FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Controller for Brushless Motors using ARM

Jorge Alexandre Fonseca Brandão

FINAL VERSION

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Filipe Neves dos Santos

July 31, 2026

Controller for Brushless Motors using ARM

Jorge Alexandre Fonseca Brandão

Mestrado em Engenharia Eletrotécnica e de Computadores

July 31, 2026

Resumo

Os motores síncronos de ímanes permanentes (Permanent Magnet Synchronous Motors (PMSMs)) desempenham atualmente um papel fundamental em aplicações de elevado desempenho, incluindo veículos elétricos, automação industrial, robótica e sistemas de controlo de movimento. O seu funcionamento eficiente depende da implementação de estratégias avançadas de controlo capazes de regular de forma precisa o binário e o fluxo magnético, frequentemente através de sistemas embebidos baseados em microcontroladores.

Esta dissertação tem como principal objetivo o desenvolvimento de uma arquitetura modular de controlo de PMSMs, concebida de forma a facilitar a sua posterior implementação em plataformas embebidas baseadas em ARM. Neste contexto, são estudadas, implementadas e validadas duas das principais estratégias de controlo utilizadas nestes motores: o Direct Torque Control (DTC) e o Field-Oriented Control (FOC). A implementação é estruturada em blocos funcionais independentes e desenvolvida em código C, permitindo que os algoritmos de controlo sejam posteriormente transferidos e integrados numa plataforma ARM com alterações mínimas.

Após uma revisão dos fundamentos teóricos relativos à estrutura e funcionamento dos motores brushless, às diferentes técnicas de controlo, aos métodos de deteção da posição do rotor e à modelação matemática da máquina, é desenvolvida uma arquitetura completa de acionamento no ambiente de simulação PSIM. Este ambiente é utilizado não apenas para simular o comportamento do motor e do inversor, mas também como plataforma de desenvolvimento e validação dos módulos de controlo em código C antes da sua implementação em hardware.

São desenvolvidos e avaliados controladores DTC e FOC, incluindo os respetivos módulos funcionais, tais como transformações de coordenadas, estimação de fluxo e binário, reguladores de corrente e geração dos sinais de comando do inversor. O desempenho de ambas as estratégias é validado através de simulações e comparado em termos de seguimento da referência de velocidade, resposta dinâmica, ondulação de binário e comportamento em regime permanente.

Como extensão ao controlador FOC, são ainda apresentados métodos de identificação dos principais parâmetros elétricos e mecânicos do motor, nomeadamente resistência estatórica, indutâncias, fluxo dos ímanes permanentes e parâmetros mecânicos, bem como uma implementação baseada em sensores Hall Estas funcionalidades seguem a mesma abordagem modular, permitindo a sua integração progressiva na arquitetura de controlo.

Os resultados obtidos demonstram que ambas as estratégias proporcionam um controlo eficaz do PMSM, evidenciando as vantagens do FOC em aplicações que requerem elevada precisão e reduzida ondulação de binário, enquanto o DTC apresenta uma resposta dinâmica mais rápida. Para além da comparação entre as estratégias, o trabalho resulta numa estrutura de software modular e reutilizável, validada em ambiente de simulação e concebida para facilitar a transição dos algoritmos desenvolvidos para plataformas ARM e aplicações de controlo em tempo real.

Palavras-chave: Motores Síncronos de Ímanes Permanentes, Controlo Orientado pelo Campo, Controlo Direto do Binário, Controlo Vetorial, ARM, PSIM, Sensores Hall, Identificação de Parâmetros.

Abstract

Permanent Magnet Synchronous Motors (PMSMs) play a fundamental role in high-performance applications, including electric vehicles, industrial automation, robotics, and motion control systems. Their efficient operation relies on advanced control strategies capable of accurately regulating electromagnetic torque and magnetic flux, often through embedded systems based on microcontrollers.

The main objective of this dissertation is the development of a modular PMSM control architecture designed to facilitate its subsequent implementation on ARM-based embedded platforms. Within this framework, two of the most widely adopted PMSM control strategies, Direct Torque Control (DTC) and Field-Oriented Control (FOC), are studied, implemented, and validated. The implementation is structured into independent functional blocks and developed in C code, allowing the control algorithms to be subsequently transferred and integrated into an ARM platform with minimal modifications.

Following a review of the theoretical foundations of brushless motor technology, control strategies, rotor position detection methods, and mathematical modeling, a complete motor drive architecture is developed within the PSIM simulation environment. PSIM is used not only to simulate the behavior of the motor and inverter, but also as a platform for developing and validating the C-based control modules prior to their implementation in hardware.

Both DTC and FOC controllers are developed and evaluated, including their respective functional modules, such as coordinate transformations, torque and flux estimation, current regulation, and inverter command signal generation. Their performance is validated through simulation and compared in terms of speed reference tracking, dynamic response, torque ripple, and steady-state behavior.

As an extension to the FOC controller, methods for identifying the main electrical and mechanical motor parameters are also presented, including stator resistance, inductances, permanent magnet flux linkage, and mechanical parameters. In addition, a Hall sensor-based rotor position and speed estimation method is implemented. These functionalities follow the same modular approach, allowing their progressive integration into the control architecture.

The obtained results demonstrate that both strategies provide effective PMSM control, with FOC offering advantages in applications requiring high precision and reduced torque ripple, while DTC provides a faster dynamic response. Beyond the comparison of the two control strategies, the work results in a modular and reusable software structure, validated in simulation and designed to facilitate the transition of the developed algorithms to ARM-based platforms and real-time motor control applications.

Keywords: Permanent Magnet Synchronous Motors, Field-Oriented Control, Direct Torque Control, Vector Control, Embedded Systems, ARM, PSIM, Hall Sensors, Parameter Identification.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) constitute a global framework adopted by all United Nations Member States to promote sustainable development through social, economic, and environmental actions. The framework consists of 17 interconnected goals that address global challenges such as poverty, inequality, climate change, responsible resource management, and technological innovation.

Although this thesis focuses on the modelling and implementation of advanced control strategies for Permanent Magnet Synchronous Motors (PMSMs), its outcomes contribute to several Sustainable Development Goals by improving the efficiency and performance of electric drive systems. In particular, the work aligns with SDG 7 (Affordable and Clean Energy), SDG 9 (Industry, Innovation and Infrastructure), and SDG 12 (Responsible Consumption and Production).

SDG 7 – Affordable and Clean Energy Ensure access to affordable, reliable, sustainable and modern energy for all.

SDG 9 – Industry, Innovation and Infrastructure Build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation.

SDG 12 – Responsible Consumption and Production Ensure sustainable consumption and production patterns.

The implementation and evaluation of Field-Oriented Control (FOC) and Direct Torque Control (DTC) strategies contribute to these goals by enabling more efficient operation of electrical machines. High-performance motor control improves energy conversion efficiency, reduces electrical losses, and minimizes torque ripple, leading to lower energy consumption and better utilization of available resources. Furthermore, advanced control algorithms support the development of innovative industrial automation and electric mobility applications, where efficient motor drives play a fundamental role.

Table 1 summarizes the relationship between this thesis and the corresponding Sustainable Development Goals.

Overall, the research presented in this dissertation contributes to the advancement of efficient electric drive technologies, supporting more sustainable industrial systems and promoting the broader adoption of energy-efficient solutions.

Table 1: Contribution of this thesis to the United Nations Sustainable Development Goals.

SDG	**Target**	**Contribution of this work**
7	7.3	Improve energy efficiency through the implementation of advanced motor control techniques that reduce losses and optimize electrical drive operation.
9	9.5	Promote technological innovation by developing and validating modern control strategies for high-performance PMSM drives.
12	12.2	Support the sustainable use of resources by increasing the efficiency of electrical energy conversion and reducing unnecessary energy consumption.

Contents

List of Figures

List of Tables

List of Acronyms

AC	Alternating Current
BLDC	Brushless Direct Current (Motor)
CPMR	Cogging Cycles Per Mechanical Revolution
DC	Direct Current
DTC	Direct Torque Control
EMF	Electromotive Force
FOC	Field-Oriented Control
GCT	Gate-Commutated Thyristor
HES	Hall-Effect Sensor
IGBT	Insulated Gate Bipolar Transistor
IPM	Interior Permanent Magnet (machine/rotor)
LCM	Least Common Multiple
NdFeB	Neodymium-Iron-Boron (magnet material)
PM	Permanent Magnet
PMBDCM	Permanent Magnet Brushless Direct Current Machine
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse-Width Modulation
SCR	Silicon-Controlled Rectifier
SPM	Surface Permanent Magnet (machine/rotor)
SVPWM	Space Vector Pulse-Width Modulation
SVM	Space Vector Modulation
Y	Wye (Star) Connection Topology
Δ	Delta Connection Topology

Chapter 1

Introduction

1.1 Motivation

Permanent Magnet Synchronous Motors (PMSMs) have become one of the most important electrical machine technologies in modern industrial and transportation applications. Their high efficiency, compact design, high torque density, and excellent dynamic performance make them particularly suitable for electric vehicles, robotics, industrial automation, aerospace systems, and precision motion control.

The widespread adoption of PMSMs has been accompanied by the development of increasingly sophisticated control strategies capable of exploiting their performance potential. Among these, Direct Torque Control (DTC) and Field-Oriented Control (FOC) have established themselves as two of the most prominent techniques for high-performance motor drives. While both methods provide independent control of electromagnetic torque and magnetic flux, they differ significantly in their operating principles, dynamic behavior, implementation complexity, and steady-state characteristics.

The design and implementation of such control algorithms require not only a solid theoretical understanding of PMSM modeling but also careful consideration of practical aspects, including reference-frame transformations, current regulation, pulse-width modulation techniques, parameter sensitivity, and rotor position information. Furthermore, the performance of advanced controllers strongly depends on the accuracy of the machine parameters used during control and estimation. In practical motor drive applications, these algorithms are typically executed on embedded microcontrollers, making computational implementation, software modularity, and portability important considerations during controller development.

Motivated by these challenges, this dissertation investigates the implementation and validation of two widely used PMSM control strategies within the PSIM simulation environment. A central focus of the work is the development of the control algorithms in a modular form, using C-based functional blocks designed to facilitate their subsequent transfer to ARM-based embedded platforms. In addition to comparing the performance characteristics of DTC and FOC, the work also addresses practical aspects such as parameter identification and rotor position estimation, providing a comprehensive framework for high-performance PMSM control and its future embedded implementation.

1.2 Approach

The primary objective of this dissertation is to study, implement, and validate advanced control techniques for Permanent Magnet Synchronous Motors through simulation-based development in PSIM, while developing the control software in a modular structure suitable for subsequent implementation on ARM-based platforms. Particular emphasis is placed on comparing Direct Torque Control and Field-Oriented Control while establishing a common drive architecture that enables a fair evaluation of both approaches.

The development and validation of the proposed control strategies are carried out using the PSIM simulation environment. PSIM was selected due to its strong focus on power electronics and electric drive applications, providing dedicated component libraries for power converters, electrical machines, and control systems. Its efficient simulation engine enables rapid evaluation of controller performance while allowing the electrical power stage and the digital control algorithms to be modeled within a unified framework. Furthermore, its support for C-based control blocks enables the controller algorithms to be developed in a form closer to their intended embedded implementation. This makes PSIM particularly well suited for implementing and comparing both Direct Torque Control and Field-Oriented Control under identical operating conditions, while simultaneously providing a development and validation stage before transferring the control code to an ARM platform.

The work begins with a comprehensive review of the theoretical foundations of brushless motor technology, including the operating principles of PMSMs, existing control strategies, rotor position detection methods, and mathematical modeling in different reference frames. This theoretical background provides the basis for the subsequent controller implementations.

The practical component focuses on the realization of complete DTC and FOC drive systems, including signal conditioning, coordinate transformations, current regulation, torque and flux estimation, switching logic, and Space Vector Pulse Width Modulation. The control algorithms are organized into modular functional blocks, with the objective of maintaining a clear separation between the individual control functions and facilitating their reuse and subsequent integration into embedded software. The implemented controllers are validated through simulation experiments that evaluate speed tracking, torque production, transient response, and steady-state behavior under different operating conditions.

Building upon the baseline FOC implementation, additional advanced functionalities are investigated. These include parameter identification methods for the stator resistance, inductances, permanent magnet flux linkage, and mechanical parameters, as well as Hall sensor based rotor angle and speed estimation techniques. These features further improve the applicability of the controller in practical implementations where accurate machine parameters and reliable position information are essential.

The resulting work provides both a theoretical and practical comparison of modern PMSM control techniques while establishing a modular control architecture developed and validated in PSIM with subsequent deployment on ARM-based embedded platforms as a central design objective.

1.3 Dissertation Structure

This dissertation is organized into five chapters.

Chapter 1 introduces the motivation for the work, defines the objectives of the dissertation, and outlines the adopted methodology and overall document structure.

Chapter 2 – Literature Review presents the theoretical background required for the development of the proposed controllers. It introduces brushless DC motors and Permanent Magnet Synchronous Motors, describes their construction and operating principles, reviews existing control strategies, discusses rotor position detection methods, and derives the mathematical models used throughout the remainder of the dissertation.

Chapter 3 – Practical Work: Controller Model Implementations in PSIM describes the implementation of the PMSM drive architecture and the realization of Direct Torque Control and Field-Oriented Control within the PSIM environment. Particular attention is given to the modular implementation of the control algorithms in C-based functional blocks, with the objective of facilitating their subsequent transfer to ARM-based embedded platforms. The chapter also presents validation results for both controllers and concludes with a comparative analysis of their performance.

Chapter 4 – Advanced Features for Field-Oriented Control extends the baseline FOC implementation by introducing parameter identification procedures for the electrical and mechanical machine parameters together with Hall sensor based angle and speed estimation techniques.

Chapter 5 summarizes the main contributions and findings of the dissertation, discusses its limitations, and proposes directions for future work. Additional implementation details and supplementary validation material are provided in the appendices.

Chapter 2

Literature Review

2.1 Introduction to Brushless DC Motors

2.1.1 Definition and Scope of BLDC Motors

The permanent magnet brushless DC machine (PMBDCM) is the technical name given to permanent magnet synchronous motors (PMSMs) that feature a trapezoidal-induced electromotive force (EMF)[1, p. 457]. The distinction between trapezoidal and sinusoidal back-EMF waveforms is illustrated in Figure 2.1.

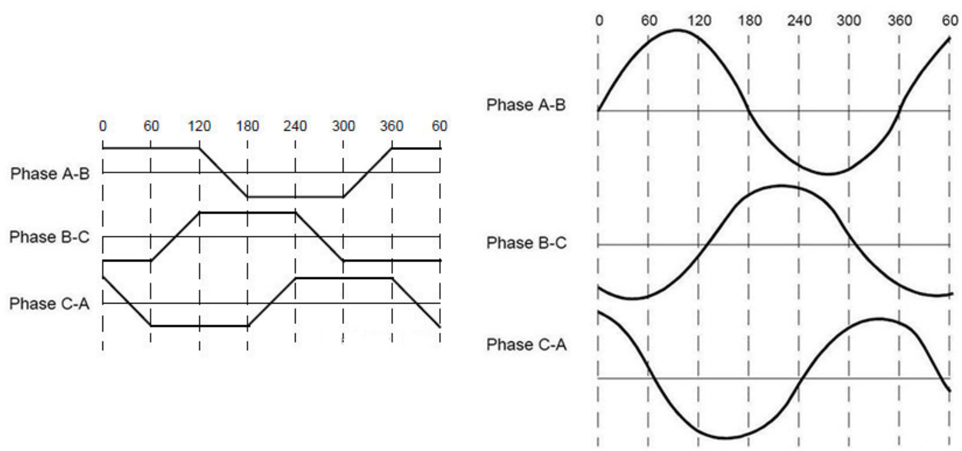


Figure 2.1: Comparison between trapezoidal (left) and sinusoidal (right) line-to-line back-EMF waveforms of a three-phase permanent magnet machine. Adapted from [2].

This machine is commonly referred to in industrial circles as a PM brushless DC machine because the phase voltage equation mirrors the armature voltage equation of a traditional DC machine, while crucially lacking brushes and a commutator[1, p. 459].

The primary characteristic that defines the operation of a PMBDCM is the method of current control: to energize the machine phases, the control system only needs to track the beginning and end of the constant flat portion of the induced EMF[1, p. 457]. This process requires information on just six discrete positions, corresponding to every 60 electrical degrees, for energizing the three stator phases[1, p. 462]. The desired phase current waveform for the BLDC is rectangular and 120 electrical degrees wide in each half cycle[1, p. 476].

BLDC drives span a wide scope of applications, particularly in high-volume, cost-sensitive sectors, including:

- High-performance applications like servos and robotics.[3, p. 29]
- High-volume commercial applications such as fans, pumps, washers, dryers, treadmills, wheelchairs, and automotives.[1, p. 497]
- Embedded control platforms, particularly in low-voltage ranges (12–48 V), where single-chip control solutions are popular[1, p. 559].

2.1.2 Historical Transition: From Brushed DC to Solid-State Commutation

The modern BLDC motor drive represents the culmination of two pivotal developments: the advent of permanent magnets (PMs) and the innovation of solid-state electronic commutation.

The Era of Brushed DC In the early days of electrical power, systems were based on direct current (DC). Although AC power systems began overtaking DC systems in the 1890s, DC motors remained a significant fraction of electrical machines purchased annually until the 1960s[4, p. 123].

However, traditional DC machines suffered from inherent disadvantages[3, p. 29], including higher manufacturing cost and rotor inertia and maintenance issues due to the commutators and brushes.

The shift to electronic commutation began fundamentaly in the 1950s with the development of PM DC field excitation, which led to the creation of more compact DC machines by replacing the heavy electromagnetic poles. Simultaneously, the advent of switching power transistors and silicon-controlled rectifier (SCR) devices in the late 1950s paved the way for solid-state technology[1, p. 31].

These semiconductor developments enabled the replacement of the mechanical commutator with an electronic commutator—an inverter. This solid-state commutation allowed the armature (winding that carries the main current) to be placed on the stator, facilitating better cooling and higher voltage operation due to increased space for insulation[1, p. 31]. This technological evolution contributed directly to the widespread adoption of AC machines (including BLDC) over DC machines in variable-speed applications. The continuous evolution of semiconductor devices, such as the introduction of high-power IGBTs and GCTs in the 1990s, further accelerated the development of high-power converters and drives[5, p. 3].

2.1.3 Core Advantages of BLDC Technology

BLDC drives offer several advantages over their predecessors, driven mainly by the permanent magnet rotor and electronic commutation system.

About the maintenance and reliability, the elimination of the brushes and mechanical commutator entirely removes the most common wear components found in traditional DC motors, leading to enhanced system reliability and significantly reduced maintenance requirements. Furthermore, certain BLDC converter topologies (such as half-wave operation designs) inherently resist electrical failures, offering high reliability because the switch is always in series with the phase winding, which prevents destructive shoot-through faults. These topologies can even be designed to operate at reduced capacity if a single switch or phase winding fails[1, p. 501].

When it comes to efficiency and power density,BLDC motors offer superior power density compared to other permanent magnet alternatives. Specifically, a PM brushless DC machine has a 15.4% higher power density than a Permanent Magnet Synchronous Machine (PMSM) for equivalent resistive losses[1, p. 50]. This density advantage is linked to the high ratio of the RMS value to the peak value of the flux linkages utilized by the machine.

Efficiency is also enhanced through the reduction of mechanical losses associated with brushes and commutation. For specific drive topologies (like the split-supply converter in half-wave operation), losses can be nearly halved compared to full-wave inverter drives, due to using fewer active switches (one switch and diode per phase) during operation[1, p. 501].

Concerning the controllability, the shift to electronic commutation facilitated modern, high-dynamic control strategies:

- Simple Commutation, because the PMBDCM control scheme is simpler than PMSM control, requiring only positional information at six discrete electrical points (60-degree increments) for accurate current switching.

- High Dynamic Control for applications demanding high dynamic performance, advanced control techniques like Field Oriented Control (FOC) and Direct Torque Control (DTC) are often employed[5, p. 321]. These schemes are effective because they decouple the torque-producing and flux-producing components of the stator current, allowing independent control of torque and magnetic flux, analogous to the superior control found in a separately excited DC motor drive[1, p. 279]. FOC and DTC can be implemented cost-effectively using digital controllers.

2.1.4 Relevance to Embedded and Motion Control Applications

The attributes of BLDC motors make them highly relevant for modern embedded systems and robotics, where precise motion control, high reliability, and compactness are critical needs.

PM drives are widely used in specialized high-performance applications such as servos and robotics that require precision motion. Robotics specifically need controlled acceleration and deceleration, necessitating sophisticated drive systems capable of controlled starts and stops[1, p. 279] (e.g., regenerative braking capability for fast deceleration). The high dynamic response enabled by modern FOC and DTC control schemes ensures the precision required for robotic actuation and instantaneous torque control below base speed[1, p. 353].

Cost-Effectiveness and Embedded Implementation The requirement for cost-effectiveness and reliability in embedded platforms is met through architectural simplifications and advanced control algorithms:

- Sensorless Control: Eliminating the rotor position sensor, which can be expensive and less reliable than the motor itself, is a major focus for enhancing cost and reliability, broadening BLDC applicability to cost-sensitive embedded systems[1, p. 423].
- Low-Cost Solutions: For low-voltage embedded applications (e.g., 12 V to 48 V), single-chip solutions are available for controlling PM brushless DC motors, supporting both sensor-based and sensorless operation[1, p. 559].
- Integration: The control schemes like FOC and DTC can be cost-effectively implemented in real-time by a digital controller, making high-performance control accessible on embedded platforms[5, p. 321].

2.2 Structure and Working of BLDC Motors

2.2.1 Mechanical Structure

This section provides a physical description of the Brushless DC (BLDC) machine's mechanical structure and outlines key design options, thereby setting the stage for understanding later performance, back-electromotive force (EMF), and control implications.

Core Components

The structure of a BLDC motor generally consists of stationary and rotating parts [6, p. 27]. The main components include:

- **Stator core:** Formed by a stack of soft iron laminations, containing slots and teeth [6, p. 97].
- **Windings:** Coils of magnet wire placed within the stator slots [6, p. 44].
- **Rotor with permanent magnets (PMs):** The rotating assembly containing permanent magnets [7, p. 215].
- **Shaft:** The central component, providing rotation and transmitting torque [6, p. 36].
- **Bearings:** Components supporting the shaft rotation, such as ball bearings [8, p. 8].
- **Housing (frame):** Provides the mechanical structure, protects internal parts, and facilitates cooling (e.g. aluminum extrusion) [6, p. 30].
- **Optional sensors:** Position sensors, such as Hall sensors or encoders, used to detect rotor position for synchronization and commutation [7, p. 28][8, p. 48].

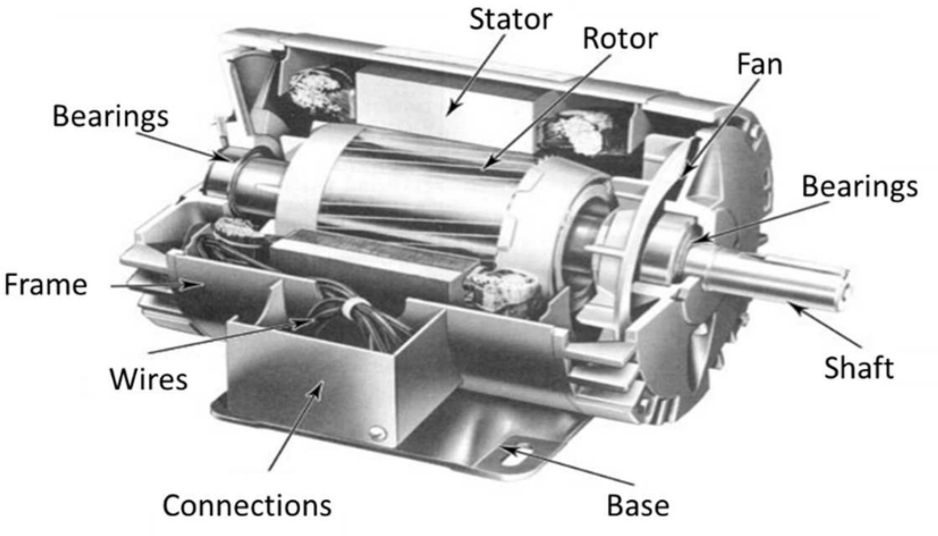


Figure 2.2: Generic cross-sectional view of a rotating electric machine illustrating the main mechanical components: stator, rotor, shaft, bearings, housing/frame, and internal wiring. Although shown here for an induction-type machine, the mechanical structure is representative of BLDC motors as well [9].

2.2.1.1 Stator Windings

The design of the stator windings is a critical choice that significantly impacts the motor's performance characteristics, including the shape of the back-EMF waveform and thermal management. The geometric distribution of coils strongly influences not only the magnetic field distribution in the air gap but also the harmonic content of the resulting back-EMF, as illustrated in Fig. 2.1.

Winding type: concentrated vs. distributed [8, p. 26] This distinction is illustrated in Fig. 2.3.

Coil pitch, slot fill, and end-turns

- **Slot-fill factor** is a metric for the density of copper packed into the stator slot. A high slot-fill factor is desirable because it reduces phase resistance (I^2R loss) and thus lowers the temperature rise [6, p. 101].
- **Coil pitch** refers to the span of the coil. Using a short-pitch (chorded) winding, where the coil span is less than a full pole pitch, can help to weaken torque harmonics and minimize the required end-turn length [7, p. 27].
- **End-turns**, i.e. the copper connections extending beyond the lamination stack, are a source of copper loss (resistive loss) and contribute to the leakage inductance [6, p. 554, p. 686]. Minimizing end-turn length is crucial for improving efficiency and reducing resistive losses [1, p. 105].

Table 2.1: Comparison of concentrated and distributed stator windings

Feature	**Concentrated winding (single-tooth)**	**Distributed winding**
Physical layout	All conductors are in one slot, spanning one pole. Coils do not overlap [8, p. 26].	Conductors are spread over several slots. Windings of different phases typically overlap [7, p. 27].
EMF shape impact	Produces a more trapezoidal EMF [1, p. 53][10, p. 65].	Produces a more sinusoidal EMF [10, p. 88].
Cogging impact	Lower cogging torque if fractional-slot, due to many cogging cycles per revolution [1, p. 53][1, p. 109].	Lower harmonic content, generally superior performance [8, p. 26].
Manufacturability	Simpler and cheaper, with less copper and shorter end windings.	More complex manufacturing.
End-turn length	Shorter end windings.	Longer end windings [8, p. 26].

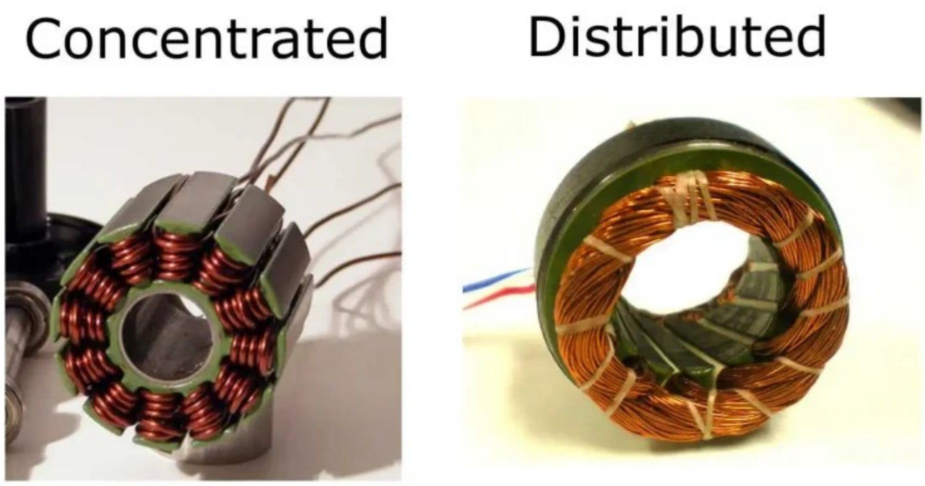


Figure 2.3: [11]Side-by-side comparison of concentrated (left) and distributed (right) stator windings. Concentrated windings place each coil around a single tooth, while distributed windings span multiple slots. The corresponding back-EMF waveforms for these winding types are shown in Fig. 2.1.

Neutral/Topology: Star/Delta/Isolated Neutral The way phase windings are connected affects voltage requirements, fault modes, and sensing capabilities:

- Star (Y) Connection: This is the most frequently adopted configuration [7, p. 27]. In a standard three-phase system, if the neutral point is not brought out, the phase voltages are difficult to detect [7, p. 38]. The equivalent line inductance of the winding in this configuration is calculated as $L_a = 2(L-M)$ [7, p. 40].

- Delta (Δ) Connection: If triplen harmonics exist in the phase EMF waveforms (often due to winding pitch or magnet pole arc design), connecting in delta will cause an undesirable circulating current, significantly increasing losses [10, p. 75] [6, p. 278].

- Line-to-Line Sensing: In a 3-wire connection (star or delta), only two line currents need to be measured, as the third current is determined by the constraint ($i_A + i_B + i_C = 0$) [6, p. 276] [7, p. 27].

- Isolated Neutral / Modular Windings: Designing specialized windings, such as a "modular" winding where no slot contains more than one coilside, can provide a high degree of isolation between phases, which is beneficial for fault tolerance [6, p. 100].

2.2.1.2 Rotor and Magnets

The rotor assembly holds the permanent magnets (PMs) that generate the excitation flux. Its design strongly influences speed capability and flux characteristics. The choice of magnet material is a trade-off between energy density, thermal limits, and cost.

Magnet materials: NdFeB vs. ferrite

- **Neodymium-iron-boron (NdFeB):** Offers the highest energy product, enabling smaller frame sizes for a given output [7, p. 17]. NdFeB magnets are temperature-sensitive, the risk of demagnetization increases above 100°C as the knee point of the demagnetization curve shifts [10, p. 33, p. 42].

- **Ferrite:** Considerably cheaper and exhibits lower remanent flux density, but is more resistant to temperature-induced demagnetization up to approximately 120°C [6, p. 660, p. 694].

Mounting: surface-mounted vs. interior

- **Surface-mounted PM (SPM):** The magnets are typically bonded directly to the surface of the rotor iron core. SPM machines are generally considered non-salient, meaning that the direct- and quadrature-axis inductances are approximately equal ($L_d \approx L_q$) [7, p. 27]. At high speeds, retaining sleeves or bands are required to withstand centrifugal forces [8, p. 24] [1, p. 34].

- **Interior PM (IPM):** Magnets are embedded within the rotor laminations; the structure is mechanically robust and characterized by high saliency ($L_q > L_d$), which supports the production of reluctance torque and effective flux weakening. IPM rotors are also less susceptible to demagnetization compared to surface-mounted types [6, p. 31].

The structural differences between surface-mounted PM (SPM) and interior PM (IPM) machines are illustrated in Fig. 2.4.

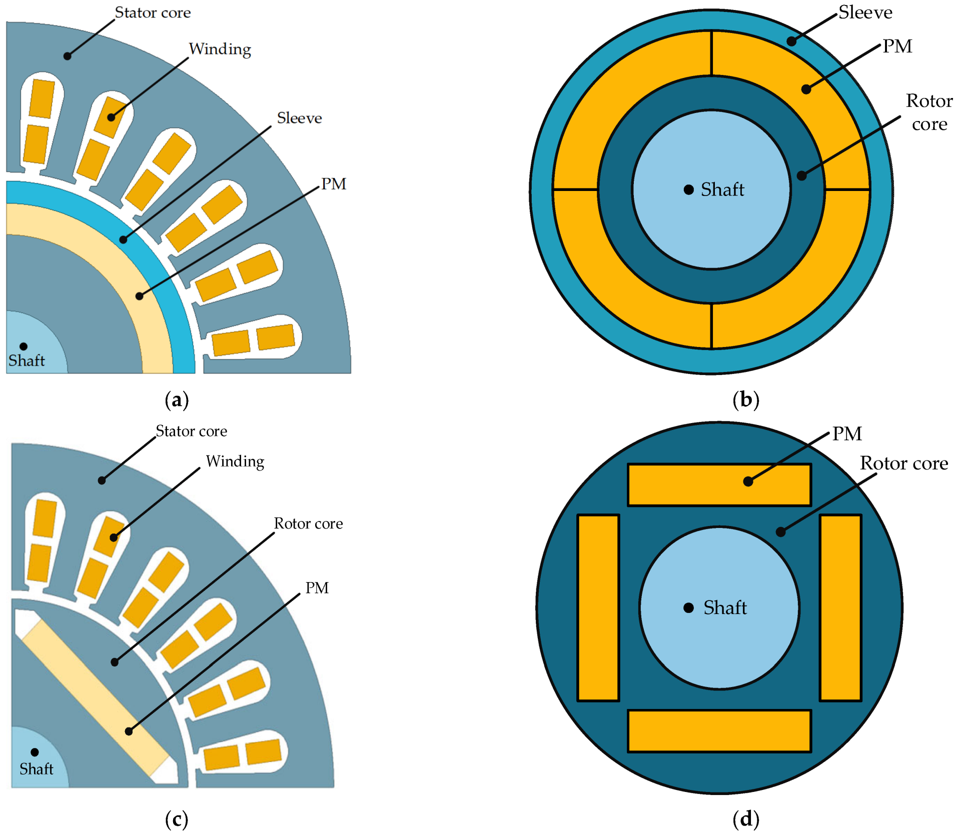


Figure 2.4: Comparison of surface-mounted permanent magnet (SPM) and interior permanent magnet (IPM) rotor structures. Figures (a)–(b) show SPM geometries with magnets mounted on the rotor surface and retained by a sleeve; Figures (c)–(d) show IPM geometries with buried magnets embedded within the rotor core. Adapted from [12].

Magnetization Pattern: radial vs. Halbach The pattern of magnet flux generation dictates the air-gap flux waveform:

- **Radial Magnetization:** The flux is directed radially (perpendicular to the rotor surface). This typically produces an air-gap flux density distribution that approximates a trapezoidal or rectangular wave, desired for standard BLDC drives[1, p. 24].

- **Halbach Array:** This advanced magnetization pattern uses an array of magnets magnetized in various directions to approximate a rotating field [1, p. 28]. Halbach magnets are capable of producing a stronger magnetic field and generating a highly sinusoidally distributed flux in the air gap, thereby reducing spatial harmonics and improving EMF waveform quality, particularly in slotless designs [6, p. 59, p. 689].

2.2.1.3 Phase Count

- **Three-phase systems** are by far the most common choice, offering high efficiency, good utilization of copper, and favorable characteristics compared to single-phase or low-phase-count systems [6, p. 71].
- **Two-phase motors** are used in applications such as low-duty fan drives, where a simpler drive circuit (requiring only two Hall sensors and two power transistors) is cost-effective [6, p. 71].
- **Multi-phase machines** (e.g. six or nine phases) are specialized designs often considered for fault tolerance and specific power requirements [1, p. 46].
- The **commutation logic** and sensor configuration must match the chosen phase count [8, p. 48]. Traditional three-phase BLDC motors rely on position sensors (such as Hall sensors) to provide six commutation signals per electrical cycle. Non-three-phase designs require corresponding changes in commutation logic and sensor configuration [1, p. 51].

2.2.1.4 Inner-Rotor vs. Outer-Rotor Configuration

Inner-rotor (rotor inside stator) The inner-rotor configuration is common in high-performance servo applications [6, p. 31–36]. Its main characteristics are:

- **Low rotor inertia:** The rotor has a relatively small diameter, leading to low inertia and enabling high dynamic response [6, p. 688].
- **Favourable cooling of the stator:** The heat-generating stator winding and core are stationary and in close thermal proximity to the frame, which allows efficient heat removal by conduction, convection, or liquid cooling.
- **High-speed capability:** The configuration is generally favoured for high-speed operation [10, p. 55].

Outer-rotor (rotor surrounding stator) The outer-rotor structure is popular in applications such as fan drives, disk drives, and electric bicycles [7, p. 6]. Its main features are:

- **High torque at low speed:** The magnets are positioned further from the shaft centre, increasing the air-gap radius and enabling high torque output at lower speeds.

- **Higher inertia:** Due to the large rotor diameter, the inertia is relatively high. This can be advantageous in constant-speed applications, as it helps to dampen speed fluctuations and provide smoother operation, at the expense of slower transient response [6, p. 33–36].
- **Application tie:** The outer-rotor structure is frequently used in electric bicycles because of its ability to generate high torque.

The fundamental geometric differences between these two configurations are illustrated in Fig. 2.5.

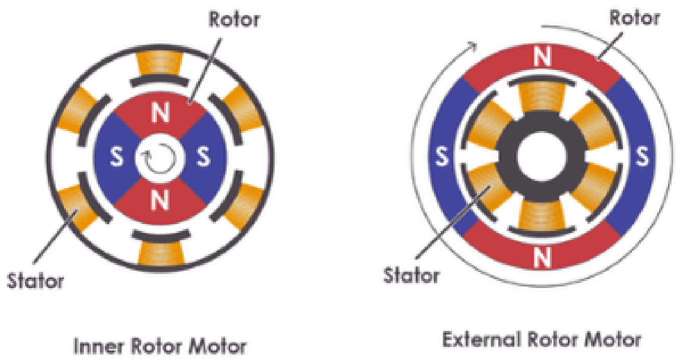


Figure 2.5: Comparison of inner-rotor and outer-rotor BLDC motor configurations, showing the relative position of rotor and stator. Adapted from [13].

2.2.1.5 Slot/Pole Selection and Cogging Mitigation

Cogging torque is a component of torque ripple caused by the interaction between the PM flux and the stator slot openings, leading to reluctance variation [7, p. 128]. It must be mitigated to reduce acoustic noise and torque pulsation [6, p. 24]. Common mitigation strategies include:

- **Fractional-slot vs. integer-slot designs:** Using fractional slots per pole increases the least common multiple (LCM) of the numbers of poles and slots, thereby raising the fundamental frequency of the cogging torque [7, p. 130]. Since cogging torque amplitude decreases with increasing frequency, this approach reduces the torque ripple magnitude [1, p. 48].
- **Skewing:** Skewing the stator laminations or rotor magnets axially by one slot pitch (or less) smooths the flux variations. While skewing effectively reduces cogging torque, it also reduces the fundamental EMF and torque, and increases leakage inductance and manufacturing cost. It additionally helps to suppress ripple caused by permeance harmonics [6, p. 108].
- **Dummy or (almost) closed slots:** Using dummy slots (auxiliary slots or grooves) or closed /almost-closed stator slots helps to achieve a more uniform flux distribution and suppress cogging torque.

- **Magnet shaping/shifting:** Methods such as profiling the magnets, using "breadloaf" magnet shapes, or adjusting magnet width can reduce cogging torque and shape the air-gap flux distribution.

Minimising cogging torque is one of the key measures to reduce overall torque ripple and the associated acoustic noise and vibration, which is particularly important in high-performance drives.

2.2.1.6 Air-Gap, Saliency, and Saturation

A larger air-gap length is typically necessary in PM machines to accommodate the magnets and any protective sleeves [10, p. 153]. Increasing the air-gap generally reduces the stator inductance and torque density [6, p. 686].

Saliency, i.e. the condition $L_q \neq L_d$, enables the generation of reluctance torque and is essential for deep flux-weakening operation and for many sensorless control methods that exploit magnetic anisotropy [6, p. 100].

Magnetic saturation reduces the incremental inductance and torque constant and introduces nonlinearities into the torque-current relationship [1, p. 92]. These nonlinearities must be considered in high-performance control and observer design.

2.2.1.7 Thermal and Packaging Considerations

The dominant losses in BLDC machines are usually Joule (copper) losses in the stator windings [6, p. 554]. In inner-rotor radial-flux machines, the heat generated in the stator core and windings is efficiently conducted away to the frame (housing) because these components are stationary and close to the exterior [6, p. 31]. Heat removal can involve conduction through mounting surfaces, heat sinks, forced-air cooling, or liquid coolants.

Potting or impregnation, i.e. the application of varnish or specialized encapsulating compounds, serves both to insulate the windings and to improve heat transfer by providing thermally conductive pathways away from the coils. This enhanced thermal management directly affects the continuous torque rating of the machine and allows higher power densities [6, p. 26].

Equations

Two qualitative relationships are particularly relevant for the subsequent analysis of BLDC machines:

1. **Back-EMF proportionality:** The induced EMF is proportional to the rotational speed ω and the total flux Φ linking the winding [10, p. 65]:

$$e \propto \omega \Phi.$$

2. **Qualitative link between winding and EMF shape:** The shape of the back-EMF depends on the winding distribution and the magnet flux profile:

    - Concentrated windings tend to produce a trapezoidal EMF [7, p. 26].
    - Distributed windings tend to produce a sinusoidal EMF [10, p. 87].

The magnetic circuit governing torque and EMF generation can be viewed analogously to a fluid-dynamic system. The permanent magnets act as a constant "pressure source" (flux Φ), while the windings behave like turbine blades that convert this flux into mechanical power. The winding distribution shapes the EMF waveform, and the rotational speed ω determines the magnitude of the counter-induced voltage.

2.2.2 Operating Principle

The operation of the BLDC motor relies on the electronic commutation of stator currents, which establishes a rotating magnetic field that interacts with the permanent magnet (PM) flux of the rotor to produce torque. A key aspect of this process is the shape and timing of the resulting back-EMF waveform, as it governs the commutation instants required to generate accurately timed stator currents typically implemented via PWM to ensure continuous and efficient rotation [14, p. 1, p. 5].

Electromagnetic Torque: Fundamental Principles

Torque in a BLDC machine is generated through the interaction between the magnetic field of the stator windings and the magnetic field of the rotor magnets. Maximum torque occurs when these two fields are orthogonal (90°), with the torque amplitude decreasing as the fields align [1, p. 51].

The developed torque (T_e) is proportional to the stator current (I) through the torque constant (K_t), which specifies the torque produced per ampere of phase current [14, p. 17]. The back-EMF constant (K_e), defined analogously, relates the induced EMF to rotor speed. Under constant field excitation (flux linkage λ_{af}), the electromagnetic torque may be expressed as a function of the quadrature-axis current component (i_{qs}) [1, p. 282].

The literature affirms that the proportional relationship between K_t and K_e holds for the fundamental representation of trapezoidal BLDC machines [15, p. 529][1, p. 542]. However, the sources do not provide a complete derivation of the commonly cited equivalence $K_t = K_e$ in SI units.

Back-EMF

Back-EMF is the voltage induced in the stator windings as a result of rotor motion and, as required by Lenz's Law, opposes the applied stator voltage [14, p. 12]. Its characteristics are central to commutation strategies and sensorless control.

- **Physical origin and dependence:** Back-EMF arises from the PM flux cutting the stator conductors [1, p. 54]. Its amplitude is proportional to the rotor angular velocity (ω) and the magnetic field density (B), following the relationship $E \propto NlrB\omega$, where N and l denote winding parameters. Once the motor is designed, the back-EMF constant K_e is fixed, such that the induced voltage depends solely on rotor speed [14, p. 11].

- **Waveform shape:** The shape of the back-EMF waveform is determined by the stator winding distribution and magnet geometry [1, p. 21]. Concentrated windings typically yield a trapezoidal back-EMF (as in BLDC machines), whereas distributed windings produce a sinusoidal back-EMF, characteristic of PMSM machines [15, p. 383][1, p. 48]. These structural differences give rise to the distinct EMF profiles observed in practice [1, p. 56]. See Fig. 2.1.

- **Phase vs. line-to-line sensing:** Zero-crossings of the back-EMF occur when the instantaneous polarity of the phase voltage changes relative to the neutral point. Detecting these zero-crossings in the non-energized phase is a fundamental method used to determine commutation timing in sensorless BLDC drives.

Electronic Commutation (Six-Step)

Electronic commutation controls the directional flow of current in the stator windings to sustain continuous torque production [14, p. 1, p. 2]. In six-step commutation, switching events are spaced every 60° of electrical rotation [14, p. 8] (Fig. 2.6).

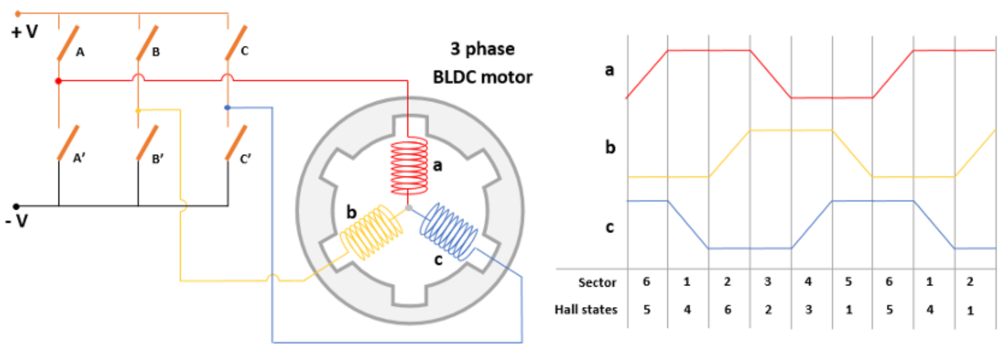


Figure 2.6: Six-step commutation sequence for a three-phase BLDC motor, showing switching states and corresponding Hall sensor signals [16].

- **120° conduction:** At any instant, two of the three motor phases conduct (one sourcing current and one sinking), while the third remains floating. Each phase conducts for 120° of electrical angle in each cycle [5, p. 312].

- **Sequence logic:** The switching sequence is selected based on rotor position, typically measured using Hall-effect sensors or estimated from back-EMF. These signals determine which inverter switches are activated at each commutation step [14, p. 8].

- **Commutation advance:** At high speeds, the current must be applied earlier in the electrical cycle to compensate for phase lag. This advance becomes necessary when the back-EMF approaches or exceeds the available phase voltage, requiring the injection of demagnetizing current to maintain controllability [1, p. 284, p. 475, p. 480].

Power Stage & Current Paths

The BLDC motor is supplied by a three-phase voltage-source inverter composed of three half-bridge legs (switches Q0 to Q5) connected to the DC link.

During each 120° conduction interval, one phase is energized from the DC link while the complementary switch conducts the return current path, establishing torque production (Fig. 2.7).

