People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Echahid Hamma Lakhdar University – El Oued
Faculty Of Science and Technology
Department Of Electrical Engineering
Master's Program: Electrical Control

MASTER'S THESIS

Submitted in Partial Fulfillment of the Requirements for the Degree of
MASTER'S DEGREE IN ELECTRICAL ENGINEERING
Specialization: ELECTRICAL CONTROL

Thesis Topic

Sensorless Field-Oriented Control of PMSM Drives Using Backstepping Control and MVT-Based Sector Nonlinearity Observers

Presented by:
Messioughi Nour-Eddine

Pr. Allag Abdelkrim	**Président**
Dr. Hamidani Bilal	**Examinateur**
Dr. Allag Meriem	**Encadreur**
Dr. Allag Nassiba	**Encadreur**

Academic year 2025-2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor **Pr. Allag Abdelkarim** for his valuable guidance, constructive feedback, and continuous support throughout the development of this work.

I also wish to thank the members of the examination committee for their time, interest, and valuable remarks.

My sincere thanks go to the professors of the Department of Electrical Engineering for the knowledge and support provided during my Master's studies.

Finally, I am deeply grateful to my family for their patience, encouragement, and unwavering support throughout my academic journey.

Dedication

I dedicate this work

To my beloved parents,

for their endless love, prayers, sacrifices, and unwavering support.

Their encouragement has always been a source of strength and inspiration throughout my journey.

To my dear family,

for their patience, understanding, and constant support during the challenging moments of this work.

To my beloved wife,

for her love, patience, encouragement, and unconditional support.

Thank you for standing beside me throughout this journey and for always believing in me.

To my adorable daughters, Layane and Nourhane,

you are the joy and light of my life.

Your smiles, love, and innocence have given me strength and motivation whenever I needed them most.

This achievement is dedicated to you, with all my love.

To all those who supported, encouraged, and believed in me,

I dedicate this work with sincere gratitude and love.

Abstract

The high-performance control of Permanent Magnet Synchronous Motors (PMSM) is frequently challenged by the motor's nonlinear dynamics, parameter variations, and the limitations of traditional linear control techniques. This thesis presents a comprehensive sensorless control strategy designed to overcome these hurdles by combining Field Oriented Control (FOC) with advanced nonlinear control and estimation theories.

The proposed approach utilizes a Backstepping controller to ensure robust tracking performance and global stability, designed through a recursive Lyapunov-based methodology. To facilitate sensorless operation, a Nonlinear Observer based on the Mean Value Theorem (MVT) is developed to accurately estimate the rotor's position and speed, effectively handling the system's nonlinearities without the need for mechanical encoders. This control suite is implemented via Space Vector Pulse Width Modulation (SVPWM) to optimize inverter efficiency and minimize harmonic distortion.

Methodologically, the stability of the entire closed-loop system is mathematically proven, and the performance is validated through rigorous testing. Results indicate that the proposed combination offers superior dynamic response, enhanced disturbance rejection, and high estimation precision compared to standard PI-based FOC systems. Ultimately, this research demonstrates that the synergy of backstepping control and MVT-based observation provides a robust, high-efficiency solution for modern industrial PMSM applications.

Keywords

Permanent Magnet Synchronous Motor (PMSM), Sensorless Control, Field-Oriented Control (FOC), Backstepping Control, Nonlinear Observer, Mean Value Theorem (MVT), Lyapunov Function, Space Vector Pulse Width Modulation (SVPWM), Rotor Position and Speed Estimation, Nonlinear Control.

الملخص

غالبًا ما يواجه التحكم عالي الأداء في المحركات المتزامنة ذات المغناطيس الدائم (PMSM) تحديات ناجمة عن الديناميكيات غير الخطية للمحرك، وتغيرات المعلمات، ومحدودية تقنيات التحكم الخطية التقليدية. تقدم هذه الأطروحة استراتيجية شاملة للتحكم **دون مستشعرات،** تهدف إلى التغلب على هذه التحديات من خلال الجمع بين **التحكم بتوجيه المجال (FOC)** ونظريات التحكم والتقدير اللاخطي المتقدمة.

تعتمد المنهجية المقترحة على **متحكم Backstepping** لضمان أداء متين في تتبع المراجع والاستقرار، وقد تم تصميمه باستخدام منهجية تكرارية قائمة على **دالة ليابونوف** .ولتحقيق التشغيل دون مستشعرات، تم تطوير **مراقب لاخطي قائم على نظرية القيمة المتوسطة (MVT)** من أجل التقدير الدقيق لموضع وسرعة الدوار، مع التعامل بفعالية مع لاخطية النظام دون الحاجة إلى مشفرات ميكانيكية. ويتم تنفيذ منظومة التحكم هذه باستخدام **تضمين عرض النبضة بمتجه الفضاء (SVPWM)**بهدف تحسين كفاءة العاكس وتقليل التشوه التوافقي.

من الناحية المنهجية، تم إثبات استقرار النظام ذي الحلقة المغلقة بالكامل رياضيًا، كما تم التحقق من أداء المنظومة من خلال اختبارات صارمة. وتبين النتائج أن الجمع المقترح يوفر **استجابة ديناميكية محسنة، وقدرة أفضل على رفض الاضطرابات، ودقة عالية في التقدير** مقارنة بأنظمة التحكم FOC التقليدية القائمة على متحكمات .PI وفي النهاية، يوضح هذا البحث أن التكامل بين **تحكم Backstepping والمراقبة القائمة على MVT** يوفر حلًا متينًا وعالي الكفاءة لتطبيقات محركات PMSM الحديثة في المجالات الصناعية.

الكلمات المفتاحية:

المحركات المتزامنة ذات المغناطيس الدائم (PMSM)، التحكم دون مستشعرات، التحكم بتوجيه المجال (FOC)، تحكم Backstepping، المراقب اللاخطي، نظرية القيمة المتوسطة (MVT)، دالة ليابونوف، تضمين عرض النبضة بمتجه الفضاء (SVPWM)، تقدير موضع وسرعة الدوار، التحكم اللاخطي.

List of Figures

Figure 2-1: permanent magnet synchronous motor 17
Figure 2-*2*: permanent magnet motor structure 18
Figure 3-1: simplified equivalent circuit of PMSM 25
Figure 3-2: Transformation from stationary to rotational reference frame 26
Figure 3-3: diagram of the PMSM in rotational $\boldsymbol{dq}$ frame 28
Figure 4-1: Sensorless FOC for PMSM 32
Figure 4-2: Mean Value Theorem definition 35
Figure 5-1: modulation and carrier signal in PWM by triangle comparison for space vector modulation and the reference 58
Figure 5-*2*: three-phase inverter diagram 59
Figure 5-3: Representation of tow-level three-phase inverter voltage space vector . in complex \alpha\beta plan [39]. 65
Figure 5-4: pulse pattern for sector 1 77
Figure 5-5: simulation of SVPWM generated by STM32 applied onto the inverter 85
Figure 5-6: shows the control system diagram of PMSM using Backstepping Controller and MVT-NSL based Observer 86
Figure 5-7: Simulation result showing the Three-Phase Currents, the Rotor Speed 86
Figure 5-8: shows the quadratic component current $\boldsymbol{iq}$, and the direct component current $\boldsymbol{id}$ 87
Figure 5-9: shows the error observer convergence 87
Figure 6-1 : DC resistance measurement 91
Figure 6-2: measurement voltage-current characteristic 91
Figure 6-*3*: Experimental d-axis test 93
Figure 6-*4*: measured d-axis current response 93
Figure 6-*5*: experimental setup Measured PMSM back-EMF waveform 94
Figure 6-*6*: No load back-EMF 94
Figure 6-*7*: back-EMF amplitude versus rotor speed 94
Figure 6-*8*: Measured rotor-speed decay 96
Figure 6-9:electromagnetic torque versus rotor speed, the slop of the fitted characteristic provides an experimental estimat3 of friction B 97
Figure 6-10: three-phase inverter diagram with IR 2110 MOSFET driers 100
Figure 6-11: the IR2110 Integrated circuit driver 101
Figure 6-12: show IR2110 ICs connection for driving high and low MOSFET 101
Figure 6-13: shows how to bootstrap of IR2110 IC MOSFET's driver. 102

Figure 6-14: shows how to use the web-based bootstrap calculator to determine the required C_B recharge current at MAX and MIN duty cycle, and provides the information required to determine the best fit bootstrap module 103
Figure 6-15: shows the bootstrap of the MOSFET driver using the capacitor and diode 104
Figure 6-16: shows the RC snubber circuit for the MOSFET 105
Figure 6-17: shows how to create a dead time within the switching interval of the same leg inverter 105
Figure 6-18: the output voltages and currents of three-phase inverter fed the PMSM 106
Figure 6-18: three- phase inverter PCB using for controlling PMSM motor 107
Figure 6-20: STM32 F407 microcontroller 108
Figure 6-21: STM32- based control architecture 110
Figure 6-22: General Software Architecture of Real Time PMSM Control 110
Figure 6-23: Complete Real Time Execution Sequence 111

List of Table

Table 5-1: comparison between SVPWM and SPWM [35]. 58
Table 5-2:Switching states of a two-level three-phase VSI 60
Table 5-3: how the six active voltage vectors divide the stationary reference frame into six identical sectors.[47] 67
Table 5-4: shows the active tow vectors that surrounding the reference vector. 69
Table 5-5: the duty cycle expressions of all sectors 77
Table 5-6: SVPWM sectors. 79
Table 5-7: the duty cycle of each phase. 83

Table of contents

Abstract **1**

الملخص 2

List of Figures **3**

List of Table **5**

Table of contents **6**

Nomenclature **9**

General Introduction **13**

Chapter1: Permanent Magnet Synchronous Motor **16**

1. Introduction: 17

2. What is a Permanent Magnet Synchronous Motor? 17

2.1. Structure of a PMSM 18

2.2. How Does a Permanent Magnet Synchronous Motor Work? 18

3. Types of Permanent Magnet Synchronous Motors 19

3.1. Surface-Mounted PMSM (SPMSM) 19

3.2. Interior PMSM (IPMSM): 20

4. Advantages of Permanent Magnet Synchronous Motors: 20

5. Disadvantages of Permanent Magnet Synchronous Motors: 21

6. PMSM applications: 21

7. Conclusion 22

Chapter2: Modeling Permanent Magnet Synchronous Motor **23**

1. Modeling the PMSM in the $\boldsymbol{dq}$ frame: 24

1.1. Electrical dynamics in the $\boldsymbol{dq}$ frame: 25

1.2. Electromagnetic torque equation: 27

1.3. Mechanical dynamics: 27

2. Complete nonlinear state-space model: 29

3. Nonlinear and uncertainties: 29

Chapter 3: Sensorless FOC control **30**

1. Control methods: 31

2. FOC (Field Oriented Control) control: 31

3. FOC structure: 31
4. Motivation for sensorless control: 33
5. Mean Value Theorem (MVT) Method in Nonlinear Control and Sensorless Observer Design: 34
5.1. Definition: 34
5.2. Application of MVT in Nonlinear Observer Design: 35
6. Importance of MVT in Sensorless Control: 36
7. Advantages of the MVT Approach 36
8. Build MVT-Nonlinear Sector (NLS) observer for PMSM motor: 37
9. Nonlinear Sector (NLS) transformation Design for PMSM: 41
9.1. Identification of Sector Variables & Physical Bounds: 41
9.2. Substitution into the MVT Matrix: 42
9.3. Stability analysis: 43
9.4. Numerical implementation of the observer: 44

Chapter 4: Backstepping Controller 46

1. Backstepping Controller: 47
1.1. Defined the speed tracking errors: 47
1.2. Define $\boldsymbol{eiq}$the $\boldsymbol{q}$-axis tracking current error: 49
1.3. Define $\boldsymbol{eid}$the $\boldsymbol{d}$-axis tracking current error: 51

Chapter 5: Space Vector PULS Width Modulation SVPWM 56

SVPWM (Space Vector Pulse Width Modulation): 57
1. Introduction: 57
2. Small Comparison with Classical SPWM: 58
3. Two-Level Voltage Source Inverter (VSI) and Switching States: 58
4. Mathematical Representation of the Space Voltage Vectors: 61
5. Voltage Vector Synthesis and Derivation of the Switching Times: 70
6. Average Voltage Principle:[48] 71
7. General Expressions for All Sectors:[48] 73
8. Physical Interpretation of the Switching Times: 73
9. Relationship with the Proposed PMSM Drive: 74
10. Duty Cycle Computation and Symmetrical Switching Sequence: 74
11. Symmetrical Distribution of the Zero Vector:[49] 74

12. Symmetrical Switching Sequence: 75
13. Advantages of the Symmetrical Switching Sequence: 78
14. Implementation in the Proposed PMSM Drive: 78
Conclusion 87

Chapter 6: Practical Design: 88

1. Parameter's Identification: 89
1.1. Laboratory Test Bench 90
1.2. The power source (three-phase inverter): 98

2. Inverter Implementation and STM32 Configuration (Programming): 107
2.1. Realized inverter board: 107
2.2. STM32 microcontroller: 107
2.3. STM32 timer peripherals: 108
2.4. Software Architecture of the Real-Time Control System: 110
2.5. Complete Real-Time Control Algorithm: 110

General Conclusion 112
References Bibliographies 116
ANNEXES 122

Nomenclature

Symbol	Description	Unit
(v_a, v_b, v_c)	Three-phase stator voltages	V
(i_a, i_b, i_c)	Three-phase stator currents	A
(v_α, v_β)	Stator voltage components in the (\alpha\beta) frame	V
(i_α, i_β)	Stator current components in the (\alpha\beta) frame	A
(v_d, v_q)	(d)- and (q)-axis stator voltages	V
(i_d, i_q)	(d)- and (q)-axis stator currents	A
(i_{dref})	Reference (d)-axis current	A
(i_{qref})	Reference (q)-axis current	A
(R_s)	Stator resistance	Ω
(L_d)	(d)-axis inductance	H
(L_q)	(q)-axis inductance	H
(Ψ_{pm})	Permanent-magnet flux linkage	Wb
(Ψ_d, Ψ_q)	(d)- and (q)-axis stator flux linkages	Wb
(T_e)	Electromagnetic torque	N·m
(T_L)	Load torque	N·m
(p)	Number of pole pairs	—
(ω_m)	Mechanical angular speed	rad/s
(ω_e)	Electrical angular speed	rad/s
(θ_m)	Mechanical rotor position	rad
(θ_e)	Electrical rotor position	rad
(f_e)	Electrical frequency	Hz

Electrical And PMSM Quantities

Symbol	Description	Unit
(J)	Rotor and load equivalent inertia	kg·m²
(B)	Viscous friction coefficient	N·m·s/rad
(ω_{mref})	Reference mechanical speed	rad/s
(e_ω)	Speed tracking error	rad/s
(e_1)	First backstepping tracking error	—
(e_2)	Second backstepping error	—

Mechanical Quantities

Symbol	Description	Unit
$(T_C larke)$	Clarke transformation matrix	—
$(T_P ark)$	Park transformation matrix	—
(α,β)	Stationary reference-frame axes	—
(d, q)	Rotating reference-frame axes	—
(θ_e)	Electrical rotor angle used in the Park transformation	rad

Coordinate Transformation

Backstepping Controller

Symbol	Description	Unit
(k_ω)	Speed-loop backstepping gain	—
(k_d)	(d)-axis current-loop gain	—
(k_q)	(q)-axis current-loop gain	—
(i_{qref})	Reference (q)-axis current generated by the speed controller	A
(v_{dref})	Reference (d)-axis voltage	V
(v_{qref})	Reference (q)-axis voltage	V
(V_{ref})	Reference voltage vector magnitude	V

MVT-NLS Observer

Symbol	Description	Unit
$(\hat{x})$	Estimated state vector	—
$(\hat{\imath}_d, \hat{\imath}_q)$	Estimated (d)- and (q)-axis currents	A
$(\hat{\omega}_m)$	Estimated mechanical angular speed	rad/s
$(\hat{\omega}_e)$	Estimated electrical angular speed	rad/s
$(\hat{\theta}_e)$	Estimated electrical rotor position	rad
(L)	Observer gain matrix	—
(A)	System/state matrix	—
(B)	Input matrix	—
(C)	Output matrix	—
(D)	Feedthrough matrix	—
(x)	State vector	—
(u)	Control/input vector	—
(y)	Measured output vector	—

SVPWM and Three Phase Inverter

Symbol	Description	Unit
(V_{dc})	DC-link voltage	V
$(V_{dc,ref})$	Reference DC-link voltage	V
(V_{ref})	Reference voltage-vector magnitude	V
(T_s)	Sampling/PWM period	s
(T_1,T_2)	Active-vector dwelling times	s
(T_0)	Zero-vector dwelling time	s
(S_a, S_b, S_c)	Switching states of the three inverter legs	—
(V_1, V_6)	Active space voltage vectors	V
(V_0,V_7)	Zero voltage vectors	V
(M)	Modulation index	—
(f_{sw})	Switching frequency	Hz
(D_a, D_b, D_c)	Duty cycles of the three inverter legs	—
(T_{dead})	Dead time	S

Abbreviation

Abbreviation	Meaning
PMSM	Permanent Magnet Synchronous Motor
FOC	Field-Oriented Control
VSI	Voltage Source Inverter
SVPWM	Space Vector Pulse Width Modulation
PWM	Pulse Width Modulation
MVT	Mean Value Theorem
NLS	Nonlinear Linear Sector
MVT–NLS	Mean Value Theorem–Nonlinear Least Squares
PI	Proportional–Integral
PID	Proportional–Integral–Derivative
LQR	Linear Quadratic Regulator
ADC	Analog-to-Digital Converter
MCU	Microcontroller Unit
DSP	Digital Signal Processor
Hall	Hall-effect position sensor
DC	Direct Current

AC	Alternating Current
dq	Direct–Quadrature reference frame
αβ	Stationary orthogonal reference frame

General Introduction

The Permanent Magnet Synchronous Motor (PMSM) has become the industry standard for high-performance applications, ranging from electric vehicles to industrial robotics, due to its high-power density, efficiency, and superior torque-to-inertia ratio. However, the inherent nonlinearity of the PMSM—stemming from magnetic saturation and time-varying parameters—presents a significant challenge for traditional linear control methods.

To achieve high dynamic performance, **Field Oriented Control (FOC)** is typically employed to decouple torque and flux. While standard FOC relies on proportional-integral (PI) controllers, these often struggle with external disturbances and parameter uncertainties. This research adopts a **Backstepping Controller**, a recursive command law that stabilizes the entire system by "stepping back" through the state equations, ensuring global asymptotic stability.

Furthermore, to reduce system cost and increase reliability, "sensorless" operation is pursued. This work utilizes a **Nonlinear Observer based on the Mean Value Theorem (MVT)** to estimate rotor position and speed, providing a robust alternative to physical sensors. Finally, **Space Vector Pulse Width Modulation (SVPWM)** is implemented to optimize DC bus utilization and reduce harmonic distortion, ensuring the control signals are translated efficiently to the motor phases.

To address the limitations of existing PMSM drive systems, this thesis seeks to answer the following questions:

- How can a **Backstepping control law** be mathematically formulated to ensure robust tracking performance under varying load conditions compared to classical PI control?
- In what ways does a **Nonlinear Observer based on MVT** improve the accuracy of state estimation in the presence of nonlinear disturbances?
- How does the integration of **SVPWM** influence the total harmonic distortion (THD) and efficiency of the backstepping-controlled PMSM drive?
- Can the stability of the combined controller-observer-modulator system be guaranteed using **Lyapunov stability theory**?

The primary objective of this research is to design and validate a high-performance, sensorless control suite for a PMSM. Specific objectives include:

Design and Simulation: To develop a comprehensive mathematical model of the PMSM and implement a Backstepping controller in a simulation environment.

Observer Integration: To design a Nonlinear State Observer using the Mean Value Theorem (MVT) to accurately estimate speed and position without the need for mechanical sensors.

Modulation Optimization: To implement an SVPWM algorithm to drive the inverter, focusing on maximizing the voltage utilization and minimizing torque ripple.

Performance Verification: To analyze the system's robustness against parameter variations (such as stator resistance changes) and external load shocks.

Chapter1: Permanent Magnet Synchronous Motor

1. Introduction:

Ever wondered how electric cars accelerate so smoothly, industrial robots move with pinpoint accuracy, or wind turbines generate energy so efficiently? The secret lies in a special type of motor called the Permanent Magnet Synchronous Motor (PMSM). Known for its exceptional efficiency, durability, and impressive energy savings, the PMSM is becoming the "heart" of many modern technologies.[1]

2. What is a Permanent Magnet Synchronous Motor?

A Permanent Magnet Synchronous Motor (PMSM) is a type of AC motor that uses permanent magnets embedded in or attached to its rotor to create a constant magnetic field. Unlike induction motors, which rely on current induced in the rotor windings to generate a magnetic field, PMSMs use high-strength magnets often made from rare-earth materials like neodymium to maintain rotor magnetism without the need for electrical excitation.

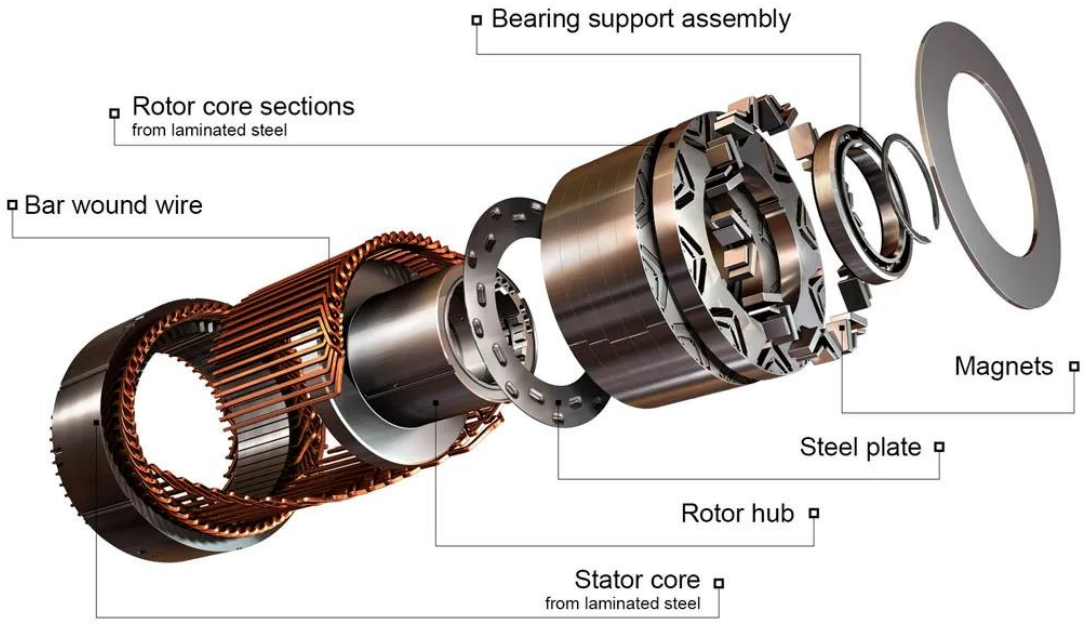


Figure 2-1: permanent magnet synchronous motor

The term "synchronous" refers to the fact that the rotor and the stator's rotating magnetic field move at the same speed, ensuring precise control over torque and position. This synchronous operation, combined with the use of permanent magnets, results in superior energy efficiency, high power density, and consistent performance across a wide range of operating conditions.[1][2]

PMSMs have gained significant attention in recent years thanks to their ability to deliver high torque at low speeds, operate quietly, and maintain efficiency even under heavy loads.

From electric vehicles to high-performance industrial machinery, the permanent magnet synchronous motor is increasingly becoming the go-to choose for engineers seeking performance and reliability.

2.1. Structure of a PMSM

- **Rotor**: Contains high-strength permanent magnets (such as neodymium or samarium-cobalt) that produce a constant magnetic field.
- **Stator:** Houses copper windings that carry alternating current to generate a rotating magnetic field.
- **Position Sensors**: Devices like encoders or resolvers are often used for precise control of rotor position, essential in applications requiring high accuracy.

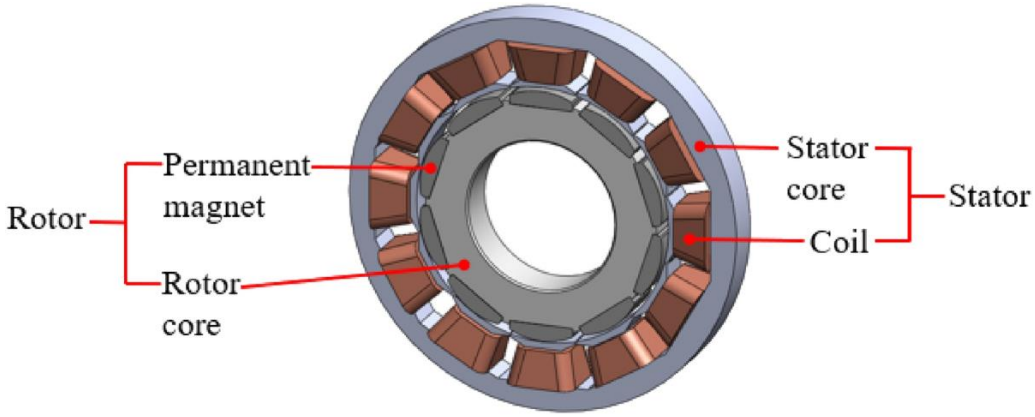


Figure 2-*2*: permanent magnet motor structure

2.2. How Does a Permanent Magnet Synchronous Motor Work?

The operation of a permanent magnet synchronous motor is based on the interaction between the magnetic field of the rotor's permanent magnets and the rotating magnetic field generated by the stator windings.

When a three-phase AC current flows through the stator, it produces a rotating magnetic field. The permanent magnets on the rotor are attracted to this field, causing the rotor to turn at the exact same speed hence the term "synchronous.

- **Current Input:** A three-phase AC supply energizes the stator windings.
- **Magnetic Field Creation:** The energized stator produces a rotating magnetic field.
- **Synchronous Rotation:** The rotor's permanent magnets lock onto the stator's field, rotating at the same speed without slip.
- **Torque Generation:** The interaction between the stator field and the rotor's magnetic field produces continuous torque output.

Unlike induction motors, PMSMs do not suffer from slip losses, allowing them to operate with higher efficiency and better torque control. Additionally, the absence of rotor windings reduces heat generation and minimizes maintenance needs.[3]

3. Types of Permanent Magnet Synchronous Motors

The PMSM is typically divided into two subcategories, the Surface permanent magnet synchronous machine (SPMSM), and the IPMSM, depending on how the magnets are located in the rotor.

The most important difference between these two machines is that the IPMSM has saliency. The inductance in the d-axis is lower than the inductance in the $q-axis$, i.e., $L_d > L_q$. This is caused by the fact that the permanent magnets has approximately the same reluctance as air. When the magnets are buried, some

part of the rotor will exhibit higher reluctance than others. Another difference is that burying the magnets provides a more robust construction, which may support operation at higher speeds [8].

An interesting side note is that the inductance in a separately excited salient synchronous machine has the exact opposite property of the IPMSM. The direct axis inductance is higher because the rotor contains more iron in the position of the field winding, i.e.,$L_d > L_q$. Due to the saliency of the IPMSM, it offers somewhat higher flexibility in the development of algorithms for estimation of rotor position. The property of position depended inductance may be exploited.

A area of active research when it comes to PMSM, is ways to minimize torque ripple. A review of various techniques, including motor design, and drive design, is presented in [9].

3.1. Surface-Mounted PMSM (SPMSM)

In a surface-mounted design, the permanent magnets are fixed directly onto the surface of the rotor.

- **Avantages:**
  - Simple rotor construction.
  - Low manufacturing cost compared to other PMSM types.
  - Excellent efficiency at low to medium speeds.
- **Disadvantages:**
  - Lower mechanical strength due to exposed magnets.
  - Limited ability to handle high-speed operations.

Typical Applications: Low- to medium-speed electric drives, pumps, fans, and general-purpose industrial equipment.

3.2. Interior PMSM (IPMSM):

In an interior design, the permanent magnets are embedded inside the rotor. This structure allows for both magnetic torque and reluctance torque generation.

❖ **Advantages:**

- Higher mechanical strength, making it suitable for high-speed operations.
- Better torque density and efficiency over a wider speed range.
- Improved protection for magnets from mechanical damage.

❖ **Disadvantages:**

- More complex and expensive to manufacture.
- Requires precise design and control strategies.

Typical Applications: Electric vehicles, high-performance industrial machinery, and servo drive systems.

4. Advantages of Permanent Magnet Synchronous Motors:

PMSMs offer several benefits that make them increasingly popular in modern engineering:

- **High Efficiency**: The rotor's magnetic field is generated by permanent magnets, eliminating the slip losses present in induction motors. This results in efficiencies often exceeding 90%.
- **High Power Density**: The use of rare-earth magnets enables PMSMs to deliver high torque output relative to their size, making them ideal for space-constrained designs.
- **Precise Torque and Speed Control**: Because the rotor rotates in perfect synchrony with the stator's magnetic field, PMSMs can achieve highly accurate torque control, which is crucial in robotics, CNC machines, and EVs.
- **Quiet and Smooth Operation**: Lower vibration and noise levels improve user comfort and reduce wear on connected components.
- **Low Maintenance Requirements**: Without brushes or rotor windings, PMSMs generate less heat and experience fewer mechanical failures, extending their service life.

In summary, these advantages explain why PMSMs are a preferred choice in applications demanding efficiency, precision, and long-term reliability.[4]

5. Disadvantages of Permanent Magnet Synchronous Motors:

Despite their strengths, PMSMs are not without challenges:

- Higher Initial Cost: Permanent magnets, particularly those made from neodymium or samarium-cobalt, are costly and contribute to a higher purchase price.
- Temperature Sensitivity: High operating temperatures can weaken the magnetic field, reducing motor performance over time.
- Complex Contrastively: PMSMs require sophisticated controllers and position sensors, increasing the system's overall cost and complexity.
- Risk of Demagensitivity: Excessive heat, overcurrent, or mechanical stress can permanently damage the magnets.

In conclusion, while PMSMs deliver exceptional performance and efficiency, their higher costs and control requirements mean they are best suited for applications where these benefits outweigh the trade-offs.

