February 2024
PhD Dissertation

Nonlinear Control and AI-Driven Strategies for Enhancing Performance in Permanent Magnet Synchronous Motor Drives

Graduate School of Chosun University

Department of Electrical Engineering

Muhammad Usama

Nonlinear Control and AI-Driven Strategies for Enhancing Performance in Permanent Magnet Synchronous Motor Drives

비선형 제어 및 AI 기법을 활용한 영구자석 동기 전동기 구동 성능 향상

February 23, 2024

Graduate School of Chosun University

Department of Electrical Engineering

Muhammad Usama

Nonlinear Control and AI-Driven Strategies for Enhancing Performance in Permanent Magnet Synchronous Motor Drives

Advisor: Jaehong Kim

A dissertation submitted in partial fulfillment of the requirements
for the Degree of Doctor of Engineering

October 2023

Graduate School of Chosun University

Department of Electrical Engineering

Muhammad Usama

우사마 무함마드 박사학위논문을 인준함

위원장 김용재 (인)

위원 최연옥 (인)

위원 백건 (인)

위원 남광희 (인)

위원 김재홍 (인)

2024 년 01 월

조선대학교 대학원

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

SPMSM	Surface-Mount Permanent Magnet Synchronous Motor
IPMSM	Interior Permanent Magnet Synchronous Motor
PI	Proportional Integral
IP	Integral Proportional
ITAE	Integral Time Absolute Error
ITSE	Integral Time squared Error
ISE	Integral Square Error
IAE	Integral Absolute Error
VV	Voltage Vector
MRAS	Model Reference Adaptive System
FCS-MPCC	Finite Control Set-Model Predictive Current Control
MPTC	Model Predictive Torque Control
SMC	Sliding Mode Controller
EMF	Electromotive Force
DTC	Direct Torque Controller
FOC	Field Oriented Control
VSI	Voltage Source Inverter
MTPA	Maximum Torque per Armature
AI	Artificial Intelligence
ML	Machine Learning
NN	Neural Network
DSR	Deep Symbolic Regression
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
DSR	Deep Symbolic Regression

PSO	Particle Swarm Optimization
CS	Cuckoo Search
FNN	Feed-Forward Neural Network
AC	Actor-Critic
THD	Total Harmonic Distortion
TRF	Torque Ripple Factor
SRF	Speed Ripple Factor
CNN	Convolutional Neural Network
DSP	Digital Signal Processing
FRAM	Ferroelectric Random Access Memory

List of Figures

List of Tables

ABSTRACT

Nonlinear Control and AI-Driven Strategies for Enhancing Performance in Permanent Magnet Synchronous Motor Drives

Muhammad Usama
Advisor: Prof. Jaehong Kim, Ph.D.
Department of Electrical Engineering
Graduate School of Chosun University

This thesis presents adequate control strategies for PMSM drive control. Different control schemes have been proposed over the years to enhanced control performance of motor drive system. In past the most of the control strategies are based on traditional linear control scheme that is proportional-integral(PI) control for cascaded control design of motor drive. PI controls are still employed in industrial application due to there simplicity and easy to implement. But due to nonlinear nature of power electronic systems with multiple inputs, and outputs the performance of PI can easily be degraded under disturbance or parametric variations. Moreover, PI controllers are suitable for linear systems with unconstrained control problems. The controller performance is highly dependent on plant model, where they are tuned for excellent performance within narrow working operating range, thus possessing excellent closed-loop performance is still challenging. To overcome this challenge the meta-heuristic approaches are implemented in this work to tune control parameters without need of the plant model. However, over the wide operating range the dynamic performance is still poses practical challenges.

In recent years, there is remarkable advancement in control strategies for Permanent Magnet Synchronous Machine(PMSM). Nonlinear control designs

are implemented to enhanced the closed-loop control performance and provide robustness in transient and steady-state working operating conditions. Also, nowadays AI-based control strategies are emerging at fast-pace for the development of intelligent control design. Moreover, these control design are capable of handling system nonlinearities and constraints with effective manner. Furthermore, these intelligent control design doesn't require prior knowledge of plant model as well as computationally less expensive then some nonlinear control strategies because the training of the network is done offline and the trained network is implemented online.

This thesis is divided into two parts. The key motivation behind the design of control strategies and the implemented control strategies in the literature are presented in the first part. Furthermore, the modeling of mathematical model of the motor drive implemented in this thesis is presented in detail. The second part presents the design and application of advanced control strategies for excellent dynamic performance of motor drives, and is divided into three chapters. The first chapter demonstrate the optimization algorithm to fine tune the linear controller gains for efficient dynamic performance without the need of plant model. The model independent tuning approach give optimized linear control superiority over traditional linear control. The second chapter is devoted to the design of advanced nonlinear control for the application motor drives. The chapter will demonstrate the reaching law sliding mode control for enhanced dynamic performance at low speed operation with effective compensation technique. This provided approach ensure excellent speed tracking performance and remain robust in presence of uncertainties and mitigate the undesirable high-frequency chattering resulting from the rapid mode switching within the control system, which aims to maintain the state on a predefined sliding surface and consequently causes abrupt and oscillatory fluctuations in the control input. The third chapter address the application of Artificial intelligence(AI) for motor drive system. Some state of

the art control design are proposed that can replace Linear or non-linear control design and give excellent performance without prior expert knowledge. Finally, this thesis thoroughly evaluates motor drive performance using a combination of experimental and simulation-based tests. The presented test results help improve motor drive systems by demonstrating exceptional performance features.

한 글 요 약

비선형 제어 및 AI 기법을 활용한 영구자석 동기 전동기 구동 성능 향상

우사마 무함마드
지도 교수: 김재홍
전기 공학과
대학원, 조선대학교

이 논문에서는 영구자석 동기기(permanent magnet synchronous machine, PMSM)의 제어 특성을 향상시키기 위하여, 비선형 제어 및 인공지능(artificial intelligence, AI) 기법을 활용한 제어전략을 제시한다. 현재까지 영구자석 동기 모터의 운전을 위해서는 대부분 비례-적분(PI) 제어기를 활용한 선형 직렬 제어 방식을 사용해왔다. PI 제어는 단순하고 구현하기 쉽기 때문에 여전히 산업용 애플리케이션에서 사용되고 있지만 구동 인버터의 비선형성으로 인해 외란 또는 파라미터 변화에 따라 PI 제어기의 성능이 쉽게 저하되는 문제가 있다. 또한 PI 제어기는 포화특성이 없는 선형 시스템에 적합하다. 제어기의 성능은 플랜트 모델에 의존적이며 좁은 범위 내에서 우수한 성능을 위해 조정되므로 넓은 운전영역에서 훌륭한 페루프 성능을 유지하는 것은 어려운 과제이다. 이러한 문제들을 극복하기 위해 이 연구에서는 meta-heuristic approach를 구현하여 정확한 플랜트 모델 없이도 제어 파라미터를 조정할 수 있는 방법을 제시한다.

지난 몇 년 동안 PMSM 제어전략에 괄목할 만한 발전이 있었다. 비선형 제어 설계를 통하여 페루프 제어 성능을 향상시키고 과도상태 및 정상상태 동작 조건에서 강인성을 높이려는 노력들이 이어져왔다. 또한 최근에는 지능형 제어설계 개발을 위해 AI 기반 제어전략들이 빠른 속도로 소개되고 있다. 이러한 제어설계 기법들은 시스템의 비선형성 및 제약 조건을 하에서 시스템을 효과

적으로 구동할 수 있게 해준다. 뿐만 아니라 지능형 제어 설계는 플랜트 모델에 대한 사전 지식이 필요하지 않으며 네트워크의 학습이 오프라인에서 이루어지고 학습된 네트워크가 온라인으로 구현되기 때문에 일부 비선형 제어전략에 비해 실시간 계산에 대한 부담이 덜하다.

이 논문은 두 부분으로 구성된다. 첫 번째는 제어전략 설계의 주요 동기와 문헌에서 구현된 제어전략들을 소개하고 PMSM 드라이브의 수학적 모델링에 대해 자세히 설명한다. 두번째는 PMSM 드라이브의 우수한 동적 성능을 위한 고급 제어전략의 설계 및 적용 방법을 제시하며, 총 3개의 장으로 구성된다. 첫 번째 장에서는 플랜트 모델 없이 선형 제어기 이득을 미세 조정하는 최적화 알고리즘을 제시한다. 이 model independent tuning approach는 기존 선형제어 튜닝방식보다 우월한 동적 성능을 제공한다. 두 번째 장에서는 PMSM 드라이브를 위한 고급 비선형 제어 기법을 제안하며, 제안된 제어전략은 reaching law approach 기반의 sliding mode control을 이용한다. 세 번째 장에서는 PMSM 구동 시스템에 대한 인공지능(AI)의 적용에 대해 다룬다. 선형 또는 비선형 제어 설계를 대체하고 사전 전문 지식 없이도 우수한 성능을 제공할 수 있는 몇 가지 최첨단 제어 설계 기법을 제안한다. 마지막으로 실험 및 시뮬레이션을 통하여 제안된 기법들의 적합성을 평가하며, 제시된 테스트 결과들을 통하여 제안된 제어 설계 기법이 PMSM 구동에 있어서 향상된 동적 특성을 보여줌을 증명한다.

I. INTRODUCTION

As the importance of energy efficiency and sustainability has increased across several industries, there has been an increase in the need for efficient electric motor drives. The inherent benefits of permanent magnet synchronous motors (PMSMs), such as their high power density, small size, and excellent controllability, have made them an attractive choice. PMSMs are used in various industries, including industrial automation, water pumps, clean energy systems, CNC machines, electric vehicle drive systems, etc. However, it is crucial to develop controls that maximize the dynamic performance of PMSMs in order to realize their potential fully.

The dynamic performance of PMSMs refers to their ability to precisely and quickly react to changes in operating conditions, such as load perturbation and reference control inputs. The dynamic performance of the motor drive can be affected by system Non-linearities, parameter uncertainties, and external disturbances. Ineffective control design may lead to problems such as poor reference tracking performance, higher total harmonic distortion and torque ripple, steady-state error, and bounded disturbance rejection, which can affect the overall dynamic system performance. Therefore, this thesis investigates and develops control strategies that address these challenges and enhance the dynamic performance and reliability of PMSMs for industrial applications.

A. Optimization Strategies

The closed-loop control performance of the control system highly depends on how well the controller parameters are tuned. Excellent disturbance rejection, stability, durability, robustness, and good tracking response without being easily impacted by noise are the main characteristics of a well-tuned controller [1]. Classical tuning technique does not provide the required control performance compared to optimization control tuning technique, which are independent of

plant model and can aid in solving problems that include multiple objectives, nonlinear constraints, and nonlinear dynamic properties [2]. Optimization-based tuning techniques, such as metaheuristic algorithms, are the best replacement for traditional tuning techniques as they can handle significant complex problems with reduced computational complexity. Moreover, simulation-based optimization poses three major concerns: an algorithm's effectiveness, the statistical simulator's accuracy and precision, and adequately choosing the appropriate methodologies for the specific problem. According to [3], several optimization algorithms are studied in the literature, but not all provide suitable solutions for all issues. Thus, this thesis will study the most effective optimization strategy with reduced computational complexity and excellent drive response.

In this thesis, the main objective of employing the optimization technique is to have optimal gains for speed control of PMSM. Nature-inspired metaheuristic algorithms such as PSO, CS, and Jaya optimization are utilized to tune the gain parameters and provide dynamic speed response. Metaheuristic algorithms provide optimal solutions by overcoming nonlinear constraints or system nonlinearity. Moreover, effectiveness in solving complex systems and good efficiency in optimizing the controller for providing excellent closed-loop performance lead to a hot topic in optimizing parameters in the control design of PMSM drives. By employing the optimization algorithm for gain tuning of the controller, it is feasible to obtain optimal gain parameters that provide reduced overshoot, less steady state error, fast rise time, and robustness against the external torque.

Additionally, the contribution of this work specifically includes:

(a) Application of metaheuristic approach for control design of PMSM

(b) Comprehensive study of the computational complexity of applied metaheuristic approach

(c) The applied metaheuristics approach is independent of the plant model; hence

accurate plant model parameters are not required for designing speed control

(d) A detailed code is provided in Appendix to aid future metaheuristic practitioners for further research studies

(e) This section is based on [4]

B. Advanced Control

A vector control strategy is widely employed in motor control and typically has a cascaded control design with internal PI current control and outer PI speed control loop [5]. The bandwidth of the inner current control loop is larger than the outer speed loop. PI controllers' performance usually degrades due to nonlinear complex systems, parametric perturbation, and external disturbance, leading to unstable response and reduced performance efficacy. To overcome these challenges, a nonlinear control structure is adopted that handles system nonlinearities and suppresses the effect of parametric variation [6], [7].

In this chapter, a nonlinear control design employing an SMC for the enhanced system's dynamic response is proposed. By employing the proposed control strategies, we can enhance the dynamic performance of PMSM at a standstill and transient conditions at low speed operation conditions, that are usually sensitive to adversarial disturbance. Moreover, the current harmonics, the torque ripple, the speed rise time, overshoot, and steady-state error are reduced in the proposed designs. In addition, the control efficiency and immunity to external disturbance and parametric variations are verified and compared with conventional control design. Thus, the main contribution of advance control design specifically includes:

(a) For the improved dynamic performance of IPMSM drives, the nonlinear cascaded control design is proposed and compared with traditional control design

(b) To enhance speed smoothness during low-speed operation of IPMSM

drives, the research offers an efficient compensating technique. The goal of this technique is to suppress the inherent torque ripple and speed oscillations that usually exist at low speeds.

(c) This section is mostly based on [8]–[10]

C. AI-Driven Control Strategies

Recently, the development of artificial intelligence-based control strategies for power electronics has marked an exceptionally fast-paced era of AI. The interest in data-driven strategies for motor drive control has gained much attention among the research community in recent years. With advancements in semiconductor technology, powerful computing platforms, and data availability, AI-based control schemes are anticipated to become increasingly popular for the high-performance control of motor drives [11].

In this thesis, state-of-the-art AI-driven control strategies are explored for the high performance of PMSM. Machine learning is the subset of AI and is categorized into three branches; (a) Supervised learning, (b) Unsupervised learning, and (c) Reinforcement learning (RL). Supervised learning will solve the multi-class classification problem at each time step to predict the optimal switching states of the inverter, resulting in reduced harmonics and excellent closed-loop performance. In addition, a novel current control employing deep symbolic regression to derive numerical expressions for the voltage reference is proposed, eliminating the need for any linear or nonlinear control, and the problem associated with parameter tuning and perturbation will be addressed.

The main contribution of this section is listed as:

(a) Model-free predictive current control utilizing a feedforward neural network is proposed with minimum features

(b) The network is trained offline, employing supervised learning to learn the responses of the FCS-MPCC under varied operating conditions

(c) The optimal current controller providing the analytical expression for the voltage references is proposed

(d) This section is mostly based on [12], [13]

II. Background Study

A. Literature Review

As the greenhouse effect becomes a severe concern, electrical machines' efficacy and reliability are more crucial than ever. The rise in need for decarbonization and the recent reduction in permanent magnet material cost and reduced greenhouse gases emission makes PMSM the best choice to be utilized in home appliances and industrial applications such as refrigerators, cooling fan washing machines, robots, electric vehicles, servo drives, high-speed train, airplane, traction, etc. [14], [15]. PMSM exhibits many advantages as compared to induction machines, such as they offer low inertia, high torque density due to high field strength, the motor size is reduced, high efficacy, no copper loss of rotor winding, and, most importantly can, incorporate reluctance torque in the field weakening range, that makes it suitable to designed for constant power speed range [16], [17].

Based on back EMFs, PM motors can be divided into two general categories. One category exhibits square or trapezoidal back EMF known as brushless DC motors, while the other exhibits sinusoidal back EMF, known as PMSM. The configurations of the magnets, cavities, and coil windings all influence how the back EMF patterns take shape [18]. For high electric machine performance, precise control and reduced ripples are essential. BLDC lacks precise shaft speed control as it employs trapezoidal back EMF and utilizes sensors feedback for commutation, which causes speed fluctuations and torque ripple, but still, if precise speed control is not essential, then BLDC can be utilized. However, due to sinusoidal back EMF, the PMSM gives precise speed control essential for high-performance applications. Excellent control efficiency, wide speed operating range, and high power density make PMSM a viable choice for industrial machinery so far [19]–[21].

Around this time, various control strategies have been proposed for optimal

control of PMSM. For efficient and stable drive response, control techniques play a critical role in regulating the position, speed, and torque of PMSM. PMSMs have been effectively controlled using conventional techniques like scalar and vector control. These techniques offer fundamental control capabilities, but they might not be able to keep up with the needs of contemporary applications, which seek improved performance [22], [23].

PI controllers are still the most favorable choice for industrial applications. These controllers are easy to implement and have simple design and reliability characteristics. However, for robust dynamic performance, the accurate parameter should be known; otherwise, it will have a sluggish response and steady-state error. Adjusting a PI controller's settings to prevent failures and provide excellent transient performance has grown increasingly challenging in recent years due to the complexity of plant operations. To overcome these issues, much work has been done to tune the controller parameters offline and online optimally. The tuning methods are classified into two categories: (a) Classical tuning strategies and (b) Optimization tuning strategies. Classical techniques are based on making assumptions about plant models and desired output. These techniques are computationally inexpensive and easy to use. However, the results are based on assumptions and hence, needed further tuning[24], [25]. In contrast, optimization techniques require the use of data modeling and optimization of decision functions to numerically tune the controller parameters [26]. The performance of controller gain tuning highly depends on decision function. However, to deal with model nonlinearities, the optimized PI control lack robust performance, and to have high performance electric machine the nonlinear control design have been proposed over the years for better working operations. Numerous control structures have been proposed employing the predictive technique, sliding mode control, fuzzy-based control, and observer-based design [27]–[29]. Low sensitivity to parametric perturbation and strong disturbance rejection are the

main features of SMC. However, chattering is a significant concern for SMC that causes low accuracy and can damage the motor. Moreover predictive control emerges as an excellent control technique that can handle nonlinear constraint and provide a robust response; however, variable switching frequency cause diffused harmonic spectrum. In addition, the performance of the predictive control strategy highly depends on model parameters [30], [31]. Thus, to deal with this problem, cascaded improved nonlinear control designs are proposed, and the robustness is verified by doing a comparative study with conventional nonlinear design [8], [10], [32].

Recently, the application of AI in power electronics gradually advanced with the emergence of machine learning which become an important research field for control, estimation, fault diagnoses, and tolerance in motor drive [33]–[35]. The data-driven approach is considered to be a viable choice in academia. Big data has become more widely accessible, which has driven the development of ever-more-powerful computer systems, and ML has effectively solved many issues that were formerly impossible to overcome. This is especially true for natural language processing using the Transformer architecture and computer vision with the CNN[36], [37]. Numerous real-world ML applications have already been implemented, including entertainment, health care, identifying fraud, AI-powered assistants, and self-driving vehicles [38], [39]. Despite technological advancements, it was also noted that commercial uses of neural networks in power electronics appear to be very few at this time. It is important to note that the primary barrier to ML applications in electric machine drives continues to be hardware constraints. The high-frequency update rate is significantly challenging for ML applications in motor drives as the update frequencies are in the range of 10KHz for electric machines, whereas just 10Hz for real-time video processing ML applications [40], [41].

B. Thesis Objective

This thesis explore and exploit different control strategies to enhance the performance of PMSM in wide operating conditions. Initially, the gain tuning method is explored to aid researchers in selecting optimization techniques in motor drive applications. The impact of the optimization technique on control performance is evaluated in-term of speed regulations and computational complexity, and parameter dependence of the optimization algorithm is presented. In addition, the nonlinear control strategies are proposed to suppress the effect of the nonlinear motor model and can handle nonlinear constraints. Furthermore, the proposed control designs are compared with the state-of-art method, and performance efficacy is verified. Finally, artificial intelligence-driven techniques are exploited for regression, classification, and prediction of control variables for performance enhancement. AI-based methods can handle complex systems and provide robust performance against parametric perturbation and independent of the plant model.

This thesis is organized as follows: The first part presents the overview presented in Chapter I. The literature review and thesis goal are mentioned in Chapter II. The mathematical model of the PMSM is described in Chapter III. The second part of the thesis presents the insight and design of control strategies. In Chapter IV, the optimization strategies are provided, whereas Chapter V explores the design of advanced nonlinear control. Furthermore, Chapter VI exploits the application of AI in motor drives. Finally, the conclusion of the thesis and an outlook for future work is given in Chapter VII.

III. Mathematical Modeling of PMSMs

Permanent Magnet eradicate the use of field exciting coils and slip ring for current conduction. Because of no field winding inside the rotor, PM motors have low inertia. The size of the motor is reduced due to high field strength. Moreover, as there is no copper loss of the secondary winding, the PM motors offers high efficacy and dynamic performance than induction machines [15].

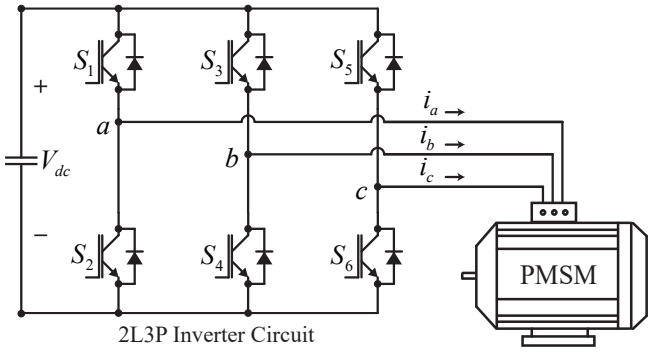


Figure 1: Three-phase two-level VSI-fed SPMSM.

For industrial applications, a VSI is the best choice because it is simple to design and is economical. The pictorial depiction of the two-level three-phase inverter connected to the PMSM is given in Figure 1.

This chapter describes the dynamical modeling and analysis of a PMSM. The three phase model of PMSM is transformed in to rotating reference frame by employing the well known park transformation. By utilizing this modification, the motor's steady-state AC signals are converted into DC values, which are simpler to compute and control.

A. Electrical Model

In a three-phase coordinate system, the voltage equation of PMSM in space vector form is derived as follows [15]

$$\bar{v}_s = R_s \bar{i}_s + \frac{d\bar{\psi}_s}{dt} \tag{1}$$

where $\bar{v}$ is the the space vector of stator voltage, $R_s\bar{i}_s$ is the voltage drop across stator resistance, and $\frac{d\bar{\psi}_s}{dt}$ is the induced voltage due to varying magnetic flux which is obtained as:

$$\bar{\psi}_s = L_s \bar{i}_s + \lambda_m e^{j\theta_e} \tag{2}$$

where λ_m is the amplitude of flux linkage of PM, L_s is the sum of mutual and leakage inductance, and θ_e is the electrical angles. Taking the derivation of (2) we get:

$$\frac{d\bar{\psi}_s}{dt} = L_s \frac{d\bar{i}_s}{dt} + \frac{d\lambda_m e^{j\theta_e}}{dt} \tag{3}$$

$$= L_s \frac{d\bar{i}_s}{dt} + j\omega_e \lambda_m e^{j\theta_e} \tag{4}$$

substituting (4) into (1), the voltage equation of PMSM in vector form is obtained as

$$\bar{v}_s = R_s \bar{i}_s + L_s \frac{d\bar{i}_s}{dt} + j\omega_e \lambda_m e^{j\theta_e} \tag{5}$$

By employing Park's transformation, dq−axis voltages are derived as:

$$v_d = r_s i_d + L_d \frac{di_{sd}}{dt} - \omega_e L_q i_{sq}. \tag{6}$$

$$v_q = r_s i_q + L_q \frac{di_{sq}}{dt} + \omega_e L_d i_{sd} + \omega_e \lambda_m. \tag{7}$$

The electromagnetic torque of the motor can be obtained from:

$$T_e = \frac{3P}{2}(\lambda_d^e i_q^e - \lambda_q^e i_d^e), \tag{8a}$$

$$\lambda_d = L_s i_d + \lambda_m. \tag{8b}$$

$$\lambda_q = L_s i_q, \tag{8c}$$

$$T_e = \frac{3P}{2}[\lambda_m i_q^e - (L_q - L_d) i_d^e i_q^e]. \tag{8d}$$

where P is the number of the poles. The value of L_d and L_q are determined by the motor geometry and arrangements of the magnets [42].

Dq saliency, which causes a difference between L_d and L_q, is a characteristic of PMSM typologies with inset-mounted or interior-radial magnets. The two inductance's can be thought of as equivalent in motors with surface-mounted magnets, where the saliency is frequently minimal and $L_d = L_q = L_s$. So the torque become the function of i_q and independent of i_d that is given as:

$$T_e = \frac{3P}{2}\lambda_m i_q^e \tag{9}$$

This means that, with the permanent magnets' flux remains constant, the machine's torque can be adjusted by controlling the q−axis current.

B. Mechanical Model

The mechanical equation of PMSM comprises of electromagnetic torque, load torque and electrical speed and given as follow:

$$T_L = T_e - B\omega_m - J\frac{d\omega_m}{dt}. \tag{10}$$

where T_L is the load torque, ω_m is the rotor mechanical speed, and J and B are the motor inertia and friction coefficient, respectively. The relationship between

electrical rotor position and shaft speed is given by $\frac{d\theta_e}{dt} = \omega_e$.

With the PMSM model established, it will be further utilized for control system design, performance analysis, and simulation , and experimental studies to optimize motor performance and efficiency.

C. Modulation Scheme

The selection of an appropriate modulation scheme is a crucial aspect in the design and implementation of three-phase inverters for PMSM. The selection of an appropriate modulation scheme directly related with the efficiency, power quality, and dynamic response of the drive. In recent years, there has been a growing interest in the development of advanced modulation schemes for industrial applications to improve efficiency and reduce the harmonic distortion of the output voltage. Among these modulation schemes, SVPWM have gained significant attention due to their superior performance characteristics.

In this section we aim to provide comprehensive performance analysis of SVPWM and Simplified SVPWM. The research aims to provide insights into the selection of an appropriate modulation scheme for industrial applications and to help engineers and researchers to design and develop more efficient and reliable power electronics and motor drive systems.

