MRAS sensorless control strategy for anisotropic PMSM

Author:
Lorenzo Tranquilli

Course : Electrical Engineering - Ingegneria Elettrica
Academic year: 2024 - 2025
ID: 242281
Advisor: Prof. Luigi Piegari

Abstract

The growing introduction of permanent magnet synchronous motor (PMSM) in industrial automation, electric mobility, and high energy-efficiency applications requires the development of increasingly advanced, reliable, and cost-effective control strategies. In this context, sensor-less techniques play a fundamental role, enabling the elimination of mechanical position and speed sensors, thereby reducing costs, increasing reliability, and improving the overall robustness of the system.

This thesis focuses on the implementation and analysis of a model reference adaptive system (MRAS)-based estimation method applied to an anisotropic permanent magnet synchronous motor. After presenting the mathematical model of the motor in the dq reference frame, the MRAS method is described and implemented by defining a reference model and an adaptive model, interconnected through an adaptation law derived according to Lyapunov stability criteria.

The work includes system simulations carried out in the MATLAB/Simulink environment and the evaluation of the algorithm's performance in both continuous-time and discrete-time domains. The results highlight the capability of the MRAS approach to accurately estimate speed (and/or rotor position) in both proposed cases, confirming it as a valid solution for sensorless applications in anisotropic PMSMs, while still requiring appropriate design considerations to ensure reliability and robustness.

The thesis demonstrates that the MRAS technique represents an effective solution for the sensorless control of anisotropic PMSMs, offering a suitable trade-off between implementation complexity and dynamic performance.

Keywords: sensorless, MRAS, electric motors, permanent magnets

Abstract in lingua italiana

La crescente diffusione dei motori sincroni a magneti permanenti (PMSM) nei settori dell'automazione industriale, della mobilità elettrica e delle applicazioni ad alta efficienza energetica richiede lo sviluppo di strategie di controllo sempre più avanzate, affidabili e a basso costo. In tale contesto, le tecniche sensorless rivestono un ruolo fondamentale, consentendo l'eliminazione dei sensori meccanici di posizione e velocità, con conseguente riduzione dei costi, aumento dell'affidabilità e miglioramento della robustezza complessiva del sistema.

Il seguente lavoro di tesi si concentra sull'implementazione e sull'analisi di un sistema di stima basato su model eeference adaptive system (MRAS) applicato a un motore sincrono a magneti permanenti anisotropo. Dopo una descrizione del modello matematico del motore nel sistema di riferimento dq, viene descritto ed implementato il metodo MRAS, progettato definendo un modello di riferimento e un modello adattativo, collegati attraverso una legge di adattamento derivata secondo criteri di stabilità basati su Lyapunov.

L'elaborato include la simulazione del sistema in ambiente MATLAB/Simulink e la valutazione delle prestazioni dell'algoritmo sia in tempo continuo che discreto. I risultati evidenziano la capacità dell'approccio MRAS di stimare accuratamente la velocità (e/o la posizione rotorica) in entrambi i casi proposti, confermandosi un una soluzione valida per applicazioni sensor-less su PMSM anisotropi, pur richiedendo adeguate accortezze progettuali per garantirne affidabilità e robustezza.

La tesi dimostra come l'impiego della tecnica MRAS rappresenti una soluzione efficace per il controllo sensorless di PMSM anisotropi, offrendo un compromesso ottimale tra complessità implementativa e prestazioni dinamiche.

Parole chiave: sensorless, MRAS, motori elettrici, magneti permanenti.

Contents

Introduction

Anisotropic permanent magnet synchronous motors are increasingly used in high-efficiency systems and high-performance applications, due to their ability to generate high torque with a high power density [1] . When properly controlled, these motors allow the optimization of operational performance by using advanced strategies such as field-oriented control (FOC) combined with maximum torque per ampere (MTPA) and maximum torque per voltage (MTPV) techniques, which improve energy efficiency and maximize available torque under different operating conditions.
The main objective of this thesis is the study and implementation of a model reference adaptive system algorithm for the sensorless control of an anisotropic PMSM. The MRAS approach, based on the definition of a reference model and an adaptive model connected through an adaptation law, provides an effective solution to accurately estimate rotor speed and position, while presenting some critical aspects regarding sensitivity to operating conditions and design parameters.

The thesis is structured into three main chapters. Chapter 1 provides a detailed mathematical description of the anisotropic PMSM and field-oriented control with MTPA and MTPV strategies, introducing the theoretical foundations necessary to understand the motor's behaviour and the possibilities for optimizing its operation. Chapter 2 introduces the MRAS algorithm, describing its theoretical basis, implementation in MATLAB/Simulink, and preliminary simulation results, with particular attention to design parameters and observer stability. Finally, Chapter 3 presents the complete implementation of the discrete-time system model, comparing the simulation results and examining the algorithm's performances.

This work aims to provide a comprehensive analysis of the MRAS algorithm applied to anisotropic PMSMs, highlighting both its potential and its limitations, with the goal of contributing to the understanding of advanced sensorless control strategies and their practical applications in high-performance motors.

1 | Anisotropic PMSM

Permanent magnet (PM) motors were introduced as an evolution of conventional electrical machines in order to overcome some limitations of traditional motor technologies. Before the widespread introduction of PM-motors, most industrial applications were based on induction motors and wound-field synchronous motors. In such machines, the magnetic field in the rotor is generated by electric current flowing through windings. Although these machines proved to be robust and reliable, they were characterized by additional energy losses, larger dimensions and a reduced efficiency, especially in variable-speed applications. In particular, the continuos excitation of the rotor windings leads to additional copper losses and thermal stress, decreasing the overall efficiency and limiting the performances of the machine under dynamic operating condition. In addition, the use of brushes as excitation system for the rotor increases the structural and control complexity of the machine, resulting in higher maintenance over its lifetime. Indeed, in PM- motors, the magnetic field is directly generated by the permanent magnets placed inside the rotor. Such excitation system allows the elimination of rotor windings resulting in a simpler structure that requires less maintenance. This technological advancement led to a significant increase in efficiency, power density and dynamic performance. These characteristics make PMSM suitable for applications as electric vehicles, robotics, industrial drives and renewable energy systems, where efficiency and fast dynamics are the main drivers [2],[3],[4].

In this chapter, the anisotropic permanent magnets synchronous machines are introduced considering their main characteristics. Then, a mathematical model is elaborated, including the control algorithm used in the next chapters.

1.1. Characteristics of anisotropic PMSM

The permanent magnets most widely used in PMSM are: NdFeB (neodymium iron boron), SmCo (samarium cobalt) and ferrite. The choice of the material implemented in the motor directly affects its performances, in particular in terms of efficiency and electromagnetic torque [5]. Their main characteristics are high energy density, high coercivity, and thermal stability, which allow the machine to generate significant magnetic flux while maintaining performance under varying temperature conditions. All of them are characterized by high reluctance, similar to that of air. Therefore, depending on their placement in the rotor, the rotor geometry may be magnetically asymmetric, unlike isotropic PMSMs where the magnetic properties are uniform along all directions.
The two main basic configurations are reported in Fig.1.1:

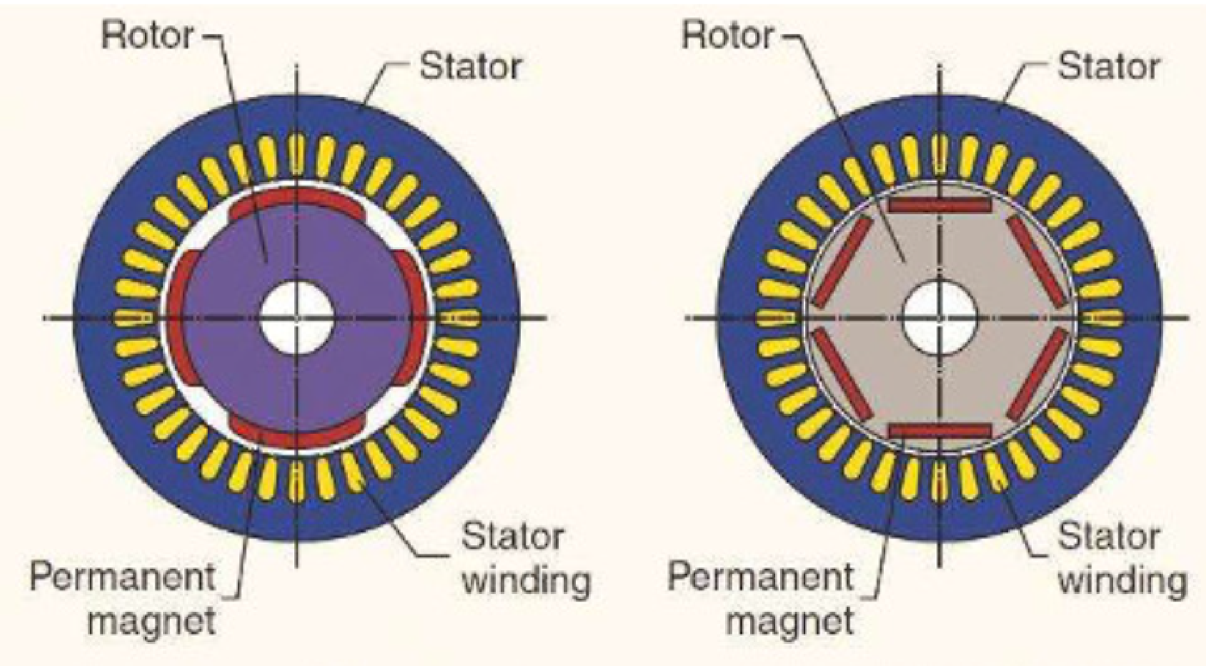

Figure 1.1: surface-mounted PMSM (left) and internal-mounted PMSM (right) [6].

In surface-mounted PMSMs (SM-PMSM), the permanent magnets are arranged around the rotor surface. Such configuration, independently on the rotor's angular position, ensures that the reluctance in the radial direction remains nearly constant. Consequently, SM-PMSMs are considered nearly magnetically isotropic, and the torque production relies almost entirely on the interaction between the stator current and the rotor magnetic field (magnetic torque).
While in the interior PMSM (IPMSM), the magnets are embedded inside the rotor, gen-

erating a non-uniform radial magnetic reluctance profile. The anisotropy introduces, in addition to the magnetic torque of the machine (resulting from the interaction between stator current and magnetic field generated by magnets), the reluctance torque, which can be exploited through a proper current control.
The anisotropic design of the IPMSM offers several advantages:

1. Higher torque density: by exploiting both magnetic and reluctance torque, the IPMSM can achieve higher torque for the same machine size compared to an SM-PMSM.

2. Improved efficiency: the reluctance torque reduces the required stator current for a given torque, reducing copper losses.

3. Better performance at high speeds: the embedded magnets reduce mechanical stress at high speeds.

However, the non-uniform rotor geometry also introduces challenges. In particular the processing of magnets and their encasement increase the complexity form the manufacturing point of view, but also the costs associated. The saliency (difference between direct-axis and quadrature-axis inductances) requires a more complex model, so the sensitivity of the parameters estimation increases and advanced control strategies are required. Also torque ripple may appear due to the variation of reluctance along the rotor.
The next chapter introduces the mathematical model of the anisotropic PMSM, providing a foundation for analyzing its dynamic behavior and designing effective control.

1.2. Mathematical model of anisotropic PMSM

The machines equations follow the space vector theory, amplitude invariant formulation, and Park transformation to obtain a suitable model oriented to the rotor flux, in particular:

$$\underline{\mathbf{x_s}} = x_\alpha + jx_\beta = \frac{2}{3}(x_a + \alpha x_b + \alpha^2 x_c) \tag{1.1}$$

$$x_d + jx_q = \underline{\mathbf{x_s}} e^{-j\theta_e} \tag{1.2}$$

where: θ_e is rotor's electrical position, $\alpha = e^{j\frac{2\pi}{3}} = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$ and $\alpha^2 = e^{j\frac{4\pi}{3}} = -\frac{1}{2} - j\frac{\sqrt{3}}{2}$.

The following electrical equations, along with their Park transformation, must refer to electrical speed and position, while the mechanical equations involve mechanical quantities. As well-known, the relationship between them includes the number of pole pairs pp:

$$x_{electrical} = pp\, x_{mechanical} \tag{1.3}$$

In the following equations, the three-phase model of a PMSM obtained through Faraday's law approach is reported:

$$\begin{cases} v_{sa} = R_{sa} i_{sa} + \dfrac{\partial \Phi_{sa}}{\partial t} & \text{(1.4a)} \\ v_{sb} = R_{sb} i_{sb} + \dfrac{\partial \Phi_{sb}}{\partial t} & \text{(1.4b)} \\ v_{sa} = R_{sc} i_{sc} + \dfrac{\partial \Phi_{sc}}{\partial t} & \text{(1.4c)} \\ \Phi_{sa} = L_{saa} i_{sa} + L_{sab} i_{sb} + L_{sac} i_{sc} + \Psi_{ma} & \text{(1.4d)} \\ \Phi_{sb} = L_{sbb} i_{sb} + L_{sab} i_{sa} + L_{sbc} i_{sc} + \Psi_{mb} & \text{(1.4e)} \\ \Phi_{sc} = L_{scc} i_{sc} + L_{sca} i_{sa} + L_{scb} i_{sb} + \Psi_{mc} & \text{(1.4f)} \\ & \text{(1.4g)} \end{cases}$$

where the first three equations represent the Kirchhoff's voltage laws (KVLs) of the machine, where v_{sabc} and i_{sabc} are the stator phase voltages and currents, R_{sa}, R_{sb} and R_{sc} are the stator phase resistances, L_{saa}, L_{sbb} and L_{scc} are the phase self-inductances, and L_{sab}, L_{sbc} and L_{sca} (assumed all equal and minus half of the previous terms) are the phase mutual-inductances. Φ_{sabc} and Ψ_{mabc} are the stator and rotor magnetic fluxes.

By assuming all the stator windings equal and symmetric, it means that $R_{sa} = R_{sb} = R_{sc} = R_s$ and $L_{saa} = L_{sbb} = L_{scc} = l_s$. Therefore, unique stator phase resistance R_s and stator self inductance ls are introduced in the equations. Considering flux conservation approach it's possible to write:

$$v_{sk} = R_s i_{sk} + l_s i_{sk} + \frac{\partial \phi_{sk,tot}}{\partial t}, \quad k = a, b, c \tag{1.5}$$

where the first term corresponds to the resistive voltage drop, the second is the self-induced leakage flux contribution, and the third term corresponds to the total mutual flux.

The evaluation of the last terms, in each phase, can be achieved by evaluating the flux density in the air gap B, which is function of stator current distribution, electrical position and the air gap angular distribution between stator and rotor (in reality, the reciprocal of the air-gap is used in literature because it enables easiest computations).

Then, once the mutual fluxes in each phase are obtained, equations (1.5) can be rewritten in space vector form introducing the stator voltage and current space vectors as:

$$\underline{v_s} = R_s\underline{i_s} + (l_s + L_m)\frac{\partial \underline{i_s}}{\partial t} + L_1\frac{\partial}{\partial t}(\underline{i_s^*}e^{j2pp\theta}) + jpp\omega_r\phi_M e^{jpp\theta} \tag{1.6}$$

where $\underline{v_s}$ and $\underline{i_s}$ are the stator voltage and current space vectors, L_m is the magnetizing inductance which depends on geometrical features of the machine and describes the stator magnetic flux which closes only in the air gap , L_1 is the term introduced to model the anisotropy of the machine, ϕ_M is the permanent magnet flux linkage, θ is the rotor's mechanical position.

The second term can also be written by introducing the stator phase inductance $L_s = ls + L_m$, and express the total self-induced stator flux. The third term represents the back-emf of the flux density produced by the PM in the stator windings.

The resulting space vector equation for the electrical torque T_{el} is:

$$T_{el} = \frac{3}{2}pp\phi_M\Im(\underline{i_s}e^{-jpp\theta}) + \frac{3}{2}ppL_1\Im(\underline{i_s}^2e^{-j2pp\theta}) \tag{1.7}$$

where the first term is generated by the interaction of stator current and permanent magnet flux, while the second term results due to the anisotropy of the machine.

For control purposes, both equations (1.6) and (1.7) have been moved in the d-q reference frame, where d-axis is aligned with rotor flux (with rotor position), while the q-axis is shifted 90° leading the d-axis. Such change of reference frame is provided by multiplying both equations by $e^{-pp\theta}$, resulting in d-q model of the anisotropic PMSM:

$$\begin{cases} v_{sd} = R_s i_{sd} + L_d\dfrac{di_{sd}}{dt} - pp\omega_r L_q i_{sq} & \text{(1.8a)} \\ v_{sq} = R_s i_{sq} + L_q\dfrac{di_{sq}}{dt} + pp\omega_r(L_d i_{sd} + \phi_M) & \text{(1.8b)} \\ T_{el} = \dfrac{3}{2}pp\phi_M i_{sq} + \dfrac{3}{2}(L_d - L_q)i_{sd}i_{sq} & \text{(1.8c)} \end{cases}$$

where v_{sd} and v_{sq} are d-q stator voltage components, i_{sd} and i_{sq} are d-q stator current components, $L_d = L_s + L_1$ and $L_q = L_s - L_1$ are the d-q inductances of the machine.

Finally, the mechanical equations of the machine can be written as:

$$
\begin{cases}
T_{el} - T_{load} = J_{eq}\dfrac{d\omega_r}{dt} + B_0\omega_r & (1.9a)\\[2ex]
\dfrac{d\theta_{el}}{dt} = pp\omega_r & (1.9b)
\end{cases}
$$

where equation (1.9a) represents the mechanical shaft equation of the motor, in which T_{load} is the resistance load torque, J_{eq} is the inertia of the mechanical system around the shaft and B_0 is the proportional coefficients with speed to model the viscous friction torque.

The Simulink model of the anisotropic PMSM is represented in the following Fig.1.2 and Fig.1.3:

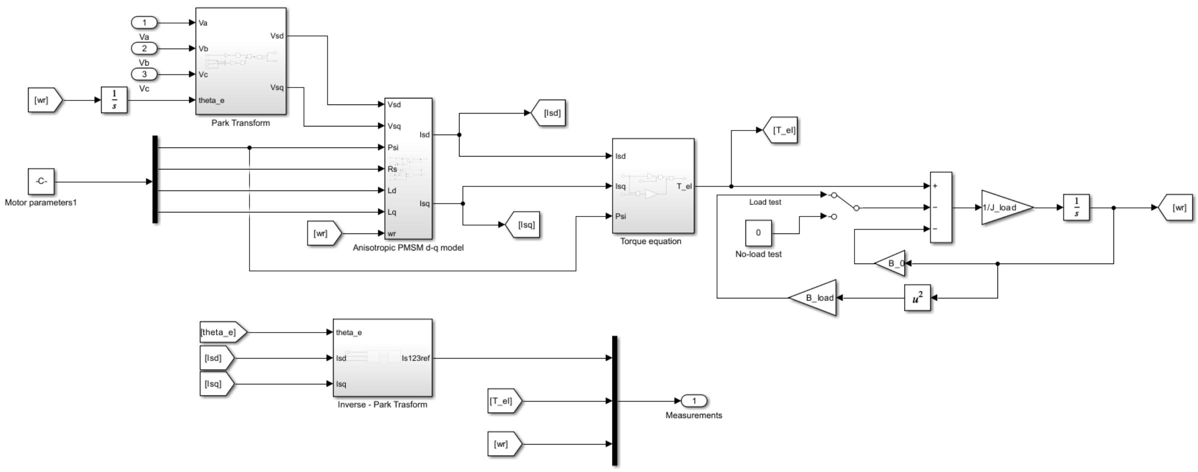

Figure 1.2: Simulink model of anisotropic PMSM.