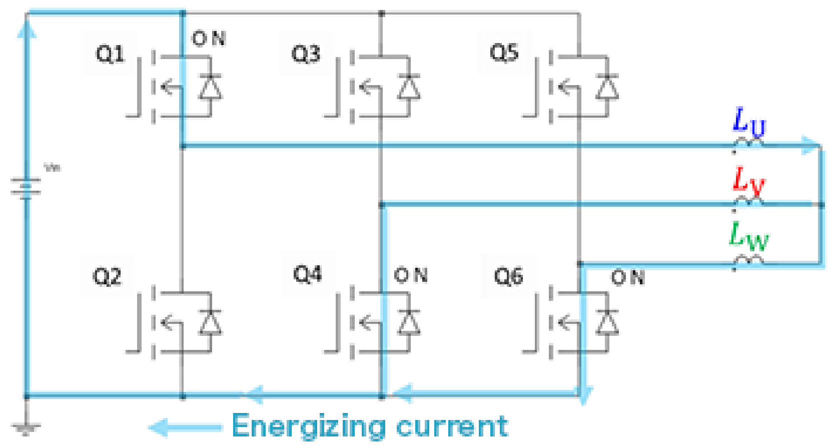


Figure 2.7: Energizing current path during six-step commutation. The conducting switches establish a direct current flow from the DC link through phase L_U, enabling torque production. Adapted from ROHM Semiconductor [17].

- **Current freewheeling:** When a conducting switch turns off, the inductive phase current must continue flowing [1, p. 139]. This current is redirected through the antiparallel diodes of the inverter, forming a freewheeling path (Fig. 2.8). Such diode conduction also contributes to low-speed voltage distortion [1, p. 152].

- **Dead-time effect:** Dead time (t_d) is intentionally inserted between complementary switch transitions to prevent shoot-through faults [15, p. 189]. However, dead time reduces the effective fundamental voltage and introduces low-frequency distortion [15, p. 191], resulting in nonlinear voltage transfer behavior particularly detrimental at low speeds [1, p. 135].

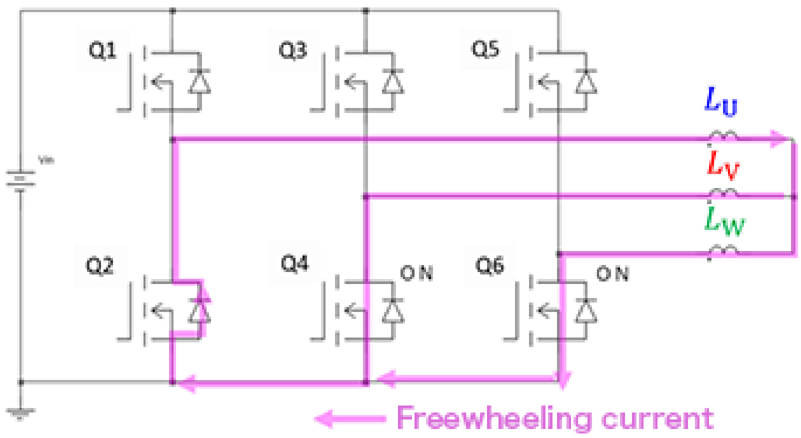


Figure 2.8: Freewheeling current path when the conducting switch is turned off. The inductive phase current continues flowing through the antiparallel diodes, illustrating the need for dead time and explaining inverter voltage distortion at low speed. Adapted from ROHM Semiconductor [17].

PWM Strategy

- **PWM modulation:** PWM controls the average stator voltage and thereby regulates motor speed.
- **Sampling:** PWM signals may be generated using symmetric sampling (carrier-frequency sampling) or unsymmetric sampling (twice the carrier frequency). Symmetric sampling produces gate pulses symmetric about the midpoint of the PWM period [1, p. 170].
- **SVPWM:** Space Vector Modulation (SVM) offers a wider linear modulation range and superior DC-bus utilization relative to sinusoidal PWM [5, p. 48]. By precisely selecting voltage vectors, SVM enables high-performance control irrespective of the back-EMF waveform [1, p. 181–182].

Current/Torque Ripple Origins and Mitigation

Torque ripple in BLDC drives arises mainly from imperfections in current waveforms due to commutation events and inherent machine characteristics [1, p. 476].

Origins

- **Misalignment and commutation:** Torque pulsations occur primarily at commutation instants [1, p. 475, p. 525]. Large deviations from an ideal trapezoidal back-EMF amplify these disturbances [1, p. 51].
- **Finite switching frequency:** Limited switching frequency introduces harmonic components into the current waveform, contributing significantly to torque ripple [15, p. 249, p. 707].

Mitigation

- **Current shaping / misalignment correction:** Coordinating the currents of the incoming and outgoing phases during commutation helps maintain constant torque, particularly when the shaping reflects the actual flux distribution [1, p. 156].

- **Higher switching frequency:** Increasing the switching frequency reduces the amplitude of current ripple.

- **Commutation dwell and fractional-slot designs:** Concentrated windings—analogous to fractional-slot configurations depending on pole count—reduce cogging torque due to their high cogging-cycle-per-revolution (CPMR) characteristics [1, p. 48].

Links Forward

- Trapezoidal vs. sinusoidal/FOC: Trapezoidal BLDC control is simple but susceptible to commutation-induced torque ripple. In contrast, PMSM drives using sinusoidal currents and FOC achieve significantly smoother torque production [1, p. 48, p. 280].

- Back-EMF and sensorless control: The trapezoidal back-EMF waveform is advantageous for sensorless operation, as its zero-crossings in the unenergized phase provide reliable commutation timing information [14, p. 13].

2.3 Control Strategies for PMSM

2.3.1 Trapezoidal Control

Trapezoidal control, often associated with Brushless DC (BLDC) motor operation, relies on a Six-Step Commutation strategy to drive the motor [14, p. 5]. This method is characterized by 120° conduction, where only two of the three motor windings are energized at any given instant, while the third winding remains in a non-energized or "floating" condition [7, p. 30].

2.3.1.1 Phase State Sequence and Rotor Position

Continuous rotation is obtained by advancing the commutation state in synchrony with rotor position. In sensored drives, three Hall devices spaced at either 60° or 120° electrical generate transitions every 60°, defining the six discrete states of one electrical cycle. In sensorless drives, the same timing is inferred from the zero crossing of the floating phase back-EMF (ZCD), with appropriate blanking and filtering to reject switching transients. Because Hall transitions occur every 60° electrical, the position estimate is quantized; many implementations interpolate between edges (e.g., time-based linearization) to reduce current-alignment error and torque ripple. ZCD detection is performed relative to a (virtual) neutral point reconstructed from the three phase voltages; the choice of divider impedance and filtering strongly affects low-speed SNR. [14, p. 4]

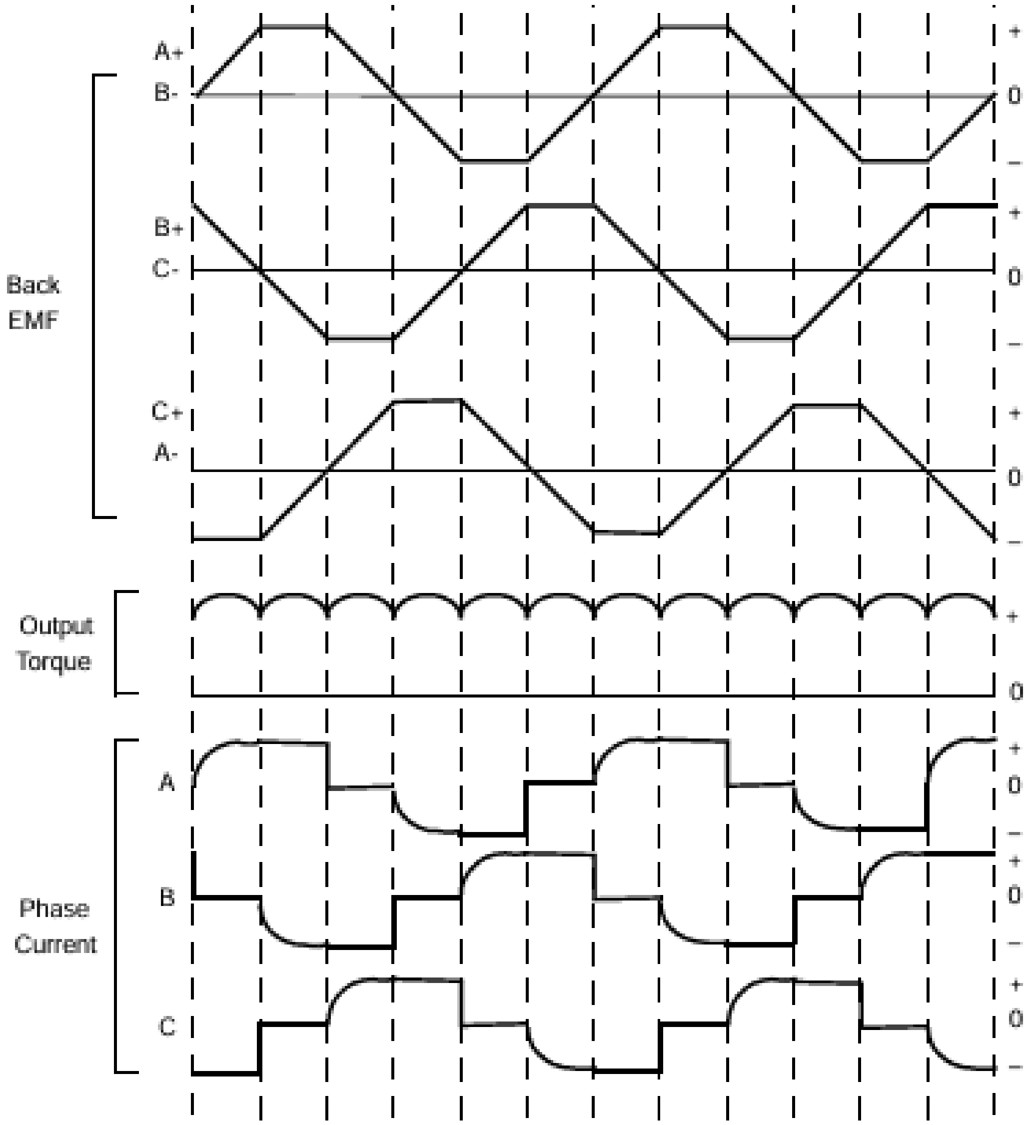


Figure 2.9: Six-step trapezoidal BLDC commutation over two electrical cycles: Hall sensor outputs (A,B,C), phase back-EMFs (A,B,C), resulting electromagnetic torque ripple, and phase currents (A,B,C). Sequence numbers (1)–(6) indicate the six commutation states per electrical cycle. Adapted from Microchip Application Note AN885 [14].

The resulting relationship between Hall signals, back-EMF, torque ripple, and phase currents is illustrated in Fig. 2.9.

2.3.1.2 Commutation Timing and Advanced Conduction

At higher speed the phase back-EMF approaches the DC-link voltage, limiting current build-up. A lead (advanced) conduction angle is therefore applied so current rises before the EMF peak. This acts as field-weakening by reducing the effective flux linkage during commutation, extending the constant-power region at the cost of added commutation-torque ripple and increased sensitivity to parameter errors. Finite inductance causes commutation overlap (current persisting in the outgoing phase), which modifies the effective conduction interval and contributes to torque ripple at step boundaries. Unlike PMSM field-weakening with negative i_d, commutation advance in trapezoidal drives alters the current–EMF timing rather than actively reducing flux; therefore the achievable constant-power range and efficiency differ from FOC-based field-weakening. [7, p. 77–79]

2.3.1.3 PWM Strategy and Dead-Time

Torque and speed are regulated by PWM of the inverter legs. Center-aligned PWM is commonly preferred because sampling at the carrier midpoint reduces measurement bias from switching transients [7, p. 18][14, p. 11]. Practical modulation choices include high-side chopping with the complementary device held on, or full-bridge chopping of both devices in a leg; the choice affects current ripple, sensing windows, and EMI [18, p. 386]. At high modulation indices, 1-shunt and some 2-shunt reconstructions exhibit missing sampling windows around zero vectors; gating patterns are often adjusted (or in-line sensing used) to guarantee valid current samples. Dead-time inserted to prevent shoot-through introduces a nonlinear voltage error that is most detrimental at low speed and small duty ratios; compensation or calibration is therefore recommended [18, p. 389][7, p. 31]. Dead-time compensation can be implemented via inverse dead-time mapping or table-based calibration using measured current polarity.

2.3.1.4 Current Regulation Options

Current regulation enforces thermal limits and delivers commanded torque [7, p. 85]. Hysteresis control is simple and fast but yields a variable switching frequency [18, p. 213, p. 360]. Fixed-frequency PI regulators in the *abc* frame (or *dq* for advanced drives) paired with PWM provide predictable spectra. Accurate current reconstruction requires synchronized sampling and careful placement of shunts (DC-link, phase-leg, or in-line), since narrow sensing windows at high modulation can otherwise distort measurements. [18, p. 461, p. 614] [7, p. 228, p. 236]

2.3.1.5 Startup Procedure

Because back-EMF is negligible at standstill, sensorless drives use a deterministic start-up [7, p. 201]: (i) rotor alignment to a known position, (ii) open-loop acceleration with a ramped electrical frequency, and (iii) hand-over to closed-loop commutation once the back-EMF magnitude and SNR support reliable ZCD timing. Practical start-up logic also includes initial rotor-polarity detection and backspin handling to avoid reverse torque impulses under residual motion or load disturbances.

2.3.1.6 Advantages and Limitations

Advantages Algorithmic simplicity, modest compute requirements, and robust operation with Hall sensors make six-step control attractive for cost-sensitive drives. Minimal angle estimation is required compared with sinusoidal FOC.

Typical Limitations Elevated torque ripple (due to mismatch between rectangular current and trapezoidal EMF, particularly during commutation), increased acoustic noise, reduced efficiency at light load or very high speed, and sensitivity to DC-bus ripple and dead-time distortion at high modulation. While 180° conduction can reduce current ripple for near-sinusoidal EMF, 120° conduction remains standard in BLDC drives targeting minimal switching and simple commutation. [7, p. 16, p. 57] [18, p. 533]

2.3.1.7 Key Equations

Per-phase voltage (including back-EMF e_a):

$$u_a = R\,i_a + (L-M)\,\frac{di_a}{dt} + e_a(\theta,\omega). \tag{2.1}$$

Equation (2.1) follows from the symmetric three-phase model $u_a = R\,i_a + L\dot{i}_a + M(\dot{i}_b + \dot{i}_c) + e_a$ together with the constraint $i_a + i_b + i_c = 0$, which implies $\dot{i}_b + \dot{i}_c = -\dot{i}_a$ and yields $u_a = R\,i_a + (L-M)\,\dot{i}_a + e_a$.

[7, p. 33, p. 48] Under ideal 120° conduction, the electromagnetic torque can be approximated by

$$T_e \approx K_t\,I_{\text{eff}}. \tag{2.2}$$

Here I_{eff} denotes the line-current magnitude in the two conducting phases during the 120° interval (or its step-average value when commutation overlap is present). [14, p. 14]

2.3.2 Sinusoidal Control and Field-Oriented Control (FOC)

Field-Oriented Control (FOC), also referred to as vector control, aims to regulate the electromagnetic torque of an AC machine instantaneously by emulating the decoupled control behavior of a separately excited DC motor [19, p. 230]. In a DC motor, the field flux and armature current are maintained orthogonal through mechanical commutation. In contrast, FOC achieves this decoupling electronically by controlling both the magnitude and phase of the stator current vector with respect to the rotor flux [20, p. 4].

A schematic overview of the implemented field-oriented control structure is shown in Fig. 2.10. The diagram highlights the hierarchical control architecture, including the inner dq-axis current controllers, the inverse coordinate transformations, and the feedback of measured phase currents and rotor position.

2.3.2.1 Clarke and Park Transforms and *dq* Decoupling

The foundation of FOC lies in transforming three-phase, time-varying quantities into time-invariant components suitable for linear control [20, p. 5].

- **Clarke Transform ($abc \to \alpha\beta$):** Projects the three-phase stator currents into a two-axis stationary orthogonal reference frame.

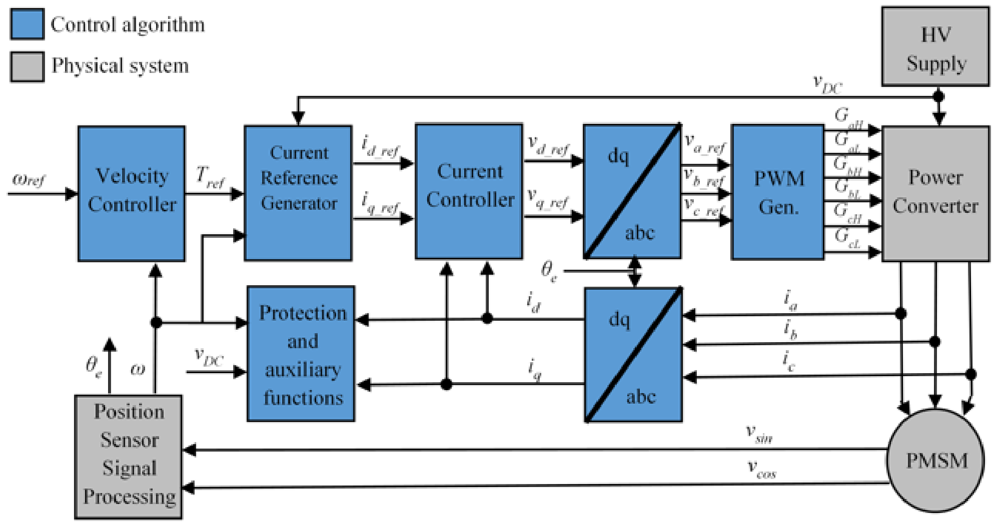


Figure 2.10: Block diagram of a field-oriented control (FOC) scheme for a PMSM, illustrating the current control loops, coordinate transformations, inverter, and feedback signals [21].

- **Park Transform** ($\alpha\beta \rightarrow dq$)**:** Rotates the stationary reference frame by the electrical rotor angle θ so that it aligns with the rotor flux.

In the rotating dq frame, the stator current is decomposed into [19, p. 174]:

- i_d (direct-axis current), which controls the magnetic flux,
- i_q (quadrature-axis current), which is orthogonal to the flux and directly controls the electromagnetic torque [5, p. 333].

To treat i_d and i_q as independent DC control variables, the controller must compensate the inherent cross-coupling terms caused by the rotating reference frame, commonly referred to as speed-voltage terms [20, p. 9]:

$$\omega_e L_q i_q \quad \text{and} \quad \omega_e L_d i_d.$$

The geometric interpretation of the Clarke and Park transformations is illustrated in Fig. 2.11. By rotating the stationary $\alpha\beta$ frame into the synchronous dq frame using the electrical angle θ, the stator current components become DC quantities under steady-state conditions.

2.3.2.2 Application to BLDC Motors with Trapezoidal Back-EMF

Although BLDC motors are designed with trapezoidal back-EMF characteristics and are traditionally driven using six-step commutation, they benefit from sinusoidal current excitation or full FOC operation. Conventional trapezoidal control introduces torque ripple during commutation intervals. By shaping the phase currents to be near-sinusoidal, torque pulsations and acoustic noise are significantly reduced. This improvement comes at the cost of a slight reduction in peak torque compared to ideal rectangular current excitation.

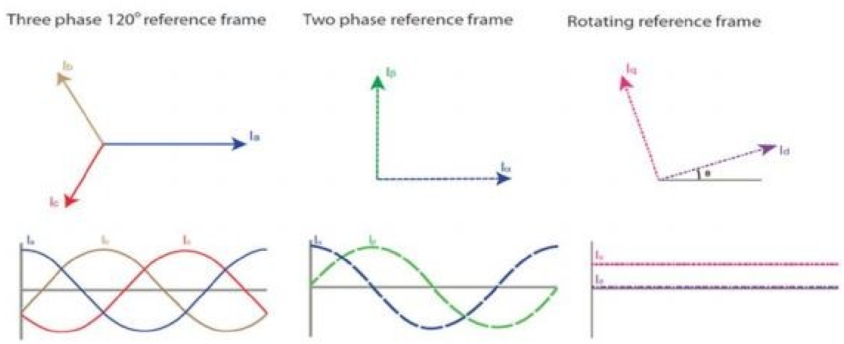


Figure 2.11: Relationship between the stationary $\alpha\beta$ reference frame and the rotating dq reference frame used in field-oriented control. The rotation by the electrical angle θ aligns the d-axis with the rotor flux.[22]

2.3.2.3 Current Control Structure

Current Measurement Phase currents are typically measured using one, two, or three shunt resistors or in-line current sensors [20, p. 12]. While three shunts provide full observability, two-shunt configurations are common in practice, as the third phase current can be reconstructed using the constraint [19, p. 333–337]:

$$i_a + i_b + i_c = 0.$$

Shunt placement affects the achievable signal-to-noise ratio and imposes constraints on the minimum PWM duty cycle due to current settling requirements before sampling.

Rotor Angle Generation Accurate rotor position information is required for the Park transformation. This information is commonly obtained using encoders or Hall-effect sensors. In sensorless drives, rotor position is estimated using observers such as sliding-mode observers, Kalman filters, or phase-locked loops (PLL) based on measured voltages and currents [5, p. 335].

PI Current Control and Bandwidth The d- and q-axis currents are regulated using PI controllers augmented with feedforward and decoupling terms to compensate back-EMF and cross-coupling effects. Anti-windup mechanisms are essential to prevent integrator saturation when the inverter voltage reaches its limits. The bandwidth of the current control loop is typically selected between one-tenth and one-twentieth of the PWM switching frequency to ensure adequate phase margin and robustness [19, p. 166 , p. 174].

Speed Control Loop An outer speed control loop generates the torque-producing current reference i_q^* [5, p. 334]. Its bandwidth must be significantly lower than that of the current loop to ensure hierarchical stability. Slew-rate limiting is often applied to reduce mechanical stress and torque transients [19, p. 168].

2.3.2.4 Space-Vector Pulse-Width Modulation (SVPWM)

Space-Vector PWM (SVPWM) is commonly employed to synthesize the commanded voltage vectors [5, p. 101, p. 238].

- **Principle:** The reference voltage vector is approximated by appropriately distributing the dwell times of two adjacent active vectors and the zero vectors within a switching period.
- **DC Bus Utilization:** SVPWM achieves approximately 15.5% higher DC bus utilization compared to sinusoidal PWM, with a maximum linear modulation index of $m_{\max} = 2/\sqrt{3} \approx 1.155$.
- **Implementation Aspects:** ADC sampling is synchronized with the PWM carrier, typically at the peak or valley, to minimize switching noise. Dead-time compensation is required to correct voltage errors introduced by non-overlapping gate signals in the inverter [19, p. 324].

2.3.2.5 Field-Weakening Operation

Operation beyond the rated base speed requires field-weakening, which is achieved by injecting a negative i_d current.

Voltage Limit Ellipse As speed increases, the stator voltage demand grows until it reaches the voltage limit imposed by the DC bus. By reducing the effective air-gap flux through negative i_d, the back-EMF is lowered, allowing operation in the constant-power region beyond base speed [19, p. 143, p. 263].

Loss Considerations Field-weakening increases the magnitude of the stator current, leading to higher copper losses and reduced efficiency at high speeds [19, p. 44].

2.3.2.6 Performance Comparison with Trapezoidal Control

Compared to trapezoidal control, FOC provides significantly lower torque ripple and acoustic noise due to the smooth rotation of the magnetic field. It enables superior efficiency at partial load conditions, improved low-speed performance, and precise torque control. These advantages come at the cost of increased computational complexity due to real-time coordinate transformations, control loops, and PWM synthesis.

2.3.2.7 Key Equations

DQ-Axis Voltage Model

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q, \tag{2.3}$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + \omega_e (L_d i_d + \psi_f). \tag{2.4}$$

Electromagnetic Torque

$$T_e = \frac{3}{2} p \left(\psi_f i_q + (L_d - L_q) i_d i_q \right). \tag{2.5}$$

For non-salient machines such as surface-mounted PMSMs and typical BLDC motors, $L_d \approx L_q$, making the reluctance torque term negligible [5, p. 357–358].

Units Consistency When expressed in consistent SI units, the torque constant K_t and the back-EMF constant K_e are numerically identical:

$$K_t = K_e.$$

2.3.3 Advanced Control Methods

2.3.3.1 Direct Torque Control (DTC)

Direct Torque Control (DTC) is an advanced control strategy for AC motor drives, distinguished by its conceptual simplicity, ease of digital implementation, and robust dynamic behavior [5, p. 344].

Operating Principle DTC regulates the electromagnetic torque and the stator flux magnitude directly using hysteresis (tolerance-band) controllers. Based on the instantaneous errors of torque and flux, together with the angular sector of the stator flux vector, a switching logic selects the appropriate inverter voltage vector from a predefined switching table. This selection is performed at every sampling instant to drive the stator flux along a desired trajectory, thereby producing the required torque [5, p. 345] [18, p. 606].

Advantages The principal advantage of DTC is its very fast torque response and excellent dynamic performance. The control scheme is inherently robust and relatively insensitive to motor parameter variations, as it typically requires only the stator resistance R_s for flux estimation [18, p. 391, p 613].

Limitations The use of hysteresis controllers results in relatively high torque and flux ripples. Furthermore, DTC operates with a variable switching frequency, which complicates the design of input and output filters and increases acoustic noise. The rapid switching of voltage vectors can also lead to increased electromagnetic interference (EMI) [15, p. 882] [5, p. 557].

Application Scope DTC is particularly well suited for applications requiring rapid torque response, such as traction drives and high-performance industrial systems, where robustness and simplicity are prioritized over the low-ripple characteristics achievable with Field-Oriented Control (FOC).

2.3.3.2 Model Predictive Control (MPC)

Model Predictive Control (MPC) has emerged as a powerful alternative for electric drive systems, leveraging a mathematical model of the motor and inverter to predict future system behavior [23, p. 4, p. 23].

Control Concept In motor drive applications, MPC is commonly implemented as a finite control set approach. All admissible switching states of the inverter are evaluated using a predictive model, and a cost function is defined that penalizes deviations in controlled variables (such as current or torque) and may include switching or constraint-related terms. The switching state that minimizes the cost function is applied during the subsequent sampling interval.

Advantages MPC naturally handles multivariable systems and allows hard constraints, such as current and voltage limits, to be incorporated directly into the control law. It offers fast dynamic response and a high degree of flexibility, including the possibility of online adaptation and optimization [18, p. 723].

Limitations The primary drawback of MPC is its high computational burden, as the optimization problem must be solved at every sampling instant. The method is also sensitive to parameter mismatches, since prediction accuracy depends strongly on the fidelity of the motor model. Additionally, tuning the weighting factors of the cost function can be nontrivial and application-dependent [23, p. 25 – 27] [18, p. 691].

Real-Time Feasibility The real-time feasibility of MPC is determined by the available computational resources and the chosen sampling period. High-performance digital signal processors (DSPs) or microcontrollers are required to complete the optimization within the tight timing constraints imposed by fast sampling rates and measurement latency [15, p. 605, p. 632, p. 892].

2.3.3.3 Implementation Trade-Offs

The implementation of advanced control strategies such as DTC and MPC involves several practical trade-offs.

Computational Complexity Advanced methods, including MPC or enhanced DTC variants (e.g., predictive or neuro-fuzzy approaches), demand a higher computational budget and increased software complexity. This can negatively affect maintainability, verification, and certification of the drive system [23, p. 22–23].

Sampling and Timing Constraints The selection of the sampling period represents a compromise between control performance and processor load. Shorter sampling periods improve tracking accuracy and dynamic response but significantly increase computational demand. Accurate synchronization of ADC sampling with the PWM carrier is essential to minimize switching noise, particularly in high-frequency inverter systems.

Hardware Cost While these advanced control strategies can be implemented on high-performance 32-bit fixed-point DSPs, the requirement for increased processing capability generally leads to higher hardware cost and system complexity [15, p. 625].

2.3.4 Comparative Analysis and Control Strategy Selection

This subsection compares trapezoidal (six-step), sinusoidal current control, field-oriented control (FOC), direct torque control (DTC), and model predictive control (MPC) against the requirements of the target platform and application. In my case, constraints include an ARM-class MCU (moderate MIPS/limited FPU), smooth low-speed behavior for precise speed/position tasks, robust operation across parameter drift and DC-link variation, and tight EMC/EMI budgets. The analysis below motivates the choice of FOC as the preferred compromise of performance, robustness, and implementation effort.

Evaluation criteria

- **Sensors required:** Halls/encoder (sensored) vs observer-based (sensorless).
- **Algorithmic complexity:** arithmetic intensity, memory footprint, control-law structure.
- **Dynamic bandwidth:** current/torque loop bandwidth; speed loop bandwidth.
- **Steady-state efficiency:** current waveform quality, switching and copper losses.
- **Torque ripple & acoustic noise:** commutation/EMF mismatch, PWM strategy, mechanical coupling.
- **Low-speed smoothness:** start-up, sub-rated speeds, standstill torque.
- **Robustness to parameter drift:** R_s, L, ψ_f, temperature, DC-link ripple, dead-time.
- **Implementation cost:** code size, calibration, commissioning/tuning effort.

Suitability for discrete-time digital control

- **Sampling & PWM sync:** mid-carrier sampling; deterministic ISR latency; minimum on-time constraints.
- **Dead-time compensation:** inverse dead-time mapping or table-based calibration; current-polarity aware.

- **ADC topology:** 1-/2-/3-shunt or in-line sensing; current reconstruction windows at high modulation.
- **Loop bandwidth targets:** $f_{\text{BW,i}} \approx f_{\text{sw}}/10 \ldots f_{\text{sw}}/5$; $f_{\text{BW},\omega} \ll f_{\text{BW,i}}$.

Qualitative performance comparison **Trapezoidal (six-step):** Lowest computational burden and BOM cost. Performs acceptably at mid–high speeds with Halls, but exhibits the highest torque ripple and acoustic noise, poorer efficiency at light load, quantized position information (if sensored), limited low-speed smoothness, and sensitivity to DC-link ripple and dead-time.
Sinusoidal current control (no full FOC): Intermediate complexity. By shaping phase currents toward sinusoidal, it reduces ripple and noise relative to six-step while keeping the controller simpler than full dq decoupling. However, cross-coupling remains, transient torque linearity is limited, and field-weakening authority is modest.
Field-Oriented Control (FOC): Best overall trade-off. Decouples torque/flux in dq, enabling low ripple, high efficiency (especially at partial load), smooth low-speed operation (with sensored or sensorless angle), accurate torque linearity, and controlled field-weakening. Requires angle and current sensing and more compute than the two previous methods, but remains well within an ARM-class MCU budget when implemented with fixed-point friendly PI loops and SVPWM.
Direct Torque Control (DTC): Very fast torque dynamics and simple structure (no modulator), but variable switching frequency, elevated torque ripple, and higher EMI/acoustic signatures make it less suitable where smoothness and noise are primary objectives.
Model Predictive Control (MPC): High flexibility (constraints, multi-objective cost functions) and excellent transient performance, but comparatively heavy computation and tuning effort. On an MCU-class ARM, maintaining short sample times with robust prediction and anti-windup is challenging; typically reserved for platforms with more headroom or stringent constraints that justify the overhead.

Implementation complexity & maintainability Trapezoidal control minimizes code size and tuning but shifts effort to acoustic/EMC mitigation and application-specific commutation timing. Sinusoidal control adds modest complexity (current regulators and angle management) while improving noise and efficiency. FOC adds dq transforms, decoupling/feedforward, SVPWM, and field-weakening logic, but benefits from established tuning procedures, well-understood failure modes, and straightforward extension to sensorless observers. DTC and MPC require more intricate commissioning (hysteresis/torque estimators for DTC; cost-function design, constraints, and timing guarantees for MPC). From a lifecycle perspective—calibration reuse, portability across similar motors, diagnostics, and safety interlocks—FOC offers the most balanced maintainability on embedded MCUs.

Selection and justification (this work) Given the application targets—*low torque ripple and acoustic noise*, *precise speed/position control at low speed*, *robustness to parameter drift and*

DC-link variation, and *ARM-class compute and I/O constraints*—FOC is selected as the control strategy. It delivers:

- **Performance:** linear torque control, reduced ripple/noise, efficient operation across load/speed, and controlled field-weakening.
- **Robust sensing:** compatibility with Halls/encoders and with sensorless observers (forward reference: 2.4).
- **Digital fit:** deterministic fixed-frequency operation with mid-carrier sampling, dead-time compensation, and 1/2/3-shunt current reconstruction.
- **Engineering economy:** mature tuning workflow (current-then-speed loops), reusable calibration, and scalable code on the target MCU.

For completeness, a sinusoidal current controller without full dq decoupling is acknowledged as a lower-complexity fallback where torque linearity and field-weakening are less critical. Trapezoidal six-step is not pursued due to ripple/noise and low-speed smoothness limitations. DTC and MPC are excluded on grounds of torque-ripple/EMI (DTC) and computational/tuning overhead (MPC) relative to the platform and application goals.

2.4 Rotor Position Detection

Accurate knowledge of the rotor position is a fundamental requirement for modern electric motor drive systems, particularly for vector-controlled machines such as Permanent Magnet Synchronous Motors (PMSMs). Rotor position information is required to correctly align the stator current reference frame with the rotor magnetic field, enabling effective decoupling of torque and flux components. Depending on application requirements, this information can be obtained either through physical sensors mounted on the motor or by estimating the position from measured electrical quantities. This section reviews the most common sensored and sensorless rotor position detection techniques, discusses their respective advantages and limitations, and provides the basis for the method selected in this thesis.

2.4.1 Sensored Methods

Accurate rotor position information is essential for providing correct commutation and reference frame alignment in electric motor drive systems [7, p. 27]. In sensored control architectures, this information is obtained through physical sensing devices that directly measure the relative position of the rotor with respect to the stator and convert it into electrical signals suitable for control purposes. Such methods are widely used in Brushless DC (BLDC) and Permanent Magnet Synchronous Motor (PMSM) drives, particularly where reliable position feedback is required across the entire speed range, including standstill [7, p. 230].

2.4.1.1 Hall Sensors

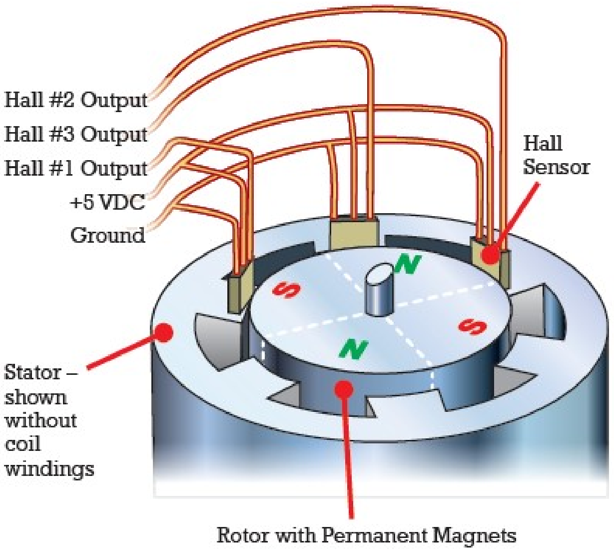


Figure 2.12: Typical placement of Hall-effect sensors in a permanent magnet motor. The sensors detect the magnetic field produced by the rotor magnets and generate discrete logic signals corresponding to fixed electrical position sectors [24].

Operating Principle As seen in Fig. 2.12, discrete Hall-effect sensors are typically mounted on the motor stator at fixed angular intervals [7, p. 215]. These magnetic sensors detect the magnetic field produced by the rotor permanent magnets and generate digital output signals as the rotor poles pass by. In three-phase machines, Hall sensors are commonly arranged with 60° or 120° electrical spacing, producing a sequence of logic signals that represent distinct rotor position sectors [7, p. 32].

Information Quality Hall sensors provide rotor position information in coarse, discrete electrical sectors. This resolution is well suited for block (six-step) commutation schemes, where current is sequentially applied to pairs of stator windings every 60° of electrical rotation. However, because only six distinct position states are available per electrical cycle, the angular resolution is insufficient for high-performance Field-Oriented Control (FOC) or fully sinusoidal commutation unless additional interpolation or estimation techniques are employed [7, p. 215] [8, p. 29, p. 50].

Advantages

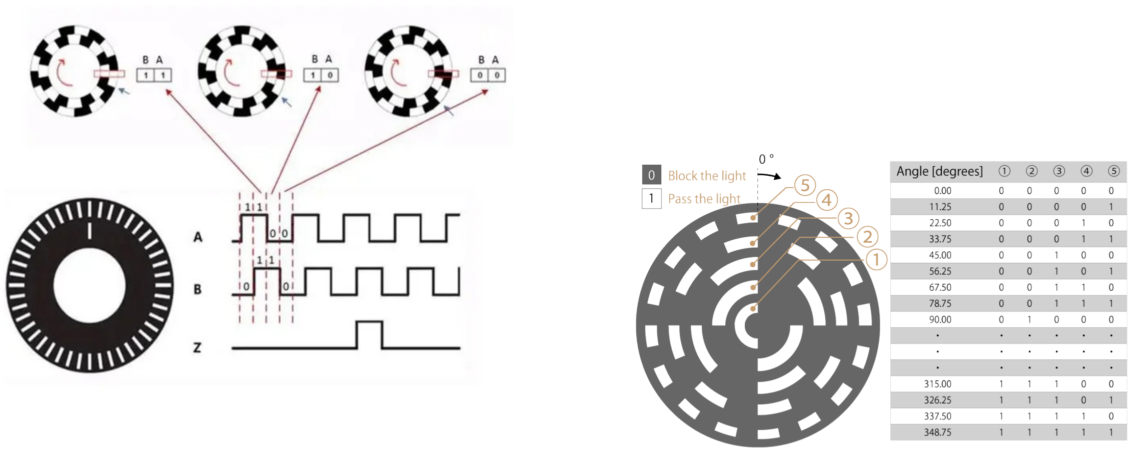
- **Simplicity:** Hall sensors are inexpensive, compact, and require minimal signal conditioning or peripheral circuitry [7, p. 237].
- **Robustness:** Due to their solid-state nature, they are widely used in industrial applications and tolerate harsh operating conditions [7, p. 216].

- **Low-Speed and Standstill Operation:** Hall sensors provide reliable position information at very low speeds and at standstill, enabling deterministic startup behavior [7, p. 167].

Limitations

- **Low Angular Resolution:** The discrete nature of the signals limits position accuracy and control smoothness [8, p. 72].
- **Torque Ripple:** When used directly for block commutation, the stator and rotor magnetic fields are not continuously maintained in an orthogonal relationship, leading to periodic torque ripple at six times the electrical frequency.
- **Limited Suitability for Precision Control:** The coarse position information makes Hall sensors unsuitable for applications requiring smooth torque production or precise position control if the signals are not refined properly [8, p. 49].

2.4.1.2 Encoders



Angle [degrees]	①	②	③	④	⑤
0.00	0	0	0	0	0
11.25	0	0	0	0	1
22.50	0	0	0	1	0
33.75	0	0	0	1	1
45.00	0	0	1	0	0
56.25	0	0	1	0	1
67.50	0	0	1	1	0
78.75	0	0	1	1	1
90.00	0	1	0	0	0
·	·	·	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·
315.00	1	1	1	0	0
326.25	1	1	1	0	1
337.50	1	1	1	1	0
348.75	1	1	1	1	1

Figure 2.13: Comparison between incremental (left - Incremental encoder with quadrature signals (A, B) and index pulse (Z), providing relative position and speed information [25]) and absolute (right - Absolute encoder using multiple tracks to encode a unique digital word for each shaft position [26]) encoders used for rotor position sensing in electric drive systems.

Types Encoders can be broadly classified into incremental and absolute types. Incremental encoders measure changes in rotor position and are commonly used for speed estimation, while absolute encoders provide a unique digital code for each shaft position, allowing direct determination of the absolute rotor angle without requiring a homing procedure [8, p. 80–81].

Operating Principle Encoders provide a direct measurement of the mechanical rotor position. Optical encoders, which are the most common type in high-performance drives, use a light source and a coded disc to generate high-frequency pulse trains as the shaft rotates. This enables very high angular resolution, often in the range of 8 to 12 bits per mechanical revolution or higher [1, p. 51].

Advantages

- **High Accuracy:** Encoders deliver the resolution required for precise stator flux orientation and accurate torque control [8, p. 50].
- **Compatibility with High-Performance Control:** They are well suited for FOC-based drive systems, where accurate knowledge of rotor position is critical for decoupled torque and flux control [8, p. 30].
- **Reliable Zero-Speed Operation:** Encoders maintain full accuracy at zero speed and standstill, ensuring stable control under static or dynamic load conditions.

Disadvantages

- **Higher Cost:** Encoders are significantly more expensive than Hall sensors and can represent a substantial portion of the overall system cost [7, p. 20].
- **Mechanical Integration Complexity:** Shaft mounting requires additional mechanical space and precise alignment, increasing system volume and assembly effort.
- **Increased Wiring and Interface Requirements:** Encoders require additional signal lines and dedicated interface hardware on the control unit, contributing to system complexity [7, p. 183].

Practical Limitations: Cost, Wiring, and Reliability The use of physical position sensors introduces a fundamental engineering trade-off between control performance and system-level constraints.

Cost Impact The most immediate drawback of sensored solutions is the added hardware cost, particularly in the case of high-resolution absolute encoders. Additional expenses arise from cabling, connectors, and the labor required for precise mechanical installation and calibration. Inaccurate sensor mounting can lead to phase misalignment, degraded torque production, and increased ripple [1].

Reliability Concerns Sensored systems are more vulnerable in harsh operating environments. Elevated temperatures, moisture, dust, and mechanical vibrations can degrade sensor performance or lead to connector and cable failures. These failure mechanisms are largely absent in sensorless control architectures.

System-Level Complexity The inclusion of sensors increases the number of potential failure points within the drive system. From an electronic perspective, additional sensor inputs raise the demand for microcontroller I/O resources and exacerbate electromagnetic compatibility (EMC) challenges. Sensor cables may act as antennas for high-frequency switching noise, often necessitating shielded wiring or isolation techniques, which further increase cost and design complexity.

2.4.2 Sensorless Methods

Sensorless control techniques eliminate the need for mechanical rotor position sensors by estimating the rotor position and speed directly from electrical quantities measured at the motor terminals, such as stator voltages and currents. By avoiding shaft-mounted sensors, these methods reduce hardware complexity, system cost, and susceptibility to mechanical failures. As a result, sensorless approaches are increasingly favored in industrial and automotive drive systems, particularly in applications exposed to harsh environmental conditions [27, p. 1].

2.4.2.1 Back-EMF-Based Approaches

Core Idea The fundamental principle of back-electromotive force (back-EMF) based methods is that rotor position information can be inferred from the voltage induced in the stator windings by the rotating permanent magnet field. In three-phase machines, this is commonly achieved through zero-crossing detection techniques, where the instant at which the back-EMF of a non-energized phase crosses zero corresponds to a known rotor position [23, p. 2] [7, p. 8].

Typical Operating Region Back-EMF-based methods are most effective in the medium to high speed operating range, where the induced voltage magnitude is sufficiently large to be reliably detected. Due to their simplicity and maturity, these techniques are widely used in standard sensorless drive applications [7, p. 168].

Advantages By eliminating mechanical sensors, back-EMF-based approaches enable more compact and lightweight drive systems. The absence of sensor cabling improves noise immunity and overall robustness, making these methods suitable for high-power automotive and industrial applications where reliability and cost-effectiveness are critical.

Key Limitations The principal limitation of back-EMF-based techniques is that the induced voltage amplitude is proportional to rotor speed. At low speeds or standstill, the back-EMF becomes too small to be distinguished from noise, leading to poor observability of the rotor position. Consequently, these methods are not well suited for applications requiring accurate control at very low speeds or during startup [27, p. 5, p. 7] [23, p. 2].

2.4.2.2 Observer-Based Methods

Observer-based sensorless methods estimate rotor position and speed using mathematical models of the motor combined with measured terminal voltages and currents. These approaches operate in a closed-loop manner, continuously correcting the estimated states based on the error between measured currents and model-predicted currents [27, p. 8, p. 15].

Examples Common high-level implementations include state observers and model-based estimators such as Extended Kalman Filters and Sliding Mode Observers. These techniques solve the motor state equations in real time to estimate the instantaneous angular position and speed of the magnetic flux vector.

Advantages Observer-based methods generally offer superior dynamic performance compared to open-loop back-EMF sensing. They are particularly well suited for Field-Oriented Control (FOC), as they provide continuous, high-resolution estimates of flux magnitude and spatial orientation, which are essential for accurate torque and flux decoupling [27, p. 38] [20, p. 4].

Challenges The accuracy of observer-based methods is strongly dependent on the precision of the motor parameters used in the model. In particular, variations in stator resistance due to temperature changes can significantly degrade estimation performance. Additionally, these methods impose a higher computational burden, often requiring high-performance microcontrollers or digital signal processors to execute the necessary real-time calculations [27, p. 8] [23, p. 29].

2.4.3 Low-Speed and Standstill Challenges

The performance of sensorless control strategies is fundamentally constrained at low rotational speeds by the physical characteristics of electric machines. Since most sensorless techniques rely on electromagnetic phenomena that are directly linked to rotor motion, their effectiveness degrades significantly when the rotor speed approaches zero or during standstill operation.

Signal-to-Noise Limitations The primary limitation at low speeds arises from the fact that the induced back-electromotive force (back-EMF) is directly proportional to the rotor speed. As the speed decreases, the amplitude of the back-EMF correspondingly diminishes and approaches zero, making reliable detection increasingly difficult.

In this operating region, the useful signal becomes comparable in magnitude to various sources of measurement noise, including sensor offsets, signal drift, quantization effects, and nonlinear distortions introduced by the power inverter. As a result, the controller is unable to clearly distinguish the back-EMF component from the surrounding electrical noise. This loss of signal observability leads to inaccurate rotor angle estimation, which directly affects commutation timing. Consequently, the drive may experience increased torque ripple, degraded dynamic response, and, in severe cases, closed-loop instability [27, p. 1, p. 9, p. 20] [23, p. 1, p. 18].