1. SVPWM

Space-vector modulation has emerged as one of the most significant PWM techniques for industrial applications today with advancements in microprocessors. Space vector modulation stands out for its simple digital implementation and broad linear modulation range for output line-to-line voltages [43]. The graphical illusion of a voltage vector corresponding to eight available logic states of VSI is shown in Figure 2. It contains two basic vectors (V_0, V_7),

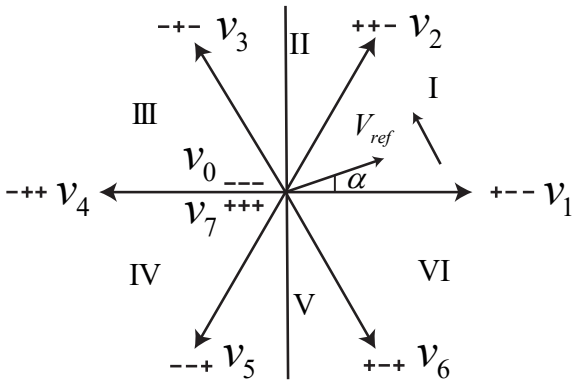


Figure 2: Voltage vectors produced by a two-level, three-phase voltage source inverter.

also known as zero voltage vectors, having an amplitude of zero, whereas six active voltage vectors $(V_1, V_2, ...V_6)$ have equal amplitude. Vector magnitudes are presented with respect to the inverter′s dc supply voltage as given in Table 2.

Table 2: Amplitude of output phase and line voltages at each sector

Sector n	Switching Vectors			Phase Voltages			Line-to-Line Voltage			Voltage Vector
	a	b	c	V_{an}	V_{bn}	V_{cn}	V_{ab}	V_{bc}	V_{ca}	V_n
1	1	0	0	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	V_{dc}	0	$-V_{dc}$	$\frac{2}{3}V_{dc}\angle 0°$
2	1	1	0	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	0	V_{dc}	$-V_{dc}$	$\frac{2}{3}V_{dc}\angle 60°$
3	0	1	0	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-V_{dc}$	V_{dc}	0	$\frac{2}{3}V_{dc}\angle 120°$
4	0	1	1	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-V_{dc}$	0	V_{dc}	$\frac{2}{3}V_{dc}\angle 180°$
5	0	0	1	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	0	$-V_{dc}$	V_{dc}	$\frac{2}{3}V_{dc}\angle 240°$
6	1	0	1	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	V_{dc}	$-V_{dc}$	0	$\frac{2}{3}V_{dc}\angle 300°$

The basic steps to implement SVPWM for nonlinear loads are given as follows:

1) To generate the SVPWM switching signals, the generalized reference voltage in the stationary reference frame is obtained using:

$$\begin{aligned} V_\alpha &= V_{ref} * \sin(\omega t) \\ V_\beta &= V_{ref} * \cos(\omega t) \end{aligned} \tag{11}$$

where $\omega = (2\pi f_c)$ and f is the fundamental frequency

2) Calculate the amplitude of the reference voltage vector

$$V_{ref} = \sqrt{V_\alpha^2 + V_\beta^2} \tag{12}$$

3) Calculate the angle of the revolving reference voltage vector.

$$\alpha = \tan^{-1}(\frac{V_\beta}{V_\alpha}) \tag{13}$$

4) The reference voltage vector is determined by the identification of the sector through a revolving reference vector that is sampled once in every T_s

5) Based on the amplitude of the reference voltage vector and the revolving angle, the dwell time for the active and basic voltage vectors for the n sector is calculated as:

$$\begin{aligned} T_1 &= T_s m_a \frac{\sin(\frac{n\pi}{3} - \alpha)}{\sin(\frac{\pi}{3})} \\ T_2 &= T_s m_a \frac{\sin(\alpha - \frac{(n-1)\pi}{3})}{\sin(\frac{\pi}{3})} \\ T_0 &= T_s - T_1 - T_2 \end{aligned} \tag{14}$$

where $m_a = \frac{3V_{ref}}{2V_{dc}}$

SVPWM schemes divide T_0 equally between two basic or zero voltage vectors and apply two active and one zero voltage vector for each sector.

6) Calculate the duration of the zero vector and the switching instants for each switch

7) Implement the PWM signals for each switch using the switching instants

8) Repeat the process for each sampling interval.

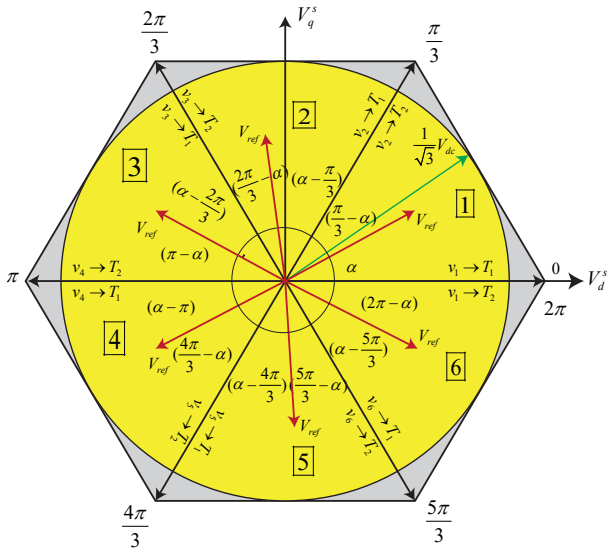


Figure 3: Calculation of dwell time for n sectors.

2. Simplified SVPWM

Simplified space vector pulse width modulation is a variation of the conventional sinusoidal pulse width modulation (SPWM) technique, where the reference voltage waveforms are employed to generate the offset voltages.

Simplified SVPWM produce the SVPWM signals directly by using the momentary reference phase voltages. The offset voltage is generated by utilizing the maximum and minimum of three phase reference signal, and this offset voltage signal is added to the instantaneous reference phase voltage, resulting in active inverter switching vectors being centred in a sampling time instant, making SPWM comparable to SVPWM scheme as illustrated in Figure 4. Two steps are required to implement the Simplified SVPWM approach for driving a two-level VSI.

a) We first need to measure the three-phase reference voltages. This can be done by sampling the voltage at each phase of the AC source. Once we have the voltage samples, we need to determine the offset voltage, which is the average

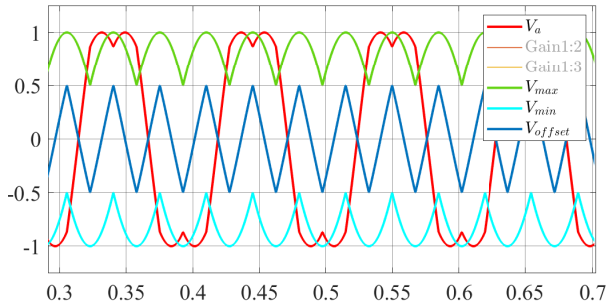


Figure 4: offset voltage.

of the maximum and minimum reference voltages. To find the maximum and minimum values, we can simply compare the voltage samples for each phase and store the highest and lowest values. Then, we can calculate the offset voltage as follow:

$$V_{offset} = -0.5 * (V_{max} + V_{min}) \tag{15}$$

b) The offset voltage is added with the reference phase voltage

The Simplified SVPWM technique offers several advantages over conventional SPWM, including reduced harmonic distortion, better voltage utilization, and improved switching efficiency. It also allows for a wider modulation index range and can generate higher output voltage levels.

Both the modulation schemes are implemented for two level three phase voltage source inverter fed with RL load having output LC filter. The test conditions are given in Table 3. The test results are illustrated in Figure 5,6.

Table 3: Simulation test parameters.

Parameter	**Description**	**Value**
F_{pwm}	Switching frequency	10 kHz
T_s	Sample time	1μs
L	Inductance	0.01H
R	Resistance	50Ω

Figure 5 illustrate the waveform of inverter phase and line voltage in linear

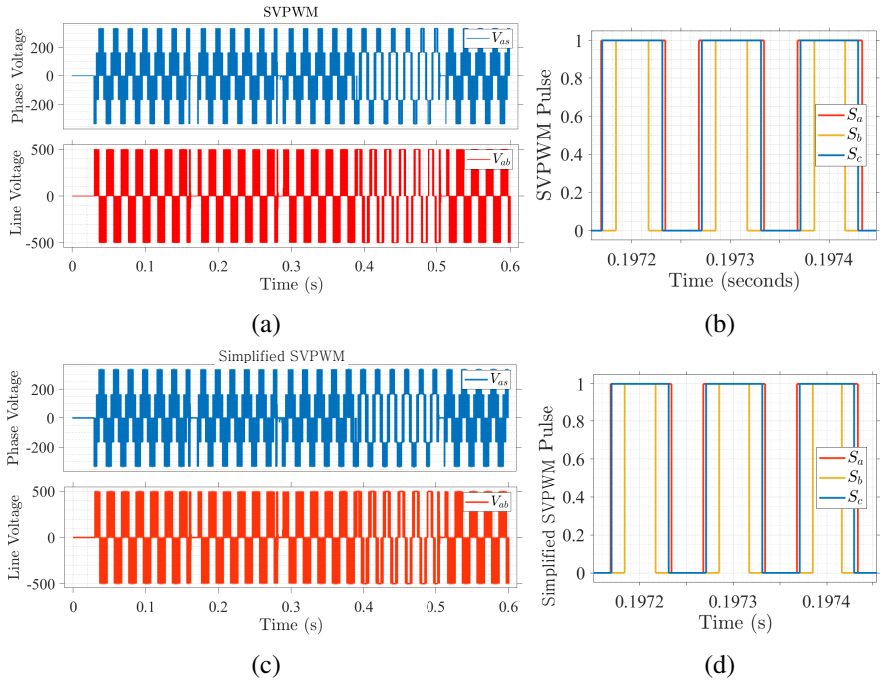
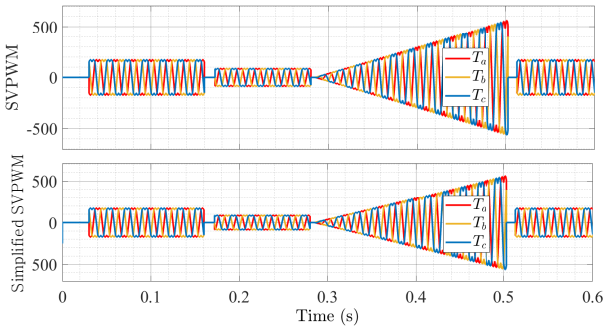


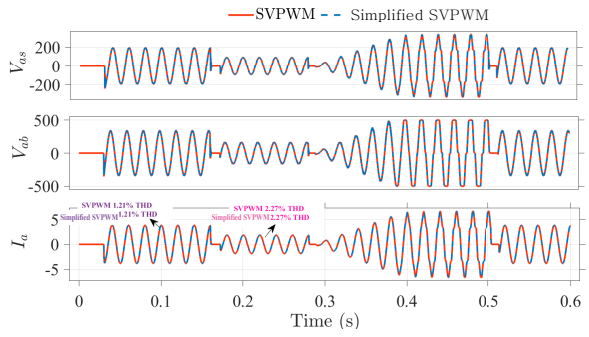
Figure 5: Phase and Line to Line voltage **(a and c)** Voltages**(b and d)** Gating Pulses.

modulation range. The gating pulses are also shown. The result waveforms are a perfect match, providing strong evidence that the simplified implementation of SVPWM is both feasible and effective in driving a three-phase two-level inverter with nonlinear load.

Figure 6 illustrate the inverter output with LC filter under same test conditions. The test result exhibits complete similarity in the transient operation. This demonstrates the capability of both methods to smoothly handle changes in voltage and current levels, ensuring stable performance across various operating conditions. Furthermore, the test results indicate that the total harmonic distortion against the phase current is also identical for both SVPWM and Simplified SVPWM. This signifies that both modulation techniques effectively mitigate harmonic distortions, resulting in cleaner and more reliable power output from the inverter. To further assess the performance of SVPWM and Simplified SVPWM, we established an experimental platform using a spindle drive system.



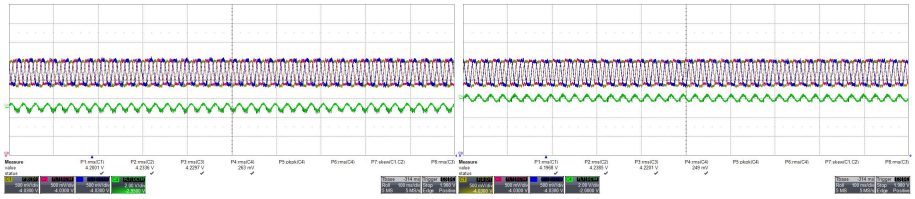
(a)



(b)

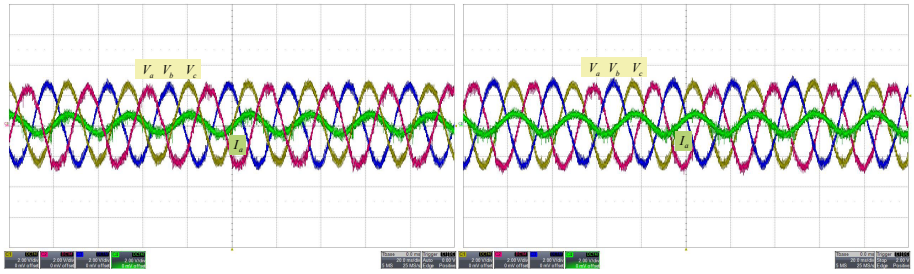
Figure 6: Inverter Output **(a)** SVPWM and Simplified SVPWM output**(b)** RL load with LC filter.

Figure 7: Experimental platform.



(a) SVPWM (b) Simplified SVPWM

Figure 8: Experimental Results: Pole voltage (T_a, T_b, T_c) and phase current .

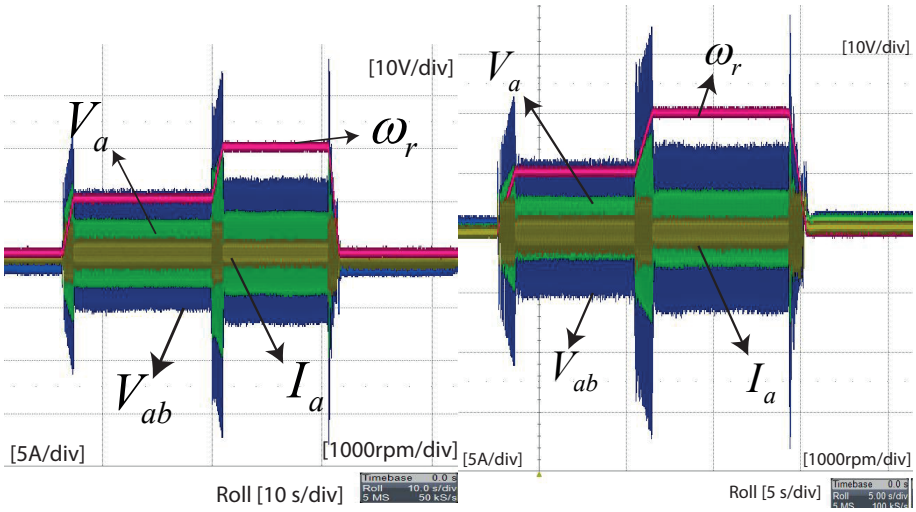


(a) SVPWM (b) Simplified SVPWM

Figure 9: Experimental Results: Inverter Output **(a)** SVPWM **(b)**Simplified SVPWM output.

The setup includes a 15kW inverter and a 3.7kW spindle motor as illustrated in Figure 7. The experimental results are presented to highlight Simplified SVPWM's advantages over SVPWM in industrial applications in term of ease of implemetation. In Figure 8, it is observed that the output of both modulation schemes and the phase current, showing similar responses. Additionally, Figure 9 displays the phase voltages and current signals, which also yield similar results under identical operating conditions. Finally, Figure 10 depicts a transient response where the motor speed increases from 0 to 1000rpm, then to 2000rpm (the rated speed), and returns to zero. Both modulation techniques exhibit the same performance.

The test results exhibit complete similarity, validating the practicality and effectiveness of the simplified implementation of SVPWM for driving a three-phase two-level inverter. Simplified SVPWM has proven to be computationally cheap, simpler to implement, and straightforward as compared to the state-of-



(a) SVPWM (b) Simplified SVPWM

Figure 10: Experimental Results: Motor drive response.

the-art SVPWM method, which relies on sector identification and lookup tables to calculate switching times for each sector. Due to its straightforward nature and reduced computational complexity, Simplified SVPWM has been chosen and adopted in this study.

IV. Optimization Strategies

The control of a system actuated by a PMSM is crucial to its performance. Model uncertainties and external disturbances are common in industrial control systems, make existing control theory based on precise models difficult to apply. Among various control strategies, still PI control is widely employed in industrial application due to its easy implementation and simplicity in design. The conventional PI control, on the other hand, has some flaws, such as large overshoot, slow rise-time, speed ripples and a restricted anti-disturbance capabilities which even make the system unstable. However due to nonlinear model of motor drive the effective performance is doubted under adversarial disturbance. Also they cause unwanted speed overshoot and sluggish behavior as a consequences of the controller's gain sensitivity and unexpected variation in load torque [44], [45]. Accurate variables and system models are crucial for PI controller efficacy. Therefore, Integral−Proportional controller is implemented to suppress aforementioned issues. To attain good controller performance, it is mandatory to manually tune the gain parameters of the controller. Thus, several novel tuning methods are presented in literature [46]–[48].The classical and optimization approaches for tuning the gain parameter is shown in Figure 11. In this work we studied most effective metaheuristic approaches for automatically offline tune the controller gain parameters for the outer speed loop and analyze the performance based on minimization of cost or decision function, computational burden and, speed ripples and transient response of the motor drive. PSO is the most commonly applied tuning approach in academia but the convergence of the algorithm need to be improved and for this reason extensive research work is done to optimize the hyper parameter of PSO algorithm for excellent tuning performance [49], [50]. To reduce computational complexity CS optimization is studied in this work. Through the course of the search, Cuckoo Search frequently keeps a higher degree of population variety. By doing so, the likelihood

of discovering global optima can be increased and premature convergence to inferior solutions can be avoided [51], [52]. Moreover, a parameter independent optimization approach named as JAYA proposed for gain tuning of the speed controller. For performance analysis ITAE is considered as the objective function to be minimized. Finally, we compute experimental and simulation results to show the efficacy of optimization algorithm for gain tuning and verify it by evaluating rise time, settling time, overshoot and speed ripples for speed control performance of PMSM.

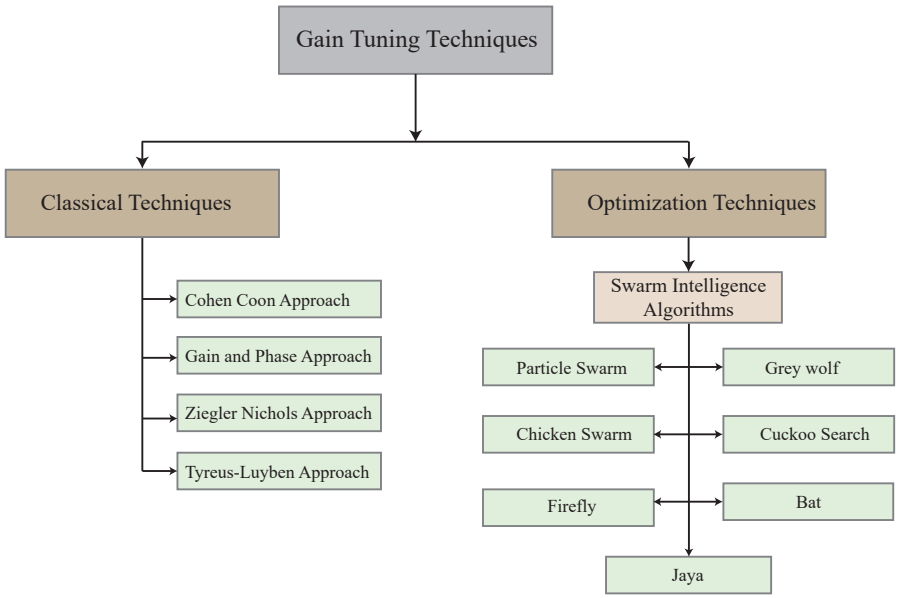


Figure 11: An overview of Controller tuning methods.

The block diagram of PI and IP speed controller for PMSM is shown in Figure 12. The closed loop transfer function for PI and IP speed controller is derived as follow:

$$G_{PI}(s) = \frac{\omega r(s)}{\omega_r^*(s)} = \frac{K_p K_t s + K_t K_i}{Js^2 + (B + K_t K_p)s + K_t K_i} \tag{16}$$

$$G_{IP}(s) = \frac{\omega_r(s)}{\omega_r^*(s)} = \frac{K_t K_i}{Js^2 + (B + K_t K_p)s + K_t K_i} \tag{17}$$

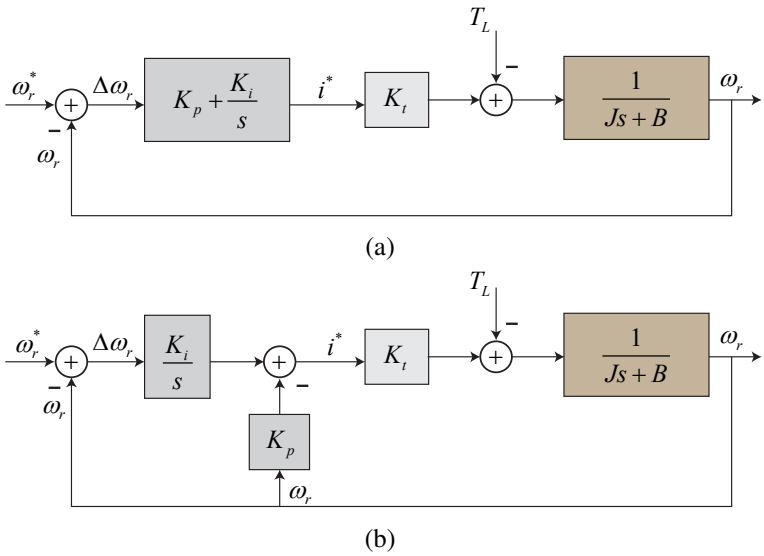


Figure 12: Block diagram of speed control loop **(a)** PI controller**(b)** IP controller.

The denominator of both functions is identical, whereas the numerators are different. Due to differential operator $'s'$ in the numerator of PI controller, by taking inverse Laplace, it appears that the output of PI is the summation of speed response of IP control in addition to the derivation of speed response of IP control [53]. The PI controller has zero steady-state error but has a large settling time. To reduce settling time, the P gain need to be increased, but it causes a large overshoot of the system. Moreover, increasing Kp can enhance tracking performance, it also makes the control system more sensitive to noise and disturbances. Any small fluctuations or noise in the speed feedback signal can cause the controller to overreact, leading to oscillations, instability, or even overshooting of the desired speed. To tackle this issue, IP controller is employed with P gain in the feedback path, thus, provides lower overshoot and undershoot at transitions,surpassing the performance of a traditional PI controller. The PI controller draws a current reference i^* larger than the IP. In addition, The PI controller's proportional action is highly responsive to instantaneous speed errors, swiftly generating a substantial control action when abrupt deviations occur

between the desired and actual speeds. This rapid response aims to promptly correct the error and may necessitate an elevated current reference for the motor to attain the target speed. In contrast, the IP controller's dominant integral action primarily addresses steady-state errors and may exhibit a less immediate reaction to momentary speed fluctuations, resulting in a comparatively lower demand for an instantaneous increase in the current reference. To avoid current peaking, the IP controller is essential for the speed loop of industrial applications. In this chapter, based on significance of IP controller, it is employed in speed control loop and optimal methods are presented to tune the gain parameters for excellent speed control performance of PMSM. The overall structural diagram for the control design is shown in Figure 13.

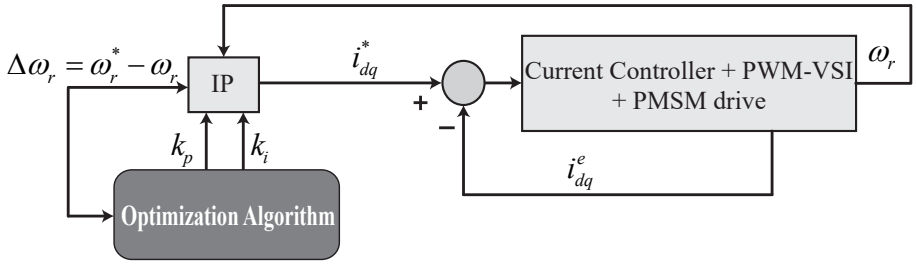


Figure 13: Structural representation of outer loop speed control of PMSM.

A. Cost Function

The cost function is used to quantify the performance of the system and guide the optimization process. The goal is to minimize the cost function by finding the optimal values for the IP gains that result in the desired performance criteria.

One commonly used performance criterion in IP gain tuning is the Integral of Time-weighted Absolute Error (ITAE). The ITAE criterion places emphasis on the magnitude and duration of the error between the system's output and the desired reference signal over time. It is defined as the integral of the absolute error multiplied by time, given by the following equation:

$$\text{ITAE} = \int_0^T t|\omega_r^*(t) - \omega_r(t)|. \tag{18}$$

Compared to ISE and ITSE, ITAE achieves faster settling time and reduced overshoot due to its time weighted nature. ITAE considers both the magnitude and duration of the error, providing a balanced evaluation of system performance. The main advantage of using the ITAE fitness function is that it provides a comprehensive measure of the control system's performance over time, taking into account both the magnitude and duration of the errors. The absolute error is used instead of squared error because ITAE fitness function places more emphasis on large errors, which are usually important for robust closed-loop control performance.

B. Particle Swarm Optimization

Particle Swarm Optimization is a population−based optimization algorithm inspired by the social behavior of bird flocking. In 1995, Eberhart and kennedy first proposed this optimization approach [54]. PSO is frequently employed to address optimization issues in a variety of industries, including engineering, data mining, finance, and machine learning.