6. PMSM applications:

Thanks to their high efficiency, precision control, and compact design, permanent magnet synchronous motors (PMSMs) have found their way into a wide variety of industries. From powering electric cars to driving industrial automation, PMSMs are a key enabler of modern technology.

- Electric Vehicles (EVs) and Hybrid Cars

PMSMs are widely used in the automotive industry because of their ability to deliver high torque at low speeds, maintain efficiency across a broad speed range, and operate quietly. These characteristics make them ideal for EVs and hybrid powertrains, where energy efficiency directly impacts driving range.

- Industrial Automation and Robotics

In manufacturing, PMSMs drive robotic arms, CNC machines, and conveyor systems with unmatched precision. Their synchronous operation ensures accurate positioning, which is critical for automated assembly lines and high-precision machining.

- HVAC Systems

Heating, ventilation, and air conditioning systems benefit from PMSMs' high efficiency and low noise levels. By reducing energy consumption, PMSMs help lower operational costs and contribute to greener building solutions.

- Renewable Energy Systems

In wind turbines and hydroelectric generators, PMSMs are used to convert mechanical energy into electrical power with minimal loss. Their robust design allows them to withstand fluctuating loads while maintaining efficiency.

- Aerospace and Marine Applications

From propulsion systems in ships to specialized equipment in aircraft, PMSMs are valued for their reliability, high power density, and reduced maintenance needs in challenging environments.[4][5]

7. Conclusion

The permanent magnet synchronous motor is more than just another electric motor, it's a driving force behind the technological shift toward greater efficiency, precision, and sustainability.

By combining the power of rare-earth magnets with synchronous operation, PMSMs deliver performance levels that traditional motors struggle to match. From electric vehicles and renewable energy systems to industrial automation and aerospace applications, their versatility knows no bounds.

It is also worth noting that PMSM technology forms the basis for other advanced motor types, such as Linear Motors, which provide direct linear motion without mechanical transmission, and Torque Motors, which deliver high torque at low speeds for applications requiring direct-drive performance. Together, these technologies illustrate how permanent magnet-based designs are shaping the future of efficient, intelligent, and versatile drive systems.

Chapter2: Modeling Permanent Magnet Synchronous Motor

1. Modeling the PMSM in the dq frame:

Three-phase AC fed to stator windings produces a rotating magnetic field. The rotor has a high- performance magnet to obtain a strong magnetic field such as neodymium iron boron or rare earth magnetic materials. The interaction between the rotor and stator magnetic field generates electromagnetic torque, which can be controlled by regulating the amplitude and phase of the three-phase currents. This can be achieved by using a three-phase inverter with Pulse Width Modulation (SPWM) techniques.

In order to facilitate analysis, the following assumptions are made: three-phase symmetrical stator winding, uniform air gap, ignoring the end effect, neglecting iron loss and magnetic saturation, linear magnetic circuit, converter provides ideal three-phase power, higher harmonics is ignored, ignoring rotor shaft friction.

The PMSM is a three-phase, star-connected machine with a sinusoidal back-EMF. however, saliency is considered, i.e., $L_d = L_q$

The inverter is assumed ideal in the modeling stage; switching effects are addressed separately in the control implementation via SVPWM.

be assumed that their resistances are identical: $R_a = R_b = R_c = R_s$. In the mathematicalmodel, the saturation and temperature effects are also neglected. [6]

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = R_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} \tag{2.1}$$

where v_a, v_b, v_c are the three phase stator voltages, R_s is the stator resistance, i_a, i_b,c are thethree phase stator currents, and ψ_a, ψ_b, ψ_c are the three phase stator flux linkages, given as:

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \psi_{pm} \begin{bmatrix} cos\,\theta_e \\ cos(\theta_e - \frac{2\pi}{3}) \\ cos(\theta_e + \frac{2\pi}{3}) \end{bmatrix} \tag{2.2}$$

Where L_{aa}, L_{bb}, L_{cc} are stator phase self-inductances, $L_{ab} = L_{ba}$, $L_{ac} = L_{ca}$, $L_{bc} = L_{cb}$are mutual inductances between respective stator phases, ψ_{pm} is the permanent magnetflux linkage, and $\theta_e$is the electrical rotor position. By means of Clarke and Park transformation, three phase voltage (2.1) and flux equations (2.2) can be transformed to the rotational reference frame dq:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R_s \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_d \\ \psi_q \end{bmatrix} + \omega_e \begin{bmatrix} -\psi_q \\ \psi_d \end{bmatrix} \tag{2.3}$$

where $v_d$and v_q are the dq axis voltages, $i_d$and i_q are the dq axis currents, ω_e is theelectrical angular velocity, and ψ_d, $\psi_q$are the dq axis flux linkages, defines as:

$$\begin{bmatrix}\psi_d\\ \psi_q\end{bmatrix} = \begin{bmatrix}L_d & 0\\ 0 & L_q\end{bmatrix}\begin{bmatrix}i_d\\ i_q\end{bmatrix} + \psi_{pm}\begin{bmatrix}1\\ 0\end{bmatrix} \tag{2.4}$$

where $L_d$and L_q are the dq axis inductances. Substituting (4) into (3), the voltage equations of the PMSM in dq coordinates lead to the following form:

$$\begin{bmatrix}v_d\\ v_q\end{bmatrix} = R_s\begin{bmatrix}i_d\\ i_q\end{bmatrix} + \begin{bmatrix}L_d & 0\\ 0 & L_q\end{bmatrix}\frac{d}{dt}\begin{bmatrix}i_d\\ i_q\end{bmatrix} + \omega_e\begin{bmatrix}-L_q\\ L_d\end{bmatrix}\begin{bmatrix}i_q\\ i_d\end{bmatrix} + \psi_{pm}\begin{bmatrix}1\\ 0\end{bmatrix} \tag{2.5}$$

From the equation (2.3) we can write the equations of the PMSM as following:

1.1. Electrical dynamics in the dq frame:

The stator voltage equations in the synchronous dqframe is given by:

$$v_d = R_s i_d + L_d\frac{di_d}{dt} - \omega_e L_q i_q \tag{2.6}$$

$$v_q = R_s i_q + L_q\frac{di_q}{dt} + \omega_e L_d i_d + \omega_e\psi_{pm} \tag{2.7}$$

A simplified per-phase equivalent circuit of the PMSM is provided in figure.3-1, valid for balanced sinusoidal steady state operation [7].

In the figure, the induced back-electromotive force (EMF) is represented by a voltage source. The voltage vector position depends on the rotor position, and this is the basis for the position estimators covered later in the project.

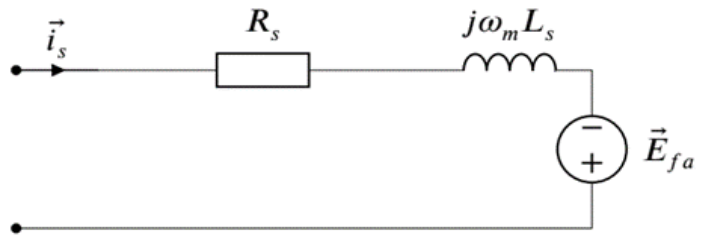


Figure 3-1: simplified equivalent circuit of PMSM

State space model:

$$\frac{di_d}{dt} = -\frac{R_s}{L_d}i_d + \omega_e i_q + \frac{1}{L_d}v_d \tag{2.8}$$

$$\frac{di_q}{dt} = -\frac{R_s}{L_q} i_q - \omega_e i_d - \frac{\psi_{pm}}{L_q} \omega_e + \frac{1}{L_q} v_q \tag{2.9}$$

The permanent magnet flux linkage ψ_{pm}may vary due to demagnetization and flux weakening at high speed. Load torque T_L is unknown and considered as a disturbance. Rotor speed and electrical position are not measured (sensorless operation).

The mathematical model of PMSM in rotational reference frame dq is very popularfor FOC structures, because both controllable quantities, current and voltage, are DC values, as is evident from Figure 1. In addition, it allows employing simple controllers to force the machine currents to the require states.

The transformation from the three-phase abc frame to the dq frame is achieved via the Clarke and Park transformations, using the electrical rotor angle θ_e.

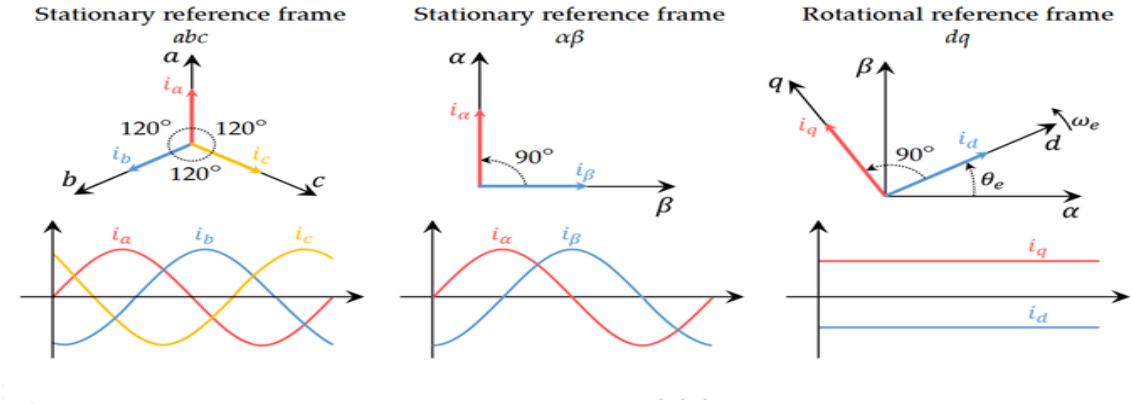


Figure 3-2: Transformation from stationary to rotational reference frame

For three phase voltage system, with V_a, V_b and $V_c$voltage and its frequency is ω, can be expressed by equation (2.10)

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = V \begin{bmatrix} cos\ \omega t \\ cos(\omega t - \frac{2\pi}{3}) \\ cos(\omega t + \frac{2\pi}{3}) \end{bmatrix} \tag{2.10}$$

For these three time-varying quantities, to transform into rotating reference frame, the first step is to transform these voltages into stationary reference frame or with orthogonal quantities system, or alfa–beta transform by projection of three-phase voltages on orthogonal axis which is called Clarke transform as expressed in equation (2.10).

$$V\alpha\beta 0 \ = \ T\ abc \ \rightarrow \ \alpha\beta 0\ Vabc$$

$$\begin{bmatrix} v_\alpha \\ v_\beta \\ v_0 \end{bmatrix} = \frac{\sqrt{3}}{2} \begin{bmatrix} 1 & cos(\frac{2\pi}{3}) & cos\left(-\frac{2\pi}{3}\right) \\ 0 & sin\left(\frac{2\pi}{3}\right) & sin\left(-\frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \frac{\sqrt{3}}{2} \begin{bmatrix} cos\,\omega t \\ sin\,\omega t \\ 0 \end{bmatrix} V \quad (2.11)$$

As shown in equation (2.10) and from Figure 1, the voltage net vector d forms an angle θ_e with the α axis of theorthogonal$\alpha\beta$ frame or the stationary frame. This vector rotates atω frequency. This means that by taking the net of the stationary reference frame by the projection of α and βcomponents, the net d vector is reduced to DC amplitudeand varying angle which is much easier to deal with inthe control section than dealing with three time-varying quantities of three-phase voltage system. This is called Park transform and can be expressed in (8) and Figure 1:

$$Vdq0 = T\alpha\beta0 \rightarrow dq0V\alpha\beta0$$

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = \begin{bmatrix} cos\,\theta_e & sin\,\theta_e & 0 \\ -sin\,\theta_e & cos\,\theta_e & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \\ v_0 \end{bmatrix} \quad (2.12)$$

1.2. Electromagnetic torque equation:

The other important quantity of the machine is the electromagnetic torque. The electromagnetic torque of the PMSM machine in rotational reference frame dq is defined as:

$$Te = \frac{3}{2} p\psi_{pm} i_q \quad (2.13)$$

This equation shows:

- Linear torque contribution from i_q ,
- Nonlinear reluctance torque term when $L_d = L_q$.

where p is the number of pole pairs, the first term in Equation (2.10) represents a synchronous torque produced by the permanent magnet flux, and the second term is a reluctance torque dependent on the rotor saliency. Both terms reflect an essential aspect of the torque production in the PMSM machine. By sublimation of the electromagnetic torque and load torque, the mathematical equation of PMSM can be presented as:

1.3. Mechanical dynamics:

The rotor mechanical dynamics are governed by Newton's law:

$$J\frac{d\omega_m}{dt} = T_e - T_L - B\omega_m \quad (2.14)$$

where T_L is the load torque, J is the moment of inertia, and ω_m is the mechanical angular velocity. The block diagram of the PMSM in rotational reference frame dq is illustrated in Figure 3-3 [6]

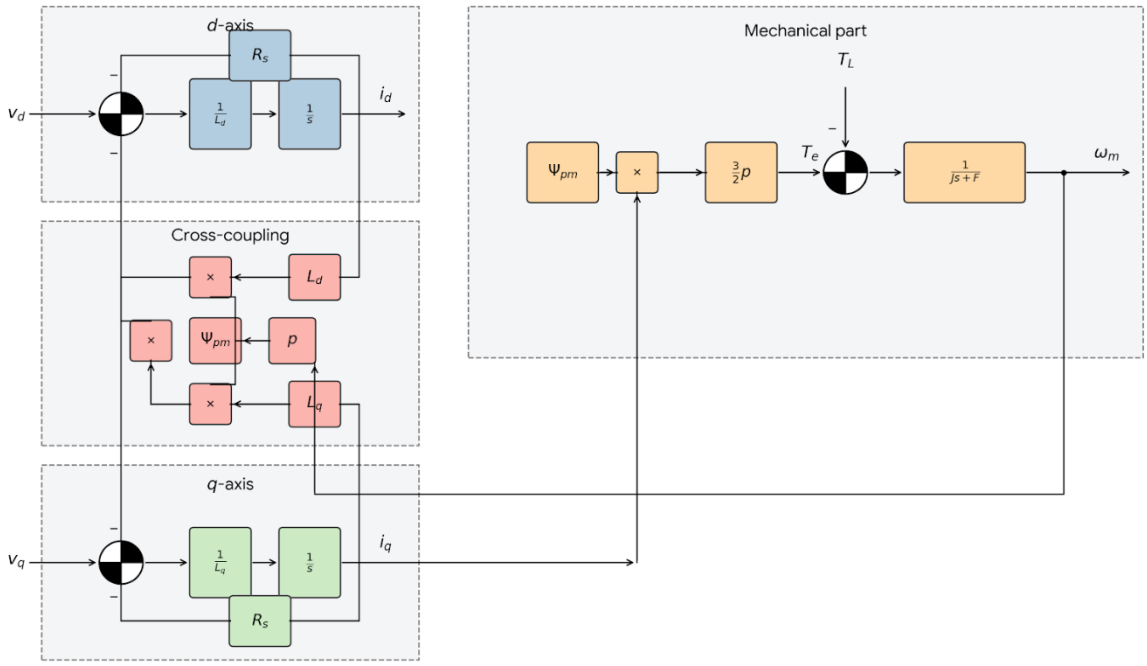


Figure 3-3: diagram of the PMSM in rotational dq frame

$$\frac{d\theta_m}{dt} = \omega_m \tag{2.15}$$

Using the electrical speed relation:

$$\omega_e = p\omega_m \tag{2.16}$$

The mechanical equations in electrical coordinates become:

$$\frac{d\omega_e}{dt} = \frac{p}{J}\left(T_e - T_L - B\frac{\omega_e}{p}\right) \tag{2.17}$$

$$\frac{d\theta_e}{dt} = \omega_e \tag{2.18}$$

The induced back-EMF is given by:

$$E_{fa} = \sqrt{\frac{3}{2}}\psi_{pm}\omega_m \tag{2.19}$$

Hence the voltage constant is given by:

$$k_e = \sqrt{\frac{3}{2}}\psi_{pm} \tag{2.20}$$

2. Complete nonlinear state-space model:

Define the state vector:

$$x = \left[i_{d,}\ \ i_q, \omega_m, \theta_e\right]^T \tag{2.21}$$

Input vector:

$$u = [vd\ vq]^T \tag{2.22}$$

Measured outputs:

$$y = \left[i_d\ \ i_q\right]^T \tag{2.23}$$

The full nonlinear PMSM model is then written as:

$$\frac{di_d}{dt} = -\frac{R_s}{L_d} i_d\ + \omega_e i_q\ + \frac{1}{L_d}\ vd \tag{2.24}$$

$$\frac{di_q}{dt}\ = -\frac{R_s}{L_q} i_q\ - \omega_e i_d\ - \frac{\psi_{pm}}{L_q}\omega_e + \frac{1}{L_q}\ vq \tag{2.25}$$

$$\frac{d\omega_m}{dt} = \frac{3p}{2J}\psi_{pm} i_q - \frac{1}{J} T_L - \frac{B}{J}\omega_m \tag{2.26}$$

3. Nonlinear and uncertainties:

This model contains:

❖ **Nonlinear terms:**

- $\omega_e i_d$
- $\omega_e i_q$
- $i_d i_q$

❖ **Uncertain parameters:**

- $Rs(T)$
- ψ_{pm}
- BJ
- $Load\ torque\ T_L$

Chapter 3: Sensorless FOC control

1. Control methods:

There are several different algorithms that may be used when designing a drive fora electric machine. The preferred method depends on the performance constraints, the cost, the motor and inverter topology, as well as the skills of the engineer responsible for the design. There are two popular methods that may be regarded as the classical control methods for motor drives. The FOC (the subject of our thesis), and the DTC. However, several other, simpler, and more advanced algorithms are available.[8]

2. FOC (Field Oriented Control) control:

FOC is a type of vector control that presents the advantages of providing smooth starting and acceleration, and of obtaining an effective dynamic response with reduced torque ripples. Hence FOC is applied frequently in PMSM drives, too. [9]

It's a method in which phase and amplitude for a motor stator voltage or current vector is controlled at the same time. FOC simplifies the control of PMSM by making it to behave like a separately excited DC machine in which developed torque and magnetizing field can be controlled in dependently [7]. For this three-phase time variant system (abc) is converted into time invariant system (dq). This causes decoupling of Torque and fluxcomponent to make control simple. Clarke and Parktransformation helps to convert abc components intodq components.

3. FOC structure:

In FOC the classical Proportional integral (PI) controller is commonly used in several cascaded loops to control the behavior of the motor. Figure 6 provides an overview of a typical FOC algorithm. Apart from the position and speed estimator, the implementation is straight forward, and in this thesis, we will discuss each part separately unless we'll replace the PI's controllers with a nonlinear one (backstepping controller).

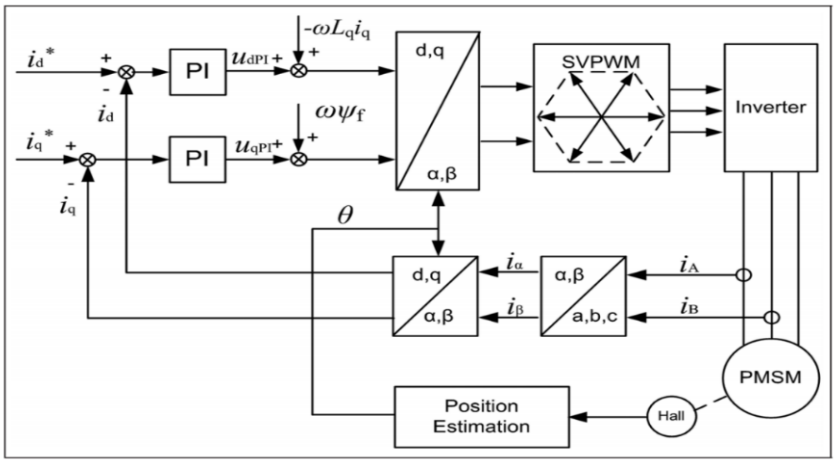


Figure 4-1: Sensorless FOC for PMSM

The three phase currents are measured1, and converted to *d-q* frame using the Clarke transformation [9]. The *d-q*-currents are used in conjunction with the *d-q*-voltages to estimate the rotor flux position. The position is in turn used to transform the *d-q*-currents to the *d-q*-frame using the Park transformation.

As the *d-q*-frame rotates synchronous with the rotor, the currents will appear as DC quantities in steady state. This enables the use of classical PI controllers / backstepping controller to control the currents. As the d-axis is aligned with the magnetic field of the rotor, the q-axis current will be the torque producing component of the current, and its reference is controlled by an external speed controller [10].

The d-axis current reference is normally set to zero for a PMSM, in order to obtain maximum torque per ampere operation. If the speed reference is above the rated motor speed, negative d-axis current is used to decrease the magnetic field, and in turn the induced back-EMF. This is known as field weakening, and the motor torque is decreased in inverse proportion to the speed.

The output from the PI current controllers / backstepping controller is the *d-q*-voltage references to the motor. These references are converted to *d-q* frame using the inverse Park transformation, and in turn fed to a SV-PWM generator. The SV-PWM computes the switching times, and generate switches signals to the inverter. The speed controller feedback is the estimated rotor speed as obtained by differentiation of the estimated position.

The reference values for d, and q-axis currents for a PMSM is given as:

$$I_{dref} = 0$$

$$I_{qref} = \frac{T_e}{\psi_m}$$

Coordinates transformations:

- Clarke transformation($\boldsymbol{abc \rightarrow \alpha\beta}$):

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{3.1}$$

$$i_\alpha = i_a \tag{3.2}$$

$$i_\beta = \frac{1}{\sqrt{3}}(i_a + 2i_b) \tag{3.3}$$

Park transformation($\alpha\beta \rightarrow dq$):

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} cos\,\hat{\theta}_e & sin\,\hat{\theta}_e \\ -sin\,\hat{\theta}_e & cos\,\hat{\theta}_e \end{bmatrix}\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{3.4}$$

The transformation uses the **estimated rotor position** $\hat{\theta}_e$, which is provided by the observer.

FOC can be implemented with or without shaft sensors. Shaft sensors such as optical encoders or resolvers are generally used in AC drives to identify the rotor position and the speed. However, systems using shaft sensors have increased complexity, volume and cost.

4. Motivation for sensorless control:

The sensorless techniques for different types of electric machines are still improving, especially in the automotive industry. PMSMs are highly demanded in such applications for their high-power density factor, high dynamic features, robustness, and high efficiency. Due to cost reduction, which is a critical factor, especially in the automotive industry, researchers are forced to find new solutions for not only electrical drive but also the battery management system, power electronics, machine design, or software modifications of different control strategies. In particular, sensorless strategies are still very popular and attractive, because they offer the elimination of mechanical speed or position sensors coupled with the rotor shaft. This eliminates potential troubles coming mainly from the mechanical connection but also coming with vibration or temperature instability. A damaged sensor or one with inappropriate functionality would completely disable further motor operation in a sensored motor drive. Therefore, a proper sensorless algorithm could improve the minimization of the final cost, and furthermore, this solution decreases drive volume or maintenance cost. However, sensorless control could also offer a specified level of redundancy in case of a sudden mechanical sensor failure, which could be even more attractive, especially in the automotive industry.

For this thesis we proposed an MVT-NLS (Mean Value Theorem) based observer as a sensorless control and through this we'll the powerful of one of many sensorless method in building a robust and smooth control without using sensors.

5. Mean Value Theorem (MVT) Method in Nonlinear Control and Sensorless Observer Design:

In nonlinear control systems, especially in sensorless control and nonlinear observer design, the nonlinear dynamics of the system often complicate stability analysis and observer synthesis. To overcome this difficulty, the **Mean Value Theorem (MVT)** is frequently employed to transform nonlinear functions into an equivalent quasi-linear form. This transformation allows the nonlinear system to be represented as a convex combination of linear models, which simplifies the analysis and facilitates the design of observers and controllers.

The MVT-based approach has become widely used in the field of sensorless electrical drives, nonlinear state estimation, and adaptive observers because it offers an effective compromise between exact nonlinear modeling and linear approximation techniques. In Permanent Magnet Synchronous Motor (PMSM) drives and induction motor applications, the MVT method enables the derivation of observer error dynamics suitable for Lyapunov stability analysis and Linear Matrix Inequality (LMI) formulations [11]

5.1. Definition:

what does the Mean Value Theorem mean?

For a differentiable nonlinear function $f(x)$, the Mean Value Theorem states that for any two points x and $\hat{x}$, there exists a point ξ lying between them such that [12]:

$$f(x) - f(\hat{x}) = \frac{\partial f(\xi)}{\partial x}(x - \hat{x}) \tag{3.5}$$

Where:

- $\frac{\partial f(\xi)}{\partial x}$ is the Jacobian matrix evaluated at an intermediate point ξ.
- x is the actual state vector,
- $\hat{x}$ is the estimated state vector.

This theorem converts the **nonlinear difference** into a linear relation depending on the Jacobian matrix.

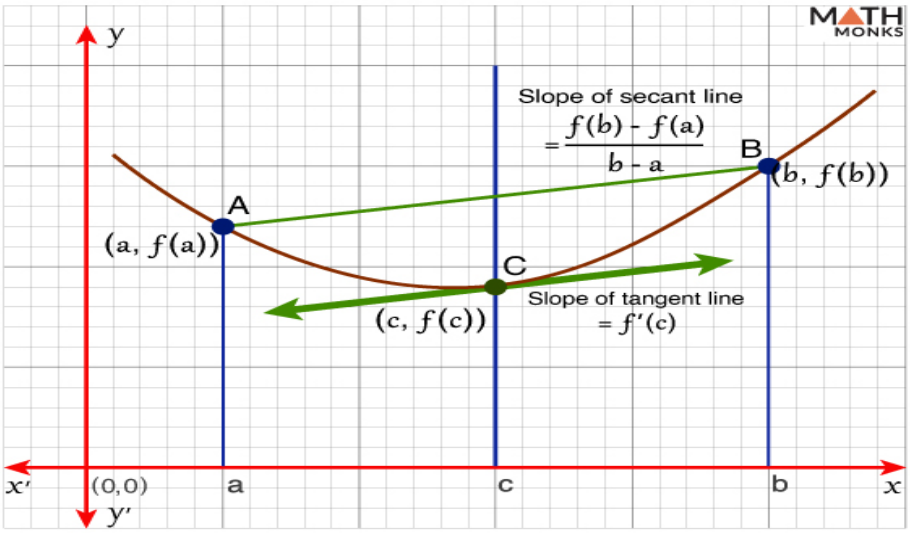


Figure 4-2: Mean Value Theorem definition

5.2. Application of MVT in Nonlinear Observer Design:

Consider the nonlinear system:

$$\dot{x}(t) = Ax(t) + Bu(t) + \phi\big(x(t)\big) \tag{3.6}$$

$$y(t) = Cx(t) \tag{3.7}$$

where:

- $A \in \mathbb{R}^{nxn}$, $B \in \mathbb{R}^{nxm}$, $C \in \mathbb{R}^{pxn}$ are constant matrices.
- $\phi\big(x(t)\big) \in \mathbb{R}^n$ is a vector function containing all non-linear coupling terms (e.g., cross-coupling currents, speed multiplication terms, or trigonometric functions of rotor angle).

A nonlinear observer can be written as:

$$\hat{\dot{x}}(t) = A\hat{x}(t) + Bu(t) + \phi\big(\hat{x}(t)\big) + L\big(y(t) - \hat{y}(t)\big) \tag{3.8}$$

with:

$$\hat{y}(t) = C\hat{x}(t) \tag{3.9}$$

Defining the estimation error:

$$e = x(t) - \hat{x}(t) \tag{3.10}$$

the observer error dynamics become:

$$\dot{e} = (A - LC)e + \phi\big(x(t)\big) - \phi\big(\hat{x}(t)\big) \tag{3.11}$$

Using the Mean Value Theorem:

$$\phi\big(x(t)\big) - \phi\big(\hat{x}(t)\big) = J(\xi)\big(x(t) - \hat{x}(t)\big) = J(\xi)e \tag{3.12}$$

Where:

$$J(\xi) = \frac{\partial\phi\big(x(t)\big)}{\partial x} | (xm < x = \xi < xM) \tag{3.13}$$

thus:

$$\dot{e} = \big(A - LC + J(\xi)\big)e \tag{3.14}$$

This representation transforms the nonlinear estimation error dynamics into a parameter-varying linear form, greatly simplifying stability analysis and observer gain design [13].

6. Importance of MVT in Sensorless Control:

In sensorless control systems, mechanical sensors such as encoders or tachometers are eliminated in order to reduce cost, improve reliability, and increase robustness. [14] Consequently, rotor speed, position, or flux must be estimated using nonlinear observers.

The nonlinear electrical machine model contains nonlinear coupling terms depending on speed, flux, and electromagnetic torque. The MVT method allows these nonlinearities to be expressed through bounded Jacobian matrices [15], enabling:

- Lyapunov-based stability analysis,
- LMI observer synthesis,
- robust estimation under parameter variations,
- reduction of modeling conservatism compared with Lipschitz approaches.

For PMSM and induction motor drives, MVT-based observers have shown excellent estimation performance and robustness in low-speed operating regions.

7. Advantages of the MVT Approach

The main advantages of the MVT method in nonlinear control are [16]:

1. Transformation of nonlinear systems into quasi-linear forms.
2. Simplification of observer and controller synthesis.
3. Reduction of conservatism compared with Lipschitz methods [17].
4. Compatibility with Lyapunov and LMI stability techniques.
5. Effective application to sensorless electrical drives and nonlinear estimation problems [18].