Startup Issues At standstill, the back-EMF is inherently zero, creating an observability gap for sensorless techniques that depend on rotor-induced voltages. Under these conditions, the initial rotor position is unknown, which presents a fundamental challenge during motor startup. If torque is applied without accurate position information, there is a high risk of generating torque in an unintended direction, potentially resulting in oscillatory behavior, delayed acceleration, or failure to initiate rotation [23, p. 28] [27, p. 75].

To overcome this limitation, sensorless drive systems typically employ dedicated startup procedures to establish an initial reference for the rotor position. Common mitigation strategies include:

- **Alignment Procedures:** Selected stator windings are energized to create a stationary magnetic field that forces the rotor into a predefined and known position prior to normal operation.
- **Open-Loop Startup:** The motor is initially driven using an open-loop voltage or frequency ramp, emulating synchronous operation. This forces the rotor to accelerate until sufficient speed is reached for the sensorless estimation algorithm to achieve reliable observability and transition into closed-loop control.

2.4.4 Selection Rationale for Position Detection

The selection of an appropriate rotor position detection strategy is a fundamental design decision in Field-Oriented Control (FOC) systems, as accurate and continuous knowledge of the rotor electrical angle is required to maintain proper alignment between the stator current reference frame and the rotor magnetic field. Several alternatives exist, ranging from fully sensored solutions to purely sensorless estimation techniques, each involving distinct trade-offs in terms of performance, cost, robustness, and implementation complexity.

Fully sensored approaches, such as high-resolution encoders, provide precise rotor position information across the entire speed range, including standstill. However, their use introduces additional hardware cost, mechanical integration challenges, and reduced robustness in harsh environments. These drawbacks motivate the widespread adoption of sensorless techniques in industrial drive systems, particularly in applications where cost efficiency and reliability are prioritized over absolute positioning accuracy.

Sensorless rotor position estimation methods exploit the intrinsic electromagnetic behavior of the machine and rely exclusively on electrical measurements that are already available for control purposes. Among these methods, observer-based techniques offer high dynamic performance and accurate position tracking by incorporating feedback mechanisms that correct estimation errors. While effective, such approaches typically require accurate machine models, careful parameter tuning, and increased computational resources, which may be undesirable or impractical in embedded control platforms with constrained processing capabilities.

Simpler sensorless strategies based on voltage or back-electromotive force (back-EMF) models provide a viable alternative in operating regions where signal observability is sufficient. These methods estimate the rotor electrical angle from reconstructed stator flux quantities derived from measured voltages and currents. Although such approaches lack the correction mechanisms associated with full state observers and are therefore sensitive to low-speed operation and parameter uncertainties, they offer reduced implementation complexity and are well suited for medium to high speed ranges.

A known limitation common to all back-EMF-based sensorless techniques is their reduced reliability at very low speeds and during standstill, where the induced voltage becomes too small to be accurately detected. For this reason, practical drive systems often combine sensorless estimation with auxiliary position sensing methods to ensure reliable startup and low-speed operation.

Based on these considerations, a hybrid rotor position detection strategy is adopted in this work. Discrete position sensing from a Hall sensor is employed to guarantee robust behavior at low speeds and during startup, while sensorless voltage-model-based estimation is utilized at higher speeds, where electromagnetic observability is sufficient. This hybrid approach represents a balanced compromise between control performance, system robustness, and implementation complexity, and reflects common practices in modern industrial PMSM drive systems.

2.5 Mathematical Modeling

2.5.1 Phase-Domain (abc) Model of the Three-Phase PMSM

The phase-domain model, commonly referred to as the *abc* model or machine-variable model, provides a fundamental description of the three-phase Permanent Magnet Synchronous Machine (PMSM) using physical stator variables[28, p. 121, p. 655]. This formulation captures the electromagnetic behavior of the machine directly in terms of phase voltages, currents, and flux linkages. Although not suitable for control design, it serves as the starting point for the derivation of more convenient control-oriented models[28, p. 325][1, p. 225].

Electrical Equations in Phase Variables

For a three-phase wye-connected PMSM, the stator voltage equations are expressed as[28, p. 121 – p. 125][1, p. 227, p. 463]:

$$v_a = R_s i_a + \frac{d\lambda_a}{dt} \tag{2.6}$$

$$v_b = R_s i_b + \frac{d\lambda_b}{dt} \tag{2.7}$$

$$v_c = R_s i_c + \frac{d\lambda_c}{dt} \tag{2.8}$$

where:

- R_s is the stator resistance per phase,

- v_a, v_b, v_c are the instantaneous phase voltages,
- i_a, i_b, i_c are the instantaneous phase currents,
- $\lambda_a, \lambda_b, \lambda_c$ are the corresponding phase flux linkages.

Each phase voltage is therefore composed of a resistive drop and a term associated with the time variation of the magnetic flux linkage.

Flux Linkage Representation

The total flux linkage of each stator phase results from two contributions, the magnetic field produced by the stator currents (armature reaction) and the magnetic field generated by the permanent magnets mounted on the rotor. In vector form, this relationship can be expressed as[1, p. 228][6, p. 245]:

$$\lambda = \mathbf{L}(\theta_r)\mathbf{i} + \lambda_m(\theta_r) \tag{2.9}$$

where:

- $\mathbf{L}(\theta_r)$ is the stator inductance matrix,
- $\mathbf{i} = [i_a\ i_b\ i_c]^T$,
- $\lambda_m(\theta_r)$ is the flux linkage vector produced by the permanent magnets,
- θ_r is the rotor electrical angle.

The inductance matrix $\mathbf{L}(\theta_r)$ is generally position-dependent[28, p. 53]. In PMSMs, particularly in salient-pole (IPMSM) designs, magnetic reluctance varies with rotor position. As a result, both self- and mutual-inductance terms contain components that vary with $\cos(2\theta_r)$ and $\sin(2\theta_r)$, making the system time-varying.

Similarly, the permanent magnet flux linkage $\lambda_m(\theta_r)$ depends on rotor position. As the rotor rotates, the flux linking each stator phase varies sinusoidally with θ_r, typically following a sinusoidal distribution for machines with sinusoidal back-EMF.

Nonlinearity and Coupling

Due to the rotor-position dependence of the inductance matrix and magnet flux, the *abc* model is inherently nonlinear and strongly coupled. The voltage equation of each phase depends not only on its own current but also on the currents of the other phases and on the instantaneous rotor position and speed. This coupling significantly complicates direct controller design in the phase domain[28, p. 81–p. 83].

Mechanical Equation

The mechanical dynamics of the PMSM are governed by the rotational equation of motion[28, p. 385][1, p. 459]:

$$J\frac{d\omega_m}{dt} = T_e - T_L - B\omega_m \tag{2.10}$$

where:

- J is the total moment of inertia of the rotor and load,
- ω_m is the mechanical angular velocity,
- T_e is the electromagnetic torque,
- T_L is the load torque,
- B is the viscous friction coefficient.

This equation couples the electrical subsystem to the mechanical dynamics, completing the full electromechanical model of the PMSM.

Discussion

Although the *abc* model accurately represents the physical behavior of the machine and is suitable for detailed electromagnetic simulations, it is not convenient for control design. The strong magnetic coupling between phases and the explicit rotor-angle dependence of the system coefficients make linear controller synthesis difficult.

For this reason, modern drive systems employ a transformation to a rotating reference frame (typically the dq frame), which eliminates position-dependent terms and produces a model better suited for high-performance field-oriented control.

- Per-phase voltage equations including back-EMF and coupling
- Non-idealities: mutual inductance, saturation, dead-time, device drops

2.5.2 State-Space Representation of the PMSM

The state-space representation reformulates the governing differential equations of the PMSM into a unified first-order vector form. This formulation is fundamental for modern control theory, as it provides a structured description of the system dynamics in terms of internal states and external inputs. It also enables systematic controller design and numerical simulation[28, p. 395].

Dynamic System Reformulation

The electrical and mechanical equations derived in the phase-domain model consist of coupled differential equations. To express the PMSM in state-space form, these equations are rewritten as a set of first-order differential equations. This representation facilitates numerical integration and establishes a framework suitable for nonlinear control analysis[1, p. 460].

Definition of the State Vector

The state vector contains the minimum set of variables required to uniquely determine the future evolution of the system, given its present condition and applied inputs. For a three-phase PMSM described in the phase domain, a suitable state vector is defined as:

$$\mathbf{x} = \begin{bmatrix} i_a & i_b & i_c & \omega_m & \theta_m \end{bmatrix}^T \tag{2.11}$$

where:

- i_a, i_b, i_c are the stator phase currents,
- ω_m is the mechanical angular velocity,
- θ_m is the mechanical rotor position.

The electrical states describe the electromagnetic behavior of the stator windings, while the mechanical states capture the rotational dynamics of the rotor-load system.

Nonlinear State-Space Form

The complete electromechanical dynamics of the PMSM can be expressed in the general nonlinear state-space form[28, p. 396][15, p. 357]:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{2.12}$$

where:

- $\mathbf{u} = [v_a \ v_b \ v_c]^T$ represents the stator phase voltages supplied by the power electronic converter,
- $\mathbf{f}(\cdot)$ is a nonlinear vector function describing the machine dynamics.

The output vector may include measurable or controlled variables such as stator currents, rotor speed, rotor position, and electromagnetic torque.

System Characteristics

The state-space formulation reveals several fundamental properties of the PMSM:

Nonlinearity The system is inherently nonlinear. This nonlinearity originates primarily from the rotor-position dependence of the inductance matrix in the phase-domain model and from the back-electromotive force (back-EMF), which depends on the product of rotor speed and magnet flux. Furthermore, the electromagnetic torque is a nonlinear function of currents and flux linkages[1, p. 262].

Electromechanical Coupling The electrical and mechanical subsystems are strongly coupled. Stator currents generate electromagnetic torque, which drives mechanical acceleration. Conversely, rotor speed influences the electrical subsystem through the induced back-EMF, directly affecting the stator voltage equations[15, p. 339][5, p. 325].

Control Perspective From a control standpoint, the primary objective in high-performance drive systems is torque regulation. Since electromagnetic torque is directly related to the stator currents, torque control is typically achieved through high-bandwidth current control. This perspective motivates the transformation of the model into a reference frame where torque-producing and flux-producing components can be independently regulated.

Discussion

Although the state-space representation compactly captures the full nonlinear dynamics of the PMSM, the explicit rotor-angle dependence and phase coupling remain present in the *abc* formulation. These characteristics complicate direct controller design. Consequently, further coordinate transformations are introduced in the following section to obtain a model better suited for field-oriented control.

2.5.3 Clarke–Park Transformation and dq Model

The Clarke–Park transformation constitutes the fundamental mathematical framework for control-oriented modeling of the Permanent Magnet Synchronous Machine (PMSM)[28, p. 87]. By applying these coordinate transformations, the position-dependent and strongly coupled *abc* model is converted into a rotating reference frame in which the steady-state electrical variables become constant. This representation enables the application of classical linear control techniques, such as PI regulation, to an otherwise nonlinear electromechanical system[15, p. 354, p. 427].

Clarke Transformation ($abc \to \alpha\beta$)

The first step is the Clarke transformation, which maps the three-phase stator variables into a two-axis stationary orthogonal reference frame[28, p. 98]. For a balanced three-phase system with zero neutral current ($i_a + i_b + i_c = 0$), the transformation can be written as:

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{2.13}$$

This transformation reduces the three-phase system to two orthogonal components while preserving power invariance[28, p. 112]. Under balanced conditions, the zero-sequence component vanishes and can be neglected for control purposes[5, p. 102].

Park Transformation ($\alpha\beta \to dq$)

The second step is the Park transformation, which rotates the stationary $\alpha\beta$ frame into a synchronous rotating reference frame aligned with the rotor magnetic field[5, p. 322]. The transformation is defined as:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{2.14}$$

where θ_e is the electrical rotor angle, related to the mechanical rotor angle θ_m by

$$\theta_e = p\theta_m \tag{2.15}$$

with p denoting the number of pole pairs[28, p. 55].

By aligning the d-axis with the rotor permanent magnet flux, the inductances become constant in the rotating frame and the time-varying coefficients present in the *abc* model are eliminated. In steady-state operation, sinusoidal phase currents correspond to constant d- and q-axis components[10, p. 100].

dq Voltage Equations

Applying the Clarke–Park transformation to the phase-domain voltage equations yields the following dynamic model in the synchronous reference frame[28, p. 126][5, p. 357]:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q \tag{2.16}$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} + \omega_e L_d i_d + \omega_e \lambda_m \tag{2.17}$$

where:

- L_d and L_q are the d- and q-axis inductances,
- $\omega_e = \frac{d\theta_e}{dt}$ is the electrical angular speed,

  The corresponding electrical angular speed is therefore

$$\omega_e = p\omega_m \tag{2.18}$$

- λ_m is the permanent magnet flux linkage.

The terms $\omega_e L_q i_q$ and $\omega_e L_d i_d$ represent speed-dependent cross-coupling between the axes, while the term $\omega_e \lambda_m$ corresponds to the back-electromotive force induced by rotor motion.

In the dq frame, the electrical dynamics are significantly simplified. Although coupling terms remain, they are algebraic and can be compensated through feedforward decoupling in the control algorithm.

Electromagnetic Torque Expression

The electromagnetic torque in the dq reference frame is given by[5, p. 357]:

$$T_e = \frac{3}{2} p \left(\lambda_m i_q + (L_d - L_q) i_d i_q\right) \tag{2.19}$$

For a surface-mounted PMSM (SPMSM), the rotor is non-salient and therefore $L_d = L_q$. In this case, the reluctance torque term vanishes and the torque expression simplifies to[6, p. 396]:

$$T_e = \frac{3}{2} p \lambda_m i_q \tag{2.20}$$

This linear relationship between electromagnetic torque and q-axis current forms the theoretical foundation of field-oriented control. By regulating i_q, the machine torque can be controlled directly, while the d-axis current is used to regulate the flux component[28, p. 140].

Chapter 3

Practical Work: Controller Model Implementations in PSIM

3.1 PMSM Drive System Architecture

The simulation platform developed for this work is based on a complete Permanent Magnet Synchronous Motor (PMSM) drive system composed of electrical and mechanical subsystems. The same plant model is used throughout this work for both the Field-Oriented Control (FOC) and Direct Torque Control (DTC) implementations, ensuring that all comparisons are performed under identical operating conditions

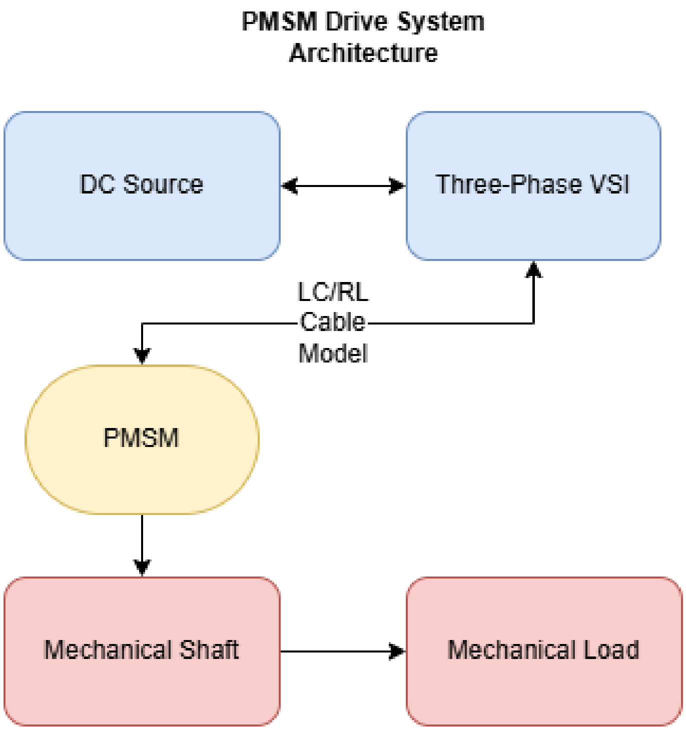


Figure 3.1: Simplified PMSM drive system architecture.

In figure 3.1 the electrical subsystem is highlighted in blue, while the mechanical subsystem is represented in red. The PMSM acts as the coupling element between both domains, converting electrical power into mechanical power through electromagnetic interaction.

The electrical subsystem comprises the DC source, the three-phase Voltage Source Inverter (VSI), the cable model, and the PMSM stator windings. The bidirectional arrows indicate that electrical power can flow in both directions, enabling regenerative operation. This part of the system is responsible for generating the controlled three-phase voltages and currents required for motor operation.

The mechanical subsystem comprises the rotor, mechanical shaft, and external mechanical load. This domain represents the conversion of electromagnetic torque into rotational motion and the interaction with external mechanical loads.

The PMSM is intentionally represented using a distinct colour because it forms the interface between the electrical and mechanical domains. Electrical quantities, such as phase voltages and currents, are converted into electromagnetic torque, which in turn drives the mechanical subsystem.

DC Source The drive system is supplied by a constant DC voltage source representing the energy storage or power conversion stage feeding the inverter. This source could be such as battery packs, rectified grid supplies, or DC buses found in industrial drive systems.

The DC source provides the energy required by the inverter to synthesize three-phase AC voltages. During operation, electrical power flows from the DC link towards the machine and is subsequently converted into mechanical power at the motor shaft.

Three-Phase Voltage Source Inverter The voltage source inverter (VSI) is responsible for converting the constant DC voltage into a set of controlled three-phase voltages.

The inverter consists of three bridge legs, each containing an upper and lower semiconductor switch. By controlling the switching states through Pulse Width Modulation (PWM), the inverter can generate a voltage vector with controllable magnitude, frequency, and phase.

From a control perspective, the inverter acts as the interface between the controller and the electrical machine. The commanded voltage references generated by the control algorithm are translated into gate signals that determine the switching sequence of the inverter devices.

Cable Model A simplified cable model is included between the inverter and the PMSM. The cable is represented by a series resistance and inductance.

Although the electrical parameters are relatively small, their inclusion improves simulation realism by accounting for voltage drops, current filtering effects, and parasitic energy storage between the inverter and the machine terminals.

Additionally, the cable model contributes to numerical stability and provides a more realistic representation of practical drive installations.

Permanent Magnet Synchronous Motor The PMSM constitutes the central element of the drive system. It converts electrical energy supplied by the stator windings into mechanical torque through the interaction between the rotating stator magnetic field and the permanent magnets embedded in the rotor.

The machine model implemented in PSIM includes both electrical and mechanical dynamics. On the electrical side, the stator voltages and currents determine the electromagnetic state of the machine. On the mechanical side, the generated electromagnetic torque interacts with the rotor inertia and external load torque to determine the resulting mechanical speed.

Because the PMSM simultaneously belongs to the electrical and mechanical domains, it acts as the energy conversion interface between both subsystems.

Mechanical Shaft The mechanical shaft transfers the electromagnetic torque generated by the PMSM to the external load. The shaft also represents the rotational inertia of the system, which determines the acceleration and deceleration dynamics according to Newton's rotational law.

Variations in electromagnetic torque produce changes in angular speed, while changes in load torque act as disturbances that must be rejected by the control system.

External Mechanical Load The load torque block represents the mechanical torque demanded by the driven application. Depending on the simulation scenario, the load may be constant, variable, or time-dependent.

The interaction between electromagnetic torque and load torque governs the mechanical operating point of the machine. At steady state, both torques become equal and the motor speed remains constant.

3.1.1 PSIM Implementation

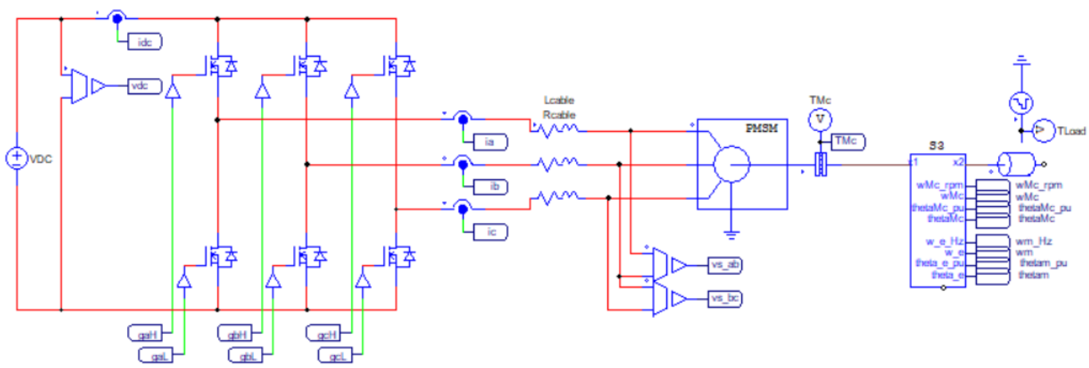


Figure 3.2: Implementation of the PMSM drive system in PSIM.

Figure 3.2 shows the complete implementation of the PMSM drive system in PSIM.

The inverter is implemented as a two-level three-phase voltage source converter supplied by a DC source.

3.1.1.1 Current Measurement Location

Accurate current feedback is essential for motor control applications, as it directly affects the performance of current controllers, observers, and estimation algorithms. Several current sensing strategies are commonly employed in PMSM drives, differing mainly in sensor placement and signal reconstruction requirements [29].

One approach is the *triple-shunt* configuration, in which a current sensor is placed in each inverter phase. This arrangement provides direct access to all phase currents and allows simultaneous current acquisition, making it particularly suitable for high-performance control schemes such as FOC. However, it requires three sensing circuits, increasing hardware cost and complexity [29].

A second option is the *dual-shunt* configuration, where only two phase currents are measured and the third current is reconstructed using Kirchhoff's current law,

$$i_a + i_b + i_c = 0.$$

This approach reduces hardware requirements while retaining most of the information available in a triple-shunt configuration. Its accuracy, however, depends on the assumption of a balanced three-phase system and proper synchronization of the measurement process [29].

The third alternative is *single-shunt* or *DC-link current sensing*, where a single sensor is placed in the DC bus. This solution offers the lowest hardware cost but requires phase-current reconstruction based on inverter switching states and PWM timing. Consequently, the associated control algorithms are considerably more complex, and continuous measurement of all phase currents is not always available [29], [30].

In the present work, current sensors are placed at the output of the voltage source inverter, immediately before the cable and PMSM connection. This location provides direct access to the stator currents required by the Clarke and Park transformations employed in FOC, as well as the current information used for flux and torque estimation in DTC. Furthermore, unlike DC-link sensing methods, the measured currents are continuously available and do not require PWM-dependent reconstruction algorithms.

Since the impedance of the cable connecting the inverter to the PMSM is very small, the measured currents can be considered equivalent to the currents entering the motor terminals. The selected sensor placement therefore provides a simple and accurate measurement solution while ensuring consistent current feedback for both control strategies investigated in this work.

3.1.1.2 Available Measurements

The PMSM model provides several electrical and mechanical quantities that are required by the implemented control algorithms. While phase currents and electromagnetic torque are obtained directly from the machine model, the speed and position signals used by the controllers are generated through the processing chain shown in Figure 3.3. The detailed PSIM implementation of the speed and position signal processing chain is provided in Appendix A for completeness.

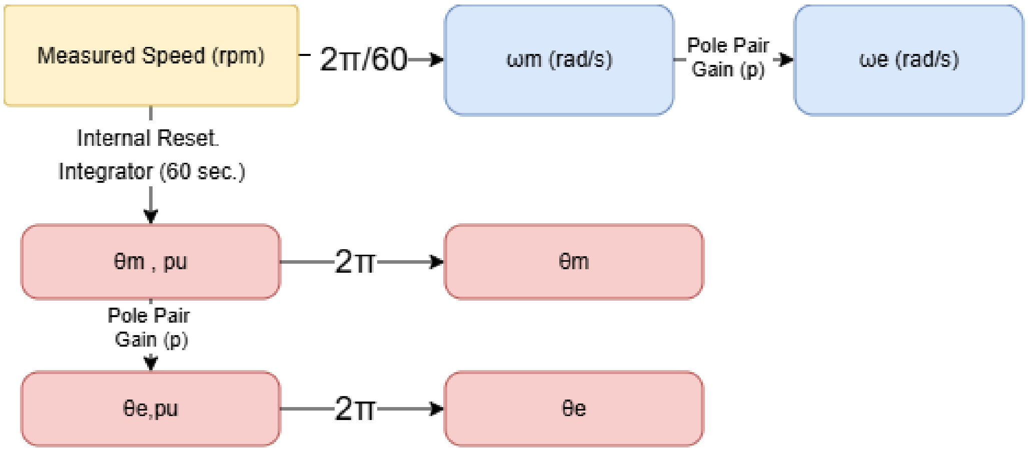


Figure 3.3: Generation of speed and position feedback signals from the measured rotor speed.

The process begins with the rotor speed provided by the PMSM model in revolutions per minute (rpm). This measured quantity serves as the basis for both the speed and position feedback paths.

In the speed-processing branch, the measured speed is converted into mechanical angular velocity according to

$$\omega_m = \frac{2\pi}{60} n_{rpm},$$

where n_{rpm} is the measured rotor speed. The electrical angular velocity is then obtained by multiplying the mechanical speed by the number of pole pairs p,

$$\omega_e = p\omega_m.$$

These quantities are used extensively throughout the control algorithms, particularly for reference-frame transformations and decoupling terms in the FOC implementation.

In parallel, the position-processing branch generates the rotor position signals. The measured speed is integrated using the internal reset integrator available within the PSIM model, producing the mechanical position in per-unit form, denoted by $\theta_{m,pu}$. In this representation, one per-unit corresponds to one complete mechanical revolution.

The mechanical position expressed in radians is obtained through

$$\theta_m = 2\pi\theta_{m,pu}.$$

The electrical position is subsequently calculated by scaling the mechanical position by the number of pole pairs,

$$\theta_{e,pu} = p\theta_{m,pu},$$

and converting the result into radians,

$$\theta_e = 2\pi\theta_{e,pu}.$$

Since electrical position is inherently periodic, the implementation additionally includes a modulo operation that constrains the electrical angle to a single electrical revolution before conversion to radians. This operation is omitted from Figure 3.3 for clarity.

Together, these signals provide the complete set of speed and position feedback variables required by the control system.

In addition to the processed speed and position signals, the PMSM drive model provides access to several electrical and mechanical quantities that are used for control, monitoring, and performance evaluation:

Table 3.1: Feedback signals available from the PMSM drive model.

Signal	**Description**
i_a, i_b, i_c	Stator phase currents
I_{dc}	DC-link current
V_{dc}	DC-link voltage
ω_m	Mechanical angular velocity
ω_e	Electrical angular velocity
θ_m	Mechanical rotor position
θ_e	Electrical rotor position
T_e	Electromagnetic torque

These measurements constitute the feedback information required for both the Field-Oriented Control and Direct Torque Control strategies developed in the following sections.

3.1.1.3 Reference PMSM Drive Parameters

To ensure consistency throughout the development, implementation, and evaluation of the proposed control strategies, a single PMSM drive configuration was adopted as the reference platform for all simulations presented in this work. The selected machine is based on the CubeMars RI100 KV105 frameless inrunner torque motor, a commercially available permanent magnet synchronous machine designed for robotics, servo-drive, and exoskeleton applications [31].

The motor was selected due to the availability of detailed manufacturer specifications, its compatibility with field-oriented control operation, and its representative characteristics of modern high-performance electromechanical systems. The same machine model, inverter configuration, and mechanical load arrangement are employed throughout all investigations, including the implementation of FOC, DTC, Hall-sensor integration, parameter estimation, and load-torque estimation techniques.

The manufacturer provides several machine parameters in line-to-line form. However, the PMSM model implemented in PSIM assumes a Y-connected stator and requires phase parameters as inputs. Consequently, the phase resistance and inductance values used in the simulations were obtained by dividing the corresponding line-to-line quantities by two. Additionally, the back-EMF constant supplied by the manufacturer was converted from its datasheet form to the peak line-to-line value required by the PSIM machine model.

The resulting parameter set adopted throughout this work is summarized in Table 3.2. Unless otherwise stated, the machine parameters were obtained from the manufacturer's published specifications [31].

Table 3.2: Reference PMSM drive parameters used throughout this work.

Category	**Parameter**	**Value**
PMSM	Motor model	CubeMars RI100 KV105
PMSM	Number of poles P	28
PMSM	Pole pairs p	14
PMSM	Stator resistance R_s	0.063 Ω
PMSM	d-axis inductance L_d	183.35 μH
PMSM	q-axis inductance L_q	183.35 μH
PMSM	Back-EMF constant $V_{pk}/krpm$	14.8 $V/krpm$
PMSM	Rotor inertia J	$2.16 \times 10^{-5}\ kg \cdot m^2$
PMSM	Rated torque	1.76 Nm
PMSM	Peak torque	4.95 Nm
PMSM	Rated current	13.6 A
PMSM	Peak current	38.6 A
PMSM	No-load speed @ 48 V	4368 rpm
Inverter	DC-link voltage V_{dc}	48 V
Inverter	Topology	Two-level three-phase VSI
PWM	Switching frequency f_{sw}	20 kHz
Control	Sampling frequency f_s	20 kHz

The selected parameter set represents a realistic low-voltage, high-torque PMSM drive suitable for robotics and servo-drive applications. By adopting a commercially available machine with documented characteristics, the simulation environment remains representative of practical motor-drive systems while providing a consistent benchmark for the comparative evaluation of the different control and estimation techniques developed in this thesis.

3.2 Direct Torque Control (DTC)

Direct Torque Control (DTC) is a control strategy for AC machines that directly regulates the electromagnetic torque and stator flux by selecting the appropriate inverter switching states. In contrast to Field-Oriented Control (FOC), which relies on cascaded current control loops and coordinate transformations for decoupled torque and flux regulation, DTC determines the inverter switching commands directly from the estimated machine states. This results in a comparatively simple control structure with fast dynamic response and reduced computational complexity.

The DTC implementation developed in this work follows the classical switching table approach. The controller continuously estimates the instantaneous stator flux and electromagnetic torque from the measured stator voltages and currents. These estimated quantities are compared with their respective reference values by hysteresis controllers, whose outputs, together with the instantaneous stator flux sector, determine the inverter voltage vector through a predefined switching table.

Figure 3.4 illustrates the overall signal flow of the implemented DTC algorithm.

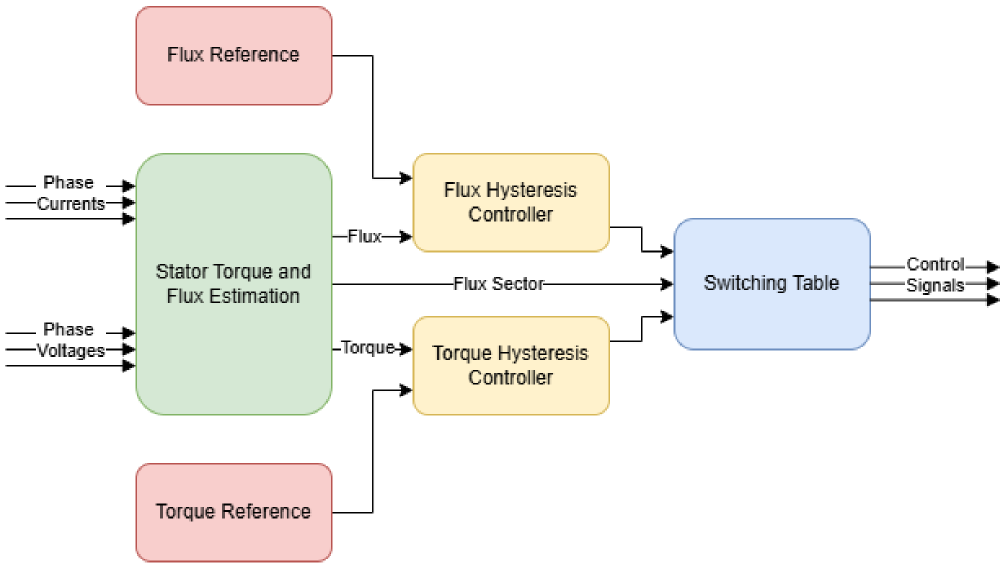


Figure 3.4: Overall structure of the implemented Direct Torque Control algorithm.

As shown in Figure 3.4, the measured three-phase stator currents and voltages serve as the primary inputs to the controller. These electrical quantities are processed by the Torque and Flux Estimation block, which calculates the instantaneous stator flux vector and electromagnetic torque. In addition to the flux magnitude, the angular position of the estimated stator flux vector is used to determine the corresponding flux sector.

The estimated stator flux is compared with the prescribed flux reference by the Flux Hysteresis Controller, generating a binary control signal that indicates whether the applied voltage vector should increase or decrease the stator flux magnitude. Similarly, the estimated electromagnetic torque is compared with the torque reference by the Torque Hysteresis Controller, producing a control signal that requests either an increase or a decrease in the developed torque.

Finally, the outputs of both hysteresis controllers, together with the identified flux sector, are supplied to the Switching (Look-up) Table. Based on these three inputs, the switching table selects the most appropriate inverter voltage vector to achieve the desired changes in torque and stator flux. The resulting control signals are then applied to the voltage source inverter, completing the DTC control loop.

The individual functional blocks shown in Figure 3.4 are described in detail in the following sections, beginning with the stator flux and torque estimation process.

3.2.1 Stator Flux and Torque Estimation

Accurate estimation of the stator flux vector and electromagnetic torque is fundamental to the operation of Direct Torque Control. Unlike current-controlled strategies, DTC relies directly on these estimated quantities to determine the appropriate inverter switching state. In the implemented controller, the estimation process begins by transforming the measured electrical quantities into the stationary $\alpha\beta$ reference frame, after which the stator flux vector and electromagnetic torque are computed.

3.2.1.1 $\alpha\beta$ Coordinate Transformation

The available measurements consist of the phase currents i_a and i_b, together with the line-to-line voltages v_{ab} and v_{bc}. Before the stator flux can be estimated, these quantities are transformed into the stationary $\alpha\beta$ reference frame using a power-invariant Clarke transformation, as represented visually in Figure 2.11. The implemented transformation is expressed as

$$i_\alpha = \sqrt{\frac{3}{2}}\, i_a, \tag{3.1}$$

$$i_\beta = \frac{1}{\sqrt{2}} i_a + \sqrt{2}\, i_b, \tag{3.2}$$

$$v_\alpha = \sqrt{\frac{2}{3}}\, v_{ab} + \frac{1}{\sqrt{6}} v_{bc}, \tag{3.3}$$

$$v_\beta = \frac{1}{\sqrt{2}} v_{bc}. \tag{3.4}$$

The resulting current and voltage components constitute the inputs for the stator flux estimator described in the following subsection.

3.2.1.2 Stator Flux Estimation

The stator flux vector is estimated using the stator voltage equation expressed in the stationary $\alpha\beta$ reference frame. Neglecting measurement offsets and assuming knowledge of the stator resistance, the stator flux components are obtained by integrating the difference between the applied stator voltage and the resistive voltage drop:

$$\psi_\alpha = \int (v_\alpha - R_s i_\alpha)\, dt, \tag{3.5}$$

$$\psi_\beta = \int \left(v_\beta - R_s i_\beta\right)\, dt. \tag{3.6}$$

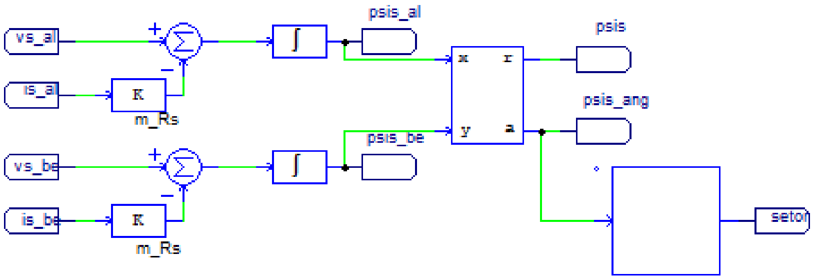


Figure 3.5: Implementation of the stator flux estimator in PSIM.

The estimated Cartesian components are subsequently converted into polar coordinates, yielding both the magnitude and angular position of the stator flux vector. The flux magnitude is calculated as

$$|\psi_s| = \sqrt{\psi_\alpha^2 + \psi_\beta^2}, \tag{3.7}$$

while the flux angle is determined using

$$\theta_\psi = \text{atan2}\left(\psi_\beta, \psi_\alpha\right). \tag{3.8}$$

The angular position of the stator flux vector is then used to identify the active sector of the $\alpha\beta$ plane. The complete electrical revolution is divided into six sectors of 60° each, numbered from 1 to 6. The implemented sector determination algorithm assigns the estimated flux angle to the corresponding sector according to predefined angular intervals. This sector information is subsequently used by the switching table to select the appropriate inverter voltage vector.

3.2.1.3 Electromagnetic Torque Estimation

Once the stator flux components have been estimated, the electromagnetic torque can be calculated directly from the cross product between the stator flux vector and the stator current vector expressed in the stationary $\alpha\beta$ reference frame.

Figure 3.6 shows the corresponding implementation in PSIM.

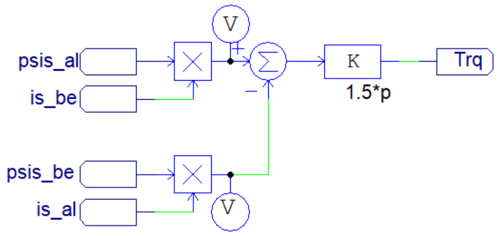


Figure 3.6: Electromagnetic torque estimation implemented in PSIM.

The estimated electromagnetic torque is computed as

$$T_e = \frac{3}{2} p \left(\psi_\alpha i_\beta - \psi_\beta i_\alpha \right), \tag{3.9}$$

where p denotes the number of pole pairs of the machine.

This expression represents the instantaneous electromagnetic torque produced by the interaction between the stator flux vector and the stator current vector. The estimated torque is subsequently compared with the reference value by the torque hysteresis controller, which determines whether the applied voltage vector should increase or decrease the generated torque.

3.2.2 Hysteresis Controllers

After estimating the stator flux magnitude and the electromagnetic torque, both quantities are compared with their corresponding reference values. The resulting errors are processed by hysteresis controllers, whose outputs determine whether the selected inverter voltage vector should increase, decrease, or maintain the controlled variable.

Figure 3.7 illustrates the implementation of the hysteresis controllers and their interaction with the switching table.

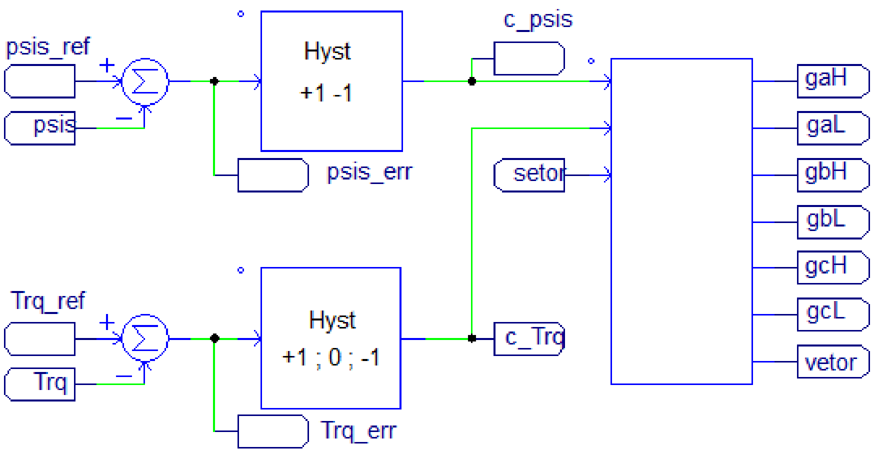


Figure 3.7: Flux and torque hysteresis controllers together with the switching table input signals.

The flux and torque controllers employ different hysteresis structures due to the distinct dynamic requirements of the two controlled variables.

3.2.2.1 Flux Hysteresis Controller

The stator flux magnitude is regulated by a two-level hysteresis controller. The controller receives the flux error

$$e_{\psi} = \psi_s^{\text{ref}} - \psi_s, \tag{3.10}$$

where ψ_s^{ref} is the reference stator flux magnitude and ψ_s is the estimated value.

The controller operates within a symmetric hysteresis band centred around zero. In the implemented controller, switching occurs whenever the error exceeds the predefined thresholds of ± 0.0005 Wb. Consequently, the controller output assumes only two possible values,

$$c_{\psi} = \begin{cases} +1, & e_{\psi} > +0.0005, \\ -1, & e_{\psi} < -0.0005. \end{cases} \tag{3.11}$$

Between both thresholds, the previous output state is maintained. This memory effect prevents unnecessary switching caused by small oscillations or measurement noise around the reference value.

The output signal c_{ψ} provides a simple indication of the required control action. A value of $+1$ requests an increase in the stator flux magnitude, whereas a value of -1 requests a reduction. The switching table subsequently interprets this command and selects the inverter voltage vector that produces the desired flux variation.

Stator Flux Reference In the implemented Direct Torque Control (DTC) scheme, the stator flux reference is kept constant and is defined as

$$\psi_s^{\text{ref}} = 0.011 \text{ Wb}. \tag{3.12}$$

For permanent magnet synchronous machines, this reference is selected according to the nominal stator flux of the motor, which is predominantly determined by the permanent magnet flux linkage. It can be approximated from the machine back-electromotive force constant as

$$\psi_f = \frac{V_{\text{pk}}}{\omega_e}, \tag{3.13}$$

where V_{pk} is the peak phase back-EMF and ω_e is the electrical angular speed. The adopted value of $\psi_s^{\text{ref}} = 0.011$ Wb corresponds closely to the estimated permanent magnet flux linkage of the simulated machine and remains fixed during all operating conditions considered in this work.

3.2.2.2 Torque Hysteresis Controller

Unlike the stator flux, the electromagnetic torque is regulated by a three-level hysteresis controller. This additional output state enables a more flexible voltage vector selection and reduces unnecessary switching when the torque error is already sufficiently close to zero.

The controller receives the torque error

$$e_T = T_e^{\mathrm{ref}} - T_e, \tag{3.14}$$

where T_e^{ref} denotes the reference torque generated by the outer speed controller and T_e is the estimated electromagnetic torque.

Two switching thresholds are defined at ± 0.5 Nm together with an intermediate zero state. Accordingly, the controller output is

$$c_T = \begin{cases} +1, & e_T > 0.5, \\ 0, & |e_T| \leq 0.5 \text{ (depending on the previous state)}, \\ -1, & e_T < -0.5. \end{cases} \tag{3.15}$$

Unlike a conventional three-level hysteresis comparator, the implemented controller incorporates state-dependent memory through the retention of its previous output state. As a result, the transition to the neutral output level is determined not only by the instantaneous torque error but also by the controller's current state.

When the controller output is $+1$, it remains in this state until the torque error becomes negative, at which point the output transitions to the neutral level (0). Similarly, when the output is -1, the controller does not immediately switch to zero upon entering the hysteresis band but only after the torque error becomes positive.

Consequently, the implemented algorithm exhibits an additional hysteresis effect around the zero crossing, reducing output chattering and avoiding unnecessary switching between adjacent states. This state-dependent behaviour provides smoother operation and improves the robustness of the voltage vector selection performed by the switching table.

Electromagnetic Torque Reference Unlike the stator flux reference, the torque reference is not prescribed directly. Instead, it is generated by an outer speed control loop that regulates the mechanical speed of the machine.

The controller computes the speed error as

$$e_\omega = \omega_{\mathrm{ref}} - \omega_m, \tag{3.16}$$

where ω_{ref} is the reference speed and ω_m is the measured mechanical speed.

A proportional–integral (PI) controller processes this error to generate the required torque reference according to

$$T_e^{\text{ref}} = K_p e_\omega + K_i \int e_\omega \, dt, \tag{3.17}$$

The proportional term provides an immediate response to speed deviations, while the integral term compensates for steady-state errors by accumulating the speed error over time. The resulting torque demand is subsequently limited to the interval

$$-2\ \text{Nm} \leq T_e^{\text{ref}} \leq 2\ \text{Nm}, \tag{3.18}$$

preventing excessive torque requests that exceed the desired operating range of the drive system.

3.2.3 Switching Table

The switching table constitutes the decision-making stage of the Direct Torque Control algorithm. Its purpose is to select the inverter voltage vector that best satisfies the requests generated by the flux and torque hysteresis controllers while taking into account the instantaneous position of the stator flux vector.

The switching table receives three inputs:

- the output of the flux hysteresis controller, $c_\psi \in \{-1, +1\}$,
- the output of the torque hysteresis controller, $c_T \in \{-1, 0, +1\}$,
- the estimated stator flux sector, $S \in \{1, \ldots, 6\}$.

Based on these three quantities, the controller selects one of the six active voltage vectors or one of the two zero voltage vectors of the two-level voltage source inverter. The selected vector is then translated into the corresponding inverter gate signals and applied during the next switching interval.

The selected voltage vector is subsequently converted into the corresponding switching states of the three-phase voltage source inverter. In the implemented controller, the active vectors V_1 to V_6 and the zero vectors V_0 and V_7 are represented by the upper switch states (g_{aH}, g_{bH}, g_{cH}), while the lower gate commands are generated as their complementary signals,

$$g_{aL} = 1 - g_{aH}, \tag{3.19}$$

$$g_{bL} = 1 - g_{bH}, \tag{3.20}$$

$$g_{cL} = 1 - g_{cH}. \tag{3.21}$$

This complementary gating strategy guarantees that only one switch in each inverter leg is conducting at any instant, thereby preventing shoot-through conditions. The resulting six gate signals are finally applied to the voltage source inverter, producing the voltage vector required to regulate both the stator flux magnitude and the electromagnetic torque.

3.2.4 DTC Validation Results

To validate the implemented Direct Torque Control strategy, a series of simulation tests was conducted under different operating conditions representative of typical industrial and robotic drive applications. The objective of these tests is to verify the controller's ability to accurately track speed references and reject external load disturbances while highlighting the dynamic characteristics of the implemented DTC algorithm.

