components to ensure efficient and smooth motor operation, usually keeping  $I_d$  as zero to obtain maximum torque. The resulting Voltages ( $V_d$  and  $V_q$ ) are transformed into three-phase voltages ( $V_a, V_b, V_c$ ) by inverse Clarke and Park transformation techniques, which are then converted into PWM signals to drive the Three-Phase Inverter (ESC) connected to the motor. Feedback from current sensors and IR sensors is continuously monitored and transformed back into the d-q frame using Clarke and Park transformations. This closed-loop system ensures precise speed and torque regulations, effectively minimizing the torque ripple and enhancing the overall performance of the PMSM [5].

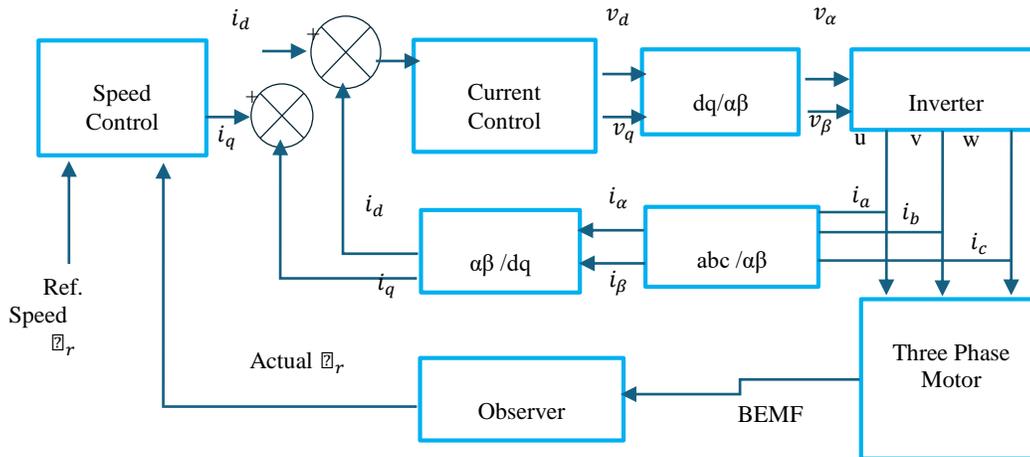


Fig. 1 Block diagram of FOC-based speed control of PMSM

### 3. Workbench Configuration

The configuration process starts by running the MC Workbench and creating a new project. The user picks the main microcontroller board, such as the STM32F401RE Nucleo, and the power electronics, like a compatible inverter

or ESC. These settings make sure the code created later is compatible with the actual hardware, including the microcontroller, ESC, and sensors. A fully built block diagram will be displayed as shown in Figure 2 with all the relevant parameters [6, 7].

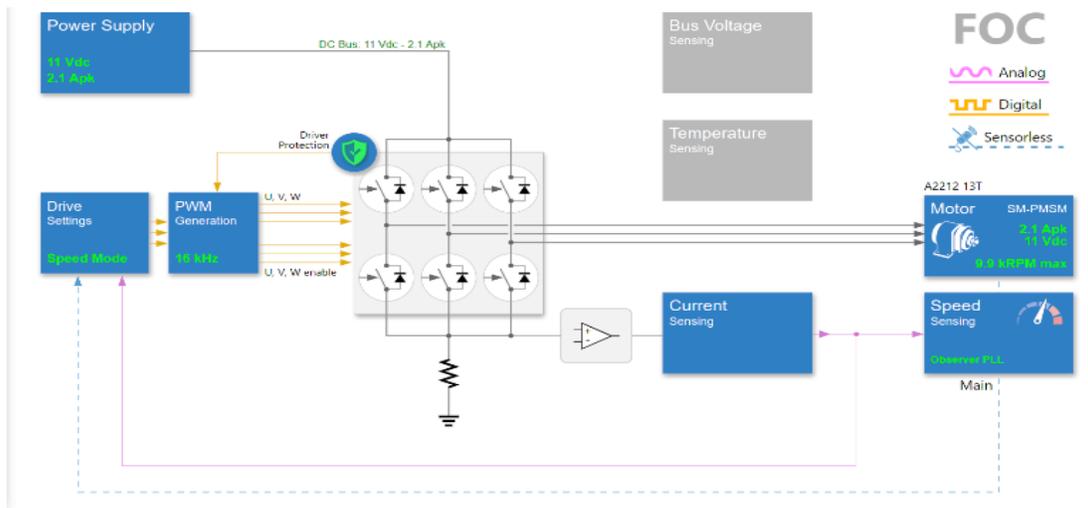


Fig. 2 Workbench configuration

### 3.1. PMSM Motor Configuration

The user selects the motor type and enters specific motor details. For this project, a 3-phase SM - PMSM motor, like the A2212 2200kV, is chosen. The user then enters values such as rated voltage, maximum current, number of pole pairs, and other motor specifications from the motor’s datasheet or through measurements. All the motor-rated values are auto-generated and adapted in the workbench, as shown in Figure 3. This setup helps the control system run accurate FOC routines that match the motor being used.

The locked values can also be theoretically verified and varied if required. These values are manually checked for their

correctness. Calculations for the Back EMF constant are shown below:

$$K_e = \frac{\text{Rated Flux} \times \text{Volts} \times \text{Pole pairs}}{60}$$

$$K_e = (0.005752 \times 2200 \times 7)/60 = 0.822 \text{ Vrms/kRPM}$$

Where:

$$\text{Rated Flux} = 0.005752 \text{ V/Hz}$$

$$\text{Pole Pairs} = 7$$

**Motor parameters**

---

Motor magnetic structure:

Pole Pairs:

**Electrical parameters**

---

Max current:  Apk  Power board maximum rated current: 2.1 Apk

Max DC Voltage:  V  Power board supported voltage range: (7 - 45 Vdc)

Rs:  Ω

Ls:  mH

B-Emf constant:  Vrms/kRPM

Rated Flux: 0.005752 V/Hz

**Fig. 3 PMSM motor configuration**

### 3.2. Power Supply Configuration

Under the “Power Supply” section, the user enters values like 12V input voltage and current limits that are provided by

an external power supply, as shown in Figure 4. They also set up protective options for overcurrent or undervoltage conditions. These settings are important for system safety and ensure the power and control systems work well together.

Provide here below the Max Current and the Bus Voltage levels that will be used by this application.  
 Please note that these values have to be provided by your Power Supply and they have to be compatible with both PowerBoard and Motor

Max. application Current:  A Up to 2.1 A

Bus Voltage:  V from 7 V to 12 V

<b>Power board Info:</b>	<b>Motor Info:</b>
<input checked="" type="checkbox"/> Maximum rated current: 2.12 Apk	<input checked="" type="checkbox"/> Max current: 2.1 Apk
<input checked="" type="checkbox"/> Supported voltage range: (7 - 45) Vdc	<input checked="" type="checkbox"/> Max DC Voltage: 12 Vdc

**Fig. 4 Power supply configuration**

### 3.3. Sensor and Feedback Configuration

The workbench lets the user choose different sensor setups. For this project, a current sensor like the ACS712 is used and connected to specific microcontroller pins. The user can choose from various current-sensing methods, such as three shunts, a single shunt, or inline sensing. They also set the sensor's gain and input mapping to match its characteristics. For feedback, the IR sensor, such as the TCRT5000 module, is selected and connected to a digital input pin. As sensorless FOC is being used, the settings related to the observer, like the startup routine and PLL tuning, are configured in the software as shown in Figure 5.



Fig. 5 Sensor and Feedback Configuration

### 3.4. PWM and PI Tuning Settings

The user then selects and adjusts the control algorithm (FOC mode) and the PWM settings. FOC is turned on for the PMSM motor. The user sets priorities for the current and speed control loops and chooses PWM frequencies that match the hardware and motor performance, such as 16kHz for the ESC, as listed in the specs.

Regulation settings, like the Proportional and Integral (PI) gains for speed and current controllers, are either set manually using control theory or with the help of auto-tuning in the software, as shown in Figure 6. These settings affect how the control system responds and stays stable [8]. The driver settings can be auto-calculated in the workbench, and the corresponding Kp and Ki values are generated for the controller as shown in Figure 7.

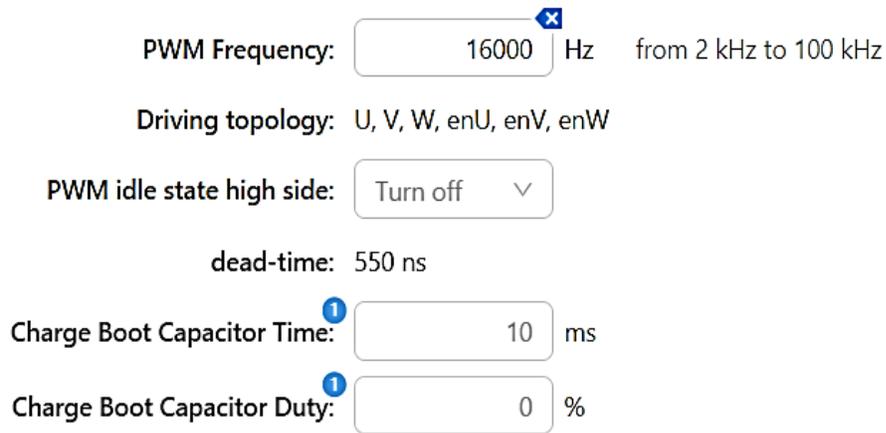


Fig. 6 PWM configuration

### Current regulator



#### Torque (Iq) current regulator

Target: from speed regulator  
 P: 3217 / 16384 0.19635  
 I: 1005 / 16384 0.06134

#### Flux (Id) current regulator

Target: 0 A  
 P: 3217 / 16384 0.19635  
 I: 1005 / 16384 0.06134

Fig. 7 Driver setting

The PI values can be calculated and verified using the standard  $K_p$  and  $K_i$  tuning formulas for the PMSM motor, as shown in the equation below. The same is represented in the workbench as shown in Figure 8.

Proportional gain:  $K_p = L_s \times \tau_c = (32 \times 10^{-6}) \times 6000 = 0.19635$

Integral Gain:  $K_i = R_s \times \tau_c = 0.16 \times 6000 = 960$

$$K_i^{digital} = K_i \times T_e = 960 \times (63 \times 10^{-6}) = 0.0613$$

Digitally stored values in STM firmware:

$$K_p^{Q14} = 0.19635 \times 16384 = 3217$$

$$K_i^{Q14} = 0.0613 \times 16384 = 1005$$

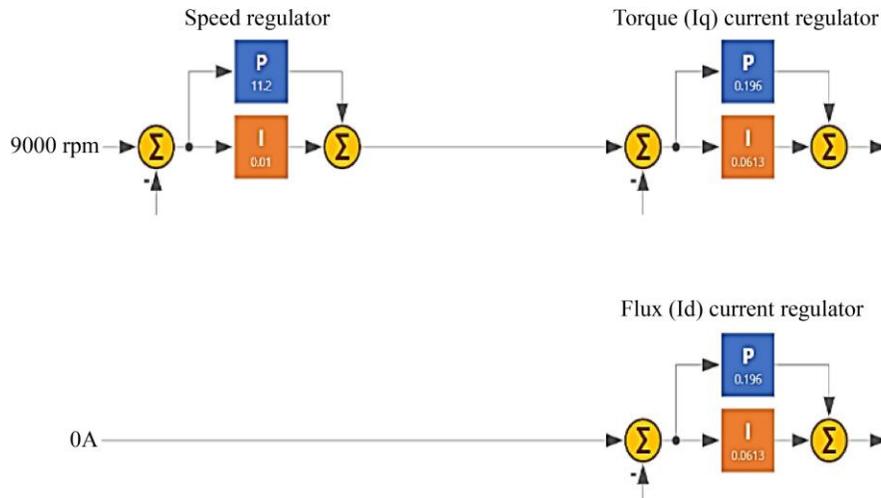


Fig. 8 PI Configuration

All signals and sensing information are mapped to physical pins on the STM32 microcontroller. This includes pins for PWM outputs to the ESC, analog inputs for current sensing, digital or analog inputs for the IR sensor, and an analog reference from the potentiometer. Once all setup steps are done, the user generates the project code. The workbench creates firmware for STM32 that is ready to compile and flash using tools like STM32CubeIDE. The generated code includes all the settings from the workbench and is ready for testing with the hardware setup [9, 10].

#### 4. Hardware Implementation

The FOC control of the PMSM motor is developed using the configuration, and the circuit diagram shows a hardware setup for controlling a 3-phase motor in closed-loop operation using the STM32 Nucleo board, as seen in Figure 9. The main power comes from a 12V DC power source that works for both the ESC and the STM32 controller. The ESC is connected directly to the STM32 and acts as a link between the controller and the motor, turning control signals into commands that control the motor's coils.

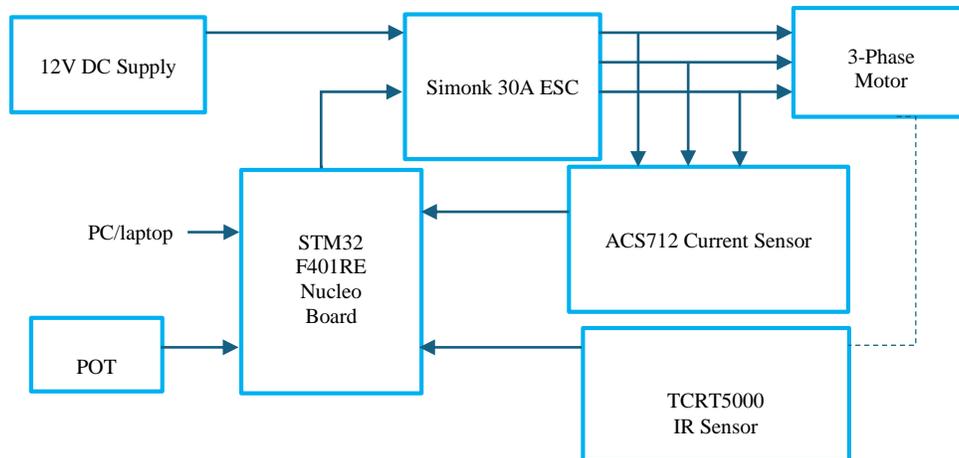


Fig. 9 Circuit diagram of FOC-based PMSM motor

To get real-time information about the motor's performance, an ACS712 current sensor is used to measure the current in each motor phase. The sensor sends this data as an analog signal to the STM32, which uses it to adjust the motor's operation and protect it from too much current. An IR sensor is connected to the STM32, giving extra feedback that can be used for monitoring the motor's position or speed. The hardware setup and implementation of FOC can be observed as shown in Figure 10.

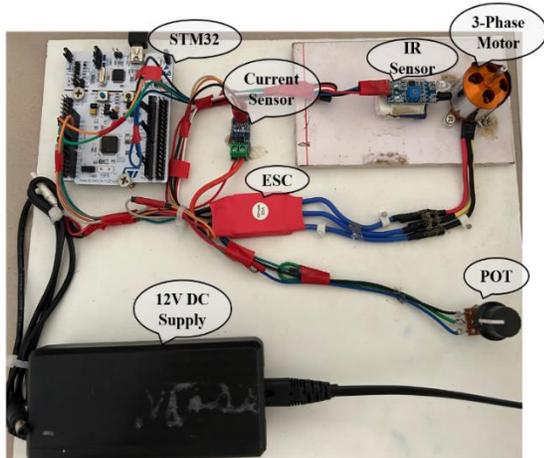


Fig. 10 Implementation of FOC-based PMSM motor

The user can adjust motor speed using a potentiometer that gives a variable voltage to the STM32. This lets the user change settings like the motor's speed or torque manually. The Nucleo board is also connected to a computer, allowing the user to write code. The output can be observed in the serial monitor or plotter and noted. This setup with the component specifications is shown in Table 1 along with their ratings and the application of their use [11, 12].

Table 1. Hardware specifications

Component	Key Specifications	Application
STM32F401RE	32-bit ARM Cortex-M3, 72MHz, 64kB to 1 MB Flash	Real-time control and signal processing
A2212 6T 2200KV 3-Phase motor	12 to 24V, 3-Phase, 2200 rpm/Volt, 0.5 to 2A	Actuator for mechanical motion
Simonk 30A ESC	30A cont., 6 to 12V, PWM 16kHz	Motor commutation and power control
ACS712 Current Sensor	Upto +30A, 4.5 to 5.5V, 100mV/A	Current monitoring for feedback/protection
IR Sensor TCRT5000	3.3V/5V range upto 20cm	Object detection or speed feedback
Potentiometer	10k to 50k ohm, 1.8 to 5.5V	User-defined reference input
12V DC Supply	100 to 270V AC in., 12V/5A DC out., >85% efficient	Main system power

## 5. Results and Discussion

For the implementation of the STM32-based motor control system using user-defined speed feedback, regulated through FOC logic, the hardware testing allowed data logging and parameter adjustments, facilitating faster convergence to optimal controller settings. The serial monitor displayed the voltage, current, torque, and speed values for variable speed through the potentiometer as seen in Figure 11. The initial torque ripple is observed to be 0.13 Nm under the FOC control for a corresponding speed of 2500 rpm. The torque ripples are reduced and maintain a constant value for varying speed.

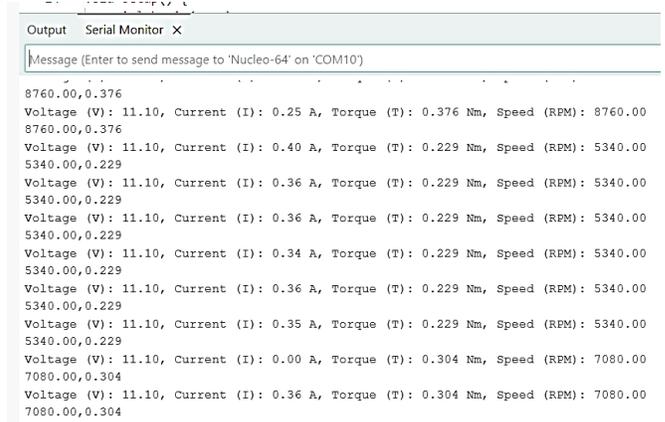


Fig. 11 Serial monitor initial speed output

For reference, at speeds of 9000 rpm and above, the torque decreases and maintains a constant value of 0.05 Nm, as shown in Figure 12. The peak torque is observed to be 0.45 Nm at a speed of 9000 rpm. The torque then decreases as the speed further increases.

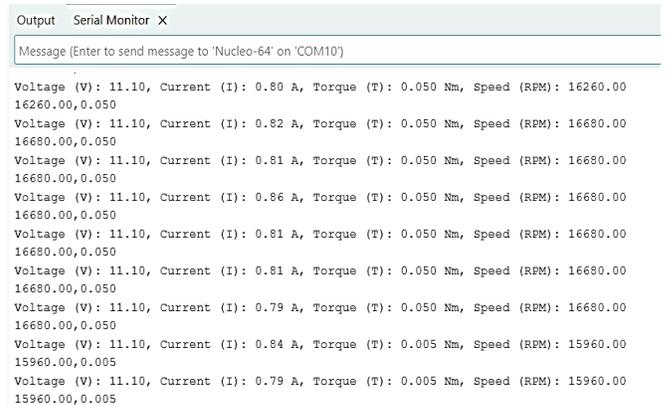


Fig. 12 Serial monitor constant torque output

From a visual inspection standpoint, the waveform representations of torque and speed via the serial interface or serial plotter corroborated the system's balanced three-phase output. The torque vs speed waveform confirmed that the torque ripples were minimized by adapting the FOC algorithm for speed control of the PMSM motor, as shown in Figures 13 and 14.

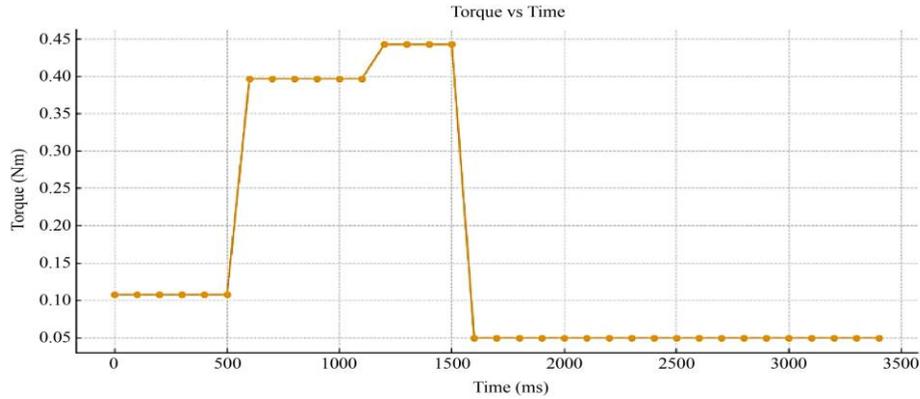


Fig. 13 Serial plotter torque output

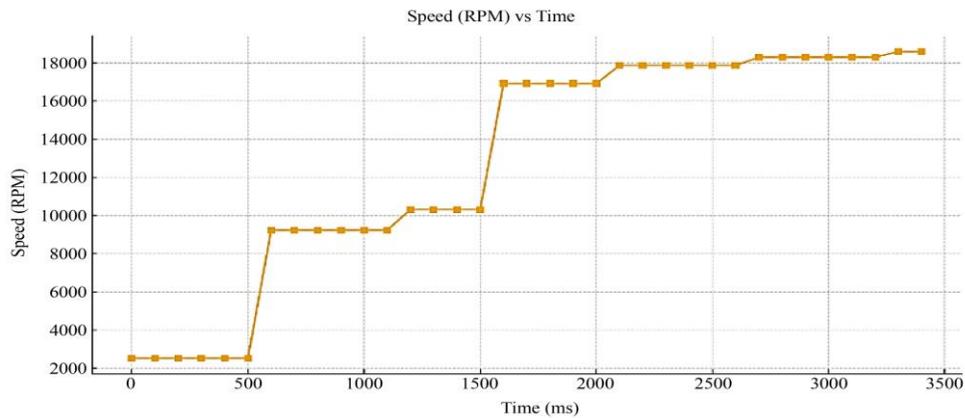


Fig. 14 Serial plotter speed output

The FOC algorithm is verified to minimize torque ripples by comparing it to six-step commutation for a similar setup as shown in Table 2. Referring to the Journal “Novel Sensor-less Six-Step Commutation Strategy for a Surface Permanent Magnet Synchronous Motor with DC Link Measurement,” the torque observed initially is around 0.6Nm at 2500rpm, which is 20% of the rated torque. By implementing the FOC algorithm, the torque observed with reduced ripples around 0.15Nm, which constitutes 5% of the total torque. According to the IEEE standard (IEEE 181-2011), the Torque ripple % for Permanent Magnet motors should be between 5-10% [13, 14].

Torque Ripple Calculation:

$$\text{Torque ripple (\%)} = \frac{(T_{\text{max}} - T_{\text{min}}) / T_{\text{avg}}}{1} \times 100$$

$$= \frac{(0.45 - 0.05) / 6.0}{1} \times 100$$

$$\text{Torque ripple (\%)} = 6.6\%$$

Table 2. Six-step commutation Vs FOC.

Control Method	Speed (rpm)	Torque (Nm)	Torque ripple (%)
Six-Step Commutation	2500	0.6	20%
	9000	1.3	
Field Orientation Control (FOC)	2500	0.15	6.6%
	9000	0.45	

## 6. Conclusion

This project successfully achieved the implementation of an FOC-based speed controller for a PMSM with the primary objective of minimizing torque ripple. The FOC strategy was deployed on the STM32F401RE microcontroller using the STM32 Motor Control Workbench and Arduino IDE.

A sensorless control scheme utilizing back-EMF estimation made the system cost-effective and compact. Hardware validation confirmed that the controller met the objectives and is a better approach compared to six-step commutation, with results aligning with IEEE standards for torque control and speed stability.

Overall, the system demonstrates a reliable, low-cost solution suitable for high-performance motor applications in industries such as robotics, electric vehicles, and automation.

## Conflicts of Interest

The developed project successfully generates torque for various speeds, and the same could be monitored via the serial monitor and serial plotter. Whereas the developed project was unable to plot the three-phase currents and display the. Thus, making the internal operation difficult to analyse.

## References

- [1] Binyu Wu et al., "Field-Oriented Control and Direct Torque Control for a Five-Phase Fault-Tolerant Flux-Switching Permanent-Magnet Motor," *Chinese Journal of Electrical Engineering*, vol. 4, no. 4, pp. 48-56, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Vinita S. Bondre, and Archana G. Thosar, "Study of Control Techniques for Torque Ripple Reduction in BLDC Motor," *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, India, pp. 1-6, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Ying Zuo, Chunyan Lai, and K. Lakshmi Varaha Iyer, "A Review of Sliding Mode Observer Based Sensorless Control Methods for PMSM Drive," *IEEE Transactions on Power Electronics*, vol. 38, no. 9, pp. 11352-11367, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Alessandro Bosso, Andrea Tilli, and Christian Conficoni, "A Robust Sensorless Controller-Observer Strategy for PMSMs with Unknown Resistance and Mechanical Model," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1536-1542, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] P. Ramesh et al., "Sensor-less Field Orientation Control for Brushless Direct Current Motor Controller for Electric Vehicles," *Materialtoday: Proceedings*, vol. 65, part 1, pp. 277-284, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Kayhan Gulez, Ali Ahmed Adam, and Halit Pastaci, "Torque Ripple and EMI Noise Minimization in PMSM Using Active Filter Topology and Field-Oriented Control," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 251-257, 2008. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Aashlesha Rajendra Raut, and Sadhana V. Jadhav, "Model Predictive Speed Control of Permanent Magnet Synchronous Motor," *IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, Jaipur, India, pp. 1-6, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Anant Kumar Singh et al., "Limitations of the PI Control with Respect to Parameter Variation in PMSM Motor Drive Systems," *IEEE International Electric Machines & Drives Conference (IEMDC)*, San Diego, CA, USA, pp. 1688-1693, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] José Carlos Gamazo-Real, Ernesto Vázquez-Sánchez, and Jaime Gómez-Gil, "Position and Speed Control of Brushless DC Motors using Sensorless Techniques and Application Trends", *Sensors*, vol. 10, no. 7, pp. 6901-6947, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Shivani Gupta, Saly George, and Viraj Awate, "PI Controller Design and Application for SVPWM Switching Technique Based FOC of PMSM," *2023 Second International Conference on Trends in Electrical, Electronics, and Computer Engineering (TEECCON)*, Bangalore, India, pp. 172-177, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] V.S. Veena, and R.P. Praveen, "Mathematical Modeling of Advanced PMPMSM Motor Drive for Aerospace Application," *2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMMD)*, Kottayam, India, pp. 1-5, 2014. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Thanh-Lam Le, "A Robust Control Strategy for Effective Field-Oriented Control of PMSMs," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18469-18475, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] A. Munteanu Ungurean et al., "Novel Sensorless Six-Step Commutation Strategy for a Surface Permanent Magnet Synchronous Motor with DC Link Measurement," *13<sup>th</sup> International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, Brasov, Romania, pp. 521-528, 2012. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] "IEEE Guide for Testing Permanent Magnet Machines," *IEEE Std 1812-2023 (Revision of IEEE Std 1812-2014)*, pp. 1-88, 2023. [[CrossRef](#)] [[Publisher Link](#)]