PSO has been widely employed in several applications of automated control systems that mainly rely on PID controllers because of its intelligent performance and simple to employ. In this study, the Particle Swarm Optimization algorithm is implemented to tune the outer speed loop. The adaptive inertia weight approach is employed to balance the explorations and exploitation abilities of the swarm instead of traditional constant inertia weight. The selection of inertia weight is based on the performance of the algorithm. If the algorithm is rapidly improving, we choose a quicker drop, but if it is sluggish we choose a slower decline or even temporarily raise it to stimulate exploration. Furthermore, PSO is used to update

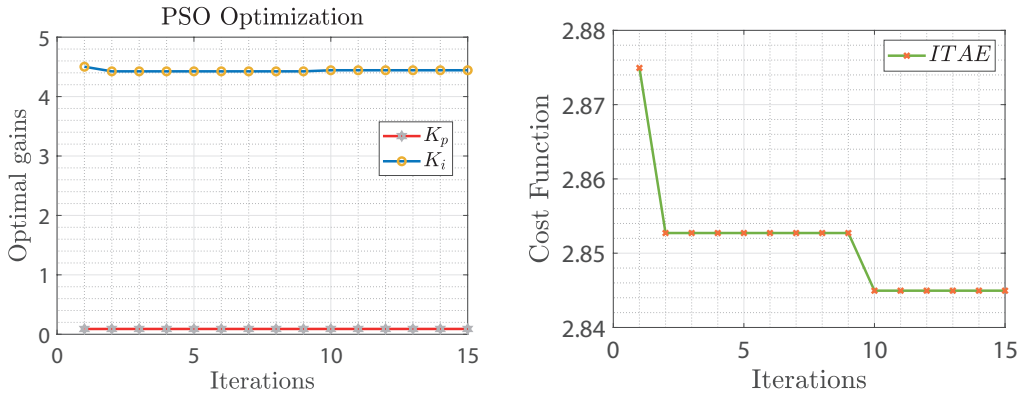
the velocity of a particle in the search space, which is a vector that represents the direction and magnitude of the particle's movement. The velocity update equation typically has two components: a cognitive component that guides the particle towards its personal best position, and a social component that guides the particle towards the global best position found by the entire swarm. The pseudo-code given in appendix for PSO Algorithm 1 describes the main steps for the gain selection process. The velocity of the each swarm particle i updated during the m-th iteration is expressed as follows [55]:

$$v_i^{m+1} = wv_i^m + c_1 r_1 (p_{best,i}^m - x_i^m) + c_2 r_2 (g_{best}^m - x_i^m). \tag{19}$$

where: v_i^{m+1} is velocity of particle i at $m+1$ iteration, w is inertia weight, $i = 1, 2, ...n$, m is the current iteration count, v_i^m is velocity of particle i at m-th iteration, c_1 is cognitive weight, r_1 is a random value in the range [0,1], $p_{best,i}^m$ is the personal best position of particle i, x_i^m is current position of particle i at m-th iteration, c_2 is social weight, r_2 is a random value in the range [0,1], g_{best}^m is the global best position found by the swarm at m-th iteration. The values of w, c_1, and c_2 are hyper-parameters that control the balance between exploration and exploitation of the search space. The values of r_1 and r_2 are random values generated at each iteration to introduce randomness into the velocity update equation.

To evaluate the fitness of each particle, the objective function is evaluated for the position of each particle. The particles are then updated based on the fitness evaluation, and the process is repeated until the best solution is found or a stopping criterion is met. The flowchart representation of implementing of PSO algorithm for gain tuning of outer speed control loop is shown in Figure. 15.

The obtained optimal gain values as shown in Figure 14 are used online in the outer speed control loop. A pseudo-code (Appendix 1) description of the Particle



(a) Optimal gains.

(b) Cost function minimization.

Figure 14: Selection of controller optimal gain using PSO.

Swarm Optimization (PSO) algorithm is used to explain the basic steps involved in the optimization process.

a) Initialize the position and velocity of each particle in the swarm from line 1.

b) Evaluate the fitness function for each particle to determine its fitness. Set the p_{best} to the particles utilizing the initial position with lowest fitness evaluation value and set g_{best} to the p_{best} of all particles from lines 2–3.

c) The '**for**' loop is run for the predefined number of iterations to update the velocity, position, personal best, and global best from lines 4–12.

d) Calculate the fitness function to update particle positions. If the updated fitness function is lower then current p_{best} and g_{best}, set the p_{best} and g_{best} to updated position.

e) Repeat steps (b) to (d) until a stopping criterion is met, such as a maximum number of iterations or a satisfactory fitness value.

f) The final output of the PSO algorithm is the g_{best} position, which corresponds to the optimal solution to the optimization problem.

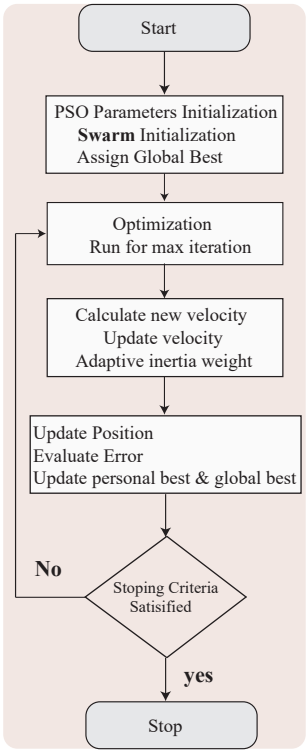


Figure 15: Flowchart of Particle Swarm Optimization.

C. Cuckoo Search Optimization

Cuckoo search is a nature−inspired optimization technique, and is based on the behaviour of cuckoo birds and their egg−laying approach. There are several applications that involve optimising a system or key components, and researchers have utilised cuckoo search in various fields such as structural optimization, parametric estimation in biological systems, MPPT for PV systems, nonlinear system identification, path planning for mobile robot [56]–[60]. In this study, nature−inspired cuckoo, a search optimization algorithm, was employed to tune the speed loop control gains, and and the performance is analyzed by comparing with other optimization technique which will aid researchers to select effective optimization technique based on there performance and computational

complexity. To enhance the convergence rate and solution accuracy in our Cuckoo Search algorithm, we've implemented an adaptive mechanism that modulates both the step size and the Lévy flight exponent. This adaptability is pivotal in striking an optimal balance between exploitation and exploration phases of the search process. By dynamically adjusting the step size, the algorithm adeptly sidesteps potential stagnation in local optima and promotes a thorough examination of the search space. Concurrently, the modulation of the Lévy exponent dictates the trajectory of the flight patterns, thus influencing the generation of new solution candidates. These strategic adaptations are instrumental in driving the algorithm towards superior solutions more efficiently.

The concept of CS is based on following rules:

a) Each cuckoo selects a nest at random and lays one egg at a time there.

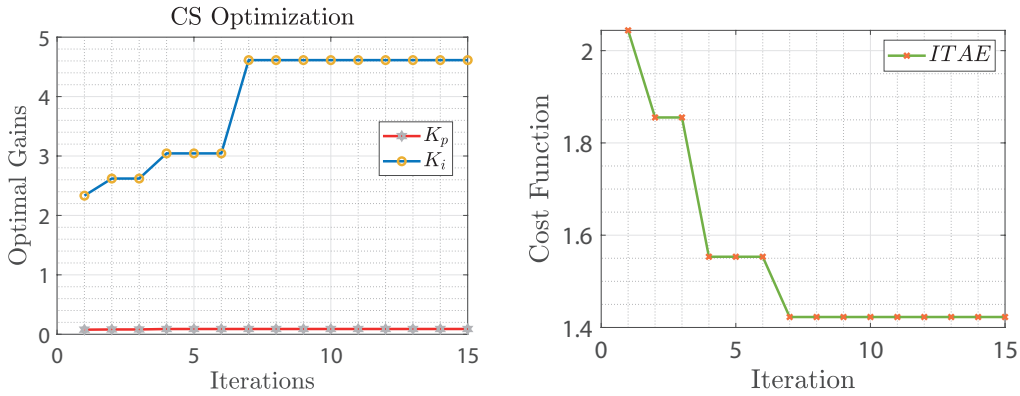
b) The best nest and best eggs can be passed on to the next generation.

c) The likelihood that the host bird will find the cuckoo's posed egg is [0, 1] based on the number of host nests.

When a host bird encounters cuckoo eggs in its nest, it has two options: it can either destroy the eggs or abandon the nest altogether. In either case, there is a chance that the host bird will build a new nest, and this probability p remains constant for a predetermined number of nests. To generate new solutions for cuckoo, a Lévy flight is executed as given below:

$$x_i(m+1) = x_i(m) + \alpha \oplus \text{Lévy}(\lambda) \tag{20}$$

where $x_i(m)$ presents samples (eggs), m is the iteration number, i is the sample number, α is the step size, and the $\oplus$ means entry wise multiplication. Lévy(λ)



(a) Optimal gains.

(b) Cost function minimization.

Figure 16: Selection of controller optimal gain using CS.

is calculated from Lévy distribution as follows:

$$\text{Lévy} \quad u = t^{-\lambda}, \qquad (1 < \lambda \leq 3) \tag{21}$$

The optimal gain values obtained by minimizing the cost function is shown in Figure 16. Figure 17 depicts the flowchart for controller optimal gain tuning procedure. The pseudocode for cuckoo search algorithm is given in the Appendix 2. The steps involve in employing cuckoo search based on listing 2 are describes as follow

a) In lines 1-3, a population of n candidates is initialized randomly and arranged in ascending order of their error scores.

b) Lines 4-14 describe the main *for* loop which runs for a defined number of iterations m_{max}.

c) Lines 5-11 perform a local search inspired by the behavior of Cuckoo birds.

d) A new candidate is generated by mutating a candidate selected at random from the population using equation (20). The mutation is applied using a

Levy flight which is a random walk while the random step length is drawn from a Levy distribution given by equation (21). If the new candidate is more fit, it replaces the previous candidate.

e) Next, lines 13 applies the parasitism operator. $(p * n)$ candidates with the worst error values are replaced by randomly generated candidates according to probability p. This operator helps maintain diversity in the population.

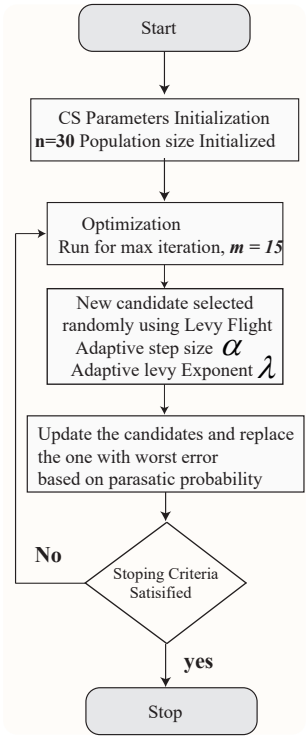


Figure 17: Flowchart of Cuckoo Search Optimization.

D. JAYA Optimization

In this work, Jaya algorithm, a simple yet effective optimization technique is provided for controller gain tuning. Jaya optimization is a population

based algorithm that comprises the feature of swarm intelligence and evolutionary techniques. Jaya algorithm is independent of algorithm specific hyper−parameters that make it a best choice for tuning the gain parameters of control system. Jaya algorithm aims to find global best solution by replacing worst solutions with best ones acquired through a combination of the best solutions in the current population [61].

The effectiveness and viability of JAYA−based algorithms allow it to be successfully employed in solving a variety of real-world applications, including feature selection, environmental engineering, and manufacturing industries [62]. It offers an effective method to automate controller gain tuning process and providing excellent control performance by minimizing control system error. The steps involve in employing Jaya optimization are given as follows:

a) Set up a population of initial candidate solutions, each of which corresponds to a set of IP controller gains.

b) Evaluate the performance of each candidate solution using a cost function that is based on control system error.

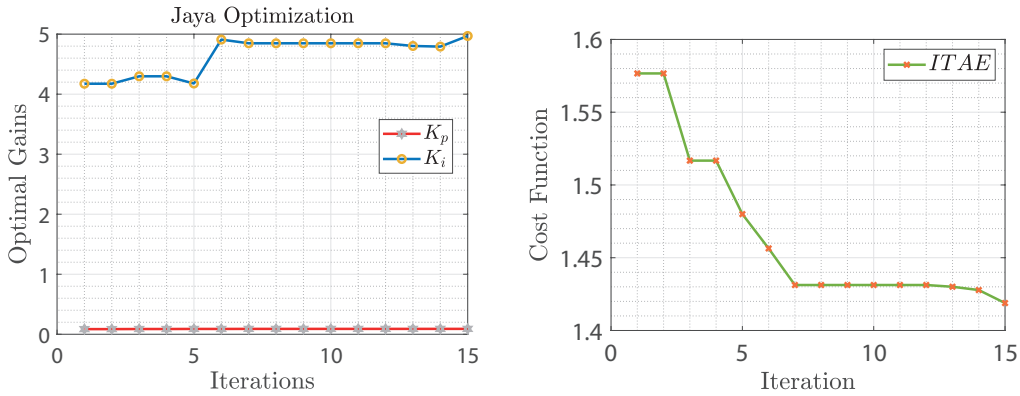
c) Jaya optimization update mechanism is applied to replace the worst candidate solutions with the new best solution in the given current population which involve adjusting the controller gains

d) The evaluation and update processes should be repeated until a termination criterion is satisfied, for a predetermined number of iterations.

Jaya optimization update mechanism is given as follow:

$$X^{'}(a,b) = X(a,b) + rand_1(X_{best}(b) - |X(a,b)|) - rand_2(X_{worst}(b) - |X(a,b)|). \tag{22}$$

where $X(a,b)$ is the value of b^{th} variable for the a^{th} candidate. Older solution



(a) Optimal gains.

(b) Cost function minimization.

Figure 18: Selection of controller optimal gain using JAYA.

is replaced with new solution based on cost function minimization. When $b_a^{new} < b_a^{prev}$ then X_a is replace with $X^{'}(a,b)$, whereas b_a^{prev} is replace with b_a^{new} respectively. After the new solution is updated, the final step is to stop the optimization, stopping criteria is evaluated that is if the current iteration is less than maximum iteration. The best solution is printed; otherwise, the process described above is repeated until the stopping criteria is met. The optimal gain values obtained using provided optimization algorithm is shown in Figure 18. Figure 19 show the flowchart for the tuning process and the Appendix 3 presents the pseudo code for jaya optimization.

Table 4: SPMSM Parameter Values.

Parameter	**Description**	**Value**
F_{pwm}	Switching frequency	16 kHz
r_s	Stator resistance	0.085Ω
L_s	Stator inductance	0.0012H
P	Poles pairs	4
K_t	Torque Constant	0.07671
J	Inertia	0.00215 $kg.m^2$
V_{dc}	DC voltage	311 V_{dc}
ω_r	Rated Speed	2000 rpm

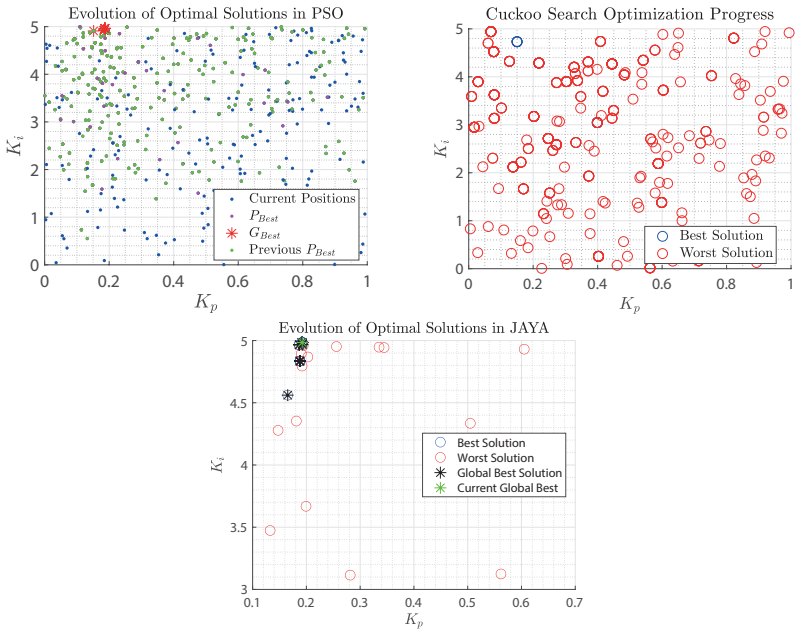


Figure 20: Optimization convergence trajectory.

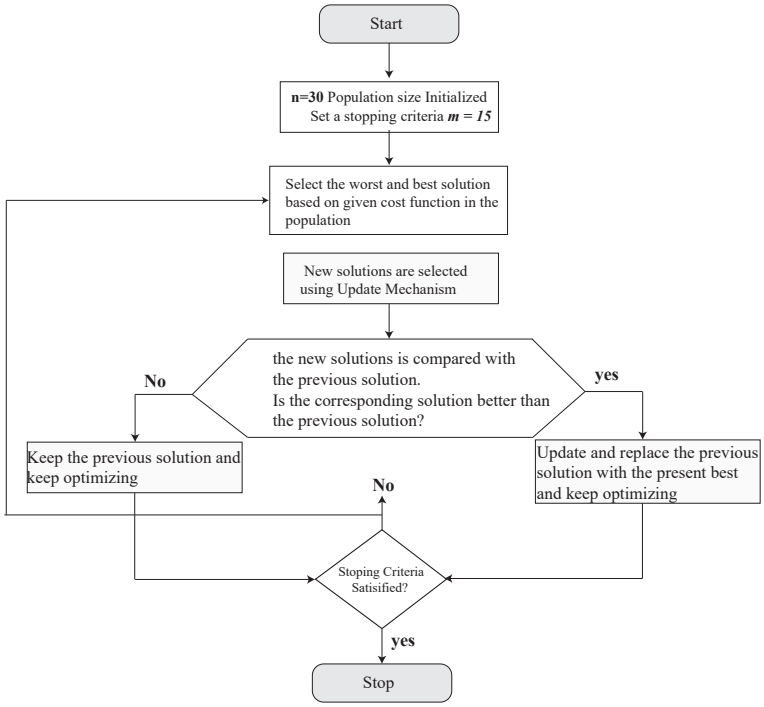


Figure 19: Flowchart of JAYA Optimization.

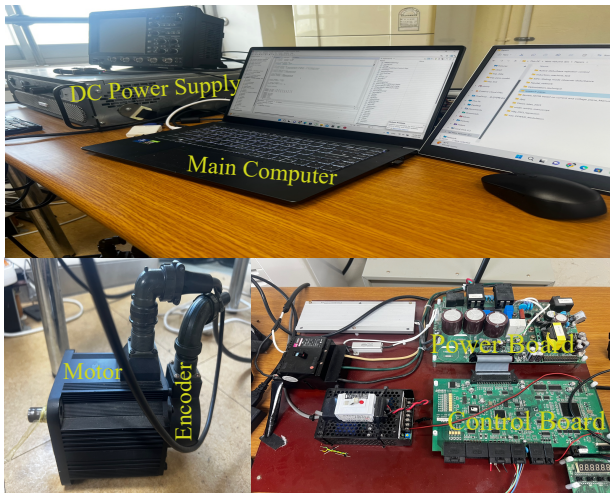


Figure 21: Experimental Platform.

E. Results and Discussion

The main objective of optimization technique is to minimize the control system error to enhance the controller performance. By employing the above mentioned optimization schemes, IP speed controller gain parameters are obtained and optimal controller control performance is verified using simulation and experiment results. The motor parameters and optimization hyperparameters are given in Table.4 and 5 respectively. The block diagram of closed loop control of SPMSM is shown in Figure 13. The optimization algorithm is employed offline Using Matlab to perform tuning and the tuned parameters are utilized online in the control system. The evolution of optimal solutions for a given metaheuristic approaches are illustrated in Figure 20. The optimization algorithms are performed via a laptop with an Intel® Core i7-1195G7 2.90GHz CPU, 16 GB RAM and NVIDIA Geforce® RTX laptop GPU 3050, and running window 11 64 bit. Furthermore, for experimental verification of given controller gain auto tuning algorithm the experiments were done with PMSM which parameters are given in Table 4. The control was implemented in a floating point

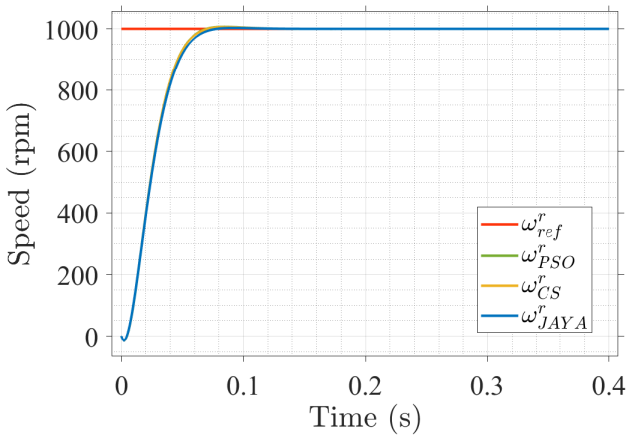
digital signal processing (DSP) board comprises of TMS320F28335. Figure 56 illustrate the experimental setup. The measured test results are saved in FRAM and plotted in Matlab for visualization.

Table 5: Optimization algorithm hyperparameters.

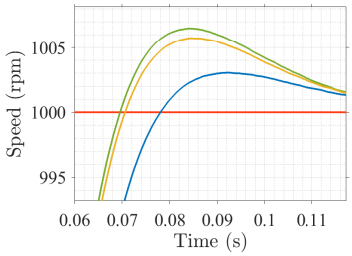
Parameter	Description	PSO	CS	JAYA
m_{max}	Maximum iterations	15	15	15
n	Population size	30	30	30
p	Parasitic probability	N/A	0.15	N/A
w	Inertia factor	0.729	N/A	N/A
c_1	Cognitive coefficient	2.0	N/A	N/A
c_2	Social coefficient	1.8	N/A	N/A
v	Upper & lower bounds	[-1.5, 1.5]	N/A	N/A

Due to the meta heuristic′s stochastic nature, a single run′s outcomes might not be reliable. Each algorithm is executed 15 times. The various algorithms processing times for calculating IP controller gains are illustrated in Table 6. Figure 22 shows simulation result for the speed response of 1000rpm at 3 $N.m$ load torque. The results for constant speed reference illustrate that different algorithm successfully converge to desired speed reference. The PSO algorithm gives higher overshoot and speed ripples whereas, jaya optimization algorithm gives minimum overshoot, and speed ripples. Figure 23 shows the experimental data which gives the step response to speed input reference of 1000rpm at no load condition for each optimization algorithm. It can be seen from test results that Jaya presents the best speed tracking performance followed by CS.

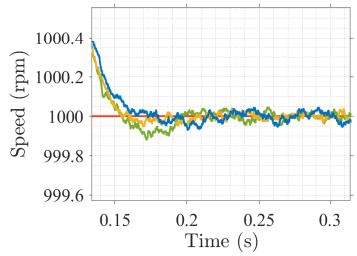
Figure 24 and Figure 25 depict the speed response to sudden change in speed step reference of each optimization algorithm in simulation and experiment. The test results illustrates the efficacy of nature− inspired optimization algorithms. The optimal IP controller gains for outer loop speed control obtained from the all optimization algorithms shows that the optimization methods have similar response to speed input change and converges to reference speed values, only



(a) Motor Step Response.



(b) Overshoot.



(c) Steady−state performance.

Figure 22: Motor response at step speed reference with constant load.

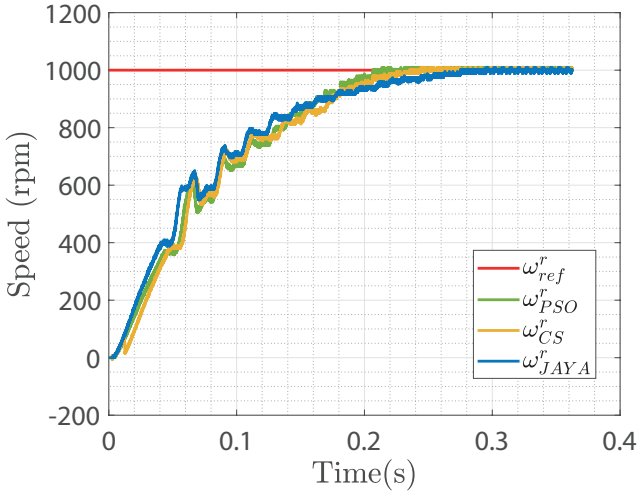


Figure 23: Experimental results for speed convergence to speed step reference.

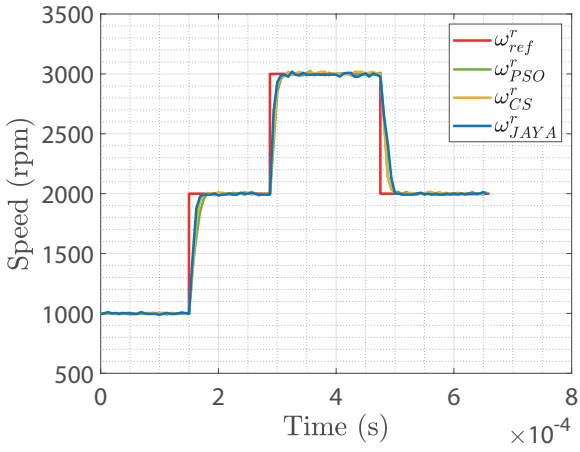


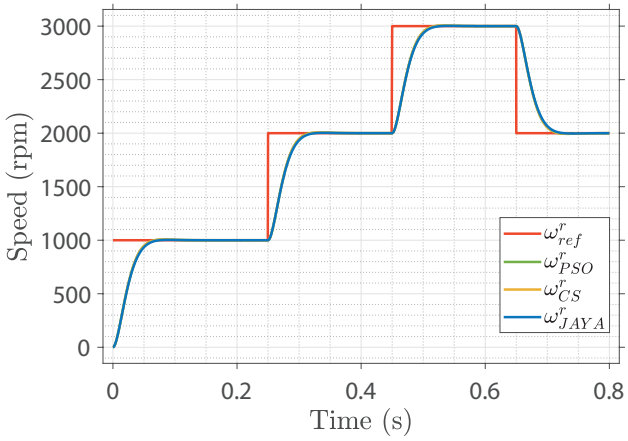
Figure 24: Experimental results for speed convergence at variable speed reference.

highlighting the jaya algorithm as the one with minimum overshoot,undershoot and speed ripples.

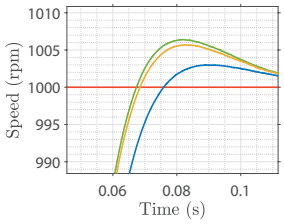
Figure 26 shows the simulation results for a speed response under varied load conditions. Motor is loaded with load torque at various time instances and the response of the controller is shown. It is obvious from the test results that all the optimized strategies shows same response to the disturbance at constant speed input. The result demonstrate, the controller ability to respond effectively to disturbance and maintain stability by converging to reference input values.