8. Build MVT-Nonlinear Sector (NLS) observer for PMSM motor:

for this study we take a simplified PMSM model with [19]:

$$L_d = L_q = L_s \tag{3.15}$$

Because that removes saliency nonlinearities:

- $i_d i_q$
- Reluctance torque term $(L_d - L_q)$

So, the PMSM equations become:

$$\begin{cases} \dot{i_d} = -\frac{R_s}{L_s} i_d + \omega_e i_q + \frac{1}{L_s} vd \\ \dot{i_q} = -\frac{R_s}{L_s} i_q - \omega_e i_d - \frac{\psi_{pm}}{L_s} \omega_e + \frac{1}{L_s} vq \\ \dot{\omega_e} = a i_q - b\omega_e - cT_L \end{cases} \tag{3.16}$$

Where:

$$a = \frac{3p^2}{2J}\psi_{pm}, \quad b = \frac{B}{J}, \quad c = \frac{P}{J}$$

$$\dot{\theta} = \omega_e = p\omega_m \tag{3.17}$$

So, the states vector:

$$x = \begin{bmatrix} i_d \\ i_q \\ \omega_e \end{bmatrix} \tag{3.18}$$

Measured outputs:

$$y = \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{3.19}$$

State space form:

$$\dot{x}(t) = Ax(t) + \phi\big(x(t)\big) + Bu(t) \tag{3.20}$$

Linear part matrix:

$$A = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & -\frac{\psi_{pm}}{L_s} \\ 0 & a & -b \end{bmatrix} \tag{3.21}$$

Nonlinear part matrix:

$$\phi(x) = \begin{bmatrix} \omega_e i_q \\ -\omega_e i_d \\ 0 \end{bmatrix} \tag{3.22}$$

$$\begin{cases} \phi_1\big(x(t)\big) = \omega_e i_q \\ \phi_2\big(x(t)\big) = -\omega_e i_d \\ \phi_3\big(x(t)\big) = 0 \end{cases} \tag{3.23}$$

Input matrix:

$$B = \begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \\ 0 & 0 \end{bmatrix} \tag{3.24}$$

Output matrix:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.25}$$

Observer equations:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + \phi\big(\hat{x}(t)\big) + Bu(t) + L\big(y(t) - \hat{y}(t)\big) \tag{3.26}$$

Gain matrix:

$$L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \tag{3.27}$$

So, the PMSM observer equations:

$$\begin{cases} \dot{\widehat{\imath_d}} = -\frac{R_s}{L_s}\widehat{\imath_d} + \widehat{\omega_e}\widehat{\imath_q} + \frac{1}{L_s}vd + L_{11}(i_d - \widehat{\imath_d}) \\ \dot{\widehat{\imath_q}} = -\frac{R_s}{L_s}\widehat{\imath_q} - \widehat{\omega_e}\widehat{\imath_d} - \frac{\psi_{pm}}{L_s}\widehat{\omega_e} + \frac{1}{L_s}vq + L_{22}\big(i_q - \widehat{\imath_q}\big) \\ \dot{\widehat{\omega_e}} = a\widehat{\imath_q} - b\widehat{\omega_e} - cT_L + L_{31}(i_d - \widehat{\imath_d}) + L_{32}\big(i_q - \widehat{\imath_q}\big) \end{cases} \tag{3.28}$$

$$\begin{cases} \phi_1(\hat{x}) = \widehat{\omega_e}\widehat{\imath_q} \\ \phi(\hat{x}) = -\widehat{\omega_e}\widehat{\imath_d} \\ \phi_3(\hat{x}) = 0 \end{cases} \tag{3.29}$$

$$e = x(t) - \hat{x}(t) \tag{3.30}$$

Define errors:

$e_d = i_d - \widehat{\imath_d}$ (Direct current error)

$e_q = i_q - \widehat{\imath_q}$ (Quadratique current error)

$e_{\omega_e} = \omega_e - \widehat{\omega_e}$ (Speed error)

$$e = \begin{bmatrix} e_d \\ e_q \\ e_{\omega_e} \end{bmatrix} \tag{3.31}$$

Error Dynamics:

$$\begin{cases} \dot{e_d} = \dot{\imath_d} - \dot{\widehat{\imath_d}} = -\dfrac{R_s}{L_s} e_d + \left(\omega_e i_q - \widehat{\omega_e}\widehat{\imath_q}\right) - L_{11} e_d \\ \dot{e_q} = \dot{\imath_q} - \dot{\widehat{\imath_q}} = -\dfrac{R_s}{L_s} e_q - \left(\omega_e i_d - \widehat{\omega_e}\widehat{\imath_d}\right) - \dfrac{\Psi_{pm}}{L_s} e_{\omega_e} - L_{22} e_q \\ \dot{e_{\omega_e}} = \dot{\omega_e} - \dot{\widehat{\omega_e}} = a e_q - b e_{\omega_e} - L_{31} e_d - L_{32} e_q \end{cases} \tag{3.32}$$

- Using MVT:

Consider two trajectories [21] :

- the real state x,
- the estimated state $\hat{x}$.

The Mean Value Theorem states that

Jacobian matrix [20]:

$$J(x) = \frac{\partial \phi(x)}{\partial x} = \begin{bmatrix} \dfrac{\partial f_1(x)}{\partial i_d} & \dfrac{\partial f_1(x)}{\partial i_q} & \dfrac{\partial f_1(x)}{\partial \omega_e} \\ \dfrac{\partial f_2(x)}{\partial i_d} & \dfrac{\partial f_2(x)}{\partial i_q} & \dfrac{\partial f_2(x)}{\partial \omega_e} \\ \dfrac{\partial f_3(x)}{\partial i_d} & \dfrac{\partial f_3(x)}{\partial i_q} & \dfrac{\partial f_3(x)}{\partial \omega_e} \end{bmatrix} \tag{3.33}$$

$$\phi(x) - \phi(\hat{x}) = \frac{\partial \phi(x)}{\partial x}(x - \hat{x}) = \frac{\partial}{\partial x} \begin{bmatrix} \omega_e i_q \\ \omega_e i_d \\ 0 \end{bmatrix} (x - \hat{x}) \tag{3.34}$$

$$\phi(x) - \phi(\hat{x}) = J(\xi) e \tag{3.35}$$

where:

$$\xi = \alpha x + (1 - \alpha)\hat{x}, \alpha \in 0{,}1 \tag{3.36}$$

with:

$$0 < \alpha < 1 \tag{3.37}$$

Thus, ξ is an intermediate point between x and $\hat{x}$./

Replacing the states by intermediate variables:

$$\xi = \begin{bmatrix} \xi_{\omega_e} \\ \xi_{i_q} \\ \xi_{i_d} \end{bmatrix} \tag{3.38}$$

$$\phi(x) - \phi(\hat{x}) = \begin{bmatrix} 0 & \xi_{\omega_e} & \xi_{i_q} \\ -\xi_{\omega_e} & 0 & -\xi_{i_d} \\ 0 & 0 & 0 \end{bmatrix} (x - \hat{x}) \tag{3.39}$$

Gives:

$$J(\xi) = \begin{bmatrix} 0 & \xi_{\omega_e} & \xi_{i_q} \\ -\xi_{\omega_e} & 0 & -\xi_{i_d} \\ 0 & 0 & 0 \end{bmatrix} \tag{3.40}$$

This matrix is fundamental because:

- it transforms the nonlinear dynamics into a quasi-linear form,
- it preserves the exact nonlinear structure,
- it becomes the basis of the sector decomposition.

❖ **Important Property of the MVT Representation [21]:**

This differs fundamentally from local linearization because:

- local linearization approximates around one operating point,
- MVT remains exact everywhere inside the admissible region.

Substitution in the errors dynamics equations:

$$\dot{e} = \big(A - LC + J(\xi)\big)e \Rightarrow \tag{3.41}$$

$$\dot{e} = \left(\begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & -\frac{\psi_{pm}}{L_s} \\ 0 & a & -b \end{bmatrix} - \begin{bmatrix} L_{11} & 0 \\ 0 & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \xi_{\omega_e} & \xi_{i_q} \\ -\xi_{\omega_e} & 0 & -\xi_{i_d} \\ 0 & 0 & 0 \end{bmatrix} \right) e \tag{3.42}$$

$$LC = \begin{bmatrix} L_{11} & 0 \\ 0 & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ 0 & L_{22} & 0 \\ L_{31} & L_{32} & 0 \end{bmatrix} \tag{3.43}$$

$$A - LC = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & -\frac{\psi_{pm}}{L_s} \\ 0 & a & -b \end{bmatrix} - \begin{bmatrix} L_{11} & 0 & 0 \\ 0 & L_{22} & 0 \\ L_{31} & L_{32} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{R_s}{L_s} - L_{11} & 0 & 0 \\ 0 & -\frac{R_s}{L_s} - L_{22} & -\frac{\psi_{pm}}{L_s} \\ -L_{31} & a - L_{32} & -b \end{bmatrix} \quad (3.44)$$

$$M(\xi) = A - LC + J(\xi)$$

$$= \begin{bmatrix} -\frac{R_s}{L_s} - L_{11} & 0 & 0 \\ 0 & -\frac{R_s}{L_s} - L_{22} & -\frac{\psi_{pm}}{L_s} \\ -L_{31} & a - L_{32} & -b \end{bmatrix} + \begin{bmatrix} 0 & \xi_{\omega_e} & \xi_{i_q} \\ -\xi_{\omega_e} & 0 & -\xi_{i_d} \\ 0 & 0 & 0 \end{bmatrix} \quad (3.45)$$

$$M(\xi) = \begin{bmatrix} -\frac{R_s}{L_s} - L_{11} & \xi_{\omega_e} & \xi_{i_q} \\ -\xi_{\omega_e} & -\frac{R_s}{L_s} - L_{22} & -\frac{\psi_{pm}}{L_s} - \xi_{i_d} \\ -L_{31} & a - L_{32} & -b \end{bmatrix} \quad (3.46)$$

9. Nonlinear Sector (NLS) transformation Design for PMSM:

The previous section established the nonlinear observer design based on the Mean Value Theorem (MVT). The complete observer structure considered in this thesis is based on the combination of [22]:

- the Mean Value Theorem (MVT), and
- the Nonlinear Sector (NLS) transformation.

9.1. Identification of Sector Variables & Physical Bounds:

Since the motor operates within physical operating limits, each variable is bounded within a closed interval $[\overline{\xi_i}, \underline{\xi_i}]$ and its weighting functions $\overline{h_i}(\xi_i) and \underline{h_i}(\xi_i)$:

$$\overline{\xi_{\omega_e}} \le \xi_{\omega_e}(t) \le \underline{\xi_{\omega_e}} \quad with \ \overline{h}_1\left(\xi_{\omega_e}(t)\right) \ and \ \underline{h}_1\left(\xi_{\omega_e}(t)\right) \quad (3.47)$$

$$\overline{\xi_{i_q}} \le \xi_{i_q}(t) \le \underline{\xi_{i_q}} \quad with \ \overline{h}_3\left(\xi_{i_q}(t)\right) \ and \ \underline{h}_3\left(\xi_{i_q}(t)\right) \quad (3.48)$$

$$\overline{\xi_{i_d}} \le \xi_{i_d}(t) \le \underline{\xi_{i_d}} \quad with \ \overline{h}_2\left(\xi_{i_d}(t)\right) \ and \ \underline{h}_2(\xi_{i_d}(t) \quad (3.49))$$

$$\xi_{\omega_e}(t) = (\overline{h}_1(\xi_{\omega_e}(t))\overline{\xi_{\omega_e}}.Ax_1 + \underline{h}_1\left(\xi_{\omega_e}(t)\right)\underline{\xi_{\omega_e}}.Ax_1 \quad (3.50)$$

$$\text{with } Ax_1 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.51}$$

$$\xi_{i_d}(t) = (\overline{h}_2(\xi_{i_d}(t))\overline{\xi_{i_d}}.Ax_2 + \underline{h}_2\left(\xi_{i_d}(t)\right)\underline{\xi_{i_d}}.Ax_2 \tag{3.52}$$

$$\text{with } Ax_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.53}$$

$$\xi_{i_q}(t) = (\overline{h}_3(\xi_{i_q}(t))\overline{\xi_{i_q}}.Ax_3 + \underline{h}_3\left(\xi_{i_q}(t)\right)\underline{\xi_{i_q}}.Ax_3 \tag{3.54}$$

$$\text{with } \quad Ax_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.55}$$

9.2. Substitution into the MVT Matrix:

$$\left(\begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 \\ 0 & -\frac{R_s}{L_s} & -\frac{\psi_{pm}}{L_s} \\ 0 & a & -b \end{bmatrix} - LC + \begin{bmatrix} 0 & \xi_{\omega_e} & \xi_{i_q} \\ -\xi_{\omega_e} & 0 & -\xi_{i_d} \\ 0 & 0 & 0 \end{bmatrix}\right) = \left(A - LC + \begin{bmatrix} 0 & \xi_{\omega_e}(t) & \xi_{i_q(t)} \\ -\xi_{\omega_e}(t) & 0 & -\xi_{i_d}(t) \\ 0 & 0 & 0 \end{bmatrix}\right) \tag{3.56}$$

To apply the MVT we should represent the following matrix as:

$$\begin{bmatrix} 0 & \xi_{\omega_e} & \xi_{i_q} \\ -\xi_{\omega_e} & 0 & -\xi_{i_d} \\ 0 & 0 & 0 \end{bmatrix} = \xi_{\omega_e}(t)\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \xi_{i_d}(t)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}\xi_{i_q}(t)\begin{bmatrix} 0 & 0 & 1 \\ -0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.57}$$

Substitute (3.52) and (3.54) in (3.57), we obtain:

$$\dot{e} = \left(A - LC + \xi_{\omega_e}(t)\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \xi_{i_d}(t)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \xi_{i_q}(t)\begin{bmatrix} 0 & 0 & 1 \\ -0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)e \tag{3,58}$$

Now, we can apply sector nonlinearities (20):

$$(\overline{h}_1(x_1(t))\overline{x}_1.Ax_1 + \underline{h}_1(x_1(t))\underline{x}_1.Ax_1) + (\overline{h}_2(x_2(t))\overline{x}_2.Ax_2 + \underline{h}_2(x_2(t))\underline{x}_2.Ax_2) \tag{3.59}$$

Thus:

$$\dot{e} = \left(A - LC + \xi_{\omega_e}(t)\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \xi_{i_d}(t)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \xi_{i_q}(t)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}\right)e \tag{3.60}$$

$$\Big(A - LC + (\overline{h}_1(\xi_{\omega_e}(t))\overline{\xi_{\omega_e}}.Ax_1 + \underline{h}_1\left(\xi_{\omega_e}(t)\right)\underline{\xi_{\omega_e}}.Ax_1 + (\overline{h}_2(\xi_{i_d}(t))\overline{\xi_{i_d}}.Ax_2 +$$

$$\underline{h}_2\left(\xi_{i_d}(t)\right)\underline{\xi_{i_d}}.Ax_2 + (\overline{h}_1(\xi_{i_q}(t))\overline{\xi_{i_q}}.Ax_3 + \quad \underline{h}_1\left(\xi_{i_q}(t)\right)\underline{\xi_{i_q}}.Ax_3\Big)e \qquad (3.61)$$

Thus, the dynamics of the estimation error from equation can be expressed as follows in general:

$$\dot{e}(t) = (A - LC)e(t) + \sum_{i=1}^{3\leq n^2}\sum_{r=1}^{2} {h_i}^r\big(\xi(t)\big)\,{\mathcal{A}_i}^r\big(x(t) - \hat{x}(t)\big) \qquad (3.62)$$

We can then reformulate the dynamics of the estimation error (3.62) as follows:

$$\dot{e}(t) = \sum_{i=1}^{2q\leq 2n^2} h_i\big(\xi(t)\big)(A_0 - LC + q\mathcal{A}_i)\,e(t) \qquad (3.63)$$

9.3. Stability analysis:

The stability analysis of the estimation error dynamics, represented by equation (.64), will be carried out using a quadratic Lyapunov function with a common matrix, as shown below:

$$V\big(e(t)\big) = e^T(t)Pe(t), \qquad P = P^T > 0 \qquad (3.64)$$

Stability is verified when the derivative of the Lyapunov function (2.39) is strictly negative:

$$\dot{V}\big(e(t)\big) \leq 0 \qquad (3.65)$$

Such that:

$$\dot{V}\big(e(t)\big) = e^T(t)\Bigg(\sum_{i=0}^{q\leq 2n^2} h_i\big(\xi(t)\big)(A_0^TP + PA_0 - C^TL^TP - PLC) + q(\mathcal{A}_i^TP$$

$$+ P\mathcal{A}_i))\Bigg)e(t) \qquad (3.66)$$

The stability of the estimation error dynamics is guaranteed if the time derivative of the Lyapunov function given by equation (3.66) is negative definite, which implies that the following linear matrix inequalities (LMIs) must be satisfied in a time-independent manner**:**

$$A_0^TP + PA_0 - C^TL^TP - PLC + q(\mathcal{A}_i^TP + P\mathcal{A}_i) < 0 \qquad (3.67)$$

In order to express inequality (3.67) in LMI form, we perform a change of variable by setting

$$N = PL \tag{3.68}$$

The corresponding LMI conditions are then obtained as follows, with the introduction of a parameter α representing the convergence rate.

Theorem: The system defined by the dynamics equation is asymptotically stable if there exists a positive definite symmetric matrix $P = P^T > 0$, such that the following inequality is satisfied

$$A_0^T P + P A_0 + q(\mathcal{A}_i^T P + P\mathcal{A}_i) - C^T N^T - NC + 2\alpha P < 0 \tag{3.69}$$

Where α represents a constant characterizing the time decay, for all $= 1, \dots, 2q$.

In this context, the observer's gain can be expressed in the following form: $L_0 = P^{-1}N$.

Our states:

$n = 3$

Measured outputs:

$y = 2$

Lyapunov matrix:

$$P \in \mathbb{R}^{3x3}$$

Symmetric positive definite

Variable matrix:

$$Y \in \mathbb{R}^{3x2}$$

Gain matrix:

$$L = P^{-1}Y \quad \Rightarrow K \in \mathbb{R}^{3x2}$$

9.4. Numerical implementation of the observer:

Result obtained by MATLAB calculations taking this motor parameters:

$R_s = 0.5\,\Omega,$

$L_s = 0.005\,H,$

$\psi_{pm} = 0.1\,Wb,$

$p = 4,$

$J = 0.01\,kg \cdot m^2,$

$B = 0.001\,N \cdot m \cdot s/rad$

$T_L = 1;$

common observer gain L:

$$1.0e+05 * \begin{bmatrix} 0.0154 & 0.0000 \\ -0.0000 & 0.0138 \\ 0.0000 & -4.6903 \end{bmatrix}$$

Lyapunov Matrix P:

$$1.0e+06 * \begin{bmatrix} 0.0312 & 0.0000 & 0.0000 \\ 0.0000 & 1.2806 & 0.0336 \\ 0.0000 & 0.0336 & 0.0010 \end{bmatrix}$$

matrix Y:

$$1.0e+08 * \begin{bmatrix} 0.4785 & 0.0000 \\ 0.0000 & 1.8947 \\ 0.0000 & -0.0178 \end{bmatrix}$$

The MATLAB script that calculates these matrices is in the Annex section.

Conclusion

This chapter presented the development of a **sensorless Field-Oriented Control (FOC)** strategy for the PMSM based on the **MVT-NLS observer**. The observer was designed to estimate the unmeasured motor states, particularly the **rotor speed and electrical position**. A key advantage of the MVT-NLS observer is its use of a **single global observer gain**, designed to ensure adequate estimation performance for all possible motor operating states within the predefined state boundaries. This allows the observer to handle the nonlinear variations of the PMSM model while maintaining a unified observer structure.

Chapter 4: Backstepping Controller

This chapter presents the design of the **Backstepping controller**. the Backstepping control strategy is developed to regulate the motor speed and the d- and q-axis currents. Then, the generated reference voltages are transformed into the stationary reference frame and applied to the **SVPWM** algorithm to generate the switching signals required for the three-phase inverter.

1. Backstepping Controller:

Backstepping control is a nonlinear control technique widely used for complex dynamic systems such as PMSM drives, robotic systems, and power converters [30]. It is based on Lyapunov stability theory and allows the controller to guarantee global system stability while handling nonlinearities, parameter variations, and external disturbances effectively. Because of its recursive design procedure, backstepping provides precise tracking performance and strong robustness in high-performance applications.

Compared with the classical PI controller, backstepping offers faster dynamic response, better robustness against uncertainties, and improved stability for nonlinear systems, whereas PI controllers are simpler but mainly suitable for linear or slowly varying systems. In motor drive applications, PI controllers may suffer from overshoot and degraded performance under parameter variations, while backstepping maintains accurate tracking and stable behavior [31].

1.1. Defined the speed tracking errors:

$$e_\omega = \omega_{ref} - \widehat{\omega_e} \tag{4.1}$$

We define the Lyapunov function [32]:

$$V_1 = \frac{1}{2} e_\omega^2 \tag{4.2}$$

We differentiate:

$$\dot{V}_1 = e_\omega \dot{e_\omega} \tag{4.3}$$

The derivative of equation (4.1) is:

$$\dot{e_\omega} = \frac{de_\omega}{dt} = \dot{\omega}_{ref} - \dot{\widehat{\omega_e}} \tag{4.4}$$

the observer equation:

$$\dot{\widehat{\omega_e}} = \frac{3p^2}{2J} \psi_{pm} \widehat{i_q} - \frac{B}{J} \widehat{\omega_e} - cT_L + observer\ correction \tag{4.5}$$

for controller derivation we temporarily group the observer correction into bounded term d_ω the equation (4.5) becomes:

$$\dot{\widehat{\omega_e}} = \frac{3p^2}{2J}\psi_{pm}\widehat{\imath_q} - \frac{B}{J}\widehat{\omega_e} - \frac{p}{J}T_L + d_\omega \tag{4.6}$$

Substitute into the equation (4.4):

$$\dot{e_\omega} = \dot{\omega}_{ref} - \frac{3p^2}{2J}\psi_{pm}\widehat{\imath_q} + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L - d_\omega \tag{4.7}$$

we choose $\widehat{\imath_q}^*$ To stabilize e_ω:

The virtual control $\widehat{\imath_q}^*$:

$$\widehat{\imath_q}^* = \frac{2J}{3p^2\psi_{pm}}\left(K_\omega e_\omega + \dot{\omega}_{ref} + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L\right) \tag{4.8}$$

With:

$$K_\omega > 0$$

the current i_q error e_{i_q}:

$$e_{i_q} = \widehat{\imath_q} - \widehat{\imath_q}^* \quad \Rightarrow \tag{4.9}$$

$$\widehat{\imath_q} = e_{i_q} + \widehat{\imath_q}^* \tag{4.10}$$

Substitute (4.8) into (4.7):

$$\dot{e_\omega} = \dot{\omega}_{ref} - \frac{3p^2}{2J}\psi_{pm}\left(e_{i_q} + \widehat{\imath_q}^*\right) + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L - d_\omega \tag{4.11}$$

Distribute:

$$\dot{e_\omega} = \dot{\omega}_{ref} - \frac{3p^2}{2J}\psi_{pm}e_{i_q} - \frac{3p^2}{2J}\psi_{pm}\widehat{\imath_q}^* + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L - d_\omega \tag{4.12}$$

Substitute (4.8) into (4.12):

$$\dot{e_\omega} = \dot{\omega}_{ref} - \frac{3p^2}{2J}\psi_{pm}e_{i_q} - \left(K_\omega e_\omega + \dot{\omega}_{ref} + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L\right) + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L - d_\omega \tag{4.13}$$

Distribute minus sign:

$$\dot{e_\omega} = \dot{\omega}_{ref} - \frac{3p^2}{2J}\psi_{pm}e_{i_q} - K_\omega e_\omega - \dot{\omega}_{ref} - \frac{B}{J}\widehat{\omega_e} - \frac{p}{J}T_L + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L - d_\omega \tag{4.14}$$

$$\dot{e_\omega} = -K_\omega e_\omega - \frac{3p^2}{2J}\psi_{pm}e_{i_q} - d_\omega \tag{4.15}$$

And this is the final speed error.

If:

$$e_{i_q} \to 0 \tag{4.16}$$

$$\dot{e_\omega} = -K_\omega e_\omega \tag{4.17}$$

Thus:

$$e_\omega \to 0 \tag{4.18}$$

So, now the speed stabilization depends on the current tracking.[32]

Lyapunov Extended Function:

$$V_2 = V_1 + \frac{1}{2} e_{i_q}^2 \tag{4.19}$$

Thus:

$$V_2 = \frac{1}{2} {e_\omega}^2 + \frac{1}{2} e_{i_q}^2 \tag{4.20}$$

1.2. Define e_{i_q} the q-axis tracking current error:

$$e_{i_q} = \widehat{i_q} - \widehat{i_q}^* \tag{4.21}$$

The Extended Lyapunov Function:

$$V_2 = \frac{1}{2} e_\omega^2 + \frac{1}{2} e_{i_q}^2 \tag{4.22}$$

Differentiate:

$$\dot{V}_2 = e_\omega \dot{e_\omega} + e_{i_q} \dot{e_{i_q}} \tag{4.23}$$

We already have:

$$\dot{e_\omega} = -K_\omega e_\omega - \frac{3p^2}{2J} \psi_{pm} e_{i_q} - d_\omega \tag{4.24}$$

Substitute into (4.23):

$$\dot{V}_2 = e_\omega \left(-K_\omega e_\omega - \frac{3p^2}{2J} \psi_{pm} e_{i_q} - d_\omega \right) + e_{i_q} \dot{e_{i_q}} \tag{4.25}$$

Thus:

$$\dot{V}_2 = -K_\omega e_\omega^2 - \frac{3p^2}{2J} \psi_{pm} e_{i_q} e_\omega - e_\omega d_\omega + e_{i_q} \dot{e_{i_q}} \tag{4.26}$$

We have:

$$e_{i_q} = \widehat{\imath_q} - \widehat{\imath_q}^* \tag{4.27}$$

Thus:

$$\dot{e_{i_q}} = \dot{\widehat{\imath_q}} - \dot{\widehat{\imath_q}}^* \tag{4.28}$$

Observer q axis dynamics:

$$\dot{\widehat{\imath_q}} = -\frac{R_s}{L_s}\widehat{\imath_q} - \widehat{\omega_e}\widehat{\imath_d} - \frac{\psi_{pm}}{L_s}\widehat{\omega_e} + \frac{1}{L_s}vq + d_q \tag{4.29}$$

Where d_q includes observer correction terms/ bounded uncertainties [32].

Substitute (4.29) into (4.28):

$$\dot{e_{i_q}} = -\frac{R_s}{L_s}\widehat{\imath_q} - \widehat{\omega_e}\widehat{\imath_d} - \frac{\psi_{pm}}{L_s}\widehat{\omega_e} + \frac{1}{L_s}vq + d_q - \dot{\widehat{\imath_q}}^* \tag{4.30}$$

Substitute (4.30) into the Lyapunov derivative (19):

$$\dot{V}_2 = -K_\omega e_\omega^2 - \frac{3p^2}{2J}\psi_{pm}e_{i_q}e_\omega - e_\omega d_\omega + e_{i_q}\left(-\frac{R_s}{L_s}\widehat{\imath_q} - \widehat{\omega_e}\widehat{\imath_d} - \frac{\psi_{pm}}{L_s}\widehat{\omega_e} + \frac{1}{L_s}vq + d_q - \dot{\widehat{\imath_q}}^*\right) \tag{4.31}$$

Backstepping control objective:

$$\dot{V}_2 < 0 \tag{4.31}$$

Choose control law vq:

We want to choose control law that:

- Cross terms canceled
- Damping introduced

So,

$$vq = L_s\left[-K_{i_q}e_{i_q} + \frac{3p^2}{2J}\psi_{pm}e_\omega + \frac{R_s}{L_s}\widehat{\imath_q} + \widehat{\omega_e}\widehat{\imath_d} + \frac{\psi_{pm}}{L_s}\widehat{\omega_e} + \dot{\widehat{\imath_q}}^*\right] \tag{4.32}$$

Where:

$$K_{i_q} > 0$$

Substitute (4.32) into (4.30):

$$\dot{e_{i_q}} = -\frac{R_s}{L_s}\widehat{i_q} - \widehat{\omega_e}\widehat{i_d} - \frac{\psi_{pm}}{L_s}\widehat{\omega_e}$$
$$+\frac{1}{L_s}L_s\left[-K_{i_q}e_{i_q} + \frac{3p^2}{2J}\psi_{pm}e_\omega + \frac{R_s}{L_s}\widehat{i_q} + \widehat{\omega_e}\widehat{i_d} + \frac{\psi_{pm}}{L_s}\widehat{\omega_e} + \dot{\widehat{i_q}}^*\right] + d_q$$
$$-\dot{\widehat{i_q}}^* \quad (4.33)$$

After simplify:

$$\dot{e_{i_q}} = -K_{i_q}e_{i_q} + \frac{3p^2}{2J}\psi_{pm}e_\omega + d_q \quad (4.34)$$

Final Lyapunov Function:

We substitute (4.30) into (4.31):

$$\dot{V}_2 = -K_\omega e_\omega^2 - \frac{3p^2}{2J}\psi_{pm}e_{i_q}e_\omega - e_\omega d_\omega + e_{i_q}\left(-K_{i_q}e_{i_q} + \frac{3p^2}{2J}\psi_{pm}e_\omega + d_q\right) \quad (4.35)$$

Distribute:

$$\dot{V}_2 = -K_\omega e_\omega^2 - \frac{3p^2}{2J}\psi_{pm}e_{i_q}e_\omega - e_\omega d_\omega - K_{i_q}e_{i_q}^2 + \frac{3p^2}{2J}\psi_{pm}e_\omega e_{i_q} + e_{i_q}d_q \quad (4.36)$$

After simplifying:

$$\dot{V}_2 = -K_\omega e_\omega^2 - K_{i_q}e_{i_q}^2 - e_\omega d_\omega + e_{i_q}d_q \quad (4.37)$$

Without disturbances:

$$\dot{V}_2 = -K_\omega e_\omega^2 - K_{i_q}e_{i_q}^2 < 0 \quad (4.38)$$

1.3. Define e_{i_d} the d-axis tracking current error:

For surface PMSM:[33]

$i_d{}^* = 0$

- For maximum torque per ampere
- Decoupling flux control
- Standard FOC operation

Since we are sensorless

$\widehat{i_d} \to 0$

As earlier we define the tracking d axis current error as:

$$e_{i_d} = \widehat{i_d} - i_d{}^* \quad (4.39)$$

$$e_{i_d} = \widehat{i_d} \quad (4.40)$$

Extended Lyapunov Function:

$$V_3 = \frac{1}{2}e_\omega^2 + \frac{1}{2}e_{i_q}^2 + \frac{1}{2}e_{i_d}^2 \tag{4.41}$$

And this represents total controller energy

We differentiate:

$$\dot{V}_3 = e_\omega \dot{e_\omega} + e_{i_q}\dot{e_{i_q}} + e_{i_d}\dot{e_{i_d}} \tag{4.42}$$

$$\dot{e_{i_d}} = \dot{\hat{\imath}_d} \tag{4.43}$$

We have:

$$\dot{\hat{\imath}_d} = -\frac{R_s}{L_s}\hat{\imath}_d + \widehat{\omega_e}\hat{\imath}_q + \frac{1}{L_s}vd + d_d \tag{4.44}$$

Where d_d includes observer correction terms.