The validation campaign consists of three complementary scenarios. The first evaluates the speed tracking capability under no-load conditions, the second investigates the controller response to abrupt load torque variations at constant speed, and the third combines speed and load changes in a representative robotic duty cycle. These test cases will later serve as a common benchmark for the comparison with the implemented Field-Oriented Control (FOC) strategy.

3.2.4.1 Speed Tracking Test

The first validation test evaluates the ability of the DTC controller to follow successive speed reference changes without the influence of external mechanical loading. During the simulation, the load torque is maintained at 0 Nm while the speed reference follows the profile summarized in Table 3.3.

Table 3.3: Speed reference profile used for the speed tracking test.

Time (s)	Reference speed (rpm)
0 – 0.025	0
0.0251 – 0.100	500
0.1001 – 0.200	1000
0.2001 – 0.250	200
0.2501 – 0.400	-500

Figure 3.8 presents the corresponding simulation results.

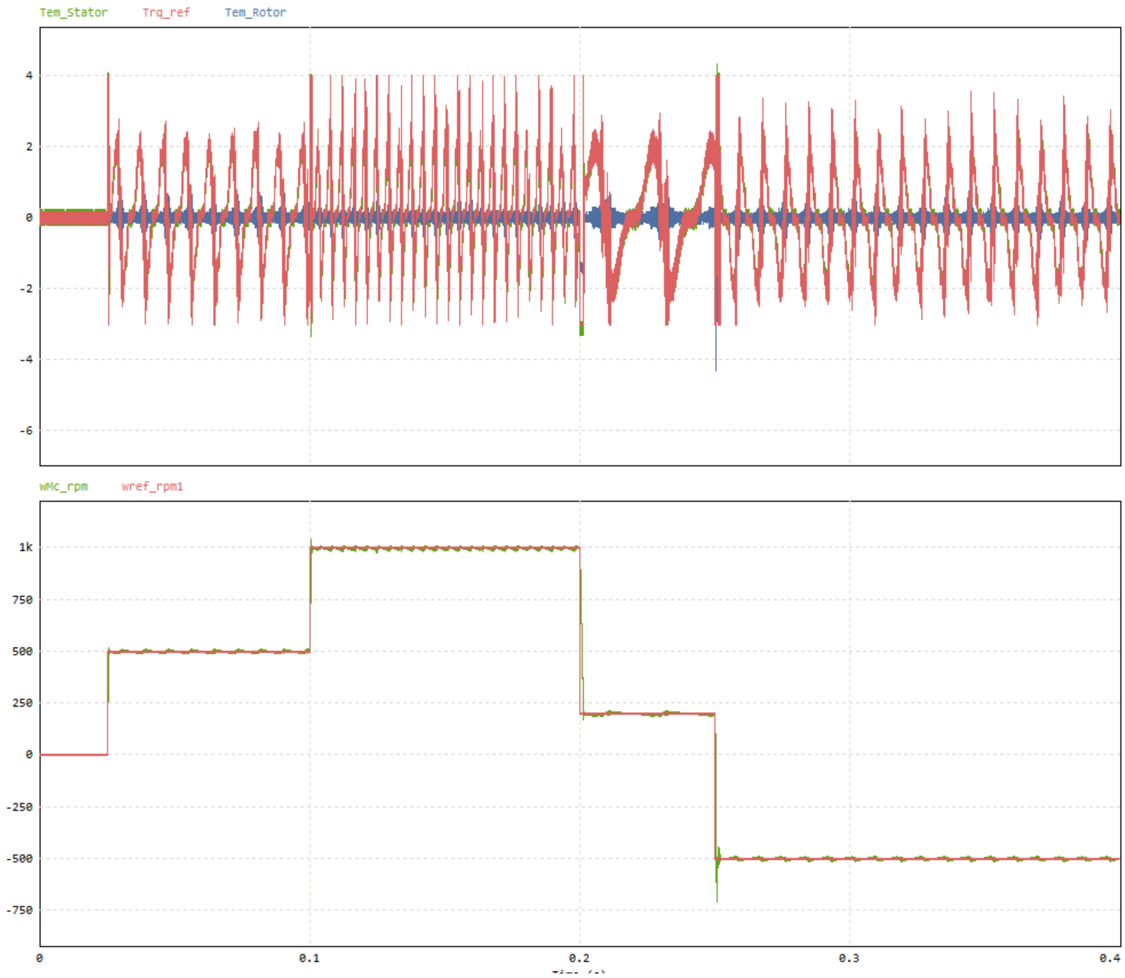


Figure 3.8: DTC speed tracking validation results.

The implemented controller successfully follows all commanded speed references with rapid dynamic response and negligible steady-state error. Small oscillations are visible around the reference values, reflecting the inherent ripple associated with classical hysteresis-based DTC. During the speed transitions, temporary overshoots are observed, with the largest transient occurring during the reversal from 200 rpm to -500 rpm. Nevertheless, the controller rapidly converges to the desired operating point while maintaining stable operation throughout the test.

3.2.4.2 Load Disturbance Rejection Test

The second validation scenario investigates the disturbance rejection capability of the controller under constant speed operation. The speed reference is fixed at 500 rpm while the applied load torque follows the sequence listed in Table 3.4.

Table 3.4: Load torque profile used for the disturbance rejection test.

Time (s)	Load torque (Nm)
0 – 0.025	0
0.0251 – 0.100	1
0.1001 – 0.175	2
0.1751 – 0.250	0
0.2501 – 0.325	-1
0.3251 – 0.400	2

The corresponding simulation results are shown in Figure 3.9.

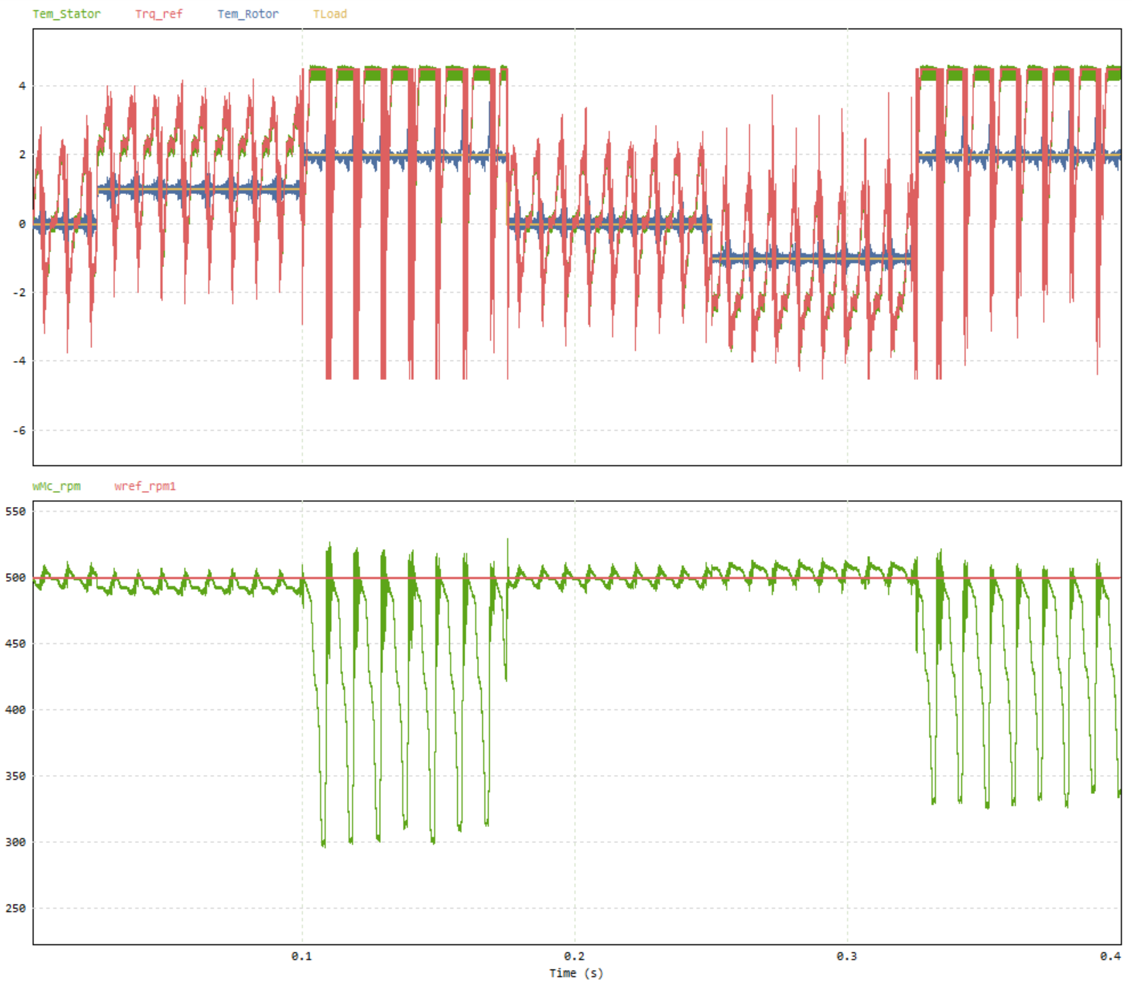


Figure 3.9: DTC load disturbance rejection validation results.

For moderate loading conditions, the controller maintains satisfactory speed regulation while producing the required electromagnetic torque to compensate for the applied disturbance. The characteristic torque ripple of classical DTC remains visible throughout the simulation but does not compromise the overall stability of the system.

When the applied load reaches 2 Nm, periodic reductions in rotational speed become apparent, indicating that the operating point approaches the practical torque capability of the motor as expected since its rated torque is around 1.8Nm. Although the average speed remains close to the desired reference, the increased oscillations reveal a degradation in tracking performance under high-load conditions.

3.2.4.3 Realistic Robotic Duty Cycle

The final validation scenario combines simultaneous speed and load variations to emulate a simplified robotic actuator duty cycle involving acceleration, payload manipulation and positioning tasks. The imposed speed and load profiles are summarized in Tables 3.5 and 3.6.

Table 3.5: Speed reference profile for the robotic duty cycle test.

Time (s)	Reference speed (rpm)
0 – 0.025	0
0.0251 – 0.100	300
0.1001 – 0.200	300
0.2001 – 0.300	600
0.3001 – 0.375	600
0.3751 – 0.450	100
0.4501 – 0.500	0

Table 3.6: Load torque profile for the robotic duty cycle test.

Time (s)	Load torque (Nm)
0 – 0.025	0
0.0251 – 0.100	0
0.1001 – 0.175	1.5
0.1751 – 0.300	1.5
0.3001 – 0.375	0
0.3751 – 0.450	2
0.4501 – 0.500	0

Figure 3.10 shows the obtained simulation results.

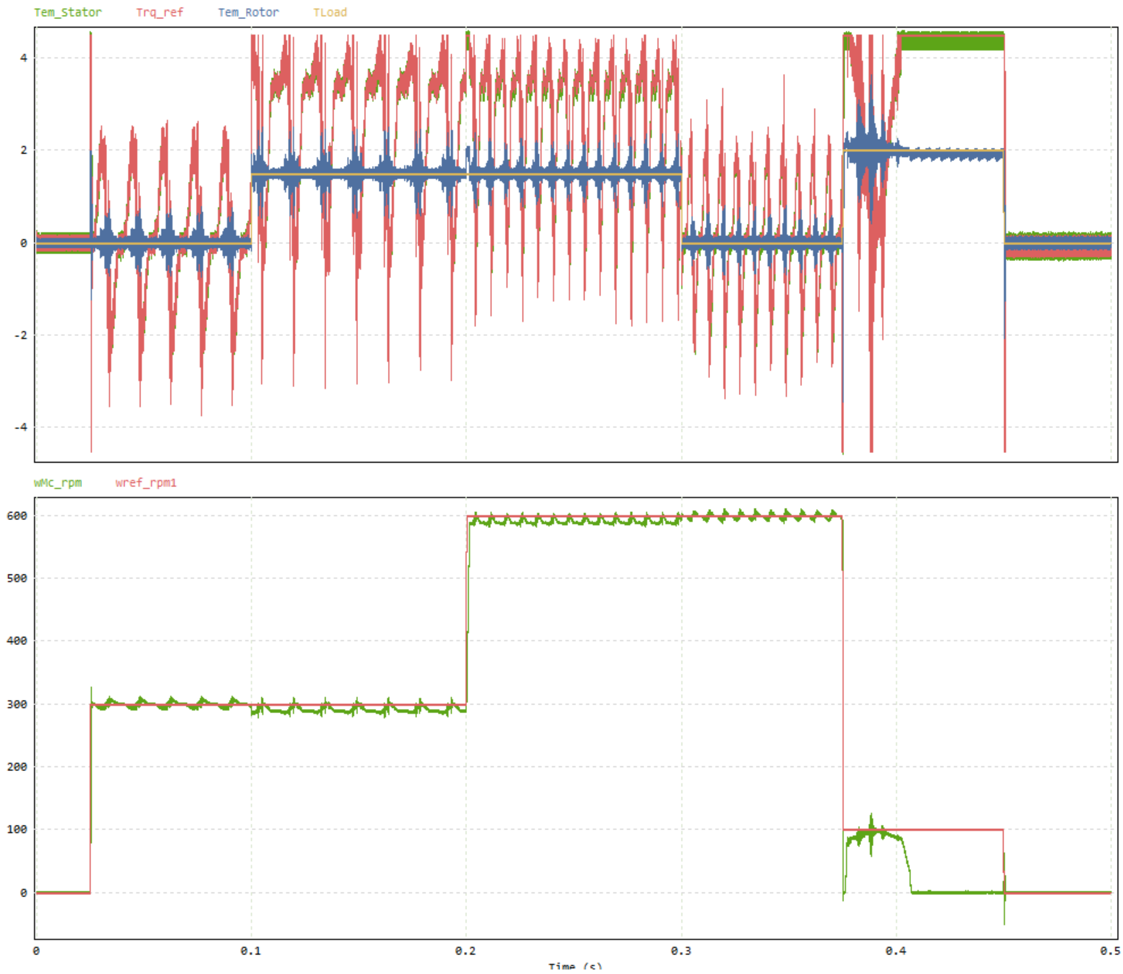


Figure 3.10: DTC validation under a representative robotic duty cycle.

Throughout the unloaded and moderate-load intervals, the controller accurately follows the commanded speed references while maintaining stable operation despite the presence of electromagnetic torque ripple. During the intervals with an applied load of 1.5 Nm, the controller continues to regulate the motor speed with only minor oscillations.

The most demanding operating condition occurs when a load torque of 2 Nm is applied while the reference speed is reduced to 100 rpm. Under these conditions, the speed regulation deteriorates and larger oscillations are observed, indicating that the requested operating point approaches the practical performance limits of the implemented DTC strategy. Once the external load is removed, the controller rapidly returns the machine to the commanded operating point.

Overall, the conducted validation tests demonstrate that the implemented Direct Torque Control algorithm provides accurate speed tracking and satisfactory dynamic performance over a wide range of operating conditions. The simulations also confirm the expected characteristics of classical DTC, namely the presence of electromagnetic torque ripple and a reduction in speed regulation capability when operating close to the maximum available torque at low speeds.

3.3 Field Oriented Control (FOC)

Field-Oriented Control (FOC) is a vector control strategy that enables independent regulation of the torque- and flux-producing current components of a permanent magnet synchronous motor (PMSM). By transforming the measured three-phase stator currents into a rotating reference frame aligned with the rotor magnetic field, the electrical dynamics of the machine can be decoupled, allowing the motor to be controlled in a manner analogous to a separately excited DC motor. This approach provides excellent dynamic performance, accurate torque control, and high steady-state efficiency over a wide operating range.

The implemented controller follows a cascaded signal-processing structure in which the measured phase currents are conditioned and transformed into the rotating *d*-*q* reference frame before being processed by independent current controllers. The resulting voltage commands are subsequently limited according to the available DC bus voltage, transformed back into the stationary reference frame, and finally converted into pulse-width modulation (PWM) signals that drive the inverter switches.

Figure 3.11 presents the overall architecture of the implemented FOC algorithm and illustrates the complete signal flow from the measured motor currents to the generated PWM gate signals.

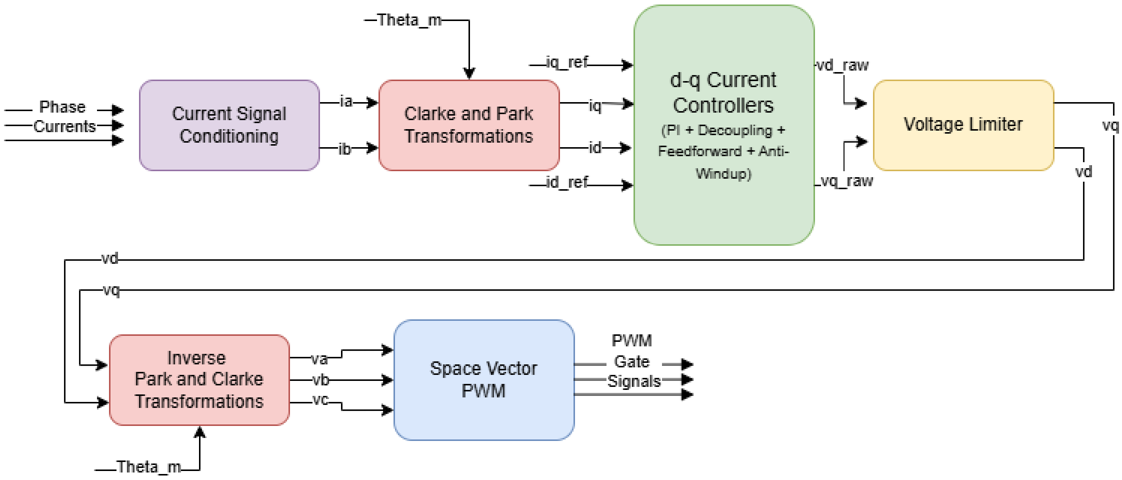


Figure 3.11: Overall structure of the implemented Field-Oriented Control (FOC) algorithm.

For clarity, the implementation is described in the same order as the signal propagates through the controller. The discussion begins with the current measurement and conditioning stage, followed by the Clarke and Park transformations, the *d*- and *q*-axis current control loops including decoupling and feedforward compensation, the voltage limiting stage, the inverse coordinate transformations, and finally the space vector pulse-width modulation (SVPWM) algorithm responsible for generating the inverter switching signals.

3.3.1 Current Measurement and Signal Conditioning

The field-oriented control algorithm requires accurate measurements of the stator phase currents to regulate the torque- and flux-producing current components. In the implemented drive system, the phase currents are measured at the output of the inverter, immediately before the terminals of the PMSM. The measured currents are then processed through a signal conditioning stage prior to being used by the controller.

Figure 3.12 illustrates the complete current measurement and conditioning chain implemented in PSIM.

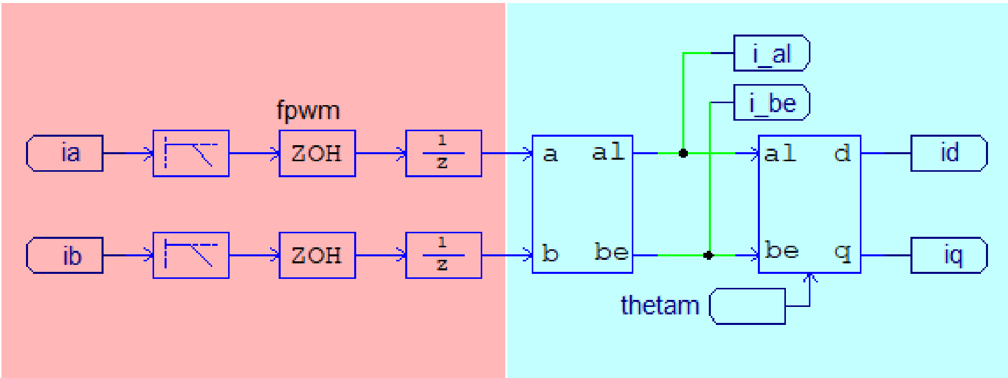


Figure 3.12: Current measurement and signal conditioning stage of the implemented FOC algorithm, highlighted in red.

The measured phase currents i_a and i_b are first passed through second-order low-pass filters with unity gain, a cut-off frequency of 100 kHz, and a damping ratio of 0.7. The purpose of these filters is to attenuate high-frequency switching noise and measurement disturbances while preserving the fundamental current components required for control.

After filtering, the signals are sampled using a zero-order hold (ZOH) block operating at the PWM switching frequency, f_{PWM}. This stage emulates the behaviour of a digital data acquisition system, where the measured values are updated once every control period and held constant until the next sampling instant.

The sampled currents subsequently pass through a unit delay block with a sampling frequency of 5 kHz, representing the discrete execution of the digital controller and introducing a one-sample computational delay. This delay also prevents algebraic loops within the simulation model and reflects the practical implementation of the control algorithm on digital hardware.

The conditioned current measurements are then provided to the coordinate transformation stage, where they are converted from the stationary reference frame into the rotating d-q reference frame used by the current controllers. The mathematical formulation of these transformations is presented in the following subsection.

3.3.2 Clarke and Park Transformations

The current controllers employed in the field-oriented control algorithm operate in the rotating d-q reference frame, where the flux- and torque-producing current components can be regulated independently. Consequently, the conditioned phase current measurements must first be transformed from the stationary three-phase reference frame into this rotating coordinate system.

Figure 3.12 highlights in blue the coordinate transformation stage implemented in the PSIM model.

The first step is the Clarke transformation, which converts the measured phase currents into the stationary orthogonal α-β reference frame. Assuming a balanced three-phase system, the transformation is given by

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \end{bmatrix}, \tag{3.22}$$

where only the measured currents i_a and i_b are required.

The resulting stationary current components are subsequently transformed into the rotating synchronous reference frame through the Park transformation. Using the electrical rotor position θ_m, the transformation is expressed as

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta_m & \sin\theta_m \\ -\sin\theta_m & \cos\theta_m \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}. \tag{3.23}$$

The direct-axis current component, i_d, is aligned with the rotor magnetic flux and primarily determines the machine flux linkage, whereas the quadrature-axis current component, i_q, is orthogonal to the rotor flux and is directly responsible for electromagnetic torque production. For surface-mounted PMSMs, maximum torque per ampere operation is typically achieved by regulating the direct-axis current reference to zero, such that the electromagnetic torque is controlled exclusively through the q-axis current.

The transformed currents i_d and i_q are then supplied to the corresponding current control loops, where they are independently regulated to track their respective reference values.

3.3.3 d-q Current Controllers

After the measured stator currents have been transformed into the synchronous rotating reference frame, the resulting d- and q-axis current components are independently regulated by two current control loops. The objective of these controllers is to ensure that the measured currents accurately track their respective reference values, thereby enabling direct control of the machine flux and electromagnetic torque.

Figure 3.13 shows the implementation of the d- and q-axis current controllers used in the developed FOC algorithm.

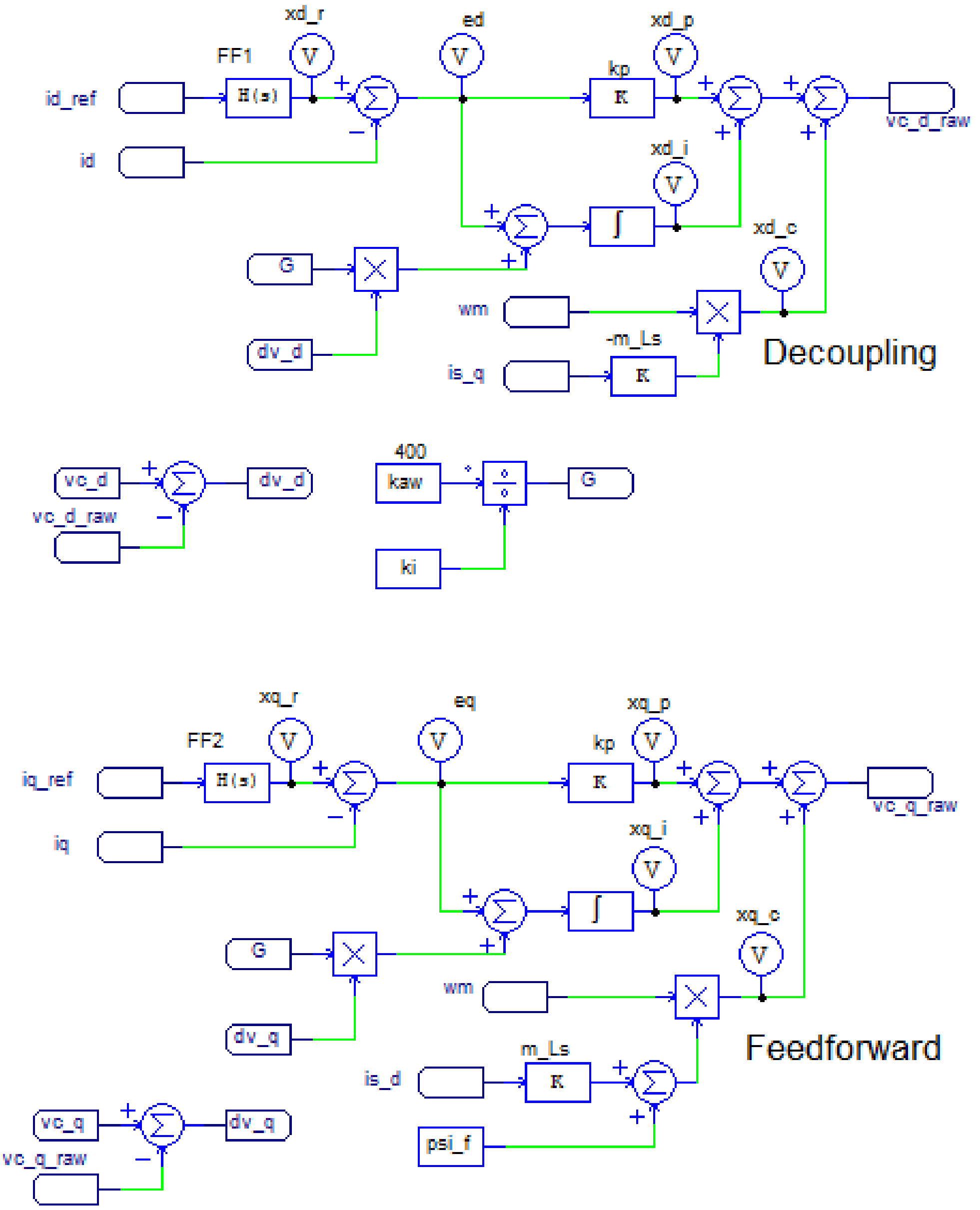


Figure 3.13: d- and q-axis current control loops with PI regulation, decoupling, feedforward compensation and anti-windup implementation.

The controller consists of two independent PI regulators operating in the synchronous reference frame. The current errors are computed as

$$e_d = i_{d,\mathrm{ref}} - i_d, \tag{3.24}$$

$$e_q = i_{q,\mathrm{ref}} - i_q, \tag{3.25}$$

where $i_{d,\mathrm{ref}}$ and $i_{q,\mathrm{ref}}$ denote the reference currents generated by the outer control loops.

Before the error calculation, the reference signals are processed through first-order prefilters (FF1 and FF2). These filters smooth abrupt reference variations and reduce high-frequency transients that could otherwise excite the inner current loops.

Each error signal is then processed by a proportional-integral (PI) controller of the form

$$v_{\mathrm{PI}} = K_p e + K_i \int e\,dt, \tag{3.26}$$

where K_p and K_i are the proportional and integral gains, respectively. The proportional term provides a fast response to current deviations, while the integral term eliminates steady-state tracking errors.

Since the PMSM equations in the synchronous reference frame contain speed-dependent coupling terms, direct application of PI control would result in interactions between the two current loops. To compensate for these effects, decoupling and feedforward terms are incorporated into the controller.

For the d-axis controller, the compensated voltage reference is given by

$$v_{d,\mathrm{raw}} = v_{\mathrm{PI},d} - \omega_e L_q i_q, \tag{3.27}$$

where ω_e is the electrical rotor speed and L_q is the q-axis inductance. This term compensates for the cross-coupling introduced by the rotating reference frame and improves the dynamic response of the controller.

Similarly, the q-axis controller generates

$$v_{q,\mathrm{raw}} = v_{\mathrm{PI},q} + \omega_e L_d i_d + \omega_e \psi_f, \tag{3.28}$$

where L_d is the d-axis inductance and ψ_f is the permanent magnet flux linkage. The first additional term compensates for the dynamic coupling with the d-axis current, while the second acts as a feedforward compensation for the back electromotive force generated by the permanent magnets.

To prevent excessive accumulation of the integral term when the voltage references are saturated, an anti-windup mechanism is included in both control loops. The difference between the limited output voltage and the raw controller output is scaled by the anti-windup gain k_{aw} and fed back to the integrator input, reducing integrator windup and improving recovery after saturation.

The outputs of the two current controllers are the unconstrained voltage references $v_{d,\mathrm{raw}}$ and $v_{q,\mathrm{raw}}$, which are subsequently processed by the voltage limiting stage.

3.3.4 Space Vector PWM Generation

The voltage references generated by the d-q current controllers are transformed back into the stationary three-phase reference frame to obtain the commanded phase voltages v_a, v_b, and v_c. Before being supplied to the modulation stage, these voltages are processed by a voltage limiting algorithm to ensure that the requested voltage vector remains within the achievable operating region of the inverter. The implementation of this voltage limiting procedure is presented in Appendix B.4.1.

Following the voltage limiting stage, the normalized phase voltage references are applied to the Space Vector Pulse Width Modulation (SVPWM) algorithm, as illustrated in Figure 3.14.

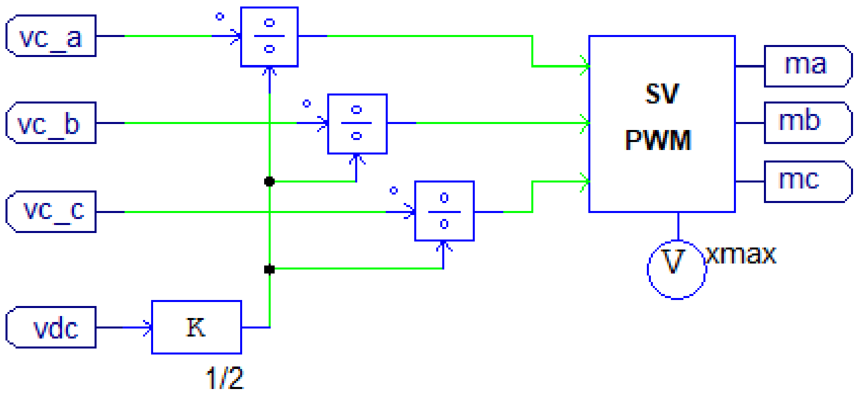


Figure 3.14: SVPWM generation stage of the implemented FOC algorithm.

The three phase voltage references are first normalized with respect to half of the DC bus voltage according to

$$m_a = \frac{v_a}{V_{dc}/2}, \qquad m_b = \frac{v_b}{V_{dc}/2}, \qquad m_c = \frac{v_c}{V_{dc}/2}, \tag{3.29}$$

where V_{dc} denotes the inverter DC-link voltage. This normalization produces the modulation signals required by the SVPWM block.

The SVPWM algorithm determines the appropriate inverter switching states by synthesizing the desired voltage vector through a sequence of active and zero voltage vectors within each switching period. Compared with conventional sinusoidal PWM, SVPWM provides improved DC bus voltage utilization and reduced harmonic distortion, making it a widely adopted modulation strategy for high-performance motor drives.

The outputs of the SVPWM block are the modulation signals m_a, m_b, and m_c, which are subsequently used to generate the gate signals for the three-phase voltage source inverter supplying the PMSM.

3.3.5 FOC Validation Results

To evaluate the performance of the implemented Field-Oriented Control algorithm, three validation tests were conducted. The tests assess the controller's ability to accurately track speed references, reject external load disturbances, and operate under a representative robotic motion profile. In all cases, the controller performance is evaluated through the motor speed response and electromagnetic torque behaviour.

The presented plots show the reference speed and measured rotor speed in the lower graph, while the upper graph presents the commanded torque, stator electromagnetic torque, rotor torque, and applied load torque.

3.3.5.1 Speed Tracking Test

The first validation test evaluates the ability of the controller to accurately follow speed reference variations under no-load conditions. The applied speed profile is defined as

Table 3.7: Speed reference profile for the speed tracking test.

Time (s)	Reference speed (rpm)
0.0000 – 0.0500	0
0.0501 – 0.2000	300
0.2001 – 0.4000	1000
0.4001 – 0.5000	200
0.5001 – 0.8000	-500

with an applied load torque of

$$T_L = 0 \text{ Nm}. \tag{3.30}$$

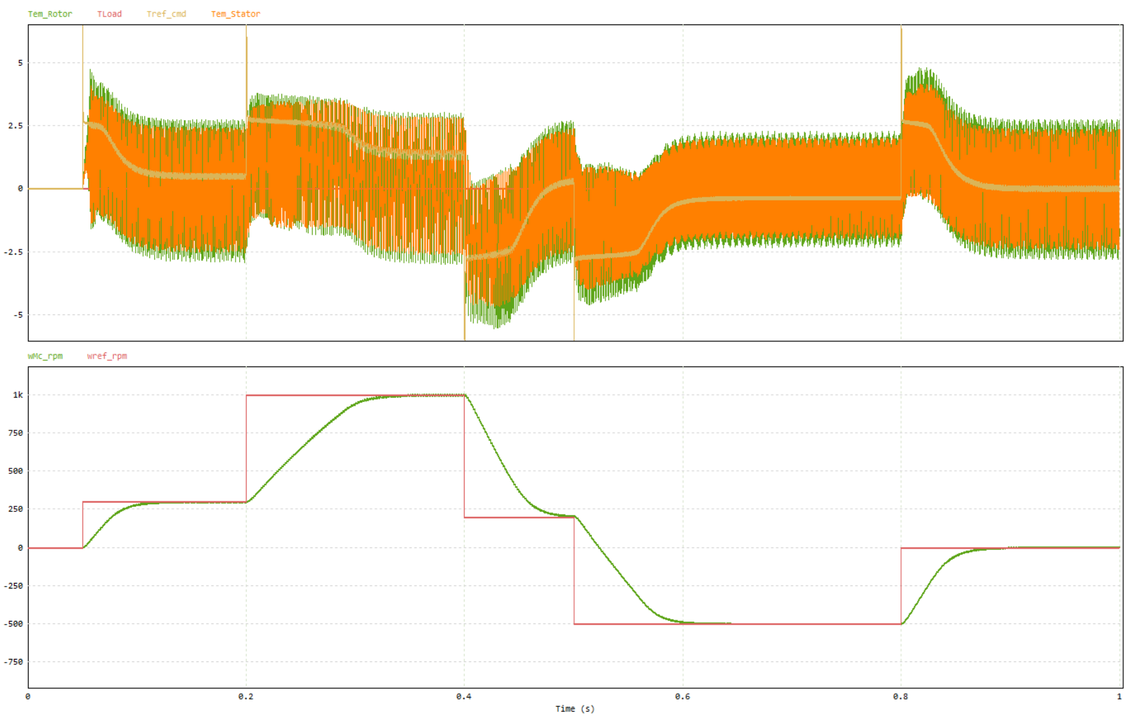


Figure 3.15: FOC speed tracking test under no-load conditions.

Figure 3.15 demonstrates that the implemented FOC controller is capable of accurately tracking all commanded speed transitions. The measured rotor speed follows the reference without observable overshoot and exhibits negligible steady-state ripple. The controller reaches the new operating point approximately 0.1 s after large speed variations of approximately 700 rpm, while smaller speed changes in the order of 300 rpm settle within approximately 0.03 s.

The torque response shows that the controller generates the required electromagnetic torque during acceleration and deceleration intervals while maintaining smooth steady-state operation. The controller exhibits a well-damped response throughout the entire speed range, including speed reversal operation, without observable overshoot or oscillatory behaviour.

3.3.5.2 Load Disturbance Rejection Test

The second validation test evaluates the disturbance rejection capability of the controller while maintaining a constant speed reference of

$$\omega_{ref} = 500 \text{ rpm}. \tag{3.31}$$

The applied load torque profile is

Table 3.8: Load torque profile for the load disturbance rejection test.

Time (s)	Load torque (Nm)
0.0000 – 0.1500	0
0.1501 – 0.3000	1
0.3001 – 0.4000	2
0.4001 – 0.5500	0
0.5501 – 0.7000	-1
0.7001 – 0.8000	2

where the torque values are given in Nm.

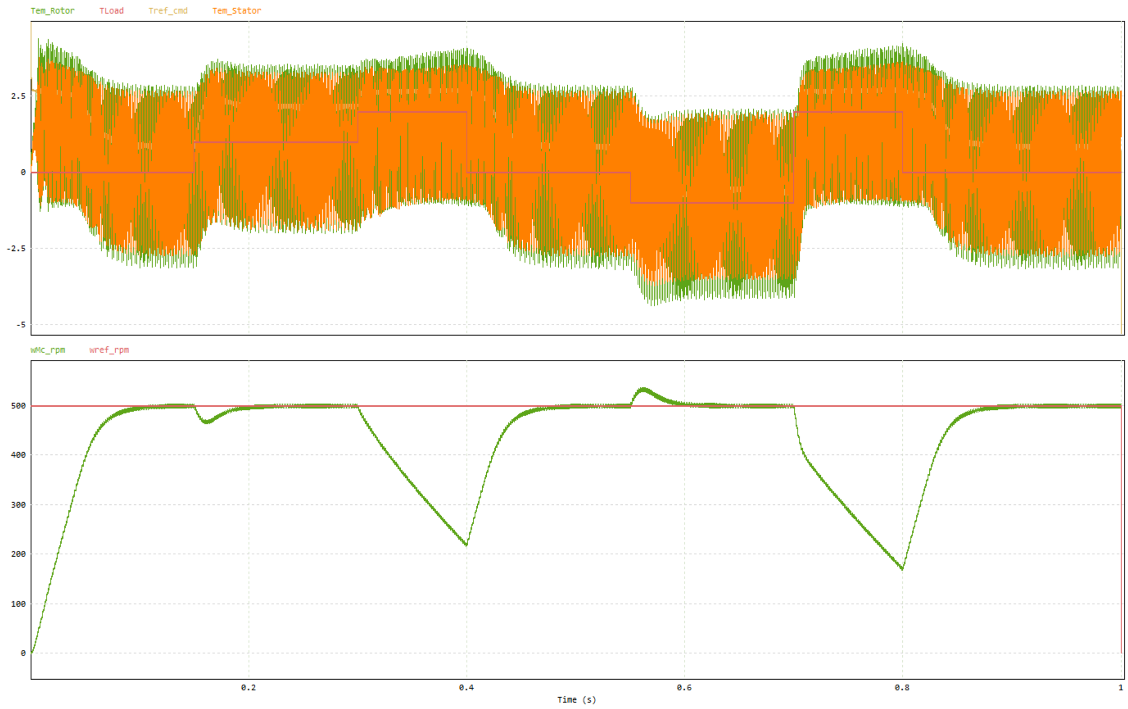


Figure 3.16: FOC load disturbance rejection test.

As shown in Figure 3.16, the controller maintains the rotor speed close to the reference despite the application of successive load disturbances, unless the required lod excedes the motors rated torque. Following each load variation, the electromagnetic torque rapidly adjusts to compensate for the imposed disturbance and restore the mechanical equilibrium of the system.

Although a small increase in torque ripple can be observed under loaded conditions, the ripple remains significantly low. NA small transient speed overshoot of approximately 6–7 percent can be observed following the removal of a negative load torque, after which the controller rapidly restores the commanded speed and it reaches the new steady-state operating condition in approximately 0.03 s for load variations of approximately 1 Nm.

These results confirm the effectiveness of the current control loops, decoupling compensation, and feedforward terms in rejecting external disturbances while maintaining accurate speed regulation.

3.3.5.3 Realistic Robotic Duty Cycle

The final validation test evaluates the controller under a representative robotic operating profile combining multiple speed and load transitions. The applied speed reference is

Table 3.9: Speed reference profile for the robotic duty cycle test.

Time (s)	Reference speed (rpm)
0.0000 – 0.0250	0
0.0251 – 0.4000	300
0.4001 – 0.7500	600
0.7501 – 0.9000	100
0.9001 – 1.0000	0

and the applied load torque profile is

Table 3.10: Load torque profile for the robotic duty cycle test.

Time (s)	Load torque (Nm)
0.0000 – 0.2000	0
0.2001 – 0.5000	1.5
0.5001 – 0.6250	0.7
0.6251 – 0.7500	0
0.7501 – 0.9000	2.0
0.9001 – 1.0000	0

with torque values expressed in Nm.

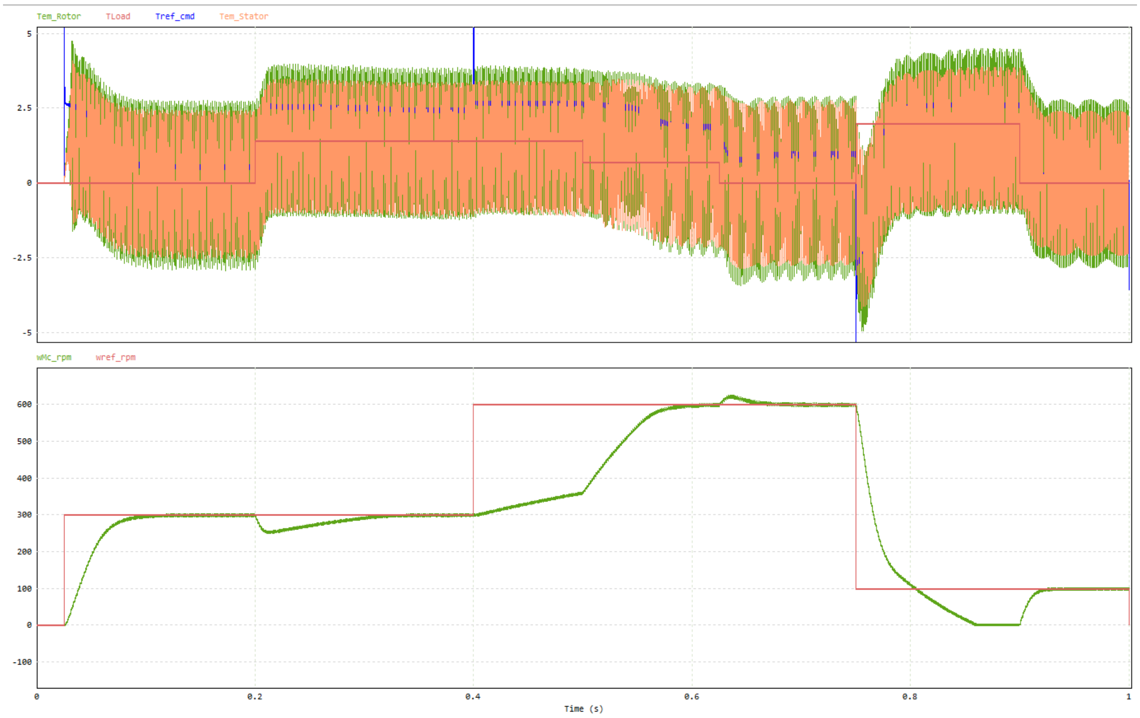


Figure 3.17: FOC performance under a realistic robotic duty cycle.

Figure 3.17 demonstrates the controller performance under combined speed and load variations representative of a robotic actuator.

Under low-load operating conditions, the speed response remains highly accurate with minimal steady-state error and negligible torque ripple. As the load torque increases, the controller requires a larger electromagnetic torque to maintain the desired speed, resulting in slightly slower transient responses. This behaviour is particularly noticeable during operating intervals with higher applied loads, where the acceleration capability of the motor becomes limited by the available torque margin.

Nevertheless, the controller maintains stable operation throughout the complete duty cycle and exhibits good tracking accuracy, smooth torque production, and effective disturbance rejection. These results confirm that the implemented FOC strategy is suitable for dynamic robotic applications requiring precise speed control and low torque ripple over a wide range of operating conditions.

3.4 Comparison of Direct Torque Control and Field-Oriented Control

Both Direct Torque Control (DTC) and Field-Oriented Control (FOC) were successfully implemented and validated in the PSIM environment. The simulation results demonstrate that each control strategy is capable of providing stable operation of the PMSM while achieving satisfactory speed and torque regulation. However, noticeable differences in their dynamic behaviour and steady-state performance can be observed.

The implemented DTC algorithm exhibited a very fast transient response due to its direct selection of inverter switching states based on the instantaneous torque and stator flux errors. This characteristic enables rapid torque production and effective disturbance rejection, making DTC particularly suitable for applications where high dynamic performance is the primary objective. Nevertheless, the hysteresis-based switching strategy inherently introduces increased torque, current, and speed ripple, which becomes evident during steady-state operation.

In contrast, the implemented FOC strategy regulates the stator currents through closed-loop PI controllers operating in the synchronous rotating reference frame. The use of coordinate transformations together with decoupling and feedforward compensation allows independent control of the flux- and torque-producing current components. As a result, the controller provides smooth current regulation, reduced torque ripple, and improved steady-state accuracy while maintaining satisfactory transient performance.

The main quantitative and implementation-related differences observed between the two control strategies are summarized in Table 3.11.

Table 3.11: Comparison between the implemented DTC and FOC strategies.

Criterion	**DTC**	**FOC**
Response time	0.001 s	0.1 s
Overshoot	10%	$< 1\%$
Steady-state ripple	5%	$< 1\%$
Steady-state speed error	2%	0.5%
Speed drop under 1 Nm load disturbance	$\Delta\omega \approx 20$ rpm	$\Delta\omega \approx 40$ rpm
Implementation complexity	≈ 25 PSIM blocks	≈ 55 PSIM blocks
Computational requirements	Clarke transformation	Clarke, Park and inverse Park transformations

The quantitative comparison confirms the trends observed throughout the validation tests. DTC provides the fastest response and exhibits a smaller speed drop following the considered load disturbance, but this performance is accompanied by considerably higher overshoot and steady-state ripple. FOC, on the other hand, achieves substantially lower overshoot, ripple, and steady-state speed error, at the expense of a slower transient response and increased implementation and computational complexity.

Considering the objectives of this work, namely the development of a high-performance PMSM controller with accurate speed regulation, low steady-state ripple, and a modular implementation suitable for subsequent deployment on an ARM-based platform, the advantages offered by FOC outweigh its reduced dynamic response. Although its greater number of functional blocks and coordinate transformations imply higher computational requirements, its improved precision and smoother operation make it the more suitable control strategy for the subsequent stages of this project.

For these reasons, the Field-Oriented Control implementation is selected as the final control architecture and will be used in the remainder of this thesis.

Chapter 4

Advanced Features for Field-Oriented Control

4.1 Chapter Overview

The previous chapter presented the implementation and validation of both Direct Torque Control (DTC) and Field-Oriented Control (FOC) strategies for Permanent Magnet Synchronous Motor (PMSM) drives in the PSIM simulation environment. Following the comparative analysis, FOC was selected as the preferred control approach due to its superior steady-state performance, lower torque and current ripple, and improved precision in speed regulation.

Building upon this controller, the present chapter introduces several additional developments aimed at improving its robustness, adaptability, and practical applicability. These enhancements extend the basic FOC implementation presented previously and explore features commonly required in real-world drive systems.

The chapter begins with the identification of the main PMSM electrical and mechanical parameters. Dedicated simulation procedures are used to estimate the stator resistance, direct- and quadrature-axis inductances, permanent magnet flux linkage, rotor inertia, and viscous friction coefficient. These parameters are essential for accurate controller operation and provide the foundation for subsequent developments.

Next, a Hall sensor-based rotor position and speed estimation method is presented. The implemented approach reconstructs the electrical angle from discrete Hall sensor transitions through interpolation, providing a higher-resolution estimate suitable for FOC operation. The modified control structure is then validated through simulation, demonstrating the feasibility of operating the controller with Hall-effect feedback while also providing the basis for a sensorless operation demonstration.

The chapter further investigates the tuning of the speed control loop by examining the characteristics of the controlled plant and applying automatic PID tuning techniques. The resulting controller parameters are analysed and compared with manually tuned solutions in order to assess their effectiveness and simplify the controller design process.

Finally, an approach for estimating the applied mechanical load torque from the machine's electrical variables is explored. By exploiting the relationship between back electromotive force (Back-EMF), motor speed, and electromagnetic torque, an estimate of the external load is obtained, providing additional information that can be useful for monitoring and advanced control applications.

Overall, this chapter demonstrates how the baseline FOC implementation can be extended with parameter identification, improved state estimation, automatic controller tuning, and load estimation capabilities, resulting in a more complete and versatile motor drive system.

4.2 Parameter Identification

4.2.1 Parameters of Interest

Accurate machine parameters are essential for the correct operation of Field-Oriented Control (FOC) and for the performance of observers and higher-level control loops. Parameters such as stator resistance, inductances, permanent magnet flux linkage, and mechanical inertia directly affect the accuracy of current control, torque production, and speed/position dynamics.

This chapter presents the identification of the main PMSM parameters used in the developed control model. The estimation is performed in two stages. First, an **offline identification** procedure is carried out using dedicated simulation tests and logged signals. These experiments allow the extraction of electrical and mechanical parameters from controlled excitation conditions.

After validating these estimates, the feasibility of an **online adaptive estimation** approach is investigated, where parameters are continuously updated during operation through algorithms implemented in the PSIM C-block.

The parameters considered in this work are those that most strongly affect the behavior of the PMSM drive and the implemented FOC control structure:

- Stator resistance R_s
- Direct and quadrature axis inductances L_d, L_q
- Permanent magnet flux linkage ψ_f
- Mechanical inertia J
- Viscous friction coefficient B

These parameters influence different parts of the control system:

- Current-loop gain and voltage model accuracy
- Decoupling terms in the dq current control
- Torque constant and back-EMF generation

- Speed and position control dynamics

Offline identification is performed through controlled simulation experiments where specific machine variables are excited and the resulting responses are analyzed. This approach allows each parameter to be estimated independently under well-defined conditions.

The following subsections describe the identification procedure for each parameter.

Figure 4.1 presents the overall architecture of the offline parameter identification framework. Measured electrical and mechanical signals are first pre-processed through filtering, differentiation, and validation stages. Based on the detected operating condition, the corresponding parameter-specific estimator is activated. The estimated parameters are then averaged, validated, and stored for subsequent use in the control system.

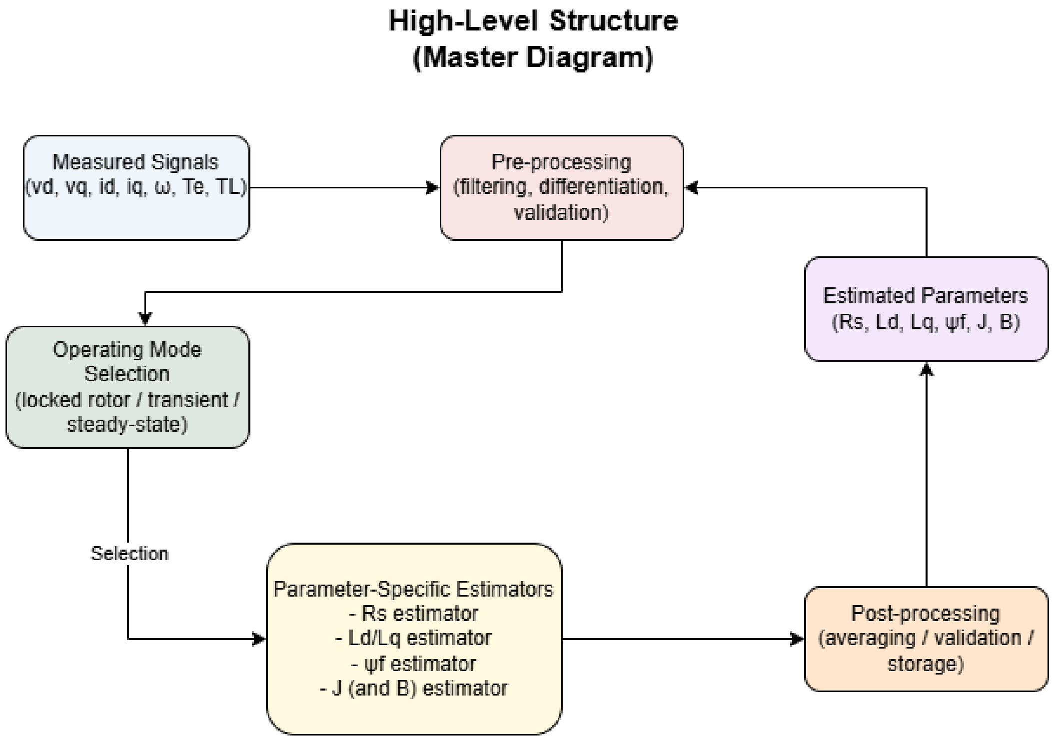


Figure 4.1: High-level structure of the offline PMSM parameter estimation framework.

4.2.2 Stator Resistance Identification (R_s)

Estimating the stator phase resistance R_s, which primarily influences the steady-state relationship between applied voltage and current in the stator windings. Accurate knowledge of R_s is important for:

- Current-loop DC gain
- Voltage model observers
- Low-speed sensorless estimation

4.2.2.1 Test Setup

Using the same build as previouslly for the 3rd iteration of the FOC, as depicted in B.9 I now locked the rotor (in PSIM: setting mechanical speed to 0, forcing $\omega = 0$). With that, I then applied a DC voltage in d-axis so current settles. There is no speed or position loop here. I run a voltage control only for this test, Vd = Vtest, vq=0. Vtest is small enough so that I avoid current limit/saturation.

4.2.2.2 Estimation Method

The stator resistance is estimated from the steady-state relationship between the applied voltage and the resulting stator current.

With the rotor mechanically locked ($\omega = 0$), the PMSM d–axis voltage equation simplifies significantly. The general dq model is

$$v_d = R_s i_d + L_d \frac{di_d}{dt} - \omega L_q i_q \tag{4.1}$$

Since the rotor is locked ($\omega = 0$) and a constant voltage is applied, the current eventually reaches steady state. At this operating point the current derivative is zero ($\frac{di_d}{dt} = 0$) and the cross-coupling term disappears. The equation therefore reduces to

$$v_d = R_s i_d \tag{4.2}$$

From this relation the stator resistance can be directly obtained as

$$R_s = \frac{v_d}{i_d} \tag{4.3}$$

In practice, the steady-state values of v_d and i_d are measured once the transient current response has settled.

The signal-processing structure of the implemented stator resistance estimator is shown in Fig. 4.2. The estimator monitors the current derivative in order to detect steady-state conditions before computing the resistance through the steady-state voltage-current relationship. The resulting estimate is filtered through an averaging stage to improve robustness against numerical noise and transient disturbances.

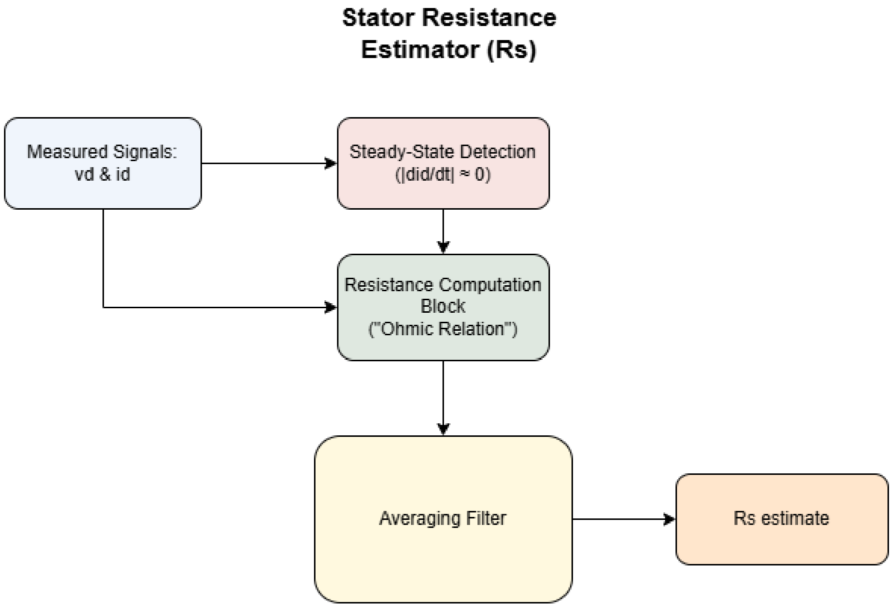


Figure 4.2: Block diagram of the stator resistance estimation algorithm.

To verify the correctness of the estimation method, the experiment is repeated using multiple injected voltages (V_{test}). This ensures that the calculated resistance is independent of the excitation level.

Additionally, the stator resistance parameter inside the PMSM model is intentionally varied and the estimation procedure repeated. This allows verification that the method remains accurate for different resistance values and therefore demonstrates its robustness.

4.2.2.3 Simulation Results

To validate the robustness of the estimation method, four simulations were performed using two different injected voltages (V_{test} = 20 V and 100 V) and two different stator resistance values (R_s = 1.5 Ω and 3 Ω) in the PMSM model.

In all cases the estimated resistance converges to the correct value, demonstrating that the method is independent of the test voltage magnitude and remains accurate for different machine parameters.

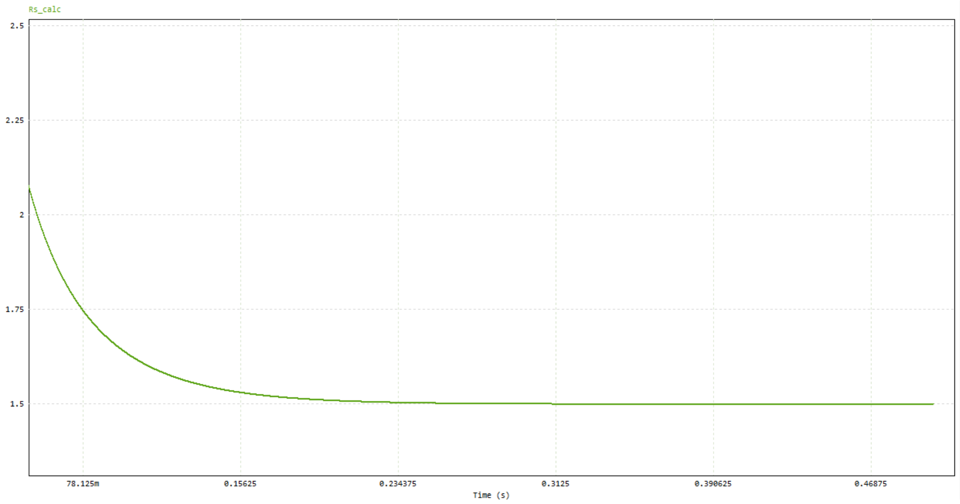


Figure 4.3: Estimated stator resistance for $R_s = 1.5\,\Omega$ using $V_{test} = 20$ V. The estimator converges to the true resistance value after the current reaches steady state.

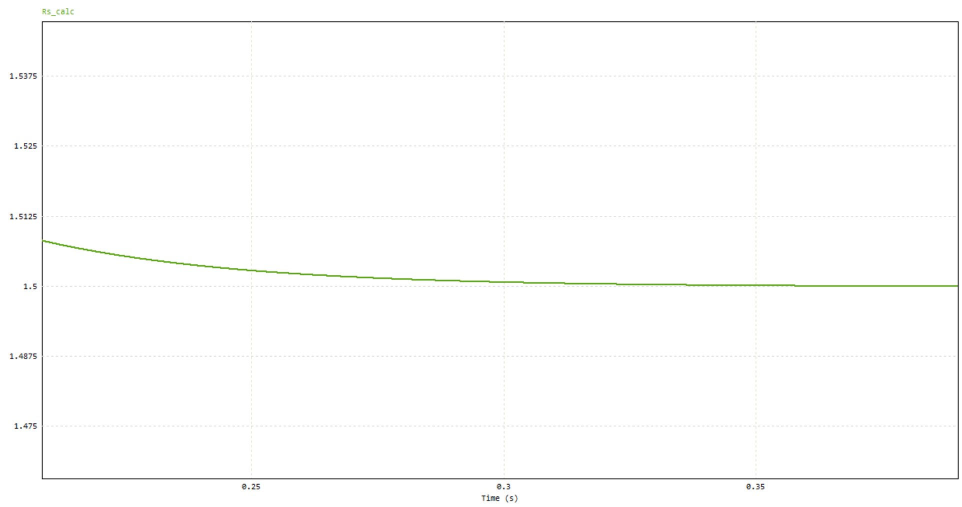


Figure 4.4: Estimated stator resistance for $R_s = 1.5\,\Omega$ using $V_{test} = 100$ V. The higher test voltage results in a larger current but the estimation still converges to the correct resistance value.

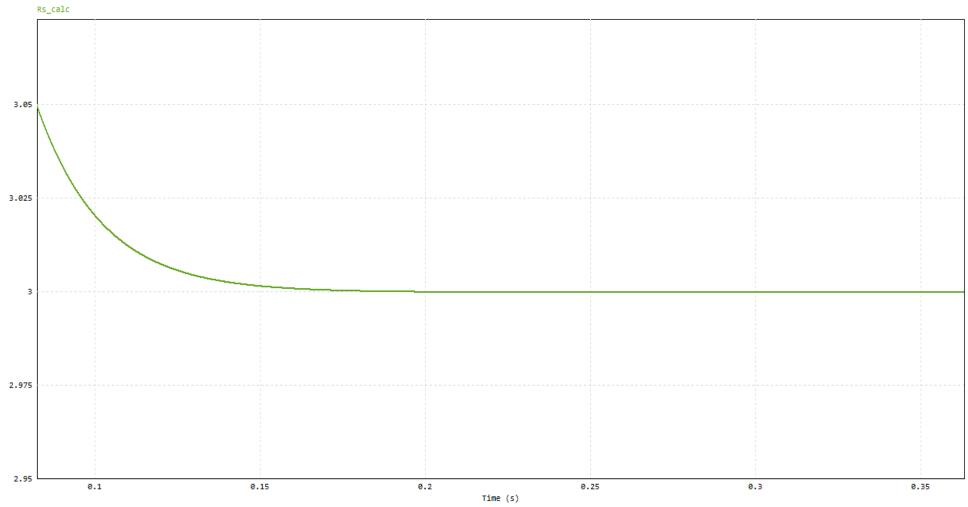


Figure 4.5: Estimated stator resistance for $R_s = 3\,\Omega$ using $V_{test} = 20$ V. The estimator correctly converges to the new resistance value of the machine model.

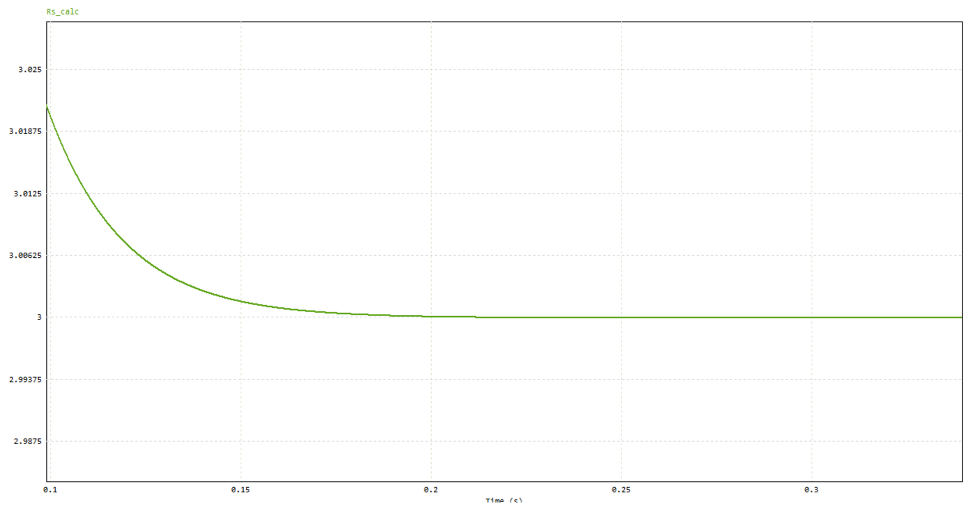


Figure 4.6: Estimated stator resistance for $R_s = 3\,\Omega$ using $V_{test} = 100$ V, confirming robustness of the estimation method with respect to both excitation voltage and parameter variation.

The estimator converges in all four cases to the correct value of R_s, confirming the validity and robustness of the proposed offline identification method.

4.2.3 Inductance Identification (L_d, L_q)

The direct and quadrature axis inductances were identified using a locked-rotor voltage injection method. These parameters are critical for accurate decoupling, current control dynamics, and observer-based estimation.

4.2.3.1 Test Setup

The identification was performed under locked-rotor conditions ($\omega_e = 0$), ensuring that all speed-dependent terms in the PMSM model are eliminated. The schematic used was once again B.9.

A sequence of voltage pulses was applied independently to the d-axis, while maintaining:

- $v_q = 0$
- $i_q \approx 0$

This guarantees that only the d-axis dynamics are excited, allowing the isolation of L_d.

Multiple pulses with increasing amplitudes were applied in order to verify robustness of the estimation across different operating points. The pulse duration was chosen to capture the transient current response without reaching steady-state.

Figure 4.7 illustrates the structure of the implemented inductance estimator. The measured current is numerically differentiated to obtain $\frac{di}{dt}$, while the resistive voltage drop is compensated using the previously identified stator resistance. A validity logic block ensures that estimation is only performed under appropriate operating conditions. The instantaneous inductance estimate is subsequently averaged in order to reduce the influence of numerical differentiation noise.

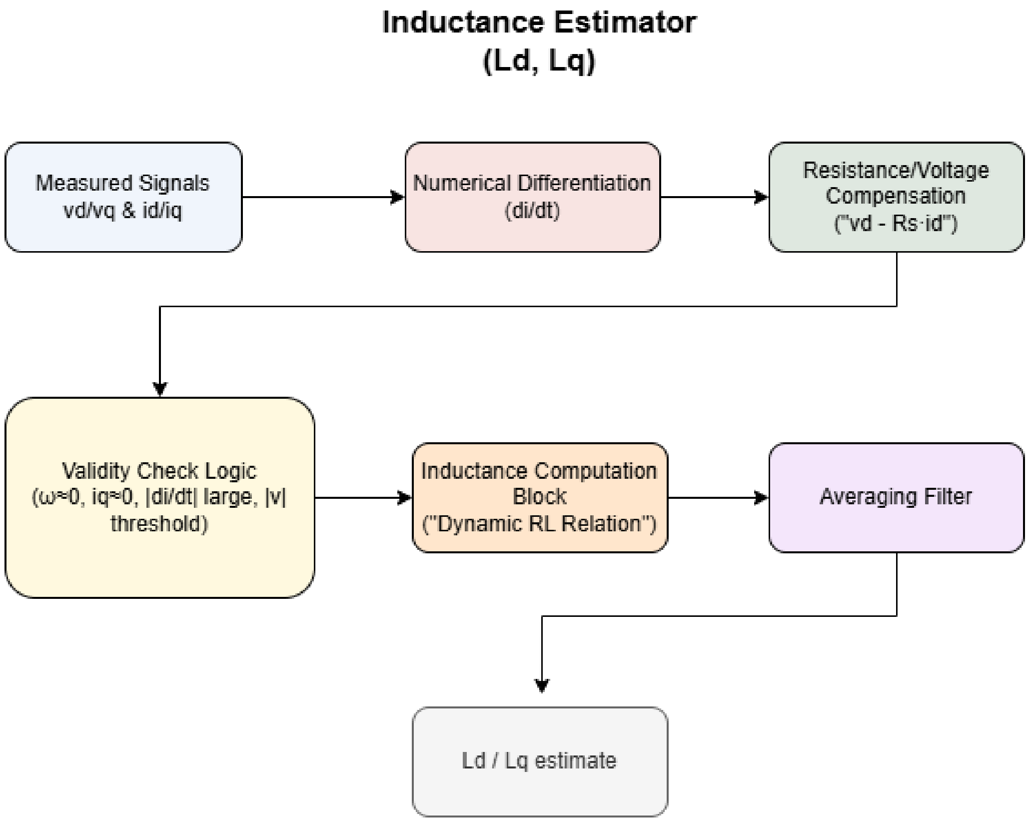


Figure 4.7: Block diagram of the L_d and L_q estimation algorithm.

4.2.3.2 Estimation Method

Under locked-rotor conditions, the PMSM voltage equations simplify to:

$$v_d = R_s i_d + L_d \frac{di_d}{dt} \tag{4.4}$$

$$v_q = R_s i_q + L_q \frac{di_q}{dt} \tag{4.5}$$

Using the previously identified stator resistance R_s, the inductance can be estimated from the transient current response:

$$L_d = \frac{v_d - R_s i_d}{\frac{di_d}{dt}} \tag{4.6}$$

The derivative $\frac{di_d}{dt}$ was computed numerically, and the estimation was only performed during a short interval of the rising edge of each voltage pulse, where the signal-to-noise ratio is highest and resistive effects are still limited.

A validity condition was applied to ensure that:

- $\omega_e \approx 0$
- $i_q \approx 0$
- $|v_d|$ is above a minimum threshold
- $|di_d/dt|$ is sufficiently large

Only samples satisfying these conditions were used to compute and average the inductance.

4.2.3.3 Simulation Results

Figure 4.8 shows the applied d-axis voltage pulses and the corresponding current response. The expected RL behavior is observed: the current increases during the voltage pulse and decays once the excitation is removed, while v_q and i_q remain negligible.

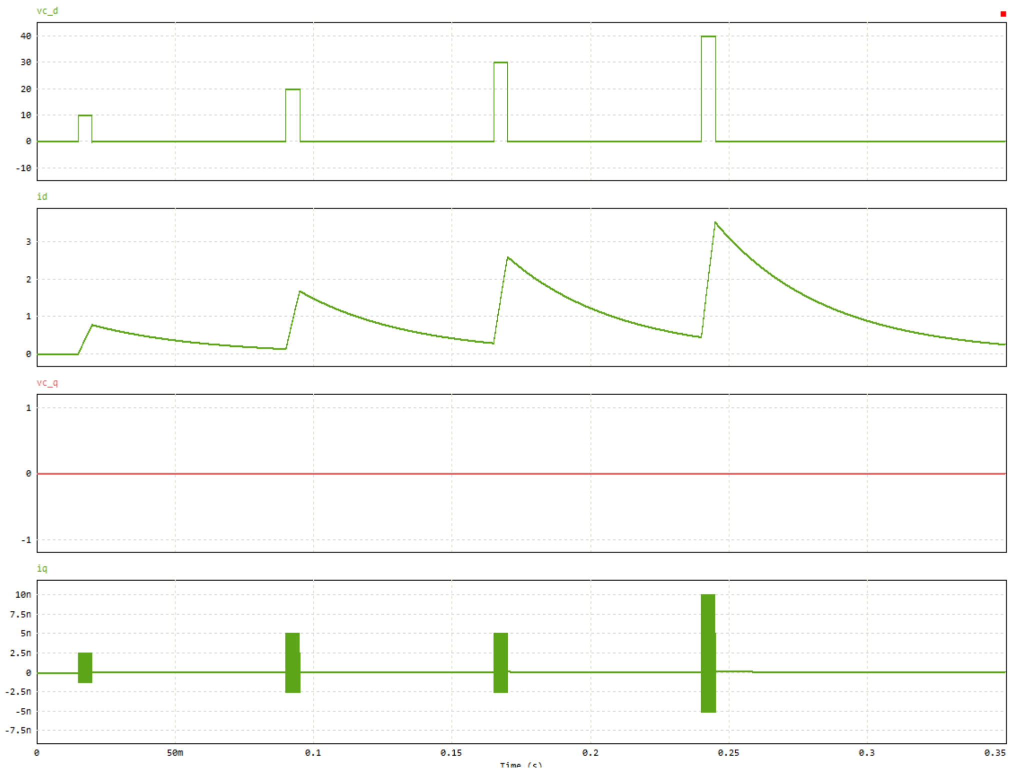


Figure 4.8: d-axis voltage excitation and corresponding current response. The q-axis remains decoupled ($v_q \approx 0$, $i_q \approx 0$).

Figure 4.9 presents the computed current derivative and the validity signal. The estimator is only active during a short interval of the transient response, ensuring that the inductance is calculated from the most informative region of the waveform.

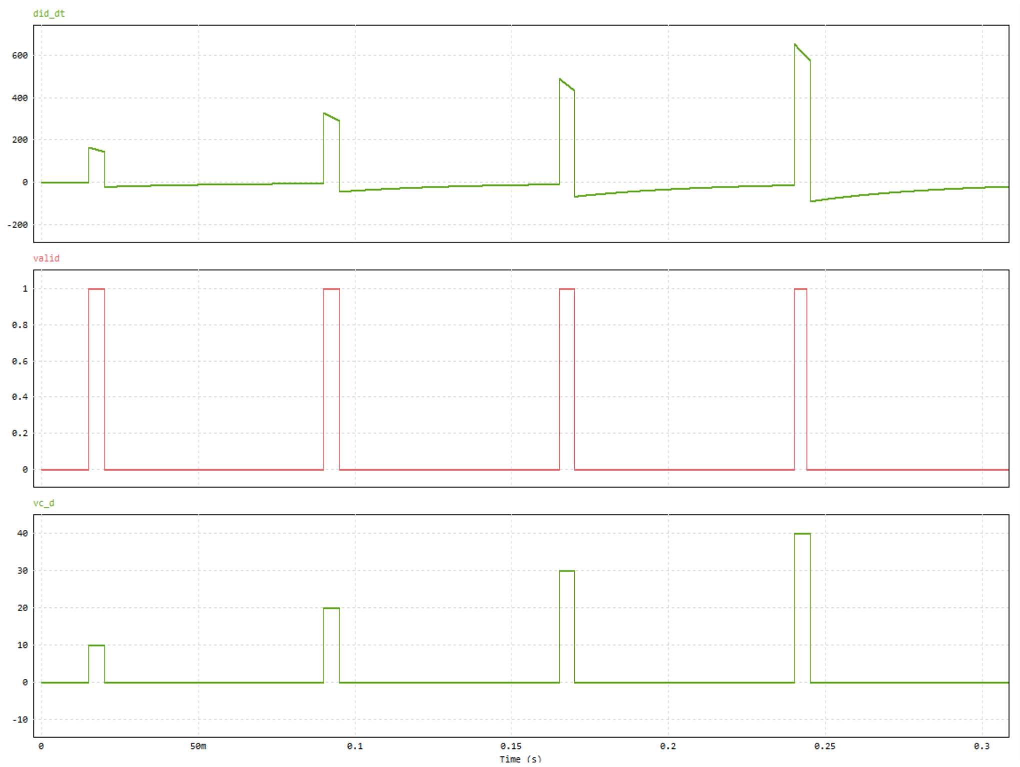


Figure 4.9: Derivative of i_d and validity signal used to gate the estimation during the transient interval.

Figure 4.10 shows both the instantaneous estimate L_d^{raw} and the averaged value L_d^{avg}. While the raw estimate exhibits spikes due to numerical differentiation and switching instants, the averaged value converges to a stable result.

Figure 4.10: Instantaneous and averaged inductance estimation. The averaged value converges to a stable result despite transient spikes.

The estimated values obtained for each voltage pulse were:

$$L_d^{(1)} = 9.01 \times 10^{-2} \text{ H} \tag{4.7}$$
$$L_d^{(2)} = 6.001 \times 10^{-2} \text{ H} \tag{4.8}$$
$$L_d^{(3)} = 6.009 \times 10^{-2} \text{ H} \tag{4.9}$$
$$L_d^{(4)} = 6.011 \times 10^{-2} \text{ H} \tag{4.10}$$

The first pulse shows a larger deviation due to the initial transient and estimator initialization. From the second pulse onwards, the results converge consistently around:

$$L_d \approx 60 \text{ mH} \tag{4.11}$$

This value is in excellent agreement with the nominal model parameter, validating both the experimental setup and the estimation method.

The same procedure is applied for the estimation of L_q, by exciting the q-axis ($v_q \neq 0$, $v_d = 0$) and observing the corresponding i_q response.

To further validate the robustness of the method, the same identification procedure was applied to the q-axis inductance using a modified machine model with L_q intentionally set to a different value ($L_q = 115$ mH). This allows verification that the estimator is capable of correctly identifying different inductance values and is not biased toward a fixed parameter.

The test conditions were identical to the L_d case, with $\omega_e = 0$, $v_d = 0$, voltage pulses applied on the q-axis ($v_q \neq 0$) and $i_d \approx 0$.

Figure 4.11 shows the instantaneous estimate L_q^{raw} and the averaged value L_q^{avg}. As in the L_d case, the raw estimate exhibits spikes due to numerical differentiation, while the averaged value converges to a stable result.

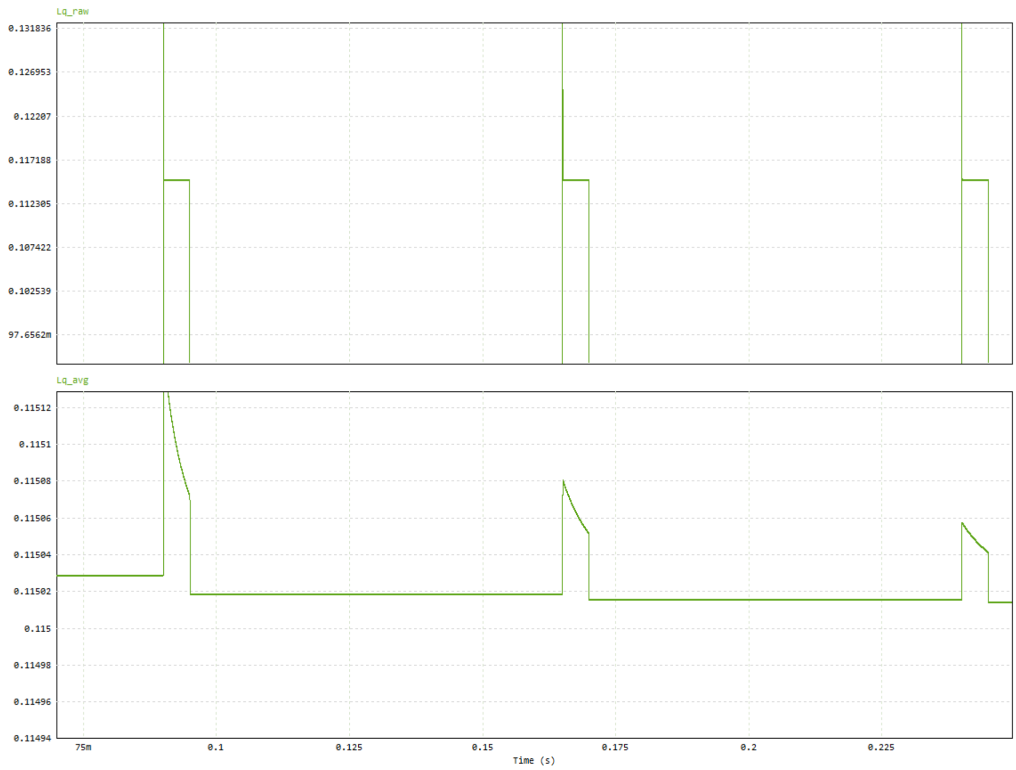


Figure 4.11: q-axis inductance estimation with $L_q = 115$ mH. The averaged estimate converges to the expected value, confirming correct operation of the method.

The estimated value converges to approximately:

$$L_q \approx 115 \text{ mH} \tag{4.12}$$

which is in excellent agreement with the imposed model parameter. This confirms that the estimation method is not only accurate, but also robust to different inductance values and independently applicable to both d- and q-axes.

Overall, the results demonstrate that the proposed offline identification approach reliably estimates both L_d and L_q, validating the method for use in subsequent control and observer design.

4.2.4 Permanent Magnet Flux Linkage Identification (ψ_f)

The permanent magnet flux linkage ψ_f defines the proportionality between electrical speed and the induced back-EMF, and is therefore a key parameter for torque production and observer-based control.

4.2.4.1 Test Setup

Using the same setup as in B.1. The estimation is performed under open-circuit conditions, where the stator terminals are disconnected and no current flows in the machine ($i_d \approx 0$, $i_q \approx 0$).

A constant mechanical speed is imposed using an external speed source. Three operating points are evaluated (300 rpm, 600 rpm, and 900 rpm) to verify consistency of the estimation.

Under these conditions, resistive and inductive voltage drops vanish, and the measured stator voltage corresponds directly to the back-EMF.

4.2.4.2 Estimation Method

From the steady-state q-axis voltage equation of the PMSM:

$$v_q = \omega_e \psi_f \tag{4.13}$$

the permanent magnet flux linkage can be directly obtained as:

$$\psi_f = \frac{v_q}{\omega_e} \tag{4.14}$$

This relation is valid under no-load conditions, where the stator current is negligible and the voltage is dominated by the back-EMF.

The implemented permanent magnet flux linkage estimator is shown in Fig. 4.12. The estimator operates under open-circuit conditions and computes the flux linkage from the ratio between the measured back-EMF voltage and electrical speed. An averaging stage is included to improve estimation stability across different operating points.

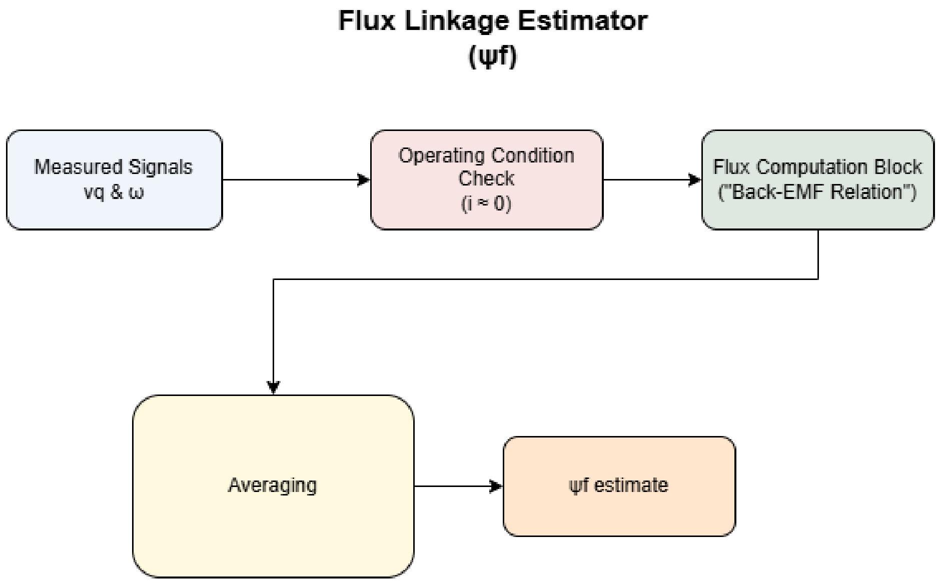


Figure 4.12: Block diagram of the permanent magnet flux linkage estimator.

4.2.4.3 Simulation Results

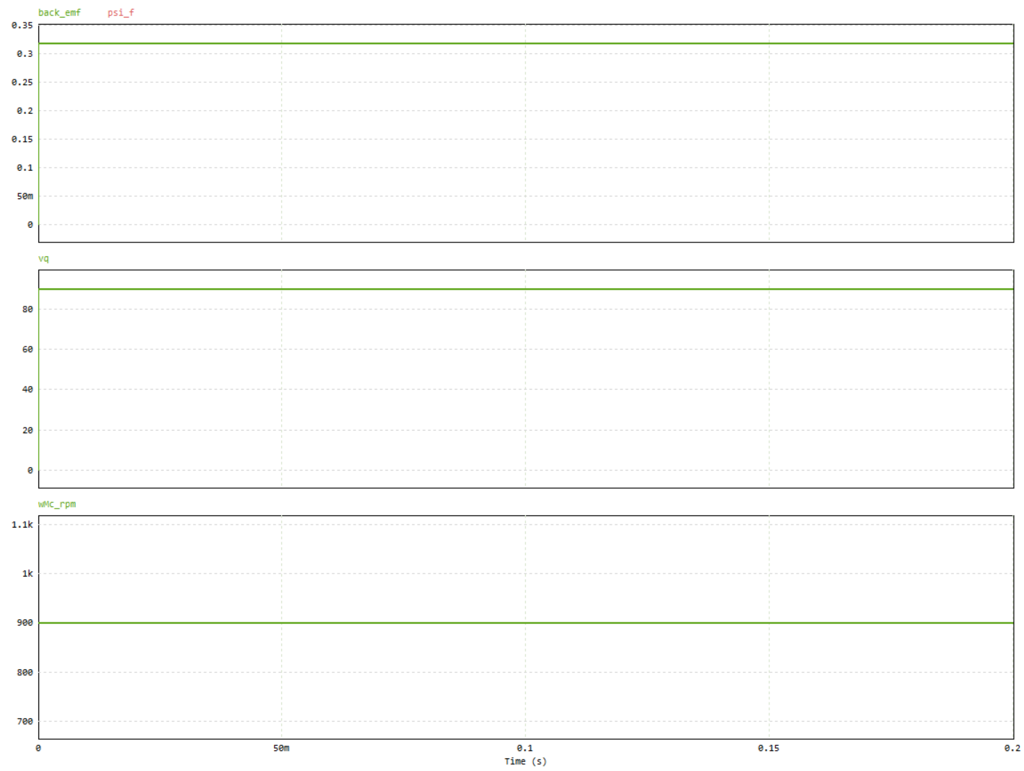


Figure 4.13: Back-EMF estimation at 300 rpm, 600 rpm, and 900 rpm. The q-axis voltage scales linearly with speed, while the estimated flux linkage ψ_f remains constant.

The results show that while v_q increases proportionally with speed, the estimated flux linkage ψ_f remains approximately constant across all operating points, confirming the validity of the method.

4.2.5 Mechanical Parameter Identification (J, B)

The mechanical parameters of the drive system, namely the inertia J and the viscous friction coefficient B, determine the dynamic behavior of the rotor and directly influence the design of higher-level control loops such as speed and position control.

4.2.5.1 Test Setup

The same simulation setup as described in Fig. B.19 is used. The machine is operated under current control, with a constant q-axis current reference i_q, resulting in a constant electromagnetic torque T_e. The d-axis current is set to zero ($i_d = 0$).

No speed control loop is employed, and the rotor is free to accelerate under the applied torque. A constant load torque T_L can be applied at the shaft. The mechanical speed ω is measured and used for parameter estimation.

4.2.5.2 Estimation Method

The mechanical dynamics of the rotor are described by:

$$J\frac{d\omega}{dt} = T_e - T_L - B\omega \tag{4.15}$$

Inertia Identification For the initial transient, the rotor speed is low and the friction term $B\omega$ can be neglected. The equation simplifies to:

$$J\frac{d\omega}{dt} \approx T_e - T_L \tag{4.16}$$

The electromagnetic torque is obtained from the q-axis current:

$$T_e = \frac{3}{2} p \psi_f i_q \tag{4.17}$$

Using the measured speed signal, the acceleration $\frac{d\omega}{dt}$ is computed numerically. The inertia can then be estimated as:

$$J = \frac{T_e - T_L}{\frac{d\omega}{dt}} \tag{4.18}$$

This method relies on the initial acceleration phase, where the influence of viscous friction is negligible and the torque can be considered constant.

Viscous Friction Identification The estimation of the viscous friction coefficient B requires a steady-state operating condition, where:

$$\frac{d\omega}{dt} = 0 \quad \Rightarrow \quad T_e = T_L + B\omega \tag{4.19}$$

In this case, B could be obtained as:

$$B = \frac{T_e - T_L}{\omega} \tag{4.20}$$

However, in the present setup, such a steady-state condition cannot be reliably achieved. The system operates under current control without speed regulation, and the rotor accelerates continuously under constant torque. As the speed increases, the back-EMF term $\omega_e \psi_f$ grows and eventually approaches the available DC bus voltage. This leads to voltage saturation in the inverter, causing the current controller to lose authority and resulting in oscillatory and unstable behavior.

Consequently, no valid equilibrium region exists in which the condition $\frac{d\omega}{dt} = 0$ is satisfied while maintaining stable operation. Therefore, a reliable estimation of the viscous friction coefficient B cannot be performed using this offline approach.

For this reason, the estimation of B is deferred to the online identification stage, where a closed-loop speed controller will enforce a stable operating point.

Figure 4.14 presents the structure of the implemented mechanical parameter estimator. The electromagnetic torque is computed from the measured q-axis current, while the rotor acceleration is obtained through numerical differentiation of the measured speed signal. Depending on the operating region, the estimator evaluates the mechanical dynamic equation to estimate the rotor inertia. The estimated parameter is subsequently filtered through an averaging stage in order to improve convergence stability.

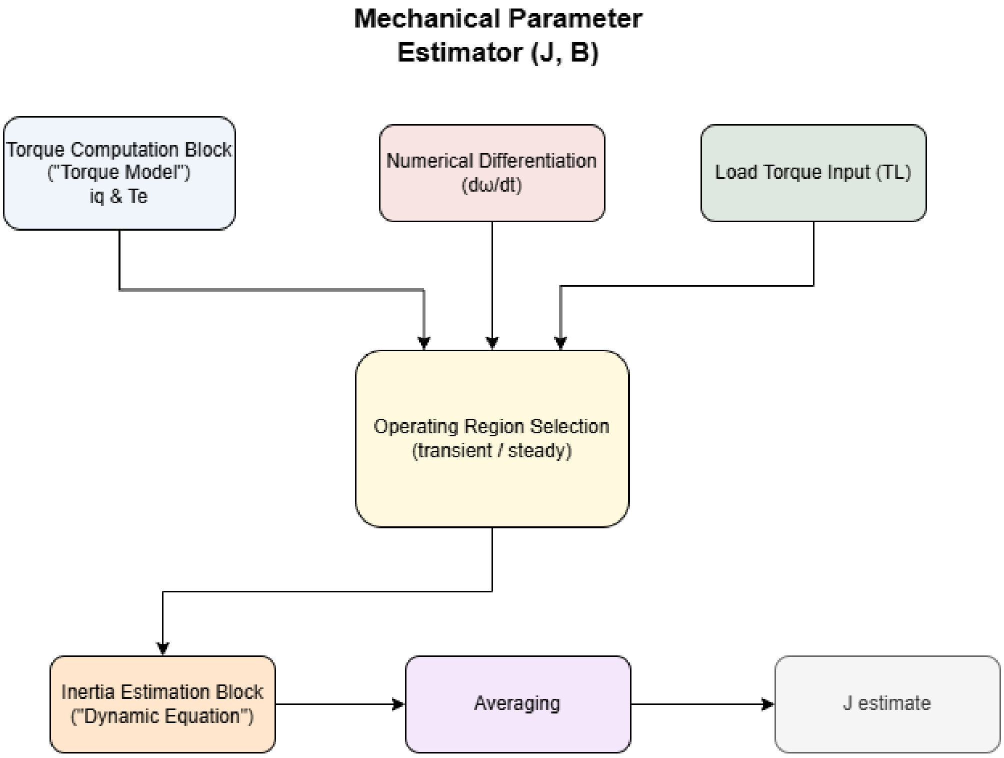
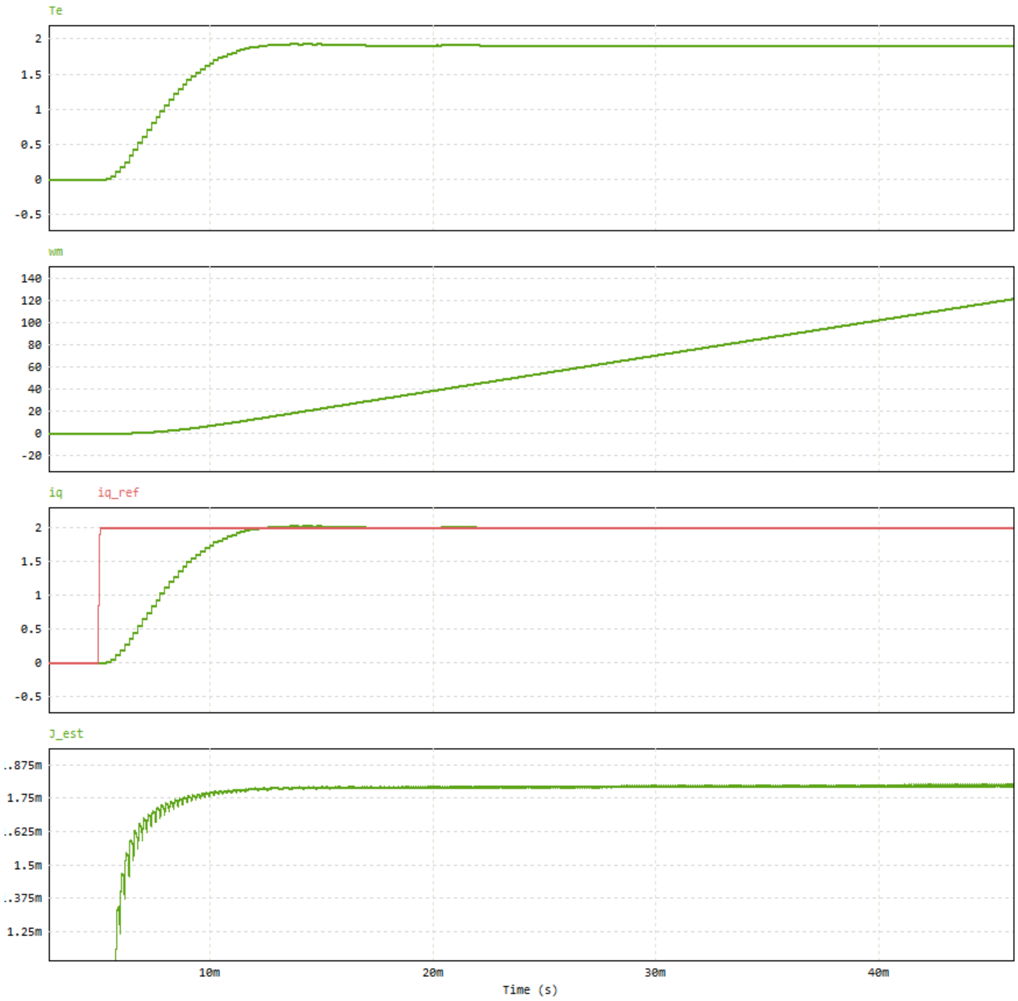
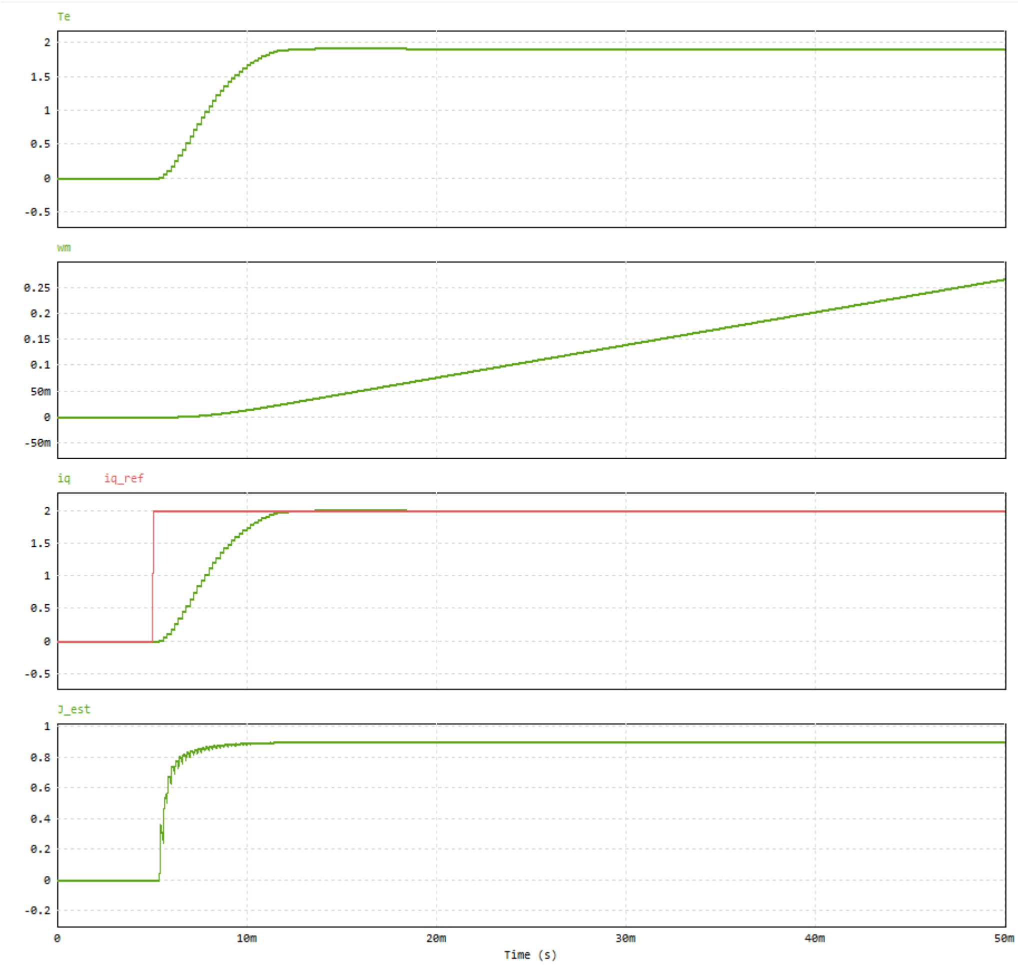


Figure 4.14: Block diagram of the mechanical parameter estimation algorithm.

4.2.5.3 Simulation Results





Te
wm
iq
iq_ref
J_est
Time (s)

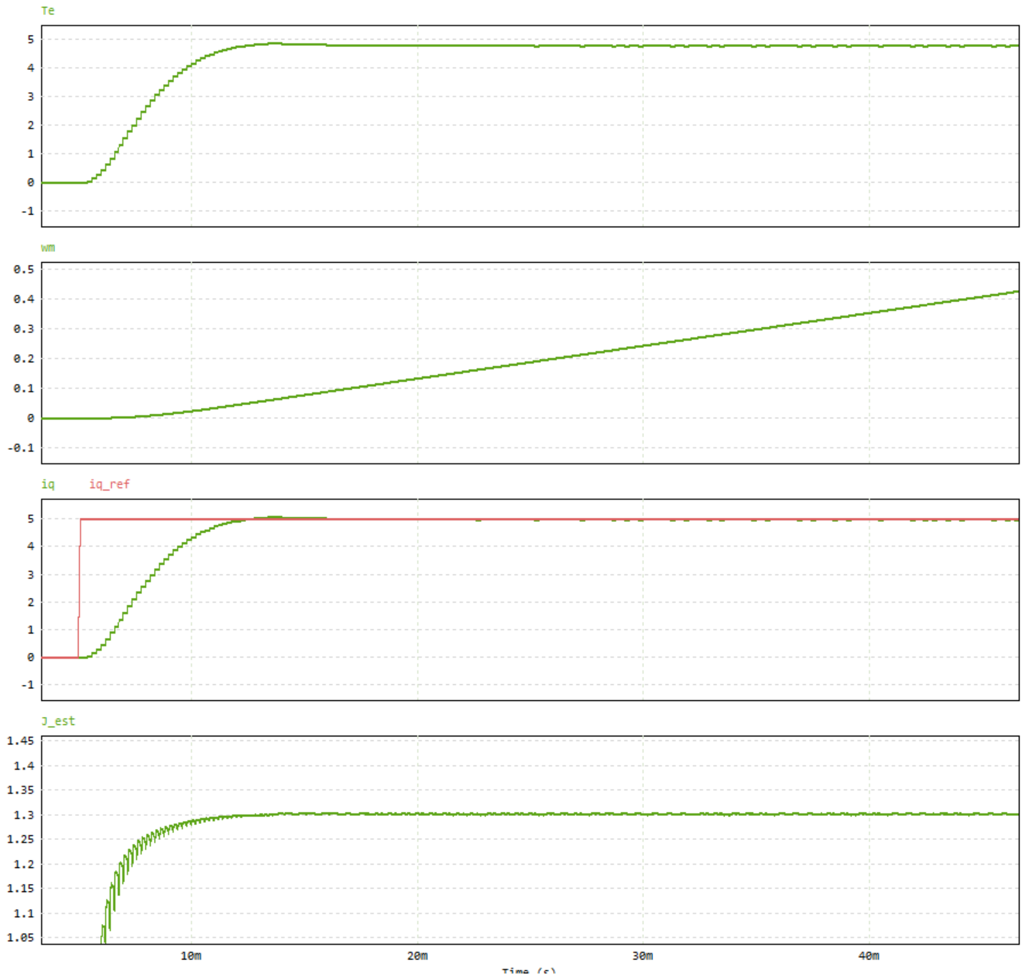


Figure 4.15: Inertia estimation results for different operating conditions. From left to right: low inertia ($J = 0.00179$), medium inertia ($J = 0.90179$), and high inertia ($J = 1.3$) with increased torque excitation.

The results demonstrate that the estimator converges rapidly to the actual inertia value in all tested scenarios. For low inertia values, the convergence is faster due to the higher acceleration, while for larger inertia values the convergence is slower but remains stable and accurate.

Additionally, increasing the torque excitation (higher i_q) improves the signal-to-noise ratio of the acceleration, resulting in smoother convergence of the estimated inertia.

Overall, the estimator shows consistent performance across a wide range of operating conditions, validating the proposed identification method.

$$J = \text{(identified value, e.g. 0.00179)} \tag{4.21}$$

$$B = \text{not identified (no valid steady-state condition)} \tag{4.22}$$

4.3 Hall Sensor Angle/Speed Estimation Demonstration

4.3.1 Objective and Model Changes

Until this stage, the FOC implementation relied on an ideal electrical angle obtained directly from the simulated rotor position. Although this approach is suitable for validating the controller during development, it assumes perfect knowledge of the rotor position and therefore does not represent the conditions encountered in a practical drive system.

The objective of this iteration is to replace the ideal angle source with an estimation algorithm based on the three digital Hall sensors available in the PMSM model. Since Hall sensors provide only six discrete position states per electrical revolution, additional processing is required to reconstruct a continuous electrical angle suitable for FOC operation. The proposed estimation chain combines Hall state decoding, sector identification, speed estimation from the time elapsed between Hall transitions, and angle interpolation within each 60° electrical sector.

To support this implementation, the simulation model was modified by enabling the Hall sensor outputs of the PMSM block and connecting the three Hall signals to a dedicated angle estimation subsystem. The estimated electrical angle generated by this subsystem replaces the previously used ideal rotor position as the feedback signal supplied to the FOC controller.

The resulting estimator represents an intermediate solution between ideal position feedback and fully sensorless operation, providing a realistic and computationally inexpensive position estimation method commonly employed in industrial PMSM drives.

4.3.2 PSIM Schematic Snapshot

Figure 4.16 illustrates the modifications introduced to the PMSM drive model. Compared to the previous implementation, the Hall sensor outputs provided by the PMSM are now routed to a dedicated estimation subsystem while the remainder of the electrical drive remains unchanged. The Hall signals are generated internally by the machine model according to the rotor position and are available at the mechanical interface of the motor, immediately before the applied mechanical load.

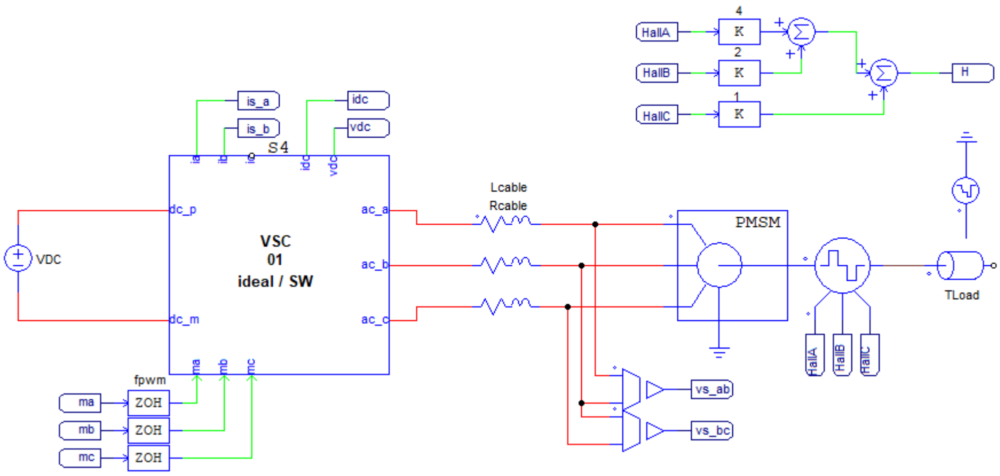


Figure 4.16: Integration of the Hall sensor outputs into the PMSM drive model.

The Hall-based estimator is composed of several functional stages that progressively convert the three digital Hall signals into a continuous estimate of the electrical rotor angle. For clarity, Figure 4.17 presents the algorithm as a simplified functional block diagram, illustrating the flow of information between the main processing stages before introducing the detailed PSIM implementation.

The Hall signals are first decoded into one of the six electrical sectors. The decoded sector is then used in two parallel processing paths. The first determines the corresponding sector base angle, while the second detects Hall transitions and measures the elapsed time between consecutive edges. This information is used to estimate the electrical speed and compute an interpolated angle increment within the current sector. Finally, the interpolated angle is combined with the sector base angle to reconstruct a continuous estimate of the electrical rotor position.

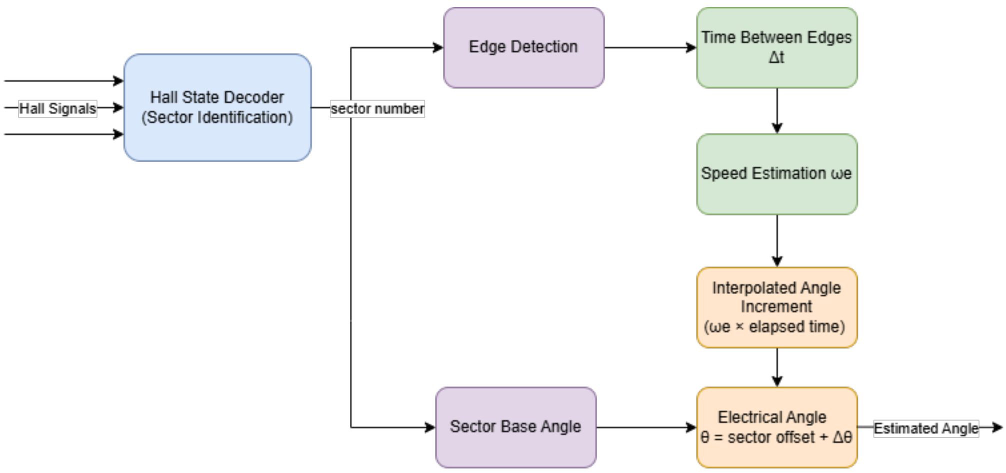


Figure 4.17: Functional block diagram of the Hall-based electrical angle estimation algorithm.

The complete implementation of the estimator in PSIM is shown in Figure 4.18. Although the functional behaviour follows the processing stages presented in Figure 4.17, the implementation consists of numerous interconnected functional blocks, making the complete schematic difficult to interpret at first glance.

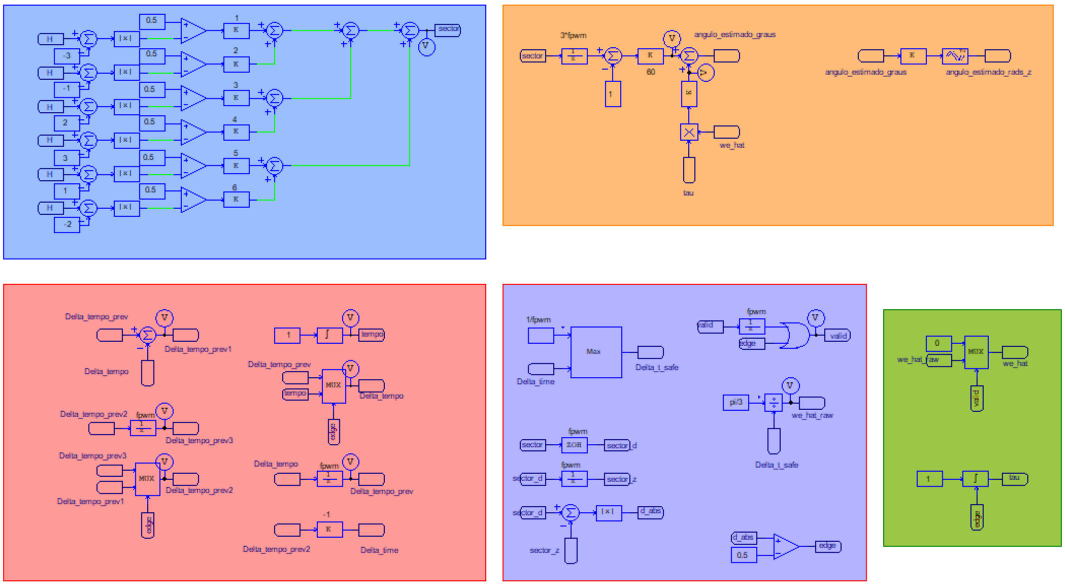


Figure 4.18: Complete PSIM implementation of the Hall-based electrical angle estimation algorithm.

4.3.3 Validation Tests

The Hall-based angle estimator was validated progressively, beginning with the verification of the raw Hall signals, followed by the reconstruction of the electrical angle and electrical speed, and finally by integrating the estimated angle into the closed-loop FOC controller. This layered approach allows each stage of the estimation algorithm to be verified independently before evaluating the overall control performance.

4.3.3.1 Electrical Angle Reconstruction

Since Hall sensors provide position information only every 60° electrical, a continuous electrical angle must be reconstructed before it can be used by the FOC algorithm. This is achieved through linear interpolation between consecutive Hall transitions according to

$$\hat{\theta}_e(t) = \theta_{\text{sector}} + \hat{\omega}_e\left(t - t_{\text{edge}}\right) \tag{4.23}$$

where θ_{sector} is the base angle associated with the current Hall sector and $\hat{\omega}_e$ is the estimated electrical speed.

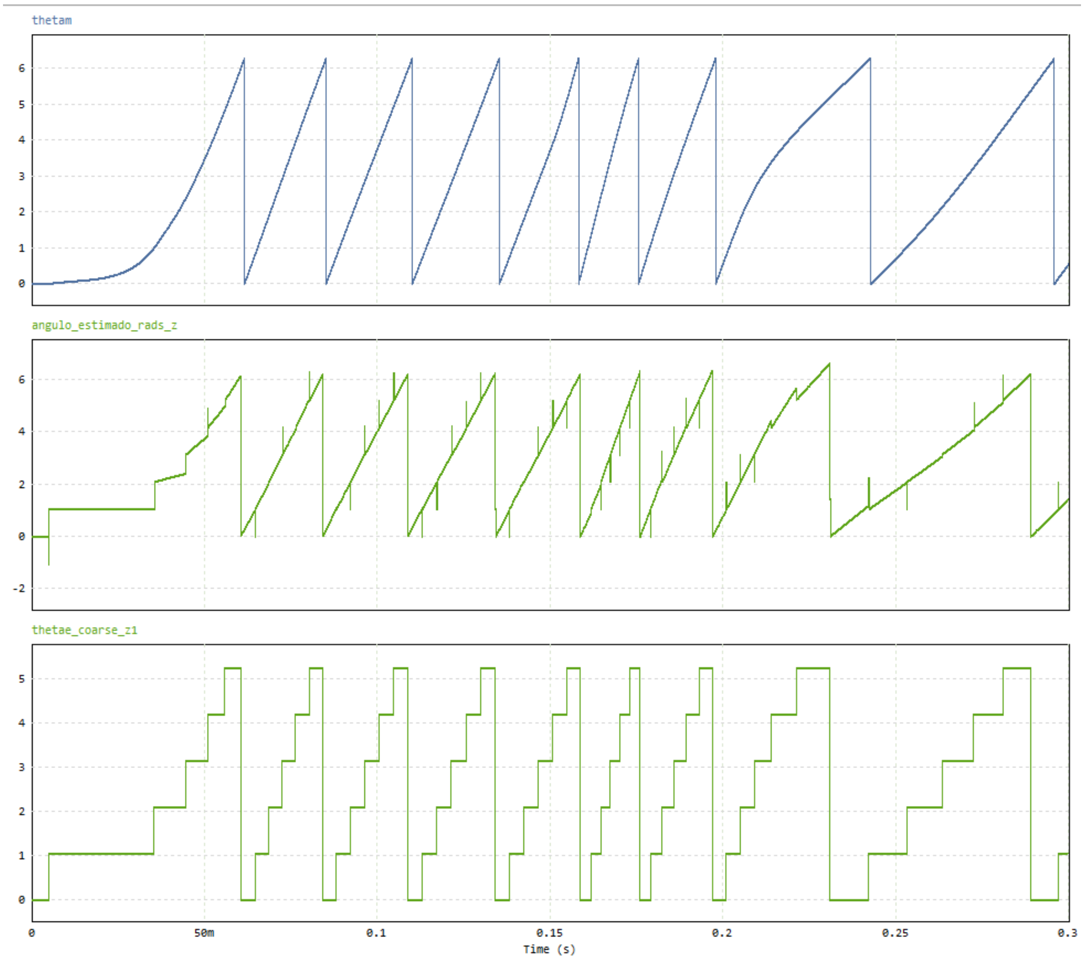


Figure 4.19: Coarse Hall angle, interpolated electrical angle and ideal electrical angle.

The lowest trace in Figure 4.19 illustrates the coarse Hall position, which changes in discrete 60° electrical increments. The interpolated angle reconstructs a continuous estimate between transitions and closely follows the ideal electrical angle shown in the upper trace. Small discontinuities appear at some Hall transitions due to the discrete update of the speed estimate; however, these remain limited and do not significantly affect the reconstructed rotor position.

4.3.3.2 Electrical Speed Estimation

The electrical speed is estimated from the elapsed time between consecutive Hall transitions according to

$$\hat{\omega}_e = \frac{\Delta\theta_{\text{sector}}}{\Delta t} \tag{4.24}$$

where

$$\Delta\theta_{\text{sector}} = \frac{\pi}{3}$$

corresponds to one electrical Hall sector.

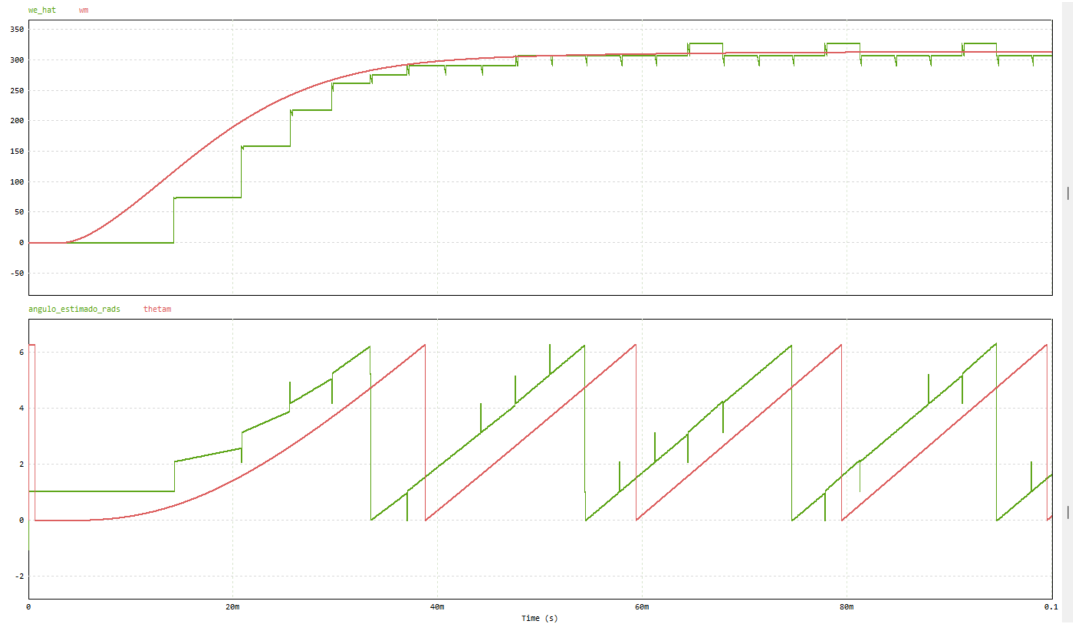


Figure 4.20: Estimated electrical speed obtained from Hall transitions compared with the actual motor speed.

As shown in Figure 4.20, the estimated speed follows the actual motor speed while exhibiting the characteristic staircase behaviour of edge-based estimation. The estimate is updated only when a Hall transition occurs and remains constant between consecutive edges. Consequently, the estimation accuracy improves with increasing rotational speed as Hall transitions become more frequent.

4.3.3.3 Closed-Loop FOC Validation Using the Hall-Based Angle

After validating the individual estimation stages, the ideal electrical angle previously used by the FOC controller was replaced with the Hall-based estimated angle. This represents the final validation of the estimator, as the controller must now rely exclusively on the reconstructed rotor position for the Park and inverse Park transformations.

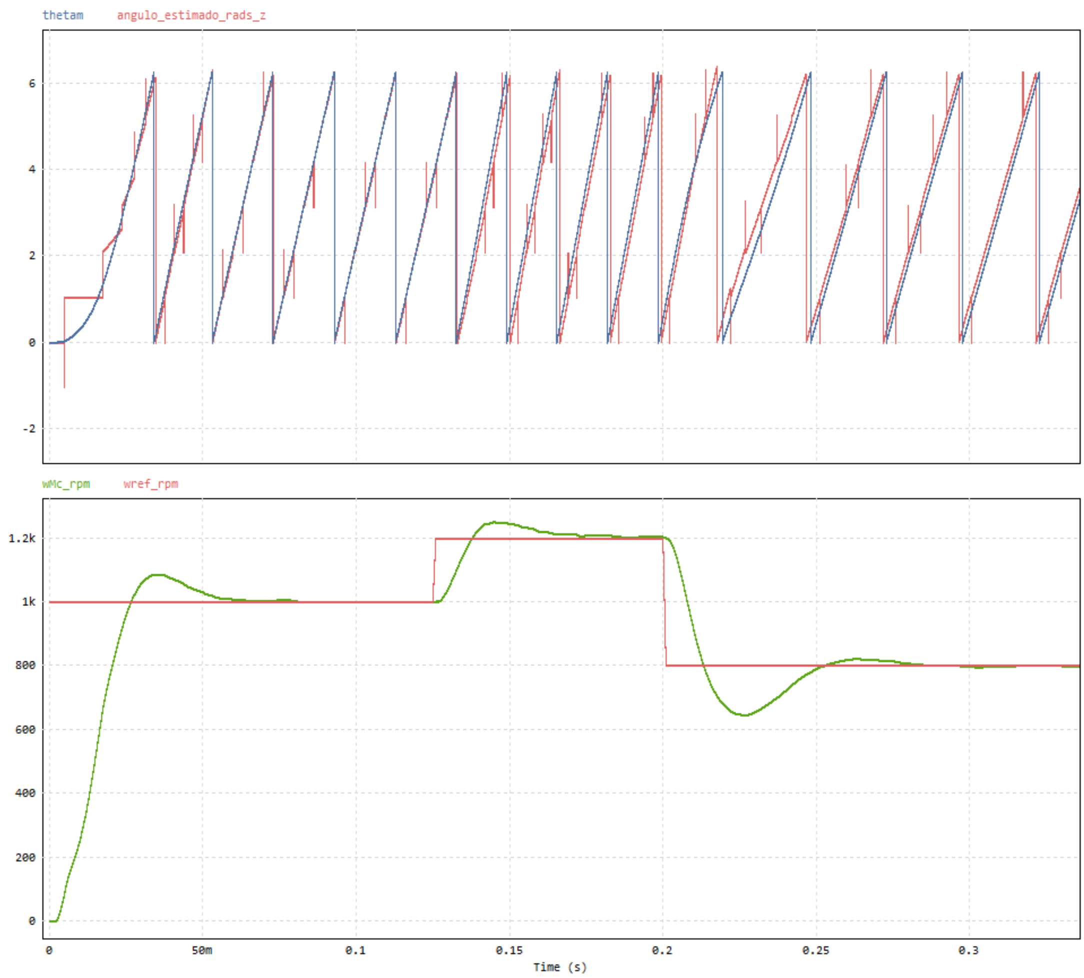


Figure 4.21: Closed-loop speed response using the Hall-based electrical angle estimation.

Figure 4.21 compares the estimated electrical angle with the ideal angle during several speed reference changes while simultaneously showing the corresponding mechanical speed response. Despite the small discontinuities introduced by the Hall transitions, the estimated angle remains well aligned with the actual electrical angle throughout the operating range.

The lower plot demonstrates that the FOC controller maintains stable operation using the estimated Hall angle as its only position feedback. The motor accurately tracks both acceleration and deceleration commands with dynamic performance comparable to that obtained using the ideal rotor position. These results confirm that the proposed Hall-based estimator provides sufficiently accurate rotor position information for closed-loop FOC operation.

4.4 Load Torque Estimation

One of the main limitations of the previously presented Field-Oriented Control (FOC) implementation is its reliance on direct measurements of the motor states without explicitly estimating external disturbances acting on the system. In practical applications, the mechanical load applied to the motor can vary over time, affecting the required torque production and consequently the dynamic response of the controller.

To address this limitation, a load torque estimation algorithm is proposed as an additional feature of the implemented control architecture. The objective is to estimate the external load torque in real time using quantities that are already available within the FOC framework, without requiring additional sensors.

For a surface-mounted Permanent Magnet Synchronous Motor (PMSM), the produced electromagnetic torque can be approximated by

$$T_e = \frac{3}{2} p \lambda_m i_q, \tag{4.25}$$

where p denotes the number of pole pairs, λ_m is the permanent magnet flux linkage, and i_q represents the quadrature-axis current.

The mechanical dynamics of the motor are governed by

$$J \frac{d\omega_m}{dt} = T_e - T_L - B\omega_m, \tag{4.26}$$

where J is the rotor inertia, B is the viscous friction coefficient, ω_m is the mechanical speed, and T_L represents the external load torque.

Rearranging the previous expression yields an estimate of the load torque:

$$\hat{T}_L = T_e - J \frac{d\omega_m}{dt} - B\omega_m. \tag{4.27}$$

The proposed implementation therefore combines the estimated electromagnetic torque with the measured rotor speed and its derivative to obtain an online estimate of the external mechanical load.

Since numerical differentiation is highly sensitive to measurement noise, the speed derivative should be filtered using a suitable low-pass filter or differentiator with noise attenuation before being employed in the estimation algorithm.

The estimated load torque could subsequently be used for disturbance compensation, condition monitoring, or adaptive control strategies. Although the complete implementation and experimental validation are outside the scope of the current work, this approach represents a natural extension of the developed FOC controller and demonstrates the potential for integrating observer-based functionalities into the existing control architecture.

4.5 Automatic PI Controller Tuning

The tuning of the proportional-integral (PI) controllers used in the current and speed loops significantly influences the overall performance of the Field-Oriented Control strategy. In the present implementation, the controller gains were manually adjusted based on simulation results and iterative refinement. Although this procedure yields satisfactory performance, it requires expert knowledge and considerable engineering effort.

To simplify the commissioning process, an automatic tuning procedure is proposed as an advanced feature of the controller.

The general objective of the auto-tuning algorithm is to identify the dynamic characteristics of the controlled system and calculate appropriate controller gains without manual intervention. Among the various available techniques, a step-response based identification method represents a practical solution due to its relatively low implementation complexity and compatibility with simulation environments.

The proposed procedure consists of the following steps:

1. Apply a predefined reference step to the controlled variable.
2. Record the resulting system response.
3. Estimate the dominant dynamic characteristics of the plant, such as rise time and settling time.
4. Calculate appropriate proportional and integral gains according to predefined tuning rules.
5. Update the PI controller parameters automatically.

The resulting control gains can then be directly integrated into the existing speed control loop shown previously in the FOC architecture.

Alternative approaches based on relay feedback experiments or optimization algorithms could also be considered. However, a step-response identification method offers a favourable compromise between implementation effort and achievable performance while remaining sufficiently robust for practical applications.

Although this automatic tuning procedure has not yet been fully implemented within the scope of this work, its integration represents a promising extension of the developed controller. Such functionality would reduce commissioning time, improve controller portability across different motor parameters, and decrease the dependence on manual parameter adjustment.

4.6 Advanced Sensorless Estimation Techniques

The Field-Oriented Control implementation presented in this work relies on Hall-effect sensors to provide the rotor position required for the Park and inverse Park transformations. Although this approach offers a simple and reliable solution, the use of mechanical position sensors increases the overall system cost, wiring complexity, and susceptibility to harsh operating environments. Consequently, considerable research has focused on the development of sensorless control techniques capable of estimating the rotor position and speed using only electrical measurements.

Sensorless control algorithms estimate the rotor position from the measured stator voltages and currents together with a mathematical model of the Permanent Magnet Synchronous Motor (PMSM). Eliminating mechanical sensors improves system reliability while reducing hardware requirements, making these techniques particularly attractive for automotive, aerospace, and industrial drive applications.

Conventional back-EMF estimation techniques perform well at medium and high rotational speeds, where the induced electromotive force is sufficiently large to provide accurate position information. However, their performance deteriorates as the motor speed approaches zero because the back-EMF amplitude becomes very small. As a result, more advanced observer-based estimation methods have been developed to achieve reliable operation over the entire speed range, including low-speed operation and standstill.

One widely adopted approach is the implementation of state observers, such as the Luenberger Observer or Sliding Mode Observer (SMO). These methods estimate the internal motor states by continuously comparing the measured electrical variables with those predicted by the mathematical motor model. The estimation error is then fed back into the observer to progressively correct the estimated rotor position and speed.

Another popular solution is the Model Reference Adaptive System (MRAS). In this approach, two mathematical models of the motor are evaluated simultaneously: a reference model based on measured quantities and an adjustable model containing the estimated rotor position. An adaptive mechanism continuously minimizes the error between both models, allowing the rotor position and speed to converge toward their true values.

For applications requiring higher estimation accuracy and increased robustness against parameter uncertainties and measurement noise, Extended Kalman Filters (EKFs) provide a more sophisticated solution. The EKF formulates the sensorless estimation problem as a recursive state estimation process, combining the nonlinear PMSM model with measured electrical signals to produce optimal estimates of the rotor position, angular velocity, and other motor states. Although EKFs generally provide superior estimation performance, they require significantly greater computational resources and careful tuning of the process and measurement covariance matrices.

A generic observer-based estimation process can be summarized as follows:

1. Measure the stator voltages and phase currents.
2. Predict the motor electrical states using the PMSM mathematical model.
3. Compare the predicted variables with the measured values.
4. Compute the estimation error.
5. Update the observer states according to the selected estimation algorithm.
6. Use the estimated rotor position and speed within the FOC algorithm in place of the Hall sensor measurements.

The estimated electrical angle, $\hat{\theta}_e$, directly replaces the measured rotor position in the Park and inverse Park transformations, allowing the Field-Oriented Control algorithm to operate without mechanical position sensors.

Although the implementation of advanced sensorless estimation techniques is beyond the scope of the present work, they represent one of the most significant future developments of the proposed drive system. Integrating observer-based sensorless control would eliminate the dependence on Hall sensors while enabling high-performance operation across the complete speed range, including low-speed operation and standstill. Such an extension would considerably increase the industrial applicability and robustness of the developed PMSM drive.

Chapter 5

Conclusions

This dissertation investigated the modelling, implementation, and comparative evaluation of advanced control techniques for PMSMs using the PSIM simulation environment. The work combined theoretical analysis with practical controller development to establish a common simulation framework capable of implementing, validating, and comparing DTC and FOC. In addition, several advanced features commonly required in industrial motor drives were explored to demonstrate how the developed controller architecture can be further extended.

The dissertation began with a review of brushless motor technology, covering the operating principles of PMSMs, their advantages over conventional brushed machines, and the evolution of modern control techniques. Different rotor position detection methods and mathematical models were also examined to establish the theoretical foundation required for controller implementation.

Based on these fundamentals, a complete PMSM drive system was developed in PSIM. A common plant model consisting of the inverter, PMSM, DC supply, and mechanical load was used throughout the work to ensure that both control strategies were evaluated under identical operating conditions. This common architecture enabled a fair comparison while simplifying the integration of additional control features.

The DTC implementation focused on direct regulation of stator flux and electromagnetic torque using hysteresis controllers and a switching table. Simulation results demonstrated the expected characteristics of DTC, including rapid torque response, simple controller structure, and independence from inner current control loops. At the same time, the results also highlighted its inherent disadvantages, namely increased torque ripple, variable switching frequency, and larger current distortion.

The FOC implementation followed the conventional cascade control structure based on Clarke and Park transformations, independent d-axis and q-axis current regulation, and Space Vector PWM (SVPWM) modulation. Validation demonstrated accurate speed tracking, smooth steady-state operation, and significantly reduced torque ripple compared to DTC. Although the controller architecture is considerably more complex and relies on accurate rotor position information, the obtained results confirmed the superior current quality and overall control performance that make FOC the preferred solution for high-performance servo applications.

A direct comparison between both control methods illustrated the practical trade-offs discussed in the literature. DTC offers a simpler implementation and excellent transient torque response but suffers from higher ripple and variable switching behaviour. Conversely, FOC requires greater computational complexity and accurate position estimation but provides improved steady-state accuracy, smoother torque production, and lower current distortion. These observations are consistent with the theoretical expectations presented in the literature review and demonstrate the suitability of each technique for different application requirements.

The work was further extended by implementing advanced features for the FOC platform. Parameter identification procedures were developed to estimate the stator resistance, d-axis and q-axis inductances, permanent magnet flux linkage, and mechanical parameters. These routines demonstrate how motor parameters can be obtained directly from the drive system without relying solely on manufacturer specifications, thereby improving model accuracy and facilitating controller tuning.

An additional Hall sensor-based rotor angle and speed estimation method was also integrated into the simulation platform. Validation confirmed that the estimated electrical angle followed the actual rotor position with the expected six-step behaviour while providing sufficiently accurate speed estimation for closed-loop control. The resulting FOC implementation successfully tracked the reference speed using only the Hall-derived position information, demonstrating the feasibility of low-cost sensored operation without requiring high-resolution encoders.

Finally, two additional topics—load torque estimation and automatic PI controller tuning—were presented as future developments. Although these methods were not implemented within the scope of this dissertation, they illustrate natural extensions of the developed platform towards more intelligent and adaptive motor control systems.

5.1 Summary of Contributions

The principal contributions of this dissertation can be summarised as follows:

- A comprehensive review of PMSM operating principles, mathematical modelling, control strategies, and rotor position detection methods.
- Development of a complete PMSM drive system in PSIM using a common simulation architecture suitable for multiple control strategies.
- Implementation and validation of a complete Direct Torque Control (DTC) drive.
- Implementation and validation of a complete Field-Oriented Control (FOC) drive based on Clarke and Park transformations and Space Vector PWM.
- Comparative evaluation of DTC and FOC under identical simulation conditions.
- Development and validation of parameter identification procedures for the principal electrical and mechanical motor parameters.

- Implementation and validation of Hall sensor-based rotor angle and speed estimation integrated within the FOC controller.
- Identification of future extensions involving load torque estimation and automatic PI controller tuning.

Together, these contributions provide a modular simulation framework that can be used for further research, controller development, and educational purposes.

5.2 Key Findings

Several important conclusions can be drawn from the work presented in this dissertation.

First, both DTC and FOC successfully achieved closed-loop control of the PMSM, confirming the correctness of their respective implementations within PSIM.

Second, the comparative study reinforces the complementary nature of the two control strategies. DTC remains attractive because of its relatively simple structure and excellent dynamic torque response, whereas FOC provides superior steady-state behaviour through lower torque ripple, smoother phase currents, and improved speed regulation.

Third, accurate motor parameter identification plays a fundamental role in high-performance vector control. The implemented identification procedures demonstrate that the key electrical and mechanical parameters required for FOC can be obtained systematically, reducing dependence on nominal manufacturer data and supporting more reliable controller tuning.

Fourth, Hall-effect sensors provide sufficient position information to achieve stable FOC operation in applications where the highest positioning accuracy is not required. Although their limited angular resolution restricts estimation accuracy compared to high-resolution encoders or advanced sensorless observers, the validation results demonstrate that Hall-based estimation represents a practical and cost-effective solution for many industrial drives.

Finally, the modular architecture developed throughout this dissertation proved well suited for extending the controller with additional estimation and automation algorithms, illustrating the flexibility of simulation-based controller development before hardware implementation.

5.3 Limitations and Future Work

Although the objectives of this dissertation were successfully achieved, several limitations remain.

The work is entirely simulation-based and therefore does not account for all practical effects encountered in real hardware implementations, such as inverter dead time, measurement noise, sensor offsets, electromagnetic interference, parameter drift due to temperature variation, and non-ideal switching behaviour. Experimental validation using a physical PMSM drive would therefore represent the next logical step.

The implemented controllers also assume relatively accurate machine parameters and ideal measurement signals. In industrial environments, parameter uncertainties and measurement errors can significantly influence controller performance, particularly for Field-Oriented Control.

Future work may therefore focus on several complementary research directions.

The first extension is the implementation of online load torque estimation to improve disturbance rejection and provide additional diagnostic information without requiring external torque sensors.

A second extension involves automatic PI controller tuning, allowing controller gains to be calculated directly from the identified motor parameters and operating conditions, thereby reducing commissioning time and improving controller robustness.

Additional research may investigate advanced sensorless estimation techniques capable of replacing Hall sensors over the complete speed range, including low-speed operation and standstill. Observer-based methods, Model Reference Adaptive Systems (MRAS), or Extended Kalman Filters (EKFs) represent promising candidates for improving estimation accuracy.

Finally, experimental implementation on embedded hardware would enable verification of the developed algorithms under real operating conditions and facilitate the evaluation of computational requirements, sampling constraints, and practical implementation issues.

Overall, the work presented in this dissertation establishes a complete simulation platform for PMSM control while demonstrating the implementation and validation of two of the most widely used industrial control strategies. The developed framework provides a solid foundation for future research into advanced estimation, adaptive control, and real-time embedded motor drive systems.

References

[1] R. Krishnan, *Permanent Magnet Synchronous and Brushless DC Motor Drives*. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2010, Electrical and Computer Engineering Department, Virginia Tech, Blacksburg, Virginia, U.S.A., ISBN: 9780824753849.

[2] P. Yedamale, *An885: Brushless dc motor fundamentals*, Application Note, Microchip Technology Inc., 2003. [Online]. Available: https://ww1.microchip.com/downloads/en/appnotes/00885a.pdf

[3] B. K. Bose, *Modern Power Electronics and AC Drives*. Upper Saddle River, NJ: Prentice Hall PTR, 2002, ISBN: 0-13-016743-6.

[4] S. J. Chapman, *Fundamentos de Máquinas Elétricas*, 5th ed., trans. by A. Laschuk. Porto Alegre, RS: AMGH Editora Ltda., 2013, Tradução da obra "Electric Machinery Fundamentals, 5th Edition", McGraw-Hill, 2012, ISBN: 978-85-8055-207-2.

[5] B. Wu and M. Narimani, *High-Power Converters and AC Drives*, 2nd ed. Hoboken, NJ: Wiley–IEEE Press, 2017, ISBN: 9781119156079.

[6] J. R. Hendershot and T. J. E. Miller, *Design of Brushless Permanent-Magnet Machines*. Venice, FL, USA: Motor Design Books LLC, 2010, Rewritten successor to *Design of Brushless Permanent-Magnet Motors* (1994), ISBN: 978-0-9840687-0-8.

[7] C.-l. Xia, *Permanent Magnet Brushless DC Motor Drives and Controls*. Singapore: John Wiley & Sons Singapore Pte. Ltd., 2012, ISBN: 978-1-118-18833-0.

[8] F. Qi, D. Scharfenstein, C. Weiss, C. Müller, and U. Schwarzer, “Motor handbook,” Institute for Power Electronics, Electrical Drives, RWTH Aachen University, and Infineon Technologies AG, Munich, Germany, Tech. Rep., version 2.1, 2019, Release date: 12.03.2019.

[9] E. Devillers, J. Besnerais, Q. Souron, and M. Hecquet, *Characterization of acoustic noise and vibrations due to magnetic forces in induction machines for transport applications using manatee software*, Sep. 2016.

[10] T. J. E. Miller, *Brushless Permanent-Magnet and Reluctance Motor Drives* (Monographs in Electrical and Electronic Engineering 21). Oxford, UK: Clarendon Press / Oxford University Press, 1989, ISBN: 0-19-859369-4.

[11] GR Winding. “Concentrated winding and distributed winding.” Accessed: 2025-09-24. [Online]. Available: https://www.grwinding.com/concentrated-winding-and-distributed-winding/

[12] G. Du, N. Li, Q. Zhou, W. Gao, L. Wang, and T. Pu, “Multi-physics comparison of surface-mounted and interior permanent magnet synchronous motor for high-speed applications,” *Machines*, vol. 10, no. 8, p. 700, 2022. DOI: 10.3390/machines10080700 [Online]. Available: https://doi.org/10.3390/machines10080700

[13] IC Components, *Brushless dc motor: How it works, types, and benefits*, https://www.ic-components.com/blog/Brushless-DC-Motor-How-it-Works,Types,and-Benefits.jsp, Accessed: 2025-01-15, 2023.

[14] P. Yedamale, "Brushless dc (bldc) motor fundamentals," Microchip Technology Inc., Tech. Rep. AN885, 2003.

[15] B. K. Bose, *Power Electronics and Motor Drives: Advances and Trends*. Burlington, MA, USA: Academic Press, Elsevier, 2006, ISBN: 978-0-12-088405-6.

[16] MathWorks. "Six-step commutation." Accessed: 2025-09-24. [Online]. Available: https://www.mathworks.com/help/mcb/ref/sixstepcommutation.html

[17] ROHM Semiconductor, *Basic operation and switching modes of a 3-phase inverter*, https://techweb.rohm.com/product/transistors-diodes/transistors/15580/, Accessed: 2025-01-10, 2023.

[18] B. M. Wilamowski and J. D. Irwin, Eds., *The Industrial Electronics Handbook: Power Electronics and Motor Drives*, 2nd ed. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2011, ISBN: 978-1-4398-0286-1.

[19] S.-K. Sul, *Control of Electric Machine Drive Systems* (IEEE Press Series on Power Engineering 55). Hoboken, NJ: John Wiley & Sons, IEEE Press, 2011, ISBN: 978-0-470-59079-9.

[20] B. Akin, M. Bhardwaj, and J. Warriner, "Sensorless field oriented control of 3-phase permanent magnet synchronous motors," Texas Instruments, Tech. Rep. SPRABY9, 2011, C2000 Systems and Applications Team.

[21] The MathWorks, Inc. "Permanent magnet synchronous machine field-oriented control." Accessed: 2026-02-03. [Online]. Available: https://www.mathworks.com/help/sps/ref/pmsmfieldorientedcontrol.html

[22] K. S. Krishnan, *Clarke and park transforms*, LinkedIn post, Published March 24, 2021. Used as a visual illustration of FOC structure, Mar. 2021.

[23] D. Mohanraj et al., "A review of bldc motor: State of art, advanced control techniques, and applications," *IEEE Access*, vol. 10, pp. 1–27, 2022. DOI: 10.1109/ACCESS.2022.317501

[24] HaydonKerk Pittman. "Connecting brushless dc motors to electronic controllers." Accessed: 2025-09-25. [Online]. Available: https://www.haydonkerkpittman.com/learningzone/whitepapers/connecting-brushless-dc-motors-to-electronic-controllers

[25] Lorentzzi. "Quadrature encoder: 6 things you need to know." Accessed: 2025-09-25. [Online]. Available: https://lorentzzi.com/quadrature-encoder-6-things-you-need-to-know/

[26] Asahi Kasei Microdevices (AKM). "Absolute encoders: Mechanism and types." Accessed: 2025-09-25. [Online]. Available: https://www.akm.com/global/en/products/rotation-angle-sensor/tutorial/type-mechanism-2/

[27] J. Holtz, "Sensorless control of induction motor drives," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1359–1394, 2002.

[28] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, and S. D. Pekarek, *Analysis of Electric Machinery and Drive Systems*, 3rd ed. Hoboken, NJ: John Wiley & Sons, IEEE Press, 2013, ISBN: 978-1-118-02429-4.

[29] NXP Semiconductors, “An14164: Current sensing techniques in motor control applications,” NXP Semiconductors, Tech. Rep. AN14164, Feb. 2024, Application Note, Rev. 1.0.

[30] MathWorks. “Sensorless field-oriented control of pmsm using dc shunt current sensing.” Accessed: 2026-06-01. [Online]. Available: https://www.mathworks.com/help/mcb/gs/sensorless-foc-of-pmsm-dc-shunt-current-sensing.html

[31] CubeMars. “Ri100 kv105 frameless inrunner torque motor.” Accessed: 2026-06-02. [Online]. Available: https://www.cubemars.com/product/ri100-kv105-frameless-inrunner-torque-motor.html

Appendix A

PSIM Implementation of the Speed and Position Processing Chain

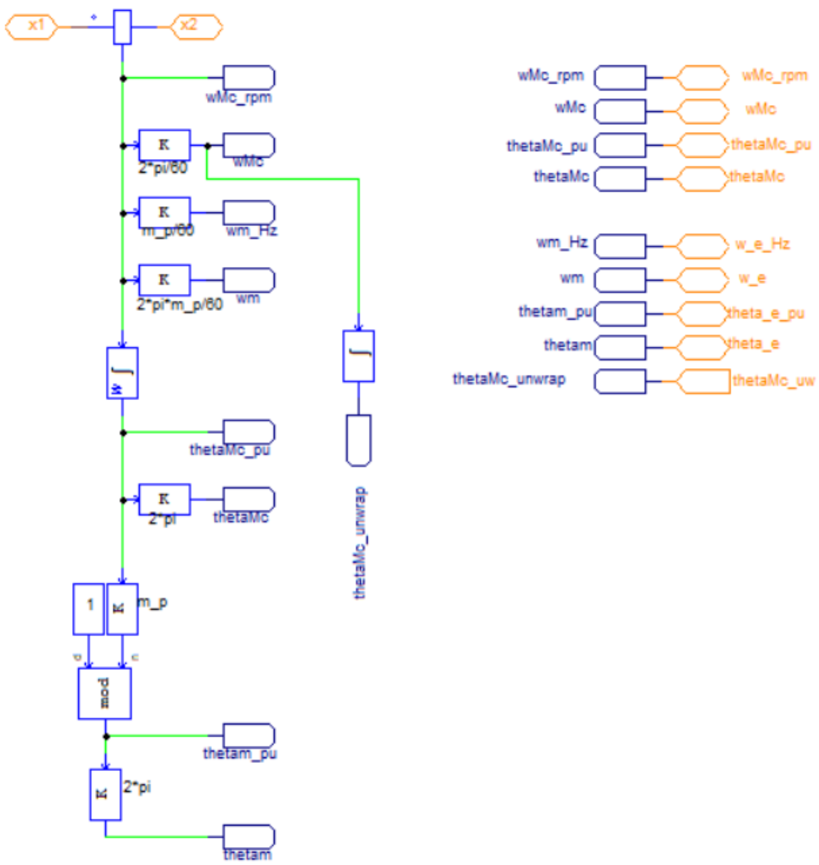


Figure A.1: PSIM implementation of the speed and position signal processing subsystem. The measured rotor speed is converted into mechanical and electrical speed quantities and integrated to obtain the corresponding mechanical and electrical position signals.

Appendix B

Validation of the Signal Flow of the FOC

B.1 Basic Elements Necessary fot the FOC

Basic Signal Flow:

$$\text{Mechanical Speed} \rightarrow \text{Electrical Speed} \rightarrow \text{Electrical Angle} \rightarrow \text{Clarke} \rightarrow \text{Park}$$

Objective: Analysing the internal signal flow of the FOC and validating the reference-frame transformations independently of any electrical excitation. The inverter stage, PWM modulation, and current control loops were intentionally not included. The PMSM was mechanically driven by a constant-speed load imposing a predefined rotor speed. This approach decoupled the mechanical motion from electromagnetic torque production, allowing a controlled validation of the angular reconstruction and coordinate transformations under deterministic conditions.

Additionally, the same sequence of operations was implemented within a custom C block, reproducing the complete transformation chain algorithmically. The equivalence between the PSIM logical blocks and the C implementation confirmed that the mathematical operations could be reliably translated into code, an essential prerequisite for later embedded implementation.

Here I established a mathematically consistent angular framework for the model. By imposing rotor motion mechanically and isolating the reference-frame processing chain, the full transformation pipeline was validated independently of current control, PWM generation, or inverter nonlinearities.

B.1.1 PSIM Schematic

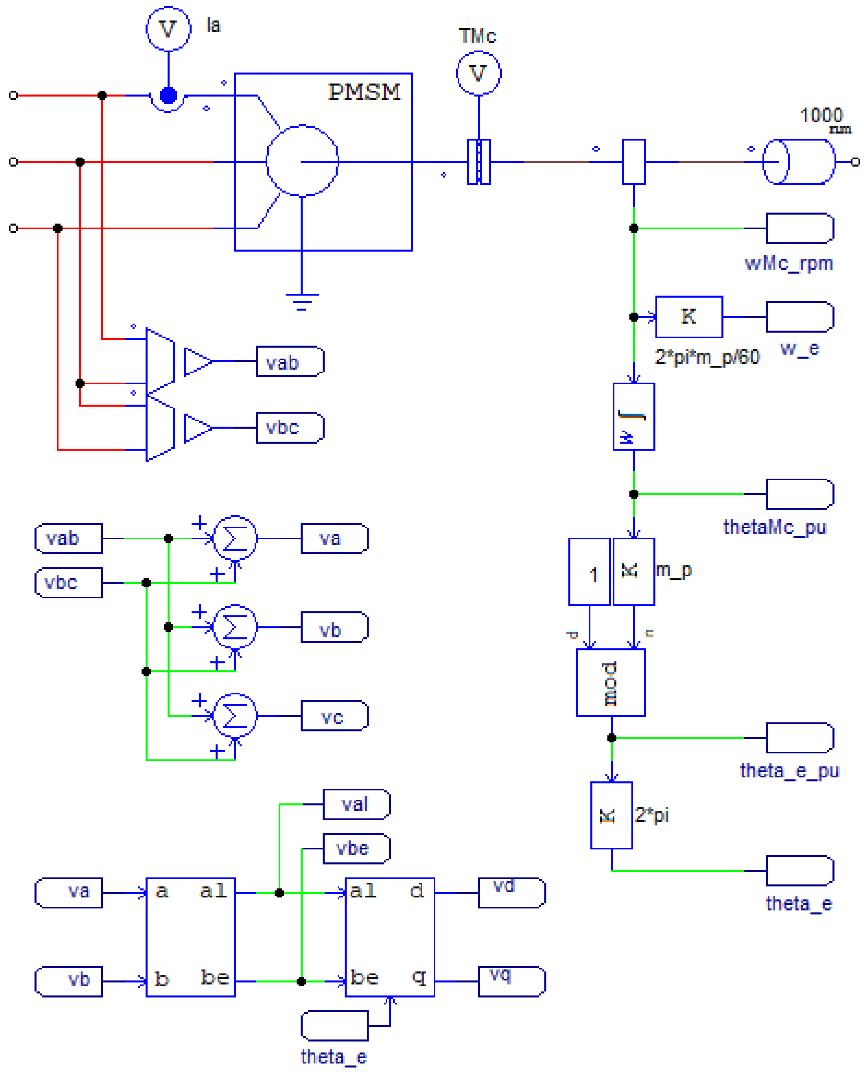


Figure B.1: baseline PSIM PMSM FOC template (overview).

Mechanical Subsystem and Imposed Speed The PMSM block is mechanically coupled to a constant-speed mechanical load. This load imposes a predefined shaft speed, effectively replacing the need for electromagnetic torque production during this validation phase.

Because the machine is not electrically excited, no stator currents are injected. The mechanical load directly enforces the rotor motion, allowing the angular quantities to be reconstructed and analyzed without the interaction of current controllers or PWM modulation.

Two mechanical quantities are measured:

- Mechanical speed $w_{Mc,\text{rpm}}$ (in rpm)
- Mechanical torque T_{Mc}

At this stage, torque serves only as a monitored variable, the imposed speed remains independent of electromagnetic dynamics.

Angle Reconstruction in Per-Unit and Radians The electrical speed signal is integrated using an internal resettable integrator. The integrator time constant is set to 60, ensuring proper scaling when integrating a speed signal originally derived from rpm.

The integrator output is the mechanical angle in per-unit:

$$\theta_{Mc,pu} \tag{B.1}$$

where one per-unit corresponds to one full mechanical revolution.

The electrical angle in per-unit is then obtained by multiplying the mechanical per-unit angle by the number of pole pairs:

$$\theta_{e,pu} = m_p \, \theta_{Mc,pu} \tag{B.2}$$

Because angular quantities are periodic, a MOD block is used to constrain the electrical angle within the interval:

$$0 \leq \theta_{e,pu} < 1 \tag{B.3}$$

Finally, the electrical angle in radians is obtained through:

$$\theta_e = 2\pi \, \theta_{e,pu} \tag{B.4}$$

This angle θ_e is the key variable required for the Park and Clarke transformations.

Voltage Measurement and Phase Reconstruction Two line-to-line voltages are measured using voltage sensors:

- v_{ab}
- v_{bc}

Assuming a balanced three-phase system with no neutral connection, the individual phase voltages are reconstructed from the measured line-to-line voltages. Using algebraic summation blocks, the phase voltages are obtained as:

$$v_a, \quad v_b, \quad v_c \tag{B.5}$$

This reconstruction enables validation of the reference-frame transformations without requiring direct phase measurements.

Clarke and Park Transformations The reconstructed phase voltages are applied to a Clarke transformation block, generating the stationary reference-frame components:

$$v_\alpha, \quad v_\beta \tag{B.6}$$

The Clarke transformation reduces the three-phase system into an orthogonal two-axis representation while preserving power invariance.

Subsequently, the Park transformation is applied using the previously computed electrical angle θ_e. This results in the rotating reference-frame components:

$$v_d, \quad v_q \tag{B.7}$$

At this stage of development, the Park transformation serves purely as a validation tool. Since the machine is not excited and no inverter switching is active, the transformed quantities serve purely as mathematical validation signals.

The transformed quantities were used to verify:

- Correct angle reconstruction,
- Proper phase ordering,
- Correct implementation of Clarke and Park transformations,
- Consistency of reference-frame alignment.

B.1.2 Validation and Expected Behaviors

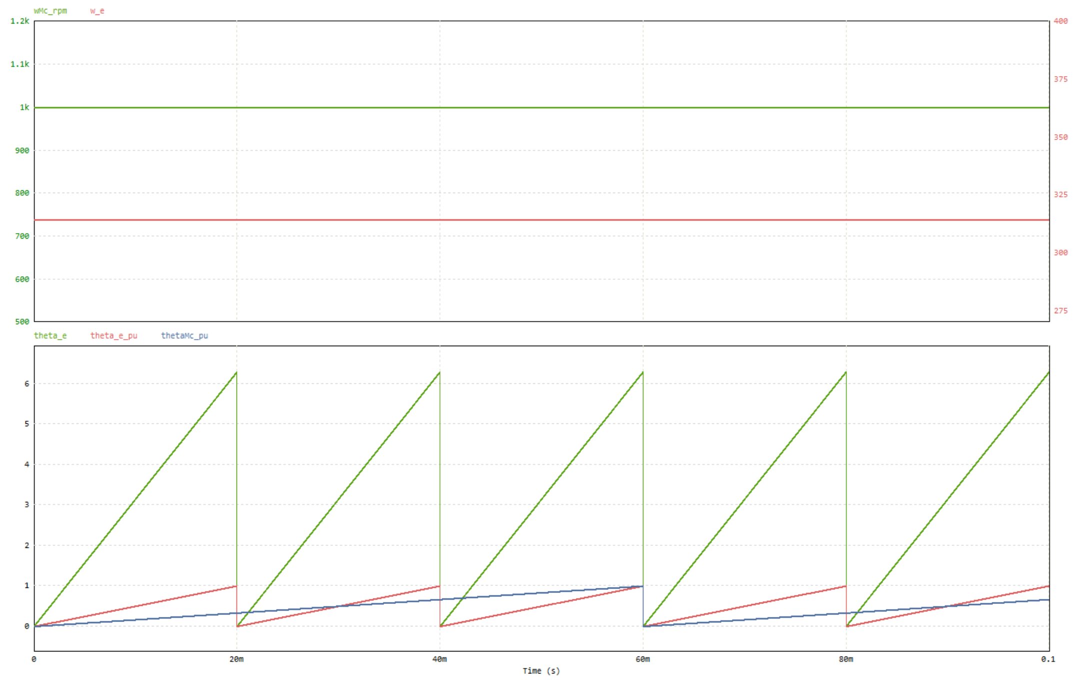


Figure B.2: Mechanical speed and reconstructed electrical quantities during imposed constant-speed operation. Top: imposed mechanical speed $w_{Mc,rpm}$ and computed electrical speed ω_e. Bottom: electrical angle θ_e (rad), electrical angle in per-unit $\theta_{e,pu}$, and mechanical angle in per-unit $\theta_{Mc,pu}$.

Figure B.2 validates the speed scaling and angular reconstruction chain. The mechanical load imposes a constant speed of 1000 rpm, resulting in a constant electrical speed of approximately $\omega_e \approx 314\,\text{rad/s}$, consistent with the conversion $\omega_e = \frac{2\pi m_p}{60} w_{Mc,rpm}$ for $m_p = 3$.

The lower plot shows the expected linear ramp behavior of the reconstructed angles. The mechanical per-unit angle $\theta_{Mc,pu}$ increases linearly and wraps at unity. The electrical per-unit angle $\theta_{e,pu}$ scales proportionally with the number of pole pairs and exhibits periodic wrapping. The electrical angle in radians θ_e forms the corresponding sawtooth waveform between 0 and 2π, confirming correct integration, scaling, and modulo implementation.

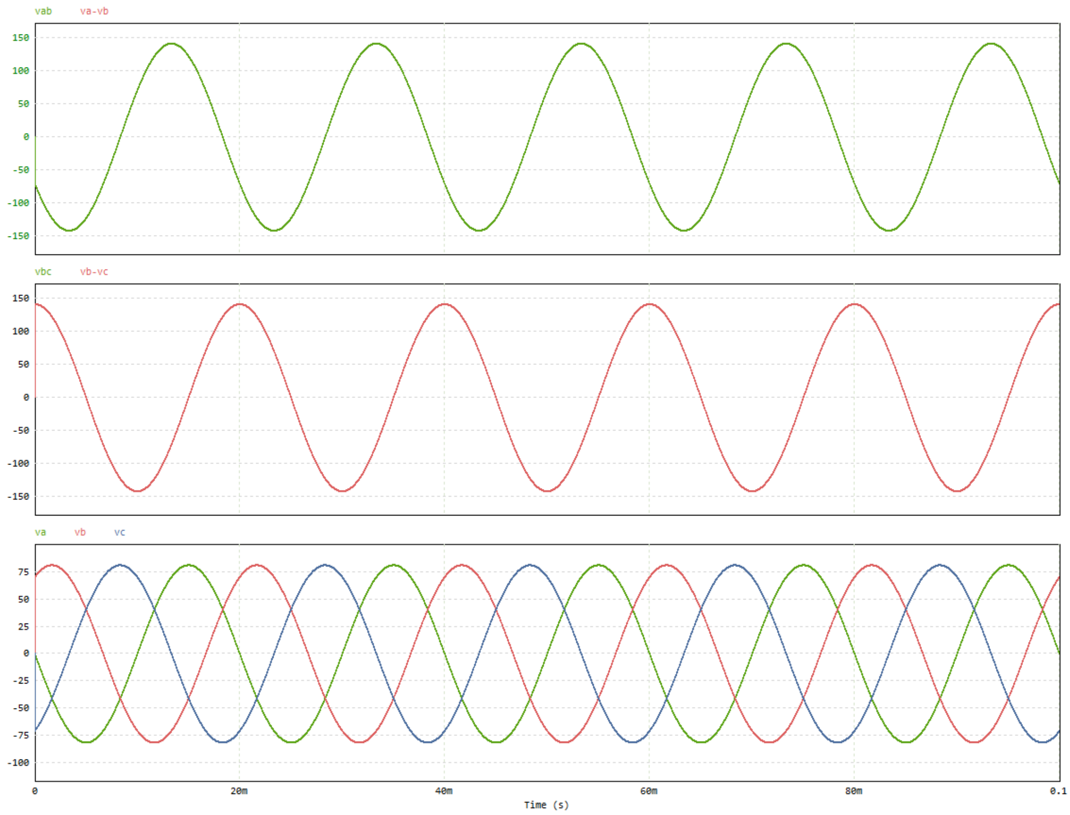


Figure B.3: Validation of phase-voltage reconstruction. Top: v_{ab} compared with $(v_a - v_b)$. Middle: v_{bc} compared with $(v_b - v_c)$. Bottom: reconstructed phase voltages v_a, v_b, and v_c.

Figure B.3 confirms the correctness of the phase-voltage reconstruction. The measured line-to-line voltages v_{ab} and v_{bc} perfectly overlap with the reconstructed differences $(v_a - v_b)$ and $(v_b - v_c)$, respectively, validating the algebraic implementation.

The reconstructed phase voltages v_a, v_b, and v_c form a balanced three-phase set with 120° phase displacement and equal amplitude. The balanced condition implies:

$$v_a + v_b + v_c \approx 0, \tag{B.8}$$

which is satisfied numerically, confirming internal consistency and correct phase ordering.

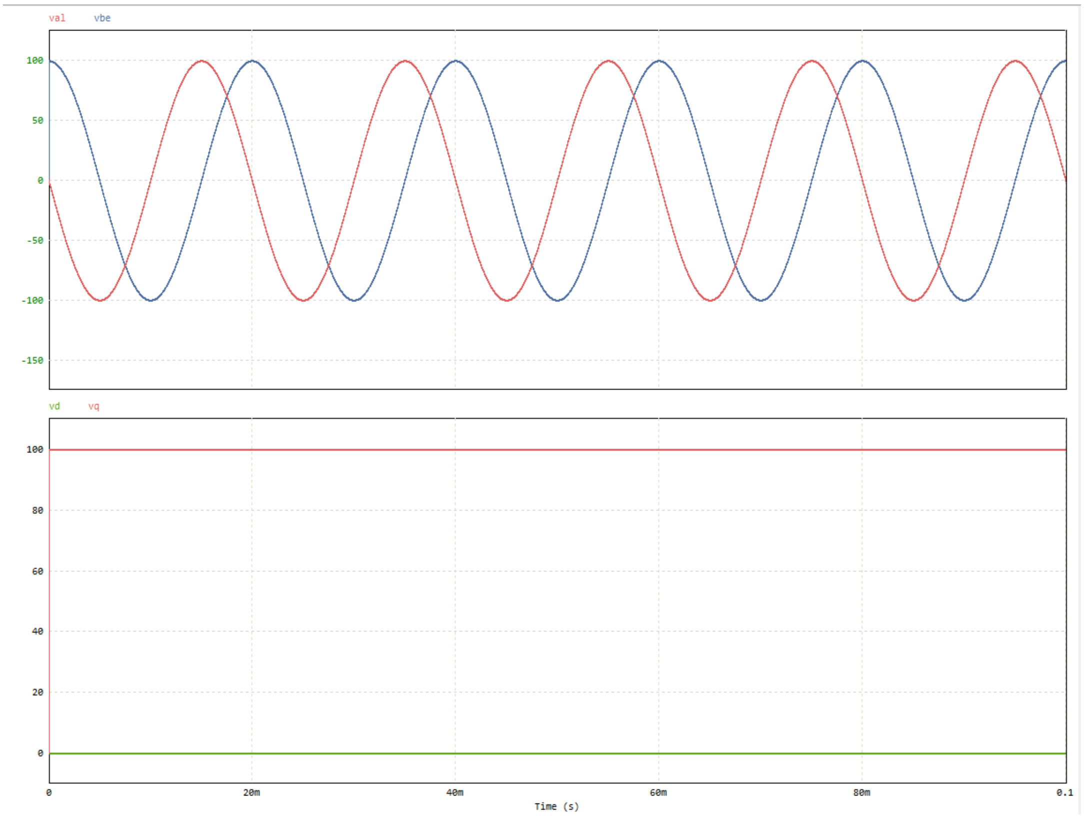


Figure B.4: Validation of Clarke and Park transformations. Top: stationary-frame voltages v_α and v_β. Bottom: rotating-frame voltages v_d and v_q.

Figure B.4 validates the Clarke and Park transformation chain. The Clarke transformation converts the balanced three-phase system into orthogonal stationary components (v_α, v_β), which exhibit sinusoidal behavior with 90° phase displacement.

Using the reconstructed electrical angle θ_e, the Park transformation maps these stationary components into the rotating reference frame. As expected for a perfectly aligned rotating frame under constant speed, one component becomes constant while the orthogonal component collapses to zero. In this case, v_q remains constant while $v_d \approx 0$, confirming correct angle reconstruction and reference-frame alignment.

This behavior verifies that the transformation pipeline preserves magnitude and correctly aligns the rotating frame with the electrical vector, completing the validation:

$$v_{abc} \rightarrow v_{\alpha\beta} \rightarrow v_{dq}$$

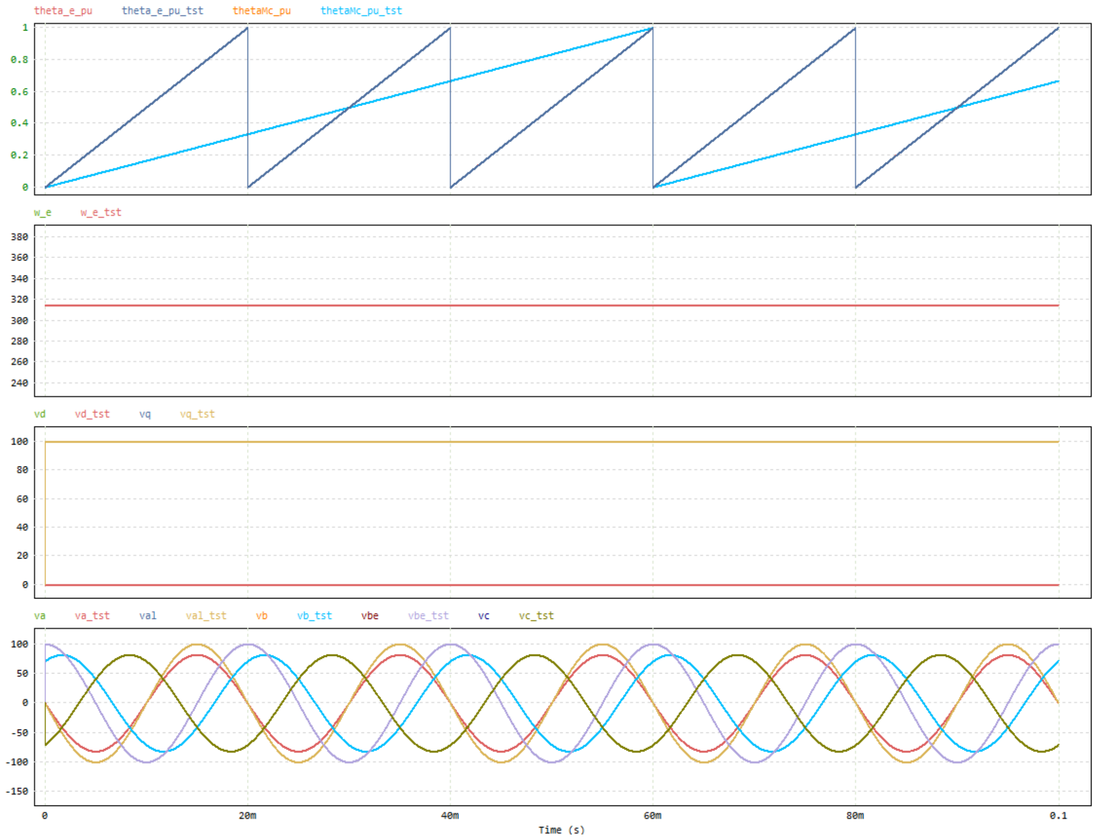


Figure B.5: C-block equivalence validation. Each C-block output (suffix `_tst`) is overlaid with the corresponding PSIM block signal for the complete chain: angle reconstruction ($\theta_{Mc,pu}$, $\theta_{e,pu}$), electrical speed ω_e, phase reconstruction (v_a, v_b, v_c), Clarke outputs (v_α, v_β), and Park outputs (v_d, v_q). The perfect overlap confirms numerical equivalence between the block-based and code-based implementations.

Figure B.5 confirms that the custom C block reproduces the PSIM logical signal chain exactly: all `_tst` traces overlap with the corresponding PSIM signals within numerical tolerance. This validates that the discretization (trapezoidal integration), angle wrapping, line-to-line voltage reconstruction, and Clarke/Park transformations can be implemented in code without changing conventions. The complete C-block implementation is provided in Appendix B.1.2. This equivalence constitutes a necessary prerequisite for the subsequent transition from simulation to real-time embedded execution.

This appendix contains the full, unaltered C-block code used to replicate the PSIM block-based signal chain (angle reconstruction and Clarke/Park transformations) for equivalence validation.

```
static double wMc_rpm_last;
static double wMc_acc_pu;

double wMc_rpm;
double vab, vbc, va, vb, vc, valpha, vbeta, vd, vq;
```

```
double thetaMc_pu;
double wm;
double thetam_pu;
double thetam;

double tmp;
double tmpcos;
double tmpsin;

#define pi 3.1415926535897932385
#define _2pi 6.2831853071795864770

//----------------------input variables---------------------
wMc_rpm = x1;
vab = x2;
vbc = x3;

//---------------------------------------
wMc_acc_pu = wMc_acc_pu + (wMc_rpm+wMc_rpm_last) * delt / (2*60.0);
         if ( wMc_acc_pu > 1 ) wMc_acc_pu = wMc_acc_pu - 1;
else if ( wMc_acc_pu < 0 ) wMc_acc_pu = wMc_acc_pu + 1;
wMc_rpm_last = wMc_rpm;

thetaMc_pu = wMc_acc_pu;

//------------------------------------------
wm = wMc_rpm * _2pi / 60 * m_p;
tmp = thetaMc_pu*m_p;

//thetam_pu = tmp - ceil(tmp) + 1;
thetam_pu = tmp - floor(tmp);   // gives [0,1)

thetam = thetam_pu * _2pi;

//---------------------------------------------
va = (2*vab +   vbc)/3;
vb = (  -vab +   vbc)/3;
vc = (  -vab - 2*vbc)/3;

valpha = sqrt(6)/2*va;
```

```
vbeta = (sqrt(2)/2)*va + sqrt(2)*vb;

tmpcos = cos(thetam);
tmpsin = sin(thetam);
vd = tmpcos*valpha + tmpsin*vbeta;
vq = -tmpsin*valpha +tmpcos*vbeta;

//------------------output variables-------------
y1 = wMc_acc_pu;
y2 = wm;
y3 = thetam_pu;
y4 = thetam;

y5 = va;
y6 = vb;
y7 = vc;
y8 = valpha;
y9 = vbeta;
y10 = vd;
y11 = vq;
```

B.2 Resistive Load and Current/Torque Validation

Objective: Verify current behavior, electromagnetic torque production, and power consistency.

While previously I validated angle reconstruction and voltage transformations under open-circuit conditions, no stator current was flowing and therefore no torque or power transfer could be evaluated. Now its done with the intruduction of a controlled electrical load to enable current generation.

Key additions:

- Addition of a balanced three-phase resistive load (Star/Y connection).
- Three identical resistors of $R = 10\,\Omega$.
- Measurement of stator phase currents i_a and i_b.
- Clarke and Park transformation of currents to obtain i_d and i_q.
- Implementation of a torque consistency test signal:

$$T_{Mc\_tst1} = m_p\,\psi_f\,i_q$$

The resistive load ensures balanced operation and purely active power transfer, simplifying interpretation of torque and power results.

B.2.1 PSIM Schematic

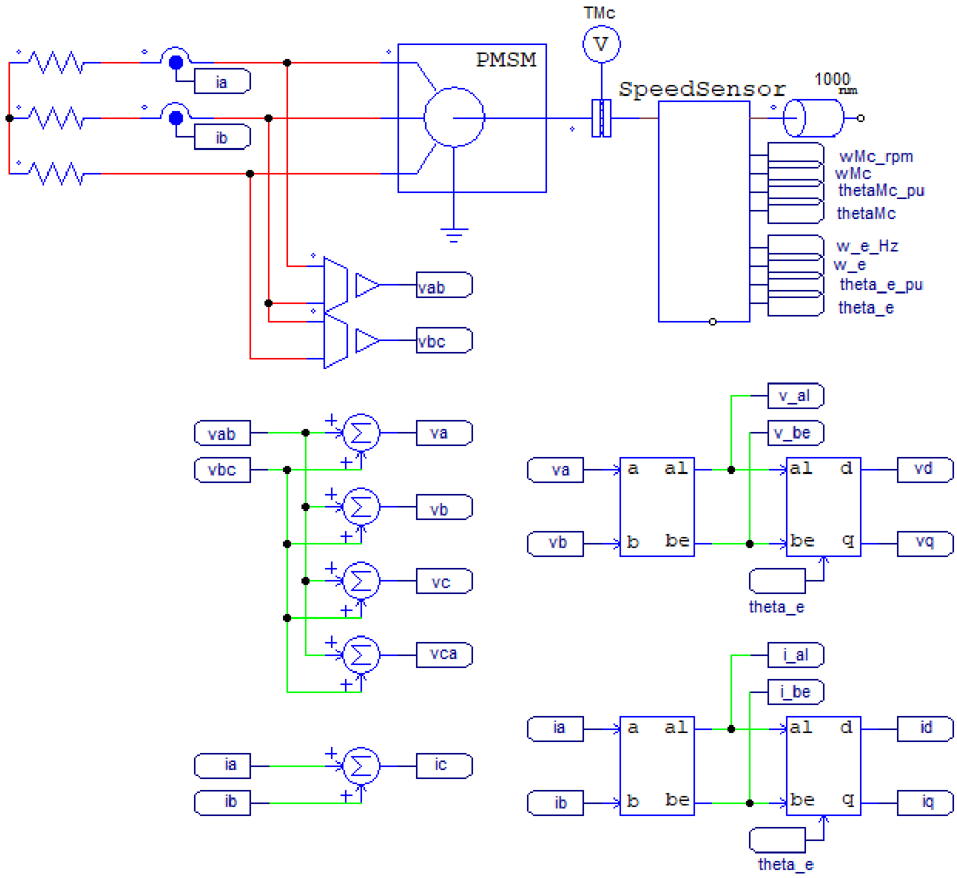


Figure B.6: PMSM connected to a balanced Y resistive load with current measurements and dq current reconstruction.

A balanced star-connected resistive load is connected to the stator terminals. Phase currents i_a and i_b are measured and used to reconstruct $i_c = -(i_a + i_b)$, assuming a balanced system.

The reconstructed phase currents are transformed using Clarke and Park transforms, in parallel with the voltage transformations already validated. This enables analysis in the synchronous reference frame.

B.2.2 Validation Tests and Results

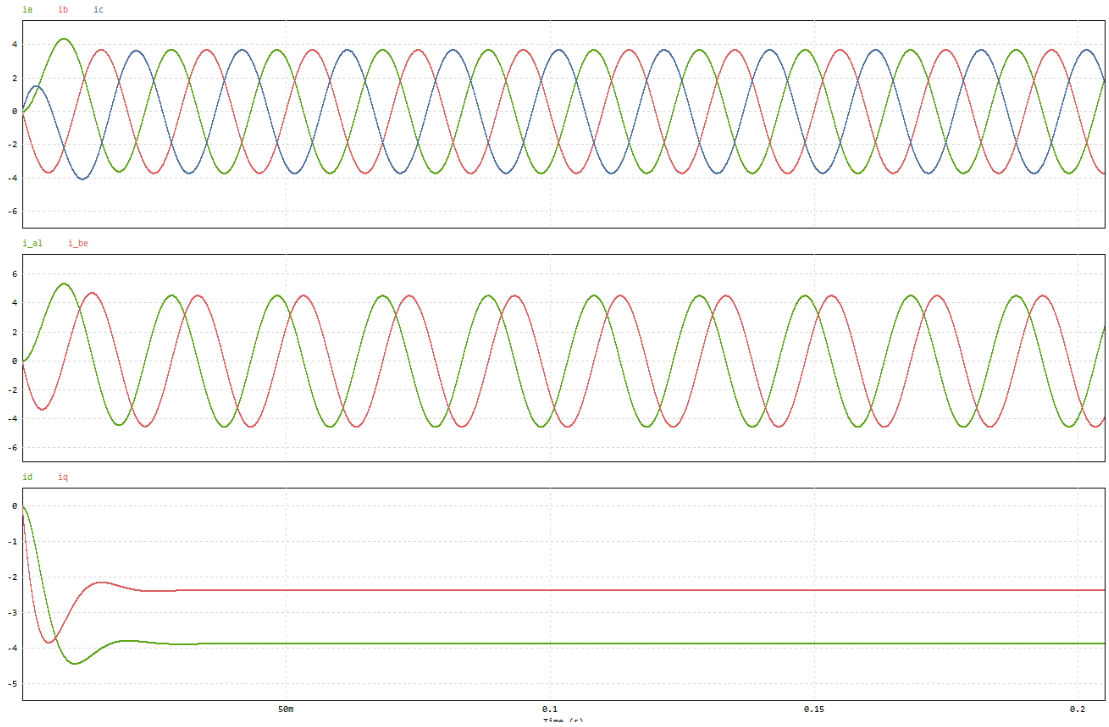


Figure B.7: Phase currents i_a, i_b, and i_c under balanced Y-connected resistive load (Top).Synchronous reference frame currents i_d and i_q (Bottom).

The phase currents i_a, i_b, and i_c are balanced and sinusoidal with 120° phase displacement and equal amplitude, confirming correct three-phase loading under the Y-connected resistive network. The steady-state behavior indicates proper voltage reconstruction and consistent electrical excitation.

The transformed currents i_d and i_q converge to constant values in steady-state. In this test, $i_d \neq 0$. This is expected, since no current control is applied. In ideal field-oriented control of a surface PMSM, i_d is typically regulated to zero to maximize torque per ampere. Here, however, the machine operates in open-loop generator mode with a passive resistive load. Without active control, there is no constraint enforcing $i_d = 0$, and the steady-state current vector orientation is determined solely by machine parameters and load impedance.

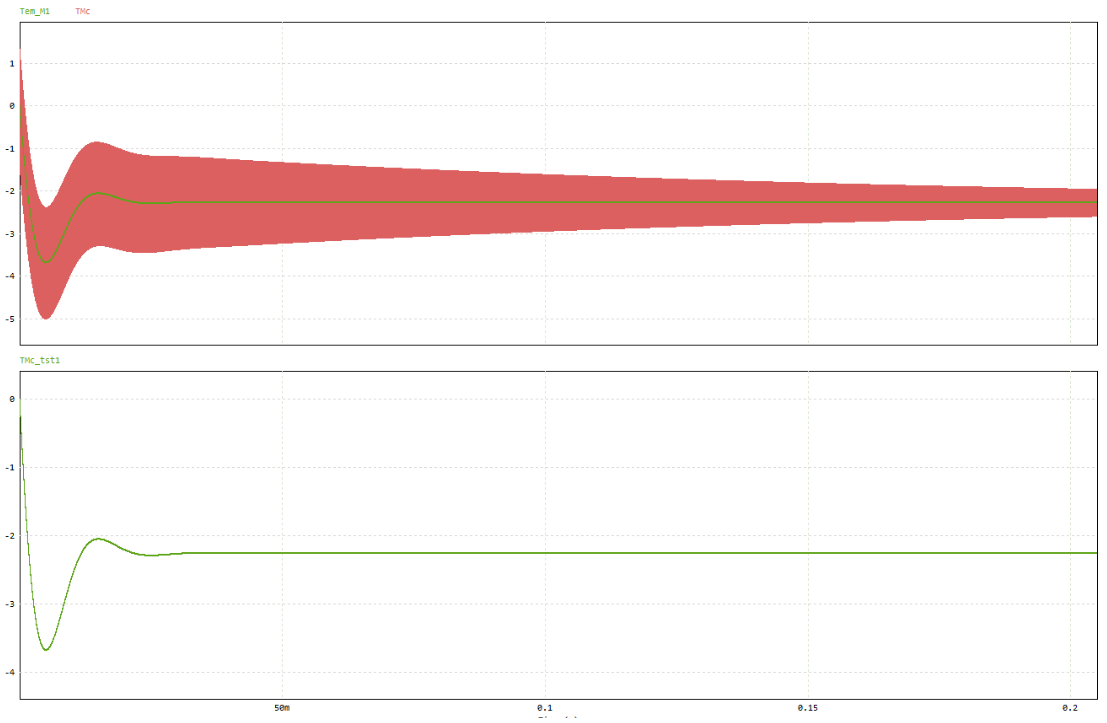


Figure B.8: Torque comparison: electromagnetic torque T_{em_M1}, analytical estimate $T_{Mc\_tst1} = m_p \psi_f i_q$, and measured shaft torque T_{Mc}.

Torque Consistency Verification Three torque signals are compared:

- T_{em_M1}: electromagnetic torque from the PMSM model,
- $T_{Mc\_tst1} = m_p \psi_f i_q$: analytical torque estimate,
- T_{Mc}: torque measured at the mechanical shaft.

The electromagnetic torque T_{em_M1} perfectly matches the analytical estimate T_{Mc_tst1} in steady-state. This confirms:

- Correct computation of i_q,
- Consistent permanent magnet flux ψ_f,
- Proper Park angle alignment.

The shaft torque T_{Mc} exhibits higher ripple compared to the internal electromagnetic torque. This distinction is physically meaningful. Electromagnetic torque is produced directly by stator–rotor interaction and can contain harmonic ripple due to electrical effects. The measured shaft torque corresponds to the torque transmitted through the mechanical path, which includes inertial effects ($J\dot{\omega}$), enforcement torque from the imposed speed source, and any numerical or torsional dynamics present in the mechanical interface.

Even under constant imposed speed, torque ripple persists because the electromagnetic torque itself is not perfectly constant in this open-loop generator configuration.

Overall, the agreement between T_{em_M1} and T_{Mc_tst1} validates the electrical model, while the observed differences at the shaft are consistent with expected mechanical dynamics.

The agreement between simulated torque and analytical estimation confirms the physical correctness of the model.

The custom C-block was reworked to include current reconstruction, dq transformation of currents, and analytical torque computation, ensuring parallel validation of the PSIM implementation.

```
static double wMc_rpm_last;
static double wMc_acc_pu;

double wMc_rpm;
double vab, vbc, va, vb, vc, valpha, vbeta, vd, vq;

double ia, ib, ic,  ialpha, ibeta, id, iq;

double thetaMc_pu;
double wm;
double thetam_pu;
double thetam;
double psi_f;
double Tem_tst1;
double pelec;

double tmp;
double tmpcos;
double tmpsin;

#define pi 3.14159265358979323885
#define _2pi 6.2831853071795864770

//------------------INPUTS-----------------------------------
wMc_rpm = x1;
vab = x2;
vbc = x3;
ia = x4;
ib = x5;
psi_f = x6;

//-------------------------------------------------------------------------------
wMc_acc_pu = wMc_acc_pu + (wMc_rpm+wMc_rpm_last) * delt / (2*60.0);
         if ( wMc_acc_pu > 1 ) wMc_acc_pu = wMc_acc_pu - 1;
```

```
else if ( wMc_acc_pu < 0 ) wMc_acc_pu = wMc_acc_pu + 1;
wMc_rpm_last = wMc_rpm;

thetaMc_pu = wMc_acc_pu;

//-----------------------------------------------------------------------
wm = wMc_rpm * _2pi / 60 * m_p;
tmp = thetaMc_pu*m_p;

//thetam_pu = tmp - ceil(tmp) + 1;
thetam_pu = tmp - floor(tmp);   // gives [0,1)

thetam = thetam_pu * _2pi;

//-----------------------------------------------------------------------
va = (2*vab +   vbc)/3;
vb = (  -vab +   vbc)/3;
vc = (  -vab - 2*vbc)/3;
ic = -(ia + ib);

tmpcos = cos(thetam);
tmpsin = sin(thetam);

valpha = sqrt(6)/2*va;
vbeta = (sqrt(2)/2)*va + sqrt(2)*vb;
vd = tmpcos*valpha + tmpsin*vbeta;
vq = -tmpsin*valpha +tmpcos*vbeta;

ialpha = sqrt(6)/2 * ia;
ibeta  = (sqrt(2)/2)*ia + sqrt(2)*ib;
id =  tmpcos*ialpha + tmpsin*ibeta;
iq = -tmpsin*ialpha + tmpcos*ibeta;
//-----------------------------------------------------------------

Tem_tst1 = m_p * psi_f * iq;
pelec = va*ia + vb*ib + vc*ic;   // instantaneous 3-phase power


//------------------OUTPUTS-----------------------------
y1 = wMc_acc_pu;
```

```
y2 = wm;
y3 = thetam_pu;
y4 = thetam;

y5 = va;
y6 = vb;
y7 = vc;
y8 = valpha;
y9 = vbeta;
y10 = vd;
y11 = vq;
y12 = ic;
y13 = ialpha;
y14 = ibeta;
y15 = id;
y16 = iq;
y17 = Tem_tst1;
y18 = pelec;
```

B.3 Inverter Integration and Voltage Modulation Validation

B.3.1 Objective and Model Changes

Objective: Introduce an ideal voltage source converter (VSC) supplied by a DC link and validate the voltage modulation and transformation chain from dq reference commands to three-phase machine terminals.

Until now, the PMSM operated either under open-circuit conditions or passive resistive loading. Here, active voltage excitation is introduced through an inverter stage, enabling controlled voltage vector injection into the machine. Important to note that the imposed speed is still active.

Key additions:

- Ideal three-phase voltage source converter (VSC).
- DC link source of $V_{dc} = 1000\,\mathrm{V}$.
- Cable model with $R_{cable} = 1\,\mathrm{m}\Omega$ and $L_{cable} = 1\,\mu\mathrm{H}$.
- Inverse Park and Clarke transformations for voltage command generation.
- Space Vector PWM (SVPWM) modulation block.

The commanded voltages $v_{c,d}$ and $v_{c,q}$ are transformed to phase commands $v_{c,a}$, $v_{c,b}$, $v_{c,c}$ via inverse Park and Clarke transformations. These serve as inputs to the SVPWM block, which generates modulation signals for the VSC.

B.3.2 PSIM Schematic

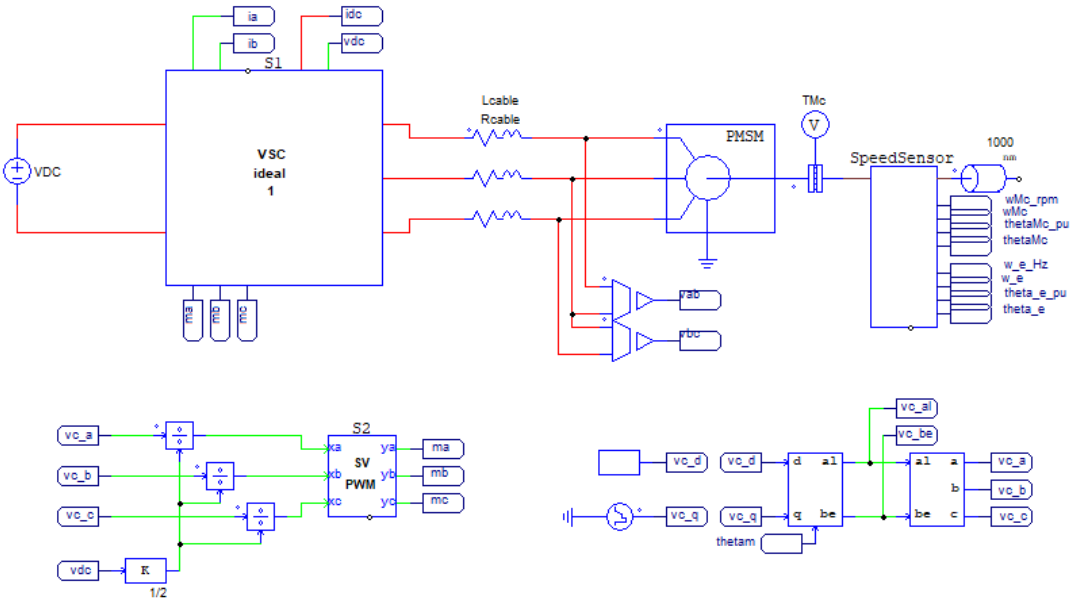


Figure B.9: Integration of ideal VSC, DC link, cable model, and SVPWM-based voltage command chain.

The ideal VSC is supplied by a 1000 V DC link and feeds the PMSM through a small cable impedance for increased physical realism.

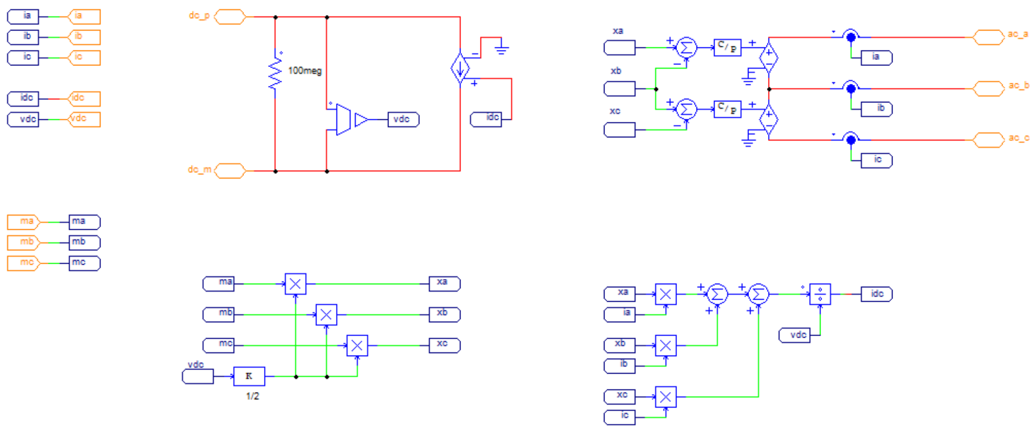


Figure B.10: Ideal voltage source converter (VSC) subcircuit implementation.

The ideal VSC represents a three-phase two-level inverter supplied by a constant DC link voltage $V_{dc} = 1000\,\text{V}$. Rather than modeling individual semiconductor switching devices in detail, the ideal converter assumes ideal switches and instantaneous transitions, allowing focus on modulation behavior and voltage synthesis.

The inverter converts DC link energy into three-phase AC voltages according to modulation signals (m_a, m_b, m_c) generated by the SVPWM block. The instantaneous phase voltages are proportional to the DC link voltage:

$$v_a = \frac{V_{dc}}{2} m_a, \quad v_b = \frac{V_{dc}}{2} m_b, \quad v_c = \frac{V_{dc}}{2} m_c$$

where the modulation indices are bounded by $|m_{a,b,c}| \leq 1$.

This introduces a fundamental limitation: the maximum achievable phase voltage amplitude depends directly on the DC link voltage. Consequently, the achievable synchronous voltage vector magnitude in the (d, q) frame is inherently limited by V_{dc}.

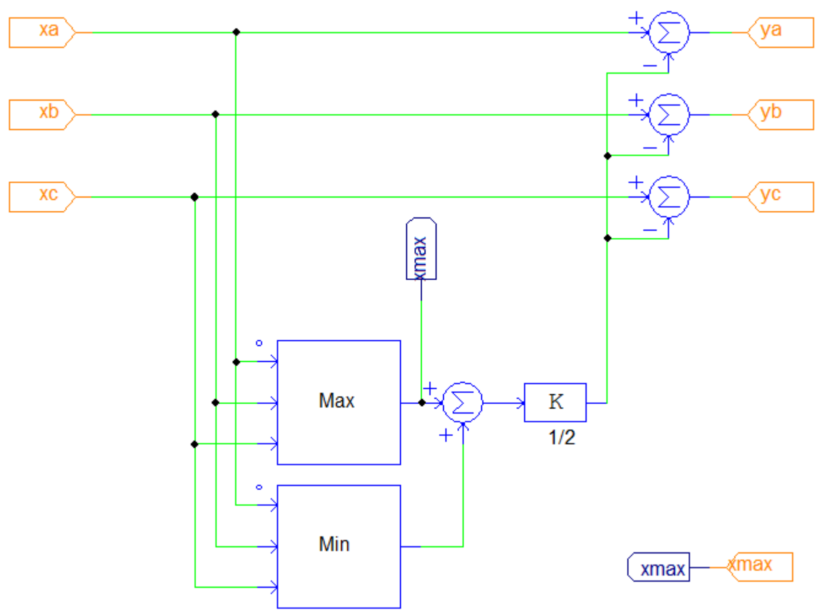


Figure B.11: SVPWM modulation block generating inverter switching commands from phase voltage references.

The SVPWM block converts the commanded phase voltages $(v_{c,a}, v_{c,b}, v_{c,c})$ into modulation signals for the inverter.

First, the commanded synchronous voltages $(v_{c,d}, v_{c,q})$ are transformed via inverse Park and Clarke transformations into three-phase references. These references represent the desired fundamental voltage vector in the stator reference frame.

SVPWM interprets the three-phase reference as a rotating space vector within the hexagonal voltage space of the two-level inverter. During each switching period, the algorithm selects adjacent active voltage vectors and distributes switching times such that their average equals the commanded voltage vector.

Together, the inverse transformations and SVPWM establish the complete voltage synthesis chain from synchronous reference commands $(v_{c,d}, v_{c,q})$ to physical inverter output voltages applied at the PMSM terminals.

B.3.3 Validation Tests and Results

To validate the inverter, transformation chain, and voltage-driven machine behavior, a sequence of step commands was applied to $(v_{c,d}, v_{c,q})$. This excites the dq dynamics under imposed constant speed, where back-EMF and cross-coupling are present and no current control is active.

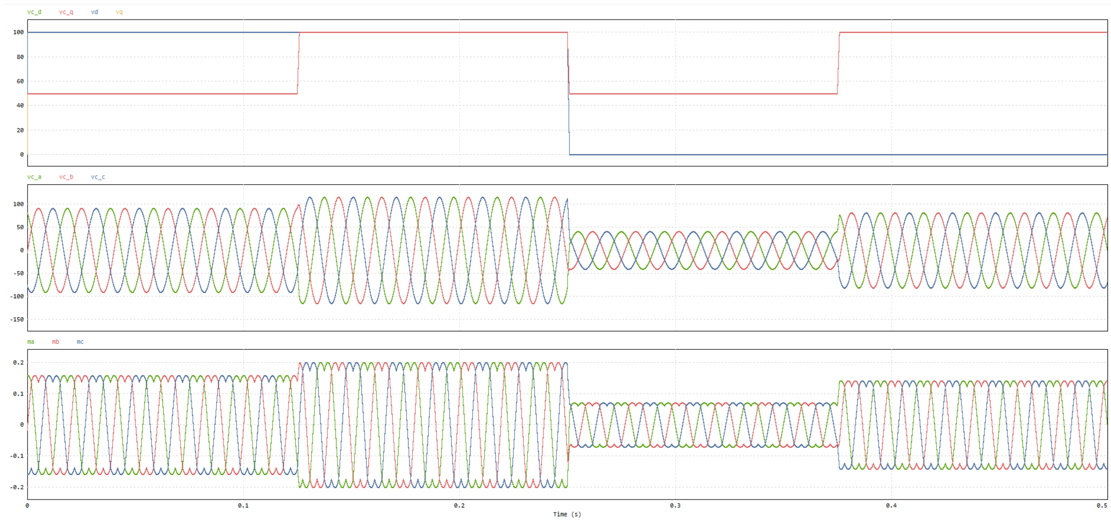


Figure B.12: Commanded voltages $(v_{c,d}, v_{c,q})$, reconstructed (v_d, v_q), resulting phase voltages, and modulation signals (m_a, m_b, m_c).

Figure B.12 validates the complete voltage synthesis chain. The reconstructed voltages (v_d, v_q) follow the commanded references $(v_{c,d}, v_{c,q})$, confirming: (i) correct inverse Park/Clarke generation of abc references, (ii) correct SVPWM operation, and (iii) consistent voltage sensing and abc→dq reconstruction. The modulation signals (m_a, m_b, m_c) scale with the commanded voltage magnitude and remain bounded. Switching ripple is expected due to PWM, the validation criterion is agreement of the averaged dq voltages with the commands.

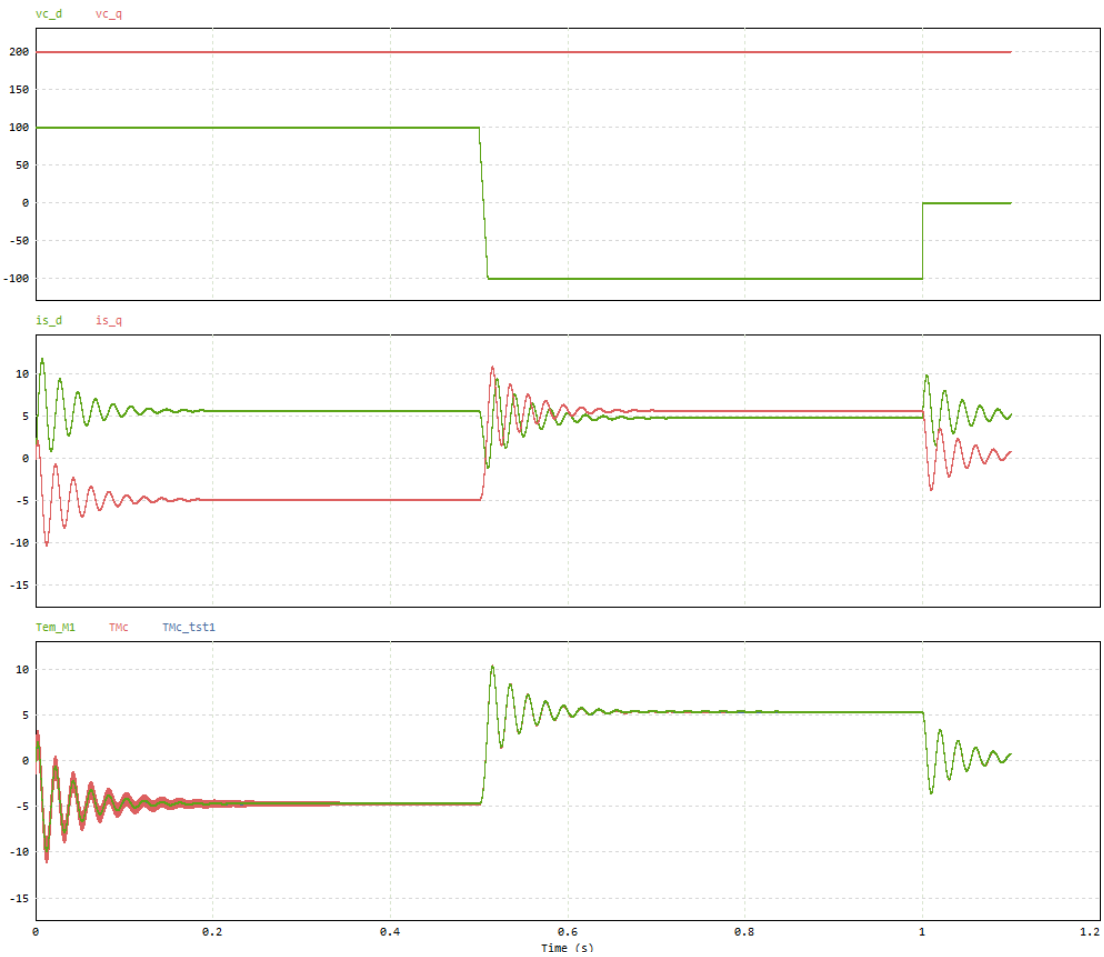


Figure B.13: Effect of $v_{c,d}$ stepping on (i_d, i_q) and electromagnetic torque.

Figure B.13 shows that stepping $v_{c,d}$ produces a dominant change in i_q in this open-loop voltage-driven configuration. The electromagnetic torque follows the evolution of i_q, confirming that torque production is primarily governed by the q-axis current component. The observed cross-axis sensitivity reflects the inherent dq coupling of PMSM dynamics under nonzero electrical speed, since no decoupling or current regulation is applied.

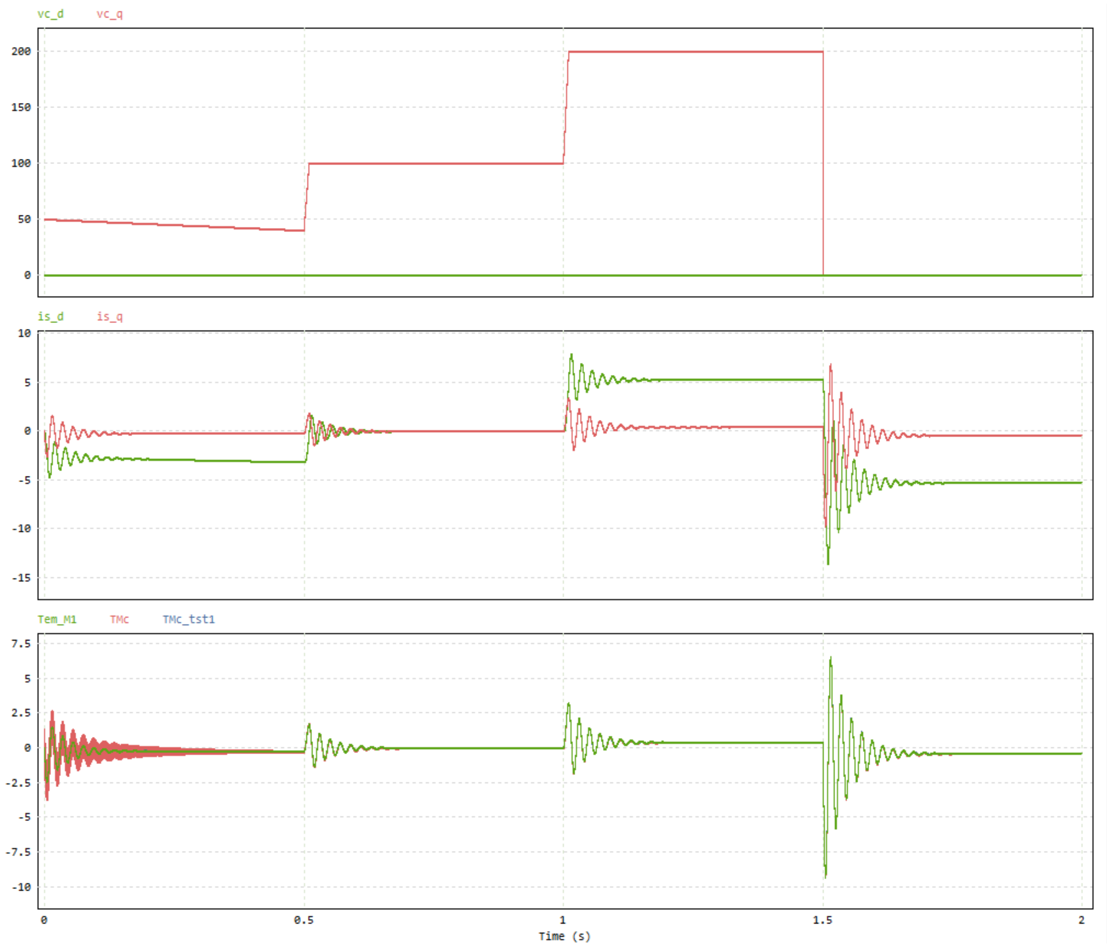
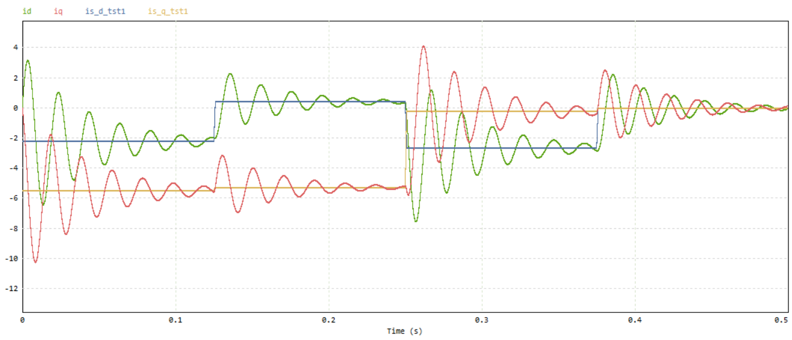


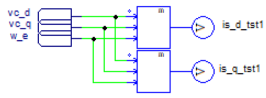
Figure B.14: Effect of $v_{c,q}$ stepping on (i_d, i_q) and electromagnetic torque.

Figure B.14 shows that stepping $v_{c,q}$ produces a dominant change in i_d, again highlighting cross-coupling between dq channels. As before, torque follows i_q, while i_d variations mainly reflect flux-axis dynamics rather than direct torque production. Switching ripple is visible in the currents due to PWM excitation and appears as high-frequency ripple superimposed on the average torque.

Overall, the results validate: (i) end-to-end consistency of dq↔abc transformations, (ii) correct SVPWM voltage synthesis from the DC link, and (iii) physically consistent coupled voltage–current–torque behavior under imposed speed and open-loop excitation.



(a) Simulated currents (i_d, i_q) vs analytical steady-state prediction $(i_{d,tst1}, i_{q,tst1})$.



(b) SVPWM modulation block used for voltage synthesis.

Figure B.15: Analytical steady-state current validation and SVPWM implementation.

The analytical steady-state model solves the dq voltage equations under the assumption $\dot{i}_d = \dot{i}_q = 0$, yielding predicted currents $(i_{d,tst1}, i_{q,tst1})$ as functions of $(v_{c,d}, v_{c,q}, \omega_e)$. The close agreement between simulated and analytical currents confirms correct implementation of machine parameters, dq cross-coupling, and voltage-driven dynamics. Minor differences are attributed to switching ripple and transient effects, which are not included in the steady-state model.

The observed dq cross-coupling under imposed speed confirms the expected open-loop voltage-driven dynamics of the PMSM. The model now represents a physically consistent voltage-fed drive and establishes the foundation for implementing decoupled, closed-loop current control in subsequent iterations

B.4 Voltage Limiting, Anti-Windup and Non-Ideal Inverter

Objective:

Introducing physical voltage constraints into the current control structure and to prevent integrator wind-up under saturation conditions. Although the current loops were already tracking their references with satisfactory dynamic performance, the modulation signals (m_a, m_b, m_c) were exceeding ± 1, clearly indicating that the controller was demanding voltage levels beyond what the DC-link and inverter could physically deliver.

This issue becomes critical once the imposed mechanical speed source is removed. In a realistic drive system, acceleration dynamics and back-EMF variation introduce large transient voltage demands during speed or load steps. Without voltage limiting, the PI regulators request infeasible voltage vectors. The inverter saturates implicitly, while the PI integrators continue accumulating error, leading to overshoot, slow recovery, and possible instability.

Therefore, two tightly coupled mechanisms were introduced:

Key additions:

- A norm-based dq voltage vector limiter placed after the current controllers.
- A back-calculation anti-windup loop embedded in each PI current regulator.

- Replacement of the ideal voltage source inverter by a non-ideal VSC including switching devices and DC-link dynamics.

Together, these modifications enforce actuator feasibility, preserve controller internal consistency during saturation, and increase simulation realism before proceeding to fully dynamic mechanical operation.

B.4.1 PSIM Schematic

1. Voltage Limiter Subsystem

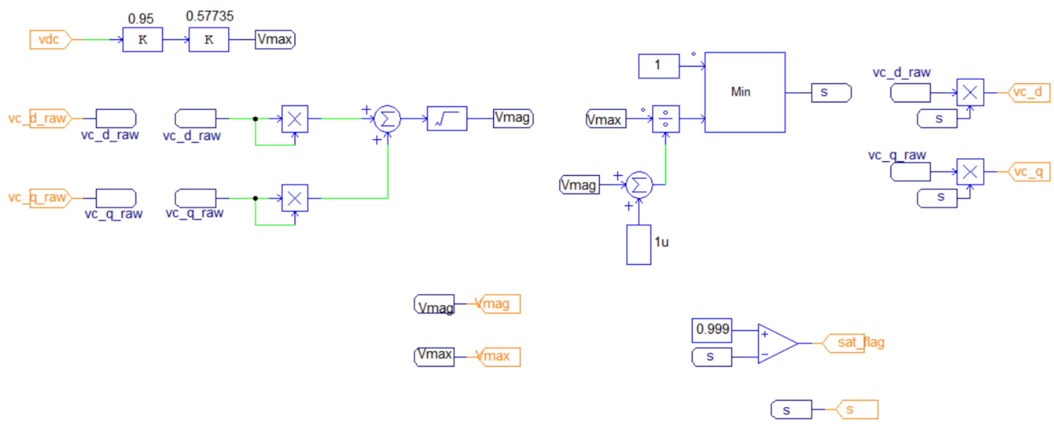


Figure B.16: dq Voltage Vector Limiter Subcircuit

The limiter operates on the raw voltage commands $v_{c,d,raw}$ and $v_{c,q,raw}$, which already include proportional action, integral action, and decoupling/feedforward compensation.

First, the Euclidean norm of the commanded voltage vector is computed:

$$V_{mag} = \sqrt{v_{d,raw}^2 + v_{q,raw}^2}$$

Voltage limits in an inverter are vector limits, not independent axis limits. Therefore, magnitude limiting preserves the direction of the voltage vector while constraining its amplitude.

The maximum allowable magnitude is computed from the DC-link voltage:

$$V_{max} = 0.95 \cdot \frac{V_{dc}}{\sqrt{3}}$$

The constant $1/\sqrt{3} = 0.57735$ corresponds to the linear modulation limit of SVPWM. The factor 0.95 provides a safety margin to avoid numerical edge effects and unintentional overmodulation.

The scaling factor is then computed as:

$$S = \min\left(1, \frac{V_{max}}{V_{mag} + \varepsilon}\right)$$

where ε prevents division by zero.

If $V_{mag} \leq V_{max}$, then $S = 1$ and no limiting occurs. If $V_{mag} > V_{max}$, then $S < 1$ and the vector is proportionally scaled:

$$v_d = S \cdot v_{d,raw}, \qquad v_q = S \cdot v_{q,raw}$$

This method ensures that the voltage vector remains inside the achievable circle defined by the inverter, preserving vector orientation and decoupling structure.

The signal `sat_flag` is derived from S and provides a diagnostic indication of active saturation.

2. Anti-Windup Integration in the PI Controllers

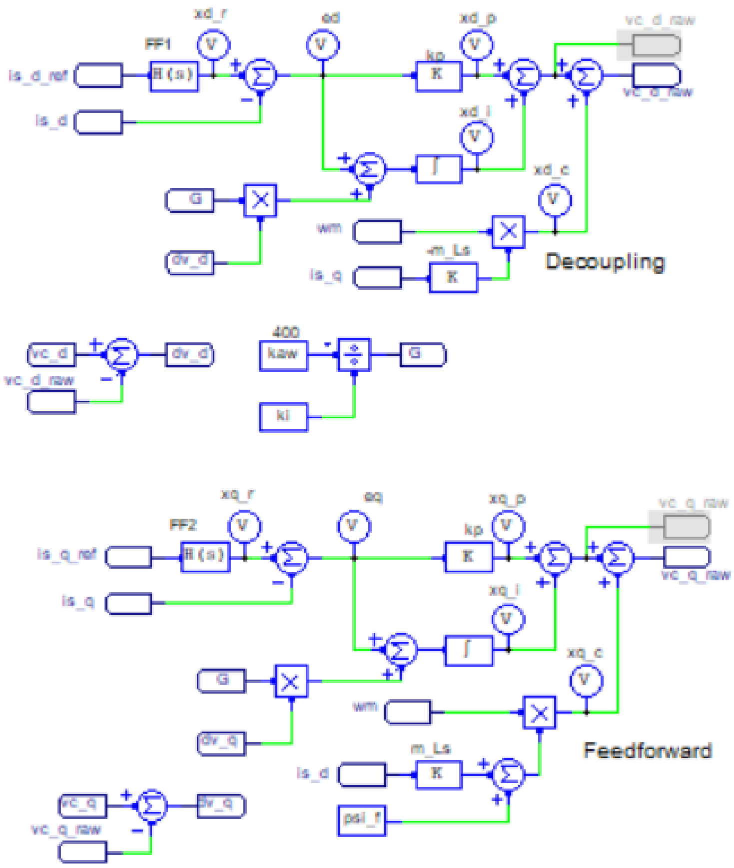


Figure B.17: Iteration 5: PI Current Controllers with Back-Calculation Anti-Windup

Each current regulator implements a PI controller with decoupling:

$$v_{d,raw} = K_p e_d + x_{d,i} + v_{d,dec}$$

where $x_{d,i}$ is the integrator state.

The anti-windup mechanism is based on back-calculation. The difference between limited and raw voltages is computed:

$$\Delta v_d = v_d - v_{d,raw}$$

The integrator dynamics become:

$$\dot{x}_{d,i} = K_i e_d + K_{aw}(v_d - v_{d,raw})$$

In linear operation ($v_d = v_{d,raw}$), the correction term is zero and the controller behaves as a standard PI.

Under saturation, $v_d \neq v_{d,raw}$ and the second term counteracts integrator growth. If the controller requests more voltage than physically possible, the injected correction drives the integrator toward a consistent state, preventing wind-up.

The same structure is implemented identically for the q-axis.

This guarantees smooth transition between linear and saturated regimes and avoids large transient overshoot when saturation clears.

3. Non-Ideal Voltage Source Converter

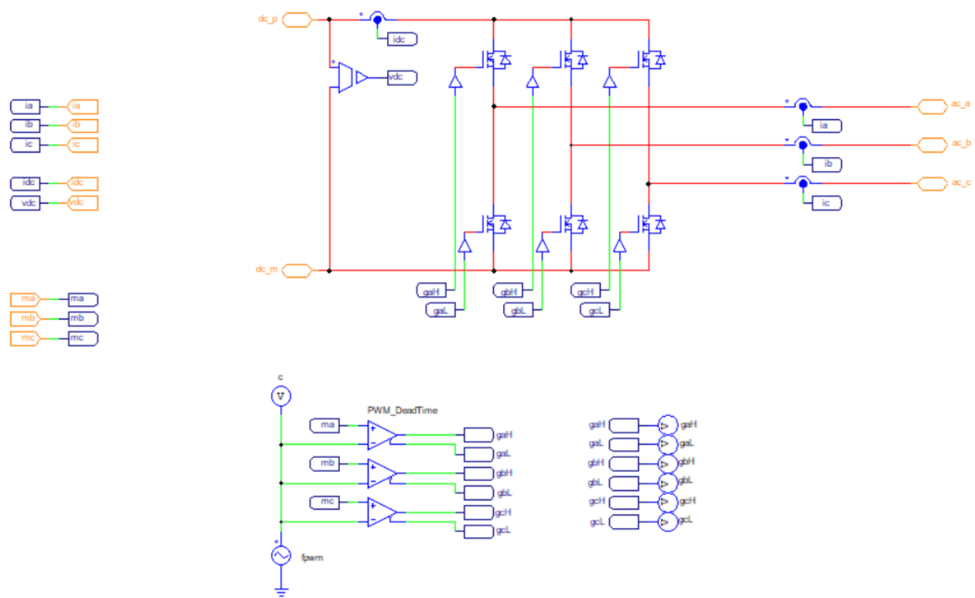


Figure B.18: Non-Ideal Three-Phase Voltage Source Converter

To increase physical realism, the ideal voltage source was replaced by a switching-based three-phase VSC including:

- Six semiconductor switches with anti-parallel diodes.
- Complementary gate signals with dead-time insertion.
- Explicit DC-link dynamics (V_{dc} measurement and current flow).
- Phase current measurement points.

The modulation signals (m_a, m_b, m_c) are converted to PWM signals, which drive complementary high-side and low-side switches. Dead-time insertion ensures non-overlapping conduction, preventing shoot-through.

Unlike an ideal averaged inverter model, this implementation introduces:

- Switching ripple,
- Dead-time induced distortion,
- Finite DC-link constraints,
- Realistic interaction between commanded voltages and applied phase voltages.

This non-ideal VSC ensures that the voltage limiter and anti-windup mechanisms are validated against physically realizable inverter behavior rather than ideal algebraic voltage injection.

With this the total implementation is replicated in C attending to all the details above mentined leaving only the speed PID loop not implemented since it is no longer a meter of the FOC logic it self and can just be a generic implementation.:

```
static double wMc_rpm_last;
static double wMc_acc_pu;

double wMc_rpm;
double vab, vbc, va, vb, vc, valpha, vbeta, vd, vq;

double ia, ib, ic,  ialpha, ibeta, id, iq;

double vc_d, vc_q;
double vc_alpha, vc_beta;
double vc_a, vc_b, vc_c;
double vdc;
double ma_tst, mb_tst, mc_tst;

double thetaMc_pu;
```

```
double wm;
double thetam_pu;
double thetam;
double psi_f;
double Tem_tst1;
double pelec;

// --- Voltage limiting / anti-windup ---
double vc_d_raw, vc_q_raw;
double Vmag, Vmax, S;
double dv_d, dv_q;
double kaw = 400.0;       // using PSIM value
double eps = 1e-6;        // like 1u block

// --- Static variables (Integrators must be static to persist) ---
static double xd_i = 0.0;
static double xq_i = 0.0;

// variables for PI Control
double id_ref, iq_ref, ed, eq, kp, ki;
double vPI_d, vPI_q;

// Constants (matching  motor parameters)
double m_Rs_tst = 1.5;
double m_Ls_tst = 0.060;

double tmp;
double tmpcos;
double tmpsin;

#define pi 3.1415926535897932385
#define _2pi 6.2831853071795864770

//------------------INPUTS-----------------------------------
wMc_rpm = x1;
vab = x2;
vbc = x3;
ia = x4;
ib = x5;
psi_f = x6;
```

```
vdc     = x7;
id_ref  = x8;
iq_ref  = x9;

//----------------------------------------------------------------------
wMc_acc_pu = wMc_acc_pu + (wMc_rpm+wMc_rpm_last) * delt / (2*60.0);
         if ( wMc_acc_pu > 1 ) wMc_acc_pu = wMc_acc_pu - 1;
else if ( wMc_acc_pu < 0 ) wMc_acc_pu = wMc_acc_pu + 1;
wMc_rpm_last = wMc_rpm;

thetaMc_pu = wMc_acc_pu;

//----------------------------------------------------------------------
wm = wMc_rpm * _2pi / 60 * m_p;
tmp = thetaMc_pu*m_p;

//thetam_pu = tmp - ceil(tmp) + 1;
thetam_pu = tmp - floor(tmp);   // gives [0,1)

thetam = thetam_pu * _2pi;

//----------------------------------------------------------------------
va = (2*vab +   vbc)/3;
vb = (  -vab +   vbc)/3;
vc = (  -vab - 2*vbc)/3;
ic = -(ia + ib);

tmpcos = cos(thetam);
tmpsin = sin(thetam);

//valpha = sqrt(6)/2*va;
//vbeta = (sqrt(2)/2)*va + sqrt(2)*vb;
//vd = tmpcos*valpha + tmpsin*vbeta;
//vq = -tmpsin*valpha +tmpcos*vbeta;

ialpha = sqrt(6)/2 * ia;
ibeta  = (sqrt(2)/2)*ia + sqrt(2)*ib;
id =  tmpcos*ialpha + tmpsin*ibeta;
iq = -tmpsin*ialpha + tmpcos*ibeta;
```

```
// ------------- PI GAIN CALCULATION --------------
{
    double Tstl = 0.005; // 5ms
    double Mp   = 1.0;   // 1%
    double eW   = 1.0;   // 1%

    double lnMp = log(Mp/100.0);
    double zeta = -lnMp / sqrt(lnMp*lnMp + pi*pi);
    double lnEW = log(eW/100.0);
    double wn   = -lnEW / (Tstl * zeta);

    ki = wn * wn * m_Ls_tst;
    kp = 2.0 * zeta * wn * m_Ls_tst - m_Rs_tst;
}
// ------------------ CURRENT CONTROL (PI + decoupling + limiter + AW) ---------
// Error signals
ed = id_ref - id;
eq = iq_ref - iq;

// PI output using previous integrator state
vPI_d = (kp * ed) + xd_i;
vPI_q = (kp * eq) + xq_i;

// Raw dq voltage request (PI + decoupling/feedforward)
vc_d_raw = vPI_d - (wm * m_Ls_tst * iq);
vc_q_raw = vPI_q + (wm * (id * m_Ls_tst + psi_f));

// Voltage vector limiter (SVPWM linear region)
Vmag = sqrt(vc_d_raw*vc_d_raw + vc_q_raw*vc_q_raw);
Vmax = 0.95 * vdc * 0.5773502691896258; // 1/sqrt(3)

if (Vmag > eps) {
    S = Vmax / (Vmag + eps);
    if (S > 1.0) S = 1.0;
    if (S < 0.0) S = 0.0;
} else {
    S = 1.0;
}

vc_d = S * vc_d_raw;
```

```
vc_q = S * vc_q_raw;

// Back-calculation anti-windup (inject limiter error)
dv_d = vc_d - vc_d_raw;
dv_q = vc_q - vc_q_raw;

xd_i += (ki * ed + kaw * dv_d) * delt;
xq_i += (ki * eq + kaw * dv_q) * delt;

// ------------------ DECOUPLING (Per PSIM Schematic) ----------
// D-axis: vc_d = vPI_d + (w_e * -m_Ls * iq_f)
//vc_d = vPI_d - (wm * m_Ls_tst * iq);

// Q-axis: vc_q = vPI_q + (w_e * (id_f * m_Ls + psi_f))
//vc_q = vPI_q + (wm * (id * m_Ls_tst + psi_f));
//---------------------------------------------------------------

Tem_tst1 = m_p * psi_f * iq;
pelec = va*ia + vb*ib + vc*ic;   // instantaneous 3-phase power

//---------------------------------------------------------------
// --- Command path replication (dq -> alpha/beta -> abc) ---
vc_alpha =  tmpcos*vc_d - tmpsin*vc_q;
vc_beta  =  tmpsin*vc_d + tmpcos*vc_q;

// Inverse of custom Clarke mapping
vc_a = (2.0/sqrt(6.0)) * vc_alpha;
vc_b = (1.0/sqrt(2.0)) * vc_beta - 0.5 * vc_a;
vc_c = -(vc_a + vc_b);

// Optional: linear utilization estimate (NOT full SVPWM)
if (vdc > 1e-6) {
  ma_tst = 2.0 * vc_a / vdc;
  mb_tst = 2.0 * vc_b / vdc;
  mc_tst = 2.0 * vc_c / vdc;
} else {
  ma_tst = mb_tst = mc_tst = 0.0;
}
//------------------OUTPUTS---------------------------
y1 = wMc_acc_pu;
```

```
y2 = wm;
y3 = thetam_pu;
y4 = thetam;

y5 = va;
y6 = vb;
y7 = vc;
y8 = valpha;
y9 = vbeta;
y10 = vd;
y11 = vq;
y12 = ic;
y13 = ialpha;
y14 = ibeta;
y15 = id;
y16 = iq;
y17 = Tem_tst1;
y18 = pelec;

y19 = vc_alpha;
y20 = vc_beta;
y21 = vc_a;
y22 = vc_b;
y23 = vc_c;
y24 = ma_tst;
y25 = mb_tst;
y26 = mc_tst;
```

B.5 Iteration 6 — Speed Loop Closure and Position Control Extension

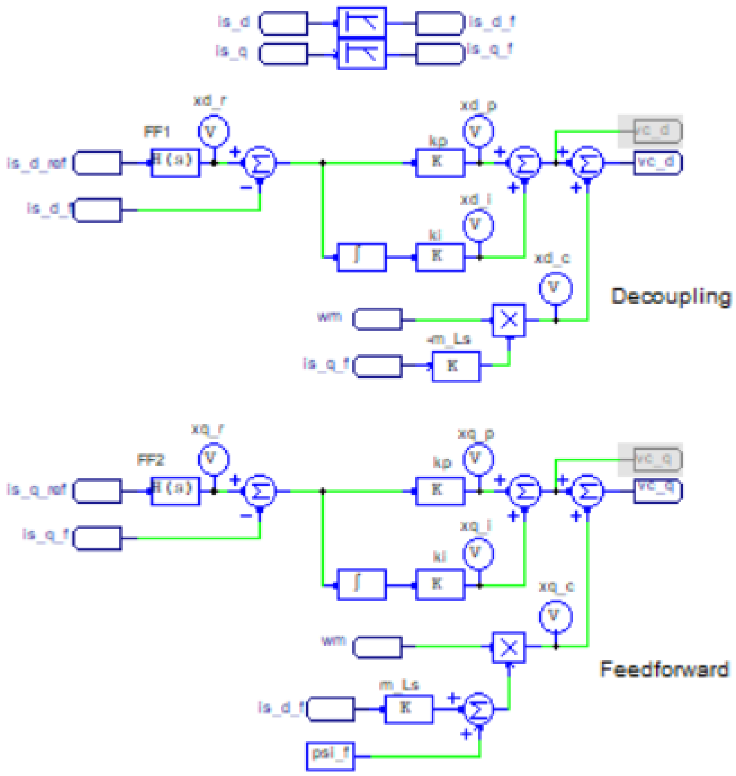


Figure B.19: Iteration 4: Closed-loop dq current control with PI regulators and decoupling compensation.

Figure B.19 shows the implementation of the synchronous dq current controller for a surface-mounted PMSM ($L_d = L_q = L$).

Control Structure Overview

The controller is implemented independently for the d and q axes, following the machine model in the synchronous reference frame:

$$v_d = R_s i_d + L\frac{di_d}{dt} - \omega_e L i_q \tag{B.9}$$

$$v_q = R_s i_q + L\frac{di_q}{dt} + \omega_e L i_d + \omega_e \psi_f \tag{B.10}$$

These equations show that the axes are dynamically coupled through the speed-dependent cross terms. Without compensation, the d and q current loops would interfere with each other.

d-Axis Loop

The d-axis error is computed as

$$e_d = i_{d,ref} - i_d.$$

This error is processed by a PI regulator:

$$v_{PI,d} = K_p e_d + K_i \int e_d dt.$$

To cancel the cross-coupling term $-\omega_e L i_q$, feedforward compensation is added:

$$v_{c,d} = v_{PI,d} + \omega_e L i_q.$$

In the schematic, this is implemented as:

- Subtraction block generating e_d
- Proportional gain block K_p
- Integrator with gain K_i
- Multiplication block $\omega_e \cdot L \cdot i_q$
- Summation of PI output and compensation term

This structure effectively linearizes the d-axis plant into a first-order RL system.

q-Axis Loop

The q-axis error is computed as

$$e_q = i_{q,ref} - i_q.$$

The PI controller is

$$v_{PI,q} = K_p e_q + K_i \int e_q dt.$$

The compensation terms cancel both cross-coupling and back-EMF:

$$v_{c,q} = v_{PI,q} - \omega_e L i_d - \omega_e \psi_f.$$

In the schematic this corresponds to:

- Error calculation block
- Proportional and integral paths
- Multiplication block $\omega_e L i_d$

- Multiplication block $\omega_e \psi_f$
- Final summation producing $v_{c,q}$

This compensation removes the speed-dependent disturbance introduced by the permanent magnet flux linkage.

Position Reference Tracking

The outer position loop is validated by applying step changes in position reference, corresponding to multiple mechanical revolutions.

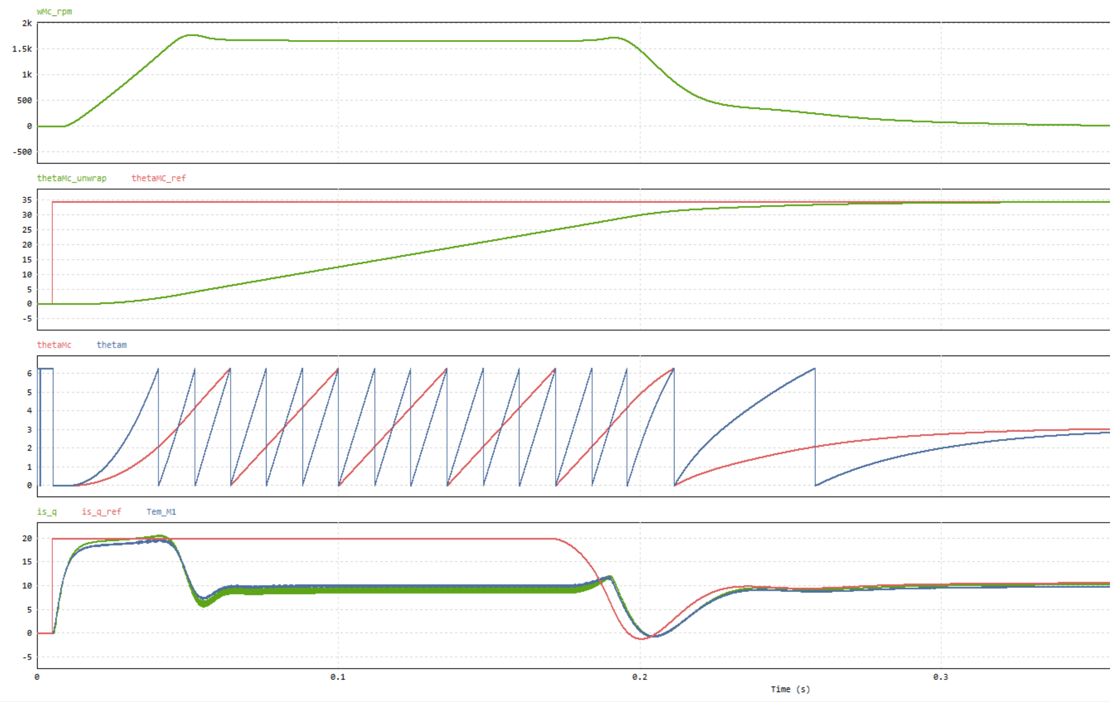


Figure B.20: Position reference tracking: mechanical angle, speed, and i_q.

The mechanical angle tracks the commanded reference accurately. The position error generates a speed reference through the proportional controller, which is then regulated by the speed PI.

During large position steps:

- The speed rises according to the generated ω_{ref}.
- The q-axis current produces the required acceleration torque.
- The system decelerates smoothly as the target position is approached.

No steady-state position offset is observed. The nested cascade behaves consistently, with smooth transition between acceleration and deceleration phases.

PI Parameter Design

The PI gains are not selected heuristically. They are derived from a desired second-order closed-loop current response specification.

The design parameters are:

- Settling time: T_{stl}
- Maximum overshoot: M_p
- Steady-state error band: e_W

From these, the damping ratio ζ and natural frequency ω_n are computed as:

$$\zeta = \frac{-\ln\left(\frac{M_p}{100}\right)}{\sqrt{\ln^2\left(\frac{M_p}{100}\right)+\pi^2}},$$

$$\omega_n = \frac{-\ln\left(\frac{e_W}{100}\right)}{T_{stl}\,\zeta}.$$

Assuming the decoupled plant reduces to

$$G(s) = \frac{1}{Ls+R_s},$$

the PI gains that produce the desired second-order response are:

$$K_i = \omega_n^2 L,$$

$$K_p = 2\zeta\omega_n L - R_s.$$

These expressions result from matching the closed-loop characteristic polynomial to the standard second-order form:

$$s^2 + 2\zeta\omega_n s + \omega_n^2.$$

This ensures predictable transient behavior, controlled damping, and systematic tuning instead of empirical gain adjustment.

Appendix C

Parameter estimation

This appendix contains the full, unaltered C-block code used in the Parameter estimation of the PSIM block.

```
static double id_prev = 0.0;
static double did_dt_f = 0.0;
static double Ld_avg = 0.0;
static int valid_count = 0;

double vd3;
double id3, iq3, we3, Rs3, en3;
double did_dt;
double Ld_raw;
double valid;

// thresholds
double eps_did = 1e-3;
double eps_iq  = 0.2;
double eps_we  = 1e-2;
// minimum applied vd to consider estimation valid
double eps_vd  = 1.0;
double id_max_est = 3.0;  // only estimate during early transient

// Inputs
vd3 = x1;
id3 = x2;
iq3 = x3;
we3 = x4;
Rs3 = x5;
en3 = x6;
```

```
// raw derivative
did_dt = (id3 - id_prev) / delt;

// simple low-pass filter on derivative
did_dt_f = did_dt_f + 0.1 * (did_dt - did_dt_f);

// default outputs
Ld_raw = 0.0;
valid = 0.0;

// Valid only during pulse transient and locked-rotor condition
if (en3 > 0.5) {
    if ((fabs(vd3) > eps_vd) &&
        (fabs(did_dt_f) > eps_did) &&
        (fabs(iq3) < eps_iq) &&
        (fabs(we3) < eps_we) &&
        (fabs(id3) < id_max_est)) {

        Ld_raw = (vd3 - Rs3 * id3) / did_dt_f;

        // reject non-physical values
        if ((Ld_raw > 0.001) && (Ld_raw < 0.2)) {
            valid = 1.0;

            valid_count++;
            if (valid_count == 1) {
                Ld_avg = Ld_raw;
            } else {
                Ld_avg = Ld_avg + (Ld_raw - Ld_avg) / valid_count;
            }
        }
    }
}

// Save previous current
id_prev = id3;

// Outputs
y1 = did_dt_f;
```

```
y2 = Ld_raw;
y3 = Ld_avg;
y4 = valid;
```