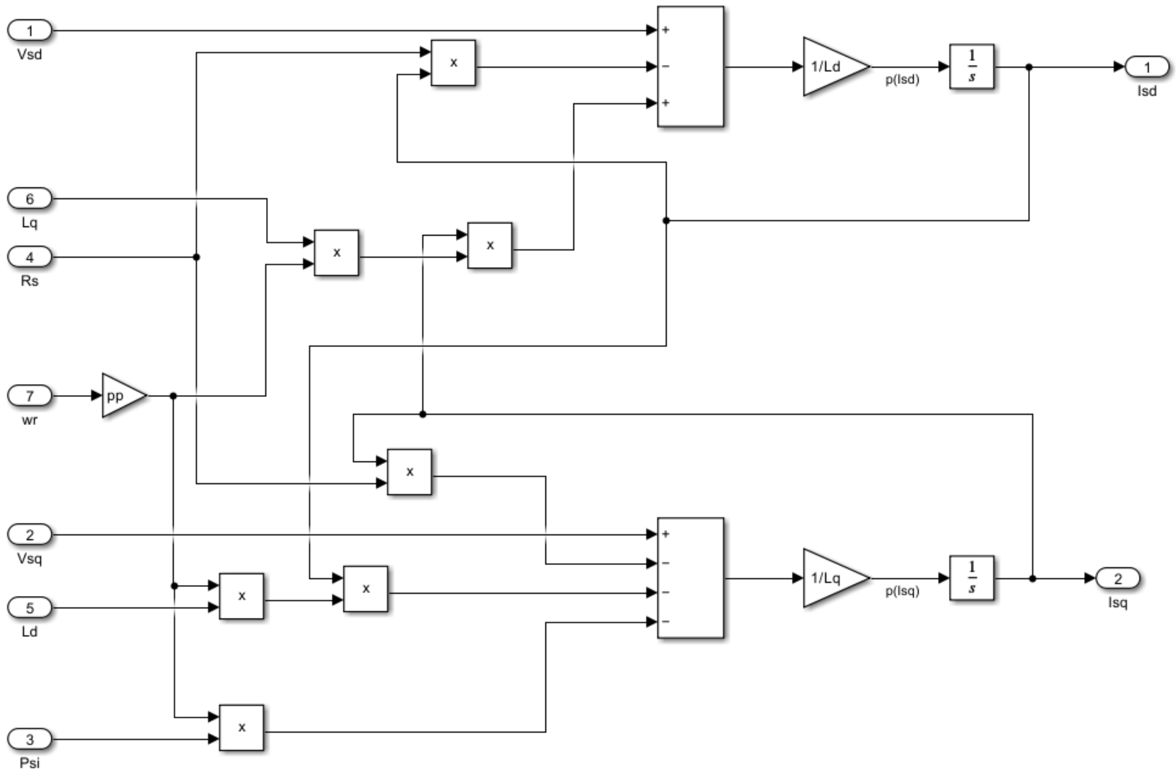

Figure 1.3: Simulink connections of anisotropic PMSM d-q model.

1.3. Field Oriented Control

The field oriented control is the standard control method to reach high performances for permanent magnets synchronous machines. Its main goal is to decouple the torque and flux controls identifying the corresponding effective current components. This simplifies the management of dynamics in the motor and optimizes its performances. Such decoupling is almost completely achievable trough the d-q model of the isotropic PMSM (which can be easily obtained from equations (2.9) , where the torque is only dependent by i_{sq} and the flux control mainly by i_{sd}). While for anisotropic PMSM, the decoupling is not possible, because of the anisotropic term in the torque equation (1.8c), which is proportional to $i_{sq}i_{sd}$. Anyway, the d-q model is still an advantageous reference frame, since it provides a structured basis for torque optimization strategies as MTPA and MTPV.

The scheme which describes the FOC is reported in Fig.1.4 :

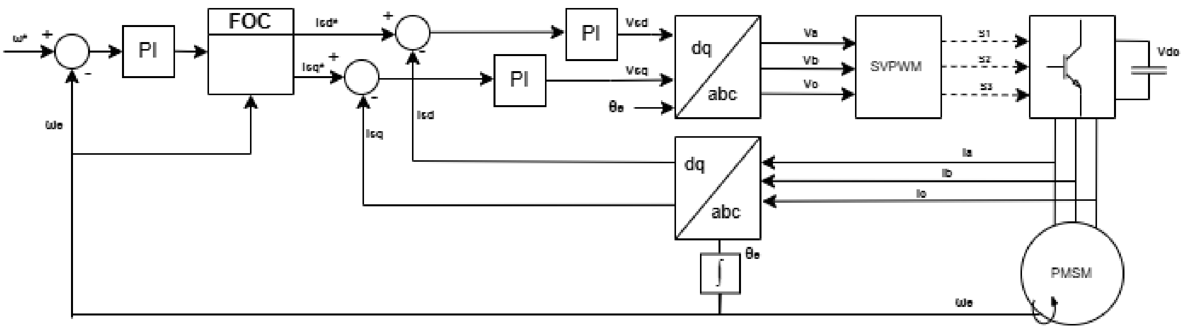

Figure 1.4: Speed control scheme with field oriented control.

Three proportional integral (PI) regulators have been used in the control system. These type of regulators are suitable for system characterized by first order liner differential equations, in order to improve the ability to follow the reference values and decouple the dynamic responses inside the system. The tuning must be performed by considering the nature of the quantity to dealing with: mechanical, electrical or electronic. For the system under study, equation (1.9a) for the speed PI-regulator, while equations (1.8a) and (1.8b) for current regulators. The algorithm used for the tuning is reported in appendix A. The overall dynamic performance of the system is limited by the inverter, the actuator. It's dynamics, related to the switching frequency f_{switch}, fixes the upper-limit in terms of bandwidth.

1.3.1. MTPA

The MTPA algorithm is used to maximize the electric torque of the machine for each value of the stator current. In d-q model of the machine, this means building a table starting from the stator current value I_s and define the corresponding pairs of currents values $(i^*_{sd}\ ;\ i^*_{sq})$.
In particular, by evaluating the equation (1.8c) introducing the stator current space vector $\underline{i_s} = I_s e^{-j\delta}$ as:

$$\begin{cases} i_{sd} = I_s\cos(\delta) & (1.10a) \\ i_{sq} = I_s\sin(\delta) & (1.10b) \end{cases}$$

where δ is the angular position of stator current space vector in d-q reference frame. Then by applying $\frac{\partial(\frac{Tel}{I_s})}{\partial\delta} = 0$ it's possible to define:

$$i^*_{sd} = \frac{-1 + \sqrt{1 + 8\frac{(L_d - L_q)^2}{\phi^2_M} I^2_s}}{\frac{4(L_d - L_q)}{\phi_M}} \tag{1.11}$$

$$i^*_{sq} = \sqrt{I^2_s - i^{*2}_{sd}}. \tag{1.12}$$

Therefore, spanning I_s from 0 [A] up to I_s^{max}, the corresponding maximum electrical torque T^*_{el} is computed according to equation (1.8c). This applies for speeds lower than the base speed, ω_{base} of the machine. Infact, at higher speed values, the the back-emf induced by the permanent magnets (proportional to the speed) must be reduced in order to avoid to overcome the maximum voltage of the machine, as explained in the following paragraph.

1.3.2. MTPV

The MTPV algorithm operates in the high speed region of the machine, computing the maximum torque achievable without overcome the voltage limit, defined by the maximum voltage V_s^{max}. In particular, in d-q reference frame, by neglecting the voltage drop on stator resistance, the voltage limit can be represented by ellipses as:

$$\frac{i^2_{sq}}{(\frac{V_s^{max}}{pp\omega_r L_q})^2} + \frac{(i_{sd} + \frac{\phi_M}{Ld})^2}{(\frac{V_s^{max}}{pp\omega_r L_d})^2} \leq 1 \tag{1.13}$$

with centre $(-\frac{\phi_M}{L_d}; 0)$, that reduce their area by increasing with speed. This behaviour implies that the MTPA algorithm can be followed until the voltage limit is respected, otherwise the pairs of currents that ensure voltage constraint and maximize the torque are computed as:

$$i^*_{sd} = \frac{-2L_d I_s \phi_M + \sqrt{4L^2_d I^2_s \phi^2_M - 4(L^2_d - L^2_q) I^2_s [\phi^2_M + L^2_q I^2_s - (\frac{V_s^{max}}{pp\omega_r})^2]}}{2(L^2_d - L^2_q) I_s} \tag{1.14}$$

$$i^*_{sq} = \sqrt{I^2_s - i^{*2}_{sd}}. \tag{1.15}$$

In the following, Fig.1.5 shows the result on the implementation of MTPA and MTPV algorithms in the I_d-I_q plane, considering maximum current constraint and three examples of voltage ellipses:

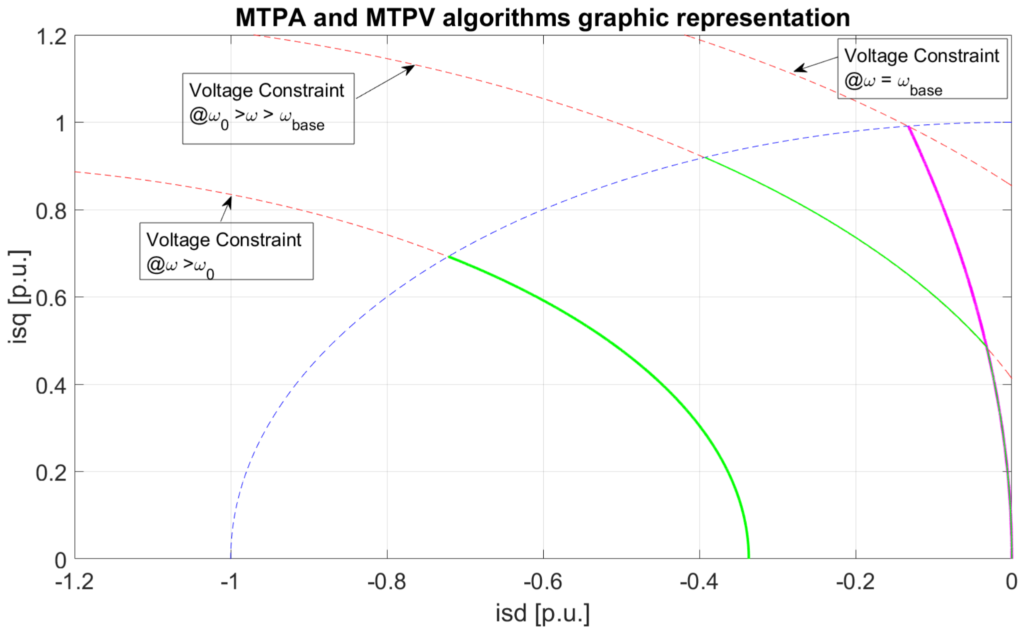

Figure 1.5: MTPA and MTPV algorithm graphic representation.

The purple curve represents the optimized current pairs which satisfy the MTPA algorithm. Such control can be followed until the voltage constraint is satisfied. Otherwise, the green curves applies, representing the pairs of currents that result from MTPV.
Tab.1.1 summarizes how the control is implemented according to the speed of the PMSM:

Speed range	Control Applied
$\omega < \omega_{base}$	MTPA
$\omega_0 > \omega > \omega_{base}$	1° MTPA / 2° MTPV
$\omega > \omega_0$	MTPV

Table 1.1: Control algorithm implementation depending on the speed.

The complete Simulink schematic and algorithm implementation of FOC is reported in appendix A.

1.4. Ideal simulation results

After the description of the mathematical d-q model of the anisotropic PMSM and the implementation of the field oriented control, the following section reports the main results obtained for a specific machine. The idea is to validate the correct implementation of the control system, focusing on the dynamic of the speed and on the coherent optimization control strategies application.

The simulation has been performed by considering the electrical and mechanical parameters present in the Tab.1.2 and Tab.1.3 respectively, coming from the machine data sheet:

Parameter	Value
pp	20
R_s	20 $m\Omega$
L_d	195 μH
L_q	233 μH
ϕ_M	0.09 Wb
I_s^{nom}	230 A
V_{s-LL}^{nom}	440 V
T_{el}^{nom}	891 Nm

Table 1.2: Electrical parameters of the anisotropic PMSM.

Parameter	Value
B_0	0.0003 Nms
B_{load}	0.25 Nms
J_{eq}	11.3 Kgm^2

Table 1.3: Mechanical parameters of the anisotropic PMSM.

According to the control scheme represented by Fig.1.4, at this stage the inverter is bypassed, it will be introduced in the followin chapters. Therefore the reference voltage values resulting from FOC in d-q coordinates are directly applied to the d-q model of the PMSM. The cut-off frequency used to tune the current PI regulators is $f_H^{current} = 1$ kHz, while for the speed PI regulator $f_H^{speed} = 1$ Hz has been selected.

The resulting coefficients are reported in :

Parameter	Speed	Current d/q
K_p	60.350	1.069 /1.272
K_i	58.11	1026.7 /1221.9

Table 1.4: PI-regulators gains.

The test consists in a step of reference mechanical speed up to the base speed of the machine. The measured mechanical speed is reported in Fig.1.6 and the error with respect to the reference value in Fig.1.7 below:

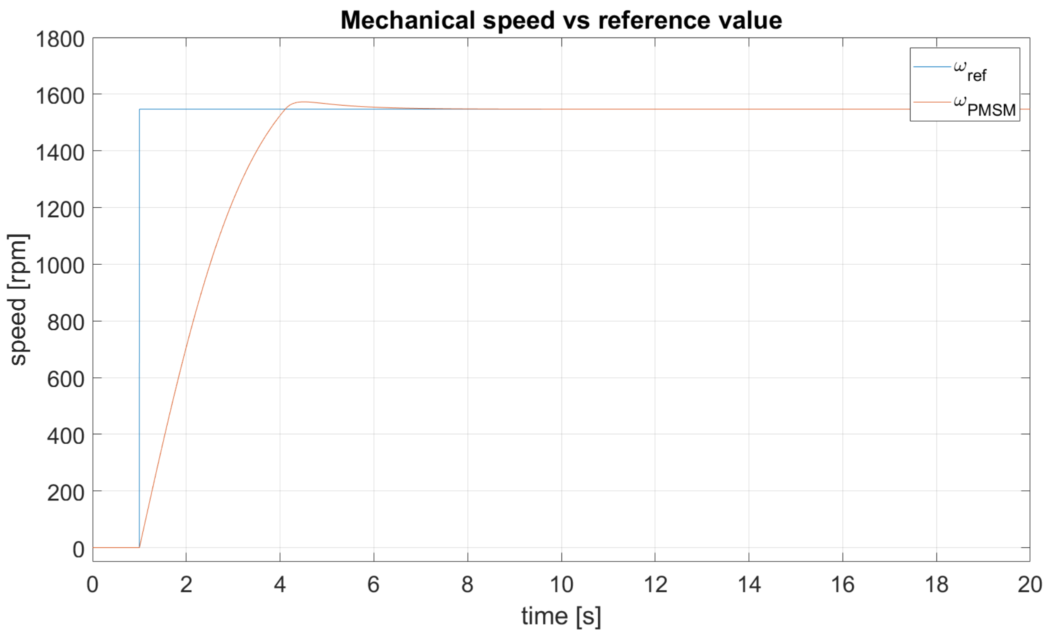

Figure 1.6: PMSM mechanical speed measures with respect the reference signal.

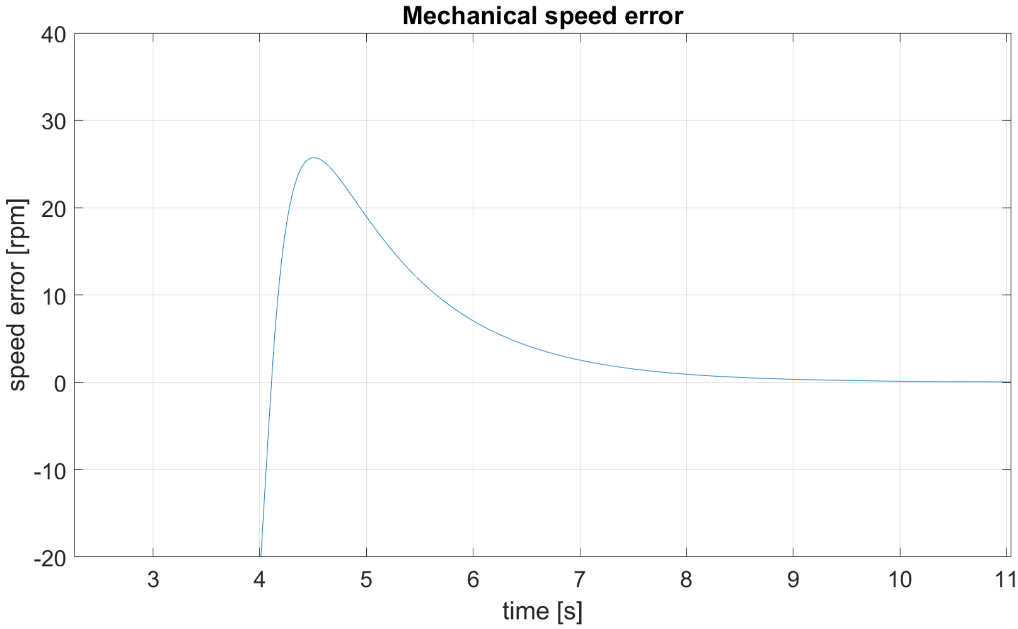

Figure 1.7: Mechanical speed error.

The mechanical speed has a slight overshoot immediately after the electromechanical transient, then the value stabilizes with the reference signal. The overshoot is mainly due to the integrator in the speed PI-controller, that requires time to discharge since anti-windup mechanism is not included.

The electrical torque and currents are reported in Fig.1.8 and Fig.1.9.

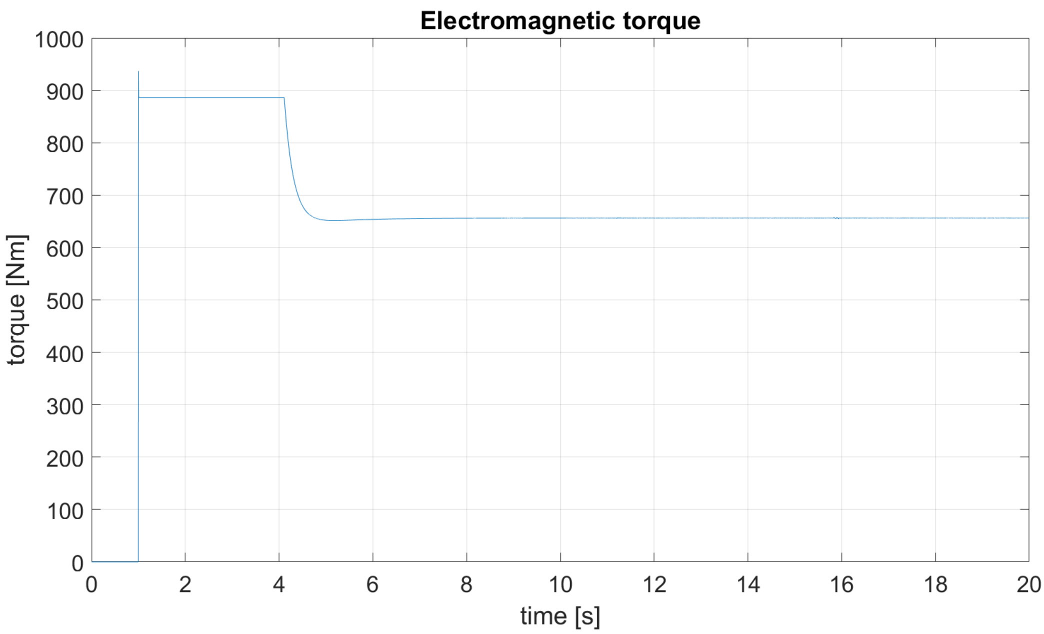

Figure 1.8: Electrical torque measure.

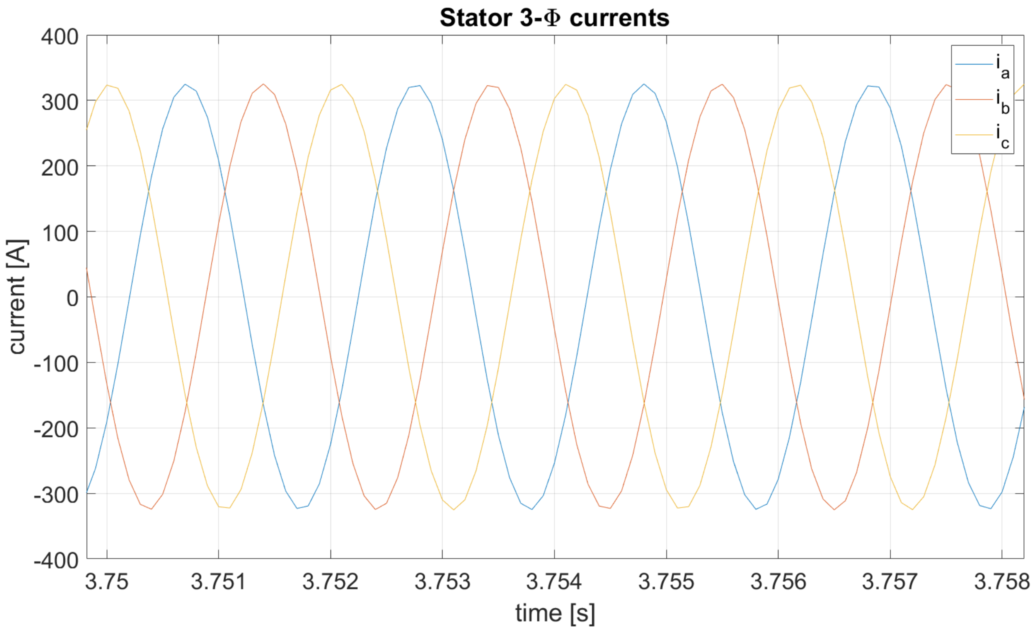

Figure 1.9: Three-phases stator current measures when maximum torque applied.