Design d axis control law:

$$vd = L_s\left[-K_{i_d}e_{i_d} + \frac{R_s}{L_s}\hat{\imath}_d - \widehat{\omega_e}\hat{\imath}_q\right] \tag{4.45}$$

Where:

$K_{i_d} > 0$

Substitute (4.44) into (4.43):

$$\dot{e_{i_d}} = -\frac{R_s}{L_s}\hat{\imath}_d + \widehat{\omega_e}\hat{\imath}_q + \frac{1}{L_s}L_s\left[-K_{i_d}e_{i_d} + \frac{R_s}{L_s}\hat{\imath}_d - \widehat{\omega_e}\hat{\imath}_q\right] + d_d \tag{4.46}$$

After simplifying:

$$\boxed{\dot{e_{i_d}} = -K_{i_d}e_{i_d} + d_d} \tag{4.47}$$

Substitute (4.47) into (4,42) and all dynamics:

$$\dot{V}_3 = e_\omega\left(-K_\omega e_\omega - \frac{3p^2}{2J}\psi_{pm}e_{i_q} - d_\omega\right) + e_{i_q}\left(-K_{i_q}e_{i_q} + \frac{3p^2}{2J}\psi_{pm}e_\omega + d_q\right) + e_{i_d}\left(-K_{i_d}e_{i_d} + d_d\right) \tag{4.48}$$

Expand and after cancellation opposite terms:

$$\boxed{\dot{V}_3 = -K_\omega e_\omega^2 - K_{i_q}e_{i_q}^2 - K_{i_d}e_{i_d}^2 - e_\omega d_\omega + e_{i_q}d_q + e_{i_d}d_d} \tag{4.49}$$

Without disturbances:

$$d_\omega = d_q = d_d = 0 \tag{4.50}$$

$$\dot{V}_3 = -K_\omega e_\omega^2 - K_{i_q} e_{i_q}^2 - K_{i_d} e_{i_d}^2 < 0 \tag{4.51}$$

Therefore:

$$e_\omega, e_{i_q}, e_{i_d} \to 0 \tag{4.52}$$

Final sensorless backstepping control law [85]:

Virtual q-axis current:

$$\boxed{\widehat{\imath_q}^* = \frac{2J}{3p^2\psi_{pm}}\left(K_\omega e_\omega + \dot{\omega}_{ref} + \frac{B}{J}\widehat{\omega_e} + \frac{p}{J}T_L\right)} \tag{4.53}$$

q-axis voltage:

$$\boxed{vq = L_s\left[-K_{i_q}e_{i_q} + \frac{3p^2}{2J}\psi_{pm}e_\omega + \frac{R_s}{L_s}\widehat{\imath_q} + \widehat{\omega_e}\widehat{\imath_d} + \frac{\psi_{pm}}{L_s}\widehat{\omega_e} + \dot{\widehat{\imath_q}}^*\right]} \tag{4.54}$$

d-axis voltage:

$$\boxed{vd = L_s\left[-K_{i_d}e_{i_d} + \frac{R_s}{L_s}\widehat{\imath_d} - \widehat{\omega_e}\widehat{\imath_q}\right]} \tag{4.55}$$

Since $e_{i_d} = \widehat{\imath_d}$

$$\boxed{vd = L_s\left[-K_{i_d}\widehat{\imath_d} + \frac{R_s}{L_s}\widehat{\imath_d} - \widehat{\omega_e}\widehat{\imath_q}\right]} \tag{4.56}$$

❖ **Numerical implementation of the Controller:**

Calculation has been done with the following parameters:

R_s = 0.5 Ω,

L_s= 0.005 H,

ψ_{pm} = 0.1 Wb,

p = 4,

J = 0.01 k$g \cdot m^2$,

B = 0.001 $N \cdot m \cdot s/rad$,

Controller gains

K_{i_d}= 400,

K_{i_q}= 400,

K_ω= 40,

and load torque

T_L=1 $N \cdot m$.

Step 1: Compute the coefficient of $\widehat{\imath_q}^*$:

The coefficient is:

$$\frac{2J}{3p^2\psi_{pm}} = \frac{2(0.01)}{3(4)^2(0.1)} = \frac{0.02}{4.8} = 0.00416667$$

$$\frac{B}{J} = \frac{0.001}{0.01} = 0.1$$

$$\frac{p}{J}T_L = \frac{4}{0.01}(1) = 400$$

Therefore:

$$\widehat{\imath_q}^* = 0.00416667\left(40e_\omega + \dot{\omega}_{ref} + 0.1\widehat{\omega_e} + 400\right)$$

Expanding:

$$\boxed{\widehat{\imath_q}^* = 0.16667e_\omega + 0.00416667\dot{\omega}_{ref} + 0.000416667\widehat{\omega_e} + 1.6667}$$

Step 2: Compute the coefficient of vq:

$$\frac{3p^2}{2J}\psi_{pm} = \frac{3(4)^2}{2(0.01)}(0.1) = 240$$

$$\frac{R_s}{L_s} = \frac{0.5}{0.005} = 100$$

$$\frac{\psi_{pm}}{L_s} = \frac{0.1}{0.005} = 20$$

Thus:

$$vq = 0.005\left[-400e_{i_q} + 240e_\omega + 100\widehat{\imath_q} + \widehat{\omega_e}\widehat{\imath_d} + 20\widehat{\omega_e} + \dot{\widehat{\imath_q}}^*\right]$$

Multiplying by

$$L_s = 0.005$$

$$\boxed{vq = -2e_{i_q} + 1.2e_\omega + 0.5\widehat{\imath_q} + 0.005\widehat{\omega_e}\widehat{\imath_d} + 0.1\widehat{\omega_e} + 0.005\dot{\widehat{\imath_q}}^*}$$

Step 3: Compute vd:

$$vd = L_s\left[-K_{i_d}\widehat{\imath_d} + \frac{R_s}{L_s}\widehat{\imath_d} - \widehat{\omega_e}\widehat{\imath_q}\right]$$

Substituting:

$$vd = 0.005\left[-400\widehat{\imath_d} + 100\widehat{\imath_d} - \widehat{\omega_e}\widehat{\imath_q}\right]$$

Therefore:

$$\boxed{vd = -1.5e_{i_d} - 0.005\widehat{\omega_e}\widehat{\imath_q}}$$

Step 4: derivative of $\widehat{\imath_q}^*$:

$$\dot{\widehat{\imath_q}}^* = 0.00416667\left(40\dot{e_\omega} + \ddot{\omega}_{ref} + 0.1\dot{\widehat{\omega_e}}\right)$$

$$\dot{e_\omega} = \dot{\omega}_{ref} - \dot{\widehat{\omega_e}}$$

Substituting:

$$\dot{\widehat{\imath_q}}^* = 0.00416667\left(40\left(\dot{\omega}_{ref} - \dot{\widehat{\omega_e}}\right) + \ddot{\omega}_{ref} + 0.1\dot{\widehat{\omega_e}}\right)$$

$$\dot{\omega}_{ref} = 0 \qquad \ddot{\omega}_{ref} = 0$$

Expanding:

$$\dot{\widehat{\imath_q}}^* = 0.00416667\left(40\dot{\widehat{\omega_e}} + 0.1\dot{\widehat{\omega_e}}\right)$$

Therefore:

$$\boxed{\dot{\widehat{\imath_q}}^* = -0.16625\dot{\widehat{\omega_e}}}$$

Chapter 5: Space Vector PULS Width Modulation SVPWM

SVPWM (Space Vector Pulse Width Modulation):

1. Introduction:

Space Vector Pulse Width Modulation (SVPWM) is the modulation technique adopted in this work to control the three-phase Voltage Source Inverter (VSI) supplying the Permanent Magnet Synchronous Motor (PMSM). The primary objective of the modulation stage is to convert the constant DC-link voltage into a balanced set of three-phase AC voltages with controllable amplitude and frequency, allowing the motor to operate under the commands generated by the Field-Oriented Control (FOC) algorithm.[35]

Since the power semiconductor switches of a two-level inverter can only assume two discrete states (ON or OFF), it is impossible to generate perfectly sinusoidal voltages directly at the inverter output. Instead, the inverter produces a sequence of discrete voltage vectors whose average value over each switching period approximates the desired sinusoidal reference voltage. Pulse Width Modulation (PWM) techniques achieve this objective by appropriately controlling the switching instants of the inverter devices.[36]

Among the various PWM methods, **Space Vector Pulse Width Modulation (SVPWM)** has become the standard modulation strategy for high-performance AC motor drives. Unlike conventional Sinusoidal Pulse Width Modulation (SPWM), which independently modulates each phase using three sinusoidal reference signals, SVPWM treats the three-phase inverter as a single vector operating in the two-dimensional stationary αβ\alpha\betaαβ reference frame. The desired reference voltage is synthesized by combining the nearest inverter voltage vectors during every switching interval, thereby producing a more accurate approximation of the commanded voltage vector.[37]

The use of SVPWM provides several advantages over classical SPWM. First, it increases the utilization of the DC-link voltage by approximately **15.5%**, enabling the inverter to generate a higher fundamental output voltage without entering the overmodulation region. Secondly, the optimized switching sequence reduces the Total Harmonic Distortion (THD) of the output voltages and currents, leading to smoother electromagnetic torque and lower acoustic noise. Furthermore, the symmetrical distribution of switching events minimizes switching losses and improves the overall efficiency of the inverter. Owing to these advantages, SVPWM has become the preferred modulation technique for high-performance PMSM drives, servo systems, electric vehicles, robotics, and industrial motion-control applications.

In the proposed sensorless Backstepping-controlled PMSM drive, the nonlinear current controllers generate the reference voltages v_d and v_q in the synchronous dq reference frame. These voltages are transformed into the stationary $\alpha\beta$ frame through the inverse park

transformation, after which the SVPWM algorithm determines the appropriate inverter switching states and computes the corresponding duty cycles required to synthesize the desired voltage vector. Consequently, SVPWM forms the final stage of the control system, translating the control commands into the gate signals applied to the inverter switches.

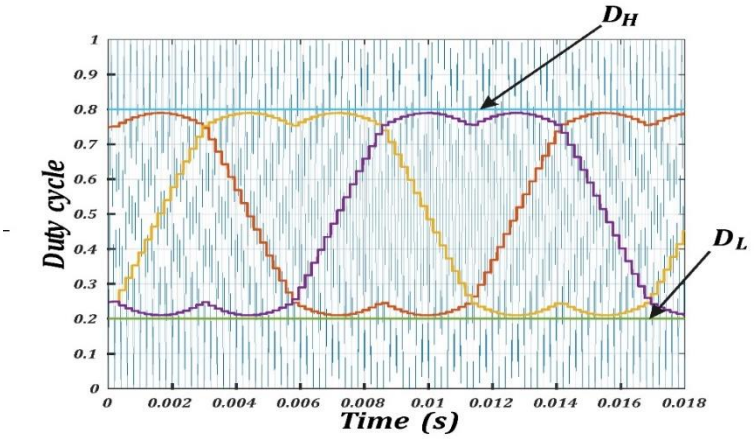


Figure 5-1: modulation and carrier signal in PWM by triangle comparison for space vector modulation and the reference

2. Small Comparison with Classical SPWM:

Table 5-1: comparison between SVPWM and SPWM [35].

Feature	SPWM	SVPWM
Principle	Compare sine with carrier	Space vector synthesis
DC bus utilization	Lower	Higher
Harmonic distortion	Higher	Lower
Switching optimization	Moderate	Better
Computational complexity	Simple	More advanced
Maximum output voltage	$0.785V_{dc}$	$0.907V_{dc}$

3. Two-Level Voltage Source Inverter (VSI) and Switching States:

The Voltage Source Inverter (VSI) constitutes the power conversion stage of the PMSM drive. Its primary function is to convert the constant DC-link voltage V_{dc} into a set of balanced three-phase AC voltages capable of driving the motor according to the voltage commands generated by the control algorithm.[39]

A conventional two-level VSI consists of three identical inverter legs corresponding to phases A, B, and C. Each leg is composed of two complementary power semiconductor switches (MOSFETs or IGBTs), connected in series across the DC bus, as illustrated in **Figure 9**. The

midpoint of each inverter leg forms one output phase connected to the stator windings of the PMSM.

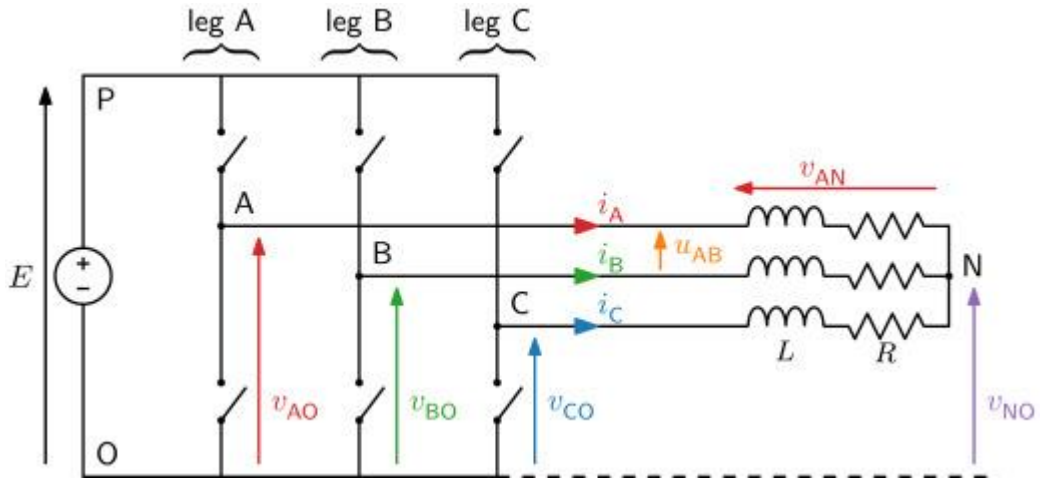


Figure 5-*2*: three-phase inverter diagram

Since the upper and lower switches of the same inverter leg are complementary devices, they must never conduct simultaneously, as this would produce a direct short circuit across the DC source, commonly referred to as **shoot-through**. Therefore, only one switch of each leg is allowed to conduct at any given instant. This complementary operation is ensured by the gate driver circuitry and the insertion of a small dead time between the switching transitions.[40]

To describe the switching configuration mathematically, a binary switching variable is assigned to each inverter leg:

$$S_a = \begin{cases} 1, upper\ switch\ ON,\ lower\ switch\ OFF \\ 0, upper\ switch\ OFF,\ lower\ switch\ ON \end{cases}$$

$$S_b = \begin{cases} 1, upper\ switch\ ON,\ lower\ switch\ OFF \\ 0, upper\ switch\ OFF,\ lower\ switch\ ON \end{cases}$$

$$S_c = \begin{cases} 1, upper\ switch\ ON,\ lower\ switch\ OFF \\ 0, upper\ switch\ OFF,\ lower\ switch\ ON \end{cases}$$

Consequently, the switching state of the entire inverter can be represented by the binary triplet

$$(S_a, S_b, S_c).$$

Since each switching variable can assume only two possible values (0 or 1), the total number of possible switching combinations is:

$$N = 2^3 = 8.$$

Thus, a two-level three-phase inverter possesses eight possible switching states. However, these eight states do not generate eight different output voltages. Instead, they produce only **six non-zero (active) voltage vectors** and **two zero voltage vectors**.[41]

The two zero vectors correspond to the switching states:

$$(0,0,0) \text{ and } (1,1,1).$$

for which all three motor terminals are connected to the same DC potential. As a result, the phase-to-phase voltages become zero, and no effective voltage is applied to the stator windings.

The remaining six switching combinations generate six different non-zero voltage vectors having identical magnitude but different spatial orientations. These vectors are uniformly distributed every 60∘60^\circ60∘ in the stationary αβ\alpha\betaαβ reference frame, forming the well-known hexagonal space-vector diagram used in SVPWM.[42]

The switching states and their corresponding voltage vectors are summarized in table 2.

Table 5-2: Switching states of a two-level three-phase VSI

State	Switching	Voltage Vector
V_0	000	zero
V_1	100	active
V_2	110	active
V_3	010	active
V_4	011	active
V_5	001	active
V_6	101	active
V_7	111	zero

The six active vectors represent the only voltage levels that the inverter can physically generate. Therefore, the fundamental objective of the SVPWM algorithm is to synthesize the desired reference voltage vector by properly combining the nearest active vectors together with the zero vectors during each switching period. The duration of each switching state is calculated such that the average output voltage over one PWM period is equal to the reference voltage commanded by the controller.

Why are there only six active vectors: At first glance, the inverter has eight switching states, so one might expect eight different output voltage vectors. However, the states (000) and (111) produce identical line-to-line voltages.

To understand this, consider the phase voltages with respect to the negative DC bus:

For the state: (000), all three output terminals are connected to the negative DC rail:

$$V_a = V_b = V_c = 0 \tag{5.1}$$

Therefore,

$$V_{ab} = V_a - V_b = 0, \tag{5.2}$$

$$V_{bc} = V_b - V_c = 0, \tag{5.3}$$

$$V_{ca} = V_c - V_a = 0. \tag{5.4}$$

Similarly, for the state (111), all three terminals are connected to the positive DC rail:

$$V_a = V_b = V_c = V_{dc}, \tag{5.6}$$

which also gives:

$$V_{ab} = V_{bc} = V_{ca} = 0. \tag{5.7}$$

Although the absolute potentials differ, the motor windings are excited only by the **line-to-line voltages**. Consequently, these two switching states produce the same electrical effect on the machine and are therefore called **zero vectors**.

In the proposed Backstepping-controlled PMSM drive, the controller computes the reference voltages $\boldsymbol{v_d^*}$ and $\boldsymbol{v_q^*}$. After transformation into the stationary reference frame, the SVPWM algorithm selects, at every switching interval, one of the eight switching states listed above. By rapidly alternating among these states according to calculated switching times, the inverter synthesizes a rotating voltage vector that closely follows the reference generated by the controller, ensuring accurate torque and flux regulation of the PMSM.

4. Mathematical Representation of the Space Voltage Vectors:

The eight switching states of the two-level Voltage Source Inverter generate discrete voltage vectors in the stationary αβ\alpha\betaαβ reference frame. These vectors constitute the basis of the Space Vector Pulse Width Modulation (SVPWM) algorithm, whose objective is to approximate a continuously rotating reference voltage vector by properly combining the available inverter vectors.[43]

To determine the location of each switching vector, the inverter output voltages must first be expressed in terms of the switching functions.

4.1. Phase Voltages as a Function of the Switching States:

For a two-level inverter supplied by a DC-link voltage V_{dc}, the switching variables:

$$S_a, S_b, S_c \in 0,1 \tag{5.8}$$

completely determine the phase voltages

The pole voltages with respect to the negative DC bus are:

$$V_{a0} = S_a * V_{dc}, \tag{5.9}$$

$$V_{b0} = S_b * V_{dc}, \tag{5.10}$$

$$V_{c0} = S_c * V_{dc}. \tag{5.11}$$

However, the PMSM windings are connected in a balanced star (Y) configuration without a neutral connection. Therefore, the motor is excited by the phase voltages measured with respect to the floating neutral point NNN, rather than with respect to the negative DC terminal. The neutral potential is obtained as:[44]

$$V_{N0} = \frac{V_{a0} + V_{b0} + V_{c0}}{3} \tag{5.12}$$

Hence, the actual stator phase voltages become:

$$V_a = V_{a0} - V_{N0}, \tag{5.13}$$

$$V_b = V_{b0} - V_{N0}, \tag{5.14}$$

$$V_c = V_{c0} - V_{N0}. \tag{5.15}$$

Substituting the expressions of the pole voltages yields:

$$V_a = \frac{V_{dc}}{3}(2S_a - S_b - S_c), \tag{5.16}$$

$$V_b = \frac{V_{dc}}{3}(2S_b - S_a - S_c), \tag{5.17}$$

$$V_c = \frac{V_{dc}}{3}(2S_c - S_a - S_b). \tag{5.18}$$

These equations establish the relationship between the inverter switching state and the three phase voltages applied to the PMSM stator.[45]

❖ **Clarke Transformation:**

The three-phase voltages are transformed into the stationary orthogonal reference frame using the Clarke transformation. This transformation converts the three coupled phase quantities into two independent orthogonal components, considerably simplifying the mathematical representation of the inverter output.

The Clarke transformation is given by:

$$\begin{bmatrix} V_\alpha \\ V_\beta \\ V_0 \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{5.19}$$

After substituting the previous expressions for the phase voltages and simplifying, the stationary-frame voltages become:

$$V_\alpha = \frac{2}{3}V_{dc}\left(S_a - \frac{1}{2}S_b - \frac{1}{2}S_c\right), \tag{5.20}$$

$$V_\beta = \frac{V_{dc}}{\sqrt{3}}(S_b - S_c). \tag{5.21}$$

These two equations are fundamental because every switching state of the inverter corresponds to a unique point in the αβ\alpha\betaαβ plane.[46]

❖ **Space Voltage Vector:**

Instead of handling the two orthogonal voltage components separately, they are combined into a single complex vector known as the **space voltage vector**:

$$\vec{V}_{ref} = V_\alpha + jV_\beta \tag{5.22}$$

Where:

- V_α is the horizontal component,
- V_β is the vertical component,
- $\sqrt{j} = -1$.

The magnitude of the s²pace vector is:

$$V_{ref} = \sqrt{{V_\alpha}^2 + {V_\beta}^2} \tag{5.23}$$

while its angular position in the stationary reference frame is:

$$\theta_s = \tan^{-1}\frac{V_\beta}{V_\alpha} \tag{5.24}$$

Consequently, each switching state corresponds to a vector having a specific magnitude and orientation in the $\alpha\beta$ plane.

❖ **Coordinates of the Six Active Voltage Vectors:**

Using the previous equations, the coordinates of each active vector can now be calculated explicitly.[47]

Vector V_1 (100)

For:

$$S_a = 1, \qquad S_b = 0, \qquad S_c = 0,$$

One obtains:

$$V_\alpha = \frac{2}{3}V_{dc}, \qquad V_\beta = 0.$$

Therefore,

$$V_1 = \left(\frac{2}{3}V_{dc}, 0\right). \tag{5.25}$$

Vector $V_2(110)$

For:

$$\mathrm{S_a} = 1, \qquad \mathrm{S_b} = 1, \qquad \mathrm{S_c} = 0,$$

$$V_\alpha = \frac{1}{3}V_{dc}, \qquad V_\beta = \frac{\sqrt{3}}{3}V_{dc}.$$

$$V_2 = \left(\frac{1}{3}V_{dc}, \frac{\sqrt{3}}{3}V_{dc}\right). \tag{5.26}$$

Vector $V_3(010)$:

$$V_3 = \left(-\frac{1}{3}V_{dc}, \frac{\sqrt{3}}{3}V_{dc}\right). \tag{5.27}$$

Vector $V_4(011)$:

$$V_4 = \left(-\frac{2}{3}V_{dc}, 0\right). \tag{5.28}$$

Vector $V_5(001)$:

$$V_5 = \left(-\frac{1}{3}V_{dc}, -\frac{\sqrt{3}}{3}V_{dc}\right) \tag{5.29}$$

Vector $V_6(101)$:

$$V_6 = \left(\frac{1}{3}V_{dc}, -\frac{\sqrt{3}}{3}V_{dc}\right) \tag{5.30}$$

❖ **Magnitude of the Active Vectors:**

The magnitude of every active vector is identical. For example, considering $V2$

$$| V_2 | = \sqrt{\left(\frac{1}{3}V_{dc}\right)^2 + \left(\frac{\sqrt{3}}{3}V_{dc}\right)^2} \tag{5.31}$$

$$| V_2 | = \sqrt{\frac{1}{9}{V_{dc}}^2 + \frac{3}{9}{V_{dc}}^2} \tag{5.32}$$

$$| V_2 | = \frac{2}{3}V_{dc} \tag{5.33}$$

The same result is obtained for all six active vectors:

$$| V_1 | = | V_2 | = \cdots \ldots = | V_6 | = \frac{2}{3}V_{dc} \tag{5.34}$$

Thus, all active vectors have the same magnitude and differ only in their angular position.

4.2. Formation of the Space Vector Hexagon:

Since every active vector has the same magnitude and each consecutive vector is separated by an electrical angle of: 60°.

their endpoints form the vertices of a regular hexagon in the αβ\alpha\betaαβ plane, as illustrated in **Figure 10[48]**

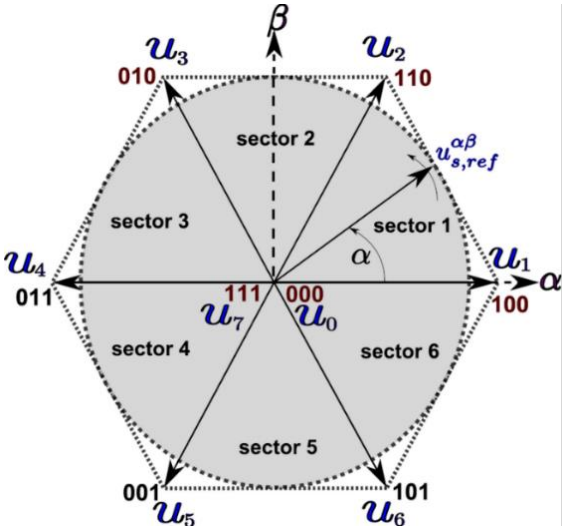


Figure 5-3: Representation of tow-level three-phase inverter voltage space vector in complex \alpha\beta plan [39].

The six vectors are located at:

$$0^o, 60^o, 120^o, 180^o, 240^o, 300^o.$$

The two zero vectors,

$$V_0 = (000),$$

and

$$V_7 = (111),$$

are both located at the origin of the coordinate system because they produce zero line-to-line voltage.

This hexagonal distribution constitutes the geometric foundation of the SVPWM algorithm. At every switching period, the desired reference voltage vector is synthesized by combining the two adjacent active vectors of the corresponding sector together with one or both zero vectors.

❖ **Sector Determination:[49]**

After transforming the reference voltages from the synchronous dq reference frame into the stationary αβ\alpha\betaαβ frame, the resulting reference voltage vector.

$$\vec{V}_{ref} = V_\alpha + jV_\beta \tag{5.35}$$

can assume any position inside the space-vector hexagon. Since the inverter is capable of generating only six active voltage vectors, the reference vector cannot be applied directly. Instead, the SVPWM algorithm first determines the sector in which the reference vector is located and then synthesizes it using the two adjacent active vectors that bound this sector.

The determination of the sector is therefore the first computational step of the SVPWM algorithm and is repeated during every switching period.

❖ **Division of the $\alpha\beta$ Plane:**

The six active voltage vectors divide the stationary reference frame into six identical sectors, each occupying an electrical angle of: 60^o

Since a complete revolution corresponds to: 360^o

the angular width of each sector is:

$$\frac{360}{60} = 60$$

The sectors are numbered counterclockwise beginning from the positive α-axis, as shown in **Figure 5-3:**

Table 5-3: how the six active voltage vectors divide the stationary reference frame into six identical sectors.[47]

Sector	Angular Interval
$sector1$	$0^o \leq \theta_s < 60$
$sector2$	$60^o \leq \theta_s < 120^o$
$sector3$	$120^o \leq \theta_s < 180^o$
$sector4$	$180^o \leq \theta_s < 240^o$
$sector5$	$240^0 \leq \theta_s < 300^o$
$sector6$	$300^o \leq \theta_s < 360^o$

❖ **Determination of the Reference Vector Angle:**

The angular position of the reference vector is obtained from its orthogonal components using the inverse tangent function:

$$\theta_s = atan2(V_\beta, V_\alpha) \tag{5.36}$$

Where:

- V_α is the horizontal component,
- V_β is the vertical component.

The function $atan2(\cdot)$ is preferred over the conventional tangent function because it automatically determines the correct quadrant of the reference vector, avoiding ambiguities that arise from the standard inverse tangent.

If the calculated angle is negative,

$$\theta s < 0, \tag{5.37}$$

it is converted into the interval

$$0 \leq \theta s < 360^o \tag{5.38}$$

by adding one complete revolution,

$$\theta s = \theta s + 360^o \tag{5.39}$$

This normalization ensures that the reference vector always lies within the operating range of the SVPWM algorithm.

❖ **Sector Identification:[47]**

Once the angle of the reference vector has been determined, the corresponding sector is easily obtained by

$$sector = \left\lfloor \frac{\theta s}{360} \right\rfloor + 1 \tag{5.40}$$

Where the $\lfloor . \rfloor$ denotes the floor operation (the result is integer).

Example 1:

Suppose:

$$V_\alpha = 120\,V, \tag{5.41}$$

$$V_\beta = 80\,V. \tag{5.42}$$

The reference vector angle becomes:

$$\theta_s = \tan^{-1}\frac{80}{120} = 33.69 \tag{5.43}$$

Since,

$$0 < 33.69 < 60, \tag{5.44}$$

the reference vector belongs to:

Sector 1:

Example 2:

Suppose:

$V_\alpha = -50\,V,$

$$V_\beta = 140\,V.$$

The reference vector angle becomes:

$$\theta_s = \tan^{-1}\frac{140}{-50} = 109.65$$

Since,

$$60 < 109.65 < 120,$$

the reference vector belongs to:

Sector 2:

❖ **Adjacent Active Vectors:**

After determining the sector, the SVPWM algorithm identifies the two neighboring active vectors that surround the reference vector. These vectors are used to synthesize the desired average voltage during one switching period.

The corresponding active vectors for each sector are summarized in **Table 4-5]**

Table 5-4: shows the active tow vectors that surrounding the reference vector.

Sector	First Active Vector	Second Active Vector
1	V_1	V_2
2	V_2	V_3
3	V_3	V_4
4	V_4	V_5
5	V_5	V_6
6	V_6	V_1

For instance, if the reference vector is located inside **Sector 3**, it is synthesized by appropriately combining the vectors V_3, V_4, together with the zero vectors V_0 and V_7.

