machines

MDPI

Article

Co-Simulation Comparison Analysis of SPWM and SVPWM in FOC of an Outer Rotor PMSM: Genetic Algorithm Based Optimization for Conventional and Fractional-Order PI Controller

Büşra Çelik [1,2,*] and Tolgay Kara [3]

[1] Graduate School of Natural and Applied Sciences, Department of Electrical and Electronics Engineering, Gaziantep University, 27310 Gaziantep, Turkey
[2] Electronics and Automation, Naci Topçuoğlu Vocational School, Gaziantep University, 27600 Gaziantep, Turkey
[3] Department of Electrical and Electronics Engineering, Gaziantep University, 27310 Gaziantep, Turkey
* Correspondence: busracelik@gantep.edu.tr

Abstract

Outer rotor permanent magnet synchronous motors (OPMSMs) are widely preferred in applications requiring high torque at low speeds due to their structural advantages. This study presents a co-simulation-based field-oriented control (FOC) of an OPMSM using ANSYS Maxwell, Twin Builder, and MATLAB/Simulink environments. The motor is modeled using the finite element method in ANSYS Maxwell, while the power electronics and control system are integrated through a co-simulation framework, enabling a realistic and high-fidelity analysis. The main objective of this study is to comparatively evaluate sinusoidal pulse width modulation (SPWM) and space vector pulse width modulation (SVPWM) techniques within the same control framework. To enhance the effectiveness and reliability of the FOC controller, a genetic algorithm (GA)-based optimization is employed to tune both conventional PI and fractional-order PI controllers in the speed control loop for the SVPWM method. The findings clearly show that SVPWM beats SPWM in all measured parameters, including current THD, DC-link voltage usage, and torque ripple, which is consistent with the well-known theoretical advantages stated in the literature. Furthermore, the GA-optimized fractional-order PI controller outperforms standard PI controllers in terms of overshoot, settling time, and torque ripple. Overall, the suggested co-simulation framework provides an effective platform for comparing modulation strategies, as well as illustrating the role of GA-tuned controllers in high-performance OPMSM drive systems.

Keywords: co-simulation; outer rotor PMSM; SPWM; SVPWM; genetic algorithm; fractional order PI

Academic Editor: Farshid Mahmouditabar

Received: 2 June 2026
Revised: 8 July 2026
Accepted: 10 July 2026
Published: 15 July 2026

1. Introduction

Permanent magnet synchronous motors (PMSMs) have become key components in modern drive systems due to their high efficiency, compact design, and better power density. PMSM development has accelerated over the past three decades, driven by advancements in magnetic materials such as permanent magnets [1] and electrical steels [2], as well as increasing demand from emerging applications, particularly in electric vehicles (EVs) [3,4]. PMSMs are frequently used in EVs, elevators, robotics, and industrial automation due to their advantages [5,6].

 https://doi.org/10.3390/machines14070801

The outer rotor permanent magnet synchronous motor (OPMSM) has clear advantages over other PMSM topologies, particularly in low-speed, high-torque applications. Because of their greater effective air-gap radius, decreased torque ripple, and enhanced mechanical stability, OPMSMs offer a higher torque density than traditional inner rotor machines. These characteristics make them ideal for uses like industrial automation systems, electric vehicles, and direct-drive wind turbines. Specifically, outer rotor OPMSMs have been extensively used in EV in-wheel motor technologies, unmanned aerial vehicles (UAVs), and hybrid propulsion systems [7].

A high-performance PMSM drive system is primarily based on a three-phase voltage source inverter (VSI) controlled by an appropriate pulse width modulation (PWM) method. The two most often used approaches are sinusoidal PWM (SPWM) and space vector PWM (SVPWM). SPWM is simple to build, but it has low DC bus utilization (about 78.5% of the DC-link voltage) and increased total harmonic distortion (THD). SVPWM is the preferred choice for high-performance drives due to its greater DC-link utilization (approximately 15.5% higher) and lower harmonic distortion achieved through optimized inverter switching states [8,9].

Recent improvements in control systems have expanded the capabilities of PMSM drives. Field-oriented control (FOC), which separates torque and flux channels, is the industry standard for providing superior dynamic performance. In field-oriented control (FOC), the three-phase motor adjusts current on the d–q axis to produce the appropriate torque or speed. Various control schemes, such as maximum torque per amp, constant torque angle, and proportional–integral (PI) control, are used to determine the torque component's reference value (iq). The stator current must be maximized to align the rotor and stator vector angles at 90° while maintaining the id component at zero. An inverter generates control signals using methods like pulse width modulation (PWM), space vector modulation (SVM), and space vector pulse width modulation (SVPWM). The d–q axis currents are adjusted based on dynamically monitored values, allowing for accurate motor speed or torque control in the feedback loop [10,11]. However, FOC's performance is significantly dependent on the accuracy of the motor model and the tuning of its controllers. Conventional integer-order PI controllers are widely used. Fractional-order PI controllers (PI^{λ}) are used as an alternative, offering an additional degree of freedom that increases robustness and transient response. However, optimal tuning of both classical and fractional-order controllers remains a difficulty, where metaheuristic algorithms such as genetic algorithms (GA) have proven particularly effective [5].

With substantial research on co-simulation, a full examination of the literature (summarized in Table 1) reveals a critical methodological gap. A large percentage of current research either uses a single modulation technique without direct comparison or relies on simplified analytical motor models. Co-simulation provides a superior choice by allowing numerous simulation platforms to work concurrently. Each subsystem is solved using its own numerical solver with continuous data sharing. This project creates a co-simulation framework in which data seamlessly flows between ANSYS software 2020 R1 (power electronics and motor) and MATLAB/Simulink 2018b (control algorithm). The SPWM and SVPWM switching methods are modeled using ANSYS/Simplorer, while the outer rotor OPMSM's two-dimensional finite element model is integrated directly using ANSYS/Maxwell. To ensure motor accuracy, each phase is connected with accurate phase resistance and coil inductance values. In addition, a load torque block is included to assess load response, and a moment of inertia block is attached to the motor's output to accurately represent mechanical qualities. The output block measures position and speed, which are subsequently relayed back to the controller to close the loop with high physical accuracy.

Table 1. Review of co-simulation studies.

Ref/Year	Machine & Tools	Control	Objective	Key Findings	Experiment
[12] (2017)	DC motor; Twin Builder + Simulink	Speed + current control	Present co-simulation methodology	Results match Simulink model; Simulink suits analysis, Twin Builder suits circuit design.	×
[13] (2019)	PMSG (3 kW); Maxwell + Simulink	Passive rectifier	Compare co-simulation with classical model	Co-simulation shows better agreement with experiments, especially during transients.	✓
[14] (2019)	PMSG (3 kW); Maxwell + Simulink	MPPT (OTC)	Wind turbine co-simulation	Core saturation captured; turbine stall observed at high wind speeds.	×
[15] (2019)	PMSG (1.5 kW); Maxwell + Simulink	DFOC	Detect partial demagnetization	D-axis current spectrum enables demagnetization detection; operation maintained up to 25% loss.	×
[16] (2019)	5-phase PMSM; Maxwell + Simulink	MTPA	Analyze demagnetization	Torque loss up to 50% at high temperature; demagnetization concentrated near air-gap.	×
[17] (2021)	5-phase PMSM; Maxwell 2D + PLECS	FOC, harmonic injection	Cross-platform validation	Enables saturation analysis; results consistent with classical simulation.	×
[18] (2021)	PMSM-EV; Simulink	FOC, SVPWM	Prevent demagnetization via temperature estimation	Temperature-based control prevents demagnetization under braking.	×
[19] (2021)	PMSG (2.2 kW); Maxwell + Simulink	FOC	Evaluate demagnetization under faults	Braking currents more critical than short-circuit; FOC maintains speed under 25% demagnetization.	✓
[20] (2022)	Induction motor (15 kW); Maxwell + Simulink	V/f control	Analyze IGBT short-circuit fault effects	Severe torque oscillations and asymmetric flux; fault detected via stator current analysis.	×
[21] (2023)	DS-PMSM; Maxwell + Simplorer	FOC, SVPWM	Compare winding layouts	Novel winding suppresses harmonics without degrading dynamics.	×
[22] (2023)	PMSM (3 kW); Maxwell 2D + Simplorer + Simulink	SVPWM, vector control	Improve simulation accuracy using the ECE model	THD improved by 36.8% and harmonic accuracy by 23.5%; results closely match experiments.	✓
[23] (2024)	PMSM-EV; Maxwell 2D + Simulink	MPC, OPP	Evaluate FEA-based optimization	FEA models ∼300× slower than analytical methods.	✓
[24] (2024)	DS-PMSM; Maxwell + Simulink	CPS-SVPWM	Reduce torque ripple	Torque ripple reduced up to 50%; magnetic losses decreased.	×
[25] (2025)	PMSM; Maxwell 2D + Twin Builder + Simulink	MTPA, PI, SPWM	Evaluate drive-cycle efficiency	Switching increases iron losses at low speed; minor efficiency difference between 5–10 kHz.	×
[26] (2025)	DTP-PMSM; Maxwell + PLECS	FOC, SVPWM	Develop unified machine model	Coordinate transformations eliminated; torque accuracy >99%.	✓
[27] (2025)	8/6 SRM; Maxwell 2D + Simulink	Hysteresis control	Compare converter topologies	Asymmetric converter gives best waveform; C-dump minimizes torque ripple.	×
[28] (2026)	HTS maglev PMSM; Maxwell + Simulink	FOC, SVPWM	Propose novel propulsion motor	Peak thrust 18.32 N; experimentally validated.	✓

https://doi.org/10.3390/machines14070801

Despite these developments, there is not a single study in the literature that includes (1) a direct comparison of SPWM and SVPWM within the same FOC framework; (2) GA-based optimization of both conventional PI and fractional-order PI controllers; and (3) a high-fidelity co-simulation environment that uses FEM to capture nonlinear electromagnetic effects. The majority of research focuses on just one of these characteristics, as Table 1 illustrates.

This study uses the combined ANSYS Maxwell, Simplorer, and MATLAB/Simulink platforms to provide a comprehensive co-simulation-based examination of an OPMSM in order to close this gap. This study makes three main contributions:

1. **A comparative analysis of modulation techniques:** In a realistic co-simulation situation, we directly evaluate SPWM and SVPWM's performance in a closed-loop FOC framework in terms of DC bus utilization, current THD, torque ripple, and dynamic speed response.
2. **GA-based controller optimization:** To improve transient performance, a genetic algorithm is utilized to optimize both conventional PI and fractional-order PI (PI^{λ}) controllers in the speed control loop using the ITAE, ISE, IAE, and ITSE criteria.
3. **High-quality co-simulation framework**: Unlike simplified Simulink-only models, our framework accurately represents electromagnetic nonlinearities, phase resistance/inductance effects, and mechanical dynamics (inertia, load torque) by employing ANSYS Maxwell's FEM solver for the motor, Simplorer for inverter switching circuits, and MATLAB for control logic.

The remainder of this paper is organized as follows: Section 2 presents the finite element method (FEM) model and design specifications of the OPMSM. Section 3 describe the SPWM and SVPWM techniques, respectively. Section 4 explains the FOC method and implemented speed controller methods, conventional PI and fractional-order PI. Section 5 outlines the genetic algorithm and GA-based optimization procedure for both controller types. Section 6 provides a comparative analysis of SPWM and SVPWM under FOC, followed by the results of the previous section, the GA-optimized controllers applied, and the results given in Section 7. Finally, Section 8 discusses the findings, presents the conclusions, and suggests directions for future research.

2. FEM Model of Outer Rotor PMSM

Outer rotor permanent magnet synchronous motors (OPMSMs) have a wider rotor radius, resulting in lower cogging torque and higher rotational inertia. Furthermore, due to the higher effective air-gap radius, more electromagnetic torque may be generated. Furthermore, outer rotor motors are suitable for applications requiring silent operation because they produce less acoustic noise and run more steadily at low speeds. Furthermore, outside rotor designs can have shorter axial lengths than inner rotor machines while delivering the same power output. Despite these advantages, outer rotor motor topologies have a few downsides. One of the most major disadvantages is increased potential. One of the key disadvantages is the increased possibility for heat buildup, which could lead to more thermal stress in the motor [7].

The OPMSM model was designed by using Ansys Maxwell. First, the RMxprt model was utilized to quickly define the motor's geometry, streamline the design space, and perform first performance assessments. Important motor characteristics were determined, including stator and rotor sizes, slot layout, stack length, winding configuration, insulating properties, and type of permanent magnet material. When the motor attained the necessary power, speed, and torque levels, the Rmxprt model produced a two-dimensional model. To check motor values such as torque, speed, and current and temperature evaluations were performed again for the 2D model. After the FEM model was completed, the 3D design of the motor was finalized and will be used in an experimental study [7,29]. Figure 1 shows the rmxprt model, the two-dimensional model, and the three-dimensional model of the motor.