While here below, Fig.1.10 shows the d-q stator currents resulting from the PMSM model:

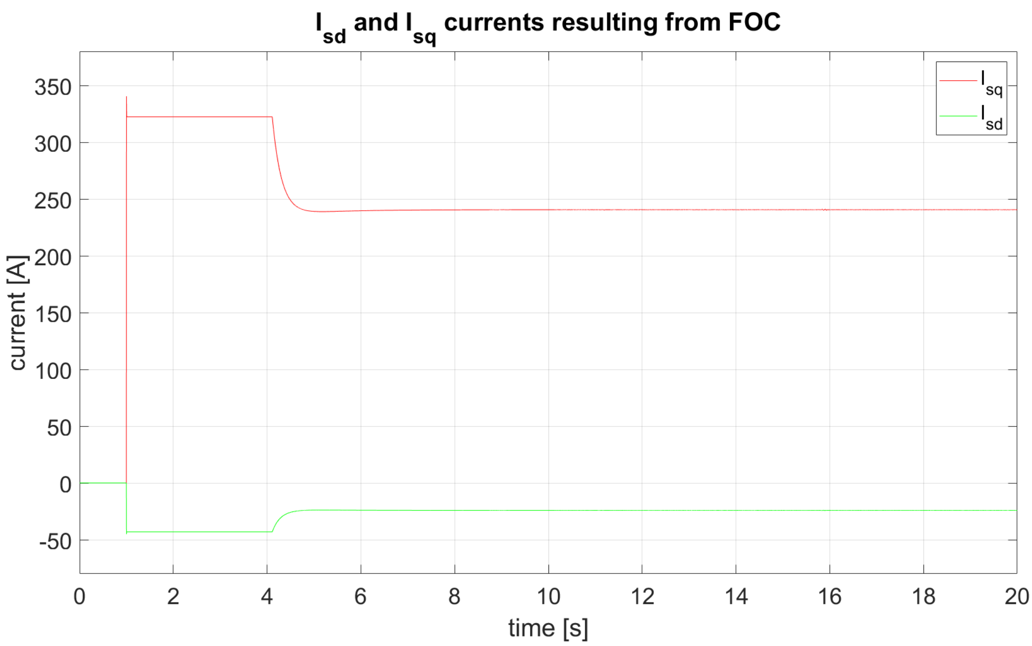

Figure 1.10: d-q stator current dynamics according to speed.

Notice that during the maximum torque application, immediately after the change of speed reference, the three-phases currents don't overcome the limit value. Moreover, the initial values of d-q stator currents during the speed transient correspond exactly to the intersection between MTPA curve and maximum current limit in Fig.1.5, properly scaled by $\sqrt{2}I_s^{nom}$. Their value then stabilize to the operating point in which the electrical torque is equal to the load torque.

2 | Sensor-less Control Strategy: MRAS

In this chapter, sensor-less control strategies are introduced, providing initially a brief an overview the existing techniques. Then the focus is moved towards the model reference adaptive system, studying its application to the anisotropic PMSM and reporting the main results from simulations performed to the machine model introduced in Section 1.4.

2.1. Overview of Sensors-less Control Strategies

In the PMSM speed control system the measurement of rotor position represent a crucial step. Generally, it's detected through mechanical sensors such as photoelectric encoder or rotary transformer . These mechanical sensors can easily and quickly obtain rotor position information but in many situations, they have some limitations such as:

1. **Higher dimensions and rotor inertia**: the introduction of mechanical sensors increases volume, weight and rotational inertia, negatively effecting the control performance;

2. **Connection to the control system circuit required**: such aspect increased the complexity of the control system;

3. **Higher susceptibility to interferences and reliability reduction**: including mechanical sensors and related wires can introduce noise in the system, affecting its performances.

Therefore, developing a low-cost and high-performance speed sensor-less control method has become a hot topic in the field of motor control research.

As reported in [7] and showed in Fig.2.1 , there are two main categories of sensor-less control schemes depending on the speed range considered: model-based methods for high-speed range and saliency-based methods for low speed range .

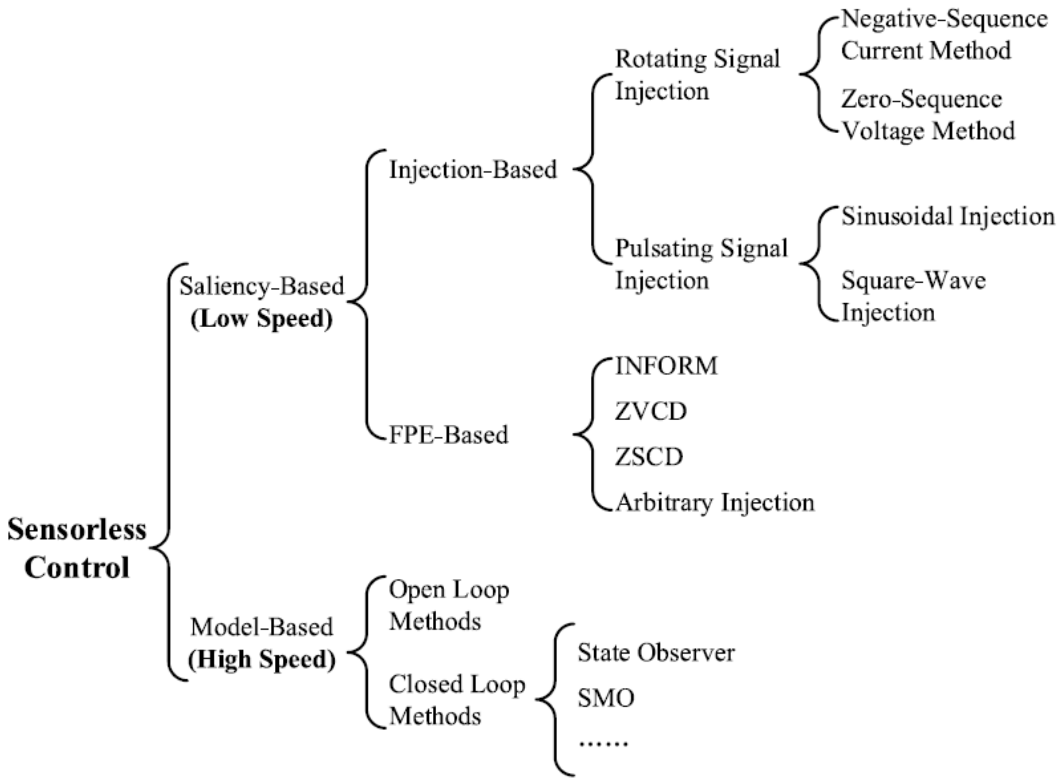

Figure 2.1: Sensorless control methods classifications [7].

Through model-based control strategies, the speed estimation is derived by evaluating physical quantities such as back electromotive force (EMF) or flux associated with the fundamental excitation. These approaches are subdivided in open loop and closed loop methods. The difference is that in the latter a feedback corrective term representing the error between estimated and measured quantities is introduced to improve the performances. Anyway, both are focused on the motor model so very sensitive to parameters uncertainty, but also on the operating state of the motor. Indeed, in the low-speed range, the back-emf is low and therefore characterized by small signal-to-noise-ratio (also affected by inverter nonlinearity etc.)

In order to cover the low-to-zero speed range, the saliency tracking-based method has been developed. They include methods based on the injection signals and approaches that exploit the fundamental PWM, known as FPE-based methods. The basic principle followed by the saliency-based techniques is to estimate the rotor position relying on the analysis of the motor response to a know high-frequency electrical excitation, exploiting the variation of the stator impedance as a function of the rotor electrical angle caused by magnetic saliency. In the inject-based strategies, an high frequency (HF) signal is added to the control voltage, while in FPE-based this is automatically provided by PWM's harmonics.

Among the model-based sensor-less techniques, the model reference adaptive system represents a widely adopted closed-loop approach for rotor speed and position estimation in the medium-to-high speed range. In the following section, the MRAS principle is introduced and its structure is described, highlighting the reference and adaptive models as well as the associated adaptation mechanism. The effectiveness of the proposed MRAS-based scheme is then evaluated through simulation results.

2.2. Model Reference Adaptive System (MRAS)

The model reference adaptive system control strategy represents one of the main sensor-less control approaches more studied in the literature and applied, thanks to solid theoretical base and simple implementation.
The basic principle of the MRAS control system consists in the comparison between two models of the system: the **reference model** and the **adaptive model**. The first is independent of the unknown variable to control, while the second includes clearly such parameter. Both the models receive the same inputs, generally the stator voltages or currents, and their outputs represent the state variable of the system. Depending on the structure of the system implemented, these could be: magnetic fluxes, back-EMF or currents. The error signal is defined by comparing the outputs of the models: such error is used by an **adaptive mechanism** to iteratively update the value of the unknown parameter. The MRAS goal is to guarantee an accurate estimation of the variable by asymptotically driving to zero the state variable error.
The general algorithm scheme of the MRAS control system is reported in Fig.2.2:

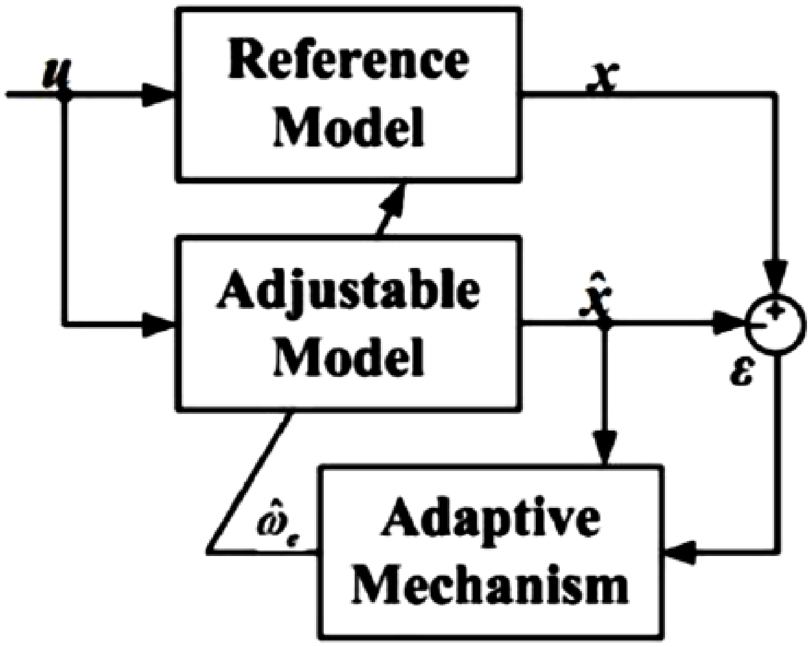

Figure 2.2: MRAS general schematic based on rotor speed estimation.[8]

The main advantage of MRAS algorithm with respect to other sensor-less controls is its simple structure, since it's implemented with electrical quantities already available in FOC. Moreover, it requires low computational effort compared to more complex algorithms as Kalman filter or AI-based control systems. The main limitations instead, derive from the parameters estimation of the machine. The stator resistance or inductances variation because of temperature or parameter uncertainties, can significantly affect the estimation resulting from the MRAS. Another drawback is the poor performance considering low-speed operational regions of the machine, due to the lack of useful information in the electrical quantities (low signal-to-noise ratio).

In the following section, the MRAS sensor-less control strategy is applied to the anisotropic PMSM, developed according to theory proposed in [9].

2.3. Reference and Adaptive Models

According to MRAS control algorithm, the reference and adaptive model must be selected. The KVLs of the d-q model of the anisotropic PMSM reported in equations (1.8a) and (1.8b) are the base of the study.
It's possible to rewrite them as:

$$\begin{cases} \dfrac{di_{sd}}{dt} = -\dfrac{R_s}{L_d} i_{sd} + pp\omega_r \dfrac{L_q}{L_d} i_{sq} + \dfrac{v_{sd}}{L_d} & (2.1a) \\ \dfrac{di_{sq}}{dt} = pp\omega_r \dfrac{L_d}{L_q} i_{sd} - \dfrac{R_s}{L_q} i_{sq} + \dfrac{v_{sq}}{L_q} - pp\omega_r \dfrac{\phi_M}{L_q} & (2.1b) \end{cases}$$

Introducing in the equations the electrical speed $\omega_e = pp\omega_r$, they can be also represented in matrix form as:

$$\frac{d}{dt}\begin{bmatrix} i_{sd} + \frac{\phi_M}{L_d} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_e \frac{L_q}{L_d} \\ -\omega_e \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_{sd} + \frac{\phi_M}{L_d} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} v_{sd} + \frac{R_s\phi_M}{L_d} \\ v_{sq} \end{bmatrix}. \tag{2.2}$$

Therefore, the system can be described by equation (2.2) which can be simplified into an equivalent system introducing the following changes of variables:

$$\begin{cases} i'_{sd} = i_{sd} + \dfrac{\phi_M}{L_d} & (2.3a) \\ i'_{sq} = i_{sq} & (2.3b) \\ v'_{sd} = v_{sd} + \dfrac{R_s\phi_M}{L_d} & (2.3c) \\ v'_{sq} = v_{sq} & (2.3d) \end{cases}$$

Finally, the equivalent d-q model of the anisotropic PMSM is described as:

$$\frac{d}{dt}\begin{bmatrix} i'_{sd} \\ i'_{sq} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_e \frac{L_q}{L_d} \\ -\omega_e \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i'_{sd} \\ i'_{sq} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} v'_{sd} \\ v'_{sq} \end{bmatrix}. \tag{2.4}$$

The estimated model is represented exactly by equation (2.4), but containing the estimated electrical speed $\hat{\omega}_e$ and the estimated d-q currents $\hat{i'_{sd}}$ and $\hat{i'_{sq}}$, resulting in:

$$\frac{d}{dt}\begin{bmatrix} \hat{i'_{sd}} \\ \hat{i'_{sq}} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \hat{\omega}_e \frac{L_q}{L_d} \\ -\hat{\omega}_e \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} \hat{i'_{sd}} \\ \hat{i'_{sq}} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} v'_{sd} \\ v'_{sq} \end{bmatrix}. \tag{2.5}$$

This thesis work consider the reference model for the MRAS sensor-less control as the d-q current model of the machine, represented in Fig.1.3 and will adopt as adjustable model the equivalent system described by equation (2.5).
In Fig.2.3, the rotor positions identified by the two systems, $d-q$ for the real model and $\gamma-\beta$ resulting from MRAS, are represented:

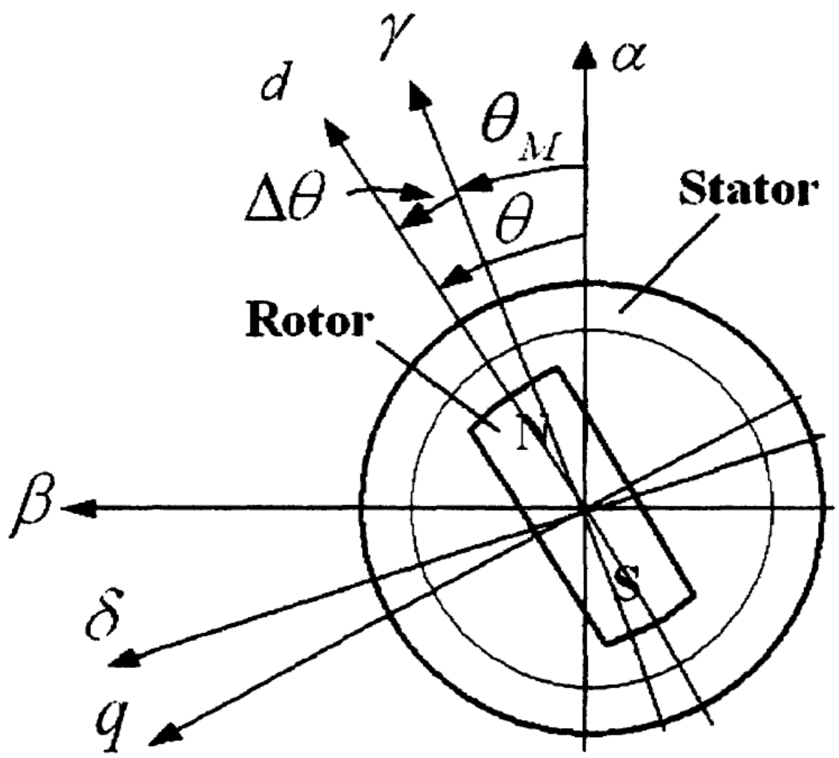

Figure 2.3: Rotor positions for real motor and MRAS algorithm [10]

Because of the errors of the estimated speed and back EMF, the currents of the two model won't be the same. The error of the currents is then used to compute the speed and position estimation. The discrepancy between the two systems will be drive to zero according to a proper adaptive mechanism described in the following section.

2.3.1. Adaptive Law

The adaptive law of the MRAS has the goal of defining the correct equation for the speed estimation. The most critical point is to get a suitable adaptive mechanism and ensure the stability of the system. In order to define the adaptive law, both the reference and adjustable systems are expressed in vector matrix form:

$$\frac{d}{dt}I'_s = AI'_s + BV'_s \tag{2.6}$$

$$\text{where: } A = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_e\frac{L_q}{L_d} \\ -\omega_e\frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix}, I'_s = \begin{bmatrix} i'_{sd} \\ i'_{sq} \end{bmatrix}, V'_s = \begin{bmatrix} v'_{sd} \\ v'_{sq} \end{bmatrix}$$

$$\frac{d}{dt}\hat{I}'_s = \hat{A}\hat{I}'_s + BV'_s \tag{2.7}$$

$$\text{where: } \hat{A} = \begin{bmatrix} -\frac{R_s}{L_d} & \hat{\omega}_e\frac{L_q}{L_d} \\ -\hat{\omega}_e\frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix}, \hat{I}'_s = \begin{bmatrix} \hat{i'}_{sd} \\ \hat{i'}_{sq} \end{bmatrix}.$$

By subtracting (2.7) from (2.6), introducing the error of state variables $\varepsilon_d = i'_{sd} - \hat{i'}_{sd}$ and $\varepsilon_q = i'_{sq} - \hat{i'}_{sq}$, the equation of the generalized error becomes:

$$\frac{d}{dt}e = Ae - W \tag{2.8}$$

where: $e = \begin{bmatrix} \varepsilon_d & \varepsilon_q \end{bmatrix}^T$, $W = (\hat{\omega}_e - \omega_e)B\hat{I}'_s$.

The system described by equation (2.8) is a standard non-linear time-varying feedback system. Indeed, ω_e in matrix A can be considered as time invariant parameter thanks to the fact that the mechanical time constant is much greater than the electrical time constant. The general schematic of the system is showed in Fig. 2.4:

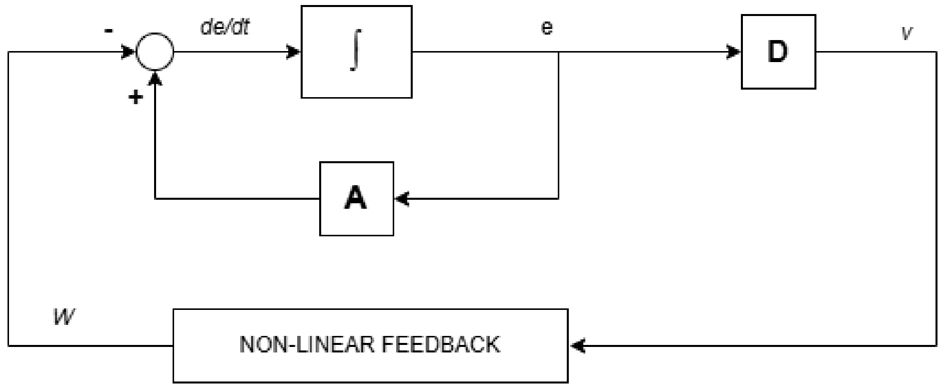

Figure 2.4: Scheme of the general non-linear feedback.

Basically, it's possible to adapt equation (2.8) as:

$$\begin{cases} \frac{d}{dt}e = Ae - W & (2.9a) \\ \nu = De & (2.9b) \end{cases}$$

where $D = I$, the identity matrix, therefore $\nu = e$.

Then, Popov's hyperstability theory can be used to model the adaptive mechanism.
To address the stability of the non-linear feedback, the following inequality must be satisfied:

$$\eta(0,t) = \int_0^t \nu^T W dt \geq -\gamma^2, \ \forall t \geq 0 \tag{2.10}$$

where γ is a limited positive number.

By introducing the expression of $\nu^T W$ according to the control system, an important constraint for the MRAS follows:

$$\eta(0,t) = \int_0^t \nu^T W dt = \int_0^t (\hat{\omega}_e - \omega_e)(\frac{L_q}{L_d}\hat{i'}_{sq}\varepsilon_d - \frac{L_d}{L_q}\hat{i'}_{sd}\varepsilon_q)dt \geq \gamma^2 \tag{2.11}$$

this constraint is always satisfied, since the adaptive signal (right part inside the integral) can be expressed as combination of terms of the form $f(t)\dot{f(t)}$ such that:

$$\int_0^t f(t)\dot{f(t)}dt = \frac{1}{2}[f(t)^2 - f(0)^2] > \frac{1}{2}f(0)^2. \tag{2.12}$$

Moreover, a standard PI regulator can be implemented as adaptive rate regulator. Therefore, considering as initial speed $\omega_e = 0$, the speed estimation is performed according to:

$$\hat{\omega}_e = (K_p + \frac{K_i}{s})[\frac{L_q}{L_d}\hat{i'_{sq}}(i'_{sd} - \hat{i'_{sd}}) - \frac{L_d}{L_q}\hat{i'_{sd}}(i'_{sq} - \hat{i'_{sq}})] \tag{2.13}$$

and, equivalently, in the traditional d-q reference frame by introducing the equations (2.3):

$$\hat{\omega}_e = (K_p + \frac{K_i}{s})[\frac{L_q}{L_d}\hat{i_{sq}}i_{sd} - \frac{L_d}{L_q}\hat{i_{sd}}i_{sq} - \frac{\phi_M}{L_q}(\hat{i_{sq}} - i_{sq}) + (\frac{L_d}{L_q} - \frac{L_q}{L_d})\hat{i_{sd}}\hat{i_{sq}}]. \tag{2.14}$$

Then, the estimated electrical position is computed as:

$$\hat{\theta}_e = \theta_e(0) + \int_0^t \hat{\omega}_e dt. \tag{2.15}$$

The control system that includes the MRAS sensorless control proposed is reported in Fig.2.5

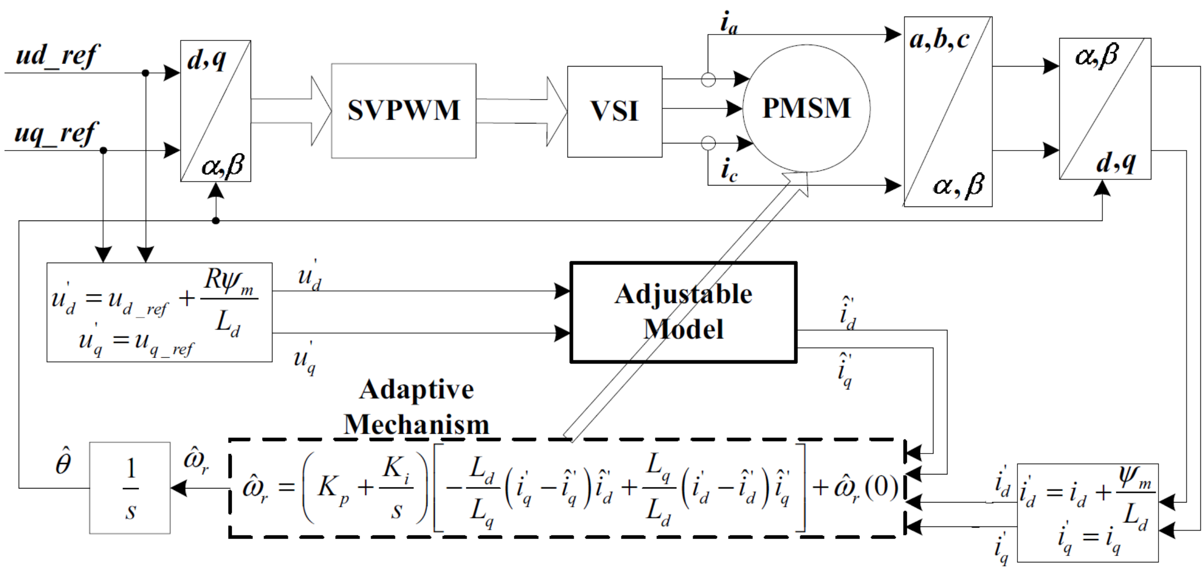

Figure 2.5: Control scheme of anisotropic PMSM including MRAS algorithm [9]

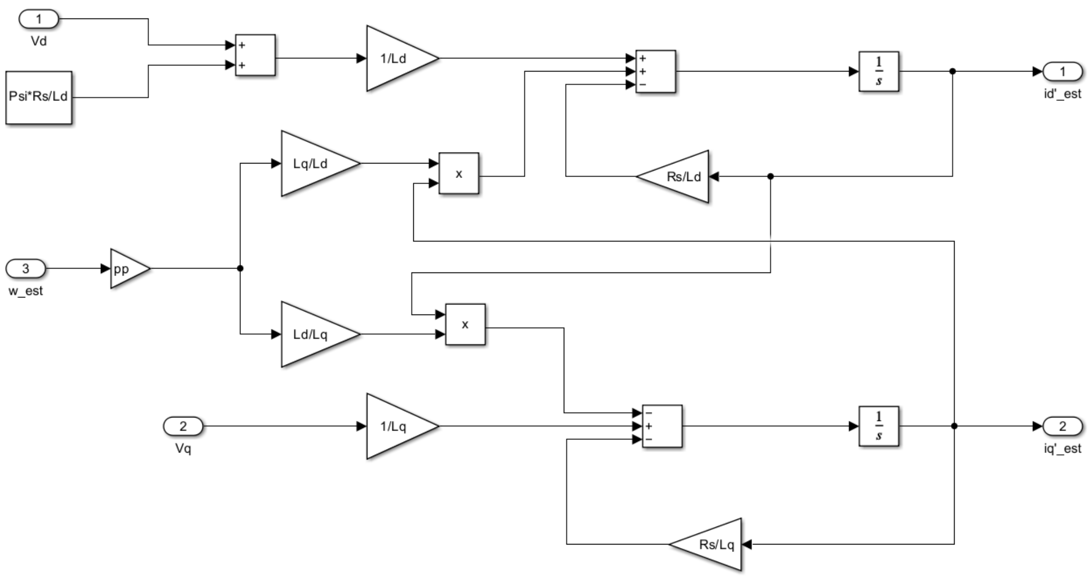

Figure 2.6: Simulink implementation of MRAS adjustable model.

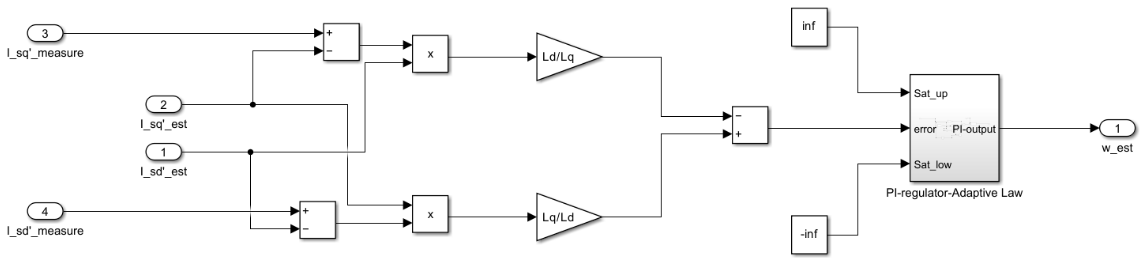

Figure 2.7: Simulink implementation of MRAS adaptive law.

2.4. Ideal results with MRAS

This section reports the simulation results of the MRAS algorithm proposed, which has been implemented in the Simulink model of the anisotropic PMSM with FOC already described in Chapter 1. The aim is to verify the effectiveness of the sensor-less control, studying its behaviour and highlighting the main problems encountered.
In order to set up the simulation, the tuning of the PI- regulator that controls the adaptive law for MRAS must be performed. Unlike standard PI loops where analytical relationships can sometimes guide the selection of proportional and integral gains, the MRAS structure does not provide closed-form solutions for optimal tuning. Therefore, try and error procedure has been adopted, typically starting by an initial guess resulting from stable simulations running the motor with real speed feedback. The two parameters used for the MRAS, speed and currents PI-regulator are reported below:

Parameter	**Speed**	**Currents d/q**	**MRAS**
K_p	60.350	1.069 /1.272	0.5
K_i	58.11	1026.7 /1221.9	5

Table 2.1: PI regulators setting specifications.

Moreover, in the speed and currents PI-regulators, anti-windup algorithms have been employed to avoid overload of integrators and so possible instabilities events in the system.

The simulation has been performed by ode23t solver in Simulink, by using a maximum step size of 10^{-5}; the main results are reported in the following pages:

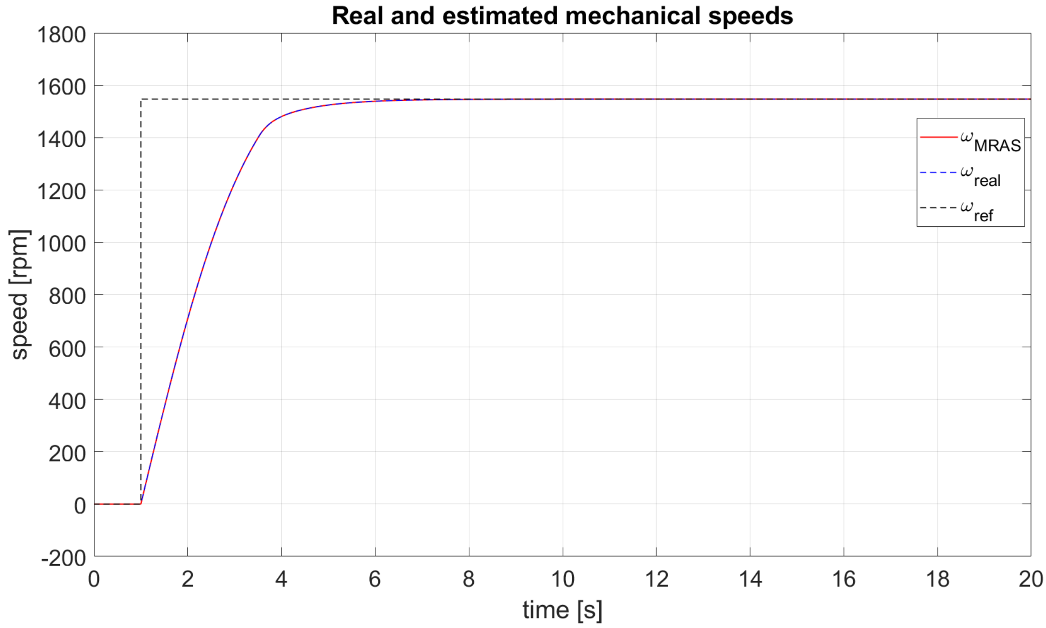

Figure 2.8: Mechanical real and estimated speeds.

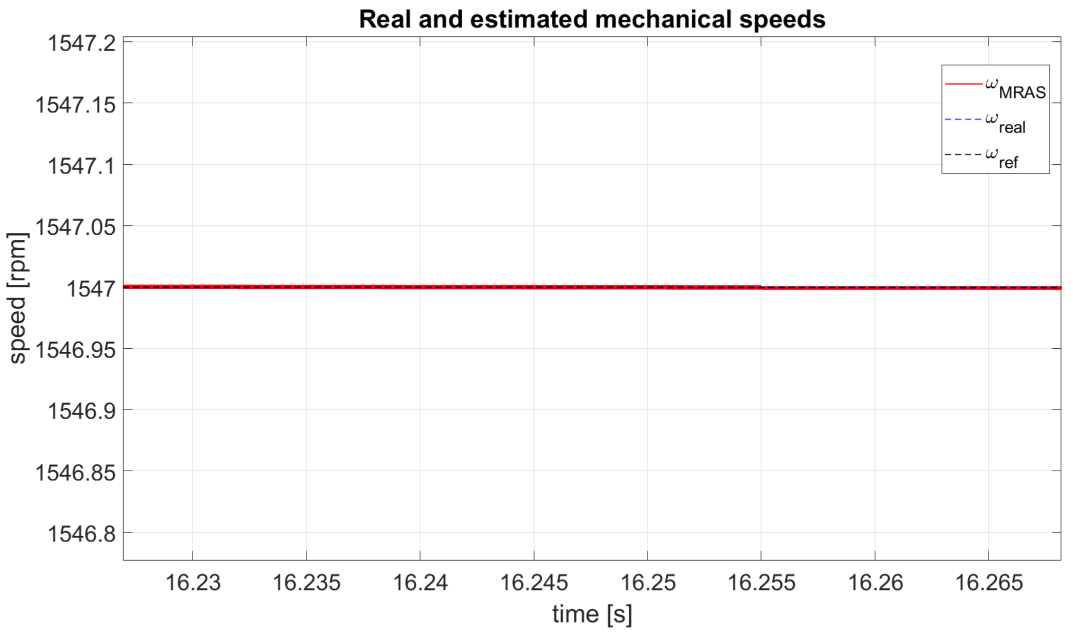

Figure 2.9: Zoom of real and estimated mechanical speeds at ω_{base}.

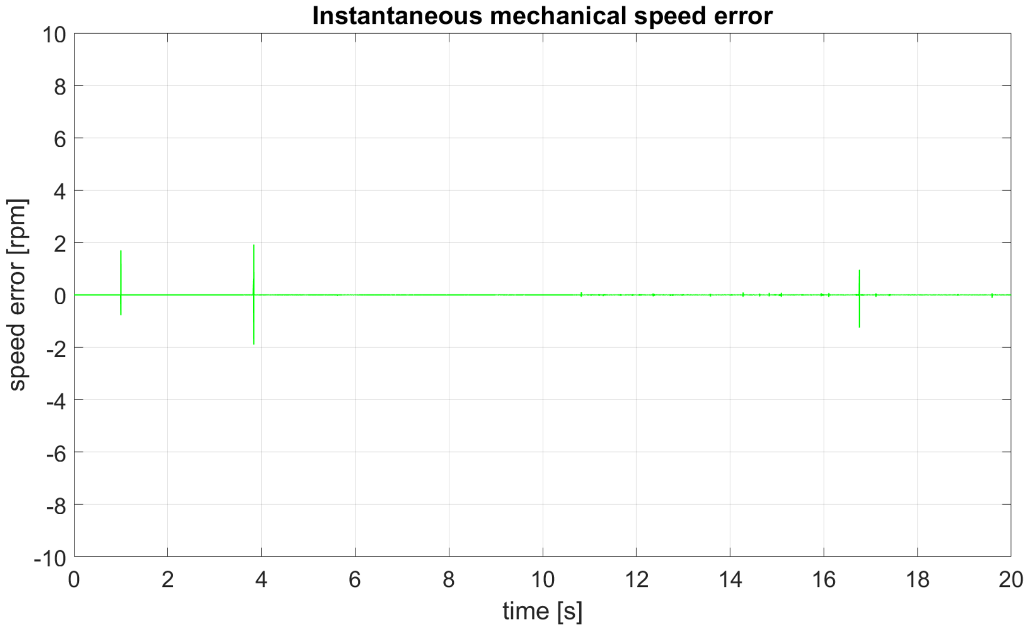

Figure 2.10: Instantaneous mechanical speed error during the simulation.

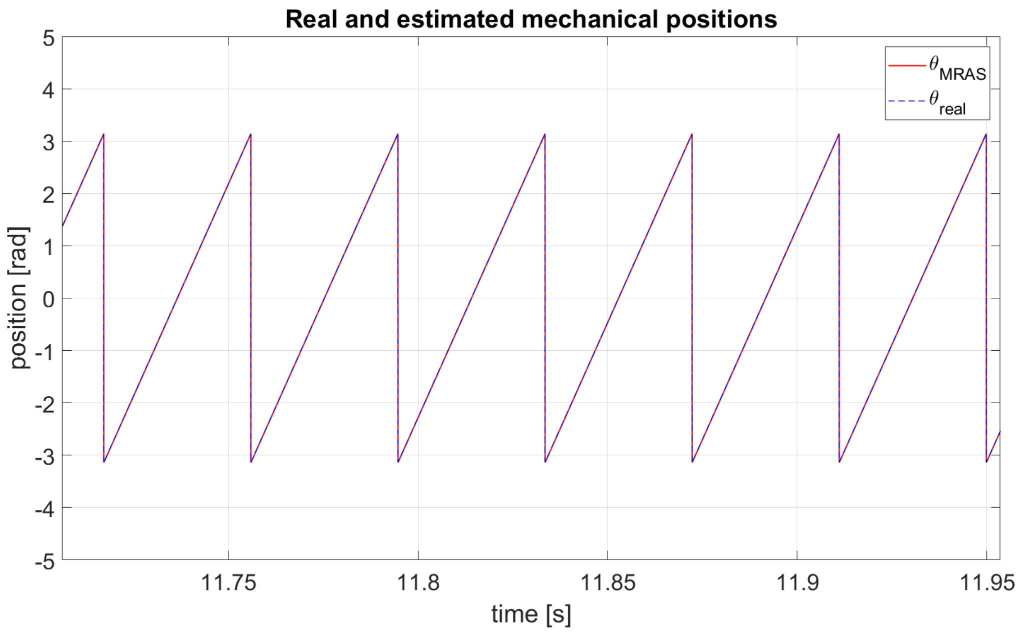

Figure 2.11: Mechanical positions at steady state.

Figure 2.12: Instantaneous mechanical position error during the simulation.

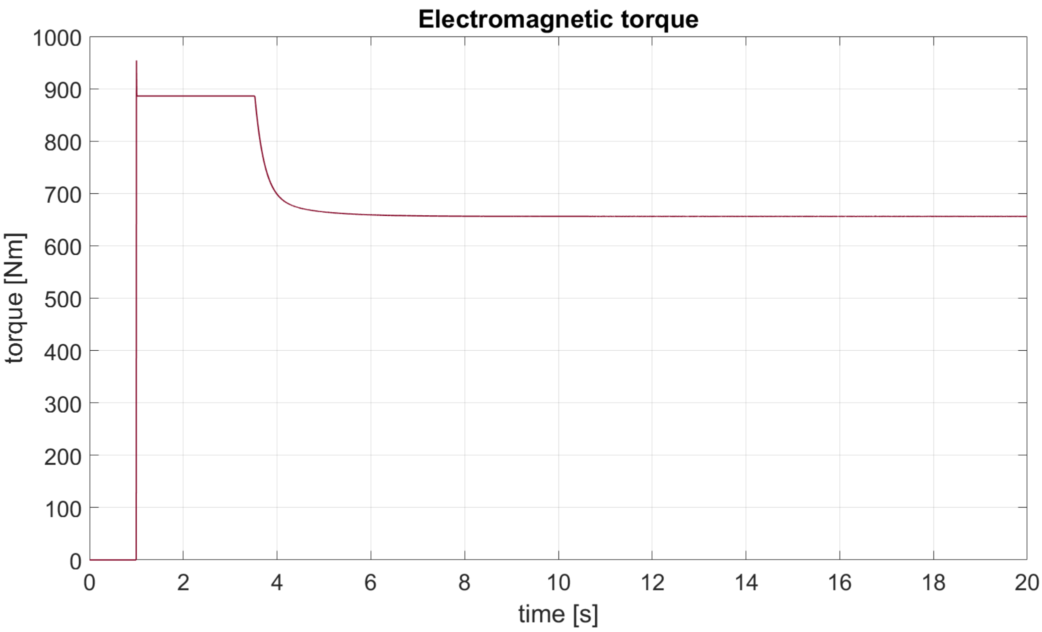

Figure 2.13: Electromagnetic torque of the PMSM.

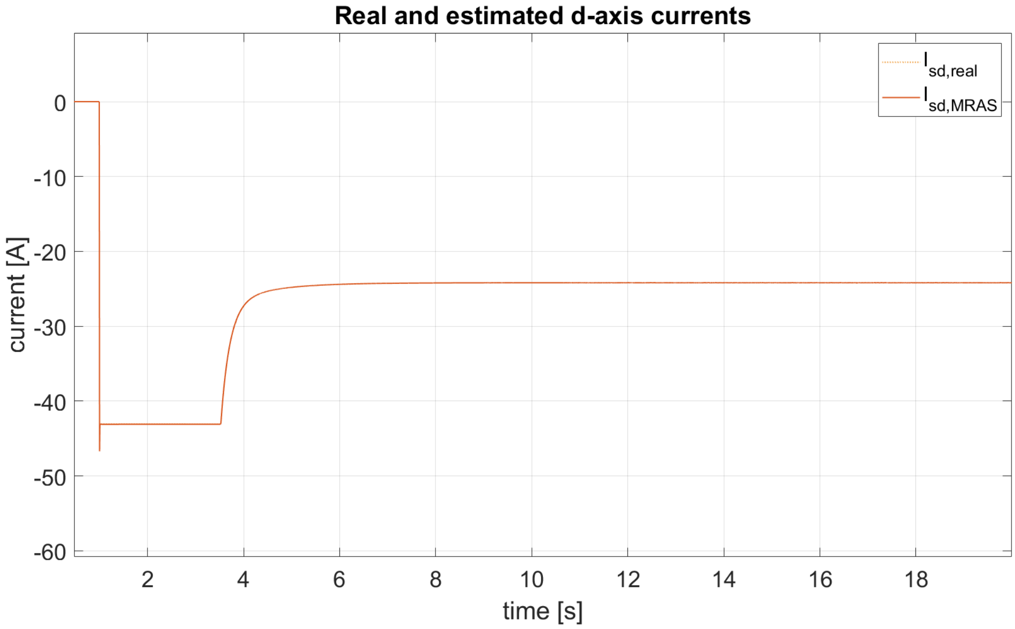

Figure 2.14: Real and estimated I_{sd} currents.

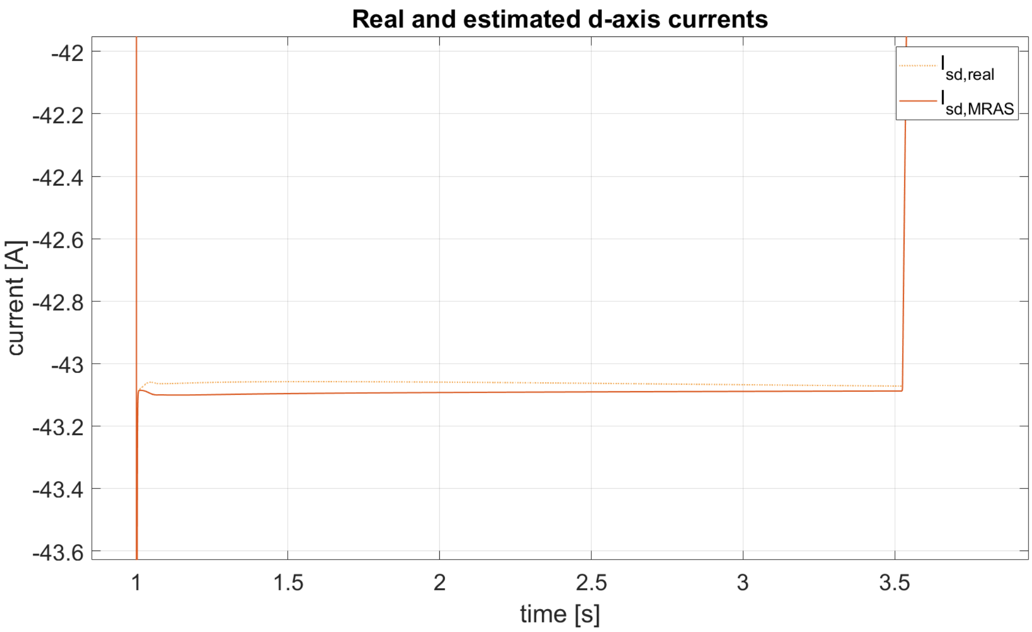

Figure 2.15: Zoom of I_{sd} currents mismatch during speed step transient.

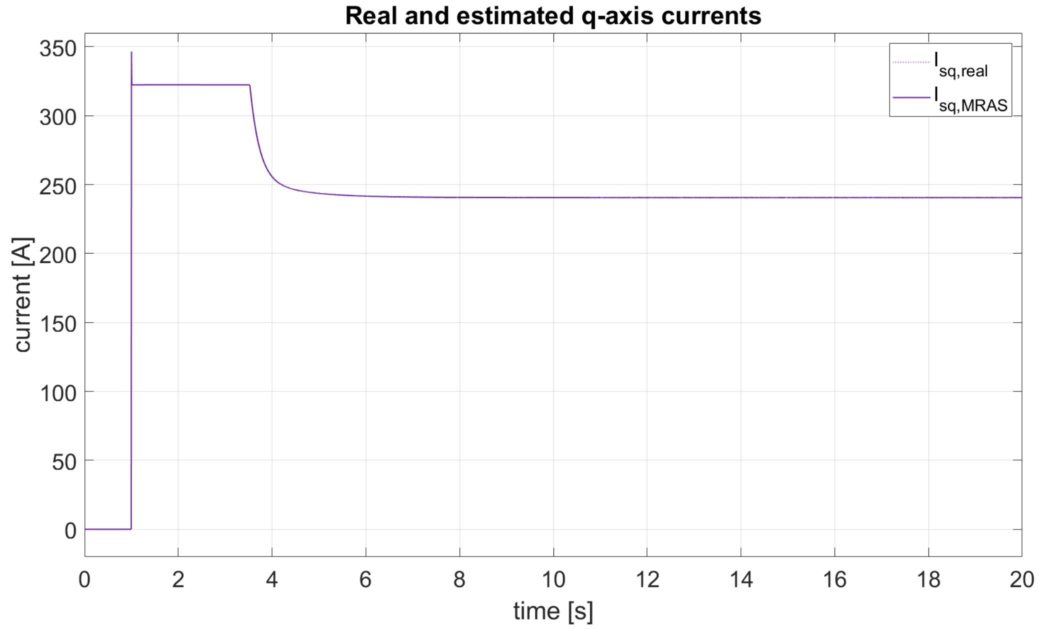

Figure 2.16: Real and estimated I_{sq} currents.

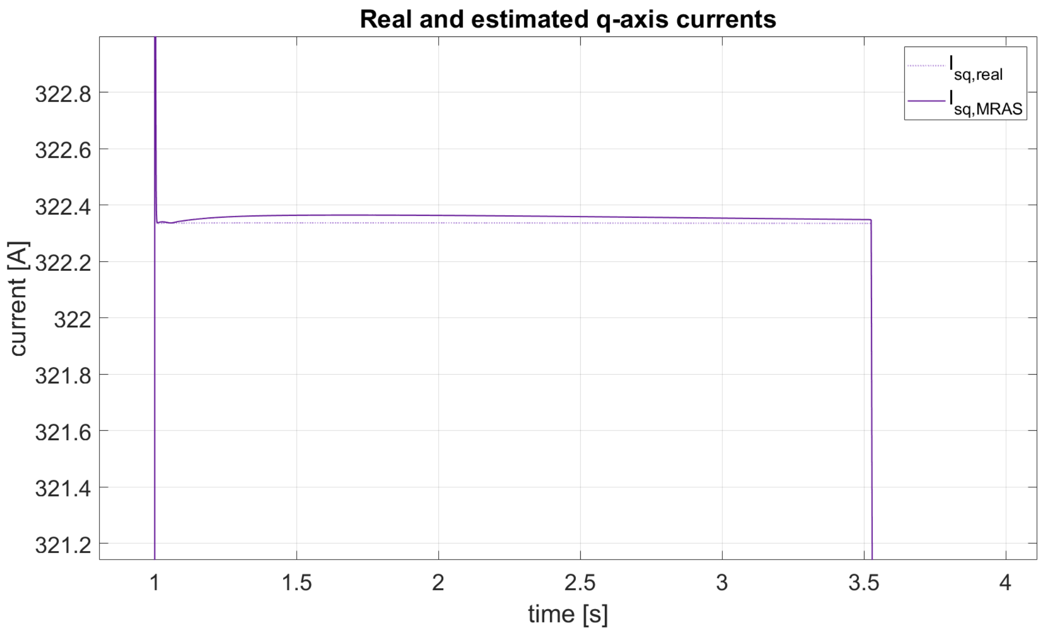

Figure 2.17: Zoom of I_{sq} currents mismatch during speed step transient.

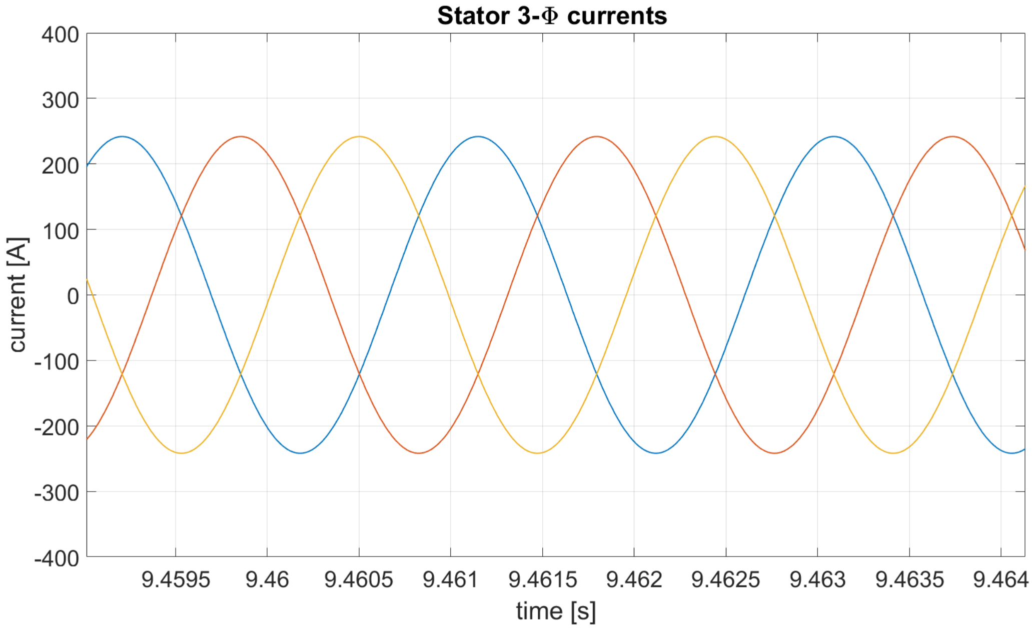

Figure 2.18: Zoom of the three-phase stator currents.

From Fig.2.8, it's possible to see that the speed estimation provided by the MRAS algorithm follows and almost perfectly match the actual value of the system. The estimated speed accurately tracks the actual rotor speed during the test, confirming the effectiveness of the algorithm implementation. As reported in Fig.2.10, the main speed errors are concentrated during the transient to reach the reference speed. Such operating region is characterized by rapid dynamic variations, during which the adaptation mechanism requires time to converge and so low instabilities occur. There are also steady state spikes that are likely attributable to the simulation process. In the steady state conditions (Fig.2.9), both the actual speed and the MRAS estimation perfectly match, indicating good performances of the estimator.

Similar, as it's showed in Fig.2.11 and Fig.2.12 (to notice the scale of y-axis in the plot), the mechanical position estimation matches almost perfectly the real one, ensuring a good synchronization between the two d-q reference systems of the model.

About the real and estimated d,q currents, the estimator closely follows the real values. Anyway, it's possible to notice in Fig.2.15 and Fig.2.17 that during the transient, there is a mismatch between the estimate and actual values: this behaviour is typical for the MRAS estimators since the adaptive law act directly on the speed, that perfectly match the actual value in this phase. The observed mismatch of the currents can be attributed to the difference in dynamic bandwidth between the FOC current PI controllers and the MRAS adaptation mechanism. The current controllers react very quickly to the speed step, generating fast current components, while the MRAS estimator is designed with a slower dynamic to ensure stability. As a result, the estimated currents cannot follow these rapid variations instantaneously, even though the estimated speed converges accurately. The important aspect is that the error generated by the adaptive law is null, ensuring the correct speed estimation convergence during the whole simulation.

3 | Discrete time implementation and simulations of MRAS

In this chapter, a more realistic system has been implemented. The aim is to develop the discretization of the anisotropic PMSM model, including the MRAS control algorithm implemented, and verify behaviour of the overall system. This step is crucial to understand the effectiveness of the sensor-less control, understanding if its implementation on real-time control board based on microcontroller could be achieved. Moreover, the space vector pulse width modulation (SV-PWM) is described and implemented on Simulink framework in order to define the voltage and frequency supply for the motor. The results of the simulations performed are reported in the last section.

3.1. Space Vector Pulse Width Modulation (SV-PWM)

The objective of this section is to introduce in the model an inverter and its control technique implemented, in order to represent the AC voltage source applied to the stator windings of the motor. A voltage source inverter (VSI) has been considered in the system, which is also the type of conversion system mostly used in electrical drives. It's preferred to a current source inverter because the desired voltage waveform is easily controlled in amplitude and period and since the load is represented by the stator windings of a PMSM (basically ohmic-inductive load), they indirectly act a low pass filter for the current waveform, therefore the harmonic content is reduced and so the resulting torque ripple. In the following is represented the classic two-level three-phases inverter, which has been selected as inverter for the model :

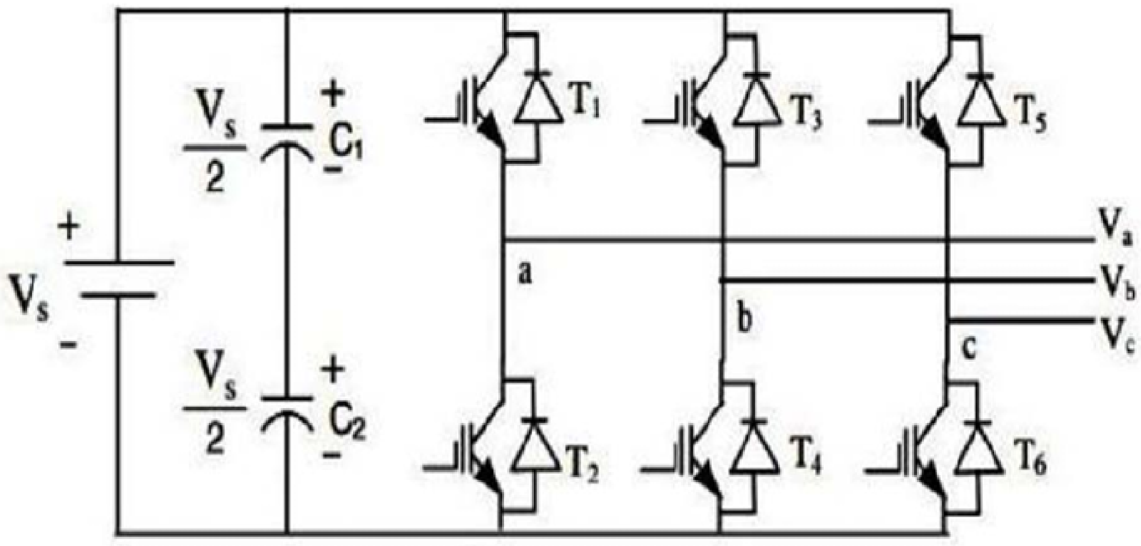

Figure 3.1: Schematic of two-level three phase voltage source inverter [11].

These inverters use capacitors in the DC–link to store the electrical energy. Three power legs are in charge of the conversion process, each of them includes two pairs of switch (typically IGBTs or MOSFETs) and freewheeling diode.

The combination of the commutation status of the six switches present in the circuit generates the three phase triplets from the direct voltage source.

The overall performance of the VSI, does not depend solely on the circuit topology, but also on the modulation strategy adopted to control the switching devices. The type of modulation — as sinusoidal PWM (SPWM), space vector PWM modulation, or hysteresis-based techniques — directly affects parameters such as the harmonic content of voltage and current, switching losses, DC-link voltage utilization, current ripple, and the electromagnetic stress on the components. Consequently, the choice of the modulation technique represents a crucial design aspect, as it determines the trade-off among waveform quality, efficiency, device stress, and control system complexity.

In this work, the traditional SV-PWM has been adopted: it allows to get a phase to neutral peak voltage amplitude $\frac{2}{\sqrt{3}}$ higher than sinusoidal PWM, therefore a better exploiting of the linear operating region of the inverter as explained in [12].

Such control technique is based on the vectorial representation of the voltage space vectors that the inverter is able to generate. Each of the three legs has two possible configurations, therefore eight voltage space vector can be generated: such limit is due to the inverter topology, not related to the control strategy adopted.

The state S_i of each leg, and consequently the status of its corresponding switches (upper switch $T_{i,up}$, down $T_{i,down}$), can be expressed by:

$$\begin{cases} S_i = 0 \rightarrow T_{i,up} \ \ OFF \ \ ; T_{i,down} \, ON & (3.1a) \\ S_i = 1 \rightarrow T_{i,up} \ \ ON \ \ ; T_{i,down} \, OFF. & (3.1b) \end{cases}$$

In the following Fig.3.2 and Tab.3.1, the space vectors of the two level three phases inverter are represented:

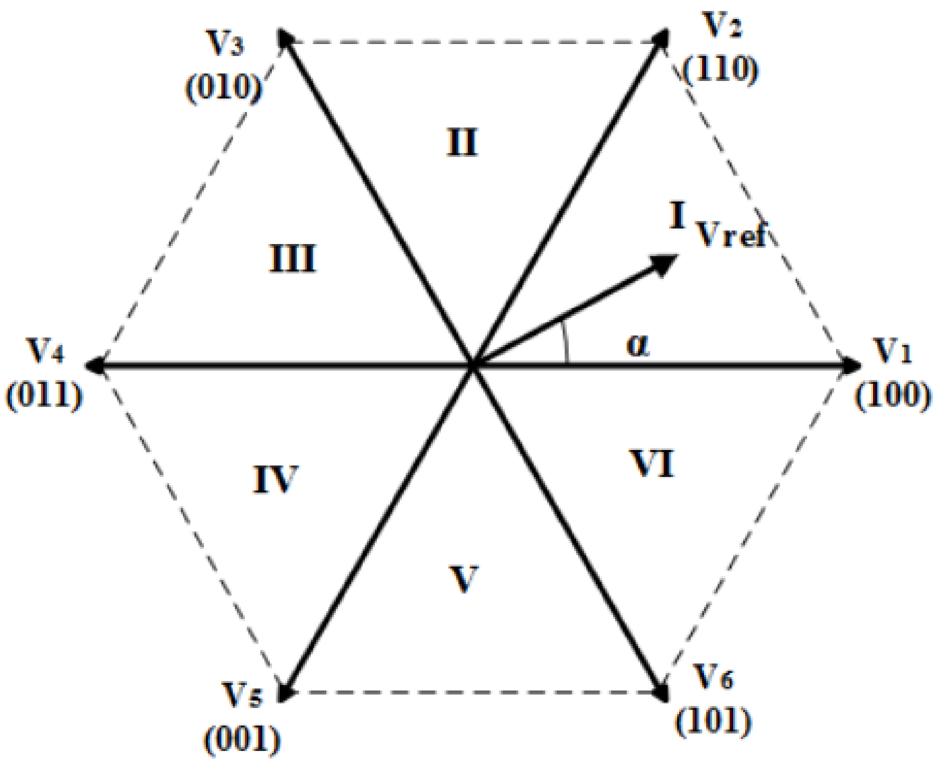

Figure 3.2: Space vectors schematic representation for two level three phase inverter [13].

Space Vector	Value
100	$\frac{2}{3}V_{dc}$
110	-$\frac{2}{3}\alpha^2 V_{dc}$
010	$\frac{2}{3}\alpha V_{dc}$
011	-$\frac{2}{3}V_{dc}$
001	$\frac{2}{3}\alpha^2 V_{dc}$
101	-$\frac{2}{3}\alpha V_{dc}$
111	0
000	0

Table 3.1: Space vectors values for two level three phase inverter.

The complex plan can be seen as the union of 6 regions: each of them is identified by the area between the inverter space vectors. Two last configurations in Tab.3.1 represent the cases in which the three phases are short-circuited (i.e. zero voltage space vector), not present in Fig.3.2 since they coincide with the origin of complex plane. The operating region of the inverter corresponds to the surface inside the hexagon identified by the space vectors.

The idea of the traditional SV-PWM technique is to reconstruct the desired reference voltage, expressed as reference space vector $\underline{V}_{ref}$, as temporal-average combination of the inverter space vectors. This computation is performed each switching period, $T_{switch} = \frac{1}{f_{switch}}$, in which the reference space vector can be considered as constant quantity since the electrical time constant $\tau_{electrical} >> T_{switch}$ .In order to define the time intervals of the inverter configurations in each switching period, the fraction time values σ, β and γ are introduced. They express the percentage of T_{switch} related to their corresponding inverter configuration.

In the following equation (3.2) is reported the system solved to identify the time intervals for a reference vector belonging to region I°, but the same applies for the others sections:

$$\begin{bmatrix} T_{switch} \\ \Re(\underline{V}_{ref}) \\ \Im(\underline{V}_{ref}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ \Re(\underline{V}_{100}) & \Re(\underline{V}_{110}) & \Re(\underline{V}_{111/000}) \\ \Im(\underline{V}_{100}) & \Im(\underline{V}_{110}) & \Im(\underline{V}_{111/000}) \end{bmatrix} \begin{bmatrix} \sigma \\ \beta \\ \gamma \end{bmatrix} \tag{3.2}$$

The next step consists in the construction of the sequence of configurations to supply the PMSM. Indeed, either because the zero vector can be defined with two configurations and each vector can be applied more than once in a switching period, there are several possible sequence that can be designed. There are many studies in literature that focus on this topic [14],[15],[16]. Not all the possible sequences are valid, but there are rules to be satisfied that applies for each subinterval.

In particular, as reported in [16]:

1. The active states must be applied at least once in a sub-cycle
2. Either the zero state 0 or the zero state 7 must be applied at least once in a sub-cycle
3. Only one phase must switch for a state transition
4. The total number of switches in a sub-cycle must be less than or equal to three

Conditions 1) and 2) are used to ensure voltage balance while 3) keeps the switching losses low. Condition 4) is to ensure that the average switching frequency is less than or equal to that of conventional SV-PWM for a given sampling frequency.
In the following Fig.3.3 , different approaches in defining the space vectors symmetric sequences are showed:

Sector	Conventional sequence	Clamping sequences	Special sequence type I	Special sequence type II
I	(0127, 7210)	(012, 210), (721, 127)	(0121, 1210), (7212, 2127)	(1012, 2101), (2721, 1272)
II	(7230, 0327)	(723, 327), (032, 230)	(7232, 2327), (0323, 3230)	(2723, 3272), (3032, 2303)
III	(0347, 7430)	(034, 430), (743, 347)	(0343, 3430), (7434, 4347)	(3034, 4303), (4743, 3474)
IV	(7450, 0547)	(745, 547), (054, 450)	(7454, 4547), (0545, 5450)	(4745, 5474), (5054, 4505)
V	(0567, 7650)	(056, 650), (765, 567)	(0565, 5650), (7656, 6567)	(5056, 6505), (6765, 5676)
VI	(7610, 0167)	(761, 167), (016, 610)	(7616, 6167), (0161, 1610)	(6761, 1676), (1016, 6101)

Figure 3.3: Symmetric space vectors sequences for SV-PWM of two level three phase inverter [16].

In this work, the conventional SV-PWM is adopted. For instance, the sequence with its corresponding time intervals in a switching period for section I° will be:

Space Vector	Application time
000	$\gamma/4$
100	$\sigma/2$
110	$\beta/2$
111	$\gamma/2$
110	$\beta/2$
100	$\sigma/2$
000	$\gamma/4$

Table 3.2: Space vector sequence adopted for region I°.

According to this design, the sequence of the active and zero vectors is distributed in a specular way inside the switching period, ensuring a temporal centring of the switching pulses and an improved harmonic distribution.

The last step involves the algorithm that outputs the phase to neutral voltages that supply the PMSM. This part is easily performed by computing, for each configuration:

$$\begin{cases} v_a = \dfrac{(2S_a - S_b - S_c)}{3} V_{dc} & \text{(3.3a)} \\ v_b = \dfrac{(2S_b - S_a - S_c)}{3} V_{dc} & \text{(3.3b)} \\ v_c = \dfrac{(2S_c - S_a - S_b)}{3} V_{dc} & \text{(3.3c)} \end{cases}$$

In the following Fig. 3.4, the Simulink schematic of the inverter controlled through conventional SV-PWM.

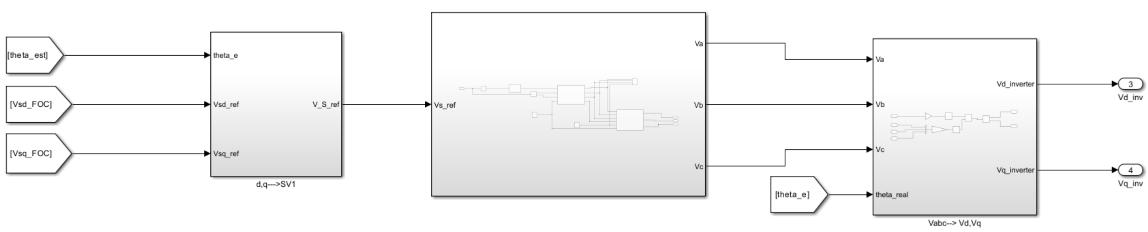

Figure 3.4: Simulink implementation of two level three phase inverter and SV-PWM.

The complete algorithm implemented for the SV-PWM is reported in detail in Appendix B.

In the next section, the discrete time implementation of the MRAS sensor-less control strategy has been studied. Then, simulation results representing the complete discretized system, including the inverter model and its CSV-PWM control techniques, will be discussed.

3.2. Discrete time implementation of MRAS

The discretization of the anisotropic PMSM control model represents a fundamental step toward future implementation on a digital platform. In particular, both the PMSM field oriented control and the estimation algorithm based on MRAS, as reported in Fig.2.5, have been converted from a continuous-time formulation to a discrete-time one, in order to accurately reproduce the behaviour of a real system based on a microcontroller, where signals are processed at finite sampling intervals. In particular, the sampling intervals are computed with a clock frequency $f_c = 10\ kHz$, corresponding to a control time $T_c = \frac{1}{f_c} = 100\ \mu s$.

The choice of sampling time must be performed by considering the Nyquist criterion:

$$f_c \geq 2f_{max} \tag{3.4}$$

where f_{max} is the maximum frequency associated with the sampled quantity.
The inverter introduced in the model is characterized by a switching frequency of $f_{switch} = 10\ kHz$, therefore the Nyquist criterion can't be respected through the observation period imposed by the microcontroller.

In the absence of an appropriate anti-aliasing filter, these components are folded back into the useful bandwidth, introducing systematic distortions in the measured quantities. This effect compromises the accuracy of the current measurement used in the MRAS speed estimation algorithm, altering the adaptation mechanism between the reference and adjustable models. As a consequence, persistent speed estimation errors, increased estimation noise, and, in the most critical cases, stability issues of the sensor-less control system may arise.

For this reason, an anti-aliasing filter must be employed in the model, with the aim to remove the higher spectral components of the PMSM currents. In particular, a discrete first order low pass filter (LPF) has been introduced, considering a cut-off frequency of $f_H^{LPF} = 5\ kHz$.

In the following Fig.3.5 is reported the Simulink implementation of the $\alpha - \beta$ stator current measurements:

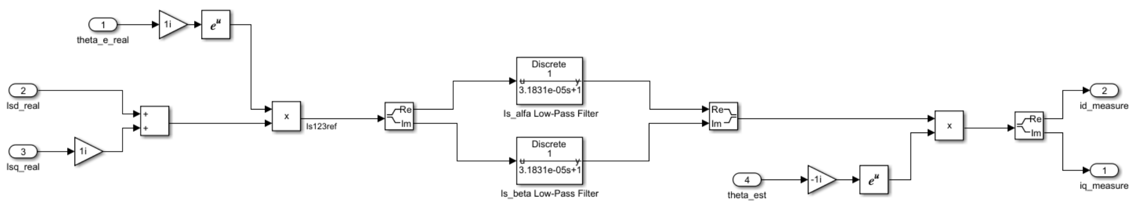

Figure 3.5: Simulink schematic of the $\alpha - \beta$ stator current measurements.

In real-world implementations, measurement acquisition, control computation, and PWM update must be properly synchronized to ensure coherent timing between sensed variables and applied control actions. However, within a MATLAB/Simulink simulation environment, such synchronization is not explicitly required. The solver operates in a deterministic and ideally time-aligned numerical framework, where measurements, control computations, and actuation updates are inherently synchronized at each simulation step, and switching noise or hardware-induced timing mismatches are not physically present.

Regarding discretization techniques, a differential equation can be numerically solved by selecting a discretization method to approximate continuos time integration [17].
The main approaches are:

1. forward Euler method: $\int_{(n-1)T_c}^{nT_c} x(t)dt \approx x(n-1)T_c$
2. backward Euler method:$\int_{(n-1)T_c}^{nT_c} x(t)dt \approx x(n)T_c$
3. trapezoidal method:$\int_{(n-1)T_c}^{nT_c} x(t)dt \approx \frac{x(n)+x(n-1)}{2}T_c$

Their graphic representation is reported in Fig.3.6

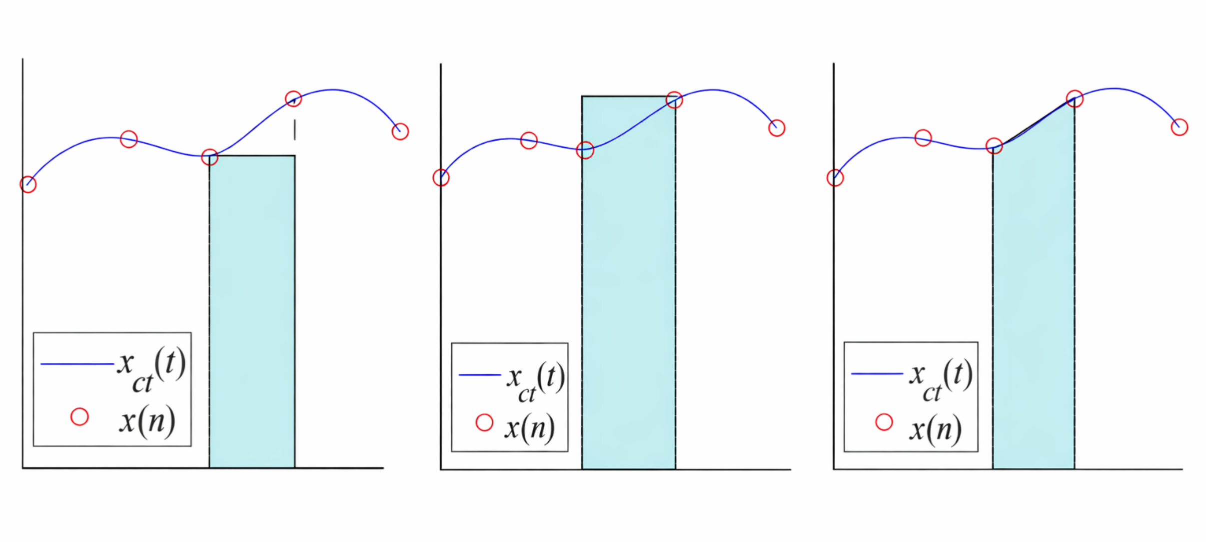

Figure 3.6: Graphic representation of: forward Euler, backward Euler and trapezoidal discretization methods [17].

For the discretization of the complete model, different strategies were adopted depending on the numerical sensitivity of the various parts of the system. In general, the trapezoidal method has been employed, as it provides a better approximation of the frequency-domain dynamics while preserving the stability properties of the continuous-time system. However, for the discrete d–q machine model used in the adaptive MRAS model, which is characterized by direct feedback of the currents, the use of the trapezoidal method led to convergence issues and algebraic loops in Simulink. Therefore, for this specific part, the Forward Euler method has been selected.

The following Fig.3.7 represents the discretized MRAS model implemented in Simulink:

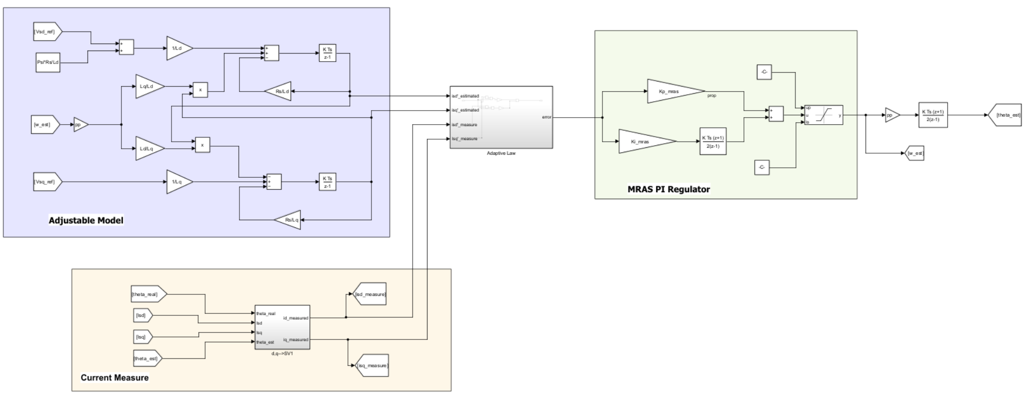

Figure 3.7: Discrete time MRAS algorithm Simulink implementation.

3.3. Simulation results for discrete time model

Following the discretization process of the anisotropic PMSM control model and the MRAS algorithm, a thorough analysis of the obtained results was carried out in order to assess the consistency between the behaviour of the discrete-time system and that of the continuous reference model. The main objective of this phase is to verify how the introduction of sampling and numerical approximation affects the dynamic performance of the vector control scheme and also the accuracy of the estimation provided by the adaptive system.

The PI controller parameters of the MRAS have been properly retuned comparing to the continuos time simulations (Tab.2.1), effectively scaling them with respect to the continuous-time design. This adjustment is necessary because discretization introduces sampling effects and modifies the equivalent dynamic behaviour of the controller, particularly in terms of integral action characteristic. Without proper scaling, the discrete implementation exhibits increased oscillations and instabilities compared to the continuous-time formulation, leading the simulations to diverge. Therefore, as reported in Tab.3.3, the PI gain parameters used in the discretized model:

Parameter	**Speed**	**Currents d/q**	**MRAS**
K_p	60.35	1.069 /1.272	0.00025
K_i	58.112	1026.7 /1221.9	0.006

Table 3.3: PI regulators setting specification for discrete time simulation.

While in Tab.3.4 are summarized the main parameters describing the inverter system and the discretization:

Parameter	**Value**
T_c	100 μs
f_{switch}	10 kHz
V_{dc}	540 V

Table 3.4: Control time and inverter switching frequency and V_{dc}.

The results of the simulation are presented in Fig.3.8 to 3.17. The system employed is the the same described in previous chapters. The Simulink model of the simulation , has been run with fixed step ode3 solver and maximum speed size 10^{-6}. Compared to the continuos time simulation in chapter 2, here in order to avoid excessive stress on the discrete-time MRAS estimator, improving numerical stability and reducing transient estimation errors, the reference speed is described by an initial step up to $\omega_{base}/2$, then, once the steady state is reached, a ramp is applied to bring the system up to the base speed.

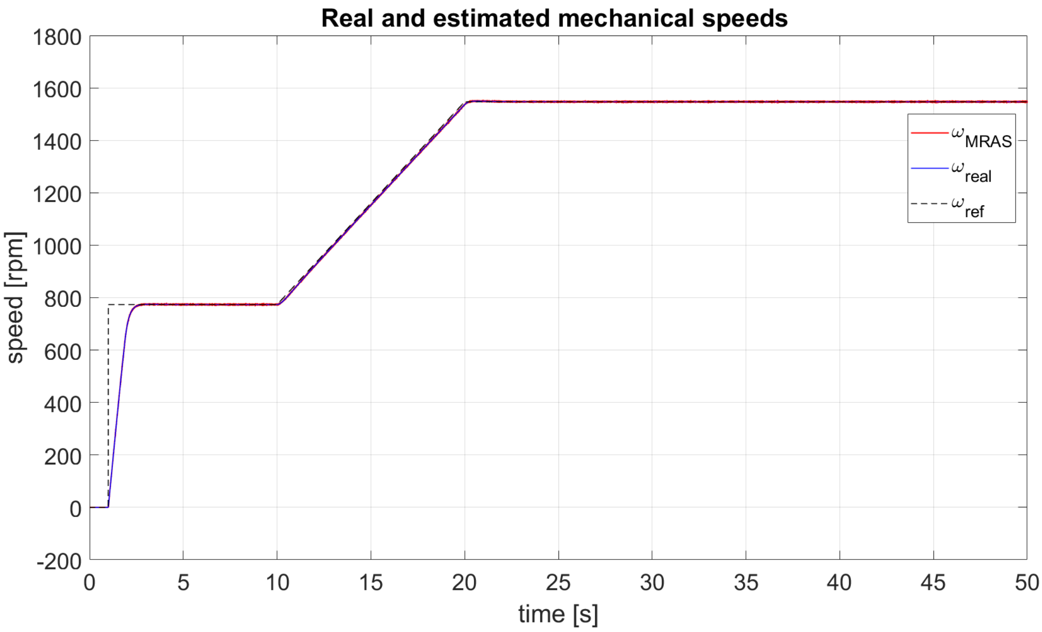

Figure 3.8: Real and estimated mechanical speeds of the discretized model.

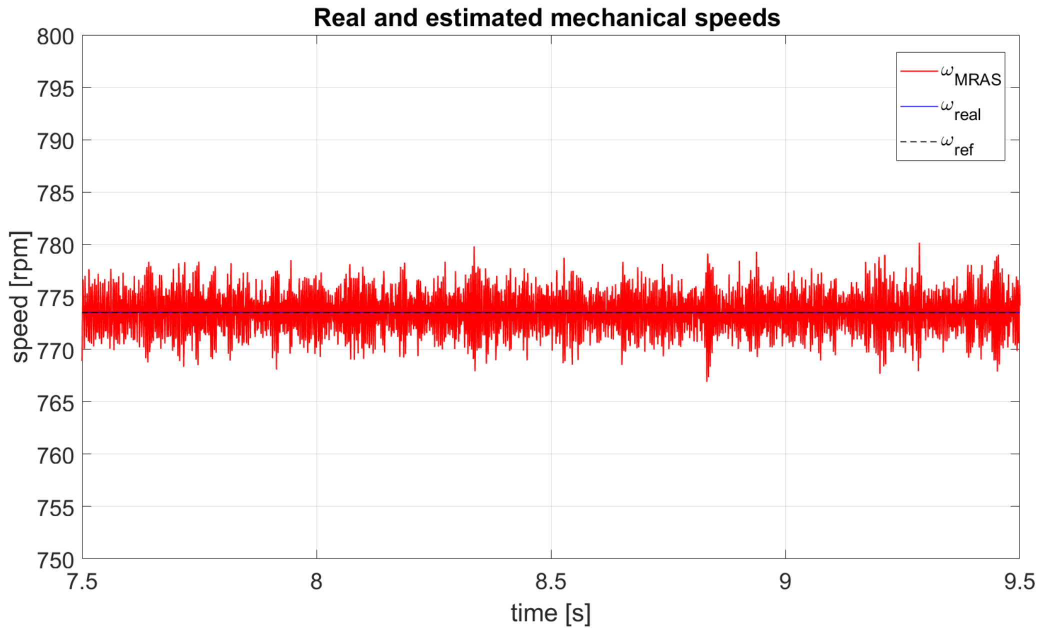

Figure 3.9: Zoom of steady state speed behavior at $\omega_{base}/2$.

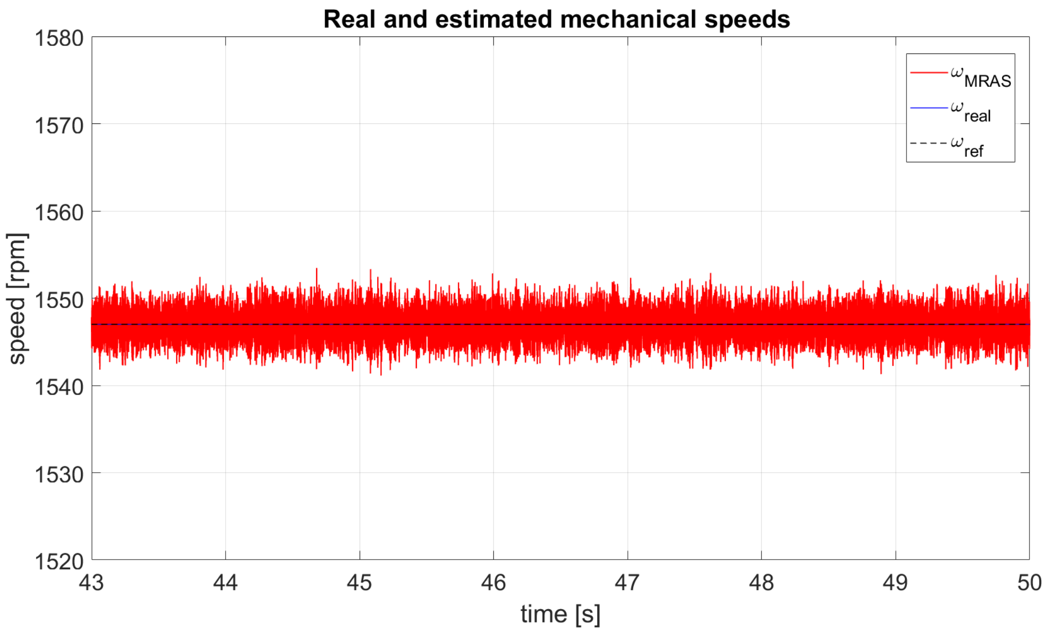

Figure 3.10: Zoom of steady state speed behaviour at ω_{base}.

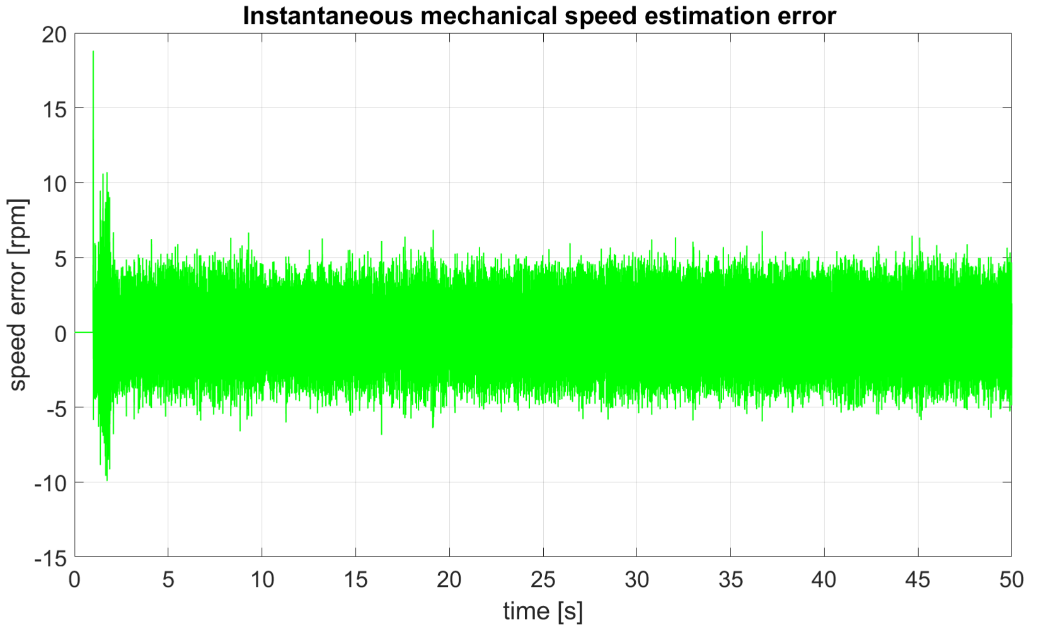

Figure 3.11: Instantaneous mechanical speed error.

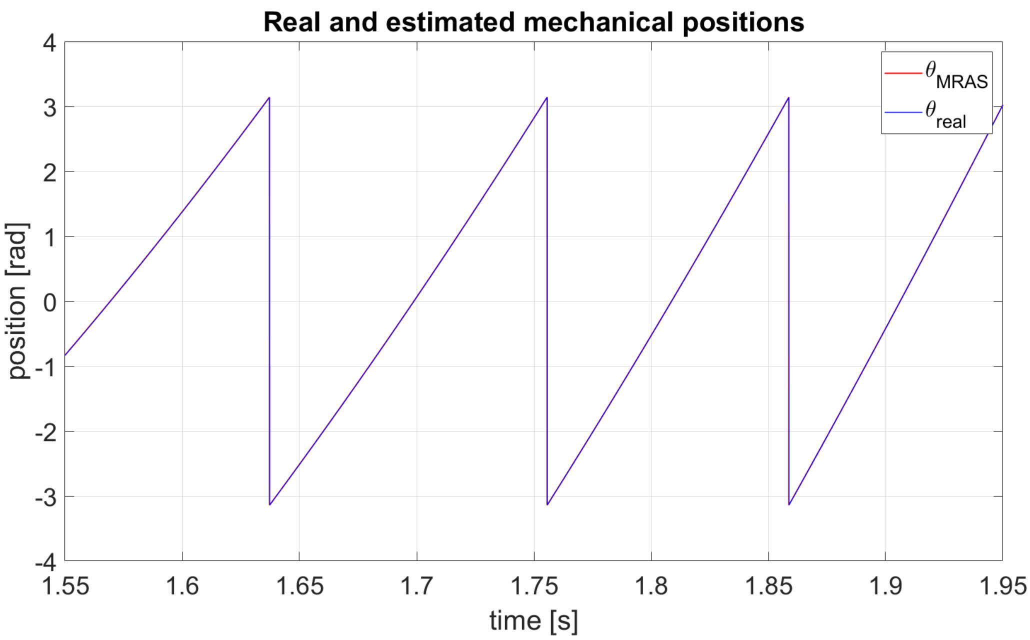

Figure 3.12: Zoom of real and estimated mechanical positions.

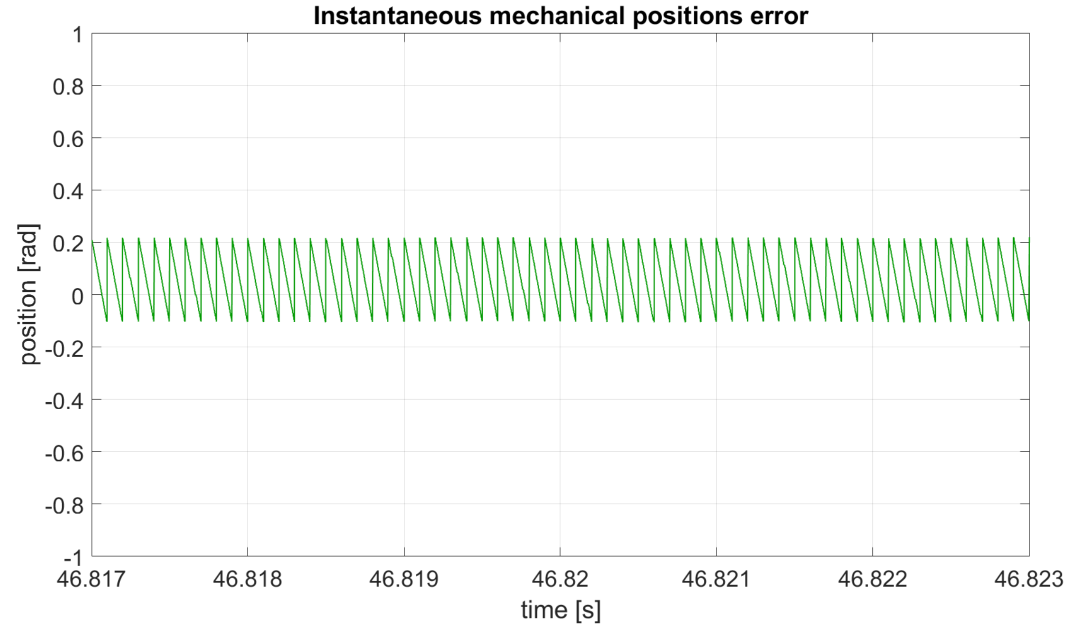

Figure 3.13: Instantaneous mechanical position error.

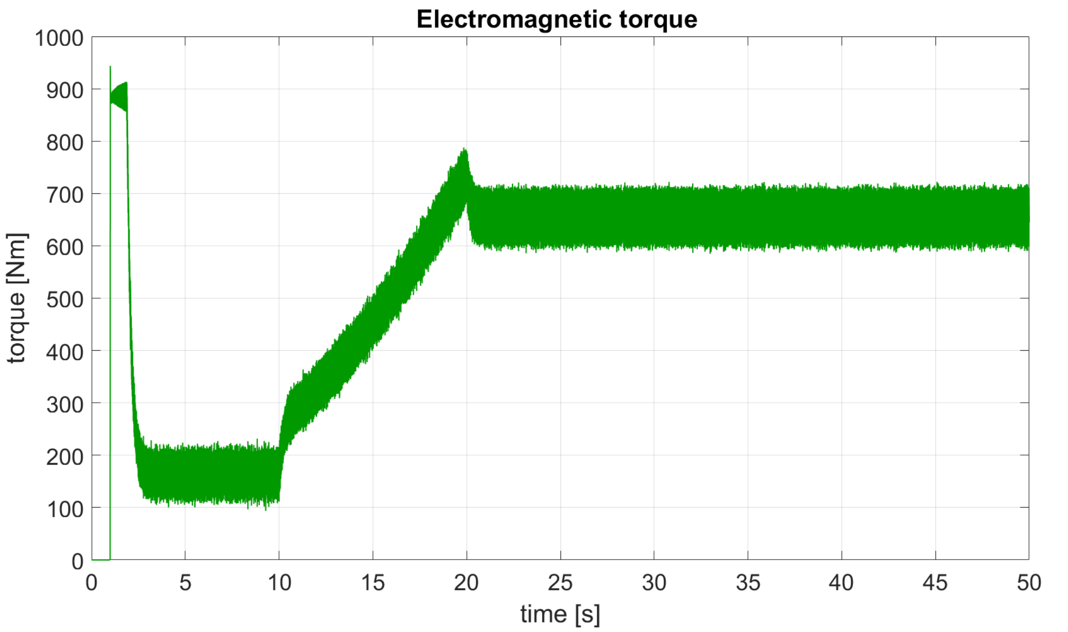

Figure 3.14: Electromagnetic torque of the discretized model.

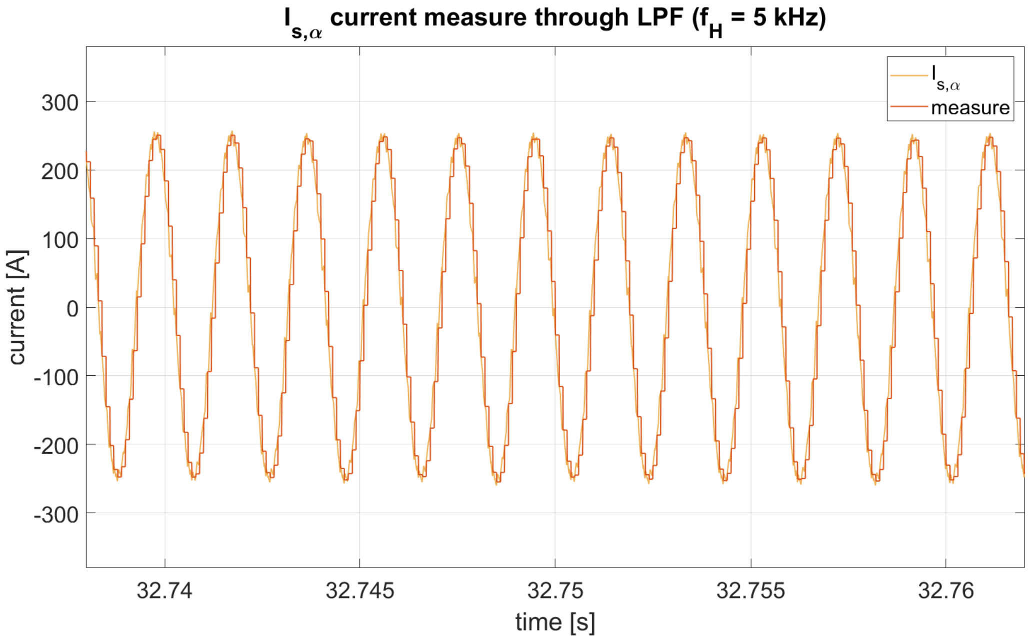

Figure 3.15: Zoom of actual and measured $I_{s,\alpha}$ currents.

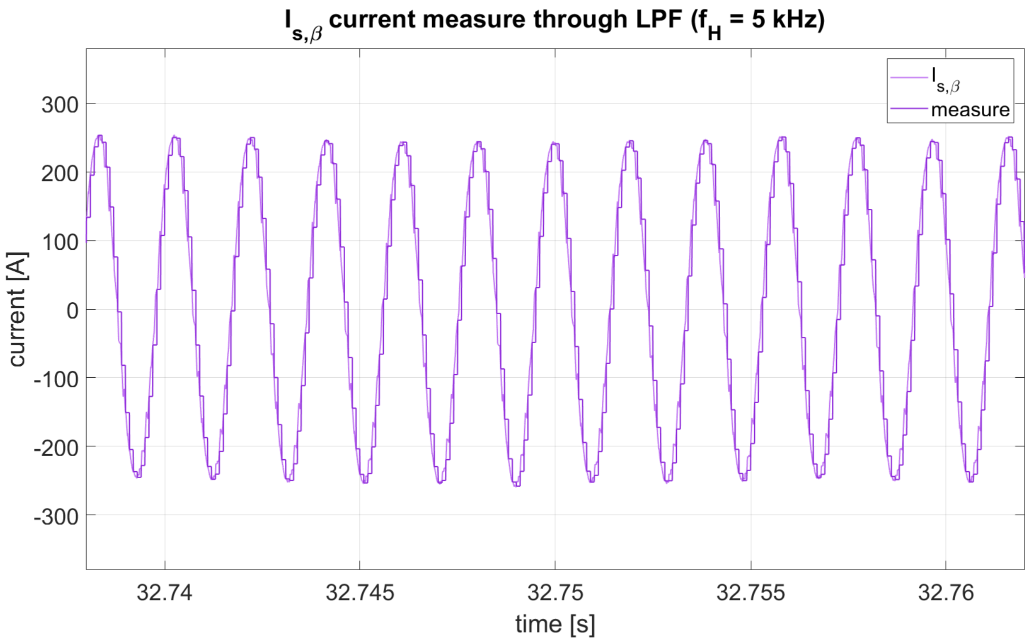

Figure 3.16: Zoom of actual and measured $I_{s,\beta}$ currents.

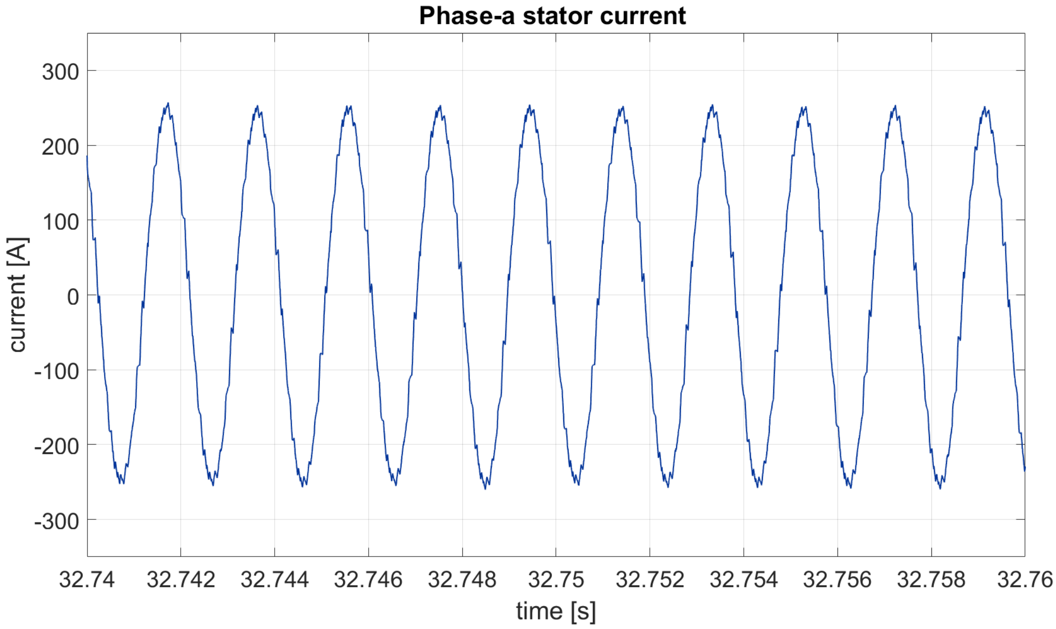

Figure 3.17: Zoom of steady state phase-a stator current at ω_{base}.

The results obtained with the discretized MRAS demonstrate that the estimated speed closely tracks the actual rotor speed over the entire operating range (Fig.3.8), confirming the effectiveness of the estimator.
The dynamic response is preserved with respect to the continuous-time implementation, and no steady-state bias is observed for the actual speed value. Small oscillations around the true speed value are present in the estimate, as it possible to observe in Fig.3.9 and in Fig.3.10; however, they remain in a range of ± 7 rpm at steady state. These oscillations present in the speed estimation affects the FOC of the motor when fed back, introducing oscillations in the reference v_{sd} and v_{sq} and therefore current ripple. The speed oscillations can be mainly attributed to discretization effects, limited sampling frequency and to the phase delay introduced by the LPF employed in current measure. Such delay resulting from the discretized model can be observed also by the position estimation in Fig. 3.12, that directly affects the d-q reference system estimation. Anyway an accurate synchronization between the estimated and actual d-q reference frame is maintained.
While according to the current measurement, in Fig.3.15 and Fig.3.16 are reported the resulting waveforms that are then used in the adaptive law of the MRAS algorithm. Here it's possible to observe the filtering effect of the LPF, that helps reducing the harmonic content of the waveform resulting in a sinusoidal shape of the current. The sinusoidal shape of the measurement reduces considering higher motor's speed. This is because the electrical frequency of the currents increase with speed while the sampling intervals is kept at $f_c = 10\ kHz$. But this does not affect the MRAS algorithm, since it presents limitations in the low-speed region due to the lower value of the back emf, where the current measures must be as precise as possible. Then the analysis of the stator phase-a current under conventional space vector PWM control, represented in in Fig.3.17, highlights the effectiveness of the modulation strategy in ensuring high current quality. The simulated waveform exhibits a nearly sinusoidal profile with limited ripple, confirming the capability of SV-PWM to reduce harmonic distortion, helping also in the limitation of the torque ripple. These results demonstrate that the discrete implementation is suitable for practical real-time applications, confirming the robustness and effectiveness of the proposed MRAS-based sensorless control strategy.

4 | Conclusions

The main objective of this thesis work has been the development and validation of a sensor-less control strategy for an anisotropic permanent magnet synchronous motor, based on MRAS algorithm. The model has been implemented in MATLAB/Simulink environment in order to analyse its performances both in continuos time and in a discrete context similar to real applications.

Firstly, the preliminary analysis of the PMSM controlled by the FOC, including MTPA and MPTV optimization strategies, has been the solid basis of the entire system. The results of Chapter 1 shows the correct decoupling between current components, according to voltage and current constraint. This phase has been the reference for the introduction of the sensor-less algorithm, so that any criticalities would have been attributed to the estimator.

Then the MRAS sensor-less control algorithm, characterized by a simple structure and low computation effort, has been defined. In particular, the definition of the reference model, adjustable model and adaptive law demonstrates to be coherent with the dynamic of the motor and FOC. Indeed, continuos time simulation results has showed an accurate speed estimate, that converge perfectly to the actual speed of the motor during the whole simulation period, confirming the theory proposed of the algorithm and its correct implementation.

The update of the model in the discrete time frame is the key aspect of the work, since it enables to understand the discretization effects and the impact of the employment of the inverter. In particular, the aliasing phenomenon has been faced, regarding the stator current measurements of the motor. This problem is raised because of the limited sampling frequency of $f_c = 10\ kHz$ compared to the maximum harmonic content of the currents: in this way, the Nyquist criterion can't be fulfilled, resulting in instabilities of the MRAS algorithm. Therefore, the use of a discrete first order low pass filter has been a decisive step to guarantee stability and avoid divergences in the system. The results show permanent oscillations in the speed estimate, confined $\pm$ 7 rpm with respect to the reference value. This aspect is related to the delays resulting from discretization and the harmonic content produced by the inverter. Nevertheless, at steady state, the estimated speed is perfectly coherent with the actual, and this last don't present any oscillations around the reference value. Therefore, compared to the continuos-time results, the discretized system maintain its performance also in a more realistic environment.

In general, the work shows that the MRAS approach represents an effective solution for sensor-less control of the anisotropic PSMS, ensuring accuracy in the speed estimation and stability of the discretization system. Possible further studies can be carried out to improve the system, as including the uncertainty on electrical parameters, degradations and more realistic PMSM model in order to understand the effects on the estimator.

Bibliography

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

[2] X. Sun, Z. Jin, M. Xue, and X. Tian, “Adaptive ecms with gear shift control by grey wolf optimization algorithm and neural network for plug-in hybrid electric buses,” *IEEE Transactions on Industrial Electronics*, vol. 71, no. 1, pp. 667–677, 2024.

[3] Z. Xiang, Y. Zhou, X. Zhu, L. Quan, D. Fan, and Q. Liu, “Research on characteristic airgap harmonics of a double-rotor flux-modulated pm motor based on harmonic dimensionality reduction,” *IEEE Transactions on Transportation Electrification*, vol. 10, no. 3, pp. 5750–5761, 2024.

[4] H. Qiu, C. Yuan, W. Chen, X. Ma, and B. Xiong, “Combined regulation performance research of the novel flux-torque regulation hybrid excitation machine with axial-radial magnetic circuit,” *IEEE Transactions on Transportation Electrification*, vol. 10, no. 4, pp. 10 420–10 427, 2024.

[5] M. E. Romero, “Analysis and comparison of the interior permanent magnet motor design with different magnetic materials,” in *2020 IEEE International Conference on Electro Information Technology (EIT)*, 2020, pp. 249–254.

[6] Control Engineering. (2024) Understanding permanent magnet motors. [Online]. Available: https://www.controleng.com/

[7] S. J. Gaolin W., Valla M., “Position sensorless permanent magnet synchronous machine drives-a review,” *IEEE*, vol. 67, 2020.

[8] N. H. Saad and A. A. El-Sattar, “Sensorless field oriented control based on improved mras speed observer for permanent magnet synchronous motor drivesensorless field oriented control based on improved mras speed observer for permanent magnet synchronous motor drive,” *IEEE*, 2016, the Department of Electrical Engineering and Applied Electronics Tsinghua University.

[9] Z. F. G. X. Z. P. Zhuang Xingming, Wen Xuhui, "Wide-speed-range sensorless control of interior pmsm based on mras," *IEEE*, 2010, institute of electrical engineering of Chinese Academic of Sciences.

[10] M. Y. X.Xiao, L.Yongdong, "A sensorless control based on mras method in interior pernanent-magnet machine drive," *IEEE*, 2005, the Department of Electrical Engineering and Applied Electronics Tsinghua University.

[11] M. R. I. Airin Rahman, Md. Moshiur Rahman, "Performance analysis of three phase inverters with different types of pwm techniques," *ICEEE*, 2017, department of Electrical and Electronic Engineering, Rajshahi University of Engineering Technology, Rajshahi, Bangladesh.

[12] H. van der Broeck, H.-C. Skudelny, and G. Stanke, "Analysis and realization of a pulsewidth modulator based on voltage space vectors," *IEEE Transactions on Industry Applications*, vol. 24, no. 1, pp. 142–150, 1988.

[13] M. Khalid, A. Mohan, and B. A. C, "Performance analysis of vector controlled pmsm drive modulated with sinusoidal pwm and space vector pwm," in *2020 IEEE International Power and Renewable Energy Conference*, 2020, pp. 1–6.

[14] G. Narayanan and V. Ranganathan, "Two novel synchronized bus-clamping pwm strategies based on space vector approach for high power drives," *IEEE Transactions on Power Electronics*, vol. 17, no. 1, pp. 84–93, 2002.

[15] M. Wang, J. Yang, and C. Zhu, "Hybrid svpwm technique for reduced torque ripple in permanent magnet synchronous motor," in *2014 International Power Electronics and Application Conference and Exposition*, 2014, pp. 1297–1302.

[16] G. Narayanan, D. Zhao, H. K. Krishnamurthy, R. Ayyanar, and V. T. Ranganathan, "Space vector based hybrid pwm techniques for reduced current ripple," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 4, pp. 1614–1627, 2008.

[17] S. Toscani, "Measurement-oriented digital signal processing," 2024, lecture slides, Politecnico di Milano.

A | Appendix A

Here below is reported the algorithm implemented for the PI regulators tuning:

Algorithm A.1 PI Tuning Algorithm

Input: [Cut-off frequency f_c, tuning parameter k_{ppu}, resistance R, inductance L]

$\omega_c = 2\pi f_c$

$K_{pMax} = R + \sqrt{2R^2 + (\omega_c L)^2}$

$K_p = \min(\max(R, k_{ppu} \cdot K_{pMax}), K_{pMax})$

$K_i = -\omega_c^2 L + \omega_c \sqrt{2(\omega_c L)^2 - K_p^2 + R^2 + 2K_p R}$

Output: Proportional gain K_p, Integral gain K_i

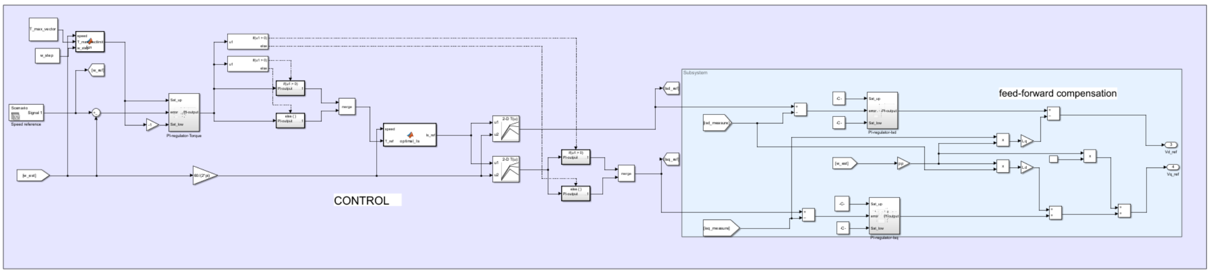

Figure A.1: Complete Simulink schematic of FOC.

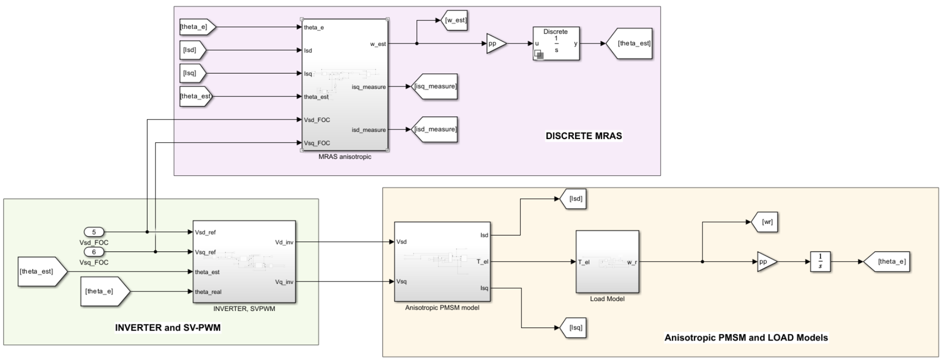

Figure A.2: Complete Simulink schematic of inverter, discretized MRAS and PMSM.

B | Appendix B

Here the MTPA/MTPV algorithm to define the FOC for the PMSM is reported :

```
function [id,iq]= MTPA_MTPV_algorithm_anisotropic(speed,Is_max,V_max,Ld,Lq,Psi,pp,w_base_est,w_0_est,I_step)

         speed_el=speed*2*pi*pp/60;

        if (speed <= w_base_est)
            j=1;
                for i=0:I_step:Is_max
                     id(j)=(1-sqrt(1+8*((Ld-Lq)*i/Psi)^2))/(4*(Lq-Ld)/Psi);
                     iq(j)=sqrt(i^2-id(j)^2);
                     j=j+1;
                end
        else if (speed> w_base_est && speed <= w_0_est)
                     j=1;
                     for i=0:I_step:Is_max
                        id(j)=(1-sqrt(1+8*((Ld-Lq)*i/Psi)^2))/(4*(Lq-Ld)/Psi);
                        iq(j)=sqrt(i^2-id(j)^2);
                        check=sqrt((speed_el*Lq*iq(j))^2+(speed_el*Ld*id(j)+speed_el*Psi)^2);
                        if (check>V_max)
                           x=Psi^2+Lq^2*i^2-(V_max/speed_el)^2;
                           id(j)=(-2*Ld*i*Psi+sqrt(4*Ld^2*i^2*Psi^2-4*(Ld^2-Lq^2)*i^2*x))/(2*(Ld^2-Lq^2)*i);
                           iq(j)=sqrt(i^2-id(j)^2);
                           j=j+1;
                         else j=j+1;
                         end
```

```
                    end
        else    % here id assumes negative value (ref) even if zero torque is applied (iq=0), to satisfy voltage constraint
                j=1;
                ref=(V_max-speed_el*Psi)/(speed_el*Ld);
                for i=0:I_step:Is_max
                    if i<abs(ref)
                        id(j)=ref+(10^-14)/(i+1); % Initially NaN, now this term is used to provide a strictly monotonically increasing behaviour for Look-up table
                        iq(j)=(10^-14)*i;         % Initially NaN, // // //
                        j=j+1;
                    else
                            x=Psi^2+Lq^2*i^2-(V_max/speed_el)^2;
                            id(j)=(-2*Ld*i*Psi+sqrt(4*Ld^2*i^2*Psi^2-4*(Ld^2-Lq^2)*i^2*x))/(2*(Ld^2-Lq^2)*i);
                            iq(j)=sqrt(i^2-id(j)^2);
                            j=j+1;
                    end
                end
        end
      end
end
```

Listing B.1: MTPA-MTPV curves implementation for FOC

In the following the two algorithms used for the SV-PWM control technique of the three-phase two-level inverter:

```

function [Sigma,Beta,Gamma,SV_pos]= switchingtimes(SV_pos,SV_real,SV_imm,Vdc)


 alfa = 1* exp (1i *(2* pi)/3);
 alfa_2 = 1* exp (1i *(4* pi)/3);
 A=zeros(3,3);
```

```
 B=[1; SV_real; SV_imm];

% Inverter SV voltages (2 null vectors + 6 active vectors )

 V_100 = (2/3)*Vdc;
 V_010 = (2/3)*(alfa*Vdc );
 V_001 = (2/3)*(alfa_2*Vdc );
 V_110 = -(2/3)*(alfa_2*Vdc );
 V_101 = -(2/3)*(alfa *Vdc );
 V_011 = -(2/3)*(Vdc);
 V_111 = 0;
 V_000 = 0;

if SV_pos >= 0 && SV_pos < (pi/3) % region 1

    A=[1 1 1; real(V_100) real(V_110) real(V_111); imag(V_100)
       imag(V_110) imag(V_111)];

elseif SV_pos >= (pi/3) && SV_pos < (2*pi/3) % region 2

    A=[1 1 1; real(V_010) real(V_110) real(V_000); imag(V_010)
       imag(V_110) imag(V_000)];

elseif SV_pos >= (2*pi/3) && SV_pos < pi % region 3

    A=[1 1 1; real(V_010) real(V_011) real(V_111); imag(V_010)
       imag(V_011) imag(V_111)];

elseif SV_pos >= pi && SV_pos < (4*pi/3) % region 4

    A=[1 1 1; real(V_001) real(V_011) real(V_000); imag(V_001)
       imag(V_011) imag(V_000)];

elseif SV_pos >= (4*pi/3) && SV_pos < (5*pi/3) % region 5

    A=[1 1 1; real(V_001) real(V_101) real(V_111); imag(V_001)
       imag(V_101) imag(V_111)];

elseif SV_pos >= (5*pi/3) && SV_pos < (2*pi) % region 6
```

```
    A=[1 1 1; real(V_100) real(V_101) real(V_000); imag(V_100)
       imag(V_101) imag(V_000)];

else
    A=[0 0 0; 0 0 0; 0 0 0];

end

times = A\B;

Sigma=times(1);
Beta=times(2);
Gamma=times(3);

SV_pos=SV_pos;

end
```

Listing B.2: Space Vector computation of switching times intervals

```
function [Va,Vb,Vc] = fcn(Alfa,Beta,Gamma,SV_pos,clock,Vdc)


    v=[Gamma/4; Alfa/2; Beta/2; Gamma/2; Beta/2; Alfa/2; Gamma/4];
    S1=0;
    S2=0;
    S3=0;
    t=cumsum(v); % returns t = [gamma/4 ; gamma/4 + alfa/2 ; ...]


    if SV_pos >=0 && SV_pos <( pi /3) % Region 1

        if clock<=t(1)
            S1=0; S2=0; S3=0;
        elseif clock<t(2)
            S1=1; S2=0; S3=0;
        elseif clock <t(3)
            S1=1; S2=1; S3=0;
        elseif clock <t(4)
```

```
            S1=1; S2=1; S3=1;
        elseif clock <t(5)
            S1=1; S2=1; S3=0;
        elseif clock <t(6)
            S1=1; S2=0; S3=0;
        else
            S1=0; S2=0; S3=0;
        end

    elseif SV_pos >=( pi /3) && SV_pos <(2* pi /3) %Region2

         if clock<=t(1)
            S1=0; S2=0; S3=0;
         elseif clock<t(2)
            S1=0; S2=1; S3=0;
         elseif clock<t(3)
            S1=1; S2=1; S3=0;
         elseif clock<t(4)
            S1=1; S2=1; S3=1
         elseif clock<t(5)
            S1=1; S2=1; S3=0;
         elseif clock<t(6)
            S1=0; S2=1; S3=0;
         else
            S1=0; S2=0; S3=0;
         end

    elseif SV_pos >=(2* pi /3) && SV_pos <( pi) %Region3

         if clock<=t(1)
            S1=0; S2=0; S3=0;
         elseif clock<t(2)
            S1=0; S2=1; S3=0;
         elseif clock<t(3)
            S1=0; S2=1; S3=1;
         elseif clock<t(4)
            S1=1; S2=1; S3=1;
         elseif clock<t(5)
            S1=0; S2=1; S3=1;
```

```
        elseif clock<t(6)
           S1=0; S2=1; S3=0;
        else
           S1=0; S2=0; S3=0;
        end

    elseif SV_pos >=( pi) && SV_pos <(4* pi /3)    %Region 4

        if clock<=t(1)
           S1=0; S2=0; S3=0;
        elseif clock<t(2)
           S1=0; S2=0; S3=1;
        elseif clock<t(3)
           S1=0; S2=1; S3=1;
        elseif clock<t(4)
           S1=1; S2=1; S3=1;
        elseif clock<t(5)
           S1=0; S2=1; S3=1;
        elseif clock<t(6)
           S1=0; S2=0; S3=1;
        else
           S1=0; S2=0; S3=0;
        end

       elseif SV_pos >=(4* pi /3) && SV_pos <(5* pi /3) %region5

        if clock<=t(1)
           S1=0; S2=0; S3=0;
        elseif clock<t(2)
           S1=0; S2=0; S3=1;
        elseif clock<t(3)
           S1=1; S2=0; S3=1;
        elseif clock<t(4)
           S1=1; S2=1; S3=1;
        elseif clock<t(5)
           S1=1; S2=0; S3=1;
        elseif clock<t(6)
           S1=0; S2=0; S3=1;
        else
```

```
                S1=0; S2=0; S3=0;
            end


    elseif SV_pos >=(5* pi /3) && SV_pos <(2* pi) %region6

            if clock<=t(1)
                S1=0; S2=0; S3=0;
            elseif clock<t(2)
                S1=1; S2=0; S3=0;
            elseif clock<t(3)
                S1=1; S2=0; S3=1;
            elseif clock<t(4)
                S1=1; S2=1; S3=1;
            elseif clock<t(5)
                S1=1; S2=0; S3=1;
            elseif clock<t(6)
                S1=1; S2=0; S3=0;
            else
                S1=0; S2=0; S3=0;
            end

    end

Va=(2*S1-S2-S3)/3*Vdc;
Vb=(2*S2-S1-S3)/3*Vdc;
Vc=(2*S3-S2-S1)/3*Vdc;

end
```

Listing B.3: Space Vector reference modulating voltages, according to switching time intervals

```
%% Simulation parameters

T_sim=20;                    % Simulation time [s]
T_step=1e-7;                 % Simulation resolution [s]
I_step= 0.05;                % current step [A]
T_controll=1e-4;             % Control time [s]
w_step =0.5;                 % speed discretization [rpm]
```

```
%% Motor [140kW] parameters

pp=20;                      % pole pairs
Rs=32e-3;                   % stator resistance [Ohm]
Ld=195e-6;                  % inductance in d-direction [H]
Lq=233e-6;                  % inductance in q-direction [H]
Psi=0.09;                   % PM flux [Wb]
Tn=891;                     % nominal torque [Nm]
Pn=140e3;                   % nominal power [W]
In=230;                     % nominal current rms [A]
Vn=440;                     % nominal voltage rms [A] phase to phase
w_base=1500;                % base speed [rpm]
w_0=1891;                   % base speed [rpm]
w_max=6383;                 % maximum speed [rpm]
J_0=0.25;                   % no-load inertia [kg*m^2]
B_0=0.0003;                 % no-load friction coeff. [Nms]
J_eq=11.3;                  % equivalent inertia load [kg*m^2]
B_load=0.025;               % proportional constant for load torque


motor_parameters=[Psi,Rs,Ld,Lq];


V_peak_n=sqrt(2/3)*Vn;      % peak-value of phase to neutral voltage
I_peak_n=sqrt(2)*In;        % peak-value at nominal rms current

%% Inverter parameters

fs=10000;                  % switching frequency [Hz]
Ts=1/fs;                   % Switching period [s]
Vdc=540;                   % DC-voltage inverter [V]

%% MRAS algorithm Kp,Ki parameters
Kp_mras =0.00025;
Ki_mras =0.006;
```

```
%% ANISOTROPIC MACHINE

w_max_est=round((V_peak_n/(pp*(-Ld*I_peak_n+Psi)))*30/pi) % [rpm]
w_0_est=round(V_peak_n/(pp*Psi)*30/pi)                     % [rpm]

Id_max=zeros(1,w_max_est);  % pre-loc Id_max vector
Iq_max=zeros(1,w_max_est);  % pre-loc Iq_max vector

j=1;
for i=0:w_step:w_max_est
[Id_max(j),Iq_max(j)] = MTPA_max(i,I_peak_n,V_peak_n,Ld,Lq,Psi,pp,
   w_base);
j=j+1;
end

% Max. torque values
T_max=3/2*pp*Psi*Iq_max+3/2*pp*(Ld-Lq)*Iq_max.*Id_max;


w_base_est=round((V_peak_n/pp)*1/(sqrt((Lq*Iq_max(1))^2+(Ld*Id_max
   (1)+Psi)^2))*30/pi); % [rpm]

% id,iq curves for FOC lookuptables
d=zeros(ceil(I_peak_n/I_step),w_max_est);
q=zeros(ceil(I_peak_n/I_step),w_max_est);

j=1;
for i=0:w_step:w_max_est
[d(:,j),q(:,j)]=MTPA_MTPV_algorithm_anisotropic(i,I_peak_n,
   V_peak_n,Ld,Lq,Psi,pp,w_base_est,w_0_est,I_step);
j=j+1;
end

T=3/2*pp*Psi*q+3/2*pp*(Ld-Lq)*q.*d;          % torque matrix


T_max_vector = T(round(I_peak_n/I_step),:);
```

```
%% PI-regulators parameters setup

[kp_t,ki_t]=PItune(1,0.85,B_0,J_eq);         % Kp,Ki parameters for
    speed PI-regulator
[kp_id,ki_id]=PItune(1000,0.85,Rs,Ld);       % Kp,Ki parameters for
    d-current PI-regulator
[kp_iq,ki_iq]=PItune(1000,0.85,Rs,Lq);       % Kp,Ki parameters for
    q-current PI-regulator
```

Listing B.4: MATLAB code for the discrete model simulation

List of Figures

List of Tables

List of Symbols

Variable	**Description**
$PMSM$	Permanent Magnet Synchronous Motor
FOC	Field Oriented Control
$MTPA$	Maximum Torque per Ampere
$MTPV$	Maximum Torque per Volt
$MRAS$	Model Reference Adaptive System
$SM-PMSM$	Surface Mounted Permanent Magnet Synchronous Motor
$IM-PMSM$	Internal Mounted Permanent Magnet Synchronous Motor
KVL	Kirchhoff's Voltage Law
EMF	Electromotive Force
VSI	Voltage Source Inverter
$SPWM$	Sinusoidal Pulse Width Modulation
$SV-PWM$	Space Vector Pulse Width Modulation
LPF	Low Pass Filter