❖ **Sector Determination Without Trigonometric Functions:[47]**

Although the angular method described above is intuitive and widely used for theoretical analysis, practical implementations on digital signal processors and microcontrollers often avoid the computation of trigonometric functions because of their relatively high computational cost.

Instead, the sector can be identified directly from the signs and relative magnitudes of the three auxiliary variables.

$$X = V_\beta, \tag{5.45}$$

$$Y = \frac{\sqrt{3}}{2} V_\alpha - \frac{1}{2} V_\beta \tag{5.46}$$

$$Z = -\frac{\sqrt{3}}{2} V_\alpha - \frac{1}{2} V_\beta \tag{5.47}$$

The signs of X, Y, and Z uniquely determine the sector of the reference vector. This method eliminates the need for evaluating inverse trigonometric functions and is therefore preferred in high-speed embedded implementations.

However, for analytical developments and mathematical derivations, the angle-based method using the $atan2(\cdot)$ function remains more convenient and provides a clearer geometrical interpretation.

4.3. Relationship with the Proposed PMSM Drive:[48]

In the proposed sensorless Backstepping-controlled PMSM drive, the nonlinear current controller generates the reference voltages:

$$v_d^* \text{ and } v_q^*$$

These voltages are transformed into the stationary frame using the inverse Park transformation

$$V_\alpha = v_d^* \cos\theta_e - v_q^* \sin\theta_e \tag{5.48}$$

$$V_\beta = v_d^* \sin\theta_e + v_q^* \cos\theta_e \tag{5.49}$$

where θ_e is the electrical rotor position estimated by the observer.

The SVPWM algorithm then determines the sector of the reference voltage vector according to the previously described procedure. Once the sector is known, the switching durations T_1, T_2 , and T_0 are computed to synthesize the commanded voltage during the next PWM period.

Key Point:

At the end of this stage, the controller has determined:

- The angular position of the reference voltage vector,
- The corresponding sector,
- The two adjacent active vectors that will be used for voltage synthesis.

The only remaining task is to calculate the duration for which each vector must be applied during one switching period. This is the core principle of SVPWM and is developed in the following subsection.

5. Voltage Vector Synthesis and Derivation of the Switching Times:

The two-level Voltage Source Inverter can generate only six active voltage vectors and two zero vectors. Consequently, the desired reference voltage vector cannot be produced directly. Instead, the principle of SVPWM consists of synthesizing the reference vector by applying the two adjacent active vectors together with the zero vectors during each switching period.[48]

Because the switching frequency is much higher than the fundamental electrical frequency of the motor, the stator windings respond to the **average voltage** over one switching interval rather than to the instantaneous switching states. Therefore, if the average voltage during one PWM period is equal to the desired reference voltage, the PMSM behaves as if it were supplied by a continuous sinusoidal voltage.

6. Average Voltage Principle:[48]

Consider a switching period T_s. During this interval, the inverter successively applies:

- the first active vector for a duration T_1,
- the second active vector for a duration T_2,
- the zero vector for the remaining time T_3.

Since these three intervals completely occupy one switching period,

$$T_s = T_1 + T_2 + T_0 \tag{5.50}$$

This equation is known as the **switching-time constraint**.

The average voltage generated during one PWM period is therefore

$$\vec{V}_{ref} T_s = \vec{V}_i T_1 + \vec{V}_{i+1} T_2 + \vec{V}_0 T_0 \tag{5.51}$$

Where:

- $\vec{V}_{ref}$ is the desired reference voltage vector,
- $\vec{V}_i$ and $\vec{V}_{i+1}$ are the two adjacent active vectors,
- $\vec{V}_0$ denotes either of the zero vectors (V_0 or V_7).

Since the zero vectors produce zero output voltage,

$\vec{V}_0 = 0$,

the average voltage equation becomes:

$$\vec{V}_{ref} T_s = \vec{V}_i T_1 + \vec{V}_{i+1} T_2 \tag{5.52}$$

This equation is the mathematical foundation of the SVPWM algorithm.

❖ **Voltage Synthesis in Sector 1:**

To derive the switching times, consider that the reference vector lies inside **Sector 1**. In this region, the adjacent active vectors are:[48]

$$V_1 = \left(\frac{2}{3} V_{dc}, 0\right) \tag{5.52}$$

$$V_2 = \left(\frac{1}{3} V_{dc}, \frac{\sqrt{3}}{3} V_{dc}\right). \tag{5.53}$$

The reference voltage vector is:

$$\vec{V}_{ref} = \left(V_\beta, V_\alpha\right) \tag{5.54}$$

Substituting these vectors into the average voltage equation yields:

$$(V_\beta, V_\alpha)T_s = \vec{V}_1 T_1 + \vec{V}_2 T_2 \tag{5.55}$$

Since vectors are equal only if their components are equal, the equation is separated into the α and β-axes.

❖ **Projection on the α -Axis:**

The horizontal components give :

$$V_\alpha T_s = \frac{2}{3} V_{dc} T_1 + \frac{1}{3} V_{dc} T_2 \tag{5.56}$$

Factoring out $\frac{1}{3} V_{dc}$

$$V_\alpha T_s = \frac{1}{3} V_{dc} (2T_1 + T_2) \tag{5.57}$$

❖ **Projection on the β -Axis:**

The horizontal components give:

$$V_\beta T_s = \frac{\sqrt{3}}{3} V_{dc} T_2 \tag{5.58}$$

Solving for T_2

$$T_2 = \frac{\sqrt{3}\, V_\beta T_s}{V_{dc}} \tag{5.59}$$

This is the first switching-time equation.

❖ **Derivation of T_1**:

Substituting the expression of T_1 into the α -axis equation,

$$V_\alpha T_s = \frac{1}{3} V_{dc} (2T_1 + T_2) \tag{5.60}$$

$$2T_1 = \frac{3V_\alpha T_s}{V_{dc}} - T_2 \tag{5.61}$$

$$2T_1 = \frac{3V_\alpha T_s}{V_{dc}} - \frac{\sqrt{3}\, V_\beta T_s}{V_{dc}} \tag{5.62}$$

$$T_1 = \frac{\sqrt{3} T_s}{V_{dc}} \left(V_\alpha - \frac{1}{\sqrt{3}} V_\beta \right) \tag{5.63}$$

which is the well-known expression for Sector 1.

❖ **Computation of the Zero Vector Time:**

Once the active vector durations have been calculated, the remaining switching interval is assigned to the zero vectors.[48]

Using the switching-time constraint,

$$T_s = T_1 + T_2 + T_0 \tag{5.64}$$

the zero-vector duration becomes:

$$T_0 = T_s - T_1 - T_2 \tag{5.65}$$

The zero-vector interval does not contribute to the average output voltage, but it plays a fundamental role in reducing harmonic distortion and obtaining symmetrical PWM waveforms.

7. General Expressions for All Sectors:[48]

The previous derivation was performed for Sector 1. The same procedure can be applied to the remaining five sectors. Although the adjacent active vectors change from one sector to another, the mathematical principle remains identical.

A convenient generalized formulation is:

$$T_1 = mT_s \sin\left(\frac{\pi}{3} - \alpha\right) \tag{5.66}$$

$$T_2 = mT_s \sin(\alpha) \tag{5.67}$$

Where:

- m is the modulation index,
- α is the angle of the reference vector measured from the beginning of the current sector.

The zero-vector duration is again

$$T_0 = T_s - T_1 - T_2 \tag{5.68}$$

These generalized expressions are widely implemented in digital signal processors because they are independent of the sector number.

8. Physical Interpretation of the Switching Times:

The switching durations have a clear geometric meaning:

- T_1 represents the time during which the first adjacent active vector is applied.
- T_2 represents the time assigned to the second adjacent active vector.
- T_0 is the remaining interval during which one or both zero vectors are applied.

As the reference vector rotates within a sector, the durations T_1 and T_2 vary continuously while T_0 adjusts automatically so that the total switching period remains constant.

For example:

- At the beginning of a sector, T_2 is close to zero and T_1 reaches its maximum value.
- At the middle of the sector, T_1 and T_2 become approximately equal.
- Near the end of the sector, T_1 decreases toward zero while T_2 reaches its maximum.

This smooth transition ensures a continuous rotation of the synthesized voltage vector and minimizes current ripple.[48]

9. Relationship with the Proposed PMSM Drive:

In the proposed sensorless Backstepping-controlled PMSM drive, the nonlinear controller computes the voltage references v_d^* and v_q^*, which are transformed into $V_\alpha \; and \; V_\beta$. The sector determination algorithm identifies the two adjacent active vectors, after which the switching times T_1, T_2, and T_0 are calculated using the previous equations. These durations are then converted into duty cycles for the three inverter legs, generating the PWM gate signals required to produce the commanded stator voltages.

10. Duty Cycle Computation and Symmetrical Switching Sequence:

After determining the switching durations T_1, T_2, and T_0, the next step of the SVPWM algorithm consists of converting these time intervals into duty cycles for the three inverter legs. These duty cycles determine the switching instants of the power semiconductor devices during each PWM period and are directly loaded into the timer compare registers of the digital controller.

To minimize harmonic distortion and reduce switching losses, SVPWM employs a **symmetrical switching pattern** in which the zero-vector time is equally distributed at the beginning and at the end of the switching period. This symmetrical arrangement ensures that the average voltage produced during one switching interval is equal to the desired reference voltage while maintaining a balanced switching sequence.

11. Symmetrical Distribution of the Zero Vector:[49]

The zero-vector interval previously obtained as:

$$T_0 = T_s - T_1 - T_2 \tag{5.69}$$

is divided equally into two identical parts, $\frac{T_0}{2}$

which are inserted before and after the active vectors.

Consequently, the switching period is composed of:

$$\frac{T_0}{2} \rightarrow T_1 \rightarrow T_2 \rightarrow \frac{T_0}{2}$$

This symmetrical arrangement guarantees that the switching waveform possesses half-wave symmetry, considerably reducing the even-order harmonics of the inverter output voltage.

12. Symmetrical Switching Sequence:

For **Sector 1**, the adjacent active vectors are V_1 and V_2. The complete switching sequence adopted by SVPWM is:[49]

$$V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_7 \rightarrow V_2 \rightarrow V_1 \rightarrow V_0$$

Where:

- V_0= (000)
- V_1= (100)
- V_2= (110)
- V_7= (111)

This sequence exhibits three important characteristics:

- it is perfectly symmetrical with respect to the center of the switching period;
- only one inverter leg changes state at each switching transition, reducing switching losses;
- the generated output voltage contains lower harmonic distortion compared with non-symmetrical switching strategies.

The same principle is applied to all remaining sectors by replacing the active vectors with the corresponding adjacent vectors.

❖ **Calculation of the Duty Cycles**

The PWM duty cycle of each inverter leg is defined as the ratio between its conduction time and the switching period,[49]

$$D = \frac{T_{ON}}{T_s} \tag{5.70}$$

Where:

- T_{ON} is the conduction time of the upper switch,
- T_s is the switching period.

The duty cycles are therefore dimensionless quantities ranging from 0 to 1 (or from 0% to 100%).

❖ **Duty Cycle Equations for Sector 1:**

For Sector 1, the switching sequence produces the following conduction intervals:

- **Phase A:**

The upper switch of phase A remains ON during:

$$T_1 + T_2 + \frac{T_0}{2} \tag{5.71}$$

Therefore,

$$D_a = \frac{T_1 + T_2 + \frac{T_0}{2}}{T_s} \tag{5.72}$$

- **Phase B:**

The upper switch of phase B conducts during:

$$T_2 + \frac{T_0}{2} \tag{5.73}$$

Hence,

$$D_b = \frac{T_2 + \frac{T_0}{2}}{T_s} \tag{5.74}$$

- **Phase C:**

The upper switch of phase B conducts during:

$$\frac{T_0}{2}$$

Consequently,

$$D_c = \frac{\frac{T_0}{2}}{T_s} \tag{5.75}$$

These duty cycles are directly written into the PWM timer compare registers (TCR).

- Duty Cycle Equations for All Six Sectors:

As the reference vector rotates, the switching sequence changes from one sector to another. Consequently, the duty-cycle equations are obtained by simply permuting the expressions of T_1, T_2, and T_0.

Table 5-5: the duty cycle expressions of all sectors

Sector	D_a	D_b	D_c
1	$\frac{T_1 + T_2 + \frac{T_0}{2}}{T_s}$	$\frac{T_2 + \frac{T_0}{2}}{T_s}$	$\frac{\frac{T_0}{2}}{T_s}$
2	$\frac{T_1 + \frac{T_0}{2}}{T_s}$	$\frac{T_1 + T_2 + \frac{T_0}{2}}{T_s}$	$\frac{\frac{T_0}{2}}{T_s}$
3	$\frac{\frac{T_0}{2}}{T_s}$	$\frac{T_1 + T_2 + \frac{T_0}{2}}{T_s}$	$\frac{T_2 + \frac{T_0}{2}}{T_s}$
4	$\frac{\frac{T_0}{2}}{T_s}$	$\frac{T_1 + \frac{T_0}{2}}{T_s}$	$\frac{T_1 + T_2 + \frac{T_0}{2}}{T_s}$
5	$\frac{T_2 + \frac{T_0}{2}}{T_s}$	$\frac{\frac{T_0}{2}}{T_s}$	$\frac{T_1 + T_2 + \frac{T_0}{2}}{T_s}$
6	$\frac{T_1 + T_2 + \frac{T_0}{2}}{T_s}$	$\frac{\frac{T_0}{2}}{T_s}$	$\frac{T_1 + \frac{T_0}{2}}{T_s}$

This table is frequently implemented in embedded software, where the sector number determines which set of duty-cycle equations is used.[50]

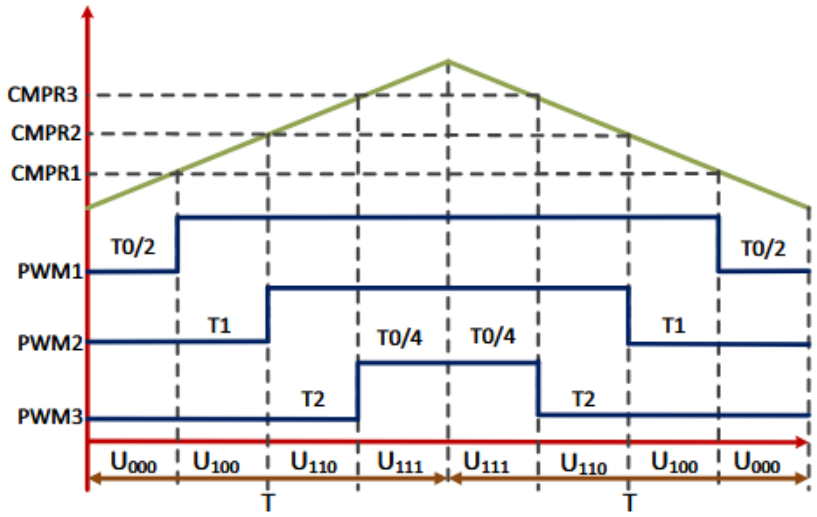


Figure 5-4: pulse pattern for sector 1

❖ **Generation of PWM Signals:**

The duty cycles are converted into timer compare values according to:

$$Compare\ value\ =\ D\ x$$

Where:

- D is the duty cycle
- N is the timer period (Auto-Reload Register, ARR).

For example, if the timer period is:

$$N\ =\ 4000$$

And

$$D_a\ =\ 0.72$$

the corresponding compare value is:

$$CCR_A\ =\ 0.72\ x\ 4000\ =\ 2880$$

The same procedure is repeated for phases B and C.

These compare values define the switching instants of the three inverter legs and are updated at every switching period.

13. Advantages of the Symmetrical Switching Sequence:

The symmetrical distribution of the switching states provides several important advantages:[50]

- It minimizes low-order harmonic components in the output voltage.
- It reduces the Total Harmonic Distortion (THD) of the stator currents.
- It decreases the switching losses by ensuring that only one inverter leg changes state at each transition.
- It reduces common-mode voltage variations.
- It improves the electromagnetic torque quality by minimizing torque ripple.
- It provides balanced thermal stress among the inverter switches.

These characteristics explain why symmetrical SVPWM is the preferred modulation technique for high-performance PMSM drives.

14. Implementation in the Proposed PMSM Drive:

In the proposed sensorless PMSM drive, the Backstepping current controller generates the reference voltages v_d^* and v_q^*. After the inverse Park transformation, the SVPWM algorithm determines the sector, computes the switching times T_1, T_2, and T_0, and finally calculates the

duty cycles D_a, D_b, and D_c. These duty cycles are loaded into the PWM timer registers of the STM32 microcontroller, which generates the complementary gate signals required to drive the three-phase inverter.

❖ **Numerical calculation of the SVPWM Algorithm used for the PMSM drive:**

At the following the complete implementation of the Space Vector Pulse Width Modulation algorithm for the proposed sensorless PMSM drive. The objective is to show how the reference voltages generated by the Backstepping controller are converted into the PWM duty cycles applied to the inverter.[50]

The inverter parameters are summarized in the table below:

Table 5-6: SVPWM sectors.

Parameter	Symbol	Value
DC bus voltage	V_{dc}	48 V
Switching frequency	f_s	10 kHz
Electrical frequency	f_e	50 Hz
PWM period	T_s	? (to calculate)
Reference d-axis voltage	v_d^*	0 V
Reference q-axis voltage	v_q^*	18 V
Electrical angle	θ_e	25°

These values are realistic because with a 48-V inverter, a q-axis voltage around 18 V corresponds to a modulation index of approximately 0.75, which is comfortably inside the linear SVPWM region.

Sector	Angular interval
1	0°–60°
2	60°–120°
3	120°–180°
4	180°–240°
5	240°–300°
6	300°–360°

Step 1: Inverse Park Transformation:

The voltage references are transformed from the rotating dq frame into the stationary αβ\alpha\betaαβ frame using:

$$V_\alpha = v_d^* \cos\theta_e - v_q^* \sin\theta_e \quad (5.76)$$

$$V_\beta = v_d^* \sin\theta_e + v_q^* \cos\theta_e \quad (5.77)$$

Since,

$$v_d^* = 0$$

The equations become:

$$V_\alpha = -18 \sin 25 \quad (5.78)$$

$$V_\beta = 18 \cos 25 \quad (5.79)$$

Using:

$$\sin 25 = 0.4226$$

$$\cos 25 = 0.9063$$

Gives:

$$V_\alpha = -7.61V$$

$$V_\beta = 16.31V$$

Step 2: Magnitude of the Reference Voltage:

The magnitude of the reference vector is:

$$V_{ref} = \sqrt{{V_\alpha}^2 + {V_\beta}^2} \quad (5.80)$$

Substitute the calculated values,

$$V_{ref} = \sqrt{(-7.61)^2 + (16.31)^2}$$

$$V_{ref} = \sqrt{57.9 + 266}$$

$$V_{ref} = \sqrt{323.9}$$

$$V_{ref} = 18V$$

which confirms that the inverse Park transformation preserves the vector magnitude.

Step 3: Determination of the Vector Angle:

The angle of the reference vector is:

$$\theta_s = atan2(V_\beta, V_\alpha) \tag{5.81}$$

Substitute:

$$\theta_s = atan2(16.31, -7.61)$$

Gives:

$$\theta_s = 115^o$$

Step 4: Sector Determination:

The six SVPWM sectors are:

Since,

$$\theta_s = 115^o$$

belongs to:

$$60^o < 115^o < 120^o$$

the reference vector is located in:

Sector 2:

Therefore, the inverter will synthesize the reference voltage using:

V_2 and V_3.

Step 5: Switching Period:

The switching period is:

$$T_s = \frac{1}{f_s} \tag{5.82}$$

Therefore,

$$T_s = \frac{1}{10000}$$

$$T_s = 100\ \mu s$$

Thus, every $\boldsymbol{100\ \mu s}$ the STM32 updates the PWM outputs.

Step 6: Local Angle Inside the Sector:

The formulas for T_1 and T_2 use the angle inside the current sector rather than the absolute angle.

Since the vector lies in Sector 2,

$$\alpha = 115^o - 60^o$$

$$\alpha = 55^o$$

Step 7: Modulation Index:

The modulation index is:

$$m = \frac{\sqrt{3}V_{ref}}{V_{dc}} \tag{5.83}$$

Substituting:

$$m = \frac{\sqrt{3}\ x\ 18}{48} = 0.6495$$

$$m = \ 0.65$$

which confirms that the inverter operates in the linear SVPWM region.

Step 8: Computation of T_1:

The switching time is:

$$T_1 = \ mT_s\ sin\left(\frac{\pi}{3} - \alpha\right) \tag{5.84}$$

Substituting:

$$T_1 = \ 0.65x\ 100\ sin(60 - 55)$$

$$sin5^o = \ 0.0872$$

$$T_1 = 65\ x\ 0.0872$$

$$T_1 = 5.67\ \mu s$$

Step 9: Computation of T_2:

$$T_2 = \ mT_s\ sin(\alpha) \tag{5.85}$$

Substituting:

$$T_2 = \ 0.65\ x\ 100\ sin\ 55^o$$

$$sin\ 55^o = \ 0.8192$$

$$T_2 = 65\ x\ 0.8192$$

$$T_2 = 53.25\ \mu s$$

Step 10: Computation of T_0 :

$$T_0 = T_s - T_1 - T_2 \tag{5.86}$$

Substituting:

$$T_0 = 100 - 5.67 - 53.25$$

$$T_0 = 41.08\ \mu s$$

$$\frac{T_0}{2} = 20.54\ \mu s$$

Step 11: Duty Cycles:

Since the reference vector is in Sector 2,

$$D_a = \frac{T_2 + \frac{T_0}{2}}{T_s} \tag{5.87}$$

$$D_a = \frac{5.67 + 20.54}{100}$$

$$D_a = 0.262$$

$$D_b = \frac{T_1 + T_2 + \frac{T_0}{2}}{T_s} \tag{5.88}$$

$$D_b = \frac{5.67 + 53.25 + 20.54}{100}$$

$$D_b = 0.794$$

$$D_c = \frac{\frac{T_0}{2}}{T_s} \tag{5.89}$$

$$D_c = \frac{20.54}{100}$$

$$D_c = 0.205$$

Thus, the PWM duty cycles are:

Table 5-7: the duty cycle of each phase.

Phase	Duty cycle
A	26.2%
B	79.4%
C	20.5%

Step 12: STM32 Timer Compare Registers:

Assume Timer 1 operates with:

$$ARR = 4000$$

The compare registers become:

$$CCR_A = 0.262 \times 4000 = 1048$$
$$CCR_B = 0.794 \times 4000 = 3176$$
$$CCR_C = 0.205 \times 4000 = 820$$

The STM32 loads these values into the timer registers:

```
TIM1->CCR1 = 1048;
TIM1->CCR2 = 3176;
TIM1->CCR3 = 820;
```

At the next PWM interrupt, these compare values are updated again according to the newly computed v_d^* and v_q^*, and θ_e.

Step 13: Relationship Between 50 Hz and 10 kHz:

The electrical period of the motor is:

$$T_e = \frac{1}{50}$$
$$T_e = 20\ ms$$

Each PWM period lasts $100\mu s$

Therefore,

$$\frac{20}{100} = 200$$

Meaning 200 PWM periods occur during one electrical cycle.

Step 14: Rotation of the Reference Vector:

Since one electrical revolution is: 360∘, the reference vector advances by:

$$\Delta\theta = \frac{360}{200}$$
$$\Delta\theta = 1.8^o$$

at every PWM update.
Consequently,

Quantity	Value
Electrical period	20 ms
PWM period	100 µs
PWM updates per electrical cycle	200
Angle increment per update	1.8°

Step 15: Transition Between the Six Sectors:

Each sector spans 60º

Therefore,

$$\frac{60^o}{1.8^o} = 33.3$$

Hence, the controller remains in each sector for approximately 33 PWM period before moving to the next one.

The sequence of sectors during one electrical revolution is:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 1$$

During every PWM interrupt, the STM32 repeats the following operations:

- Read the estimated electrical angle from the observer.
- Compute v_d^* and v_q^* using the Backstepping controller.
- Perform the inverse Park transformation to obtain V_α and V_β .
- Determine the current SVPWM sector.
- Compute T_1, T_2, and T_0.
- Calculate the duty cycles D_a, D_b, and D_c.
- Update the timer compare registers (CCR_A, CCR_B, and CCR_C).
- Generate new complementary PWM signals for the three inverter legs.

Step 16: Simulation of SVPWM Applied onto The Three-Phase-Inverter That Excited the PMSM:

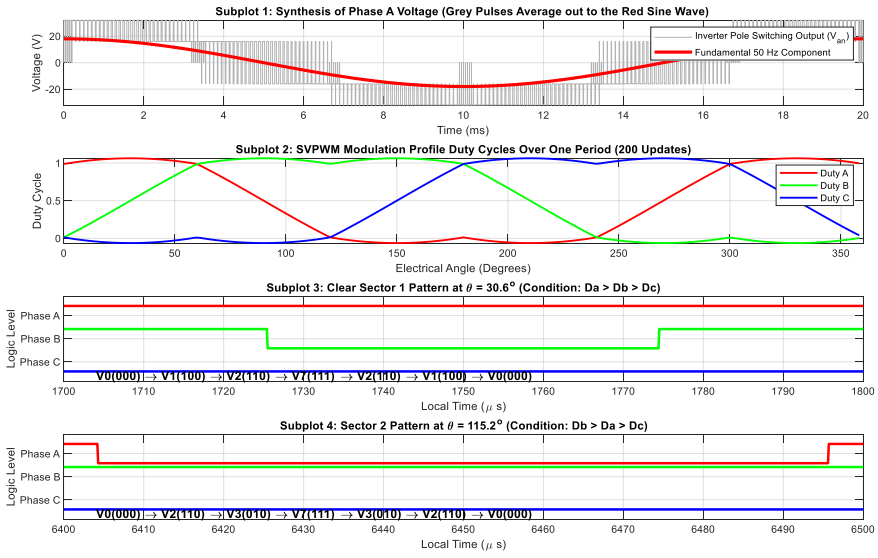


Figure 5-5: simulation of SVPWM generated by STM32 applied onto the inverter

the whole system diagram in MATLAB/SUMILINK is represented in the figure 6:

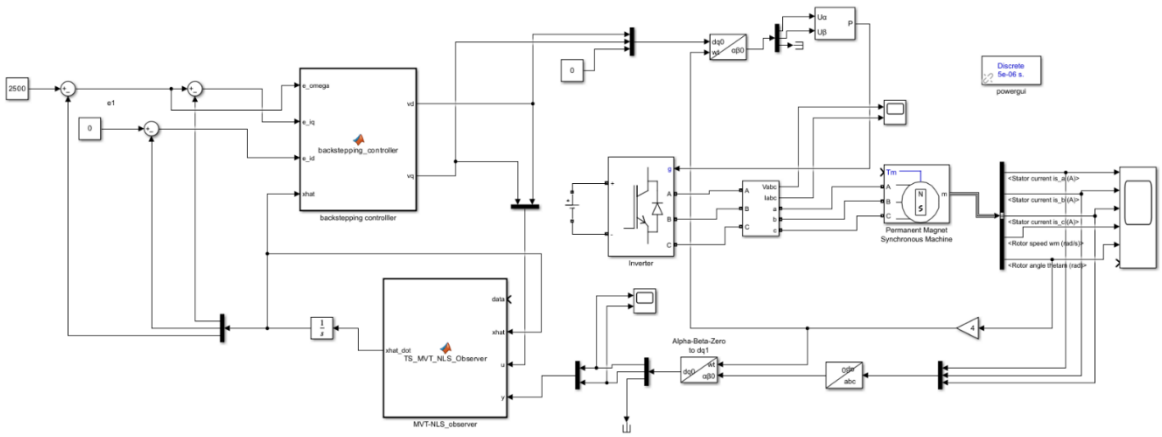


Figure 5-6: shows the control system diagram of PMSM using Backstepping Controller and MVT-NSL based Observer

The simulation result of loaded motor represented in the figure 5-16 below:

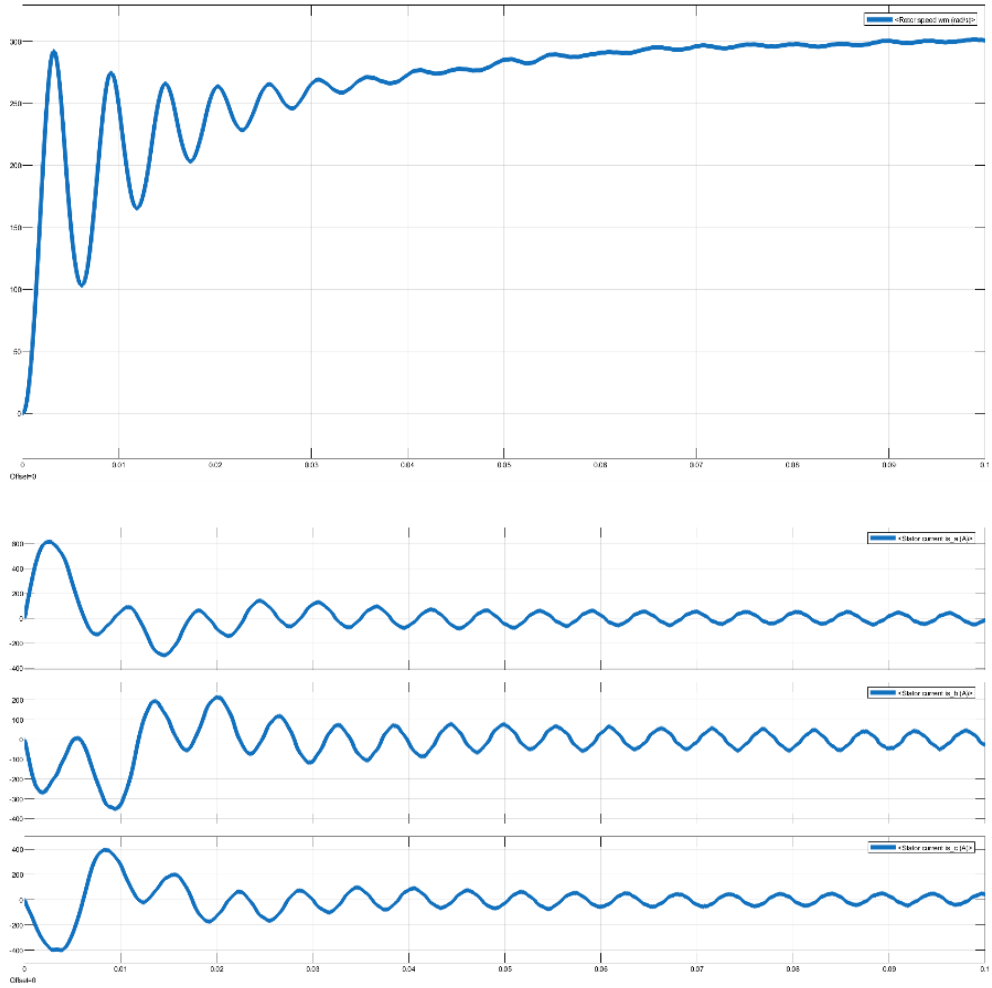


Figure 5-7: Simulation result showing the Three-Phase Currents, the Rotor Speed

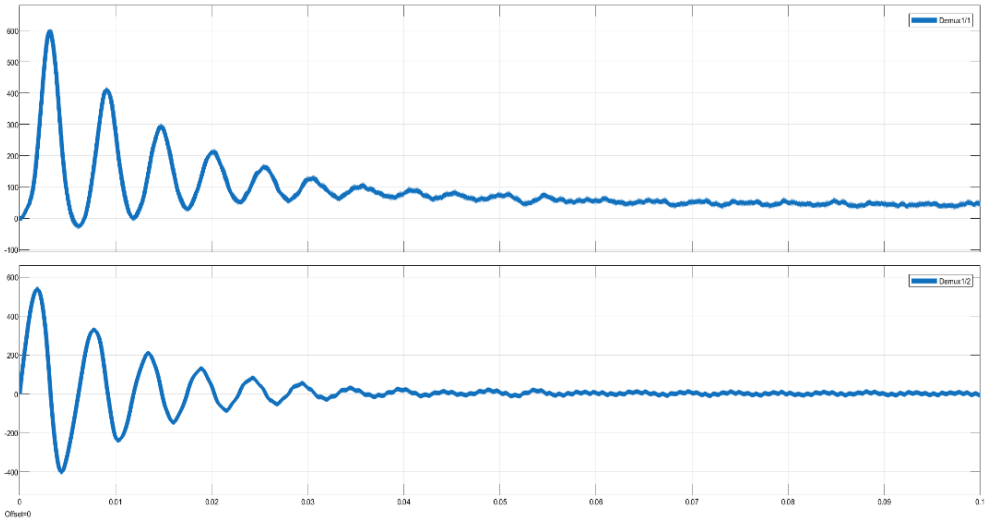


Figure 5-8: shows the quadratic component current i_q, and the direct component current i_d

As: $t \rightarrow \infty$

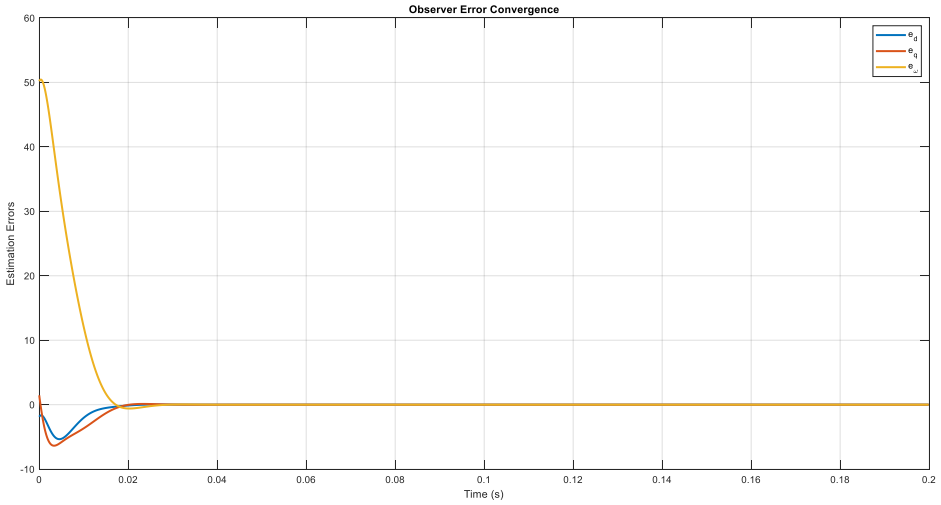


Figure 5-9: shows the error observer convergence

Conclusion

This chapter presented the design and implementation of a **Backstepping control strategy** for the PMSM, with the objective of regulating the rotor speed and controlling the d- and q-axis currents through the generation of the reference voltages v_d and v_q. The resulting control voltages were then transformed into three-phase reference voltages and applied to an **SVPWM modulation scheme** to generate the appropriate inverter switching signals. The simulation results demonstrated the ability of the proposed control structure to regulate the PMSM variables, while the SVPWM technique provided the required three-phase switching pattern with appropriate voltage utilization and reduced harmonic content.

Chapter 6: Practical Design:

1. Parameter's Identification:

In this work, the identification of the PMSM parameters aims to obtain the practical electrical and mechanical characteristics of the motor and to use the identified values in the **experimental implementation of the proposed control strategy**. Accurate knowledge of these parameters is essential for obtaining a reliable PMSM model and ensuring that the Backstepping controller and observer are properly tuned for practical operation. Therefore, the main parameters required for the implementation are experimentally determined and used in the subsequent stages of the work.

In particular, the stator resistance $\boldsymbol{R_s}$, the direct- and quadrature-axis inductances $\boldsymbol{L_d}$ and $\boldsymbol{L_q}$, the permanent-magnet flux linkage $\boldsymbol{\Psi_{pm}}$, the rotor inertia $\boldsymbol{J}$, and the viscous friction coefficient $\boldsymbol{B}$ directly affect the electrical and mechanical dynamics of the PMSM. (Back-EMF), and dynamic behavior.

This chapter presents the experimental procedures used to identify the principal electrical and mechanical parameters of the PMSM. The identification is performed through a series of controlled tests in which the required electrical and mechanical quantities are measured and subsequently used to calculate the corresponding parameters.

The electrical parameters are determined first. The stator resistance $\boldsymbol{R_s}$ is obtained from a DC resistance test, while the $\boldsymbol{d}$- and $\boldsymbol{q}$-**axis** inductances are determined through inductance measurements performed under appropriate rotor-position conditions. The permanent-magnet flux linkage $\boldsymbol{\Psi_{pm}}$ is then identified from the back electromotive force (**back-EMF**) generated during rotor rotation.

The mechanical parameters are identified separately. The moment of inertia $\boldsymbol{J}$ and viscous friction coefficient $\boldsymbol{B}$ are obtained from the motor's deceleration behavior under no-load conditions. Finally, the identified parameters are implemented in the PMSM mathematical model and compared with the experimental measurements in order to validate the obtained parameter set.

The experimental results obtained in this chapter will subsequently be used in the simulation and experimental validation of the proposed sensorless control system.

1.1. Laboratory Test Bench

❖ **General Description:**

The experimental setup consists of the PMSM under investigation, electrical measuring instruments, a DC or AC excitation source, a mechanical locking system, and, for the back-EMF experiment, a variable-speed prime mover.

The laboratory setup should be configured according to the specific test being performed. Since the resistance, inductance, and flux linkage are determined using different experimental procedures, the motor is connected differently for each test.

The experimental setup consists essentially of the following elements:

- PMSM under test
- Three-phase voltage-source inverter
- DC power supply
- Voltage measurement system
- Three-phase current measurement system
- Rotor-speed measurement system
- Data-acquisition/control system
- Computer for data recording and processing

1.1.1. Define the Motor Resistance$R_s(\Omega)$:

The stator resistance is one of the fundamental electrical parameters of the PMSM. It determines the copper voltage drop and the resistive power losses in the stator windings. It also has a direct influence on the accuracy of the PMSM electrical model, particularly at low and medium speeds.

The resistance can be determined experimentally using a DC test. Since the rotor is stationary during this test, no back-EMF is generated. Consequently, the voltage applied to a stator winding is related to the winding current primarily through its resistance.

For a single stator phase, the resistance is determined from Ohm's law:

$$R_s = \frac{V_{dc}}{I_{dc}} = \frac{\mathbf{1.2}}{\mathbf{1.5}} = \mathbf{0.8}\,\Omega \tag{6.1}$$

where V_{dc} is the measured DC voltage applied to the winding and I_{dc} is the corresponding measured current.

If the measurement is performed between two motor terminals, the measured resistance corresponds to the resistance of two stator windings connected in series. In that case,

$$R_{LL} = 2R_s \tag{6.2}$$

and therefore,

$$R_s = \frac{R_{LL}}{\mathbf{2}} = \frac{\mathbf{0.8}}{\mathbf{2}} = \mathbf{0.4}\,\Omega$$

for a symmetrical star-connected PMSM.

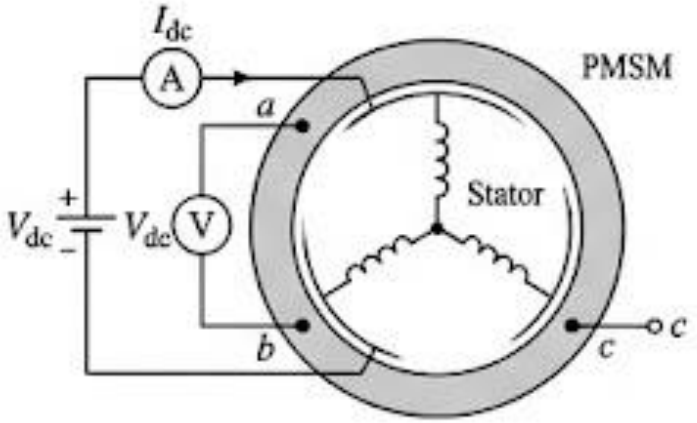


Figure 6-1 : DC resistance measurement

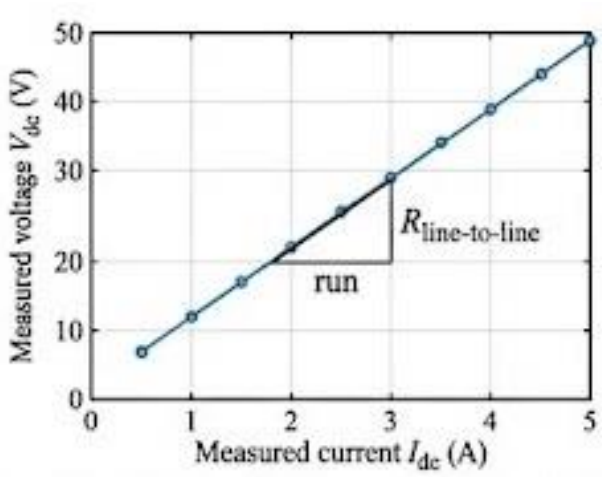


Figure 6-2: measurement voltage-current characteristic

1.1.2. Experimental Identification of the Inductances L_d and $L_q(H)$:

The ***d*-** and ***q*-axis** inductances are important parameters in the PMSM electrical model because they determine the relationship between the stator currents and flux linkages in the rotating reference frame.

For a SPMSM, the voltage equations in the ***dq*** reference frame where $L_d = L_q = L_s$ are expressed as:

$$v_d = R_s i_d + L_s \frac{di_d}{dt} - \omega_e L_s i_q \tag{6.3}$$

and

$$v_q = R_s i_q + L_s \frac{di_q}{dt} + \omega_e \left(L_s i_d + \psi_{pm}\right) \tag{6.4}$$

where $\boldsymbol{L_d}$ and $\boldsymbol{L_q}$ represent the inductances along the direct and quadrature axes, respectively.

The experimental identification therefore requires the rotor to be positioned appropriately so that the applied test signal can be associated with either the d-axis or the $q - axis$.

❖ **Identification of $\boldsymbol{L_s}$:**

The rotor is first positioned such that the selected stator excitation is aligned with the magnetic $\boldsymbol{d}$ −axis. A low-frequency or DC test signal is then applied to the corresponding winding configuration, and the resulting current response is measured.

For a voltage step applied to an equivalent $R - L$ circuit, the current response can be expressed as:

$$i(t) = \frac{V}{R_s}\left(1 - \boldsymbol{e}^{-t/\tau}\right) \tag{6.5}$$

Where:

$$\tau = \frac{R_s}{L_s} \tag{6.6}$$

is the electrical time constant.

Therefore, once the time constant has been obtained from the measured current response, the inductance can be calculated from

$$L_s = R_s \tau = \ 0.4 \ x \ 0.0125 = \ 0.005 \ H$$

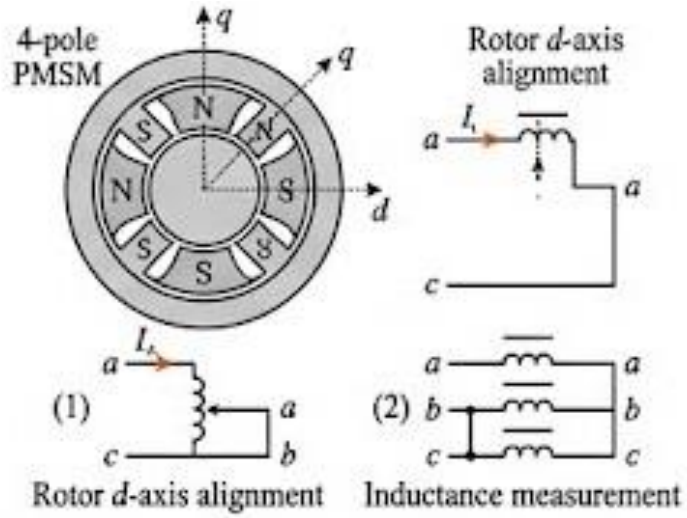


Figure 6-*3*: Experimental d-axis test

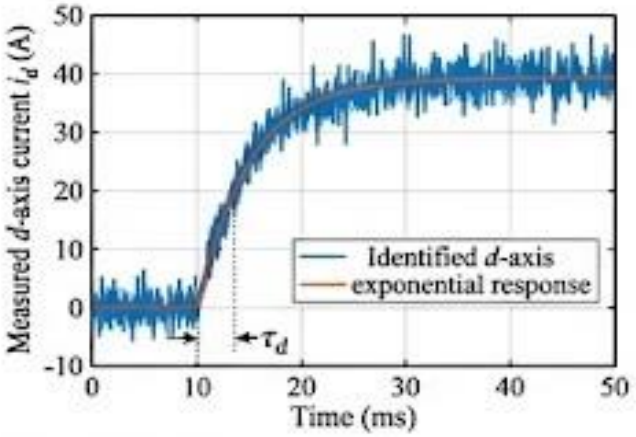


Figure 6-*4*: measured d-axis current response

1.1.3. Experimental Identification of the Permanent-Magnet Flux Linkage $\psi_{pm}(Wb)$:

The permanent-magnet flux linkage is another fundamental parameter of the PMSM. It determines the magnitude of the back-EMF generated by the machine and strongly influences the electromagnetic torque.

The parameter can be experimentally identified using a no-load back-EMF test. In this experiment, the rotor is mechanically driven while the stator windings remain electrically open. As the rotor rotates, the permanent magnets produce a rotating magnetic field that induces a voltage in the stator windings.

The generated phase back-EMF can be expressed as:

$$e_a = -\omega_e \psi_{pm} sin(\theta_e) \tag{6.7}$$

with the corresponding phase amplitude related to the electrical angular velocity by

$$E_{phase} = \omega_e \psi_{pm} \tag{6.8}$$

when E_{phase} represents the peak phase back-EMF.

Therefore,

$$\psi_{pm} = \frac{E_{phase}}{\omega_e} \tag{6.9}$$

The electrical angular speed is related to the mechanical speed ω_m which is in this test $\omega_m = 1500\ rpm$ by:

$$\omega_e = p\omega_m = 4 * \frac{1500 * 2\pi}{60} = 628.32\ rad/s \tag{6.10}$$

where p is the number of pole pairs.

Thus, at no-load test around 1500 rpm we measured a phase back-EMF of approximately **62.8 V peak**, or so, from the equation:

$$\psi_{pm} = \frac{62.83\ V}{628.32} = 0.1 Wb \tag{6.11}$$

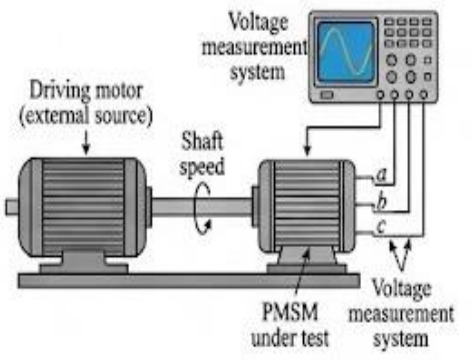


Figure 6-5: experimental setup Measured PMSM back-EMF waveform

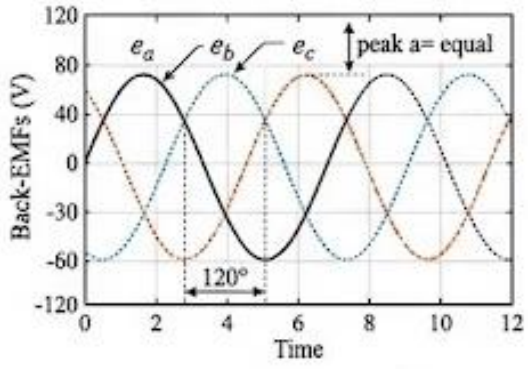


Figure 6-6: No load back-EMF

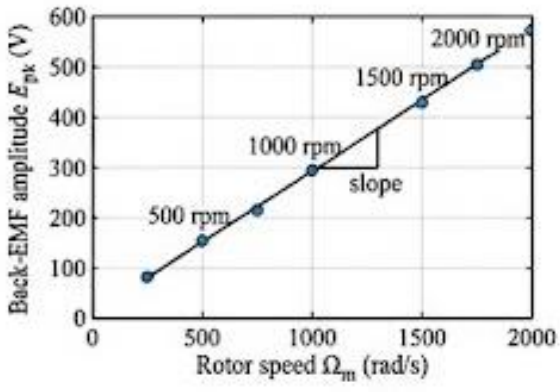


Figure 6-7: back-EMF amplitude versus rotor speed

1.2.4. Experimental Determination of the Mechanical Parameters:

The mechanical behavior of the PMSM is described by

$$J\frac{d\omega_m}{dt} = T_e - T_L - B\omega_m \tag{6.12}$$

where J is the rotor and load inertia and B is the viscous friction coefficient.

For the parameter identification, the motor is operated under no-load conditions. The load torque is therefore minimized, and the motor's deceleration behavior can be used to identify the mechanical parameters.

❖ **Identification of the Moment of Inertia $J(kg \cdot m^2)$:**

The moment of inertia can be estimated using a coast-down test. The motor is first accelerated to a known speed and then electrically disconnected or brought to a condition in which the electromagnetic torque is approximately zero.

During the deceleration phase, the mechanical equation becomes approximately:

$$J\frac{d\omega_m}{dt} = -B\omega_m - T_f \tag{6.13}$$

where T_f represents the mechanical friction torque.

If the friction model is approximated as viscous friction only, then

$$J\frac{d\omega_m}{dt} = -B\omega_m \tag{6.14}$$

Which gives

$$\omega_m(t) = \omega_{m0}\boldsymbol{e}^{\frac{-B}{J}t} \tag{6.15}$$

The measured rotor-speed decay can therefore be used to determine the mechanical time constant.

$$\tau_m = \frac{B}{J} \tag{6.16}$$

However, J and B cannot generally be determined independently from one simple exponential decay test alone. An additional relationship or experiment is required to separate them.

This is important for the thesis: we should not claim that a single coast-down curve independently identifies both J and B.

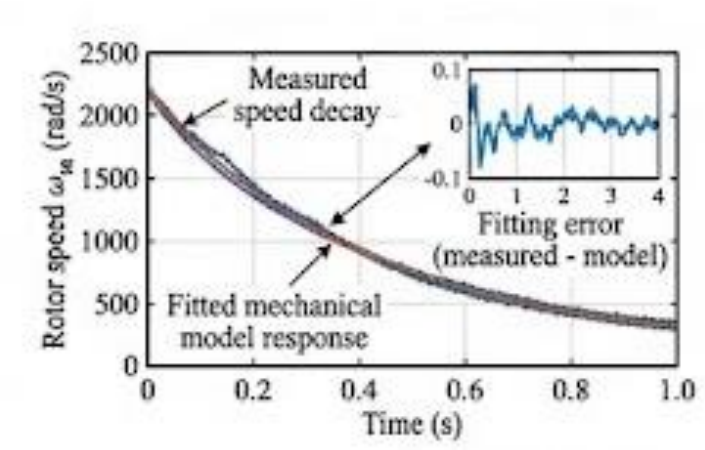


Figure 6-*8*: Measured rotor-speed decay

❖ **Identification of the Viscous Friction Coefficient $B(N \cdot m \cdot s/rad)$:**

The viscous friction coefficient can be determined from steady-state no-load operation.

At steady state,

$$\frac{d\omega_m}{dt} = 0 \tag{6.17}$$

and therefore

$$T_e = T_L + B\omega_m \tag{6.18}$$

Under no-load conditions,

$$T_L \approx 0$$

So that

$$T_e \approx B\omega_m \tag{6.19}$$

If the electromagnetic torque is known or can be calculated from the measured electrical quantities, the viscous friction coefficient can be estimated from:

$$B = \frac{T_e}{\omega_m} \tag{6.20}$$

over a suitable range of speeds.

A plot of electromagnetic torque against mechanical angular speed can therefore be used to identify the viscous friction coefficient from the slope of the approximately linear region.

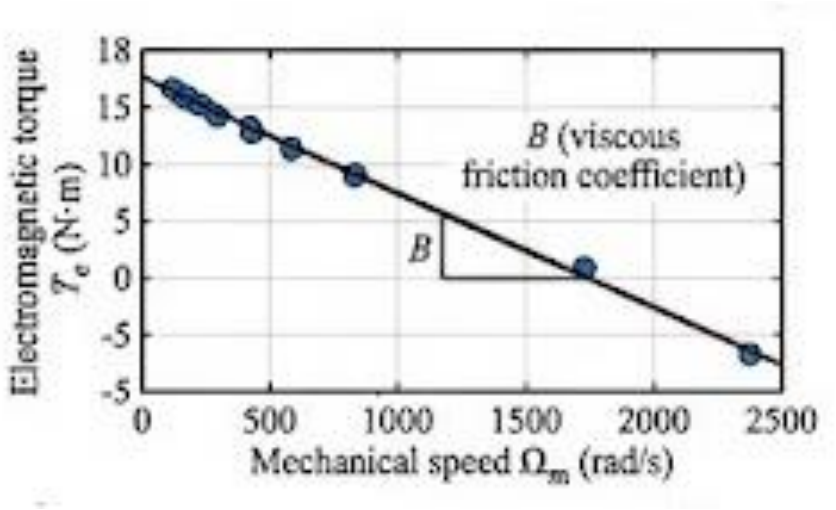


Figure 6-9:electromagnetic torque versus rotor speed, the slop of the fitted characteristic provides an experimental estimat3 of friction B

From the graph shown in the figure above we can extract the value of B at mechanical speed

$$\omega_m = 1500 \;\Rightarrow\; T_e = 2.5\, N.m$$

Therefore

$$B = \frac{2.5}{1500} = 0.001 N \cdot m \cdot s/rad$$

As τ_m= 10s,

$$J = \; 10 * 0.001 = 0.01 kg \cdot m^2$$

This chapter presented the experimental procedures used to identify the principal electrical and mechanical parameters of the PMSM. The stator resistance R_s, the direct- and quadrature-axis inductances L_d and L_q, and the permanent-magnet flux linkage ψ_{pm} were identified through dedicated electrical tests. The mechanical parameters J and B were determined from the dynamic and steady-state behavior of the motor under no-load conditions.

The identified parameters provide a more representative description of the actual PMSM than nominal values obtained solely from manufacturer data. They are therefore used in the subsequent modeling and simulation stages of this work.

Finally, the identified model is validated by comparing its simulated responses with the experimentally measured motor behavior. This validation establishes the basis for using the identified PMSM model in the development and evaluation of the proposed control and sensorless observation strategies.

1.2. The power source (three-phase inverter):

The Voltage Source Inverter (VSI) is the power conversion stage that interfaces the DC power supply with the Permanent Magnet Synchronous Motor (PMSM). Its primary function is to convert the constant DC voltage into a balanced three-phase AC voltage with controllable amplitude and frequency. In modern electric drive systems, the inverter acts as the link between the control algorithm and the electric machine, allowing precise regulation of the motor torque and speed. The design of the inverter for BLDC motor driver in Proteus software is shown in Fig. 2. There are MOSFET and its driver, voltage, and current sensor, and two Arduino as controller. The supply voltage enters the circuit and passes through the ACS712 current sensor. The results of this sensor reading will be forwarded to the Arduino UNO to be processed as overcurrent protection feedback. The supply voltage is then used to supply the driver and the microcontroller as well as the supply for the MOSFET.

Arduino Nano is used to drive the IR2104 IC with a 3-phase PWM configuration that is on/off according to the BLDC motor configuration.

The IR2104 was chosen because it has several advantages compared to another MOSFET driver ICs.The design of the inverter for BLDC motor driver in Proteus software is shown in Fig. 2. There are MOSFET and its driver, voltage, and current sensor, and two Arduino as controller. The supply voltage enters the circuit and passes through the ACS712 current sensor. The results of this sensor reading will be forwarded to the Arduino UNO to be processed as overcurrent protection feedback. The supply voltage is then used to supply the driver and the microcontroller as well as the supply for the MOSFET.

Arduino Nano is used to drive the IR2104 IC with a 3-phase PWM configuration that is on/off according to the BLDC motor configuration.

The IR2104 was chosen because it has several advantages compared to another MOSFET driver ICs. This The design of the inverter for BLDC motor driver in Proteus software is shown in Fig.2. There are MOSFET and its driver, voltage, and current sensor, and two Arduino as controller. The supply voltage enters the circuit and passes through the ACS712 current sensor. The results of this sensor reading will be forwarded to the Arduino UNO to be processed as overcurrent protection feedback. The supply voltage is then used to supply the driver and the microcontroller as well as the supply for the MOSFET.

Arduino Nano is used to drive the IR2104 IC with a 3-phase PWM configuration that is on/off according to the BLDC motor configuration.

The design of the inverter for BLDC motor driver in Proteus software is shown in Fig. 2. There are MOSFET and its driver, voltage, and current sensor, and two Arduino as controller.

The supply voltage enters the circuit and passes through the ACS712 current sensor. The results of this sensor reading will be forwarded to the Arduino UNO to be processed as overcurrent protection feedback. The supply voltage is then used to supply the driver and the microcontroller as well as the supply for the MOSFET.

Arduino Nano is used to drive the IR2104 IC with a 3-phase PWM configuration that is on/off according to the BLDC motor configuration.

The design of the inverter for BLDC motor driver in Proteus software is shown in Fig. 2.

The Voltage Source Inverter (VSI) is the power conversion stage that interfaces the DC power supply with the Permanent Magnet Synchronous Motor (PMSM). Its primary function is to convert the constant DC voltage into a balanced three-phase AC voltage with controllable amplitude and frequency.[51] In modern electric drive systems, the inverter acts as the link between the control algorithm and the electric machine, allowing precise regulation of the motor torque and speed.

In the proposed PMSM drive, the Backstepping controller generates the reference voltage components v_d^* and v_q^* in the rotating (dq) reference frame. These voltage references are transformed into the stationary (\alpha\beta) frame using the inverse Park transformation and subsequently into the three-phase abc frame through the inverse Clarke transformation. The resulting three-phase voltage references are processed by the Space Vector Pulse Width Modulation (SVPWM) algorithm, which determines the switching sequence and duty cycles required to control the inverter power switches.

The inverter employed in this work is a two-level three-phase Voltage Source Inverter operating from a 300V DC supply. It consists of six (06) power **MOSFETs** arranged in three bridge legs and driven by three (03) **IR2110** high-side/low-side gate driver circuits. The IR2110 drivers provide the necessary gate voltages for both the upper and lower MOSFETs while ensuring safe operation through a bootstrap circuit and complementary switching with dead time.[51]

MOSFETs stand for Metal Oxide Semiconductor Field Effect Transistors. These transistors are voltage-controlled devices, which means their conductivity is determined by the voltage applied at the gate terminal, known as Vgs (gate to source voltage). The MOSFET transistors come as a three-terminal device known as Gate, Source, and Drain.