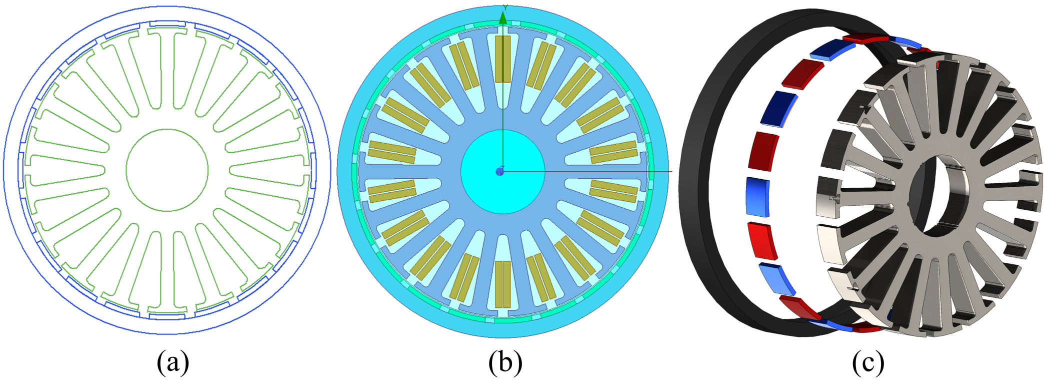

Figure 1. (**a**) Rmxprt model, (**b**) 2D model, and (**c**) 3D model of OPMSM.

During the design process, magnetic, geometric, and electrical components are considered. Static metrics like the motor's power, voltage, and number of poles remain constant, while additional criteria are used to evaluate performance. Table 2 summarizes the basic features of the OPMSM. Every rotor configuration, tooth width, stack length, and magnet property are optimized. Prior to putting the model into Ansys/Twin Builder for further analysis and co-simulation, the finite element analysis is augmented using specific mesh and boundary conditions.

Table 2. Technical specifications of the OPMSM.

Parameter	Value
Rated Power	125 W
Rated Speed	600 rpm
DC Bus Voltage	200 V
Number of Poles	20
Number of Slots	18
Phase Resistance	10.0589 Ω
Phase Inductance	0.000746879 H
Moment of Inertia	0.00331582 kg·m^2
Rated Torque	1.95 Nm
Stator Material	JFE_Steel_35JN230
Rotor Material	Iron
Permanent Magnet Material	NdFe30
Winding Configuration	Concentrated
Efficiency	83.7782%

3. PWM-Based Modulation Strategies for VSI

By integrating the right PWM method, an inverter can be efficiently controlled. In a three-phase, two-level VSI, switching signals control six power semiconductor switches (S1–S6). When the higher transistor (S1, S3, S5) is turned on, the lower transistor (S4, S2, S6) should be turned off. Consequently, 3-phase AC voltages produced from DC bus voltages are employed to power the motor. Space vector PWM and sinusoidal PWM are common methods for regulating VSI switches in a PMSM drive system [8].

3.1. Sinusoidal PWM Technique

Sinusoidal pulse width modulation (SPWM) is a popular modulation technique used in voltage-source inverters (VSIs) to generate alternating voltage at various frequencies and amplitudes. This method produces switching pulses by comparing a high-frequency triangular carrier signal to sinusoidal reference voltage signals that represent the fundamental component at the inverter output. In the SPWM method, the amplitude and frequency modulation indices determine the amplitude and frequency of the output voltage, respectively [9].

$$m_a = \frac{V_{\text{cont}}}{V_{\text{car}}}, \qquad m_f = \frac{f_{\text{car}}}{f_{\text{cont}}} \tag{1}$$

Linear operating region for SPWM,

$$m_f \geq 9 \quad \text{ve} \quad m_a \leq 1, \tag{2}$$

is defined under these conditions. Under these conditions, the amplitudes of the fundamental phase and line voltages are expressed as follows, where the DC bus voltage is V_d:

$$V_{f1} = \frac{1}{\sqrt{2}} \cdot \frac{m_a V_d}{2}, \qquad V_{h1} = \sqrt{3} V_{f1} \tag{3}$$

In the SPWM method, the maximum fundamental phase voltage accessible under the condition $m_a \leq 1$ is around 78.5% of the DC bus voltage. This scenario restricts the SPWM method's DC voltage consumption ratio. The SPWM technique is frequently utilized in power electronics and motor drives because of its simple design and implementation. As SPWM provides natural filtering by moving harmonic components to higher frequencies, switching frequencies are typically chosen in the 2 to 15 kHz range to minimize switching losses [8].

3.2. Space Vector Pulse Width Modulation (SVPWM)

In two-level three-phase voltage source inverters (VSIs), space vector pulse width modulation (SVPWM) is a complex technique for generating the reference voltage by modeling three-phase output voltages as a rotating space vector. SVPWM outperforms traditional sinusoidal pulse width modulation (SPWM) in terms of DC bus voltage usage, harmonic distortion, and output voltage quality. This makes it excellent for high-performance motor drives and power electronics [30]. A summary block diagram showing the procedure of the SVPWM approach is given in Figure 2.

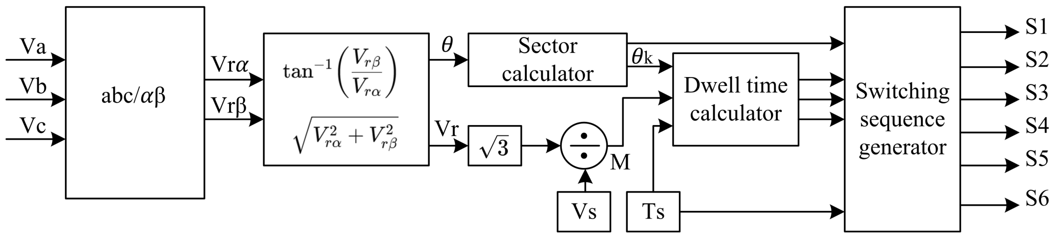

Figure 2. Summary block diagram of the SVPWM procedure.

The ratio of the reference voltage magnitude to the highest voltage that can be reached within the linear modulation zone is represented by the modulation index m in SVPWM.

$$m = \frac{\sqrt{3}\, V_{\text{ref}}}{V_{dc}} \tag{4}$$

Step 1: Clarke Transformation and Reference Vector Calculation

In SVPWM, the three-phase reference voltages are first transformed into a stationary orthogonal reference frame using the Clarke transformation. This transformation converts the three-phase quantities into two orthogonal components (V_α and V_β) in the α–β plane.

The reference voltage vector V_{ref} is then defined by its magnitude and angular position as follows [31]:

$$V_\alpha = V_{an} - \frac{1}{2}(V_{bn} + V_{cn}) \tag{5}$$

$$V_\beta = \frac{\sqrt{3}}{2}V_{bn} - \frac{\sqrt{3}}{2}V_{cn} \tag{6}$$

$$|V_{ref}| = \sqrt{V_\alpha^2 + V_\beta^2} \tag{7}$$

$$\alpha = \tan^{-1}\left(\frac{V_\beta}{V_\alpha}\right) \tag{8}$$

Step 2: Sector Identification

Based on the calculated angle α, the sector in which the reference voltage vector is located is identified. The α–β plane is divided into six equal sectors, each spanning 60°. There are eight possible switching configurations for a two-level three-phase VSI. Six of them are active voltage vectors (V_1–V_6) with the same magnitude and are spatially separated by 60° in the α–β plane. The two remaining states, V_0 and V_7, are called zero-voltage vectors. The reference voltage vector resides in one of these six sectors, which are formed by these active vectors dividing the space vector plane, as illustrated in Figure 3 [32].

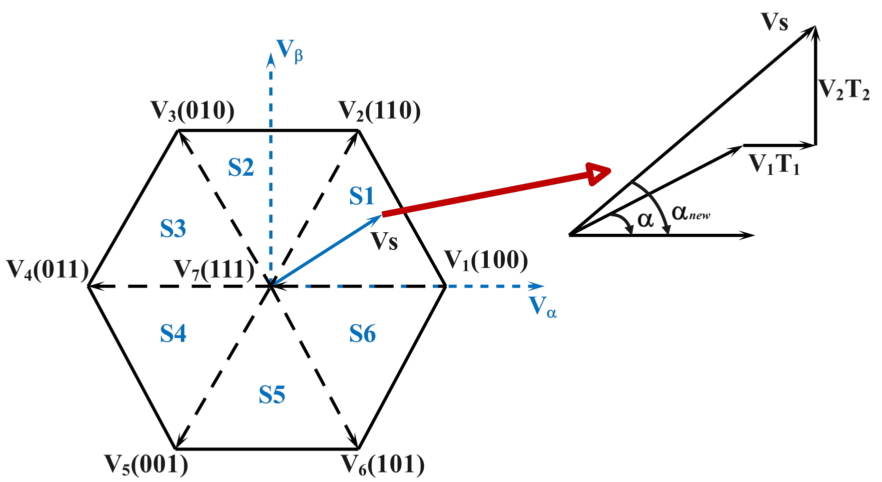

Figure 3. Space vector representation of SVPWM showing six sectors and active voltage vectors.

Step 3: Dwell Time Calculation

The dwell periods of the zero-voltage vector and the two nearby active voltage vectors are computed once the sector has been determined. These dwell times are denoted as T_1, T_2, and T_0, respectively, and they satisfy the following condition [33]:

$$T_1 + T_2 + T_0 = T_s \tag{9}$$

where T_s is the switching period. The dwell times are calculated as follows [33]:

$$T_1 = \frac{\sqrt{3}\,T_s\,V_{ref}}{V_{dc}}\left(\sin\frac{n\pi}{3}\cos\alpha - \cos\frac{n\pi}{3}\sin\alpha\right) \tag{10}$$

$$T_2 = \frac{\sqrt{3}\,T_s\,V_{ref}}{V_{dc}}\left(-\sin\frac{(n-1)\pi}{3}\cos\alpha + \cos\frac{(n-1)\pi}{3}\sin\alpha\right) \tag{11}$$

$$T_0 = T_s - T_1 - T_2 \tag{12}$$

Step 4: Switching Sequence and ON-Time Determination

The switching sequence is created based on the detected sector and predicted dwell periods. To eliminate torque ripple and minimize switching losses, the switching sequence is often chosen to be symmetrical throughout each switching interval. The zero vector time is applied evenly at the beginning and end of the switching cycle. The modulation cycle is completed by inverting the switching pattern [31]. The switching sequences for all six sectors are listed in Table 3.

Table 3. Switching sequences for six sectors.

Sector	Switching Sequence
I	$V_0\ V_1\ V_2\ V_7\ V_2\ V_1\ V_0$
II	$V_0\ V_2\ V_3\ V_7\ V_3\ V_2\ V_0$
III	$V_0\ V_3\ V_4\ V_7\ V_4\ V_3\ V_0$
IV	$V_0\ V_4\ V_5\ V_7\ V_5\ V_4\ V_0$
V	$V_0\ V_5\ V_6\ V_7\ V_6\ V_5\ V_0$
VI	$V_0\ V_6\ V_1\ V_7\ V_1\ V_6\ V_0$

As a result, the SVPWM method has several important advantages, including the following [34]:

- Improved DC bus voltage utilization: Achieving approximately 90.7% of the DC bus voltage, which is 15.5% higher than traditional SPWM.
- Constant switching frequency for predictable losses.
- Better dynamic performance during transients.
- Lower total harmonic distortion (THD) and smoother motor currents.

Despite being more complex than SPWM, SVPWM outperforms in terms of efficiency, voltage consumption, and output quality, making it the preferred choice for digital signal processor-based implementations, renewable energy systems, uninterruptible power supply, and high-performance motor drives.

4. Field-Oriented Control—Speed Control

Field-oriented control (FOC) maintains consistent motor performance by maintaining the reference speed or torque via dual control loops for speed and current. FOC increases dynamic system behavior, resulting in shorter reaction times and better torque and speed performance across a wide range. Figure 4 illustrates how a three-phase motor manages current on the d–q axis to produce the desired reference torque or speed. Several control schemes are used to calculate the reference value (i_q) of the torque component, such as maximum torque per amp, constant torque angle, and the traditional $i_d = 0$. To begin control, the reference speed or torque must be supplied. To achieve the biggest torque output, the stator current must be maximized so that the rotor and stator vector angles are aligned at 90°. This involves keeping the i_d component at zero and optimizing the stator current's i_q component. This control is carried out utilizing the switching signals generated by the inverter using methods such as pulse width modulation (PWM), space vector modulation (SVM), or space vector pulse width modulation (SVPWM). The d–q axis currents are adjusted based on the dynamically measured d–q axis current and the motor's real speed or torque. Feeding the d–q axis voltage values from this control output into the selected switching mechanism allows for precise motor speed or torque control within the feedback loop [11].

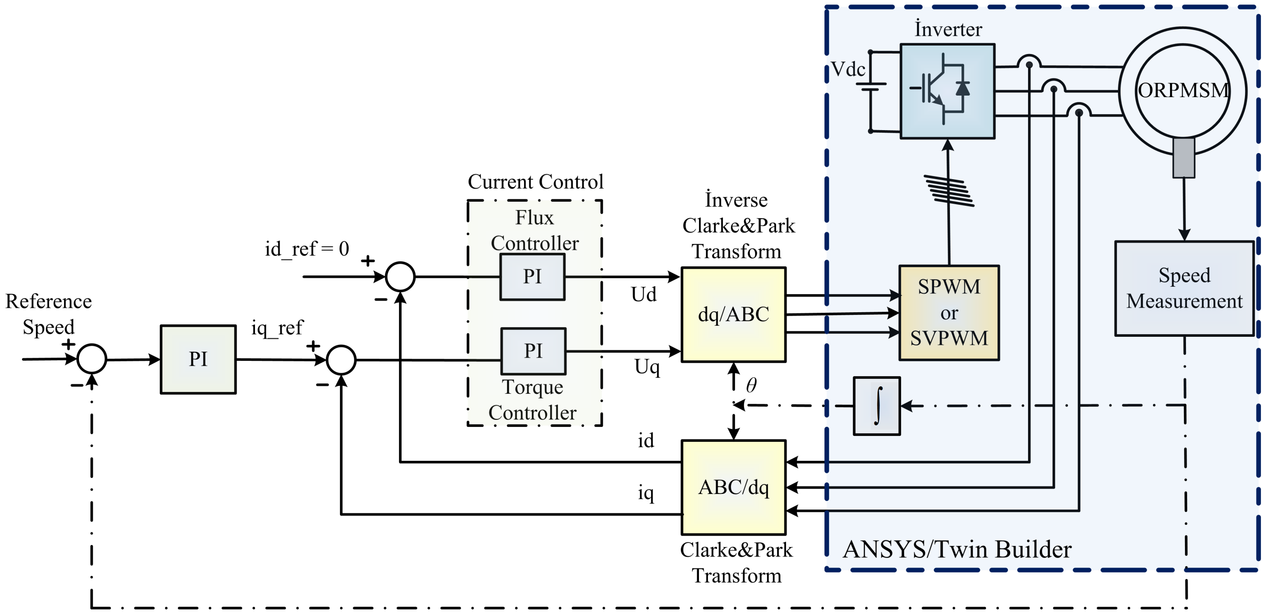

Figure 4. Co-simulation of OPMSM FOC block diagram.

The MATLAB and ANSYS implementation of power circuits and control loops is shown in Figure 4. The finite element (FEM) motor model imported from ANSYS Maxwell has been merged with the inverter and modulation techniques in ANSYS Twin Builder, while control loops are implemented in MATLAB Simulink. The co-simulation framework that emerges from the interaction between the two settings is the primary focus of this work.

Conventional PI and fractional-order PI controllers: Fractional-order PI (PI^{λ}) controllers add a fractional integral term to the traditional PI structure, providing additional flexibility for controller designers. The controller is defined as follows [29]:

$$C(s) = k_p + \frac{k_i}{s^{\lambda}}, \quad 0 < \lambda < 1 \tag{13}$$

where λ indicates the fractional integral order and k_p and k_i indicate for the proportional and integral gains, respectively. The controller simplifies to the traditional PI structure when $\lambda = 1$. The controller can be more precisely tuned, improving dynamic response and robustness, by allowing λ to take noninteger values [35].

The digital implementation of the fractional-order PI controller requires a numerical approximation of the operator $1/s^{\lambda}$, which makes it more computationally demanding than the traditional PI controller even if it offers better tuning and resilience. The computational complexity and memory requirements are increased by this approximation using rational transfer functions or finite-order filters. However, when employing contemporary digital signal processors (DSPs) and microcontrollers, low-order approximations for permanent magnet synchronous motor (PMSM) drives are still possible [35]. The suggested controller is tested in a co-simulation setting in this study; real-time hardware implementation will be investigated in subsequent research. Furthermore, co-simulation analyses were performed for both the conventional PI and the proposed FOPI controllers under the identical conditions. The simulations were performed using a workstation equipped with an Intel® Core™ Ultra 9 285K CPU, 64 GB DDR5 RAM, and NVIDIA RTX 4070 GPU from Ekosan, Gaziantep, Turkey. Each 1.5 s co-simulation for FOPI and conventional PI took approximately 7.5 h to complete using these system characteristics. Comparison of conventional PI and fractional-order PI controllers can be found in Table 4.

Table 4. Comparison of conventional PI and fractional-order PI controllers [35].

Feature	Conventional PI	Fractional-Order PI (PI^{λ})
Controller structure	$k_p + \frac{k_i}{s}$	$k_p + \frac{k_i}{s^{\lambda}}$
Integral order	Fixed ($\lambda = 1$)	Adjustable ($0 < \lambda < 1$)
Number of parameters	2 (k_p, k_i)	3 (k_p, k_i, λ)
Flexibility	Limited	High
Tuning complexity	Simple	More complex
Dynamic performance	Standard	Improved tuning capability
Robustness	Moderate	Higher
Implementation	Easy	Requires fractional operator approximation

5. Genetic-Algorithm-Based Parameter Optimization

Compared to conventional optimization techniques, a genetic algorithm is a metaheuristic optimization technique that efficiently scans the solution space. It is more effective at solving the local minimum entrapment problem due to its stochastic structure, and it is appropriate for problems involving noisy or irregular signals because it is derivative-free. Each potential solution in this method is called a chromosome, and a generation is the process by which these solutions change over the course of subsequent generations. GAs use probabilistic transition rules to repeatedly improve the population of potential solutions. An objective function, parameters, and constraints are established, and the algorithm is run in compliance with a termination criterion, just like in other optimization techniques [36].

5.1. GA Fundamental Operators

GA is built upon three fundamental operators:

Selection: Individuals are evaluated based on their fitness function values, with the best individuals advancing to the next generation. Common approaches include roulette wheel selection, tournament selection, and rank-based selection. The elitism technique, in which the fittest people are transmitted down without selection, is also often utilized [37,38].

Crossover: Following the selection phase, chromosome pairs from the mating pool exchange genetic information, potentially producing superior offspring. Common strategies include single-point, multi-point, and arithmetic crossover [37,38].

Mutation: Small random alterations are performed to the offspring bit string to increase genetic variety and the algorithm's ability to reach a global optimum. A mutation rate of 1% to 5% is typically used to balance diversity with the risk of premature convergence [37,38].

The algorithmic workflow can be summarized as follows: Generate a random beginning population, compute fitness values, apply selection, crossover, and mutation, and verify the termination criterion. The cycle continues until the requirement is met. Increasing the number of iterations enhances the solution quality; nevertheless, the user determines the required number of iterations based on the nature of the problem [39].

5.2. Tuning of PI Parameters Using GA

A PI controller provides the control signal based on the error and its integral, which is mathematically stated as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau \tag{14}$$

The error signal is represented by $e(t)$, the proportional gain by K_p, and the integral gain by K_i. For both transient performance and steady-state accuracy, appropriate parameter selection is essential. In complex systems, traditional tuning methods may perform inadequately due to their reliance on basic assumptions [40].

The literature proposes various ways for tuning PI controller settings, including new optimization-based approaches and classic methods like Ziegler–Nichols, Cohen–Coon, and Tyreus–Luyben [36]. The genetic algorithm (GA) was chosen for this study over other metaheuristic methods due to its strong global search capability, derivative-free operation, and native availability within the MATLAB Optimization Toolbox, allowing for seamless integration with the existing co-simulation framework. Although alternatives such as particle swarm optimization (PSO) and artificial bee colony (ABC) have been utilized to comparable controller tuning difficulties, the GA provides well-established convergence behavior for multi-parameter optimization in motor drive applications [40]. The genetic algorithm (GA) is frequently used for intelligent tuning due to its capacity to search large parameter spaces without the need for derivative information or expert knowledge. The GA-based approach optimizes each chromosome's parameter vector $[K_p K_i]$ within predetermined constraints. The objective is to identify the set of parameters that minimizes a particular fitness function [40,41]. The most prevalent integral performance criteria are as follows [37]:

$$\text{ISE} = \int_0^\infty e^2(t)\,dt \tag{15}$$

$$\text{IAE} = \int_0^\infty |e(t)|\,dt \tag{16}$$

$$\text{ITSE} = \int_0^\infty t\,e^2(t)\,dt \tag{17}$$

$$\text{ITAE} = \int_0^\infty t\,|e(t)|\,dt \tag{18}$$

ISE improves initial response time and targets severe errors, but it is not particularly sensitive to small steady-state flaws. IAE aims to limit the overall accumulation of errors by assigning equal weight to each error. ITSE and ITAE use time weighting to prevent errors occurring later in the response period. ITAE increases both settling time and steady-state performance by taking error amount and duration into account, making it the ideal fitness function for GA-based PI tuning [37,40].

While a high K_i eliminates steady-state error at the expense of decreased stability, increasing K_p decreases steady-state error but may cause overshoot and oscillation. By minimizing the chosen performance index, most notably ITAE, GA iteratively finds the best balance between K_p and K_i. The GA process is as follows: a random beginning population is created and each individual's fitness value is assessed, superior individuals are found using the selection operator, crossover generates new candidate solutions, and mutation preserves population diversity. Until the target performance criterion or the specified number of generations are met, this cycle is repeated. GA-based PI controllers provide lower overshoot, faster settling times, and lower steady-state errors when compared to conventional techniques. Additionally, they show better adaptability to fluctuating load situations and disruptions [39,40].

https://doi.org/10.3390/machines14070801

6. Comparative Analysis and Results of SPWM and SVPWM Under FOC

The proposed approach is developed utilizing a MATLAB–ANSYS co-simulation framework, in which ANSYS Maxwell solves the outer rotor permanent magnet synchronous motor (OPMSM)'s finite element model while MATLAB/Simulink performs the closed-loop control method. The AnsoftSFunction interface connects the two environments and allows data exchange. The Simulink-based control structure is illustrated in Figure 5.

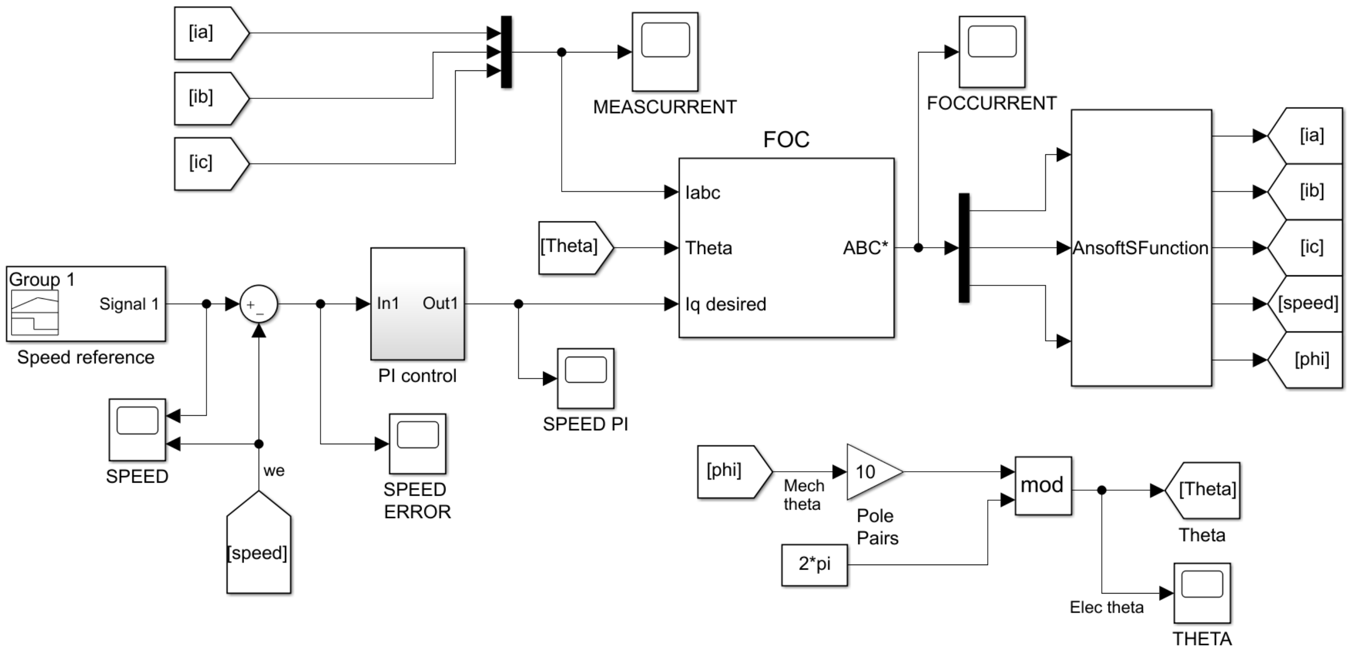

Figure 5. Block diagram of the closed-loop FOC-based speed control system in MATLAB/Simulink.

This structure calculates the speed error by comparing the reference speed ω^* with the measured rotor speed ω. The PI controller processes this error signal to provide the torque-generating current reference I_q^*. The field-oriented control (FOC) block computes the three-phase voltage references ABC* based on the measured phase currents $[i_a, i_b, i_c]$, the electrical angle θ_{elec}, and the reference I_q^*.

$N_p = 10$, so the electrical angle is obtained from the mechanical angle ϕ as

$$\theta_{\text{elec}} = \text{mod}\big(N_p\,\phi,\, 2\pi\big) \tag{19}$$

To provide a fair comparison, the SPWM and SVPWM approaches use the same control structure and controller parameters. The only difference between the two approaches is the modulation method, which is implemented in the ANSYS environment.

6.1. Closed-Loop Control with Sinusoidal PWM

The ANSYS implementation of the SPWM-based inverter and OPMSM model is shown in Figure 6.

The ABC* reference signals generated by the FOC block are sent to ANSYS through the AnsoftSFunction. The switching signals are generated by comparing these sinusoidal references to a high-frequency triangle carrier signal. The resultant switching signals operate the inverter switches, which generate the phase voltages used by the OPMSM. The electromagnetic response of the motor is simulated using finite element analysis in ANSYS, and the resulting phase currents, rotor speed, and mechanical angle are sent back into the Simulink model, effectively closing the control loop.

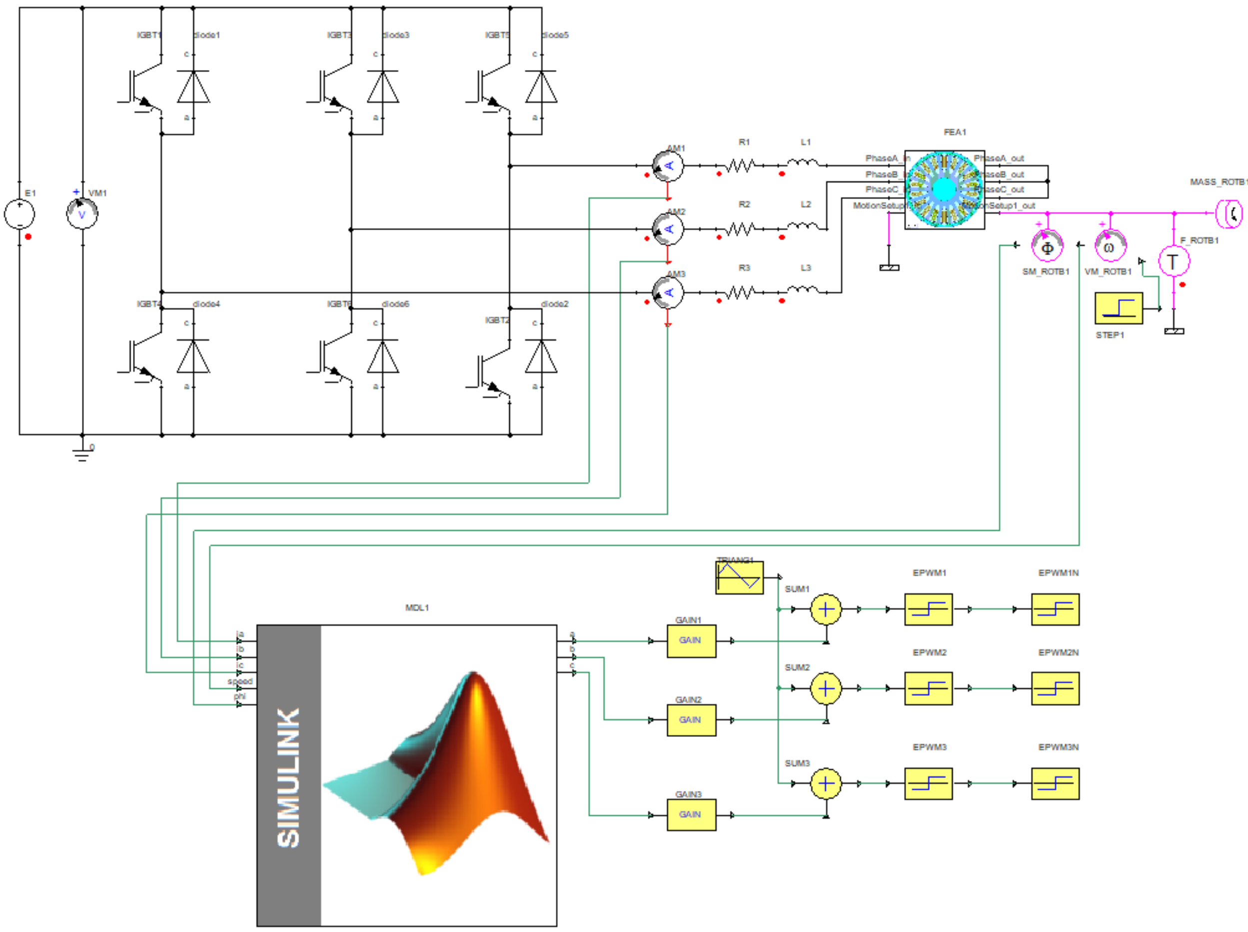

Figure 6. ANSYS/Twin builder simulation with FEM model of motor with SPWM-based inverter structure and MATLAB/Simulink connection.

6.2. Closed-Loop Control with Space Vector PWM

The ANSYS implementation of the SVPWM-based inverter and OPMSM model is shown in Figure 7.

The SVPWM approach transforms the ABC* reference signals into the α–β plane to generate a rotating space vector. The inverter switching states are governed by the sector of this vector. These times are utilized to construct the inverter switching sequence, which increases DC-link voltage usage and reduces harmonic distortion. As with SPWM, the inverter outputs feed the OPMSM model in ANSYS, and the computed electromagnetic variables are supplied back to Simulink to ensure closed-loop functioning. Overall, the control structure in MATLAB/Simulink remains unchanged, but the modulation approach has a major impact on system performance, particularly harmonic distortion, voltage usage, and dynamic response.

6.3. Results and Discussion

In this section, the performance of the OPMSM drive system was assessed using two modulation techniques: sinusoidal pulse width modulation (SPWM) and space vector pulse width modulation (SVPWM). The co-simulation environment created by ANSYS/Maxwell/Twin Builder and MATLAB/Simulink offered a high-fidelity platform for observing electromagnetic and dynamic responses.

The modulation signals in Figure 8 show the fundamental distinction between the two control techniques. SVPWM injects a zero-sequence voltage component to provide the distinctive saddle-shaped modulation signal, as seen in the upper panel. This method

https://doi.org/10.3390/machines14070801

can raise DC-link voltage consumption by roughly 15.5% when compared to conventional SPWM. The inverter may generate a higher fundamental voltage component without going into the overmodulation region thanks to the modulation signal's smoothed peak. The lower panel, on the other hand, shows pure sinusoidal references for SPWM, which are inherently constrained by the DC-link voltage peak. This restriction results in decreased voltage gain and early saturation at high speeds, which directly affects the drive's maximum speed and torque.

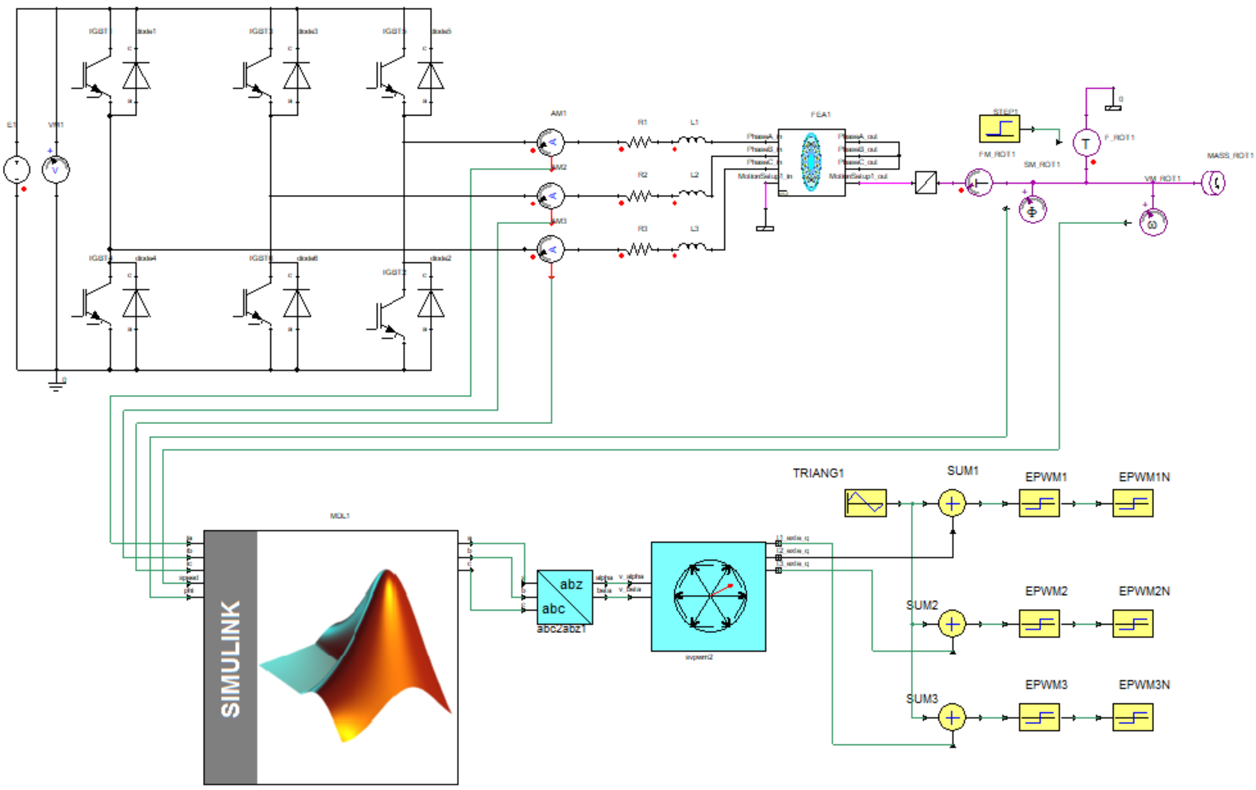

Figure 7. ANSYS/Twin builder simulation with FEM model of motor with SVPWM-based inverter structure and MATLAB/Simulink connection.

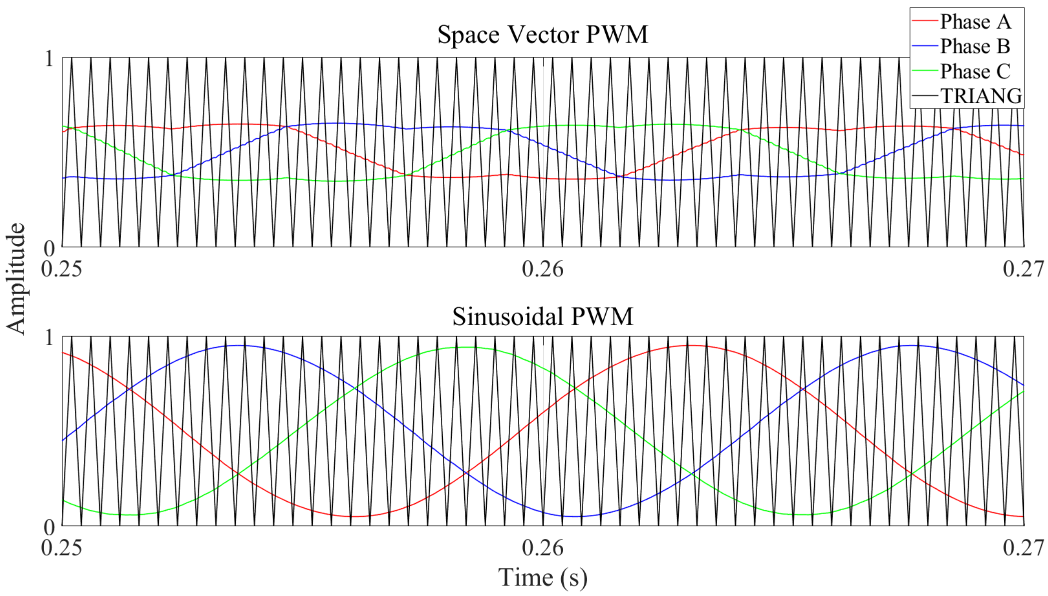

Figure 8. Comparison of modulation signals for SVPWM and SPWM.

As seen if Figure 9, a comparison of the space vector PWM (SVPWM) and sinusoidal PWM (SPWM) methods over the time interval t = 0.2–0.23 s reveals that SVPWM produces well-balanced three-phase currents with approximately 120° phase displacement and minimal distortion, resulting in high-quality sinusoidal waveforms. SPWM, on the other hand, deviates significantly from the ideal sinusoidal form, resulting in higher harmonic distortion and high-frequency ripples. Overall, SVPWM surpasses SPWM in waveform quality, leading in smaller losses, less thermal stress, and higher efficiency in motor drive applications.

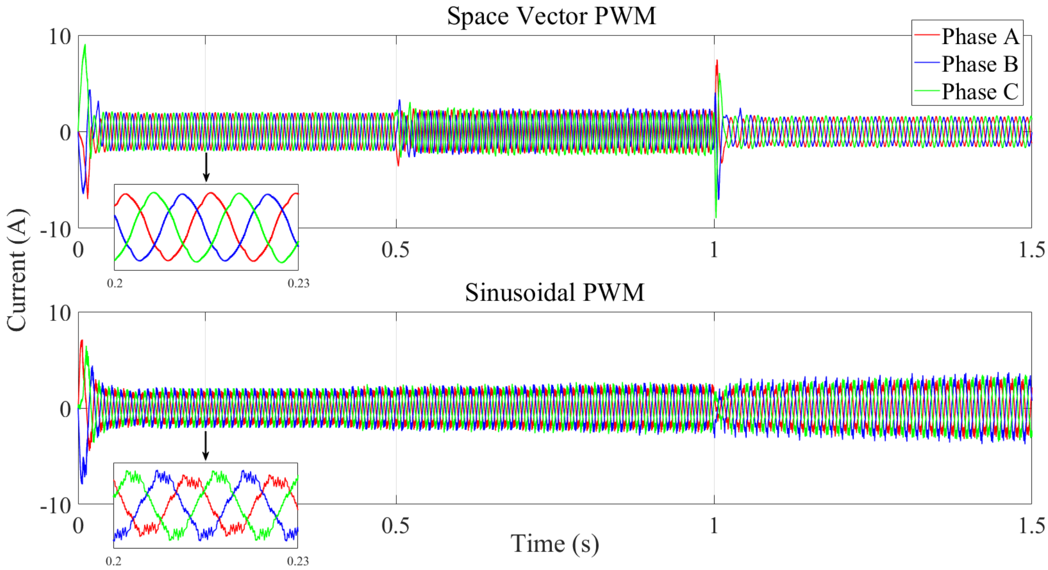

Figure 9. Three-phase current waveforms under SVPWM and SPWM.

Figure 10 shows the total harmonic distortion (THD) levels. SVPWM significantly decreases current harmonics throughout all three phases. In Phase A, THD reduces from 7.86% (SPWM) to 2.61% (SVPWM), a 66.8% drop. In Phase B, THD drops from 11.06% (SPWM) to 3.08% (SVPWM), a reduction of 72.2%. In Phase C, THD drops from 5.10% (SPWM) to 2.95% (SVPWM), a 42.2% reduction. The substantial THD reported in SPWM (up to 11.06% in Phase B) suggests higher copper losses and increased thermal stress on motor windings, thereby lowering the machine's operating lifetime. SVPWM achieves almost three times lower THD than SPWM.

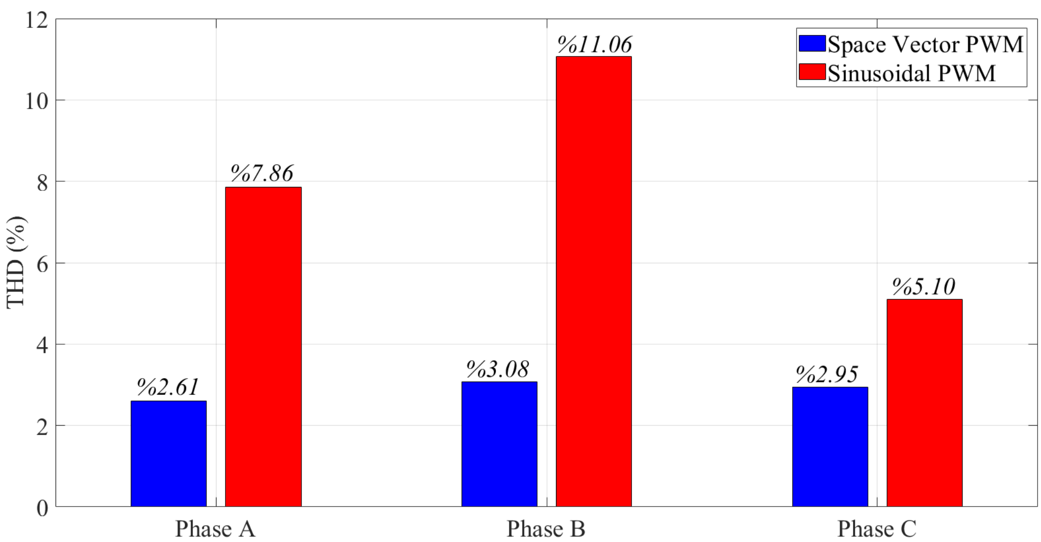

Figure 10. THD analysis of stator phase currents for both methods.

https://doi.org/10.3390/machines14070801

Figure 11 depicts dynamic speed tracking performance. The SVPWM controller maintains the reference speed with minimal settling time and no steady-state inaccuracy. SVPWM provides precise transient response with little overshoot during acceleration from 46 rad/s to 67 rad/s (at t = 0.5 s) and deceleration from 65.5 rad/s to 35 rad/s (at t = 1.00 s). However, the SPWM controller has significant tracking limitations. SPWM's lower voltage gain (due to 15.5% lower DC-link utilization) causes it to fall short of the reference speed during transient acceleration phases. Figure 11 shows that at some operating points, the SPWM response lags behind the reference and has a slight steady-state inaccuracy. This constraint renders SPWM unsuitable for high-dynamic-performance applications.

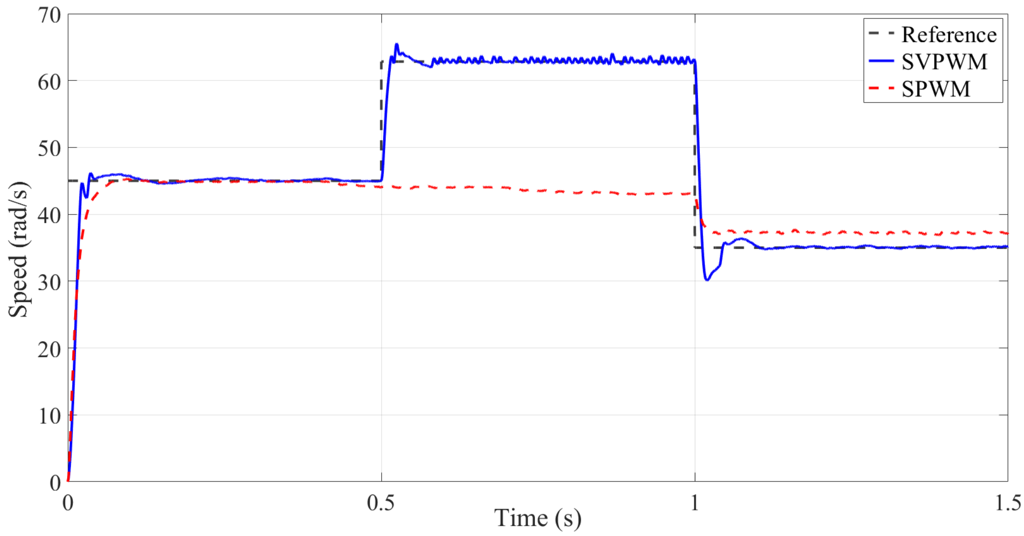

Figure 11. Dynamic speed response comparison under reference step changes.

Figure 12 displays the electromagnetic torque responses. It is clear that the chosen modulation approach affects the torque output capability during the initial transient (t = 0 s). In particular, the SVPWM approach produces a higher peak torque of 9.54 Nm as opposed to SPWM's 8.75 Nm. This discrepancy can be explained by SVPWM's improved DC bus voltage usage, which raises the inverter's effective phase voltage and boosts its capacity to produce electromagnetic torque at startup. Additionally, SVPWM shows a more responsive and consistent torque behavior throughout the intervals when the speed reference changes, while SPWM is unable to sufficiently track the required operating circumstances. The smaller linear modulation range and inadequate use of the DC bus voltage in SPWM are the main causes of this constraint, which limits the available voltage margin needed for dynamic torque generation. Degraded dynamic performance results from SPWM's inability to generate the torque required to adapt quickly to changes in the speed reference.

Figure 13 shows the way that the d–q axis current decoupling works. The torque requirement is minimal when the motor is operating without any load; hence, in steady-state, iq stays around 0 with only slight variations during transient events. Even though id's reference value is set to 0 A, in reality, it cannot be kept precisely at zero. Because of its larger ripple content and lower DC bus voltage consumption, SPWM shows more noticeable deviations, while SVPWM maintains id closer to the reference with smaller oscillations. While SPWM exhibits stronger oscillatory behavior and observable cross-coupling effects, SVPWM offers a quicker, more stable response with less oscillations under transitory situations.

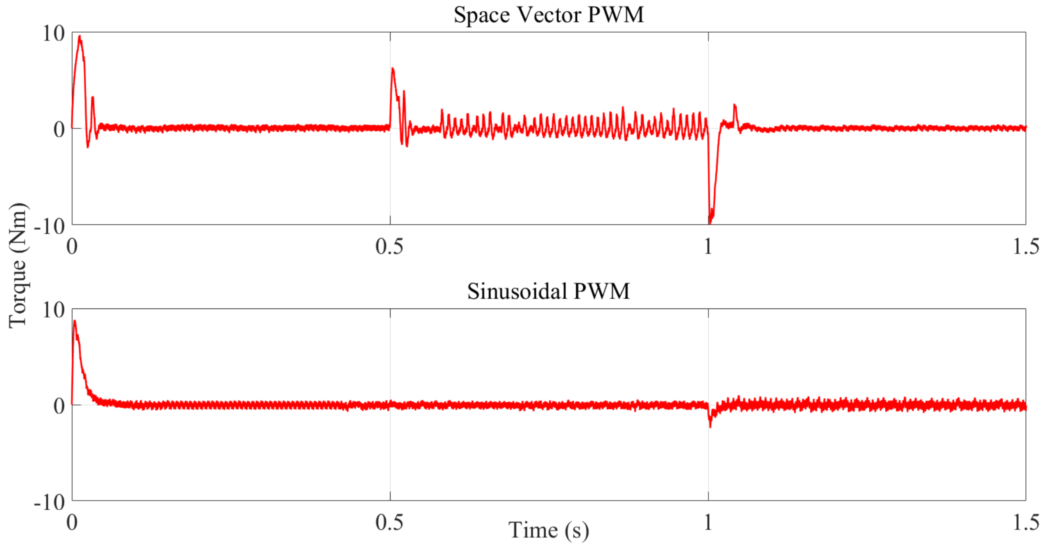

Figure 12. Electromagnetic torque comparison between SVPWM and SPWM.

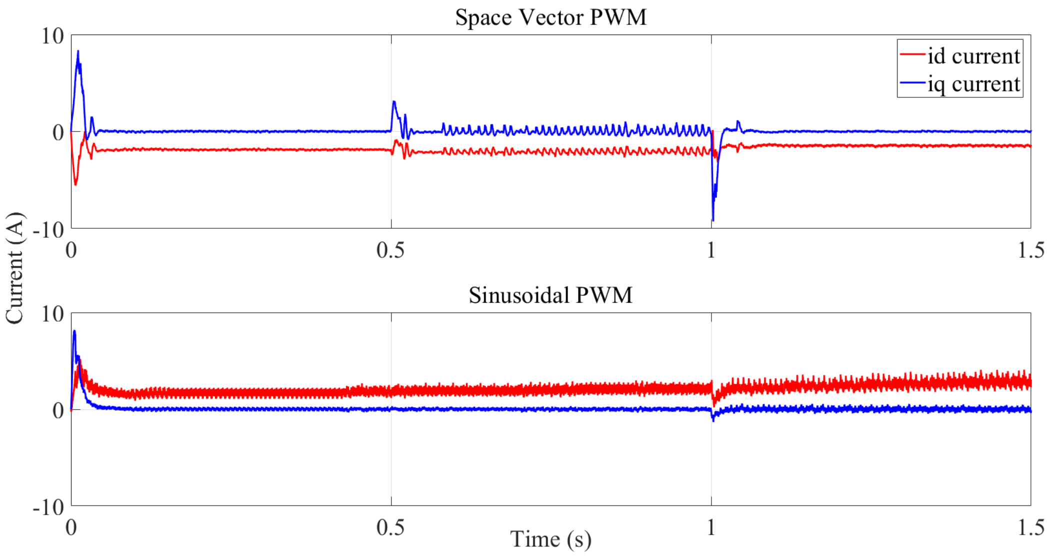

Figure 13. i_d and i_q current regulation in synchronous reference frame for SVPWM and SPWM.

Figure 14 illustrates how SPWM and SVPWM approaches differ under field-oriented control (FOC) when magnetic flux density (B) distributions are compared using finite element analysis (FEA). SPWM attained a maximum flux density of 2.0537 T, with uneven distributions and saturation zones due to the high harmonic content. SVPWM had a lower maximum flux density of 1.9169 T and a more uniform distribution, reducing stator tooth saturation through superior DC bus voltage consumption. This efficiency results in less iron loss, less torque ripple, and better thermal load distribution, demonstrating SVPWM's superior electromagnetic performance in FOC applications.

Figure 15 compares the current density (J) distributions obtained using SPWM and SVPWM under the same operating conditions. The SPWM technique has a greater peak current density of around 1.27×10^7 A/m^2, showing that the inverter runs closer to its modulation limit while striving to meet the torque demand. In comparison, SVPWM has a lower peak current density of around 6.04×10^6 A/m^2, which is due to its greater DC-link voltage usage. As a result, SVPWM can provide the necessary electromagnetic torque more effectively while maintaining precise reference speed monitoring, whereas SPWM has a steady-state speed error under the same operating conditions.

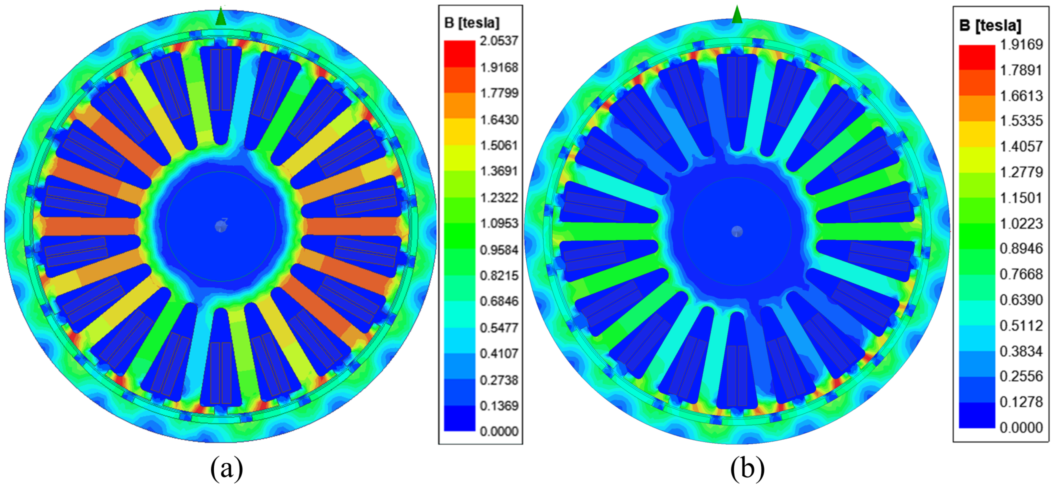

Figure 14. Magnetic flux density distribution obtained under FOC control: (**a**) SPWM, (**b**) SVPWM.

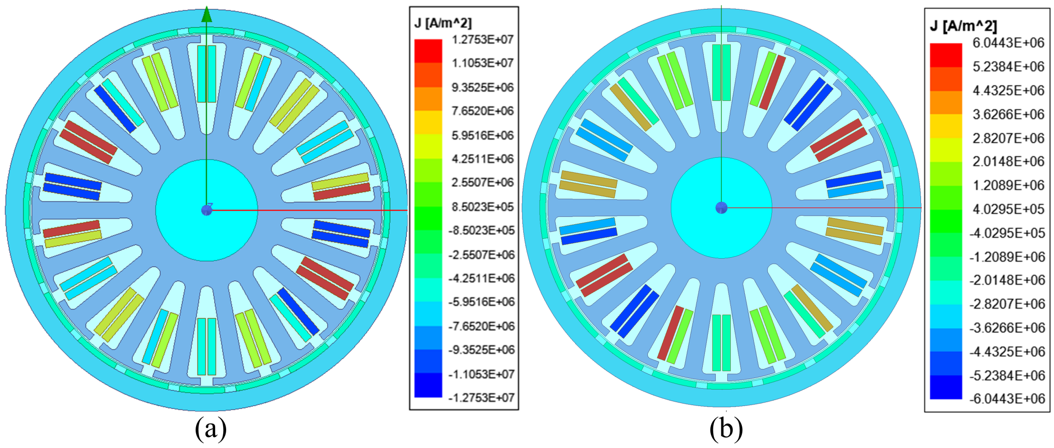

Figure 15. Current density distribution obtained under FOC control: (**a**) SPWM, (**b**) SVPWM.

Figure 16 illustrates the speed responses of the SPWM and SVPWM approaches to a 0.5 Nm load torque applied at t = 0.15 s. Both approaches properly regulate the motor speed around the 40 rad/s reference value; however, SPWM has a smoother transient response with less speed ripple, whereas SVPWM has a small speed dip followed by a quick recovery. Nonetheless, this conclusion is particular to the operating conditions under consideration, in which the motor voltage demand remains within SPWM's modulation capacity. As the load torque grows, SPWM's limited DC-link voltage usage forces it to exceed its modulation limit, resulting in a large steady-state speed inaccuracy.

The co-simulation results presented in figures, which are compiled in Table 5, show that SVPWM-based FOC outperforms conventional SPWM in every evaluated aspect. The following are the primary conclusions: (1) SVPWM achieves approximately three times lower THD, reducing copper losses and thermal stress on the motor windings; (2) SVPWM achieves 15.5% higher DC-link voltage utilization through zero-sequence voltage injection, which directly translates to higher maximum speed and torque capability; (3) SVPWM provides crisp, zero-error speed and current tracking, whereas SPWM exhibits cross-coupling effects in the d–q frame; (4) The finite element study results show that SVPWM reduces peak flux density, removing local saturation regions in the stator teeth and resulting in a more homogenous flux distribution. Furthermore, SVPWM's lower peak current density exhibits greater and more efficient electromagnetic torque generation.

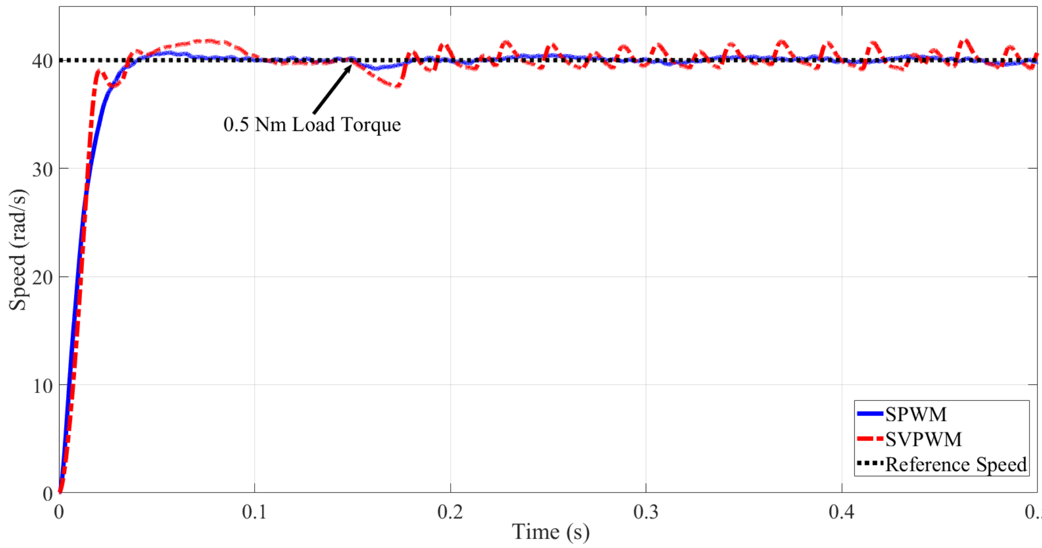

Figure 16. Speed response under load torque.

Table 5. Comprehensive performance comparison between SVPWM and SPWM for FOC-controlled OPMSM.

Performance Metric	SVPWM	SPWM	Improvement with SVPWM
Phase A THD (%)	2.61	7.86	↓ 66.8% (3.01× better)
Phase B THD (%)	3.08	11.06	↓ 72.2%
Phase C THD (%)	2.95	5.10	↓ 42.2%
DC-Link Voltage Utilization	15.5% higher	Baseline	↑ 15.5%
Starting Peak Torque (Nm)	9.54	8.75	↑ 8.2%
Speed Tracking	Fast and stable	Lag and minor error	Superior
Current Waveform Symmetry	Balanced	Slight asymmetry	Superior
Peak Current Density (A/m^2)	6.04×10^6	1.27×10^7	↓ 52.6%
Max. Flux Density B_{max} (T)	1.9169	2.0537	↓ 6.7% reduction
System Efficiency η (%)	Higher	Baseline	Superior

7. Genetic-Algorithm-Optimized Speed Controller: Conventional PI vs. Fractional-Order PI

This section presents the GA-based optimization of PI and fractional-order PI speed controllers within a co-simulation framework, and evaluates their performance under FOC using SVPWM.

7.1. Co-Simulation with GA Optimization

The GA-based PI controller optimization is implemented using the MATLAB-ANSYS co-simulation system, as seen in Figure 17. In this design, the genetic algorithm tool available in Matlab works through a custom .m script, which iteratively executes the Simulink model using the sim() command. During each simulation step, the Simulink model communicates with ANSYS Maxwell to acquire high-fidelity electromagnetic responses from the OPMSM.

The entire optimization process can be summarized as follows:

1. The GA generates a random solution vector $[K_p, K_i]$, which represents the PI controller parameters.
2. These parameters are transmitted to the .m script, which launches the Simulink model using the sim() function.
3. At each time step, ANSYS solves the motor's electromagnetic model and delivers the phase currents $[i_a, i_b, i_c]$, mechanical angle ϕ, and rotor speed ω.

4. The speed error is defined. The PI controller processes the error signal and uses its output as the I_q^* reference for the field-oriented control (FOC) scheme. The FOC block generates the corresponding ABC* voltage references for the inverter.
5. In parallel, the error signal is utilized to assess control performance using integral performance indices (IAE, ITAE, ISE, and ITSE). Simulink implements these indices with weighting blocks and $1/s$ integrators.
6. After each simulation, the computed performance indices are exported over the sys_out port and returned to the GA as the fitness value. Fitness function is as follows:

$$\begin{aligned} \text{Fitness} = {} & 35\,\text{abs}(ss1.\text{RiseTime}) + 35\,\text{abs}(ss1.\text{Peak}) \\ & + 30\,\text{abs}(ss1.\text{Overshoot}) + 10\,\text{ITSE.Data}(\text{end}) \\ & + 30\,\text{ISE.Data}(\text{end}) + 40\,\text{ITAE.Data}(\text{end}) \\ & + 10\,\text{IAE.Data}(\text{end}) \end{aligned} \tag{20}$$

7. The GA examines the population, uses selection, crossover, and mutation operators, and creates a new generation. The best-performing option is chosen for reporting after the GA-based tuning process was conducted several times; the results are determined to be sufficiently consistent between cycles.

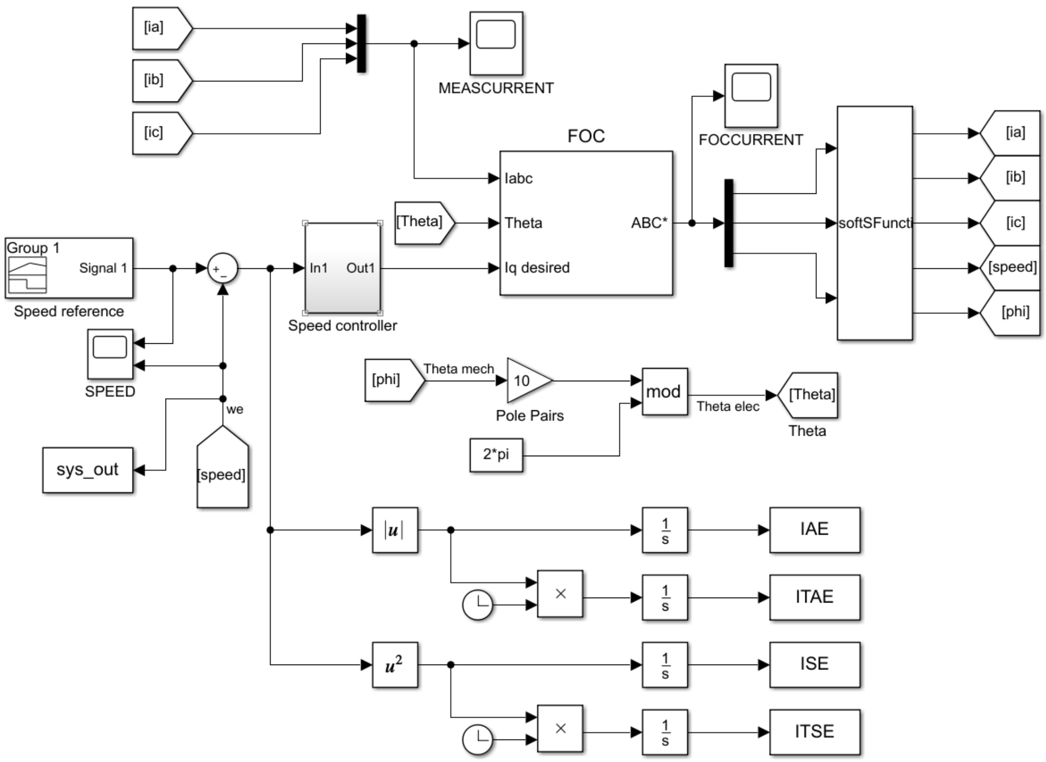

Figure 17. GA optimization for speed controller in Matlab.

In this configuration, the Simulink model functions as both a closed-loop system simulator and a fitness tester for the GA. The connection with ANSYS allows for the inclusion of electromagnetic nonlinearities in the optimization process, resulting in more precise and reliable controller tuning than traditional analytical models.

7.2. Fractional-Order PI Structure

Figure 18 displays the fractional-order PI controller's Simulink block diagram. After passing through a pre-gain block, the input signal is split into two parallel channels that represent the integral and proportional actions.

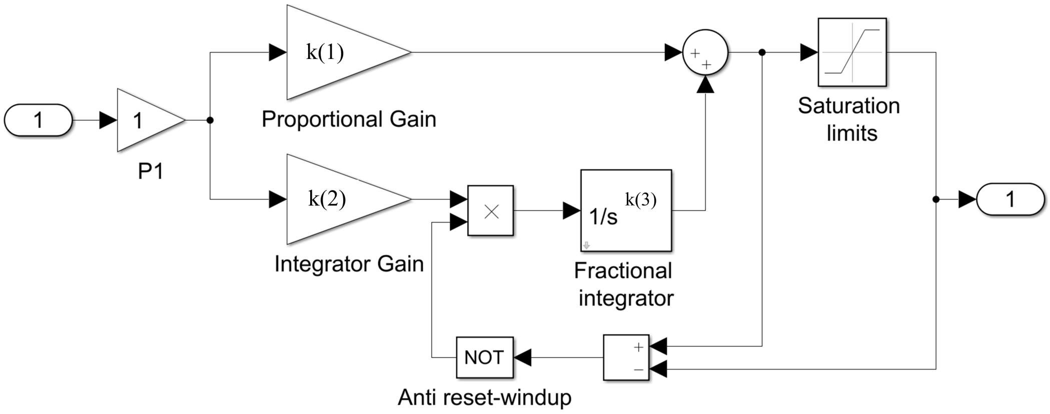

Figure 18. Simulink block diagram of the fractional-order PI.

The error signal is multiplied by the proportional gain K_p in the first path and scaled by the integral gain K_i in the second path before being applied to the fractional integrator block $1/s^{\lambda}$. In this case, λ stands for the fractional integration order, which is usually chosen between 0 and 1. It offers more tuning freedom than traditional integer-order controllers. The controller output is created by adding the outputs from the two routes and passing it through a saturation block to take actuator constraints into consideration. An anti-windup device is integrated into the control framework to prevent integrator windup under saturation situations. In this study, the parameters are defined as

$$k(1) = K_p, \quad k(2) = K_i, \quad k(3) = \lambda \tag{21}$$

and are optimized using the genetic algorithm.

Notably, the suggested FO-PI controller is a generalized version of the traditional PI controller. The controller reduces to the conventional PI structure when the fractional order is chosen as $\lambda = 1$. Thus, in this study, PI and fractional-order PI controllers are implemented and assessed inside the same architecture.

The transfer function of the controller is given by

$$C(s) = K_p + \frac{K_i}{s^{\lambda}} \tag{22}$$

7.3. Results and Discussions

The effectiveness of the suggested control strategies was evaluated using a space vector pulse width modulation (SVPWM)-based inverter with step changes in reference speed. Three different controllers were evaluated: GA-optimized fractional-order PI (FOPI), empirically tuned PI, and GA-tuned PI. Figures 19 and 20 depict the corresponding electromagnetic torque and speed responses.

As illustrated in Figure 19, all controllers correctly follow the reference speed and achieve comparable steady-state performance. However, considerable variations are detected in transitory situations. The empirically tuned PI controller experiences severe overshoot and oscillations while increasing speed by 0.5 s before reaching steady state. The GA-tuned PI controller increases performance by minimizing overshoot and partially dampening oscillations. In comparison, the GA-optimized FOPI controller has better dynamic performance, as seen by lower overshoot, faster settling time, and smoother transition. A similar trend is noticed with the step decrease in speed at around 1.0 s. The empirically tuned PI controller exhibits a distinct undershoot and oscillatory behavior, but the GA-tuned PI controller generates a more regulated response. The FOPI controller once again outperforms the others, ensuring a smoother transition with less undershoot and faster stabilization. These distinctions are underlined in the zoomed-in sections of Figure 19, where transient features are plainly visible.

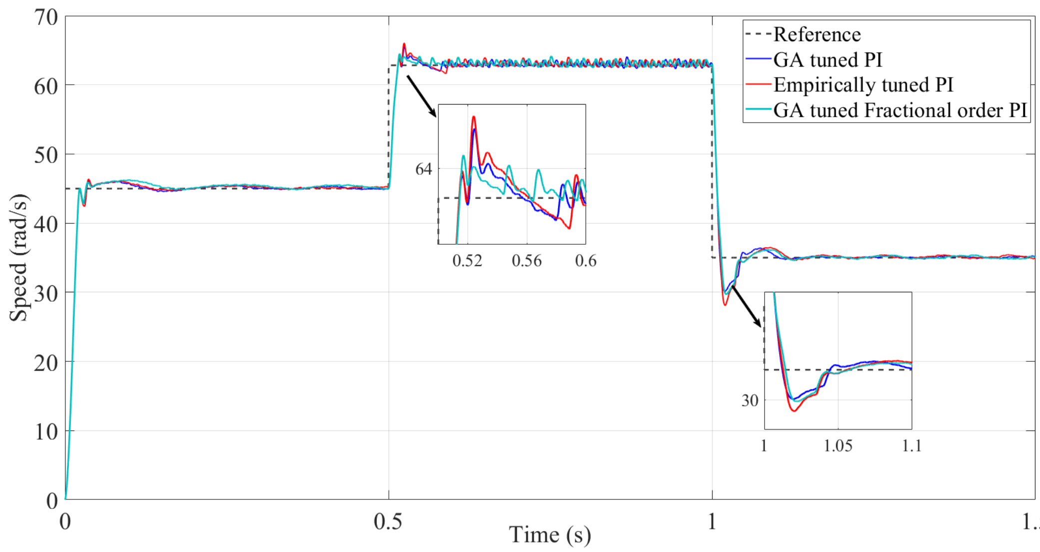

Figure 19. Comparison of speed responses for GA-tuned PI, empirically tuned PI, and GA-tuned fractional-order PI using SVPWM.

Figure 20 illustrates the system's dynamic properties through its electromagnetic torque responses. All controllers produce a large torque peak during startup due to the acceleration requirement. However, the empirically tuned PI controller has more dramatic fluctuations than the GA-based controllers. The empirically tuned PI controller exhibits larger torque ripple at speed transitions about 0.5 s, while the GA-tuned PI controller minimizes oscillations. The GA-optimized FOPI controller has the lowest torque ripple, suggesting improved current regulation and system stability. All controllers exhibit a negative torque peak during the 1.0 s speed reduction, as expected. However, the size and oscillatory form of this reaction vary. The empirically tuned PI controller has the most abrupt and oscillatory behavior, whereas the FOPI controller has a smoother torque response and faster damping characteristics.

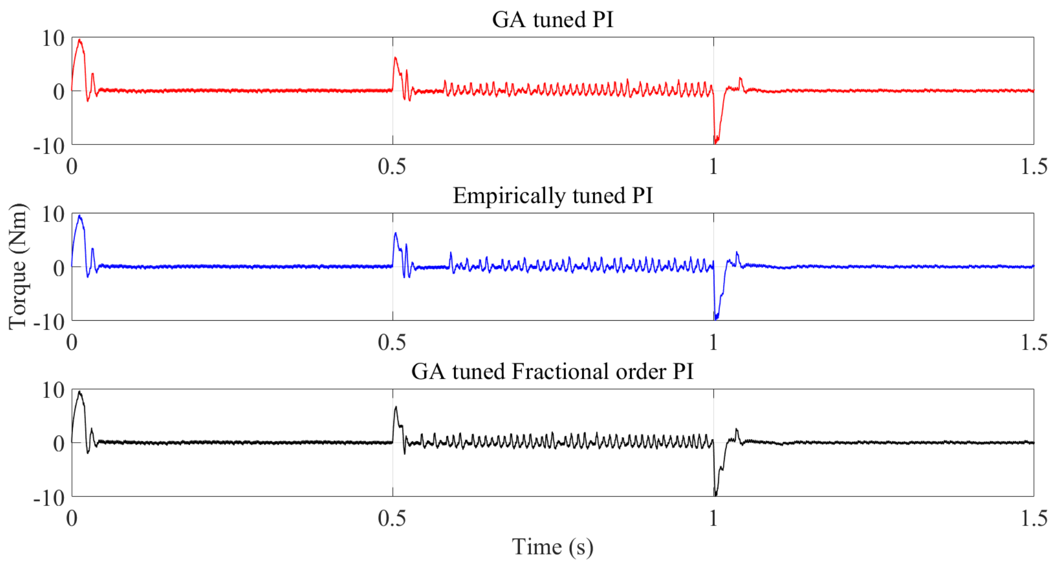

Figure 20. Comparison of electromagnetic torque responses for GA-tuned PI, empirically tuned PI, and GA-tuned fractional-order PI using SVPWM.

Overall, the results show that GA optimization improves the performance of traditional PI controllers, whereas fractional-order control adds a degree of freedom that greatly improves transient response and minimizes torque ripple. As a result, the GA-optimized FOPI controller provides higher dynamic performance while maintaining steady-state accuracy, making it a suitable approach for high-performance OPMSM drive applications. The results are summarized in Table 6.

Table 6. Dynamic speed response comparison of different control algorithms.

Reference Change	Controller	Rise/Fall Time (s)	Max. Overshoot (%)	MSE
45 $\rightarrow$ 62.832 rad/s	GA-tuned fractional-order PI	0.01010	9.43	3.3978
	GA-tuned PI	0.01010	15.27	3.3315
	Empirically tuned PI	0.01015	18.05	3.3131
62.832 $\rightarrow$ 35 rad/s	GA-tuned fractional-order PI	0.01020	19.00	8.1491
	GA-tuned PI	0.00845	17.65	7.8612
	Empirically tuned PI	0.00855	24.86	8.8004

8. Conclusions

This study presented a complete co-simulation-based comparative study of SPWM and SVPWM modulation approaches within a closed-loop FOC framework for an outer rotor permanent magnet synchronous motor (OPMSM), using a high-fidelity platform that included ANSYS Maxwell, Twin Builder, and MATLAB/Simulink environments. The finite element method was used to accurately capture the motor's nonlinear electromagnetic behavior, and the power electronics and control algorithms were implemented in a bidirectional co-simulation architecture, allowing for more realistic and reliable performance evaluation than conventional analytical models. The comparing results clearly showed that SVPWM outperformed SPWM on all performance criteria. SVPWM's zero-sequence voltage injection increased DC-link voltage usage by 15.5%, resulting in higher maximum torque and speed capabilities. The current total harmonic distortion was reduced by roughly 66.8%, 72.2%, and 42.2% in phases A, B, and C, respectively, resulting in nearly three times lower THD compared to SPWM. This significant reduction in harmonic content helps to reduce copper losses and thermal stress on the motor windings, extending the machine's operating lifetime. SVPWM reduced steady-state torque ripple by over 85%, improved id = 0 regulation precision, and achieved exact reference speed tracking with minimal steady-state errors. In contrast, SPWM had substantial limits under transient settings, failing to reach the reference speed due to inherent DC-link voltage constraints and causing undesired cross-coupling effects between the d- and q-axis current components. In terms of speed controller optimization, the GA-based tuning methodology significantly improved the transient and steady-state performance of both conventional PI and fractional-order PI controllers. The GA-optimized fractional-order PI controller consistently beat both the empirically tuned PI and the GA-tuned classical PI controllers in terms of overshoot reduction, settling time, and torque ripple suppression. The fractional integration order λ allows for more precise and robust controller tuning, resulting in smoother dynamic transitions and greater system stability under different operating situations.

By precisely capturing electromagnetic nonlinearities that are often missed in traditional models, the suggested co-simulation framework offers a high-fidelity virtual prototyping platform for electric drive systems. The framework provides quick speed response and minimal acoustic emissions in OPMSM drives by combining SVPWM with GA-optimized FOPI control, which makes it ideal for demanding applications in robotics, aerospace, and electric cars. This framework's Maxwell-based finite element analysis allows for accurate representation of magnetic saturation and core loss effects, but because it is a virtual prototype, it operates under idealized conditions and ignores real-world issues like sensor noise, inverter dead-time, and thermal parameter drift. Future study will include thermal variation sweeps, with experimental validation carried out via hardware-in-the-loop testing.

Author Contributions: Conceptualization, B.Ç.; methodology, B.Ç.; software, B.Ç.; validation, B.Ç.; formal analysis, B.Ç.; investigation, B.Ç.; data curation, B.Ç.; writing—original draft preparation, B.Ç.; visualization, B.Ç.; supervision, T.K.; writing—review and editing, T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors gratefully acknowledge the anonymous reviewers for their time, expertise, and invaluable feedback. Their insightful critiques and constructive suggestions were indispensable to achieving significant enhancements in the quality and rigor of this manuscript, and they are deeply appreciated.

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

Abbreviations

The following abbreviations are used in this manuscript:

PMSM	Permanent Magnet Synchronous Motor
OPMSM	Outer Rotor Permanent Magnet Synchronous Motor
PMSG	Permanent Magnet Synchronous Generator
SRM	Switched Reluctance Motor
FOC	Field-Oriented Control
DFOC	Direct Field-Oriented Control
MPC	Model Predictive Control
OPP	Optimal Pulse Pattern
MTPA	Maximum Torque Per Ampere
PWM	Pulse Width Modulation
SPWM	Sinusoidal Pulse Width Modulation
SVPWM	Space Vector Pulse Width Modulation
SVM	Space Vector Modulation
VSI	Voltage Source Inverter
PI	Proportional-Integral
FOPI	Fractional-Order Proportional-Integral Controller
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
ABC	Artificial Bee Colony
THD	Total Harmonic Distortion
FEM	Finite Element Method
ISE	Integral of Squared Error
IAE	Integral of Absolute Error
ITSE	Integral of Time-weighted Squared Error
ITAE	Integral of Time-weighted Absolute Error
EV	Electric Vehicle
UAV	Unmanned Aerial Vehicle
HTS	High-Temperature Superconducting
ECE	Equivalent Circuit Extraction
CPS-SVPWM	Carrier Phase Shifted Space Vector Pulse Width Modulation
DC	Direct Current
AC	Alternating Current

IGBT	Insulated Gate Bipolar Transistor

References

1. Gutfleisch, O.; Willard, M.A.; Brück, E.; Chen, C.H.; Sankar, S.G.; Liu, J.P. Magnetic Materials and Devices for the 21st Century: Stronger, Lighter, and More Energy Efficient. *Adv. Mater.* **2011**, *23*, 821–842. [CrossRef] [PubMed]
2. Dems, M.; Komeza, K.; Szulakowski, J.; Kubiak, W. Modeling of core loss for non-oriented electrical steel. *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.* **2019**, *38*, 1874–1884. [CrossRef]
3. Zhang, J.; Yin, D. A Design of Interior Permanent Magnet Synchronous Motor for Electric Vehicles. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *486*, 012130. [CrossRef]
4. Yamazaki, K.; Togashi, Y.; Ikemi, T.; Ohki, S.; Mizokami, R. Reduction of Inverter Carrier Harmonic Losses in Interior Permanent Magnet Synchronous Motors by Optimizing Rotor and Stator Shapes. *IEEE Trans. Ind. Appl.* **2018**, *55*, 306–315. [CrossRef]
5. Nicola, M.; Nicola, C.I. Tuning of PI Speed Controller for PMSM Control System Using Computational Intelligence. In Proceedings of the 2021 21st International Symposium on Power Electronics (Ee), Novi Sad, Serbia, 27–30 October 2021; pp. 1–6. [CrossRef]
6. Çelik, B. SMSM Vektör Kontrolünde Akım Kontrolünün Geleneksel PI Kontrolör ve Pekiştirmeli Öğrenme Algoritması ile Uygulanması ve Karşılaştırılması. *Fırat Üniv. Mühendis. Bilim. Derg.* **2022**, *1*, 137–142.
7. Soyaslan, M.; Avsar, Y.; Fenercioglu, A.; Eldogan, O. Cogging Torque Reduction in External Rotor PM Synchronous Motors by Optimum Pole Embrace. In Proceedings of the 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 11–13 October 2019; pp. 1–4. [CrossRef]
8. Sabarad, J.; Kulkarni, G. Comparative analysis of SVPWM and SPWM techniques for multilevel inverter. In Proceedings of the 2015 International Conference on Power and Advanced Control Engineering (ICPACE), Bengaluru, India, 12–14 August 2015; pp. 232–237. [CrossRef]
9. Harasis, S.; Khan, I.; Massoud, A. Investigating the Impact of PWM Schemes on Transient Response of Power Converters: Comparative Study Between SPWM, SVPWM, and THIPWM. In Proceedings of the 2024 IEEE 8th Energy Conference (ENERGYCON), Doha, Qatar, 4–7 March 2024; pp. 1–6. [CrossRef]
10. Swierczynski, D.; Kazmierkowski, M. Direct torque control of permanent magnet synchronous motor (PMSM) using space vector modulation (DTC-SVM)-simulation and experimental results. In Proceedings of the IEEE 2002 28th Annual Conference of the Industrial Electronics Society, IECON 02, Seville, Spain, 5–8 November 2002; Volume 1, pp. 751–755. [CrossRef]
11. Çelik, B.; Kara, T. A review of fault-tolerant control methods for permanent magnet synchronous motors with demagnetization fault. *Int. J. Dyn. Control* **2025**, *13*, 277. [CrossRef]
12. Rihar, A.; Zajec, P.; Vončina, D. Cosimulation of Ansys Simplorer and MATLAB/Simulink. In Proceedings of the 2017 19th International Conference on Electrical Drives and Power Electronics (EDPE), Dubrovnik, Croatia, 4–6 October 2017.
13. Quintal-Palomo, R. Modelling and Co-Simulation of Small Wind Turbine with Permanent Magnet Synchronous Generator. *Prz. Elektrotech.* **2019**, *1*, 210–215. [CrossRef]
14. Quintal-Palomo, R.E.; Gwozdziewicz, M.; Dybkowski, M. Modelling and co-simulation of a permanent magnet synchronous generator. *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.* **2019**, *38*, 1904–1917. [CrossRef]
15. Quintal-Palomo, R.E. Analysis of Current Signals in a Partially Demagnetized Vector Controlled Interior Permanent Magnet Generator. *Power Electron. Drives* **2019**, *4*, 179–190. [CrossRef]
16. Dos Santos Moraes, T.J.; Semail, E.; Nguyen, N.K. Demagnetization analysis of an open-end windings 5-phase PMSM under transistor short-circuit fault. In *Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019*; IEEE: New York, NY, USA, 2019; pp. 6206–6211. [CrossRef]
17. Sun, K.; Torresan, H.; Summers, T. Modelling and Control of Multi-phase PMSM in ANSYS and PLECS. In *Proceedings of the 2021 31st Australasian Universities Power Engineering Conference (AUPEC), Perth, Australia, 26–30 September 2021*; IEEE: New York, NY, USA, 2019; pp. 1–6. [CrossRef]
18. A, A.; A C, B.K. Optimal solution for PMSM rotor magnet demagnetization based on temperature estimation for EV application. In *Proceedings of the 2021 International Conference on Communication, Control and Information Sciences (ICCISc), Montreal, QC, Canada, 14–18 June 2021*; IEEE: New York, NY, USA, 2021; pp. 1–6. [CrossRef]
19. Quintal-Palomo, R.E.; Flota-Banuelos, M.; Bassam, A.; Peon-Escalante, R.; Penunuri, F.; Dybkowski, M. Post-Fault Demagnetization of a PMSG Under Field Oriented Control Operation. *IEEE Access* **2021**, *9*, 53838–53848. [CrossRef]
20. Sun, D.; Çira, F. Sonlu Elemanlar Yöntemi ile Asenkron Motorun PWM Arızasının Analizi. *Harran Üniv. Mühendis. Derg.* **2022**, *7*, 217–228. [CrossRef]
21. Winter, J.; Zipprich, R.; Tekir, B.; Ziegler, M. Investigation of the Influence Caused by Coupling the Winding Systems in a Dual-Stator PMSM Using MATLAB/Maxwell CO-SIM. In Proceedings of the PCIM Europe 2023: International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, Nuremberg, Germany, 9–11 May 2023; pp. 1–8. [CrossRef]

22. Zhao, S.; Tian, D.; Ni, Y.; Wang, Y.; Chen, J.; Xu, J. The Co-simulation of PMSM Control System Based on the Equivalent Circuit Extraction Model. In *Proceedings of the 2023 IEEE/ACIS 23rd International Conference on Computer and Information Science (ICIS), Wuxi, China, 23–24 June 2023*; IEEE: New York, NY, USA, 2023; pp. 115–119. [CrossRef]
23. Hölsch, L.; Brosch, A.; Steckel, R.; Braun, T.; Wendel, S.; Böcker, J.; Wallscheid, O. Insights and Challenges of Co-Simulation-Based Optimal Pulse Pattern Evaluation for Electric Drives. *IEEE Trans. Energy Convers.* **2024**, *39*, 2094–2105. [CrossRef]
24. Tekir, B.; Zipprich, R.; Winter, J.; Ziegler, M. Reduction of PWM Harmonics with Carrier Phase Shifting in a Dual-Stator PMSM with Magnetic Coupled Windings. In Proceedings of the PCIM Europe 2024: International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, Nuremberg, Germany, 11–13 June 2024; pp. 672–680. [CrossRef]
25. Lind, E.; Eriksson, S. Co-simulation model combining dynamic control and FEM for evaluation of PMSM drive cycle performance. *Results Eng.* **2025**, *27*, 106149. [CrossRef]
26. Shi, B.; Li, Y.; Ramones, A.I.; Fu, T.; Kallur-Krishnamoorthy, R.; Bo, J.; Andert, J.L. A Unified Model and Flux Observer for Dual Three-Phase Electric Machines. *IEEE Trans. Transp. Electrif.* **2025**, *11*, 12210–12221. [CrossRef]
27. H, V.; R, A.; M, D.; B, A. Integrated Modeling and Control of an 8/6 SRM Drive using ANSYS and MATLAB Co-Simulation. In *Proceedings of the 2025 International Conference on Modern Sustainable Systems (CMSS), Shah Alam, Malaysia, 12–14 August 2025*; IEEE: New York, NY, USA, 2025; pp. 466–473. [CrossRef]
28. Ding, Z.; Bi, J.; Lin, W.; Zhao, R.; Hu, C.; Chi, M.; Deng, Z. Design and Analysis of a Novel In-Wheel Machine-Based Powertrain for HTS Maglev Systems. *IEEE Trans. Transp. Electrif.* 2026, *early access*. [CrossRef]
29. Jussila, H.; Salminen, P.; Niemela, M.; Pyrhonen, J. Guidelines for Designing Concentrated Winding Fractional Slot Permanent Magnet Machines. In Proceedings of the 2007 International Conference on Power Engineering, Energy and Electrical Drives, Setubal, Portugal, 12–14 April 2007; pp. 191–194. [CrossRef]
30. Pandya, H.M.; Pathak, A.R.; Budhrani, A.H.; Adeshara, A.S. Implementation and Analysis of SPWM and SVPWM Techniques Using STM32F407 Microcontroller. In Proceedings of the 2024 International Conference on Computer, Electronics, Electrical Engineering & Their Applications (IC2E3), Bhopal, India, 24–25 February 2024; pp. 1–5. [CrossRef]
31. Chethan, G.N.; Kodeeswara kumaran, G. Performance Analysis of PMSM Drive with SpaceVector PWM and Sinusoidal PWM fed VSI. In Proceedings of the 2019 International Conference on Power Electronics Applications and Technology in Present Energy Scenario (PETPES), Mangalore, India, 29–31 August 2019; pp. 1–6. [CrossRef]
32. Sanyasi Naidu, A.; Samanta, S. Analysis of IPMSM Motor by Using SPWM and SVPWM Based Vector Controller with MTPA Control Strategy. In Proceedings of the 2022 IEEE 10th Power India International Conference (PIICON), New Delhi, India, 25–27 November 2022; pp. 1–5. [CrossRef]
33. Prakash, N.; Jacob, J.; Reshmi, V. Comparison of DVR performance with Sinusoidal and Space Vector PWM techniques. In Proceedings of the 2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), Koovappally, India, 24–26 July 2014; pp. 1–6. [CrossRef]
34. Turksoy, O.; Yilmaz, U.; Tan, A.; Teke, A. A Comparison Study of Sinusoidal PWM and Space Vector PWM Techniques for Voltage Source Inverter. *Nat. Eng. Sci.* **2017**, *2*, 73–84. [CrossRef]
35. Wang, Y.; Dong, L.; Liao, X. Robust stabilizing regions of fractional-order PI controllers for permanent magnet synchronous motor control system. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; pp. 5065–5070. [CrossRef]
36. Dogruer, T.; Can, M.S. Design and robustness analysis of fuzzy PID controller for automatic voltage regulator system using genetic algorithm. *Trans. Inst. Meas. Control* **2022**, *44*, 1862–1873. [CrossRef]
37. Dogruer, T.; Tan, N. Design of PI Controller using Optimization Method in Fractional Order Control Systems. *IFAC-PapersOnLine* **2018**, *51*, 841–846. [CrossRef]
38. Hong, Y.; Wang, W. PID Speed Control of PMSM Based on GA-PSO Algorithm. In Proceedings of the 2025 Third International Conference on Cyber-Energy Systems and Intelligent Energy (ICCSIE), Shenyang, China, 3–5 October 2025; pp. 1–5. [CrossRef]
39. Jawad, H.; Mhawi, E.; Abdullah, H. Performance Improvement of PMSM Using PID and GA-PID Controllers. *Int. J. Electr. Electron. Res.* **2025**, *13*, 186–192. [CrossRef]
40. Ortatepe, Z. Genetic Algorithm based PID Tuning Software Design and Implementation for a DC Motor Control System. *Gazi Univ. J. Sci. Part A Eng. Innov.* **2023**, *10*, 286–300. [CrossRef]
41. Chahbi, A.; Yessef, M.; Amharech, A.; El-Barbary, Z.; Mohammad Irshad, S.; Ameziane, H. Intelligent GA and PSO-Tuned Controllers for High-Performance Direct Field-Oriented Control in DFIG-Based WECS. *IEEE Access* **2026**, *14*, 23866–23893. [CrossRef]