Table 6 is provided to give qualitative and quantitative analysis of optimization strategies to further verify the efficacy of the given optimization algorithms which will aid the researchers to select efficient strategy for optimal gain tuning of Linear controller that are usually employed in vector control of PMSM. When observing the algorithm fitness value as a performance indicator,the JAYA outperformed all other algorithms as it provides lowest fitness value $(ITAE)$ for all the runs.

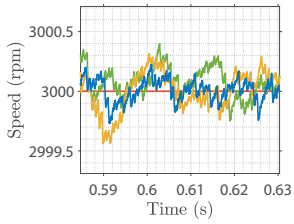
overall, the conducted simulations and experiments illustrates promising results for the speed control of PMSM. the jaya optimization algorithm



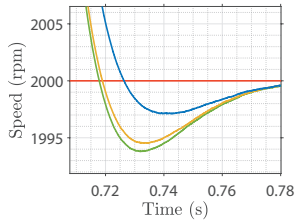
(a) Motor Step Response.



(b) Overshoot.

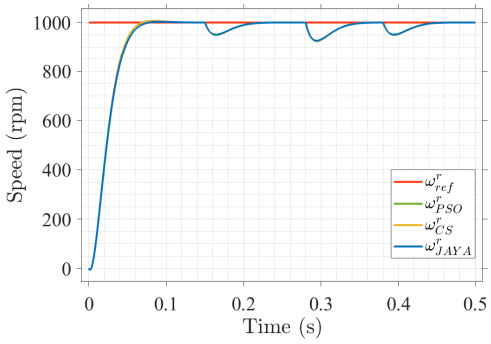


(c) Motor response at step reference.

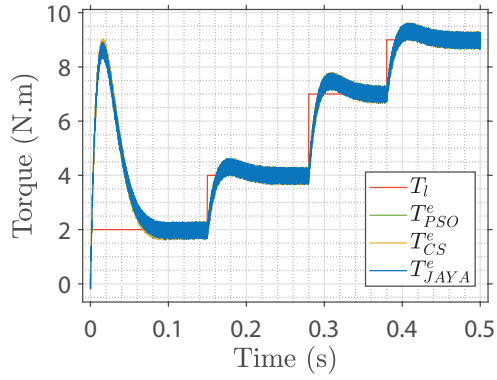


(d) undershoot.

Figure 25: Motor response at variable step speed reference.



(a) Motor Step Response.



(b) Motor response at Variable load input.

Figure 26: Motor response at step speed reference with Variable load input.

Table 6: Algorithm performance comparison.

Run	Initial Error	Optimized Error	Elapsed Time	Mean Elapsed Time
		PSO		
#1	2.8749	2.8450	2 hr 59 min	∼3 hr 40 min
#2	1.5266	1.4618	4 hr 26 min	
#3	2.432	1.872	3 hr 28 min	
		CS		
#1	2.0440	1.4227	2 hr 7 min	∼2 hr 5 min
#2	1.533	1.4643	1 hr 40 min	
#3	2.145	1.654	2 hr 30 min	
		JAYA		
#1	1.57766	1.4189	3 hr 55 min	∼3 hr 15 min
#2	1.4643	1.4250	3 hr 2 min	
#3	1.8721	1.230	2 hr 55 min	

Table 7: Control Performance Comparison

Parameters	**PSO**	**CS**	**JAYA**
Speed ripples	0.074	0.087	0.065
Speed overshoot $(\%)$	0.64	0.56	0.29
Speed settling time (s)	0.19	0.2	0.18
Computational burden	high	low	medium

outperforms other nature inspired algorithm discuss in this chapter in term of minimum overshoot,undershoot and speed ripples as given in Table 7. Even though the rise time of jaya algorithm is higher than other optimization methods but the speed tracking performance and robustness under transient condition is achieved by minimizing the control system error.

V. Advanced Control

Interior Permanent Magnet Synchronous Motor (IPMSM) drives have garnered considerable interest as variable speed drives in industrial applications. This is primarily due to their exceptional attributes, including high efficiency, high power density, compact size, and rapid dynamics, all of which contribute to low maintenance costs. By replacing rotor windings with permanent magnets, these motors achieve swift responses due to their low inertia. The objective of this chapter is to showcase the outstanding performance of an Interior Permanent Magnet Synchronous Motor (IPMSM) under variable load conditions, demonstrating its robustness and effectiveness. A distinct control design for Interior Permanent Magnet Synchronous Motors (IPMSM) is introduced in this chapter, and control performance is systematically compared with the traditional approach. The aim is to verify and demonstrate the effectiveness of the proposed design in enhancing the motor's overall closed−loop performance.

An effective compensation scheme for improved stand still and transient performance of IPMSM at low speed is investigated in this chapter. The vector control method has been a conventional choice for controlling IPMSM drives, relying on precise rotor shaft speed and position information for achieving high-performance control. However, at low speeds, external disturbances such as parasitic torque ripples and parameter uncertainties arising from imperfect machine design can lead to uncertainties in sensor measurements due to noise and the influence of mechanical loads or power electronic switches. Consequently, the speed trajectory tracking performance suffers, and the overall system robustness is compromised [63]–[65].

To ensure a high-precision trajectory tracking performance for IPMSM drives, it becomes essential to address these challenges. The primary goals are to mitigate periodic torque ripples and overcome the effects of parametric variation. By doing so, the control system can maintain superior performance even at low speeds and

effectively handle external disturbances and uncertainties, ensuring the reliability and efficiency of the IPMSM drive.

In this study, an effective compensation scheme coupled with a sliding-mode speed controller was implemented to enhance the dynamic and standstill response of an IPMSM drive at low speeds. The proposed control strategies ensure a reduction in speed ripple under low-speed operating conditions. The speed control loop was designed based on the exponential reaching law sliding-mode control (ERL-SMC) method, which remains independent of the motor model. Additionally, a compensation signal was injected into the reference current and the output of the speed control loop. This approach leveraged low-frequency current and torque disturbances to adapt actual reference variables and reject unwanted signals that could disturb the system.

The performance of the proposed control scheme was compared with conventional field-oriented control [66]. The results demonstrated that the proposed scheme effectively compensated for torque ripples, leading to significant improvements in drive performance. Specifically, rotor shaft speed ripple minimization and enhanced transient/steady-state performance were achieved, indicating the efficacy of the proposed approach in mitigating low-speed challenges and ensuring smooth and reliable operation of the IPMSM drive.

A. Maximum Torque per Armature

Due to the inherent salient pole structure of IPMSM drives, the d-axis inductance (L_d) differs from the q-axis inductance (L_q), leading to the presence of a reluctance torque component. This complex relationship between the dq-axis reference currents makes the implementation of the maximum torque per armature (MTPA) control challenging. To address this issue, a simplified dq-axis reference current equation was derived in this study based on known motor parameters as given in Table 8.

A simple equation for the d-axis reference was derived based on the q-axis reference current. Employing the provided motor parameters, a linear relationship between the torque and q-axis reference current was derived. Considering

$$I_s = \sqrt{i_d^2 + i_q^2}, \tag{23}$$

the equation for i_d^* reference current can be expressed in terms of i_q^* reference current as [67]:

$$i_d^* = \frac{\lambda_m}{2\left(L_q - L_d\right)} - \sqrt{\frac{\lambda_m^2}{4\left(L_q - L_d\right)^2} + i_q^{*2}}. \tag{24}$$

For simplicity, the above d-axis reference current is expanded by a Taylor series expansion across a point reaching zero. Substituting the given motor parameters and neglecting high-order terms because of the smaller magnitude yield,

$$i_d^* = \frac{\left(L_d - L_q\right)}{\lambda_m} i_q^{*2}. \tag{25}$$

Here, i_q^* is calculated by substituting drive parameters in (24) and expressed as

$$i_d^* = 16.72 - \sqrt{279.62 + i_q^{*2}}. \tag{26}$$

(26) is expanded by the Taylor series expansion at $i_q^* = 0.001$. Eliminating the smaller magnitude term yields

$$i_d^* = -0.0299(i_q^* - 0.001)^2. \tag{27}$$

Substituting (27) into (8d) with the given motor parameters and solving for i_q^* yields

$$i_q^* = 0.437T_e + 0.001. \tag{28}$$

(25) and (28) are employed online for the MTPA control of IPMSM drive. These equations help utilize reluctance torque and increase the efficiency of the drive.

B. Torque Pulsation Analysis

The periodic torque ripples observed in the IPMSM can be categorized into two groups: The first group arises from inherent motor drive structural factors, such as flux harmonics, cogging torque, and phase imbalances. The second group is attributed to drive control issues, such as gain mismatches, sensor offset errors, processing delays, and the effect of inverter dead time. The electromagnetic torque of the motor is dependent on the magnetic flux, reference current, and mechanical speed, as indicated by the motor drive transfer function. Consequently, at low operating speeds, the shaft speed experiences oscillations at the same harmonic frequency as the changes in electromagnetic torque, leading to fluctuations in the speed performance at steady state. This oscillation negatively impacts the control performance of the motor drive. Figure 27 shows a pictorial

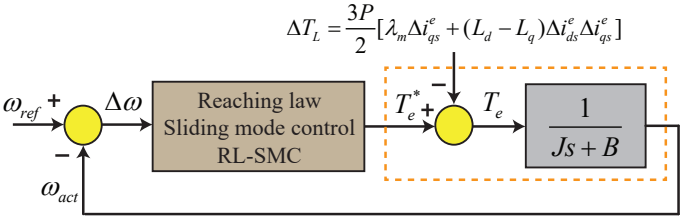


Figure 27: Block diagram of speed control loop.

representation of the speed control loop. In this section, the detailed discussion of how various disturbances contribute to the generation of periodic torque ripple and their subsequent effects on the motor drive performance.

1. The cogging torque in a motor is a result of the interaction between the magnetic flux and stator slots. This interaction leads to torque fluctuations due to the variable energy storage in the air gap between the stator slots.

Constructing an efficient mathematical model to compensate for cogging torque proves challenging due to the lack of suitable cogging torque models.

Previous studies [68] have demonstrated that cogging torque exhibits periodicity with respect to the rotor shaft position. The cogging torque ripples occur at frequencies that are multiples of the number of stator slots per pole pair and the electrical frequency of the rotor. The cogging torque is directly influenced by the structural design of the motor body, including factors like winding distribution, slot skewing, and the number of slots per pair. The motor drive design process is non-repeatable, meaning that once the motor drive is finalized, it cannot be modified [69], although further discussion on this aspect is beyond the scope of this article.

2. Secondly, the presence of harmonics in the air-gap flux is a key factor contributing to torque pulsation in the IPMSM. In real drive systems, achieving a pure sinusoidal flux density distribution across the air gap proves challenging, leading to torque ripples. These ripples occur due to the interaction of the standard stator phase current with a non-sinusoidal flux density distribution in the air gap.

In a three-phase system, the flux linkage between the drive phase current and the magnet contains harmonics of the order 5, 7, 11, and so on. When expressed in the dq reference frame, the corresponding torque exhibits harmonics of the 6^{th}, 12^{th}, and other multiples of the sixth harmonic. These harmonics further contribute to the torque pulsation observed in the IPMSM.

C. Reaching Law Sliding Mode Control

SMC has several advantages over other nonlinear control schemes. It is easy to implement, robust to variation, and has a fast response, which leads to excellent dynamic performance. Nonetheless, a prominent limitation of this controller manifests in the form of chattering phenomena, which incite high-frequency dynamics. A pivotal facet in the design of Sliding Mode Control involves the

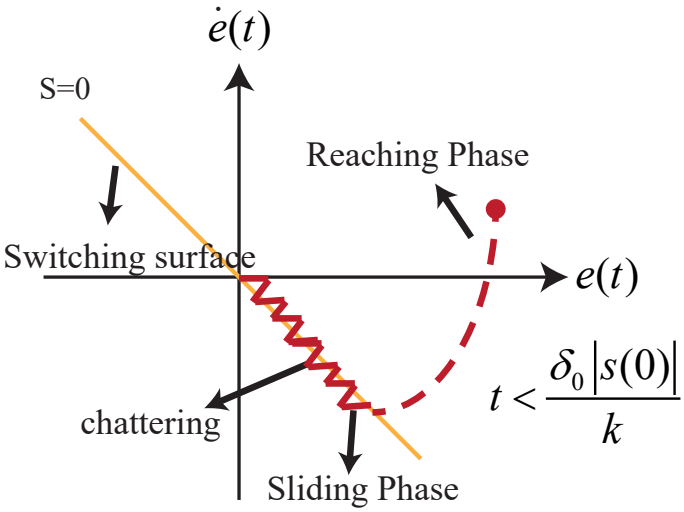


Figure 28: Sliding phase mechanism of the SMC.

mitigation of this chattering issue, alongside the augmentation of controller robustness and to reduce the reaching time. In response to these challenges, a reaching law is designed based on an exponential term that modifies the change in the system states and sliding surface.

To design SMC, the sliding surface was designed, and a control input was constructed that forced the system trajectory toward the sliding surface, guaranteeing a condition reaching the sliding mode. The sliding mode mechanism in the phase plane is shown in Figure 28, and the nonlinear model to express the SMC is

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = A(x) + B(x)u \end{cases} \tag{29}$$

where $x = [x_1 \;\; x_2]^T = [\omega_r \;\; \frac{p}{j}((L_d - L_q)i_d i_q + \lambda_m i_q)]^T$ is the system states. A and B are nonlinear functions in term of x, and B is invertible. The tracking error dynamics are given as:

$$\begin{cases} e_1 = x_1 - x_1^d \\ e_2 = x_2 - x_2^d \end{cases} \tag{30}$$

where $\dot{e}_1 = e_2$, and x_1^d is reference motor speed and x_2^d is virtual control input respectively. The nonlinear system model to express SMC following the reaching law is given as:

$$s = \dot{e}_1 + ce_1 = e_2 + ce_1, \qquad c > 0. \tag{31}$$

(31) ensures the stability of the sliding mode, whereas the convergence of the system trajectory to the sliding surface is related to the value of c in the sliding-mode surface. Once the sliding surface is selected, the next step is to select the switching controller u in such a way that it enables the error vector to reach the sliding surface and ensure the sliding mode reaching condition, which is expressed as

$$s.\dot{s} < 0. \tag{32}$$

To satisfy (32), the conventional reaching law is

$$\dot{s} = -k_r \cdot sign(s), \qquad k_r > 0. \tag{33}$$

The reaching time t_r required by the error vector to converge to the sliding surface is obtained by integrating (33) with respect to time, which is derived based on Lyapunov stability:

$$\begin{aligned} V &= \frac{1}{2}s^2, \\ \dot{V} &\leq -K_r|s|, \\ \dot{s}s &\leq -K_r|s|, \\ \dot{s} &\leq -K_r \operatorname{sign}(s), \end{aligned}$$

Now, integrating both sides from 0 to t_r:

$$\int_0^{t_r} \dot{s}\,dt \leq \int_0^{t_r} -K_r \operatorname{sign}(s)\,dt$$

The right-hand side depends on the sign of s:

$$\int_0^{t_r} \text{sign}(s)\, dt = \begin{cases} 0, & \text{if } s(0) = 0 \text{ and } s(t_r) = 0 \\ t_r, & \text{if } s(0) < 0 \text{ and } s(t_r) > 0 \\ -t_r, & \text{if } s(0) > 0 \text{ and } s(t_r) < 0 \end{cases} \tag{34}$$

So, the right-hand side simplifies to:

$$-K_r \int_0^{t_r} \text{sign}(s)\, dt = -K_r t_r$$

Now, combine the left and right side of inequality:

$$s(t_r) - s(0) \leq -K_r t_r.$$

The expression is simplified, assuming the system starts at $s(0)$, reaches the sliding surface at t_r, and $s(t_r) = 0$, signifies that the system has achieved the sliding surface.

Now, rearrange the terms and isolate t_r, the reaching time constant is derived as

$$t_r = \frac{|s(0)|}{k_r}. \tag{35}$$

(35) reveals that the reaching speed is increased by increasing the value of k_r and excellent robustness can be achieved; however, this causes the chattering problem because the chattering increases with the increase in k_r [68].

To overcome this interdependence between the reaching law and chattering level, an exponential reaching-law-based sliding-mode controller was designed. The reaching law was selected based on the choice of the exponential term that

adapts to the change in the sliding-mode switching function. The reaching law is

$$\dot{s} = -\frac{k}{N(s)} \cdot \text{sign}(s), \quad k > 0, \tag{36}$$

where

$$N(s) = \delta_0 + \left(1 + \frac{1}{|e_1|}\right) e^{-a|s|}.$$

Here, k is strictly positive integer, $0 < \delta_0 < 1$ and $a > 0$.

- δ_0 is a positive constant, ensuring that the denominator $N(s)$ remains non-zero.
- The term $e^{-a|s|}$ is an exponential function that decreases as $|s|$ increases. This allows the control action to adapt to the change in the switching function.
- The term $\left(1 + \frac{1}{|e_1|}\right)$ scales the exponential function. Here, e_1 is the error of the system.

This exponential reaching law has no effect on the stability of the control design because $N(s)$ is always strictly positive. (36) shows that, if $|s|$ decreases, the denominator of term $k/N(s)$ reaches to $1+1/|e_1|$, causing $k/N(s)$ converge to $k|e_1|/(1+|e_1|)$. This means that the system trajectory reaches to sliding surface, in which e_1 decrease to zero under the control input, which helps to reduce the chattering level. In addition, if $|s|$ increases, then the term $k/N(s)$ converges to k/δ_0 which is greater than k, thus ensuring faster reaching time. The designed controller can easily adapt to the variation of the switching controller by allowing $k/N(s)$ to vary between k/δ_0 and k. The reaching time in which the system trajectories approach the sliding surface is obtained by integrating (36), noticing that $s(t) = 0$, and integrating the limit from $\int_0^t$, which yields

$$t_r = \frac{1}{k}\left[\delta_0 |s(0)| + \frac{(1+1/|e_1|)}{a}\left(1 - e^{-a|s(0)|}\right)\right]. \tag{37}$$

Because $(1-e^{-a|s(0)|}) < 1$, the inequality expression is

$$t < \frac{1}{k}\left(\delta_0|s(0)| + \frac{1+1/|e_1|}{a}\right). \tag{38}$$

if a is selected such that $a \gg \frac{1+\frac{1}{|x1|}}{\delta_0 s(0)}$, implying a as the ratio of the scaling factor of the exponential term to the leading term$(\delta_0|s(0)|)$. If a (the exponent in $e^{-a|s|}$ is much greater than this ratio, then the exponential term becomes very small and fast decaying, rendering its impact on the overall sum becomes negligible. (38) is simplified as

$$t < \frac{1}{k}(\delta_0|s(0)|). \tag{39}$$

(35) is subtracted from (39) with the condition that gain $k_r = k$ yields

$$t - t_r < \frac{1}{k}(\delta_0|s(0)|) - \frac{1}{k_r}(|s(0)|) = \frac{1}{k}(\delta_0 - 1)|s(0)|. \tag{40}$$

Here, $(\delta_0 - 1)$ is always negative as $0 < \delta_0 < 1$ and the term $|s(0)|/k$ is strictly positive, which implies that $t - t_r < 0$ and the robustness of the control increases with the fast-reaching speed of the sliding surface with the same gain. However, if the condition $t = t_r$ is set, then

$$k < \delta_0 k_r. \tag{41}$$

Therefore, the reduction in chattering level is guaranteed as $K < K_r$, which means that a fast reaching speed with less chattering can be achieved. For the speed control mechanism, the speed tracking error is introduced as $e = \omega_{ref} - \omega_m$ which maintains the actual speed at the reference speed. The sliding-mode surface based on the aforementioned scheme is defined as

$$s = e = \omega_{ref} - \omega_m. \tag{42}$$

Taking the derivative of the tracking error and speed, the equivalent controller is calculated by setting $d(s)/dt = 0$ yielding

$$\frac{d(s)}{dt} = \dot{\omega}_{ref} - \dot{\omega}_m, \tag{43}$$

where $\dot{\omega}_m = (T_e - B\omega_m)/J$. Finally, the control law is the sum of (43) and (36), expressed as

$$\frac{d(s)}{dt} = \dot{\omega}_{ref} - \frac{T_e + B\omega_m}{J} - \frac{k}{N(S)} \cdot \text{sign}(S) = 0. \tag{44}$$

Therefore, the control input i_q^* is derived as

$$i_q^* = \left(\frac{2J}{3P[\lambda_m + (L_d - L_q)i_d^*]}\right)\left[\dot{\omega}_{ref} + \frac{B\omega_m}{J} - \frac{k}{N(S)} \cdot \text{sign}(S)\right]. \tag{45}$$

By utilizing (28) the torque reference is obtained.

Table 8: Test parameters.

Parameter	Description	Value
F_{pwm}	Switching frequency	10 kHz
T_s	Sampling time	1μs
r_s	Stator resistance	2.5 Ω
L_d	d-axis inductance	15.025 mH
L_q	q-axis inductance	30.175 mH
λ_m	Rotor flux	0.5283 Wb
J	Moment of inertia	0.00365 kgm^2
B	Damping coefficient	0.0011 Nm

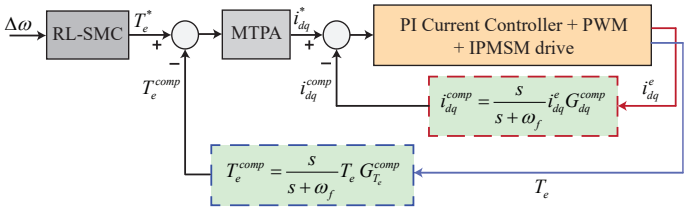


Figure 29: Structural diagram of proposed compensation scheme.

To minimize the speed ripple at a low operating speed, the reference dq-axis current and the output of the speed controller are modified by the current variations and disturbances in the electromagnetic torque, which minimizes the disturbance effect and increases the robustness under load variation. Figure 29 shows the structural design of the control scheme. The compensation signals are obtained by employing a high-pass filter. The mathematical expression derived for the dq-axis current compensation signals is

$$i_{dq}^{comp} = \frac{s}{s+\omega_f} i_{dq}^{e} G_{dq}^{comp}. \tag{46}$$

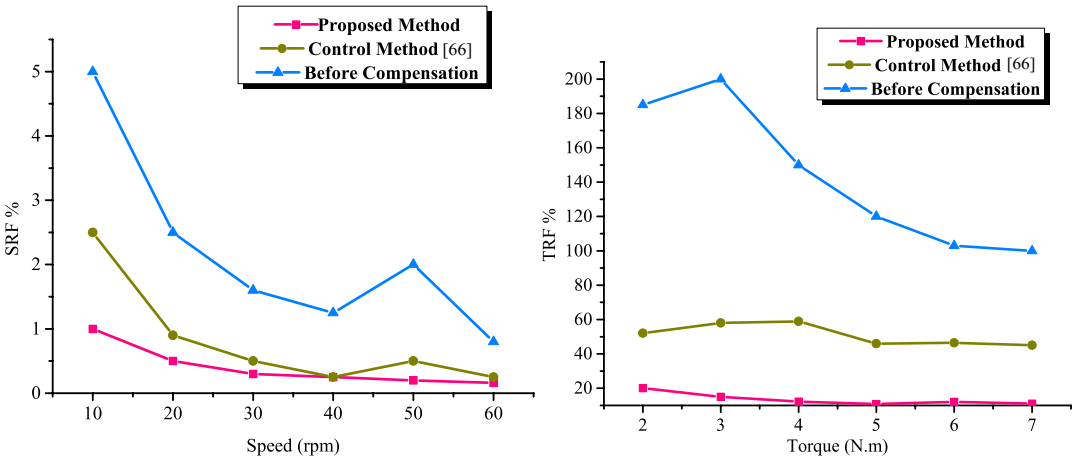
The expression for the torque compensation signal is given as

$$T_e^{comp} = \frac{s}{s+\omega_f} T_e G_{T_e}^{comp}. \tag{47}$$

where ω_f and G^{comp} are the filter cut-off frequency and the compensation gain.

D. Results and Discussion

To showcase the effectiveness of the proposed control design, a simulation model was developed. The performance evaluation was carried out using an IPMSM drive with specific parameters outlined in Table 8. A comparative study was then conducted using simulation tools, highlighting the control design's robustness and capability in achieving excellent standstill performance, particularly under low-speed working conditions. The simulation results demonstrated the superiority of the proposed control approach in handling various test scenarios and its potential to provide reliable and precise motor performance even at challenging operating conditions. The compensation gains for the electromagnetic torque and dq-axis current loop are configured as $K_{T_e}^{comp} = 10$ and $K_{i_{dq}}^{comp} = 2$, respectively. Furthermore, the cutoff frequency is set at 50



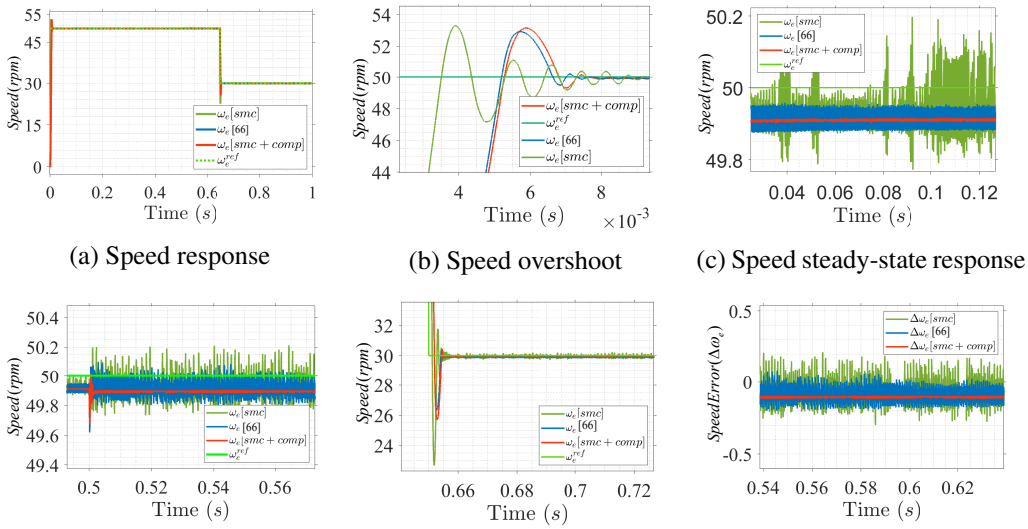
(a) Speed ripple factor. (b) Torque ripple factor.

Figure 30: Analysis of SRF for various speed and torque.

rad/s. The ripple factors for the speed and torque of the IPMSM drive are presented in Figure 30. These factors, known as the torque ripple factor (TRF) and speed ripple factor (SRF), are calculated by taking the ratio of the peak-to-peak torque and speed ripples to the rated torque and speed of the IPMSM drive. The expressions for TRF and SRF are derived as follows:

$$TRF = \frac{T_{e,pk-pk}}{T_e} \times 100 \quad (\%),$$
$$SRF = \frac{\omega_{e,pk-pk}}{\omega_e} \times 100 \quad (\%).$$

Figure 30a and 30b show the evaluation of the efficacy of the designed control algorithm with conventional control designs. For Figure 30a the speed is varied [10,20,30,40,50,60] rpm under constant load torque of 7 Nm, whereas in Figure 30b the torque is varied [2,3,4,5,6,7] Nm under constant speed of 50 rpm. The result reveals that employing the proposed control design ripple across the speed and torque effectively results in a reduction when the reference value increases. Compared with the conventional design, the ripples are greatly minimized with



(a) Speed response (b) Speed overshoot (c) Speed steady-state response

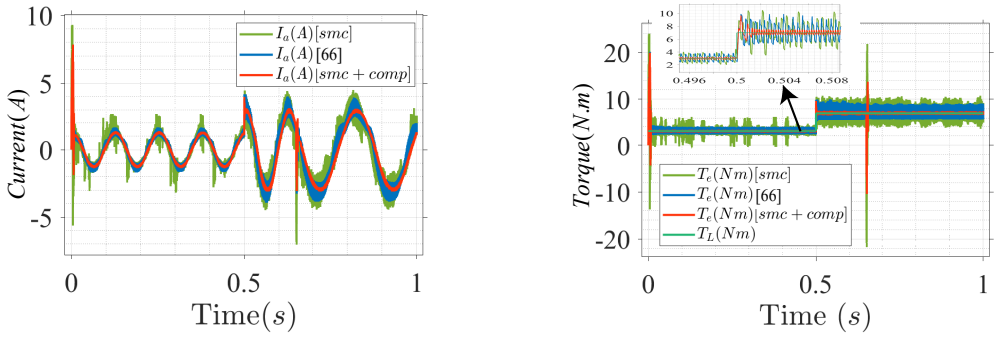
(d) Speed during load applied (e) Speed at 30 rpm (f) Steady-state speed error

Figure 31: Speed profile at low speed operating range under load variation.

the proposed control design with increasing robustness, especially at low speed.

Figure 31 shows the speed response under various load conditions. A sliding-mode speed controller is applied to eliminate linear PI controllers that are sensitive to external disturbances. Initially, a reference speed of 50 rpm is applied under a load torque of 3 rpm that is shown in Figure 31a. Figure 31b and 31c show the settling time and steady-state error of the drive, respectively. The settling time of the proposed design is fast, and the steady-state error is reduced. At 0.5 s, a load torque of 7 Nm is applied with a speed change from 50 to 30 rpm at 0.6 s. Figure 31d-31f show that, with load variation, the speed response is good, and the ripples across the speed at standstill are smaller than those in the conventional control design.

Figure 32a shows the *a*-phase current response of the IPMSM drive. The waveform of the *a*-phase current has a sinusoidal shape in the case of the proposed model, which indicates the regular operation of the IPMSM drive. A comparison of the *a*-phase current shows that the designed algorithm has a smaller current



(a) Motor phase current

(b) Torque response

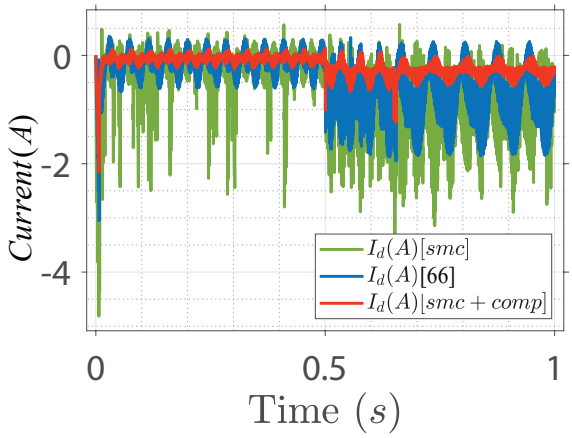
Figure 32: IPMSM current and torque response under nominal parameters with load step variation.

ripple, and the harmonic distortion is suppressed, ensuring the efficacy of the design model.

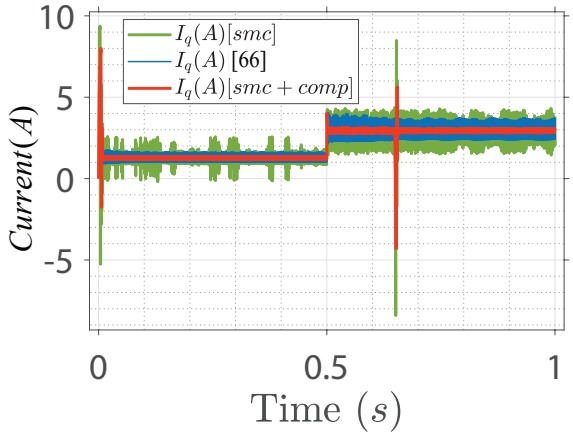
Figure 32b shows the response of the reference torque and the output electromagnetic torque across the drive. By employing the MTPA control scheme, the torque was obtained by utilizing the minimum current value. It can be seen that T_e has a higher ripple in the conventional method, whereas ripples are reduced in the proposed control structure design.

Figure 33 shows the drive phase current along the d and q-axis. Based on conventional methods, the d and q-axis currents show higher current ripples, whereas the proposed control design shows minimum ripples across the d and q-axis current. To verify the robustness of the proposed control design, the speed convergence characteristic of the IPMSM was simulated based on a sinusoidal reference speed input with an amplitude of 10 and a frequency of 30 Hz.

Figure 34 displays sinusoidal speed response and error curves achieved through different control designs. The proposed design exhibits a remarkably converged actual speed to the reference speed, with minimal error, indicating the effectiveness in reducing steady-state errors when employing this control method. Furthermore, the controller demonstrates exceptional dynamic performance,

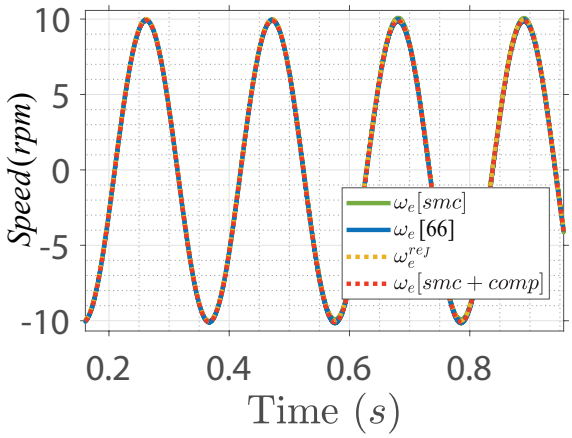


(a) D-axis current

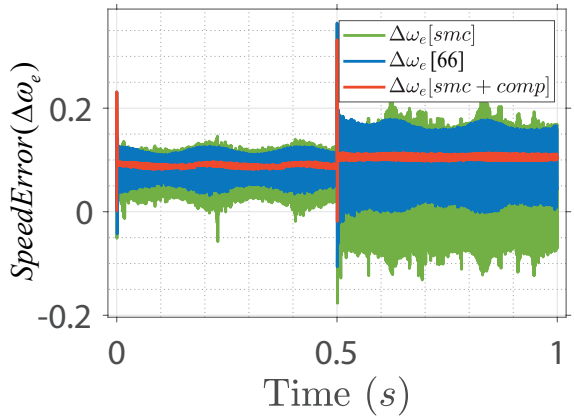


(b) Q-axis current

Figure 33: Reference output state's variable.



(a) Sinusoidal speed response



(b) Speed error

Figure 34: IPMSM drive speed response under sinusoidal speed input.

Table 9: Comparative Performance of conventional and proposed control scheme.

Parameters	**Conventional Control**	**Proposed Control**
Speed Tracking	Good	Excellent
Speed Steady-state Error	Moderate	Minimum
Speed overshoot $(\%)$	high	less
Speed settling time (ms)	high	less
Speed ripple $(\%)$	0.02	0.001
Torque ripple	Higher	lower
Current harmonics	Higher	Lower
Chattering Analysis	severe chattering	minimal chattering
Control efficiency	Lower	Higher

enhancing system robustness by effectively countering load disturbances. As a result, the overall performance of IPMSM speed control is improved, especially under uncertain conditions and load variations. The comparison of performance between the conventional control and the proposed method is summarized in Table 9.

The conclusion of this study introduces a compensation technique with a sliding-mode speed controller, aiming to mitigate torque ripples and improve the overall performance of the control system. This method offers simplicity and does not incur additional computational complexity. By adopting the sliding-mode control technique, the speed controller becomes independent of specific parameters. The compensation technique effectively reduces torque ripples, which are known to negatively impact control system performance, particularly at low speeds.

In the proposed approach, compensation signals are incorporated into the reference signal, facilitating the elimination of undesired ripples within the control system. The study thoroughly discussed the main sources of torque ripple, the structure of the speed controller, and the compensation technique's role in

minimizing ripples within the control system. Simulation results confirmed the feasibility of the control structure, showcasing its efficacy.

A comparative study against the conventional control method was performed, demonstrating that the proposed design substantially reduces speed ripples during standstill and maintains superior speed performance even when faced with external disturbances. This validates the effectiveness of the proposed compensation technique and sliding-mode speed controller in achieving smoother and more robust control of the system.

VI. AI-Driven Control Strategies

The focus of researchers on the application of artificial intelligence in field of electric motor drive has boomed recently as the neural network has the potential to show fast dynamic response and excellent performance compared to other control schemes. Various identification strategies, such as fuzzy logic, reinforcement learning, and local linear regression, have been proposed in the literature. Local linearization (local linear regression) and complex calculation are common downsides of these approaches [70], [71]. Recently, artificial intelligence (AI) has been implemented to solve complicated tasks such as artificial neural network (ANN)-based current control, AI-based motor parameter identification, ANN-torque observers, and ANN-based speed sensorless control [72]–[74].

The handling of non-linear, time-varying, and uncertain systems, which are difficult for conventional control techniques, has been proven for neural network control. However, they also have their drawbacks, such as the requirement for significant processing resources, sample inefficiency during training, and potential stability problems. Despite these challenges, AI−based control strategies hold the promise of transforming industries by providing intelligent, adaptive, and autonomous control solutions. our primary focus is on the application of AI−based control algorithms, specifically Neural Network Control, in the context of Permanent Magnet Synchronous Motors (PMSM). This study aims to investigate the effectiveness of neural network control for PMSM systems and compare it to more conventional control strategies. By studying and comparing the strengths and weaknesses of Neural Network Control and traditional control methods, i seek to gain valuable insights into their respective suitability for PMSM control tasks.

Integrating Neural Network Control with PMSM systems holds great promise as it allows us to create intelligent and autonomous control solutions. Neural

networks can learn from data and adapt to dynamic changes in the motor's behavior, making them well−suited for dealing with the inherent nonlinearities and uncertainties in PMSMs. Additionally, the ability of neural networks to approximate complex functions makes them attractive candidates for controlling PMSM systems without requiring precise mathematical models.

A. Feed-Forward Neural Network for Switching Classification

The key idea of this section is to handle the switching pulses of the two-level three-phase (2L3P) voltage source inverter (VSI) with the model-free technique employing artificial intelligence. Model predictive controller with a single prediction horizon is first employed for data generation that is utilized offline in the training phase. The MPC of VSI is considered a classification problem because there are finite switching states that are applied at each instant by reducing a pre-defined cost function for MPC. Based on generated data, the network is trained to employ supervised learning to map the input data to output switching pulses for a given input-output data pair. Based on the model-free neural network, predictive control the system input values are mapped to output during training, and then the network is employed online by eliminating MPC, that will overcome the effect of model mismatch.

1. Mathematical Model of VSI

The MPC approach predicts the future value of phase current for each possible voltage vector with the system's discrete-time model. To determine the necessary conditions for model prediction, the mathematical model of SPMSM is discretized. Over a sample period T_s, an internal discrete-time model of the motor drive was used to anticipate the future state of the output state variable for input control. In order to identify the best output voltage vector, MPC examined

the 2L3P VSI model. The output of the inverter by modulating input dc voltage through switching pulses is expressed as:

$$S = \frac{2}{3}\left[S_1 + S_3 e^{j\frac{2\pi}{3}} + S_5 e^{j\frac{4\pi}{3}}\right], \tag{48}$$

where S_1, S_3, and S_5 are the switching functions of a, b, and c phase, respectively. Their values are either 0 or 1. The three-phase voltage induced by the VSI is then transformed into the synchronous dq frame by employing Park's transformation as (49), where $\theta_r(k)$ and V_{dc} denote the rotor position angle and DC link voltage, respectively.

$$\begin{bmatrix} v_d(k) \\ v_q(k) \\ v_0(k) \end{bmatrix} = \frac{2V_{dc}}{9} \begin{bmatrix} \cos\theta_r(k) & \cos(\theta_r(k)-\frac{2\pi}{3}) & \cos(\theta_r(k)+\frac{2\pi}{3}) \\ \sin\theta_r(k) & \sin(\theta_r(k)-\frac{2\pi}{3}) & \sin(\theta_r(k)+\frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} S_1 \\ S_3 \\ S_5 \end{bmatrix}, \tag{49}$$

2L3P VSI has eight possible voltage vectors with zero voltage vectors, V_0 and V_7, while the rest $V_1, \cdots, V_6$ are active voltage vectors. One of the zero vectors is considered to reduce the computational burden of MPC. The discrete-time model of SPMSM is derived with the Eular approximation and given as:

$$i_d(k+1) = i_d(k) - \frac{T_s}{L_s}[-r_s i_d(k) + L_s \omega_r(k) i_q(k) + v_d(k)], \tag{50}$$

$$i_q(k+1) = i_q(k) - \frac{T_s}{L_s}[-r_s i_q(k) - L_s \omega_r(k) i_d(k) - \lambda_m \omega_r(k) + v_q(k)], \tag{51}$$

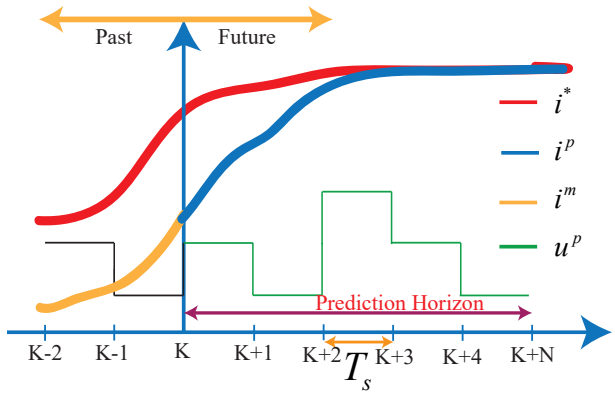


Figure 35: Visualization of mpc scheme.

where L_s, r_s, and λ_m are the stator inductance, stator resistance, and rotor magnetic flux, respectively. $\omega_r(k)$ is the rotor angular velocity, i.e., $\omega_r(k) = (\theta_r(k) - \theta_r(k-1))/T_s$.

2. Model-Based Predictive Current Control

MPC approaches have lately been seen in a variety of industrial applications due to their rapid dynamic response and efficient control performance. A schematic representation of mpc is shown in Figure 35 An important part of MPC is the selection of cost function to determine the optimal output voltage vector. In the current control mechanism, the inverter keeps the most important consideration of keep tracking the reference currents to measure current accurately. The phase currents are directly controlled in FCS-MPCC, and the cost function h is selected as

$$h = [i_d^*(k) - i_d(k+1)]^2 + [i_q^*(k) - i_q(k+1)]^2, \tag{52}$$

where $i_d^*(k)$ and $i_q^*(k)$ are the d and q axis current reference values,$i_d(k+1)$ and $i_q(k+1)$ are the predicted current values, respectively.

MPCC has a simple principle and is easy to implement but has a significant

drawback of requiring a huge number of online calculations in order to solve optimization problem which increases the computational complexity of the control system. Moreover, it is highly dependent on the model of the system. To address these problems, this paper presents a feed-forward ANN-based controller for optimal current control performance by utilizing the flexibility of MPC for data generation for training a network offline.

3. Proposed Model-free Predictive Current Control

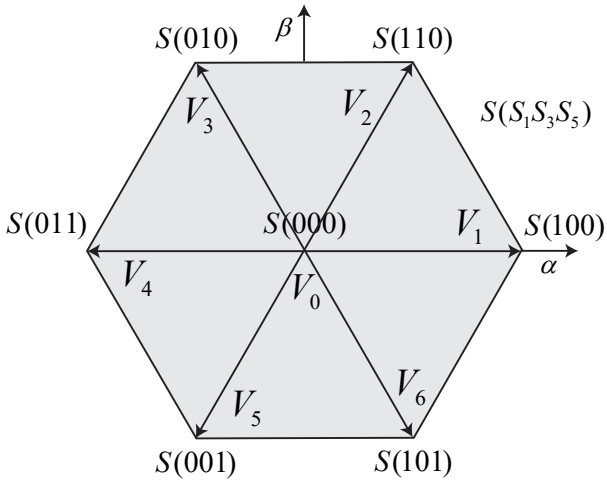
Various data-driven techniques, especially AI-based control designs, are emerging in the field of the power conversion system. A well-designed ANN can significantly increase the performance of the learning system. Nodes, weight, and Layers are the component for the formation of various ANN models. The general expression to define a neuron in NN is as follows:

$$N = \left(\sum_{i=1}^{k} w_i x_i \right) + b, \tag{53}$$

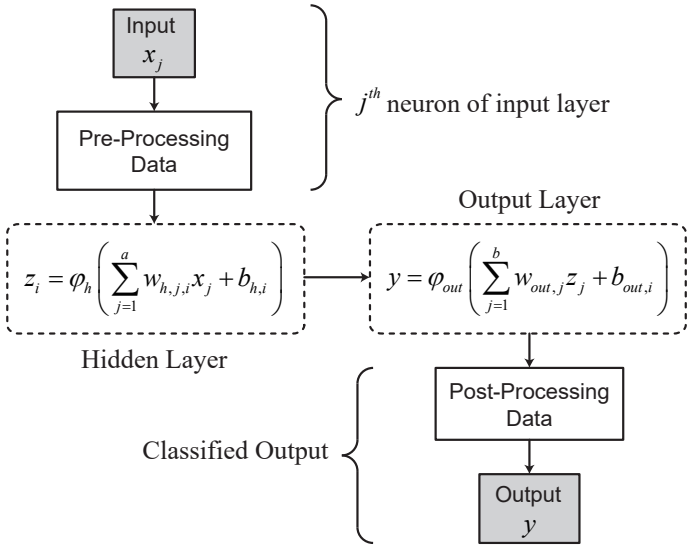
where k is the number of elements, x_i is the input vector, w_i is the interconnected weight, and b is the bias factor. N is processes utilizing activation function that is usually a nonlinear function to ensure universal approximation property [75] and is given as

$$f(N) = f[\left(\sum_{i=1}^{k} w_i x_i \right) + b]. \tag{54}$$

Considering the fixed number of switching states for VSI, the states can be considered as classes. As a result, the challenge of optimal voltage vector can be considered as a multi-class classification problem at each time step, with the optimum switching states considered as the patterns to be learned using a FNN. In this paper, a feedforward NN (FNN) with seven input neurons is structured which are responsible for mapping the output of the network. A neural network



(a) Possible combinations of switching pulses.



(b) FNN architecture.

Figure 36: Three-layer FNN with measured and reference currents ($I_{\alpha\beta}^{[k,k-1]}$) and switching states ($S_{1,3,5}^{k-1}$) as inputs, whereas the output constitutes optimal voltage vector V_i.

Table 10: Inverter switching configuration.

Class	Phase Voltage			Voltage Vector	Switching States		
C	V_{as}	V_{bs}	V_{cs}	V_i	S_1	S_3	S_5
1	0	0	0	v_0	0	0	0
2	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	v_1	1	0	0
3	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	v_2	1	1	0
4	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	v_3	0	1	0
5	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	v_4	0	1	1
6	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	v_5	0	0	1
7	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	v_6	1	0	1

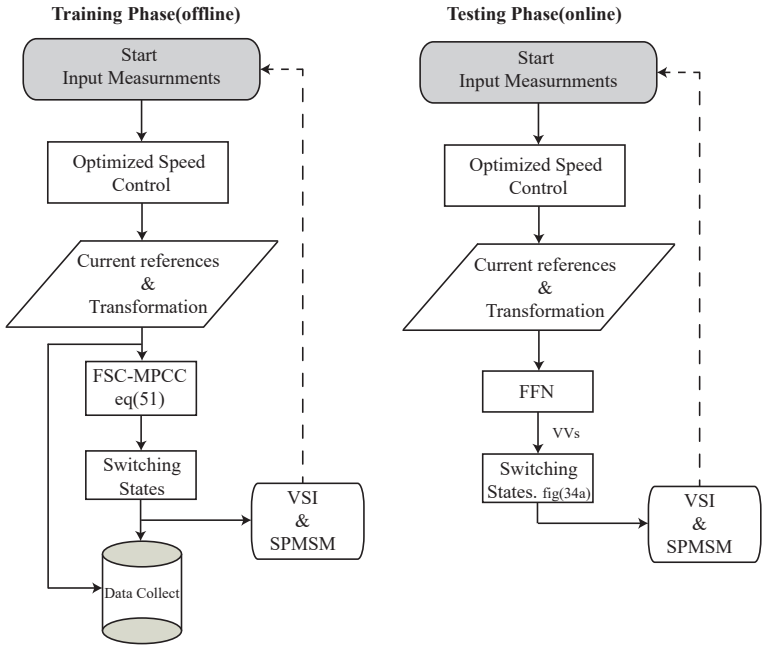


Figure 37: An overview of proposed design.

is designed for specifying the switching states of the inverter as depicted in Fig. 36a. All the inputs are separated layers without links between the layers, and the information is processed by one-way connection. Model-based predictive current control provides optimal voltage vectors based on minimizing the cost function to track the reference trajectories by providing finite switching pulses. As the set of switching states are finite for each optimal voltage vector, i considered predicting switching states of the two-level three-phase inverter as a classification problem. A large number of voltage vectors results to more precise gridding in stationary reference frame, which can significantly enhance the learning algorithm prediction accuracy. By employing smaller grid structure, the algorithm will perform better and provide more accurate predictions which will lead to accuracy improvement of the trained network. The reference and measured currents obtained from the FCS-MPCC in rotating reference frame is transformed into stationary reference frame by utilizing inverse Park′s transformation as follow:

$$\begin{cases} i_\alpha = i_d \cos(\theta_r) - i_q \sin(\theta_r) \\ i_\beta = i_d \sin(\theta_r) + i_q \cos(\theta_r) \end{cases} \tag{55}$$

The currents in stationary plane are utilized for training data to enhance the prediction accuracy of the network. There are seven neurons in the input layer, where four input neuron represents the error between measured and reference phase current in the stationary reference frame at the previous and present time instant $[k-1, k]$ as given in (56), while the other three represent the optimal gate pulses for the previous instant $(k-1)$. The representation of FNN is shown in Fig.36b, where a is the total input connections, x_j is the inputs, $w_{h,j}$ is the weight, $b_{h,j}$ is the bias of the hidden layer, and φ_h is the hidden layer activation function.

$$\begin{cases} \Delta I_{\alpha\beta}^{[k]} = I_{\alpha\beta}^{[k]} - I_{\alpha\beta}^{*[k]} \\ \Delta I_{\alpha\beta}^{[k-1]} = I_{\alpha\beta}^{[k-1]} - I_{\alpha\beta}^{*[k-1]} \end{cases} \tag{56}$$

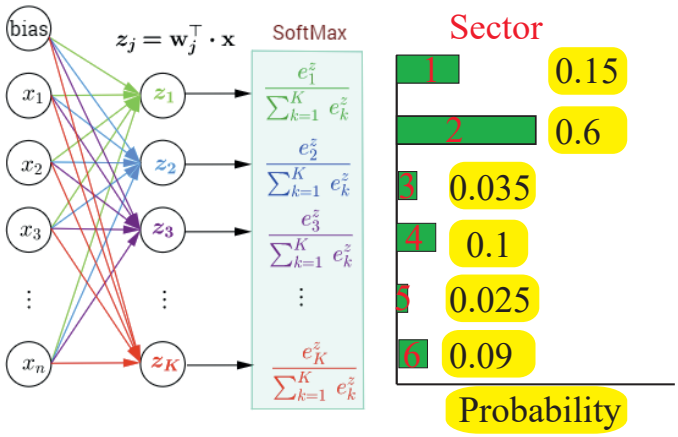


Figure 38: Visualization of Softmax output in a trained network.

The activation function (φ) is the core of ANN and, based on the layer locations either input, hidden or output layer different activation functions have proven beneficial. In this paper, hyperbolic-tangent (tanh) function is used as an activation function with 20 neurons in the hidden layer. At the output layer, the softmax function is used as an activation function which gives the probability between 0 and 1. Moreover, the sum of probability will be 1 at each instant. The illustration of working mechanism of softmax activation function at output not is given in Figure 38 The voltage vector across output with maximum probability will be active for each sampling instant. The flow chart of the control design is depicted in Figure 37.

4. Results and Discussion

The training is conducted offline, using MPC as the baseline model. For performance analysis, an optimized speed controller, employing a meta-heuristic technique in combination with the proposed model-free predictive current controller, is employed. The trained neural network is employed online to acquire optimal voltage vector using the proposed and conventional scheme via laptop with an Intel® Core i7-1195G7 2.90GHz CPU, 16 GB RAM and NVIDIA

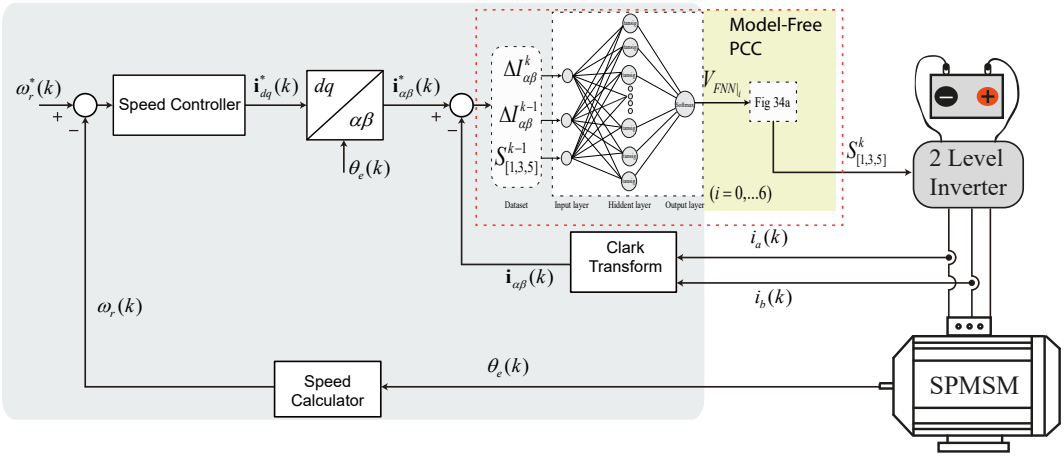


Figure 39: Model-free current controller of an SPMSM fed by 2L3P VSI.

Table 11: Performance analysis of trained NN under different number of hidden layers and test cases.

Training Test Cases		Observation	Number of Hidden Layer	Performance error	Accuracy	Training Algorithm
30	NN	150,000,30	10	0.1135	67%	SCG
			20	0.0996	71.8%	
			30	0.1035	70.1%	
45	NN	225,000,45	10	0.0562	86%	SCG
			20	0.0340	90.4%	
			30	0.0440	87.6%	

Geforce® RTX laptop GPU 3050, and running window 10 64 bit.

Training Procedure:

The training process consists of three main steps:

a) Data Generation employing MPC as the baseline model for both training and testing

b) Post and pre-processing of the generated data for the training and testing phase utilized offline and online in the control system

c) Setting up FNN hyper-parameters

The measured and reference variables of the phase current in the stationary reference frame and one-step delay switching signals are fed into the FNN. A total of seven signals are taken as input features to the neural network, i.e., k = 7. The optimal voltage vector $V_{FNN}|_i (i = 0, ...6)$ to be applied at each sampling moment is the FNN output. The softmax activation function at the output layer squashes

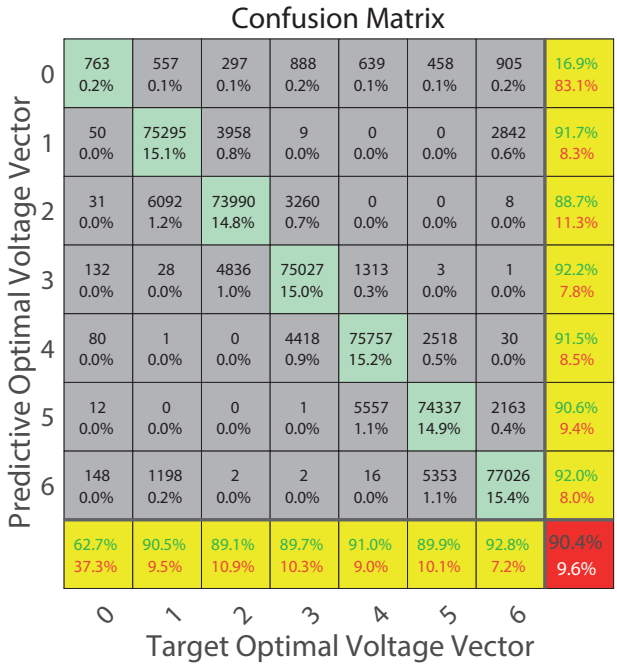


Figure 40: Confusion matrix for multi-class classification.

the output to be between 0 and 1, which tells the optimal VV to be applied at each sampling period Ts. Moreover, the output layer is a seven-dimensional array that contains the indexes of the seven different voltage vectors Vi that the inverter generates. NN Output is one-hot encoding, which means that for each sampling time period, only the index of the optimal VV will be active(i.e., having higher probability among all the classes as given in Table 10). The dataset and the number of hidden layers are selected based on higher training accuracy as given in Table 11. The dataset has been divided into three parts: 70% of the data is randomly selected for training, and 15% has been randomly selected for testing and validation, respectively. The training was done utilizing a scaled conjugate gradient scheme, which exploits the good convergence property of conjugate gradient optimization and offers less computational complexity. To analyze the prediction accuracy of the proposed model-free PCC, the confusion matrix is shown in Fig 40. The overall prediction accuracy of the trained network is 90.4%,

which shows how well the network is predicting the classes as given in Table 10.

Test Results:

To assess the efficacy of the proposed model-free current control, I conducted an evaluation of the dynamic performance of SPMSM. I employed both conventional and proposed control designs, utilizing the parameters detailed in Table 12, across a range of operating conditions. This comprehensive assessment provides valuable insights into the control system's performance.

*Case*1 : Speed response at no Load condition

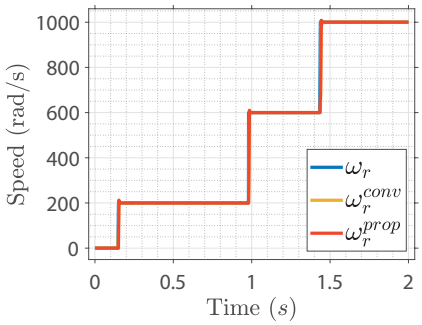
In the first scenario, I investigated the machine's speed response in the absence of any load employing proposed and conventional control design. The motor was made to experience step variations in the reference speed. The Model-free Neural Network based predictive current control was effective than that of conventional control technique as shown in Figure 41. The AI-based controllers responded to the changes in the reference speed promptly, with minimum overshoot or settling time. The data-driven control solution excels in current tracking performance, surpassing traditional current controllers. This superior performance underscores its potential for enhancing motor drive systems.

*Case*2 : Variable Load Conditions and a Fixed Reference Speed

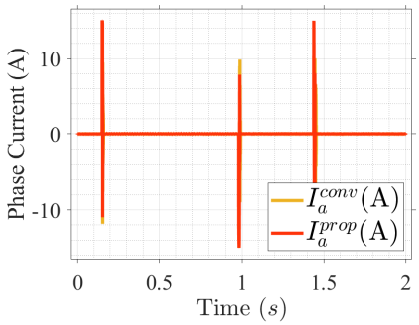
In the second test case, the effectiveness of the AI-driven controllers with a fixed reference speed of $800 rad/s$ was assessed under various load scenarios.

Table 12: Motor Parameters.

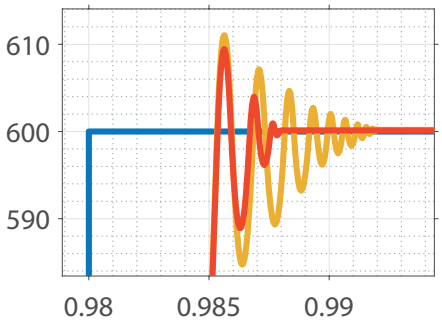
Parameter	**Description**	**Value**
T_s	Sampling time	$1\ \mu$s
r_s	Stator resistance	0.2Ω
L_s	Stator inductance	$0.00835H$
λ_m	Flux linkage	$0.175Wb$
P	Pole pairs	4
J	Inertia	$0.0008\ kg.m^2$



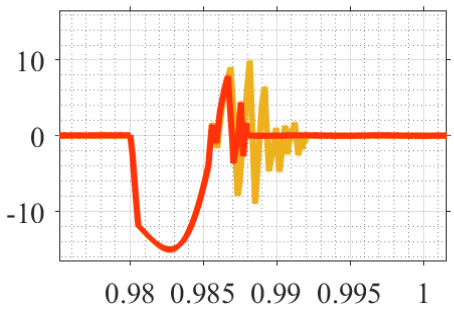
(a) Speed step response



(b) Motor phase current

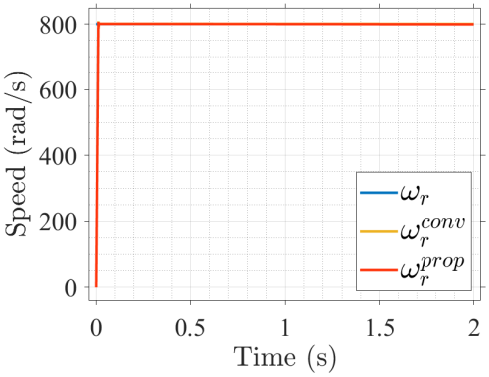


(c) Expanded result of 41a

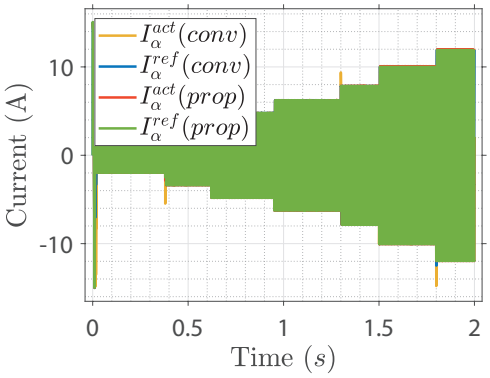


(d) Expanded result of 41b

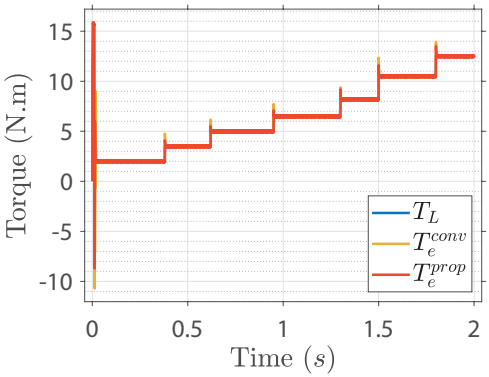
Figure 41: Motor speed response in the wide speed operating range under no load condition.



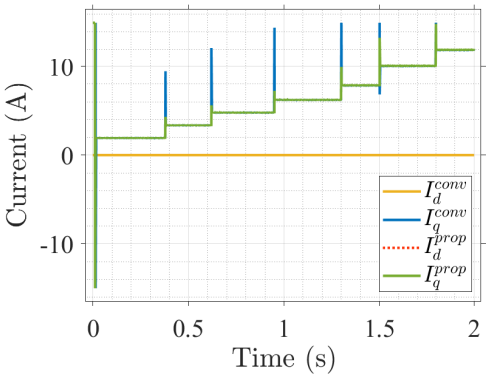
(a) Speed convergence under varied load torque



(b) Phase current in stationary reference frame



(c) Electromagnetic torque response



(d) dq-axis current response

Figure 42: Motor dynamic response under varied load conditions at 800 rad/s.

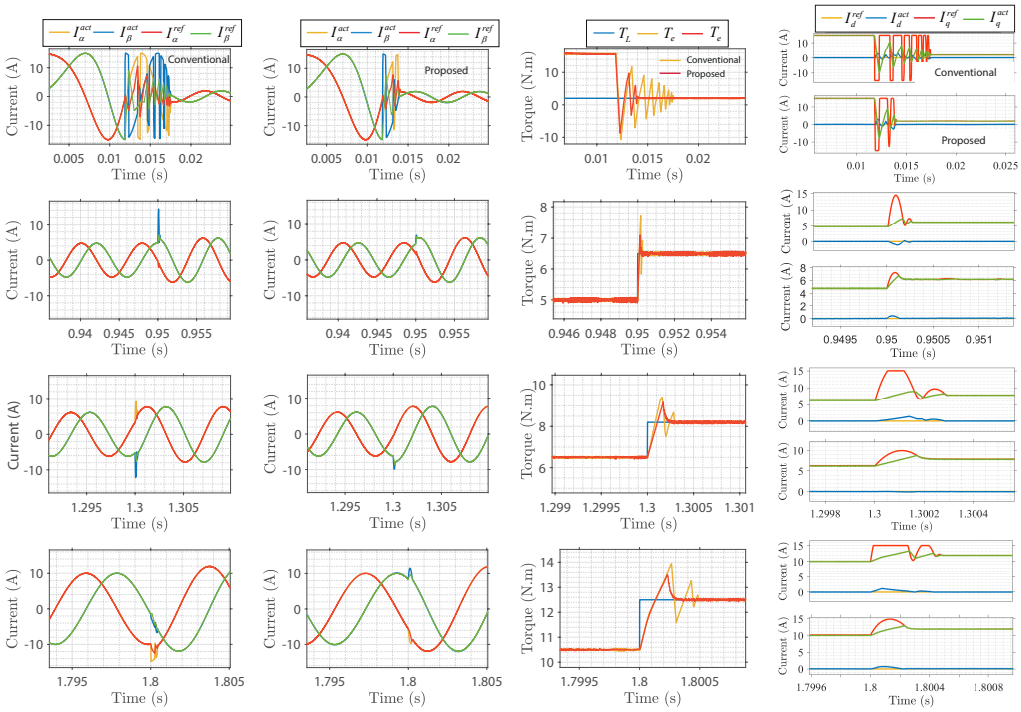
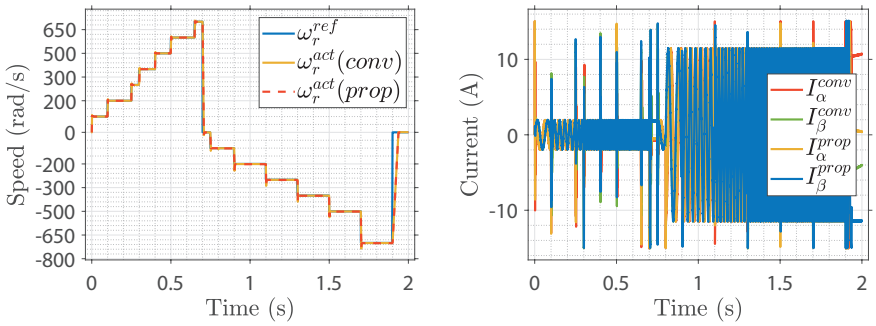


Figure 43: Motor dynamic response under varied load conditions.

I changed the load torque in steps while maintaining the reference speed as illustrated in Figure 42a. The AI-based model-free predictive current controller shown impressive resilience and consistency in maintaining the required speed. Traditional control technique, on the other hand, had trouble adapting to changes in load, had longer response times, and had higher overshot and settling time. Figure 43 demonstrating the dynamic motor drive performance at different load torque stages when zoomed in. The efficiency of the model-free control strategy is demonstrated by the motor driving phase current in stationary and rotating reference frames. The suggested design significantly outperforms the standard method in terms of transient responsiveness, overshoot, settling time, and tracking rate.

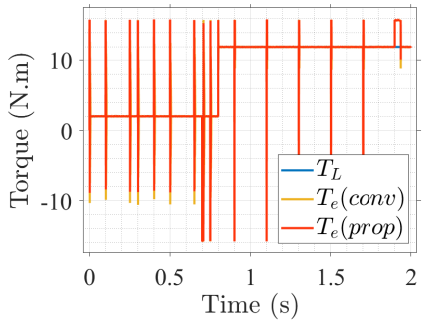
*Case*3 : Variable Speed at constant step Load.

In the third scenario, i looked into how well the AI−based controllers



(a) Speed convergence under varied load torque

(b) Phase current in stationary reference frame

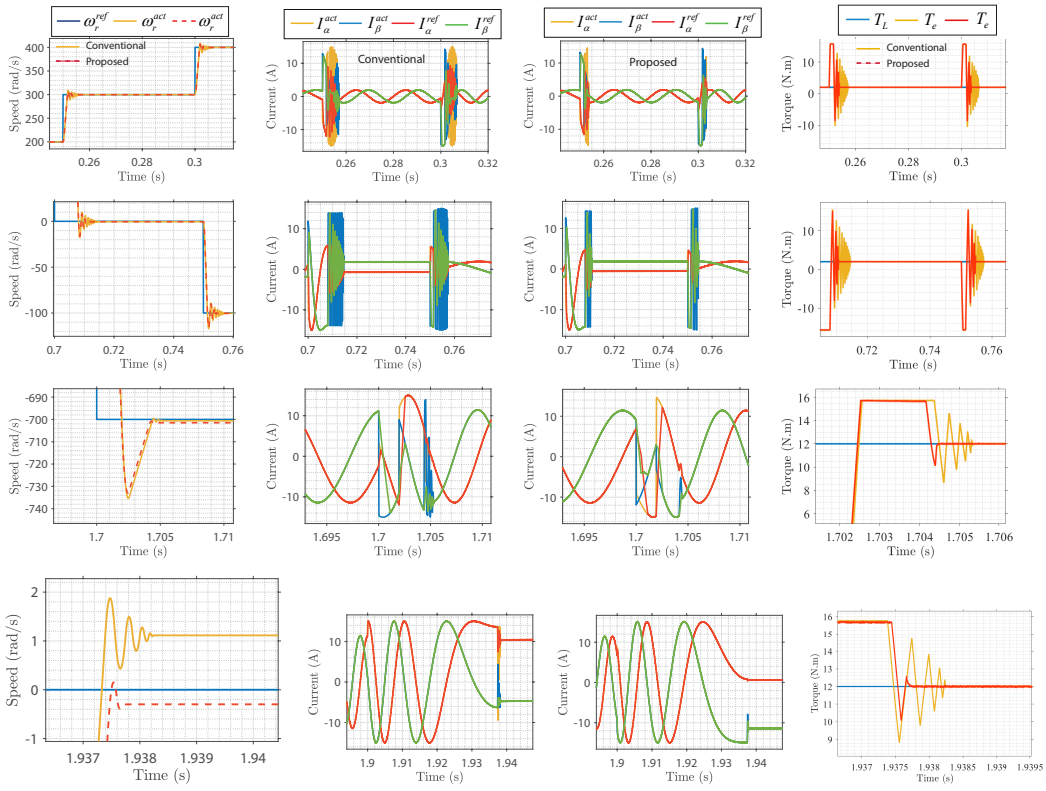


(c) Electromagnetic torque response

Figure 44: Dynamic performance at wide speed operation range under load conditions.

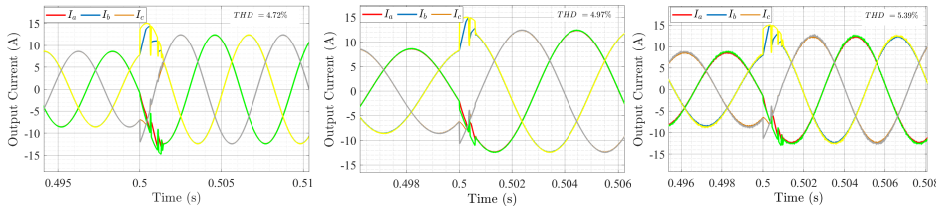
could manage a range of speed profiles under constant load conditions as shown in Figure 44a. The motor were exposed to various speed changes, and the control algorithms based on AI showed good monitoring of the speed profiles. With smoother speed profiles compared to conventional model-based predictive current control, the Neural Network Control technique demonstrated higher tracking accuracy. Furthermore, proposed control methods outperformed conventional algorithm in terms of transient response and showed minimum steady-state error and settling time as it can be seen in Figure 45.

Figure 46 illustrates the evaluation of performance under normal conditions, parameter mismatch scenarios, and variations in sampling time. It can be seen that that the Total Harmonic Distortion (THD) of output currents in the

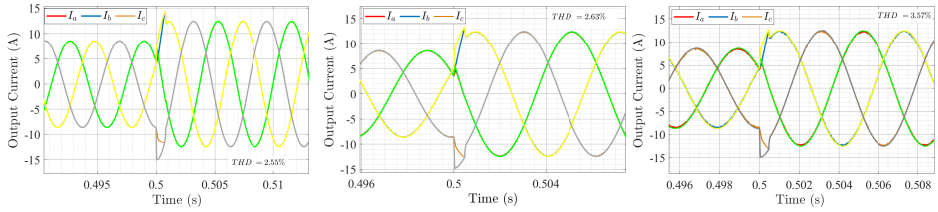


(a) Motor speed response (b) Current convergence in conventional model based control design (c) Current convergence in proposed model free control design (d) Electromagnetic torque response

Figure 45: Expanded result of Figure 44, wide speed operating range under load torque.



(a) Conventional MPCC technique



(b) Proposed Data-driven solution

Figure 46: Performance evaluation of proposed and conventional control approaches in steady-state and transient-state under normal, parameter mismatch ($L_s = 5.5mH$) and variation in sample time conditions(left to right).

state-of-the-art Model Predictive Current Control (MPCC) stands at 4.72%, 4.97%, and 5.39%, respectively. In contrast, the proposed control scheme achieves significantly lower THD values of 2.55%, 2.63%, and 3.57% for the same scenarios. Furthermore, the tracking error in the proposed control design outperforms the conventional MPCC scheme, demonstrating superior control performance. Furthermore, the given results clearly demonstrates exceptional control and tracking performance even when subjected to parameter mismatches and unknown perturbations. Additionally, it maintains a high dynamic transient similar to the standard MPCC approach.

Table 13 presents a quantitative comparison of the proposed AI-based architecture with the traditional control approach. The table lists many performance indicators that were determined under varied operational conditions. Finally, the quantitative analysis shown in Table 13 highlights the enormous benefits of the proposed AI-based control over the conventional model−based predictive current control. The AI-based technique is an excellent choice for

Table 13: Comparative analysis of model-based and model-free control design.

Parameters	**Conventional**	**Proposed**
Dynamics	Average	Excellent
Steady-state Error	Moderate	Minimum
Speed Ripples	Larger	Moderate
Torque Response	Slow	fast
Control Efficiency	Average	Excellent
Computational Complexity	Higher	Moderate

creating effective and intelligent machine control in industrial applications because it excels at speed tracking, flexibility to load and speed fluctuations, and show good performance in wide operation range.

B. Deep Symbolic Regression (DSR)

In this study, deep symbolic regression (DSR) is used to provide a novel technique of current regulation in a surface-mounted permanent magnet synchronous machine (SPMSM). The proposed DSR−based optimum current control technique has the ability to estimate unknown data and is capable of producing an analytical dynamic numerical expression through training and fitting. Since the performance of the conventional linear proportional-integral (PI) current controller significantly depends on the correctness of the system model, the primary goal of this work is to overcome its limitations. To validate the efficacy of the proposed design, various test cases are applied, which included simulations and experimental setups, involving different speed and load conditions. The obtained results are then compared with those of a traditional vector control design. The findings suggest that the DSR-based control

design outperforms traditional approaches and exhibits the ability to extrapolate beyond the training dataset. This breakthrough showcases the potential of deep learning techniques in power conversion applications, paving the way for further advancements in the field.

Various studies in the field of power electronics have shown promising results using neural networks (NN) [76]–[78]. However, it is important to acknowledge its limitations. Neural network architectures are often challenging to comprehend and interpret, resembling black boxes that lack transparency. Although data-driven techniques can accurately fit system dynamics, they provide black box models with low interpretability.

Surprisingly, despite its potential benefits, deep symbolic regression remains relatively underexplored, especially in power conversion applications. The authors note that, to the best of their knowledge, deep symbolic regression has not been previously employed for designing numerical models or model-based control laws. As an innovative approach, the study harnesses deep symbolic regression as the state-of-the-art method to construct the proposed optimal current control design based on an analytical model. This unique application holds the promise of overcoming the limitations of black box models and offers the potential for more interpretable and insightful control solutions.

1. Proposed Current Controller

In the realm of artificial intelligence, the challenge of finding a manageable numerical expression that best describes a dataset is a long-standing problem known as symbolic regression. Traditionally, this task has been tackled using genetic programming, which employs metaheuristic algorithms to fit the data with a population of numerical expressions, extracting underlying laws of physical systems. However, genetic programming faces challenges with scalability for large systems and is prone to overfitting.

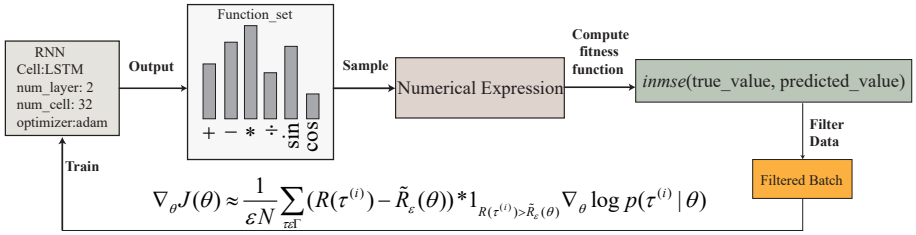
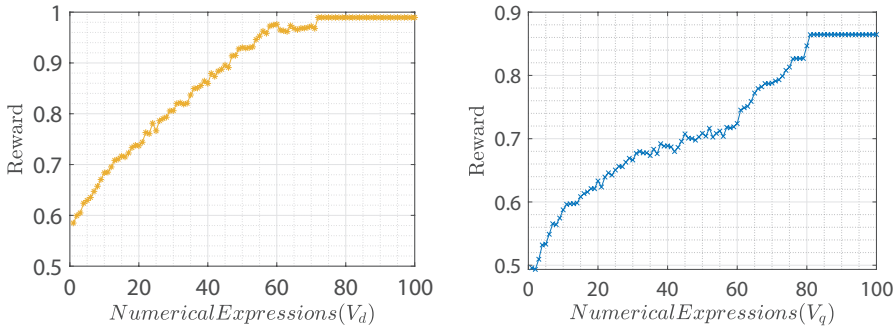


Figure 47: Deep symbolic regression core architecture.

To address these limitations, deep symbolic regression (DSR) emerges as a promising method for symbolic regression, leveraging policy gradients from deep learning. DSR has demonstrated remarkable results in recovering symbolic equations compared to genetic programming in various test cases [79]. In this study, DSR is applied to design an optimal current controller. The algorithm takes advantage of a deep neural network to generate interpretable and generalized models. The core architecture of DSR involves a recurrent neural network (RNN) that outputs a distribution of numerical expressions. These expressions are randomly chosen from the distribution, instantiated, and assessed to fit the dataset. Figure 47 illustrate the pictorial representation of the core algorithm. The training was performed offline using python, and inverse normalized mean square error bounded by sigmoid activation function is selected as a decision function. The fitness serves as a reward signal to train the RNN using a unique risk-seeking policy gradient approach. As training progresses, the RNN adjusts the probabilities of numeric expressions based on their rewards, giving higher probabilities to better-performing expressions. The final expressions with a maximum reward for the v_d command reference and for the v_q command reference were obtained and is given as follows:

$$v_d^* = -x_1(x_1 - x_4) + 11x_1 - x_4 - sin(x_2) + sin(x_1 + x_2),$$
$$v_q^* = -x_1x_2 + 12x_2 - x_3x_4 - x_3 + x_4(-2x_2(-2x_2 + x_3 - x_4) + x_4) - 2x_4 + cos(x_1 - 3x_3 + x_4) \quad (57)$$



(a) V_d rewards. (b) V_q rewards.

Figure 48: Rewards for generated numerical expression.

where $x_1 = e_d$, $x_2 = e_q$, $x_3 = \int e_d$, and $x_4 = \int e_q$, respectively.

The control scheme of the proposed DSR-based current controller is described as follows:

(a) Measure the current errors and their integration in the synchronous reference frame at sampling time T_s.

(b) The integral error information is fed into the system, ensuring that there is no steady-state error in the reference tracking.

(c) The DSR algorithm employing the risk-seeking policy gradient generates numerical expressions that are easy to understand and fit the data.

(d) The generated expressions are employed in an online model as an optimal current controller.

2. Test Setup

To evaluate the effectiveness of the DSR-based optimal current control technique and compare it with the traditional current control, I implemented the models using Python 3.6 and Matlab Simulink (2023). Figure 56 and Figure 49illustrates the setup for the implementation.

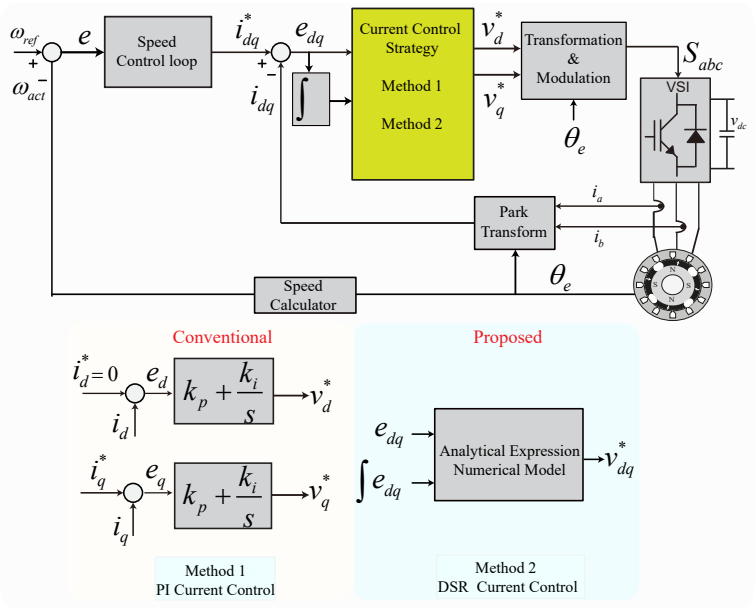


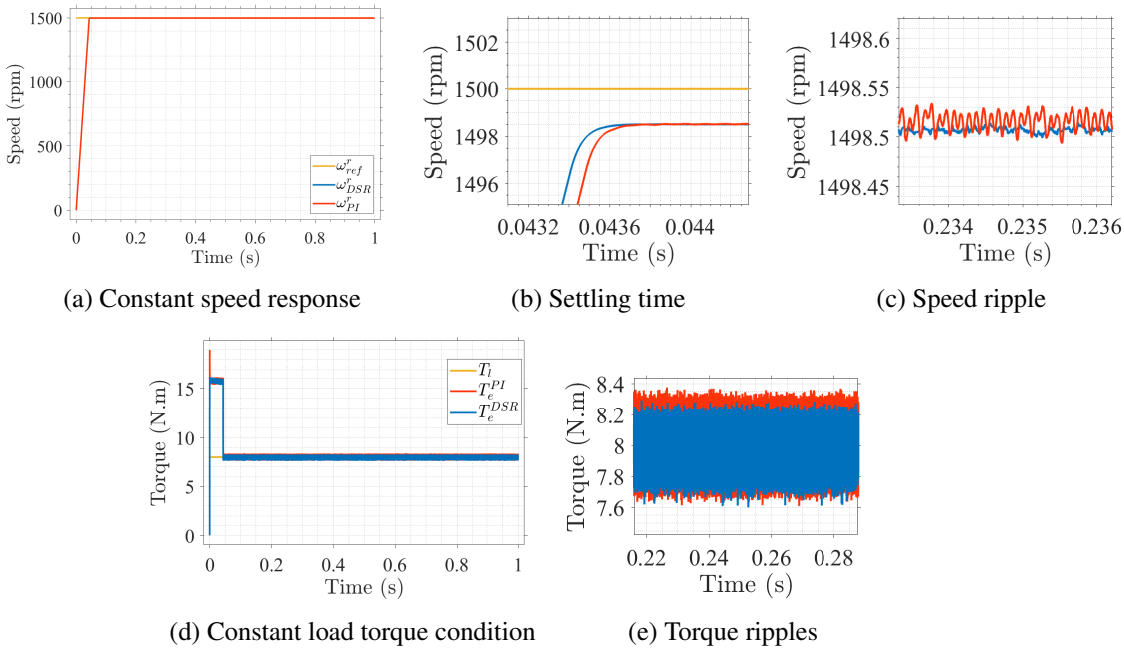
Figure 49: Conventional and proposed field-oriented control architecture of SPMSM.

The training process was conducted offline on the generated datasets. The mathematical model derived from DSR was then applied online for real-time performance. I executed the model on a laptop with an Intel® Core i7-1195G7 2.90 GHz CPU, 16 GB RAM, and NVIDIA GeForce® RTX laptop GPU 3050, running Windows 10 64-bit operating system.

3. Test Results

By conducting these experiments, i aimed to demonstrate the superiority of the proposed DSR-based optimal current control technique over traditional approaches, showcasing its potential benefits for power electronics applications.

The steadystate controller performance under constant speed and torque reference is shown in Figure 50. The results illustrate the motor′s drive response, demonstrating proposed controller ability to rapidly settle at the desired speed while maintaining a consistent load. Notably, the proposed controller showcase commendable performance attributes, evidenced by its remarkably short speed



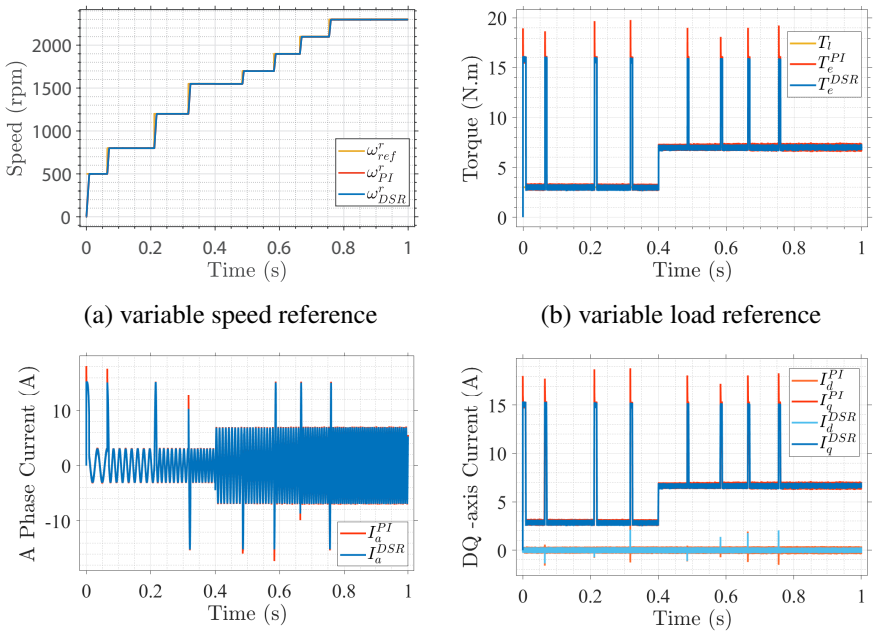
(a) Constant speed response (b) Settling time (c) Speed ripple

(d) Constant load torque condition (e) Torque ripples

Figure 50: Simulation Results – Steadystate Controller Performance: Constant Speed and Torque Responses.

settling time. Furthurmore, the result demonstrate minimal fluctuations or ripples in the motor′s output, highlighting the effectiveness of the proposed control strategy in achieving smooth and stable operation.

The performance parameters of the Permanent Magnet Synchronous Motor under various speed and load conditions are thoroughly analysed in this Figure 51. The test results of the proposed design show excellent standalone and dynamic performance under various working conditions. Moreover, it can be seen that the generated numerical analytical model is well fit to online data. Thus, providing fast response to any change in the control system. Figure 52 provides a comprehensive pictorial representation of the harmonic distortion in current waveforms generated by both the proposed and traditional control strategies for a Permanent Magnet Synchronous Motor (PMSM). The bar graph highlights the distinct levels of harmonic content present in each control approach, emphasizing



(a) variable speed reference

(b) variable load reference

(c) Motor phase A current

(d) dq−current response

Figure 51: Simulation Results − Controller Performance: under Varied Speed and Load Conditions.

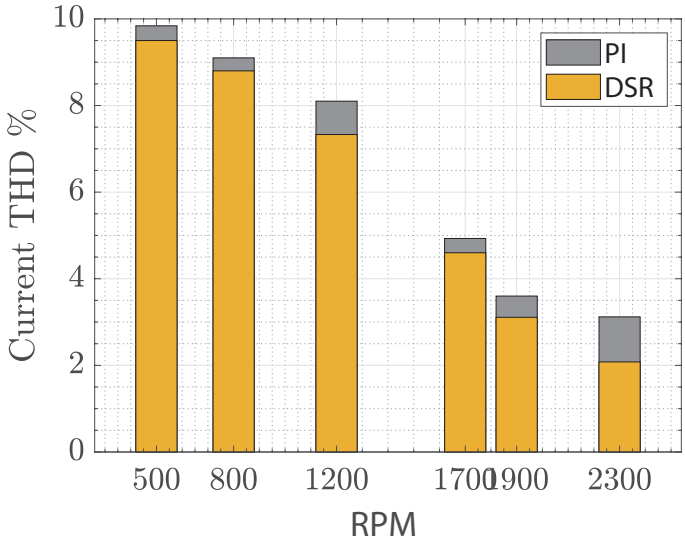
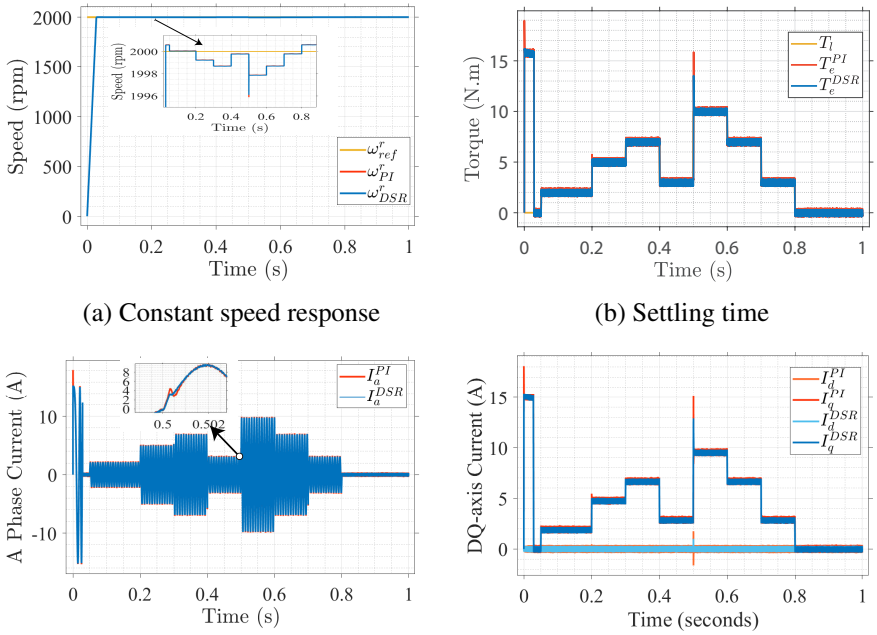


Figure 52: Comparative analysis of current THD between proposed and traditional control method for Permanent Magnet Synchronous Motor.



(a) Constant speed response

(b) Settling time

(c) Speed ripple

(d) Constant load torque condition

Figure 53: Simulation Results – Dynamic load response of SPMSM with Constant Speed.

the potential benefits of the proposed method in mitigating current harmonics. This analysis offers valuable insights into the performance enhancements achieved by the novel control strategy, thus contributing to the advancement of efficient and reliable PMSM drive systems in various industrial and commercial applications.

Figure 53 illustrate the dynamic load response of a Permanent Magnet Synchronous Motor (PMSM) while maintaining a constant speed. The test results depicts the motor′s remarkable adaptability as it navigates through varying load conditions while keeping a steady rotational speed. This analysis underscores the PMSM′s ability to accommodate changing mechanical demands, showcasing its resilience and effectiveness in scenarios where consistent speed is a requirement. The test results demonstrate the performance of the proposed controller shines prominently under varying load conditions, affirming its efficacy and potential in

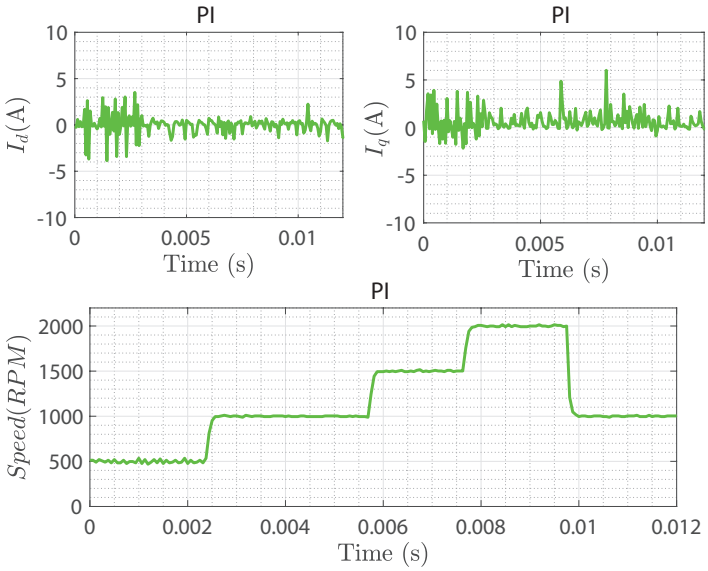


Figure 54: Experimental Result: Motor Response at varied step Speed reference employing PI current Controller

ensuring robust and adaptable motor operation.

Figure 54-55 illustrate the experimental results to regulate the performance of the motor drive in response to a positive step input reference signal with no load condition. During the low speed, there was notable distortion in the current response of traditional current controller, that can be attributed to various factors, including the controller's tuning parameters. This distortion can result in undesired fluctuations and transient behavior in the motor's performance, potentially impacting the overall system's stability and efficiency. However, the proposed control system demonstrated distinct response characteristics. The proposed control system showcased significantly lower levels of current response distortion. This reduction in distortion indicates that the proposed control system effectively mitigates transient behavior, leading to smoother and more stable motor performance following a step input change.

In addition, reduced distortion can lead to enhanced motor efficiency, minimized wear and tear, and improved accuracy in applications where precision

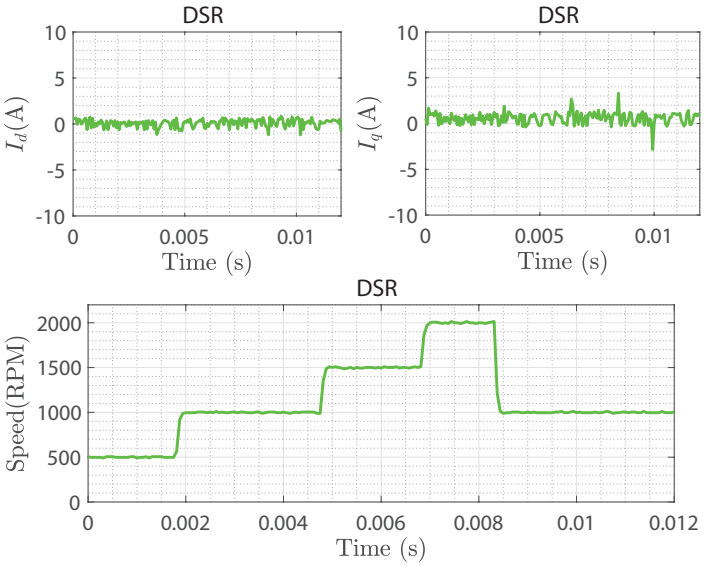


Figure 55: Experimental Result: Motor Response at varied step Speed reference employing DSR current Controller

control is essential. The test results indicate that the proposed control system holds promise for applications where precise and stable motor performance is paramount. This outcome suggests that the application of Artificial Intelligence (AI) in the field of power conversion holds substantial value and potential benefits. Thus, AI-driven research and development in power conversion can lead to breakthroughs in technology and systems, further advancing the field.

VII. CONCLUSION

In this study, i have explored and exploited different distinct control methodologies, for PMSMs. The motor drive has been empowered through the implementation of advanced and intelligent control techniques. At first, this work presents a comprehensive analysis of three optimization techniques, namely Particle Swarm Optimization (PSO), Cuckoo Search, and Jaya Optimization, applied to tune the gains of an Speed Controller for enhancing the performance of Permanent Magnet Synchronous Motors (PMSMs). Through rigorous experimentation and testing, i have demonstrated the efficacy of these optimization algorithms in achieving improved control parameters. The results obtained from the tests provide valuable insights into the performance enhancements achieved by each optimization method. The comparison table offers a concise overview of their respective strengths and limitations, aiding researchers and practitioners in selecting the most suitable optimization technique for their specific applications.

Furthermore, a novel approach is introduced to address speed and torque irregularities during low-speed operation by implementing an efficient compensation technique utilizing a sliding-mode speed controller for interior permanent magnet synchronous motor. This method aims to mitigate undesired speed and torque fluctuations by introducing a scheme that renders the speed controller's parameters independent. Detailed discussion is provided on the factors contributing to torque ripples, the structure of the speed controller, and the compensatory measures for minimizing speed irregularities. The proposed technique offers simplicity without incurring additional computational overhead, making it suitable for drive applications where the presence of low-order harmonics is undesirable. To counteract the chattering associated with sliding-mode control, an exponential reaching law is introduced. This innovation brings about swift convergence and adaptability to switching function variations, thus

reinforce the robustness of the speed controller. The efficacy of this control approach is substantiated through comprehensive simulations conducted under diverse conditions, including variable low-speed scenarios, fluctuating loads, and sinusoidal speed reference inputs. A comparative analysis is carried out against conventional control methods, revealing that the proposed controller significantly improves the stable−state and dynamic performance of IPMSM, particularly in low−speed domains. Notable benefits include reduced speed oscillations, diminished steady-state error, and accelerated transient responses.

Finally, this thesis has explored and showcased the integration of AI driven control strategies for Permanent Magnet Synchronous Motors (PMSMs). Through a meticulous investigation, i have demonstrated the effectiveness and potential of utilizing AI techniques to enhance the performance of PMSMs in various operational scenarios. At first, an improved model−free strategy for current control, leveraging a neural network for surface mount Permanent Magnet Synchronous Motor (SPMSM) drives. The effectiveness of this approach has been substantiated through simulations encompassing diverse load and speed scenarios. The neural network is initially pre−trained offline using traditional Model Predictive Control (MPC) as a reference, constructing comprehensive input−output datasets. Subsequently, this well-trained network is deployed in online replacing traditional linear current control as an Artificial Neural Network (ANN)−based current controller, showcasing notable proficiency in both steady−state and dynamic performance. Furthermore, the network's training process employs a minimized set of input data signals, effectively reducing the computational overhead during training and increase the performance efficacy to 90% as compared to previously studied AI control designs. At the end of this thesis, A novel approach to control design was employed, merging deep learning with symbolic regression to proposed a non−linear controller. Deep Symbolic Regression based non−linear controller effectively countered

the constraints inherent in traditional linear control techniques. These methods often encountered difficulties due to the difficult task of tuning gain parameters, which carried notable significance. Furthermore, the difficulty arises from parameter dependency frequently introduces hurdles, impacting the efficacy of linear control strategies. The strategic incorporation of optimal gain parameters played a pivotal role in driving dynamic and excellent performance within the framework of a closed-loop linear control system The employment of DSR to generate a numerical expression for voltage commands represents a cutting−edge advancement in motor drive systems. This achievement signifies a pivotal milestone in the application of deep learning techniques within power energy conversion domains.

A. Future Work

- I intend to study multi−objective optimization for tuning the PI controller gains for cascaded control design. Moreover, the performance of these algorithm under parametric variation and drive uncertainties will be studied.A statistical analysis will be incorporated in the same way to determine the significance of the results.

- The application of DSR based current controller will be explored for Induction machine.

- The potential application of the proposed FFNN based switching controller for three−level NPC inverters and possibly multi−level inverters is of great interest. The performance comparison of the proposed controller against traditional switching schemes is to be compared.

- Employing RL, research studies on drive speed sensorless control, fault monitoring, and tolerance control will also be explored in the future.

A Appendix

To adjust the integral proportional (IP) controller gains for the control system, we provide the pseudo code for Particle Swarm Optimisation , Cuckoo search, and JAYA algorithm. By searching the search space iteratively and adjusting the gains in response to the control system's performance. By applying optimization algorithms for integral proportional gain tuning, we aim to automate the gain optimization process and enhance the control system's performance by efficiently finding near-optimal controller gains for various control applications. The MATLAB optimization code snippets are included with a caption and label for reference. Researcher can modify the code and the description to fit there specific optimization scenario.

1. Particle Swarm Optimization

Algorithm 1 Particle Swarm Optimization.

Input: $(n, m_{max}, w, c_1, c_2)$
Output: $g^m_{best} = kp, ki$

Initialize positions x_i and velocity v_i of n particles
Calculate the cost function $(ITAE)$: $x = (x_1, ..., x_n)$
Assign present best p_{best} of all swarm particle to global best g_{best}
for $(m = 0; m < m_{max})$ **do**
    for all n particles **do**
        Calculate new velocity $v_i(m+1)$ using equation (19)
        Check and amend boundary violations in $v_i(m+1)$
        Update positions $x_i(m+1) = x_i(m) + v_i(m+1)$
        Check and amend boundary violations in $x_i(m+1)$
        Calculate error at updated positions $x_i(m+1)$
        Find the current best for each particle x^{best}_i
    end for
    Update the global best g^m_{best}
end for

```
% MATLAB code PSO
for iteration = 1:maxIter
    for particle = 1:swarmSize
```

```
        % For each particle
4         for dimension = 1:nDimensions

            % Calculate new velocity for this dimension
5             currPos = swarm{particle,dimensionCol}{
                  position,dimension};
6 % Get current position
7             personalBestPos   = swarm{particle,
                  dimensionCol}{personalBest,dimension};
8 % Get best position
9             currVel = swarm{particle,dimensionCol}{
                  velocity,dimension};
10 % Get current velocity
11            gBestPos = gBest{1,dimensionCol}{position,
                  dimension};                                  %
                   Get global best position
12            newVel = updateVelocity(currPos,
                  personalBestPos,currVel,gBestPos,w,c1,c2)
                  ;         % Calculate new velocity
13 % Check and amend boundary violation (velocity)
14            if newVel < lbVelocity

                % Check for lower boundary violation
15               newVel = lbVelocity;

                    % If boundary violated, adjust
                    velocity
16            elseif newVel > ubVelocity

                % Check for upper boundary violation
```

```
17                  newVel = ubVelocity;

                        % If boundary violated, adjust
                        velocity
18              end
19 % Update velocity of this dimension
20              swarm{particle,dimensionCol}{velocity,
                    dimension} = newVel;
21 % Update position of this dimension
22              newPos = currPos + newVel;

                    % Calculate new position
23              swarm{particle,dimensionCol}{position,
                    dimension} = newPos;
24 % Assign new position
25 % Check and amend boundary violation (position)
26              if newPos < bounds{lb,dimension} || newPos >
                     bounds{ub,dimension}
27 % Check for boundary violation
28                  swarm{particle,dimensionCol}{position,
                        dimension} = ...
                                                % If boundary
                        violated, adjust position
29                       Init(bounds{lb,dimension}, bounds{ub
                            ,dimension});
30              end
31            end
32          % Evaluate this particle for err
33          err = fobj();
```

```
            % Call objective function
34          % Update personal best position for all
               dimensions
35          if err <= swarm{particle,errCol}
36              for dimension = 1:nDimensions

                    % For each dimension
37                  swarm{particle,dimensionCol}{
                        personalBest,dimension} = ...
                                                % Update personal
                        best with current position
38                      swarm{particle,dimensionCol}{
                            position,dimension};
39              end
40          end
41          % Update err of current particle
42          swarm{particle,errCol} = err;
43          % Update global best
44          if swarm{particle,errCol} <= gBest{1,errCol}
                                                              %
               Check if err of this particle <= global best
               err
45              gBest{1,errCol} = swarm{particle,errCol};
                                                                %
                    Assign new err to global best
46              % Update positions for all dimensions of
                   gbest
47              for dimension = 1:nDimensions

                    % For each dimension
```

```
                    gBest{1,dimensionCol}{position,dimension
                        } = ...
                         swarm{particle,dimensionCol}{
                            position,dimension};
                end

            end
            if gBest{1,errCol} >= prevBestErr
             stagnateCounter = stagnateCounter + 1;
             else
             stagnateCounter = 0; % Reset counter if there's
                 improvement
             end
%      % Adjust inertia weight if stagnation threshold is
    reached
             if stagnateCounter >= stagnateThreshold
             w = w + w_increase;
             if w > w_max  % Ensure w does not exceed w_max
                 w = w_max;
             end
             stagnateCounter = 0; % Reset the counter
       end
     end
```

Listing 1: PSO Implementation

2. Cuckoo Search Optimization

```
% MATLAB code CS
for k=1:max_iter
        for i=1:discovery_eggs
```

Algorithm 2 Cuckoo Search Algorithm.

Input: (n, p, m_{max})
Output: $x_{best} = K_p, K_i$

Randomly initialize n candidates $\bar{x}_i = (i = 1, 2, ..., n)$
Calculate the error (ITAE) of each candidate x_i
Sort the population in ascending order of error $(ITAE)$
for $(m = 0; m < m_{max})$ **do**
  for $(i = 0 : i < p * n)$ **do**
    Pick a candidate x_i with error $ITAE$ at random
    Generate x_j by mutating x_i using equation (20)
    Calculate error e_j of x_j
    if $(e_j < ITAE)$ **then**
      Replace x_i with x_j
    end if
  end for
  Sort the population in ascending order of error $ITAE$
  Initialize $(p * n)$ worst candidates randomly
  keep the best solution
end for

```
        % Pick an egg from nest at random
        index_egg = round( rand() * nest_size );
        if index_egg == 0   % Because there is no egg at
            0
            index_egg = 1;
        end
        egg = cell(1,3);
        [ egg{1,1}, egg{1,2}, egg{1,3} ] = deal( nest{
            index_egg,1}, nest{index_egg,2}, nest{
            index_egg,3} );
        % Create a new egg from this egg
        new_egg = egg; % First get the egg in variable
            new_egg
        % Modify kp
        if rand() < 0.5      % Toss a coin
            new_egg{1,1} = new_egg{1,1} - alpha*rand()^(
```

```
            lambda);
        else
            new_egg{1,1} = new_egg{1,1} + alpha*rand()^(
            lambda);
        end
        % Check boundary
        if new_egg{1,1} < lb_kp || new_egg{1,1} > ub_kp
            new_egg{1,1} = Init(lb_kp, ub_kp);
        end
        % Modify ki
        if rand() < 0.5      % Toss a coin
            new_egg{1,2} = new_egg{1,2} - alpha*rand()^(
            lambda);
        else
            new_egg{1,2} = new_egg{1,2} + alpha*rand()^(
            lambda);
        end
        % Check boundary
        if new_egg{1,2} < lb_ki || new_egg{1,2} > ub_ki
            new_egg{1,2} = Init(lb_ki, ub_ki);
        end
        % Calculate err of new_egg
        [kp, ki, new_egg{1,3}] = deal( new_egg{1,1},
           new_egg{1,2}, fobj() );
        % If new_egg is better than egg, new_egg will
           replace egg in the nest
        if new_egg{1,3} <= egg{1,3}  % if err(new_egg) <
            err(egg)
            [nest{index_egg, 1}, nest{index_egg, 2},
            nest{index_egg, 3}] = deal( new_egg{1,1},
```

```
                    new_egg{1,2}, new_egg{1,3} );
38          end
39      end
40      for i = (nest_size - discovery_eggs)+1 : nest_size
41          [nest{i,1}, kp] = deal( Init(lb_kp, ub_kp) );
               % kp
42          [nest{i,2}, ki] = deal( Init(lb_ki, ub_ki) );
               % ki
43          nest{i,3} = fobj();     % err
44      end
45  end
```

Listing 2: CS Implementation

3. Jaya Optimization

```
1  % MATLAB code JAYA
2  for iteration = 1:maxIter

      % For each iteration
3      for particle = 1:swarmSize

          % For each particle
4          [ placeHolder{1,dimensionCol}, placeHolder{1,
              errCol} ] = ...
5              deal( swarm{particle,dimensionCol}, swarm{
                  particle,errCol} );
6  % Save this particle before modification
7          for dimension = 1:nDimensions

              % For each dimension  % Modify particle
```

Algorithm 3 Jaya Optimization Algorithm

Input: n, $m_{\max}$
Output: K_i^*, K_p^*
Randomly initialize n population with random IP gains: K_i^i, K_p^i for $i = 1,2,\ldots,n$
$m \leftarrow 0$
while $m < m_{\max}$ **do**
  for each candidate in n **do**
    Evaluate the IP controller with K_i^i and K_p^i
    Calculate cost function (ITAE) and store it for candidate i
  end for
  Find the candidate with the lowest performance criterion as best_solution
  Find the candidate with the highest performance criterion as worst_solution

  for each candidate in n **do**
    $K_i^i \leftarrow K_i^i + \text{rand}() \times (\text{best}.K_i - K_i^i)$
    $K_p^i \leftarrow K_p^i + \text{rand}() \times (\text{best}.K_p - K_p^i)$
  end for
  if Stopping Criteria Satisfied **then**
    break
  end if
  $m \leftarrow m + 1$
end while
$K_i^* \leftarrow \text{best}.K_i$
$K_p^* \leftarrow \text{best}.K_p$

```
            position
8               currPos = swarm{particle,dimensionCol}{
                    position,dimension};
9   % Get current particle position
10              bestPos = swarm{bestParticle,dimensionCol}{
                    position,dimension};
11  % Get best particle position
12              worstPos = swarm{worstParticle,dimensionCol
                    }{position,dimension};
13  % Get worst particle position
14              newPos = updatePosition(currPos,bestPos,
                    worstPos);
                                                    %
                    Calculate new position
15              swarm{particle,dimensionCol}{position,
                    dimension} = newPos;
16  % Assign new position
17              % Check and amend boundary violation (
                    position)
18              if newPos < bounds{lb,dimension} || newPos >
                     bounds{ub,dimension}
19  % Check for boundary violation
20                  swarm{particle,dimensionCol}{position,
                        dimension} = ...
                                                % If boundary
                        violated, adjust position
21                      Init(bounds{lb,dimension}, bounds{ub
                            ,dimension});
22              end
23          end
```

```
        % Evaluate this particle for err
        swarm{particle,errCol} = fobj();

           % Call objective function, get err
        if placeHolder{1,errCol} < swarm{particle,errCol
           }                                          % Check
            if previous version of particle was better
            [ swarm{particle,dimensionCol}, swarm{
               particle,errCol} ] = ...
                                   % If true, restore
               previous particle
                deal( placeHolder{1,dimensionCol},
                   placeHolder{1,errCol} );
        end
        % Shuffle by 'err' column since best/worst could
            change
        swarm = sortrows(swarm,errCol);
        end
    end
```

Listing 3: Proposed Jaya Implementation

4. Experimental Results of DSR

The test results focused on evaluating the effectiveness of DSR in the control of AC machines. These results showcase that DSR not only successfully substitutes the traditional linear current controller but also enhances closed-loop performance in AC machine control systems. The data and analysis underline the robustness and efficiency of DSR, marking a significant advancement in the field of AC machine control

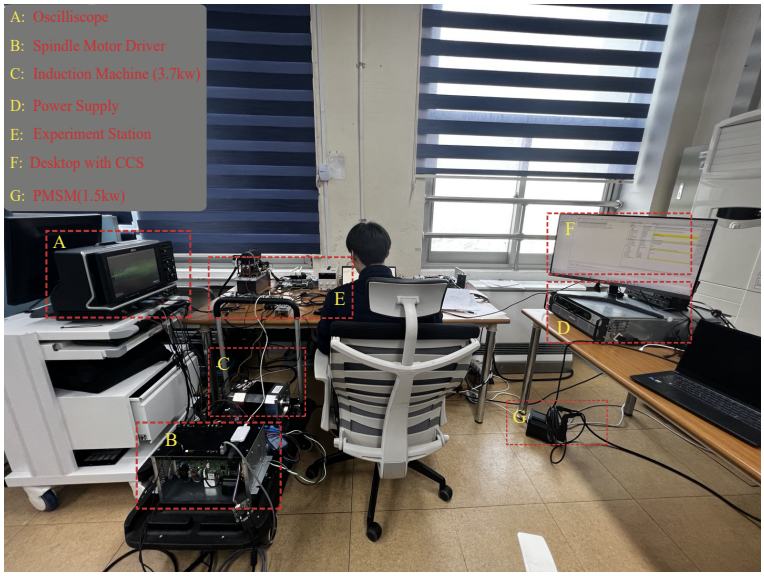


Figure 56: Experimental platform.

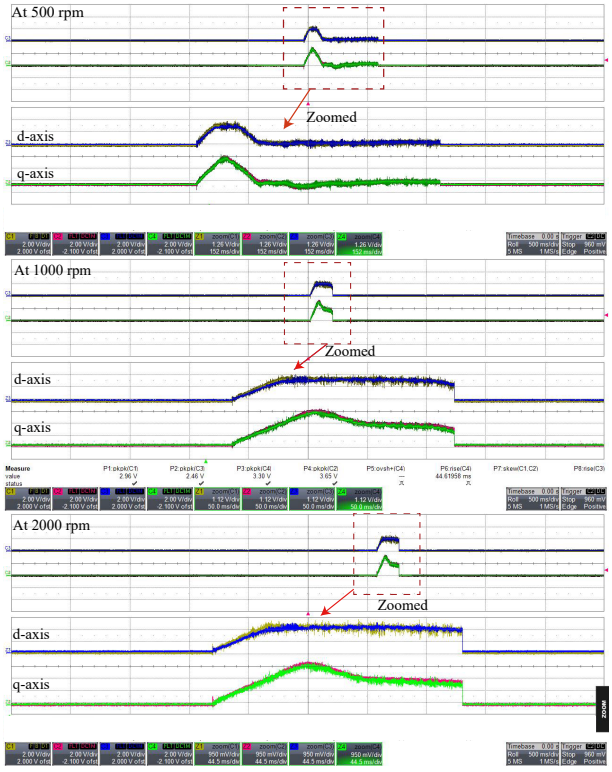
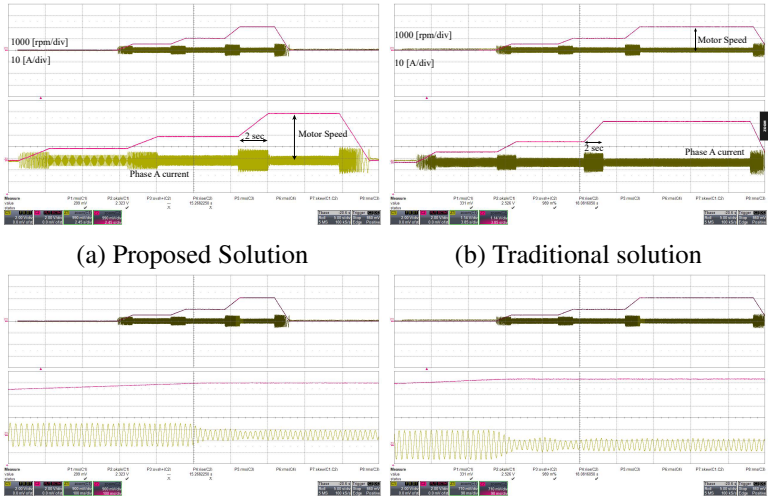


Figure 57: Motor drive dq-current tracking response at varied speed.



(a) Proposed Solution (b) Traditional solution

(c) Zoomed (a) (d) Zoomed (b)

Figure 58: IM performance

REFERENCES

[1] J. M. S. Ribeiro, M. F. Santos, M. J. Carmo, and M. F. Silva, "Comparison of pid controller tuning methods: Analytical/classical techniques versus optimization algorithms," in *2017 18th International Carpathian Control Conference (ICCC)*, 2017, pp. 533–538. DOI: 10.1109/CarpathianCC.2017.7970458.

[2] S. B. Joseph, E. G. Dada, A. A., D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for pid controller parameters tuning: Review, approaches and open problems," *Heliyon*, vol. 8, 5 2022. DOI: 10.1016/j.heliyon.2022.e09005.

[3] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997. DOI: 10.1109/4235.585893.

[4] M Usama, A Umer, and J Kim, "Evolutionary and swarm intelligence in optimal gain tuning for dynamic speed control of spmsm," in

Interdisciplinary Conference on Mechanics, Computers and Electrics (ICMECE 2022), Barcelona, Spain, 2022.

[5] M. Kazmierkowski, R Krishnan, and F Blaabjerg, *Control in Power Electronics*. New York, USA: Academic Press, 2002.

[6] M. S. Rafaq and J.-W. Jung, "A comprehensive review of state-of-the-art parameter estimation techniques for permanent magnet synchronous motors in wide speed range," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4747–4758, 2020. DOI: 10.1109/TII.2019.2944413.

[7] M. S. Rafaq, W. Midgley, and T. Steffen, "A review of the state of the art of torque ripple minimization techniques for permanent magnet synchronous motors," *IEEE Transactions on Industrial Informatics*, pp. 1–13, 2023. DOI: 10.1109/TII.2023.3272689.

[8] M. Usama and J. Kim, "Improved self-sensing speed control of ipmsm drive based on cascaded nonlinear control," *Energies*, vol. 14, no. 8, p. 2205, 2021. DOI: 10.3390/en14082205.

[9] M. Usama, Y.-O. Choi, and J. Kim, "Speed sensorless control based on adaptive luenberger observer for ipmsm drive," in *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, 2021, pp. 602–607. DOI: 10.1109/PEMC48073.2021.9432536.

[10] M. Usama and J. Kim, "Low-speed transient and steady-state performance analysis of ipmsm for nonlinear speed regulator with effective compensation scheme," *Energies*, vol. 14, no. 20, p. 6679, 2021. DOI: 10.3390/en14206679.

[11] S. Zhang, O. Wallscheid, and M. Porrmann, "Machine learning for the control and monitoring of electric machine drives: Advances and trends," *IEEE Open Journal of Industry Applications*, vol. 4, pp. 188–214, 2023. DOI: 10.1109/OJIA.2023.3284717.

[12] M. Usama and I.-Y. Lee, "Data-driven non-linear current controller based on deep symbolic regression for spmsm," *Sensors*, vol. 22, no. 21, p. 8240, 2022. DOI: 10.3390/s22218240.

[13] M. Usama and J. Kim, "A reinforcement learning speed controller for surface-mount permanent magnet synchronous motors," in *Conference on Information and Control Systems(CICS) 2021*, 2022.

[14] G. Pellegrino, A. Vagati, P. Guglielmi, and B. Boazzo, "Performance comparison between surface-mounted and interior pm motor drives for electric vehicle application," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 803–811, 2012. DOI: 10.1109/TIE.2011.2151825.

[15] K. Nam, *AC motor control and electric vehicle applications*. New York, USA: CRC Press, 2010, ch. Permanent Magnet AC Motors.

[16] Y. Dai, L. Song, and S. Cui, "Development of pmsm drives for hybrid electric car applications," *IEEE Transactions on Magnetics*, vol. 43, no. 1, pp. 434–437, 2007. DOI: 10.1109/TMAG.2006.887718.

[17] T. Renyuan, *Theory and Design of the Modern Permanent Magnet Electric Machines*. Beijing, China: Publishing House of Mechanical Industry, 1997, pp. 268–270.

[18] P. Krause, O. Wasynczuk, and S. Sudhoff, *Analysis of Electric Machinery*. IEEE Press, 1995.

[19] J. Hendershot and T. Miller, *Design of Brushless Permanent Magnet Motor*. Oxford Science Publications, 1994.

[20] D. Mohanraj, R. Aruldavid, R. Verma, *et al.*, "A review of bldc motor: State of art, advanced control techniques, and applications," *IEEE Access*, vol. 10, pp. 54 833–54 869, 2022. DOI: 10.1109/ACCESS.2022.3175011.

[21] P. Pillay and R. Krishnan, "Modeling of permanent magnet motor drives," *IEEE Transactions on Industrial Electronics*, vol. 35, no. 4, pp. 537–541, 1988. DOI: 10.1109/41.9176.

[22] I. Boldea, "Control issues in adjustable speed drives," *IEEE Industrial Electronics Magazine*, vol. 2, no. 3, pp. 32–50, 2008.

[23] D. Stellas, "Sensorless scalar and vector control of a subsea pmsm," Ph.D. dissertation, Chalmers University of Technology, 2013.

[24] J. Ziegler and N. Nichols, "Optimum settings for automatic controllers," 1993.

[25] G. Cohen, "Theoretical consideration of retarded control," *Transactions of the American Society of Mechanical Engineers*, vol. 75, pp. 827–834, 1953.

[26] D. Kim and J. Cho, "A biologically inspired intelligent pid controller tuning for avr systems," *International Journal of Control, Automation, and Systems*, vol. 4, pp. 624–636, 2006.

[27] S. Li and H. Gu, "Fuzzy adaptive internal model control schemes for pmsm speed-regulation system," *IEEE Transactions on Industrial Informatics*, vol. 8, pp. 767–779, 2012.

[28] W. Tu, G. Luo, Z. Chen, L. Cui, and R. Kennel, "Predictive cascaded speed and current control for pmsm drives with multi-timescale optimization," *IEEE Transactions on Power Electronics*, vol. 34, pp. 11 046–11 061, 2019.

[29] W. Gao and J. Hung, "Variable structure control of nonlinear systems: A new approach," *IEEE Transactions on Industrial Electronics*, vol. 40, pp. 45–55, 1993.

[30] T. Li, X. Sun, G. Lei, Y. Guo, Z. Yang, and J. Zhu, "Finite-control-set model predictive control of permanent magnet synchronous motor drive systems—an overview," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 12, pp. 2087–2105, 2022. DOI: 10.1109/JAS.2022.105851.

[31] V. Lioudis, "Chattering reduction applied in pmsm sensorless control using second order sliding mode observer," in *Proceedings of the 2015 9th International Conference on Compatibility and Power Electronics (CPE)*, Costa da Caparica, Portugal, 2015, pp. 240–245.

[32] M. Usama and J. Kim, "Performance improvement of speed control of ipmsm drive based on nonlinear current control," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 4, Article 9, 2021.

[33] B. Bose, "Artificial intelligence techniques: How can it solve problems in power electronics? an advancing frontier," *IEEE Power Electronics Magazine*, vol. 7, no. 4, pp. 19–27, 2020.

[34] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Transactions on Power Electronics*, vol. 36, no. 4, pp. 4633–4658, 2021.

[35] M. Cirrincione, M. Pucci, and G. Vitale, *Power Converters and AC Electrical Drives With Linear Neural Networks*. Boca Raton, FL, USA: CRC Press, 2017.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2012.

[37] A. Vaswani and et al., "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 1–11.

[38] Y.-m. You, "Multi-objective optimal design of permanent magnet synchronous motor for electric vehicle based on deep learning," *Applied Sciences*, vol. 10, no. 2, Art. no. 482, 2020.

[39] S. Zhang, O. Wallscheid, and M. Porrmann, "Machine learning for the control and monitoring of electric machine drives: Advances and trends," *IEEE Open Journal of Industry Applications*, vol. 4, pp. 188–214, 2023. DOI: 10.1109/OJIA.2023.3284717.

[40] B. Bose, "Neural network applications in power electronics and motor drives – an introduction and perspective," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 14–33, 2007.

[41] B. Bose, "Global energy scenario and impact of power electronics in the 21st century," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 7, pp. 2638–2651, 2013.

[42] K. Chau, C. Chan, and C. Liu, "Overview of permanent-magnet brushless drives for electric and hybrid electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2246–2257, 2008.

[43] F. Blaabjerg, S. Freysson, H.-H. Hansen, and S. Hansen, "A new optimized space-vector modulation strategy for a component-minimized voltage source inverter," *IEEE Transactions on Power Electronics*, vol. 12, no. 4, pp. 704–714, 1997. DOI: 10.1109/63.602566.

[44] M. Kushwah and A. Patra, "Tuning pid controller for speed control of dc motor using soft computing techniques - a review," *Advances in Electronics and Electrical Engineering*, vol. 4, pp. 141–148, 2014.

[45] E. Abbasi and N. Naghavi, "Offline auto-tuning of a pid controller using extended classifier system (xcs) algorithm," *Journal of Advanced Computer Engineering and Technology*, vol. 3, pp. 41–44, 2017.

[46] J.-W. Jung, V. Q. Leu, T. D. Do, E.-K. Kim, and H. H. Choi, "Adaptive pid speed control design for permanent magnet synchronous motor drives," *IEEE Transactions on Power Electronics*, vol. 30, no. 2, pp. 900–908, 2015.

[47] K. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[48] T. Higashiyama *et al.*, "Auto-tuning of motor drive system by simple adaptive control approach," in *Proceedings of the IEEE Control Applications International Conference*, 2000, pp. 868–873.

[49] M. Calvini, M. Carpita, A. Formentini, and M. Marchesoni, "Pso-based self-commissioning of electrical motor drives," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 768–776, 2015.

[50] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[51] X. Yang, *Cuckoo Search and Firefly Algorithm: Overview and Analysis*. Springer, 2014, pp. 1–26.

[52] X. Yang and S. Deb, "Engineering optimization by cuckoo search," *Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.

[53] K. Nam, *AC motor control and electric vehicle applications*. New York, USA: CRC Press, 2010, ch. Preliminaries for Motor Control.

[54] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[55] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, 1998, pp. 591–600.

[56] X. Yang and S. Deb, "Cuckoo search: state-of-the-art and opportunities," in *2017 IEEE 4th International Conference on Soft Computing and Machine Intelligence (ISCMI)*, 2017, pp. 55–59. DOI: 10.1109/ISCMI.2017.8279597.

[57] J. Ahmed and Z. Salam, "A maximum power point tracking (mppt) for pv system using cuckoo search with partial shading capability," *Applied Energy*, vol. 119, pp. 118–130, 2014. DOI: 10.1016/j.apenergy.2014.01.047.

[58] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013. DOI: 10.1007/s00366-011-0241-y.

[59] A. Gotmare, R. Patidar, and N. V. George, "Nonlinear system identification using a cuckoo search optimized adaptive hammerstein model," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2538–2546, 2015. DOI: 10.1016/j.eswa.2014.11.026.

[60] H. Rakhshani, E. Dehghanian, and A. Rahati, "Hierarchy cuckoo search algorithm for parameter estimation in biological systems," *Chemometrics*

and Intelligent Laboratory Systems, vol. 159, pp. 97–107, 2016. DOI: 10.1016/j.chemolab.2016.09.007.

[61] R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.

[62] R. A. Zitar, M. A. Al-Betar, M. A. Awadallah, and et al., "An intensive and comprehensive overview of jaya algorithm, its versions and applications," *Archives of Computational Methods in Engineering*, vol. 29, pp. 763–792, 2022. DOI: 10.1007/s11831-021-09585-8.

[63] X. Song, J. Fang, and B. Han, "High-precision rotor position detection for high-speed surface pmsm drive based on linear hall-effect sensors," *IEEE Transactions on Power Electronics*, vol. 31, pp. 4720–4731, 2016. DOI: 10.1109/TPEL.2015.2468258.

[64] G. Feng, C. Lai, and N. C. Kar, "Practical testing solutions to optimal stator harmonic current design for pmsm torque ripple minimization using speed harmonics," *IEEE Transactions on Power Electronics*, vol. 33, pp. 5181–5191, 2018. DOI: 10.1109/TPEL.2017.2737881.

[65] L. Harnefors, S. E. Saarakkala, and M. Hinkkanen, "Speed control of electrical drives using classical control methods," *IEEE Transactions on Industry Applications*, vol. 49, no. 2, pp. 889–898, 2013. DOI: 10.1109/TIA.2012.2220271.

[66] A. Houari, A. Bouabdallah, A. Djerioui, *et al.*, "An effective compensation technique for speed smoothness at low-speed operation of pmsm drives," *IEEE Transactions on Industry Applications*, vol. 54, pp. 647–655, 2018. DOI: 10.1109/TIA.2018.2793610.

[67] M. Usama and J. Kim, "Robust adaptive observer-based finite control set model predictive current control for sensorless speed control of surface permanent magnet synchronous motor," *Transactions of the Institute of Measurement and Control*, vol. 43, pp. 1416–1429, 2021. DOI: 10.1177/0142331220919123.

[68] C. J. Fallaha, M. Saad, H. Y. Kanaan, and K. Al-Haddad, "Sliding-mode robot control with exponential reaching law," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 600–610, 2011. DOI: 10.1109/TIE.2010.2048850.

[69] W. Qian, S. K. Panda, and J.-X. Xu, "Torque ripple minimization in pm synchronous motors using iterative learning control," *IEEE Transactions on Power Electronics*, vol. 19, 272–279, 2004. DOI: 10.1109/TPEL.2003.821821.

[70] J. L. Junell, T. Mannucci, Y. Zhou, and E. V. Kampen, "Self-tuning gains of a quadrotor using a simple model for policy gradient reinforcement learning," in *AIAA Guidance, Navigation, & Control Conference*, 2016, pp. 1–15.

[71] I. Grondman, M. Vaandrager, L. Busoniu, R. Babuska, and E. Schuitema, "Efficient model learning methods for actor-critic control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 42, no. 3, pp. 591–602, 2012.

[72] M. Wishart and R. Harley, "Identification and control of induction machines using artificial neural networks," *IEEE Transactions on Industry Applications*, vol. 31, no. 3, pp. 612–619, 1995. DOI: 10.1109/28.382123.

[73] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Deep residual convolutional and recurrent neural networks for temperature estimation

in permanent magnet synchronous motors," in *2019 IEEE International Electric Machines and Drives Conference (IEMDC)*, 2019, pp. 1439–1446. DOI: 10.1109/IEMDC.2019.8785109.

[74] P. Garcia, F. Briz, D. Raca, and R. D. Lorenz, "Saliency-tracking-based sensorless control of ac machines using structured neural networks," *IEEE Transactions on Industry Applications*, vol. 43, no. 1, pp. 77–86, 2007. DOI: 10.1109/TIA.2006.887309.

[75] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[76] S. Li, H. Won, X. Fu, M. Fairbank, D. Wunsch, and E. Alonso, "Neural-network vector controller for permanent-magnet synchronous motor drives: Simulated and hardware-validated results," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3218–3230, 2020.

[77] F. El-Sousy, "Intelligent optimal recurrent wavelet elman neural network control system for permanent-magnet synchronous motor servo drive," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1986–2003, 2013.

[78] P. Garcia, F. Briz, D. Raca, and R. Lorenz, "Saliency-tracking-based sensorless control of ac machines using structured neural networks," *IEEE Transactions on Industry Applications*, vol. 43, no. 1, pp. 77–86, 2007.

[79] B. K. Petersen, M. Landajuela, T. N. Mundhenk, C. P. Santiago, S. K. Kim, and J. T. Kim, "Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients," in *Proceedings of the International Conference on Learning Representations*, Virtual Event, 2021.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to Almighty ALLAH, whose blessings and strength have been the guiding force behind the completion of this paper. I am truly grateful for the courage and patience granted to me throughout this journey.

I extend my sincere appreciation to my family and friends for their unwavering support and encouragement during my research endeavors. Their constant presence and belief in my capabilities have been invaluable sources of motivation.

A special thank you goes to my advisor, Dr. Jaehong Kim, whose unwavering support and guidance have been instrumental in shaping my academic journey at Chosun University. His mentorship, encouragement, and dedication have profoundly influenced my problem-solving skills and approach to engineering. His vast knowledge, diligent research, and patient teaching have inspired and nurtured my growth as a critical thinker. It has been an esteemed privilege to be under his supervision.

I would also like to acknowledge and thank my committee members for their valuable input, feedback, and time invested in reviewing this thesis. Their involvement has enriched the quality of this work, and I am grateful for their contributions.

Once again, I extend my heartfelt appreciation to all those who have contributed to the successful completion of this endeavor. Your support and encouragement have been invaluable, and I am deeply grateful for each and every one of you.