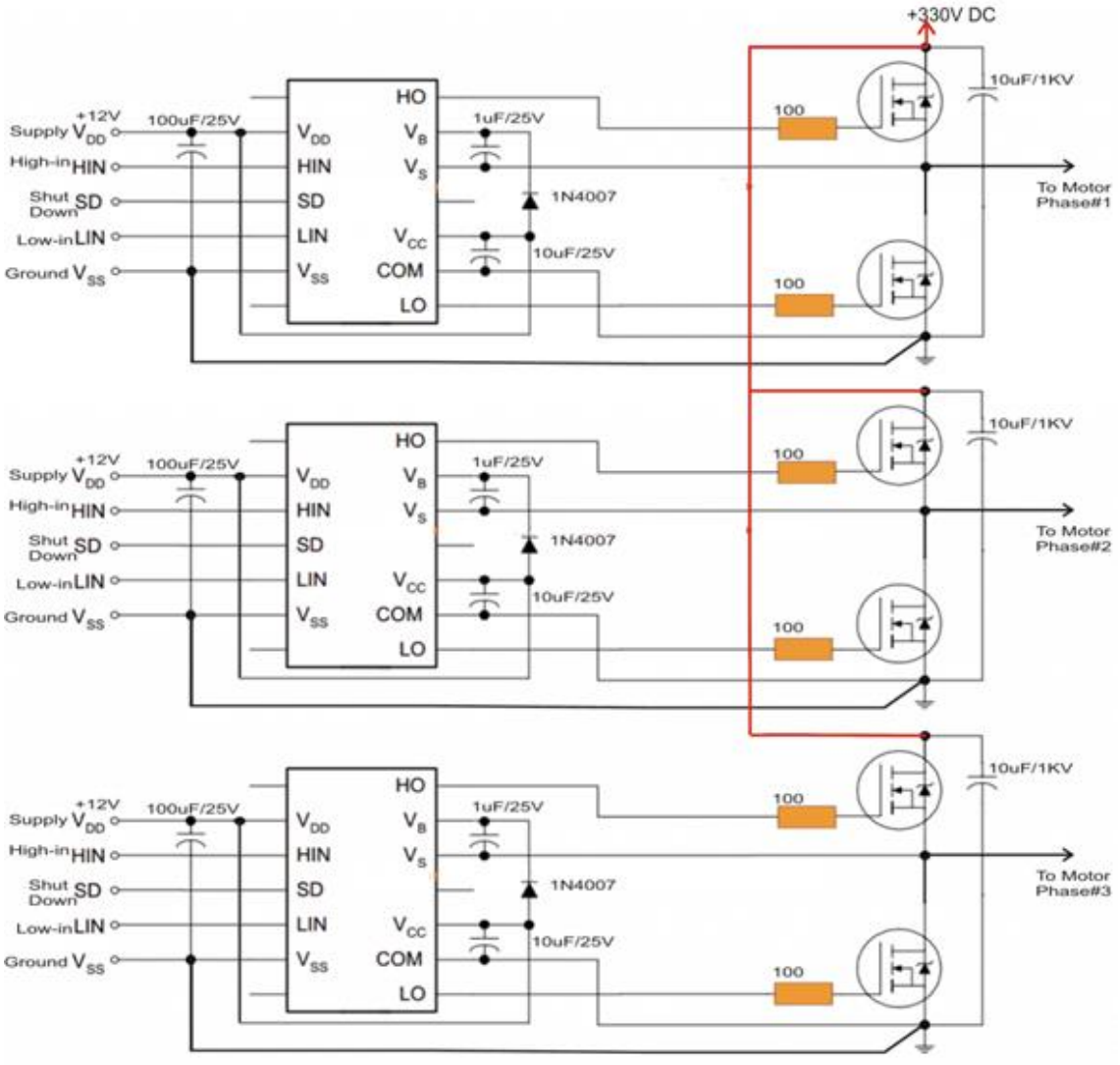


Figure 6-10: three-phase inverter diagram with IR 2110 MOSFET driers

In many electronics applications, such as three-phase inverters, there is a need to simultaneously drive low-side and high-side MOSFET transistors. In such situations, MOSFET drivers are used, and one such popular driver is the IR2110 MOSFET Driver. The IR2110 MOSFET drivers are specializing circuits because they are capable of controlling the gate voltage of both low-side and high-side transistors simultaneously and making the MOSFETs switch ON and OFF quickly. Due to the IR2110 driver's capability of controlling high voltage and high power MOSFETs, IR2110 MOSFET drivers are widely use in various applications, PMSM motor drive circuits, switch-mode power supplies, and inverter circuits.[52]

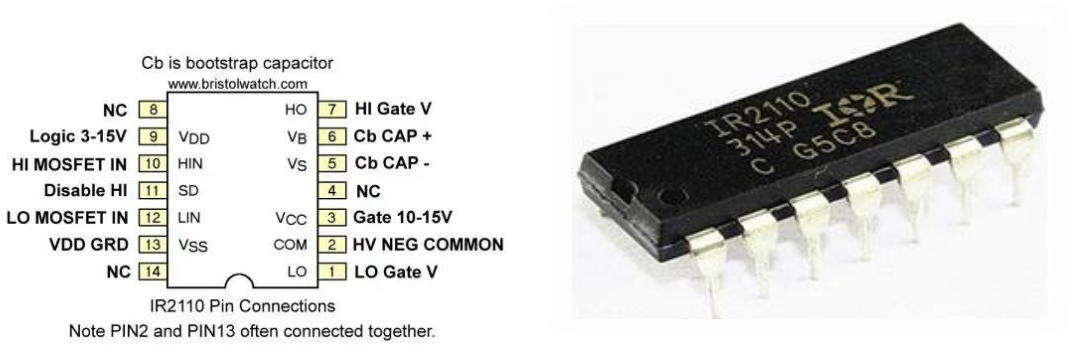


Figure 6-11: the IR2110 Integrated circuit driver

1.2.1. IR2110 as MOSFET Driver:

The IR211O MOSFET driver ICs are high-voltage, high-power driver circuits that are widely use in three-phase inverter to drive both high and low side switches. These drivers consist of independent low-side and high-side reference output channels. The operating voltage range of IR2110 is between 10-20 volts with an output current of 2.0 Amps. Generally, the IR2110 driver comes in 16-PIN SOIC and 14-PIN PDIP packages as it's shown in **Figure 12**. It also consists of a specialize channel design to handle high-side transistor output using a bootstrap operation.[52]

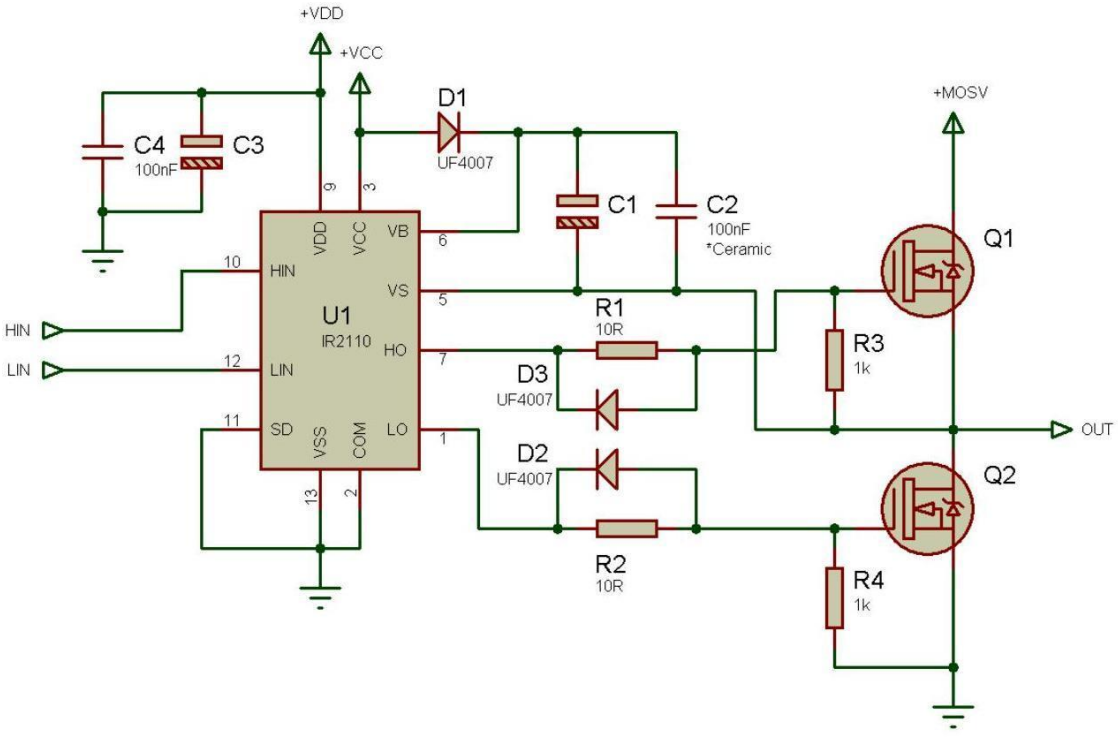


Figure 6-12: show IR2110 ICs connection for driving high and low MOSFET

❖ **Bootstrap Driver**

The gate drive circuit is a fundamental component of the three-phase inverter because it provides the necessary voltage and current required to switch the power semiconductor devices efficiently. In this work, the **IR2110** high- and low-side gate driver is employed due to its high switching capability, wide operating voltage range, and ease of interfacing with digital controllers.

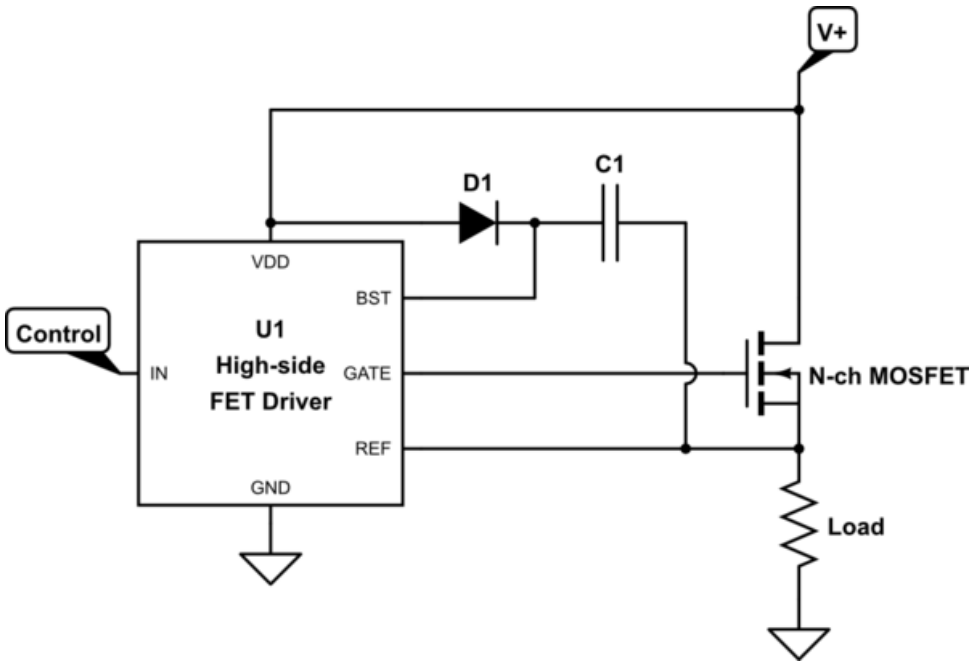


Figure 6-13: shows how to bootstrap of IR2110 IC MOSFET's driver.

The IR2110 integrates two independent gate drivers within a single integrated circuit:

- A **low-side driver** referenced to the system ground.
- A **floating high-side driver** referenced to the switching node.

The high-side driver presents a particular challenge because its source terminal is not fixed at ground potential but instead follows the switching node voltage. Consequently, an isolated floating supply is required to maintain an adequate gate-to-source voltage during switching.

To overcome this limitation, the IR2110 employs a **bootstrap circuit** consisting primarily of a bootstrap capacitor and a fast recovery diode.

When the low-side switch is ON, the switching node is pulled close to ground, allowing the bootstrap capacitor to charge through the bootstrap diode from the low-voltage supply. Once the high-side switch is commanded ON, the stored charge within the capacitor provides the floating supply required to raise the high-side gate voltage above the source potential.

The bootstrap capacitor voltage is approximately:

$$V_{BS} \approx V_{CC} - V_D \tag{6.21}$$

Where:

V_{CC}: is the driver supply voltage and.

V_D: is the forward voltage drop of the bootstrap diode.

The minimum bootstrap capacitance is commonly selected using:

$$C_{BS} \geq \frac{Q_G}{\Delta V_{BS}} \tag{6.22}$$

Q_G: is the total gate charge of the power switch and

ΔV_{BS}: is the maximum allowable voltage ripple across the bootstrap capacitor.

Here is how to calculate the bootstrap capacitor in web-based bootstrap capacitor calculator:

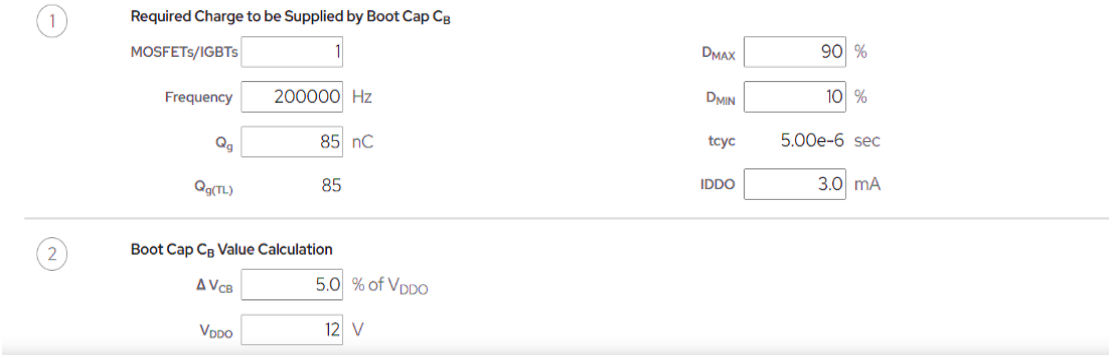


Figure 6-14: shows how to use the web-based bootstrap calculator to determine the required C_B recharge current at MAX and MIN duty cycle, and provides the information required to determine the best fit bootstrap module

The result of calculation of the C_B using the given parameter above:

Q_G = 9.85e-8

C_B = 164 nF

I_{MAX} = 197 mA

I_{AVG} = 20 mA

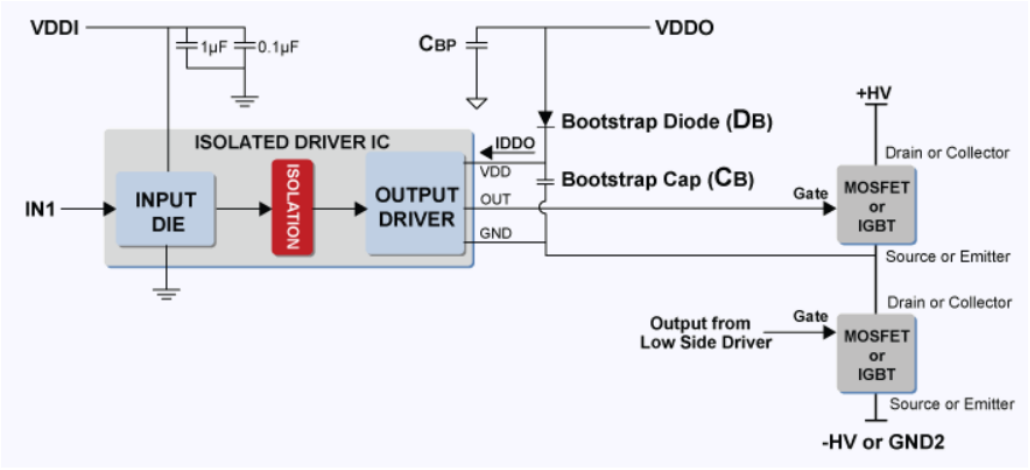


Figure 6-15: shows the bootstrap of the MOSFET driver using the capacitor and diode

❖ **Snubber Circuit**

Fast switching transitions inevitably produce voltage overshoots due to the parasitic inductances associated with the circuit layout, device packaging, and resonant connections. These voltage spikes may exceed the maximum blocking voltage of the semiconductor devices, leading to excessive switching losses or permanent device failure.

To suppress these undesirable transients, snubber circuits are incorporated across the switching devices.

The most common snubber configurations include:

- RC snubber
- RCD snubber
- TVS diode protection

The RC snubber dissipates the stored energy through a resistor while limiting the rate of voltage rise

$$\frac{dv}{dt}$$

Across the switching device. The RCD snubber provides improved energy recovery and is particularly suitable for higher-power converters.

Transient Voltage Suppressor (TVS) diodes offer additional protection against abnormal overvoltage conditions by clamping the voltage to a predetermined safe level.

The choice of snubber depends upon the operating frequency, switching energy, and converter power level.

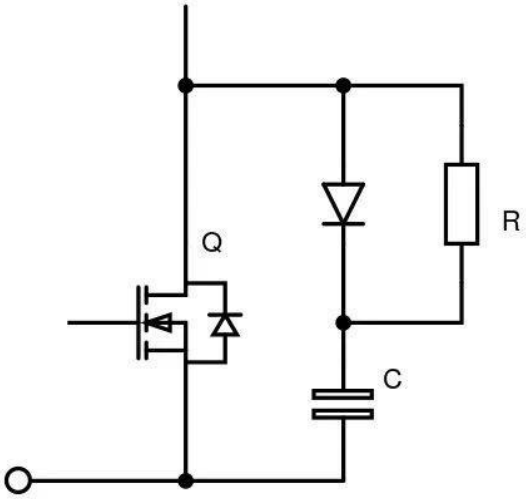


Figure 6-16: shows the RC snubber circuit for the MOSFET

❖ **Dead-Time Insertion**

To prevent simultaneous conduction of the two switches within the same inverter leg, a short non-overlapping interval, commonly referred to as **dead time**, is introduced between the switching commands. During this interval, both switches remain OFF, thereby eliminating shoot-through faults that could short-circuit the DC supply.

The dead time must be carefully selected. An excessively short dead time increases the risk of cross-conduction, whereas an excessively long dead time introduces waveform distortion and increases harmonic content. In practical induction heating systems, dead times typically range from several hundred nanoseconds to a few microseconds depending on the switching device characteristics.

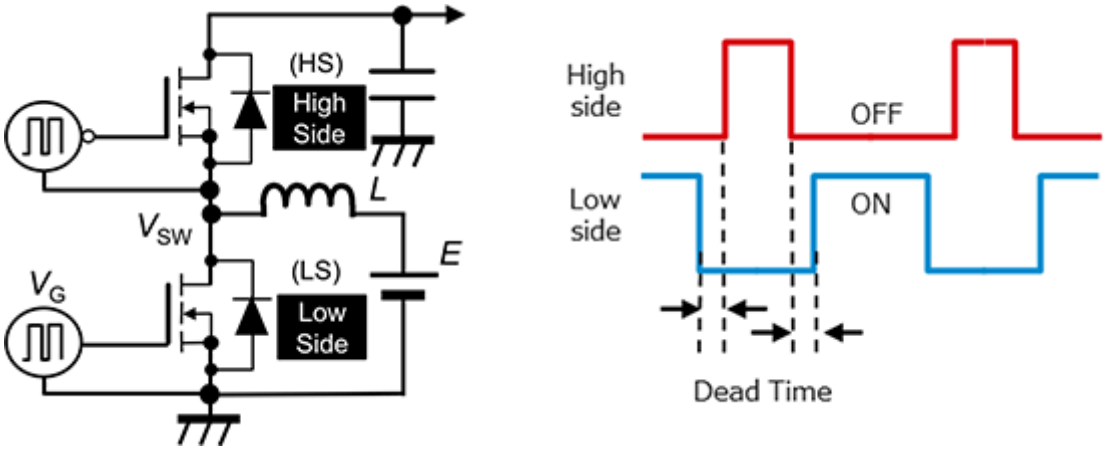


Figure 6-17: shows how to create a dead time within the switching interval of the same leg inverter

1.2.2. The output voltages and currents waveforms of three phase inverter:

❖ **Simulation:**

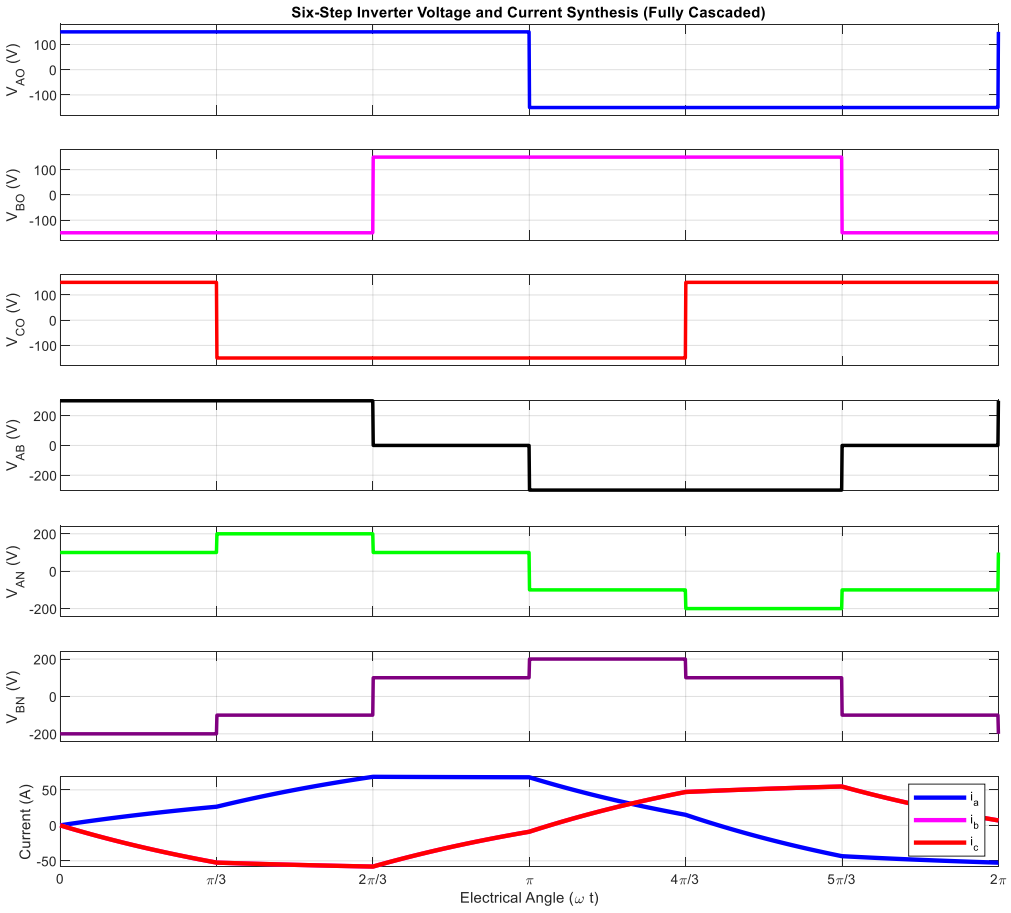


Figure 6-18: the output voltages and currents of three-phase inverter fed the PMSM

2. Inverter Implementation and STM32 Configuration (Programming):

2.1. Realized inverter board:

The inverter was physically assembled on a PCB containing the power switches, gate-driver circuits, passive components, connectors, and the required measurement and protection interfaces.

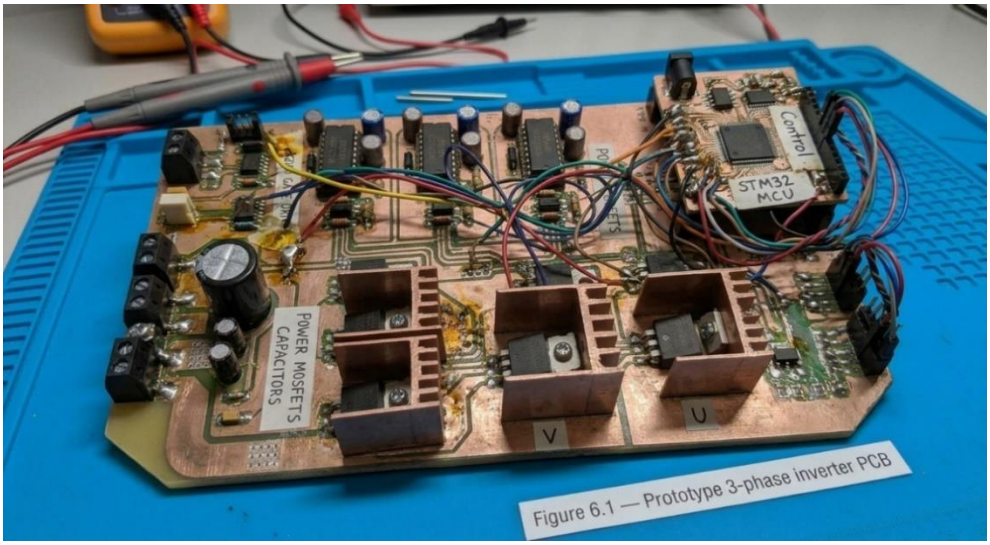


Figure 6-18: three- phase inverter PCB using for controlling PMSM motor

2.2. STM32 microcontroller:

The STM32 is a family of 32-bit microcontrollers developed by STMicroelectronics and based on ARM Cortex processor cores. It integrates the processing unit, memory, timers, analog-to-digital converters, communication peripherals, and general-purpose input/output interfaces required for real-time embedded control.

For the proposed PMSM drive, the STM32 performs the complete digital control sequence. It acquires the motor currents and DC-link voltage through its ADC channels, executes the MVT–SNL observer and Backstepping controller, calculates the reference voltage vector, and finally generates the three-phase SVPWM signals used to control the inverter.

The general control structure implemented in the microcontroller can be represented as ADC measurements→Observer→Backstepping controller→SVPWM→PWM

timers. The use of a single microcontroller for these operations allows the different stages of the control algorithm to be executed within a fixed sampling period.

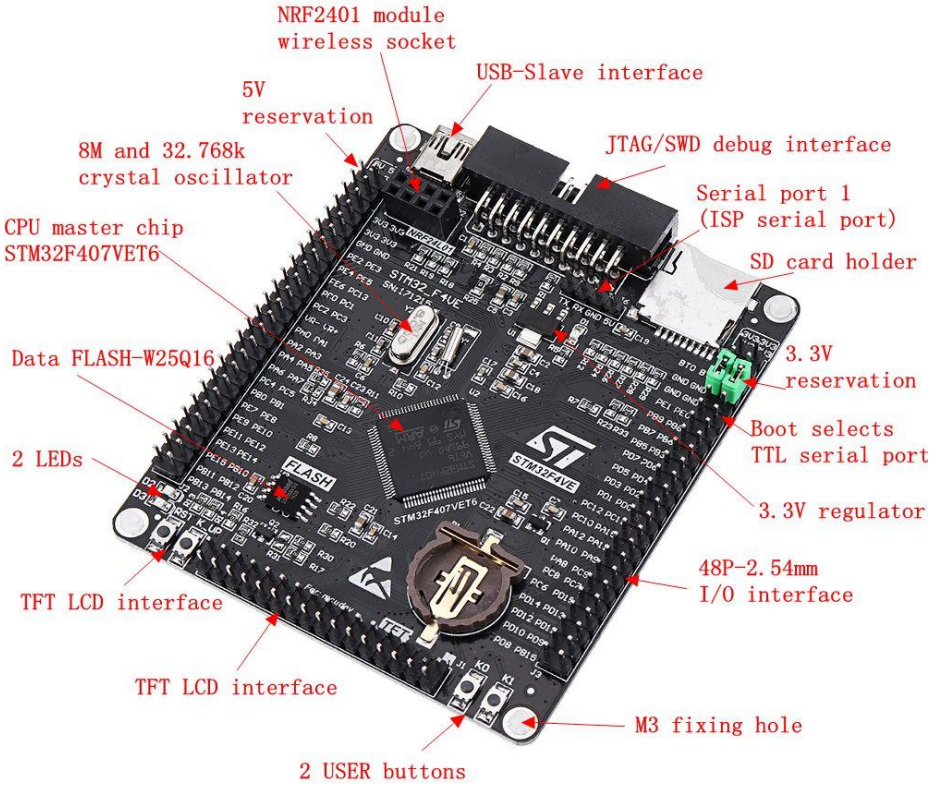


Figure 6-20: STM32 F407 microcontroller

2.3. STM32 timer peripherals:

Timers are particularly important in the implementation of a motor-control system because they are responsible for generating the PWM signals with precise timing.

A timer essentially consists of a counter that increments or decrements according to its operating mode. The counter is compared with one or more capture/compare registers to determine the switching instants of the PWM outputs.

For a basic edge-aligned PWM configuration, the PWM frequency can be expressed as ARR

$$f_{PWM} = \frac{f_{TIM}}{(PSC + 1)(ARR + 1)} \tag{6.22}$$

- f_{PWM}: is the timer input clock frequency;
- PSC: is the prescaler value;
- ARR: is the auto-reload register value.

The duty cycle of a PWM channel is controlled through its capture/compare register CCR

$$D = \frac{CCR}{ARR + 1} \tag{6.23}$$

Consequently, when the SVPWM algorithm calculates the duty cycles D_a, D_b, and D_c, these values are converted into the corresponding timer compare values:

$$CCR_a = D_a(ARR + 1) \tag{6.24}$$

$$CCR_b = D_b(ARR + 1) \tag{6.25}$$

$$CCR_c = D_c(ARR + 1) \tag{6.26}$$

The timer then generates the corresponding switching signals automatically in hardware.

❖ **Example of timer configuration:**

As a simple example, suppose that the timer clock is:

$$f_{TIM} = 72\ MHz$$

and a PWM frequency of

$$f_{PWM} = 10KHz$$

is required.

If the prescaler is selected as

$$PSC = 0$$

the required auto-reload value for edge-aligned PWM is:

$$ARR = \frac{72\ x10^6}{10\ x\ 10^6} - 1$$

$$ARR = 7199$$

Therefore, the timer counts from 0 to 7199, corresponding to a PWM period of

$$TIM_{PWM} = \frac{1}{10\ 000} = 100\ \mu s$$

For example, if the SVPWM algorithm produces

$$D_a = 0.60$$

the corresponding compare value is approximately

$$CCR_a = 0.60(7200) = 4320.$$

The same procedure is applied to phases b and c.

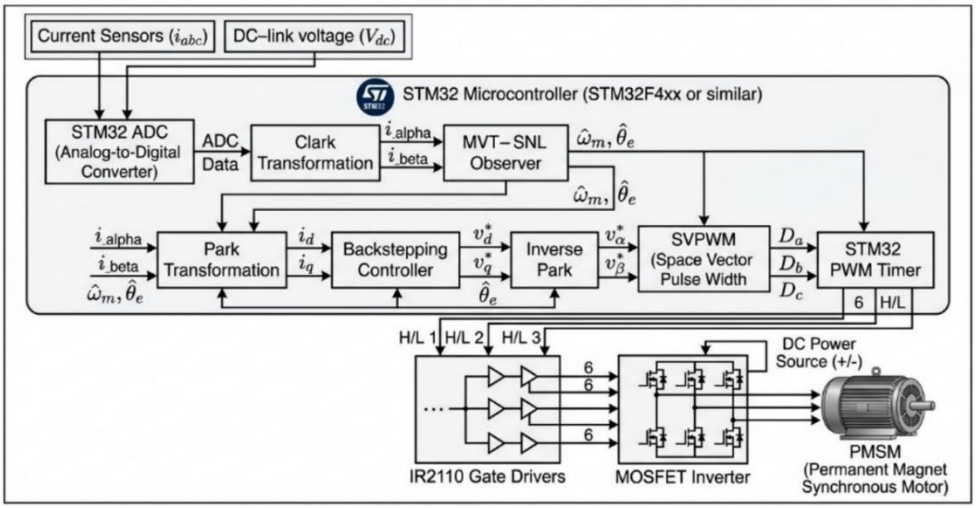


Figure 6-21: STM32- based control architecture

2.4. Software Architecture of the Real-Time Control System:

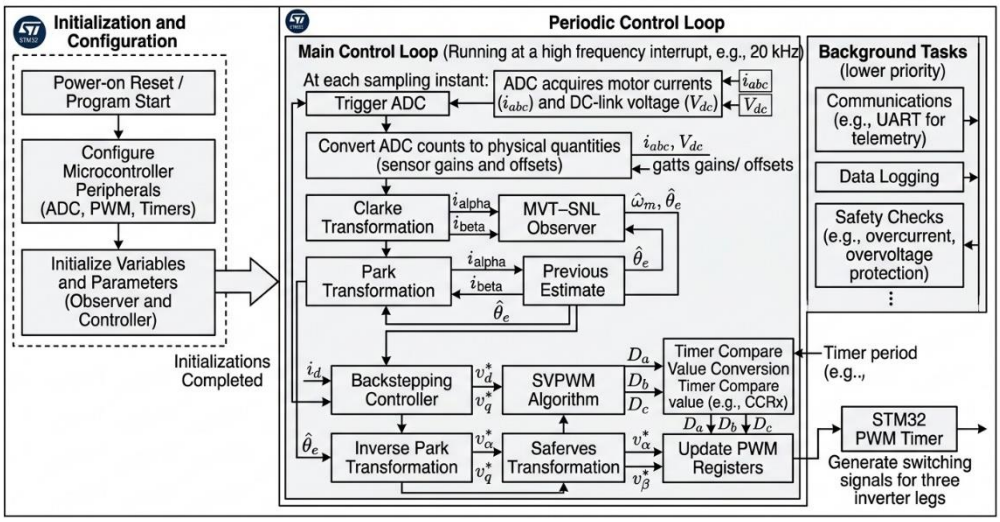


Figure 6-22: General Software Architecture of Real Time PMSM Control

2.5. Complete Real-Time Control Algorithm:

After configuring the hardware peripherals, the complete sensorless PMSM control algorithm is implemented on the STM32 microcontroller. The program combines the MVT–SNL observer, Backstepping controller, coordinate transformations, and SVPWM algorithm into a single real-time control loop.

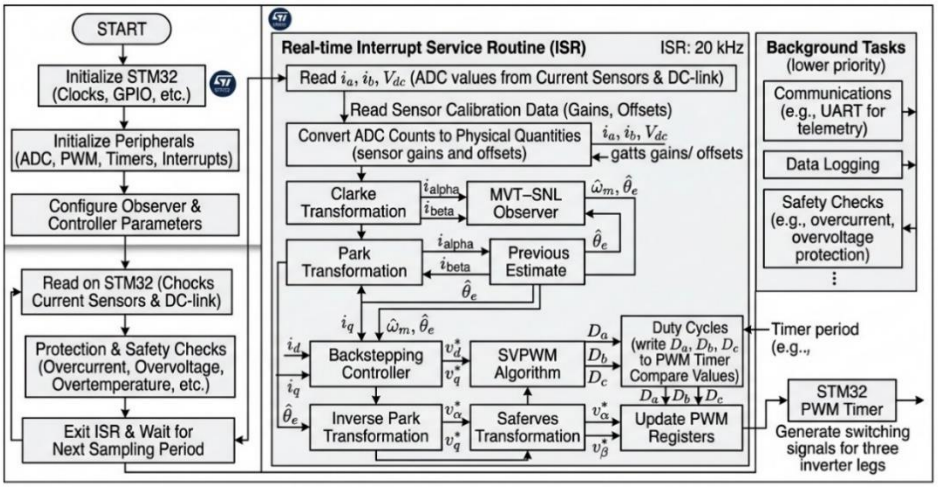


Figure 6-23: Complete Real Time Execution Sequence

General Conclusion

This thesis was devoted to the modeling, control, observation, and experimental implementation of a permanent magnet synchronous motor (PMSM) using a sensorless control strategy based on Backstepping control combined with an MVT–SNL observer. The main objective was to develop a control system capable of achieving accurate speed regulation while estimating the rotor variables and relevant uncertain parameters without relying on a mechanical position or speed sensor.

The work began with the mathematical modeling of the PMSM in the three-phase and rotating dq reference frames. The electrical and mechanical equations were established, providing the foundation for the subsequent controller and observer design. The influence of the motor parameters, including the stator resistance, inductances, permanent-magnet flux linkage, inertia, and viscous friction coefficient, was also considered in the development of the model.

A Backstepping nonlinear controller was then developed to regulate the rotor speed and stator currents. The recursive structure of the Backstepping methodology allowed the speed-control objective to be progressively transferred to the electrical subsystem, resulting in the calculation of the required stator voltage references. Particular attention was given to the treatment of the motor dynamics and uncertain quantities in order to obtain a robust control law suitable for practical implementation.

To achieve sensorless operation, an MVT–SNL observer was developed for estimating the unmeasured motor variables. The observer uses the available electrical measurements to reconstruct the rotor speed and electrical position required by the field-oriented control structure. The combination of the observer with the Backstepping controller makes it possible to reduce the dependence on conventional mechanical sensors while maintaining the required information for the control algorithm.

The voltage references generated by the controller were subsequently converted into inverter switching signals using Space Vector Pulse Width Modulation (SVPWM). The complete modulation procedure, including the transformation of the reference voltages, sector determination, switching-time calculation, and duty-cycle generation, was developed and implemented. The use of SVPWM provides an effective means of exploiting the available DC-link voltage while producing the three-phase voltage required by the PMSM.

An important part of the work concerned the experimental identification of the PMSM parameters. Dedicated tests were considered for determining the stator resistance R_s, the d- and q-axis inductances L_d and L_q, the permanent-magnet flux linkage $\boldsymbol{\Psi_{pm}}$, the rotor inertia J, and the viscous friction coefficient B. The identified parameters provide a more representative

motor model and establish the connection between the theoretical model and the physical machine.

The final stage of the thesis addressed the real-time implementation of the proposed control system. A three-phase voltage-source inverter was physically constructed using power MOSFETs and IR2110 gate-driver circuits. The STM32 microcontroller was selected as the digital control platform. Its timer peripherals were used to generate the complementary PWM signals and introduce the required dead time, while the ADC peripherals provide the measurements required by the control and observation algorithms. The resulting hardware and software architecture establishes the complete real-time control chain from measurement acquisition to inverter switching.

The overall control structure can therefore be summarized as

Measurements→MVT–SNL Observer→Backstepping Controller→SVPWM→Three-Phase VSI→PMSM

The different stages of the work demonstrate that the proposed approach provides a coherent framework for sensorless PMSM control, beginning with mathematical modeling and parameter identification and extending to digital implementation on an embedded platform.

Nevertheless, several aspects remain open for further improvement. In particular, the sensorless performance at very low and zero speed remains a challenging issue because the back-EMF information becomes weak in this operating region. Future work could therefore investigate improved low-speed estimation techniques, more robust adaptive laws, and hybrid sensorless strategies combining different observers according to the operating speed. Further experimental investigations could also consider parameter variations caused by temperature, magnetic saturation, inverter nonidealities, and load disturbances.

In addition, the real-time implementation can be further optimized by reducing the computational burden of the observer and controller, improving ADC/PWM synchronization, and exploiting the computational capabilities of more advanced STM32 devices or dedicated motor-control microcontrollers. Experimental validation under different speed references, load conditions, and parameter variations would provide an even more comprehensive assessment of the robustness of the proposed control strategy.

In conclusion, this thesis established a complete path from PMSM modeling and parameter identification to nonlinear sensorless control and real-time inverter implementation. The combination of the Backstepping controller, MVT–SNL observer, and SVPWM technique provides a suitable foundation for the development of a practical sensorless PMSM drive and

demonstrates the feasibility of transferring the proposed theoretical control methodology to an embedded experimental platform.

.

References Bibliographies

[1] Krishnan, R., Permanent Magnet Synchronous and Brushless DC Motor Drives, CRC Press, 2010.

[2] Pillay, P., and Krishnan, R., "Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives, Part II: The Brushless DC Motor Drive," IEEE Transactions on Industry Applications, vol. 25, no. 2, pp. 274–279, 1989.

[3] Bose, B. K., Modern Power Electronics and AC Drives, Prentice Hall, 2002.

[4] Hassan K. Khalil, Nonlinear Systems, 3rd ed., Prentice Hall, 2002.

[5] Isidori, A., Nonlinear Control Systems, 3rd ed., Springer, 1995.

[6] Tanaka, K., and Wang, H. O., Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach, Wiley, 2001.

[7] Takagi, T., and Sugeno, M., "Fuzzy Identification of Systems and Its Applications to Modeling and Control," IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 1, pp. 116–132, 1985.

[8] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., Linear Matrix Inequalities in System and Control Theory, SIAM Studies in Applied Mathematics, 1994.

[9] Slotine, J.-J. E., and Li, W., Applied Nonlinear Control, Prentice Hall, 1991.

[10] Utkin, V. I., Sliding Modes in Control and Optimization, Springer-Verlag, 1992.

[11] Marino, R., Tomei, P., and Verrelli, C. M., Induction Motor Control Design, Springer, 2010.

[13] Chen, B., Liu, X., Tong, S., and Lin, C., "Observer-Based Stabilization of T–S Fuzzy Systems," IEEE Transactions on Fuzzy Systems, vol. 16, no. 3, pp. 652–663, 2008.

[14] Tanaka, K., Ikeda, T., and Wang, H. O., "Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-Based Designs," IEEE Transactions on Fuzzy Systems, vol. 6, no. 2, pp. 250–265, 1998.

[15] Ortega, R., Loria, A., Nicklasson, P. J., and Sira-Ramirez, H., Passivity-Based Control of Euler–Lagrange Systems, Springer, 1998.

[16] Farza, M., M'Saad, M., and Rossignol, L., "Observer Design for Nonlinear Systems Application to Speed Sensorless Induction Motor," Control Engineering Practice, vol. 8, no. 12, pp. 1353–1361, 2000.

[17] Besançon, G., Nonlinear Observers and Applications, Springer, 2007.

[18] Rajamani, R., Observers for Lipschitz Nonlinear Systems, Springer, 1998.

[19] Krishnan, R., Permanent Magnet Synchronous and Brushless DC Motor Drives, CRC Press, 2010.

[20] Zemouche, A., Boutayeb, M., and Bara, G. I., "Observers for a Class of Lipschitz Systems with Extension to Fuzzy Systems," IEEE Transactions on Circuits and Systems II, vol. 55, no. 1, pp. 93–97, 2008.

[21] Chen, W.-H., Ballance, D. J., Gawthrop, P. J., and O'Reilly, J., "A Nonlinear Disturbance Observer for Robotic Manipulators," IEEE Transactions on Industrial Electronics, vol. 47, no. 4, pp. 932–938, 2000.

[22] Yu, W., and Kaynak, O., "Sliding-Mode Control With Soft Computing: A Survey," IEEE Transactions on Industrial Electronics, vol. 56, no. 9, pp. 3275–3285, 2009.

[23] Takagi, T., and Sugeno, M., "Fuzzy Identification of Systems and Its Applications to Modeling and Control," IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 1, pp. 116–132, 1985.

[24] Tanaka, K., and Wang, H. O., Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach, Wiley, 2001.

[25] Taniguchi, T., Tanaka, K., Wang, H. O., and Yamafuji, K., "Model Construction, Rule Reduction, and Robust Compensation for Generalized Form of Takagi–Sugeno Fuzzy Systems," IEEE Transactions on Fuzzy Systems, vol. 9, no. 4, pp. 525–538, 2001.

[26] Tanaka, K., Ikeda, T., and Wang, H. O., "Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-Based Designs," IEEE Transactions on Fuzzy Systems, vol. 6, no. 2, pp. 250–265, 1998.

[27] Ichalal, D., Marx, B., Ragot, J., and Maquin, D., "State Estimation of Takagi–Sugeno Systems with Unmeasurable Premise Variables," IET Control Theory & Applications, vol. 4, no. 5, pp. 897–908, 2010.

[28] Lendek, Z., Babuška, R., De Schutter, B., and Hellendoorn, H., Stability Analysis and Nonlinear Observer Design Using Takagi–Sugeno Fuzzy Models, Springer, 2010.

[29] Guerra, T. M., Kruszewski, A., Vermeiren, L., and Tirmant, H., "Conditions of Output Stabilization for Nonlinear Models in the Takagi–Sugeno's Form," Fuzzy Sets and Systems, vol. 157, no. 9, pp. 1248–1259, 2006.

[30] Zribi, M., Chiasson, J., and Stanković, A., "Observer-Based Backstepping Control of Permanent Magnet Synchronous Motors," IEEE Transactions on Control Systems Technology, vol. 18, no. 6, pp. 1353–1365, 2010.

[31] Kchaou, M., Chaari, A., and Boussak, M., "Backstepping Sensorless Control for PMSM Drives Based on Nonlinear Observer," International Journal of Electrical Power & Energy Systems, vol. 81, pp. 354–364, 2016.

[32] Xu, D., Zhao, J., and Zhang, Y., "Sensorless PMSM Drive Based on Adaptive Backstepping Sliding Mode Control," IEEE Transactions on Power Electronics, vol. 27, no. 8, pp. 3865–3875, 2012.

[33] Wang, H., Karimi, H. R., Liu, P. X., and Xiang, Y., "Adaptive Neural Control of PMSM Servo System with Unknown Dead-Zone," IEEE Transactions on Industrial Electronics, vol. 61, no. 1, pp. 503–513, 2014.

[34] D. G. Holmes and T. A. Lipo, Pulse Width Modulation for Power Converters: Principles and Practice, IEEE Press/Wiley, 2003.

[35] B. K. Bose, *Modern Power Electronics and AC Drives*, Prentice Hall, 2002.

[36] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd ed., John Wiley & Sons, 2003.

[37] D. G. Holmes and T. A. Lipo, Pulse Width Modulation for Power Converters: Principles and Practice, IEEE Press/Wiley, 2003.

[38] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd ed., John Wiley & Sons, 2003.

[39] B. K. Bose, *Modern Power Electronics and AC Drives*, Prentice Hall, 2002.

Mathematical Space vector representation + sector determination

[40] Holmes, D. G., and Lipo, T. A., Pulse Width Modulation for Power Converters: Principles and Practice, IEEE Press/Wiley, 2003.

[41] Bose, B. K., *Modern Power Electronics and AC Drives*, Prentice Hall, 2002.

[42] Mohan, N., Undeland, T. M., and Robbins, W. P., *Power Electronics: Converters, Applications, and Design*, 3rd ed., John Wiley & Sons, 2003.

[43] J Sastry, P Bakas, H Kim, L Wang, A Marinopoulos. Evaluation of cascaded H-bridge inverter for utility-scale photovoltaic systems. Renew. Energy. 2014; 69: 208-218.

[44] AK Gupta, S Member, AM Khambadkone. A Space Vector PWM Scheme for Multilevel Inverters Based on Two-Level Space Vector PWM. 2006; 53(5): 1631-1639.

[45] M Valan Rajkumar, PS Manoharan. FPGA based multilevel cascaded inverters with SVPWM algorithm for photovoltaic system. Sol. Energy. 2013; 87: 229-245.

[46] S Wei, B Wu. A general space vector PWM control algorithm for multilevel inverters. Eighteenth Annu. IEEE Appl. Power Electron. Conf. Expo. 2003. APEC ’03. 2003; 1(1): 562-568.

[47] X Yang, C Wang, L Shi, Z Xia. Generalized Space Vector Pulse Width Modulation Technique for Cascaded Multilevel Inverters. Int. J. Control Autom. 2014 ; 7(1) : 11-26.

[48] X Xu, Y Zou, K Ding, F Liu. Cascade multilevel inverter with phase-shift SPWM and its application in STATCOM. 30th Annu. Conf. IEEE Ind. Electron. Soc. 2004. IECON 2004. 2004; 2: 1139-1143.

[49] D. Prasad, and C. Dhanamjayulu,” A Review of Control Techniques and Energy Storage for Inverter-Based Dynamic Voltage Restorer in Grid-Integrated Renewable Sources,” Hindawi Mathematical Problems in Engineering Volume Article ID 6389132, 43 pages, 2022 https://doi.org/10.1155/2022/6389132.

[50]. M. Agarwal, and D. K. Joshi, “Design consideration for insulated gate induction motors,” February 2018.

[51]. M. Sharma, and M. Lalwani, “Performance Evaluation of Three-Phase Induction Motor Drive Fed from Z-Source Inverter,” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 8 ,2018 pp. 6098-6109. © Research India Publications. http://www.ripublication.com.

[51]. S. Janiga, S.S. Nawaz, S.K. Tummala, and S.V. Pinni, “Speed Control of an Induction Motor Fed by an Inverter Using dSPACE Controller,” E3S Web of Conferences 87 SeFet 2019. https://doi.org/10.1051/e3sconf/201 0 0, 01002 2019.

[52]. V. K. Pal, and S. Singh, “Design of Sinusoidal Pulse Width Modulation 3 Phase Bridge Inverter,” International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056. Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072.

ANNEXES

Annex A-Matlab Program

MATLAB code for PMSM Observer (Figure 25: Observer Convergence)

```
clc;
clear;
close all;

%% PMSM Parameters
Rs = 1.4;
L  = 0.006;
p  = 4;
Phi_f = 0.15;
J = 0.001;
f = 0.0001;

%% Observer gains
L1 = 200;
L2 = 200;
L3 = 50;
L4 = 50;

%% Initial conditions
x  = [0;0;100];
xh = [2;-2;50];

%% Inputs
vd = 10;
vq = 20;

%% Simulation
dt = 1e-4;
Tf = 0.2;
N = Tf/dt;
```

```
E = zeros(3,N);
for k=1:N

id = x(1);
iq = x(2);
w  = x(3);

%% Real system
xid_dot = -Rs/L*id + p*w*iq + vd/L;

xiq_dot = -Rs/L*iq ...
- p*w*id ...
- p*Phi_f/L*w ...
+ vq/L;

xw_dot  = 3*p*Phi_f/(2*J)*iq - f/J*w;

xdot = [xid_dot;
xiq_dot;
xw_dot];

x = x + dt*xdot;

%% Observer
idh = xh(1);
iqh = xh(2);
wh  = xh(3);

y  = [id;iq];
yh = [idh;iqh];

e = y - yh;
```

```
xidh_dot = -Rs/L*idh ...
+ p*wh*iqh ...
+ vd/L ...
+ k1*e(1);

xiqh_dot = -Rs/L*iqh ...
- p*wh*idh ...
- p*Phi_f/L*wh ...
+ vq/L ...
+ k2*e(2);

xwh_dot  = 3*p*Phi_f/(2*J)*iqh ...
- f/J*wh ...
+ k3*e(1) ...
+ k4*e(2);

xhdot = [xidh_dot;
xiqh_dot;
xwh_dot];

xh = xh + dt*xhdot;

%% Save errors
E(:,k) = x - xh;
end

%% Plot errors
t = 0:dt:Tf-dt;

figure;
plot(t,E(1,:),'LineWidth',2);
hold on;
plot(t,E(2,:),'LineWidth',2);
```

```
plot(t,E(3,:),'LineWidth',2);
xlabel('Time (s)');
ylabel('Estimation Errors');
legend('e_d','e_q','e_\omega');
grid on;
title('Observer Error Convergence');
```

MATLAB code for Full PMSM, MVT-NLS, BACKSTEPPING Controller

(Figure 26)

Observer MATLAB Function code:

```
function [x_hat_dot, x_hat] = MVT_NLS_Observer(y_meas, u_ctrl, x_hat_prev, dt)
% MVT_NLS_OBSERVER Hardcoded SPMSM State Estimator using MVT-NLS LMI Gains.
%
% Inputs:
%   y_meas     : Measured currents [i_d; i_q] (2x1) [A]
%   u_ctrl     : Applied control voltages [v_d; v_q] (2x1) [V]
%   x_hat_prev : Previous estimated states [i_d_hat; i_q_hat; w_e_hat] (3x1)
%   dt         : Integration step size / sample time [s]
%
% Outputs:
%   x_hat_dot  : Continuous state derivative vector [d_id; d_iq; d_we] (3x1)
%   x_hat      : Integrated state estimates [i_d_hat; i_q_hat; w_e_hat] (3x1)

%
================================================================
======
% PRE-CALCULATED MOTOR COEFFICIENTS
%
================================================================
======
Rs     = 0.5;
Ls     = 0.01;
```

```
psi_pm  = 0.1;
J       = 0.01;
B       = 0.001;
TL      = 1;
p       = 4;

a = (3*p^2*psi_pm)/(2*J);
b = B/J;

%
==========================================================================
======
% 1. Computed LMI Gain Matrix L (3x2)
L = [ -107.9 , -0.1 ;
0.2 , -108.0 ;
41.0 , 2335.7 ];

% 2. Extract Inputs and Previous States
i_d = y_meas(1);
i_q = y_meas(2);

v_d = u_ctrl(1);
v_q = u_ctrl(2);

i_d_hat = x_hat_prev(1);
i_q_hat = x_hat_prev(2);
w_e_hat = x_hat_prev(3);

% 3. Output Estimation Errors e = y - y_hat
e_id = i_d - i_d_hat;
e_iq = i_q - i_q_hat;
% 4. Non-Linear Dynamics Vector f(x_hat, u)
f_id = -115.172414 * i_d_hat + 3 * w_e_hat * i_q_hat + 68.965517 * v_d;
```

```
f_iq = -115.172414 * i_q_hat - 3 * w_e_hat * i_d_hat - 35.172414 * w_e_hat + 68.965517 * v_q;
f_we = 7650.0 * i_q_hat - 43.333333 * w_e_hat - 10000.0;
% 5. Apply LMI Observer Feedback Gain
d_id_hat = f_id + L(1,1)*e_id + L(1,2)*e_iq;
d_iq_hat = f_iq + L(2,1)*e_id + L(2,2)*e_iq;
d_we_hat = f_we + L(3,1)*e_id + L(3,2)*e_iq;

x_hat_dot = [d_id_hat; d_iq_hat; d_we_hat];

% 6. Numerical Integration Step (Euler 1st Order)
x_hat = x_hat_prev + x_hat_dot * dt;

end
```

Backstepping Controller MATLAB Function code:

```
function [vd, vq] = backstepping_controller(e_omega,e_iq,e_id, xhat)
% ============================================================
% BACKSTEPPING CONTROLLER MATCHING THESIS COEFFICIENTS
% Input: xhat = [id_hat; iq_hat; we_hat]
% ============================================================
%% Extract Estimated States
id_hat = xhat(1);
iq_hat = xhat(2);
we_hat = xhat(3);

%% Target Reference
wref = 261.8;

e_omega = wref - we_hat;

iq_ref = 0.16667*e_omega + 0.000416667*we_hat + 1.6667;
e_iq = iq_hat - iq_ref;
e_id = id_hat; % since id_ref = 0
```

```
% we_hat_dot = 240*iq_hat - 0.1*we_hat - 400*TL (where TL = 1)
we_hat_dot = 240 * iq_hat - 0.1 * we_hat - 400;

diq_ref = -0.16625 * we_hat_dot;

%% Step 3: d-axis voltage calculation
% vd = -1.6*e_id - 0.005*we_hat*iq_hat
vd = -1.6 * e_id - 0.005 * we_hat * iq_hat;

%% Step 2: q-axis voltage calculation
% vq = -2*e_iq + 1.2*e_omega + 0.4*iq_hat + 0.005*we_hat*id_hat + 0.1*we_hat +
0.005*diq_ref
vq = -2 * e_iq + 1.2 * e_omega + 0.4 * iq_hat + 0.005 * we_hat * id_hat + 0.1 * we_hat
+ 0.005 * diq_ref;

Vdc = 300;
Vmax = Vdc / sqrt(3); % 173.2V maximum phase voltage magnitude
Vmag = sqrt(vd^2 + vq^2);

if Vmag > Vmax
vd = (vd / Vmag) * Vmax;
vq = (vq / Vmag) * Vmax;
end
end
```

machine parameters:

$R_s = 0.5\,\Omega,$

$L_s = 0.005\,H,$

$\psi_{pm} = 0.1\,Wb,$

$p = 4,$

$J = 0.01\,kg \cdot m^2,$

$B = 0.001\,N \cdot m \cdot s/rad$

Annex B-Data-Sheets:

International
IOR Rectifier

Data Sheet No. PD60147 rev.V

IR2110(S)PbF/IR2113(S)PbF

HIGH AND LOW SIDE DRIVER

Features

- Floating channel designed for bootstrap operation
  Fully operational to +500V or +600V
  Tolerant to negative transient voltage
  dV/dt immune
- Gate drive supply range from 10 to 20V
- Undervoltage lockout for both channels
- 3.3V logic compatible
  Separate logic supply range from 3.3V to 20V
  Logic and power ground ±5V offset
- CMOS Schmitt-triggered inputs with pull-down
- Cycle by cycle edge-triggered shutdown logic
- Matched propagation delay for both channels
- Outputs in phase with inputs

Product Summary

V_{OFFSET} (IR2110)	500V max.
(IR2113)	600V max.
I_O+/-	2A / 2A
V_{OUT}	10 - 20V
$t_{on/off}$ (typ.)	120 & 94 ns
Delay Matching (IR2110)	10 ns max.
(IR2113)	20ns max.

Packages



14-Lead PDIP
IR2110/IR2113

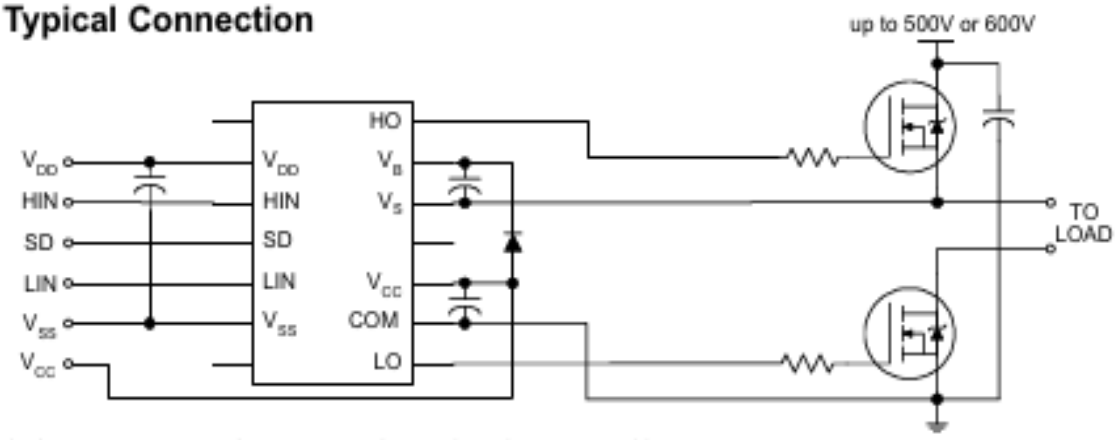


16-Lead SOIC
IR2110S/IR2113S

Description

The IR2110/IR2113 are high voltage, high speed power MOSFET and IGBT drivers with independent high and low side referenced output channels. Proprietary HVIC and latch immune CMOS technologies enable ruggedized monolithic construction. Logic inputs are compatible with standard CMOS or LSTTL output, down to 3.3V logic. The output drivers feature a high pulse current buffer stage designed for minimum driver cross-conduction. Propagation delays are matched to simplify use in high frequency applications. The floating channel can be used to drive an N-channel power MOSFET or IGBT in the high side configuration which operates up to 500 or 600 volts.

Typical Connection



(Refer to Lead Assignments for correct pin configuration). This/These diagram(s) show electrical connections only. Please refer to our Application Notes and DesignTips for proper circuit board layout.

life.augmented

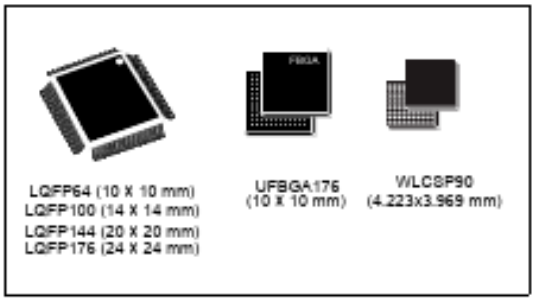
STM32F405xx STM32F407xx

Arm® Cortex®-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera

Datasheet - production data

Features

- Core: Arm® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/ 1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Memories
  - Up to 1 Mbyte of Flash memory
  - Up to 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM
  - 512 bytes of OTP memory
  - Flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
- LCD parallel interface, 8080/6800 modes
- Clock, reset and supply management
  - 1.8 V to 3.6 V application supply and I/Os
  - POR, PDR, PVD and BOR
  - 4-to-26 MHz crystal oscillator
  - Internal 16 MHz factory-trimmed RC (1% accuracy)
  - 32 kHz oscillator for RTC with calibration
  - Internal 32 kHz RC with calibration
- Low-power operation
  - Sleep, Stop and Standby modes
  - V_{BAT} supply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
- 3×12-bit, 2.4 MSPS A/D converters: up to 24 channels and 7.2 MSPS in triple interleaved mode
- 2×12-bit D/A converters
- General-purpose DMA: 16-stream DMA controller with FIFOs and burst support



- Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 168 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
- Debug mode
  - Serial wire debug (SWD) & JTAG interfaces
  - Cortex-M4 Embedded Trace Macrocell™
- Up to 140 I/O ports with interrupt capability
  - Up to 136 fast I/Os up to 84 MHz
  - Up to 138 5 V-tolerant I/Os
- Up to 15 communication interfaces
  - Up to 3 × I^2C interfaces (SMBus/PMBus)
  - Up to 4 USARTs/2 UARTs (10.5 Mbit/s, ISO 7816 interface, LIN, IrDA, modem control)
  - Up to 3 SPIs (42 Mbits/s), 2 with muxed full-duplex I^2S to achieve audio class accuracy via internal audio PLL or external clock
  - 2 × CAN interfaces (2.0B Active)
  - SDIO interface
- Advanced connectivity
  - USB 2.0 full-speed device/host/OTG controller with on-chip PHY
  - USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
  - 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII