

**Alma Mater Studiorum - Università di Bologna**

---

**SCUOLA DI INGEGNERIA ED ARCHITETTURA  
DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E  
DELL'INFORMAZIONE "GUGLIELMO MARCONI" - DEI**

**SECOND CYCLE DEGREE  
IN  
Electric Vehicle Engineering**

**DISSERTATION**

in

Automotive Power Circuits And Electric Motor Design M I.c.

**Application of the Extended Kalman filter for speed  
sensorless control of PM synchronous machines**

Candidate:  
Jacopo Ferretti

Supervisor:  
Prof. Gabriele Rizzoli

Co-supervisors:  
Giacomo Piero Schiapparelli, PhD

II Session  
Academic year 2021/2022

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>6</b>  |
| 1.1      | Thesis objective . . . . .                                   | 6         |
| 1.2      | The Electric Vehicles sector . . . . .                       | 8         |
| 1.2.1    | Trends in electric Light Duty Vehicles . . . . .             | 8         |
| 1.2.2    | Trends in electric Heavy Duty Vehicles . . . . .             | 9         |
| 1.3      | Electric Motor . . . . .                                     | 9         |
| 1.3.1    | PMSM motor drives . . . . .                                  | 9         |
| 1.3.2    | Inverter . . . . .   | 10        |
| 1.3.3    | PMSM control . . . . .                                       | 10        |
| 1.3.4    | Sensorless control technique . . . . .                       | 11        |
| 1.4      | Software used . . . . .                                      | 12        |
| <b>2</b> | <b>Permanent Magnet Synchronous Machine (PMSM)</b>           | <b>13</b> |
| 2.1      | Reference frames . . . . .                                   | 13        |
| 2.1.1    | $ABC$ Reference frame . . . . .                              | 13        |
| 2.1.2    | $\alpha\beta$ Reference frame . . . . .                      | 14        |
| 2.1.3    | $dq$ Reference frame . . . . .                               | 15        |
| 2.2      | Mathematic model of a SPM motor . . . . .                    | 17        |
| 2.2.1    | Three-phase balanced system model . . . . .                  | 18        |
| 2.2.2    | SPM motor model in $ABC$ reference frame . . . . .           | 19        |
| 2.2.3    | SPM motor model in $dq$ reference frame . . . . .            | 20        |
| 2.2.4    | SPM motor model in $\alpha\beta$ reference frame . . . . .   | 21        |
| 2.3      | PMSM Model . . . . .   | 22        |
| 2.3.1    | General PMSM In $ABC$ reference frame . . . . .              | 22        |
| 2.3.2    | Inductance matrix in $ABC$ reference frame . . . . .         | 23        |
| 2.3.3    | Inductance matrix in $dq$ reference frame . . . . .          | 27        |
| 2.3.4    | Inductance matrix in other notations . . . . .               | 27        |
| 2.3.5    | Inductance matrix in $\alpha\beta$ Reference frame . . . . . | 28        |
| 2.3.6    | General PMSM In $\alpha\beta$ reference frame . . . . .      | 29        |
| 2.3.7    | General PMSM in $dq$ reference frame . . . . .               | 29        |
| 2.3.8    | General SPM in $dq$ reference frame . . . . .                | 30        |
| <b>3</b> | <b>Three-phase inverter</b>                                  | <b>31</b> |
| 3.1      | Control Algorithm . . . . .                                  | 31        |
| 3.1.1    | FOC implementation . . . . .                                 | 33        |
| 3.2      | PWM generator - modulation techniques . . . . .              | 34        |
| 3.3      | Rotor angle observers . . . . .                              | 35        |
| 3.3.1    | Sliding mode observer method . . . . .                       | 35        |

|          |   |           |
|----------|---|-----------|
| 3.3.2    | Luenberger observer . . . . .                               | 41        |
| 3.3.3    | Kalman observer . . . . .                                   | 42        |
| 3.3.4    | Kalman Covariance Matrix optimization . . . . .             | 49        |
| <b>4</b> | <b>Simulation Model</b>                                     | <b>53</b> |
| 4.1      | Simplified Simulation Model . . . . .                       | 53        |
| 4.1.1    | Simplified motor parameters . . . . .                       | 53        |
| 4.1.2    | Torque-Power maps . . . . .                                 | 53        |
| 4.1.3    | Motor simulation . . . . .                                  | 54        |
| 4.1.4    | Motor Control . . . . .                                     | 55        |
| 4.1.5    | Inverter and battery . . . . .                              | 56        |
| 4.2      | Three-phase inverter full-replica Switching model . . . . . | 56        |
| 4.2.1    | Motor parameters . . . . .                                  | 57        |
| 4.2.2    | Motor simulation . . . . .                                  | 57        |
| 4.2.3    | Inverter simulation . . . . .                               | 57        |
| 4.2.4    | Switching loss . . . . .                                    | 58        |
| 4.2.5    | Motor Control . . . . .                                     | 58        |
| 4.2.6    | Battery . . . . .   | 59        |
| <b>5</b> | <b>Real Hardware test bench</b>                             | <b>60</b> |
| 5.1      | Hardware . . . . .  | 60        |
| 5.1.1    | Three-phase SPM Motor . . . . .                             | 60        |
| 5.1.2    | DC motor . . . . .  | 61        |
| 5.1.3    | Load . . . . .  | 62        |
| 5.1.4    | Control HW . . . . .  | 63        |
| 5.1.5    | Power supply and measurements . . . . .                     | 65        |
| 5.2      | Software . . . . .  | 65        |
| 5.2.1    | Control algorithm . . . . .                                 | 66        |
| <b>6</b> | <b>Results and Conclusions</b>                              | <b>68</b> |
| 6.1      | Motor 1: Simplified simulation . . . . .                    | 68        |
| 6.1.1    | Parameters . . . . .  | 69        |
| 6.1.2    | Observer algorithm steady state evaluation . . . . .        | 70        |
| 6.1.3    | Observer algorithm dynamic evaluation . . . . .             | 70        |
| 6.2      | Motor 2: Simplified simulation . . . . .                    | 75        |
| 6.2.1    | Parameters . . . . .  | 75        |
| 6.2.2    | Observer algorithm steady state evaluation . . . . .        | 76        |
| 6.2.3    | Observer algorithm dynamic state evaluation . . . . .       | 77        |
| 6.3      | Motor 1: Full-replica simulation . . . . .                  | 81        |
| 6.3.1    | Parameters . . . . .  | 81        |
| 6.3.2    | Observer algorithm steady state evaluation . . . . .        | 82        |

---

|                   |  |            |
|-------------------|--|------------|
| 6.3.3             | Observer algorithm dynamic evaluation . . . . .                    | 83         |
| 6.4               | Motor 2: Full-replica simulation . . . . .                         | 86         |
| 6.4.1             | Parameters . . . . .   | 86         |
| 6.4.2             | Observer algorithm steady state evaluation . . . . .               | 87         |
| 6.4.3             | Observer algorithm dynamic evaluation . . . . .                    | 87         |
| 6.5               | Real Hardware test bench . . . . .                                 | 91         |
| 6.5.1             | Parameters . . . . .   | 91         |
| 6.5.2             | Observer algorithm steady state sensed . . . . .                   | 92         |
| 6.5.3             | Observer algorithm step change sensed . . . . .                    | 93         |
| 6.5.4             | Observer algorithm step change Sensorless . . . . .                | 93         |
| 6.5.5             | Observer algorithm positive speed step change Sensorless . . . . . | 94         |
| 6.6               | Conclusion . . . . .   | 95         |
| <b>Glossary</b>   |  | <b>97</b>  |
| <b>Appendix A</b> |  | <b>105</b> |
| <b>Appendix B</b> |  | <b>106</b> |
| <b>Appendix C</b> |  | <b>110</b> |
| <b>Appendix D</b> |  | <b>112</b> |

## Nomenclature

### Physics Constants

$c$  Speed of light in a vacuum

### E-Motor electromagnetic

$\omega_e$  Electrical speed [rad/s]

$\omega_m$  Mechanical speed [rad/s]

$\theta_e$  Electrical angle [rad]

$\theta_m$  Mechanical angle [rad]

$T_l$  Load torque [N m]

$T_m$  Mechanical torque [N m]

$\varphi_x$  Stator magnetic flux in  $x$  reference frame [Wb]

$e_x$  Stator back-EMF in  $x$  reference frame [V]

$I_x$  Stator phase current in  $x$  reference frame [A]

$V_x$  Stator phase voltage in  $x$  reference frame [V]

### E-Motor mechanical

$J_s$  Stator rotational inertia

$p$  Pole Pairs

### Physics nomenclature

$J$  Rotational Inertia [kgm<sup>2</sup>]

### Abstract

The increasing interest in the decarbonization process led to a rapidly growing trend of electrification strategies in the automotive industry. In particular, OEMs are pushing towards the development and production of efficient electric vehicles. Moreover, research on electric motors and their control are exploding in popularity.

The increase of computational power in embedded control hardware is allowing the development of new control algorithm, such as sensorless control strategy. Such control strategy allows the reduction of the number of sensors, which implies reduced costs and increased system reliability.

The thesis objective is to realize a sensorless control for high-performance automotive motors.

Several algorithms for rotor angle observers are implemented in the MATLAB and Simulink environment, with emphasis on the Kalman observer. One of the Kalman algorithms already available in the literature has been selected, implemented and benchmarked, with emphasis on its comparison with the Sliding Mode observer.

Different models characterized by increasing levels of complexity are simulated. A simplified synchronous motor with "constant parameters", controlled by an ideal inverter is first analyzed; followed by a complete model defined by real motor maps, and controlled by a switching inverter.

Finally, it was possible to test the developed algorithm on a real electric motor mounted on a test bench.

A wide range of different electric motors have been simulated, which led to an exhaustive review of the sensorless control algorithm. The final results underline the capability of the Kalman observer to effectively control the motor on a real test bench.

# 1 Introduction

Electrification in automotive field is a rapidly growing trend [1], highly supported by worldwide politics. Indeed, to support the transition towards electrification, the EU Council decided to ban Internal Combustion Engine (ICE) for new cars and vans by 2035 [2], incentivizing the production of efficient electric powertrains. Several projects were born to increase the engagement of industries and universities towards research on more efficient, greener and cost-effective solutions in Electric Vehicles (EV) development [3] [4].

One of the most discussed research topic in the EV field is the electric powertrain. Permanent Magnet Synchronous Machine (PMSM) are particularly appreciated for their high power density, and high efficiency [5] [6]. One critical part on the implementation of PMSM is the control technique: many advanced and optimized technique are implemented focusing on efficiency and reliability [7]. The high-performance control of PMSM depends on accurate rotor position information, this is why high performant and expensive mechanical sensors are used in automotive traction electric motors [8]. Nevertheless, using sensorless control is appreciated in PMSM drives due to cost reduction, system downsizing, and reliability enhancement [9] [10].

**Experience in HPE COXA** HPE COXA [11] is an innovative provider of engineering solutions, technology projects and products for automotive, motorsport, off-highway, automation and defense sectors. It is also active in the vehicle electrification with the design and realization of electric motors and power electronic devices.

The objective of the internship activity was the development of advanced control strategies for electric motors. I had the possibility to get in touch with experienced and talented engineers, building up a comprehensive know how on electric motors, power electronics and control engineering.

## 1.1 Thesis objective

The thesis objective consists of implementing and benchmarking sensorless control technique to assess its feasibility and reliability in high performance EV powertrain application. The developed control has been evaluated using benchmark with increasing level of complexity:

- Simplified simulation.
  - Simplified high performance motor (constant parameters).
  - Ideal inverter.

- Current and Speed control.
- Complete simulation.
  - Flux mapped high performance motor.
  - Switching inverter.
  - Only Current control.
- Real test bench.
  - 3-phase electric motor.
  - Constant load.
  - Real inverter, fully customizable.
  - Current and Speed control.

The dissertation is structured as follows:

- Section 1: Introduction and thesis activity description.
- Section 2: Comprehensive description of 3-phase motors in math model.
- Section 3: State of the art of 3-phase inverter control.
- Section 4: Simulation model description.
- Section 5: Real world test bench description.
- Section 6: Discussion of the results and conclusions.

**Workflow architecture** For each motor test bench, the following workflow was used:

1. **Motor parameters evaluation:** Finite Element Model analysis was used to evaluate, parametrize and build a practical motor model running in simulation.
2. **Motor control model:** the motor control technique are implemented, as well as the sensorless algorithm.
3. **Sensored control evaluation:** sensed control algorithm is deployed, evaluating its performances and robustness. Model-based design will be used.
4. **Sensorless control using observers in control loop:** observed rotor angle is inserted in control loop, testing the accuracy and robustness of the new control.

5. **Final evaluation and benchmark:** comprehensive tests will be made, comparing different sensorless control algorithm in different condition and different motor.

## 1.2 The Electric Vehicles sector

Interest in EV is a continuous growing trend in academics and industries [12] [13], which is also part of the decarbonization process required by worldwide political institutions [2].

EVs can be subdivided by their application sector:

- Light Duty Vehicles (LDV): mobile machine that is primarily used to transport passengers and cargo [14].
- Heavy Duty Vehicles (HDV): comprise trucks, buses, and coaches. HDVs are defined as freight vehicles of more than 3.5 tonnes (trucks) or passenger transport vehicles of more than 8 seats (buses and coaches) [15].

Each EV can be classified according to the geometry its of electric propulsion system [16].

- Battery Electric Vehicles (BEV): run entirely on a battery-powered electric drivetrain.
- Hybrid Electric Vehicles (HEV): they are characterized by both an ICE and an electric motor. Different configuration of hybrid can be arranged, with pros and cons.
- Plug-in Hybrid Electric Vehicles (PHEV): they are subcategories of HEVs, in which the battery can be recharged through the electric grid (plug-in). PHEVs have both engine and electric motor.
- Fuel Cell Electric Vehicles (FCEV): they run entirely on a battery-powered electric drivetrain. The difference with BEV is the energy, which is stored using fuel cell technology.

### 1.2.1 Trends in electric Light Duty Vehicles

EVs sales for LDV is a continuous growing trend: a new sales record was set in 2021 despite the Covid-19 pandemic and supply chain challenges (16.5 million) [12]. EVs sales accounted for 9% of the global car market in 2021. During such year, EV sales doubled in China with respect to 2020, accounting for 82% of new electric car sales (3.3 million). Generally talking, european EV sales were not as impressive as chinese one; despite this, european EV market penetration is the highest, accounting for 17% of Europe's auto sales in 2021 [12].

### 1.2.2 Trends in electric Heavy Duty Vehicles

Electric HDVs are increasing in popularity too; in particular in 2021 electric bus accounts for 4% of the global fleet for buses [12].

## 1.3 Electric Motor

Electric motors are machines that convert electrical energy into mechanical energy, typically in the form of a rotating shaft. These types of motors are typically divided in two categories:

- Synchronous Electric Motor.
- Asynchronous Electric Motor.

Synchronous motors are the most popular in EV applications [6]. They are machines whose rotor excitation flux and stator magnetic field rotates at the same speed. The thesis will focus on the application of such motors.

Synchronous motors can also be divided in different categories, but one of the most popular is the Permanent Magnet Synchronous Machine, in which the excitation flux is generated by permanent magnets present in the rotor.

### 1.3.1 PMSM motor drives

As already mentioned, PMSM motors are the most attractive propulsion in EV applications [5] [6]. Different structures are being used in EV powertrains:

- Radial flux machine.
- Transverse flux machine.
- Axial flux machine.
- Doubly salient permanent magnet machine.
- Dual mechanical port electrical machine.

In the thesis, radial flux machines are taken in account, which are the most popular in EV powertrain application.

The standard configuration of a PMSM consists in two essential elements: the stator generating the electromotive force, and the rotor generating the magnetic field.

The power supply of a PMSM is usually a single phase or multiphase sinusoidal voltage generator, but the standard of this type of machines is 3-phase.

### 1.3.2 Inverter

Usually, electric LDVs gets powered by a battery pack, which is a DC voltage source; indeed an DC-AC converter is needed to convert the constant voltage available from the battery pack to an appropriate variable frequency voltage.

Those type of electric converters are also called inverters. Their role is to convert the direct current available from the battery pack to variable frequency alternating current [17].

In Fig. 1.1, a simplified representation of a three-phase traction inverter for traction application is shown.

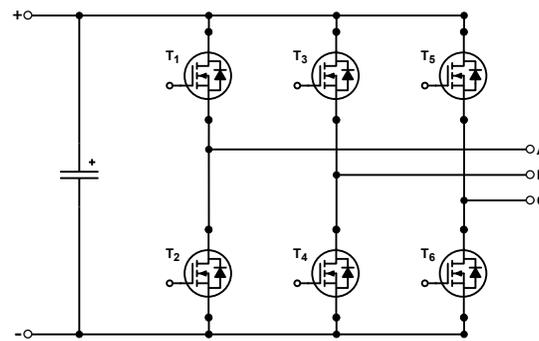


Figure 1.1: Standard three-phase voltage source inverter topology.

Different and more complex circuits can be used in traction inverters, exploiting multiphase and multilevel capabilities [18].

The inverter is responsible for the synchronous motor control, and it provides safety features too.

### 1.3.3 PMSM control

Different control techniques can be used to control PMSMs, which can be divided in scalar and vector strategies [19].

Scalar techniques are:

- $V/f$  control [20] [21].
- $I/f$  control [22] [23].

Those are open loop techniques (no speed feedback), whose principles are valid only for stationary regimes. Those are useful for the startup phase, but can not be used for high dynamic control [23].

Vector control techniques are:

- Field Oriented Control (FOC).

- Direct Torque Control (DTC).
- Direct Self Control (DSC).

Vector control techniques are aware of the stator and rotor flux vector, making them more stable and performant. The drawback of those techniques are the complexity of the control; nevertheless, vector control techniques are the only option in case high dynamics are requested.

The two most used vector control are Field Oriented Control (FOC) and Direct Torque Control (DTC). A comparison between those two techniques is shown in Tab. 1.1.

|  | <b>Field Oriented Control</b>           | <b>Direct Torque Control</b>     |
|--|---|----------------------------------|
| Dynamic response for torque                                | Slower                                  | Faster                           |
| Steady-state behavior for torque, stator flux and currents | Lower ripple and distortion             | Higher ripple and distortion     |
| Parameter sensitivity                                      | $L_d L_q \varphi_e$                     | $R_s$                            |
| Controllers  | Linear PI                               | Hysteresis control               |
| Implementation Complexity                                  | High                                    | Medium                           |
| Sensed variables   | Rotor mechanical speed, Stator currents | Stator voltages, Stator currents |

Table 1.1: Comparison of FOC and DTC for a PMSM [24].

The main advantage of DTC is to not utilize a rotor mechanical speed sensor. FOC is considered is the most suitable solution in automotive applications.

#### 1.3.4 Sensorless control technique

For sensorless control technique of PMSM, it is intended a control scheme which does not use a rotor angle position sensor. There are two main categories of sensorless control: model based (high speed) and saliency based (low speed) techniques [9].

#### 1.3.4.1 Saliency based

Saliency based sensorless control are often used for low speed. Natural saliency of the PMSM is used to detect the position of the rotor, often by means of an injected high frequency signal to the windings. Usually this kind of control scheme are used to overcome to the bad performances of model based control schemes in low speed [9].

#### 1.3.4.2 Model based

PMSMs are often characterized by wide speed range, indeed model based sensorless control is usually the favorite choice over signal based. Model based algorithms can be divided in open loop and closed loop [9] [10]. The open loop methods obtain the rotor position information from the machine model and from direct sensing of the motor voltage terminals, without using an internal correction mechanism; meanwhile closed loop methods are based on observer schemes that use an internal correction mechanism [10]. The thesis is focused on this last type of sensorless control.

### 1.4 Software used

The main software used is Matlab [25], Simulink [26] and Plecs [27]. A PMSM with realistic characteristics is simulated, and different rotor angle observers have been implemented within these software.

## 2 Permanent Magnet Synchronous Machine (PMSM)

General information about the PMSM will be introduced.

The thesis focuses on a *3-phase star connected sinusoidal PMSM*. It is worth to take in consideration that the most used motor geometry in automotive is the Internal Permanent Magnet (IPM) motor. Despite this, Surface Permanent Magnet (SPM) motors are the most used in academics since their control is less complex than IPMs.

**Note:** Through the following chapter about the reference frame, the rotating vector  $\bar{I}$  has been used as an example, but those transformations can be used for any rotating vector.

### 2.1 Reference frames

Different reference frames are described in the following section.

#### 2.1.1 ABC Reference frame

From an electrical point of view, the PMSM can be described as a three-phase system: it is characterized by three sinusoidal voltage signals with a phase offset of  $120^\circ$ . Moreover, their frequency and amplitude is variable at the stator coils, crating a rotating magnetic filed in the airgap.

Fig. 2.1 shows those sinusoidal signals in time-domain. It is possible to build

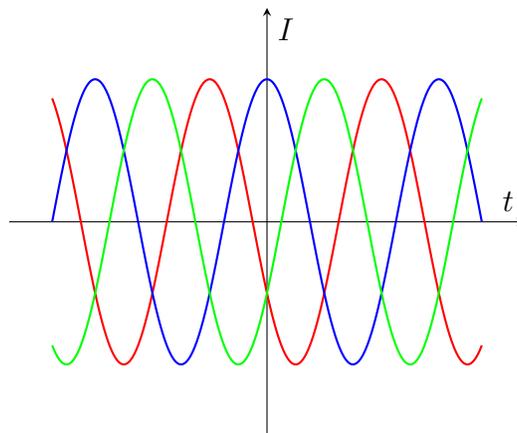
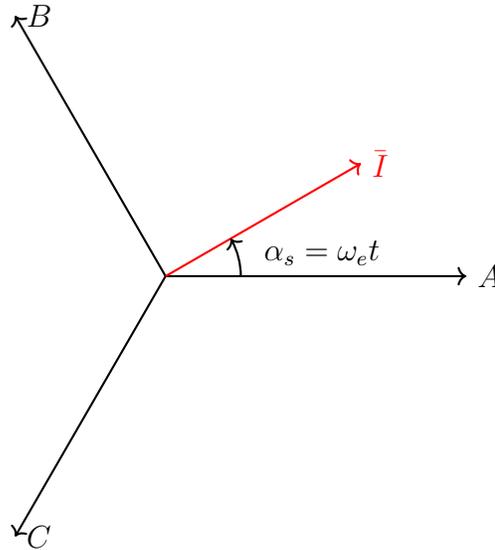


Figure 2.1: Phase currents in time.

a spatial vector reference frame, coherent with the position of the stator windings (Fig. 2.6).

Figure 2.2: Phase currents in  $ABC$ .

In this last reference frame, it is possible to represent the currents inside the rotor as the sum of the three-phase currents in the spacial reference frame  $ABC$  (Fig. 2.2). Such reference frame is fixed with the stator, therefore the current vector  $\bar{I}$  moves with a speed  $\omega_e$  (speed of the rotating magnetic filed). The three sinusoidal balanced current components are:

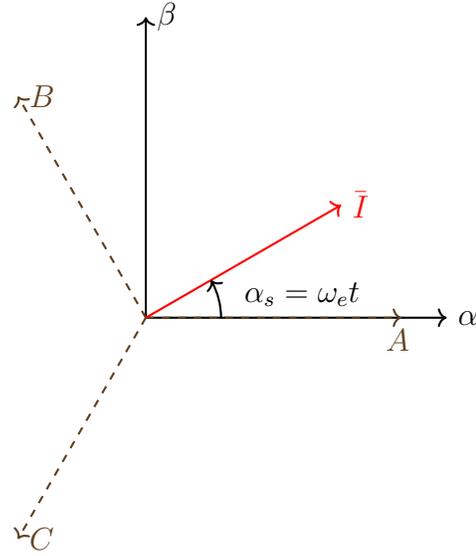
$$\begin{aligned} i_a &= |\bar{I}|\cos(\omega_e t) \\ i_b &= |\bar{I}|\cos(\omega_e t - \frac{2\pi}{3}) \\ i_c &= |\bar{I}|\cos(\omega_e t + \frac{2\pi}{3}) \end{aligned} \quad (2.1)$$

$i_a$ ,  $i_b$  and  $i_c$  are the three current components in the electric reference frame. Their sum in the electric reference frame is always zero, while in the  $ABC$  stator reference frame, the sum does result in the rotating vector  $\bar{I}$ .

### 2.1.2 $\alpha\beta$ Reference frame

Using Clarke Transform, it is possible to move from a three-axis reference frame, to a two axis reference frame (the third axis is redundant). The new current components are  $i_\alpha, i_\beta, i_0$ .

The new reference frame is sinusoidal with the stator as well, so the current will move inside the  $\alpha\beta$  reference frame with the speed  $\omega_e$ .

Figure 2.3: Phase currents in  $\alpha\beta$ .

If the  $ABC$  current components are known,  $\alpha\beta$  components are obtained by using the Clarke Transform.

$$\begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = K_{clarke} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (2.2)$$

Assuming the system is balanced,  $i_0 = 0$ . The inverse Clarke Transform is:

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} \quad (2.3)$$

The current vector can be represented by a phasor  $I$ , which significantly simplifies the mathematical notation.

$$\bar{I} = i_\alpha + j i_\beta = |\bar{I}| e^{j\alpha_s} = \frac{2}{3} \left( i_a + i_b e^{\frac{j2\pi}{3}} + i_c e^{-\frac{j2\pi}{3}} \right) \quad (2.4)$$

$\alpha_s$  is the angle of the stator current vector.

### 2.1.3 $dq$ Reference frame

The next step is to make the reference frame rotating with the same electric speed of the rotor. The motor torque control becomes simple to control the motor torque

since the current vector is not rotating anymore, and the torque is proportional to the  $q$  component of the current  $\bar{I}$ .

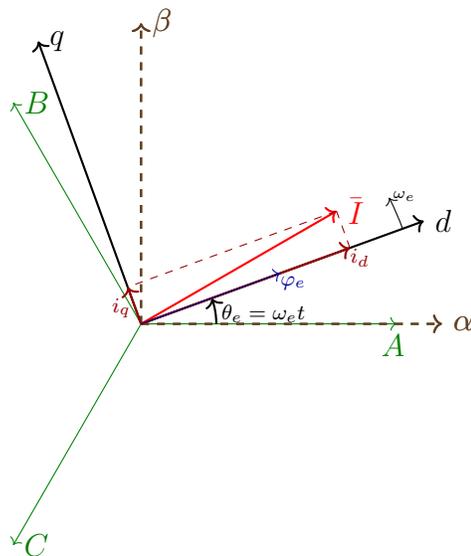


Figure 2.4: Phase currents in rotating reference frame  $dq$ .

$\theta_e$  is the angle between the  $A$  axis and the excitation flux  $\varphi_e$ . The resulting reference frame taken in consideration is shown in Fig. 2.5.

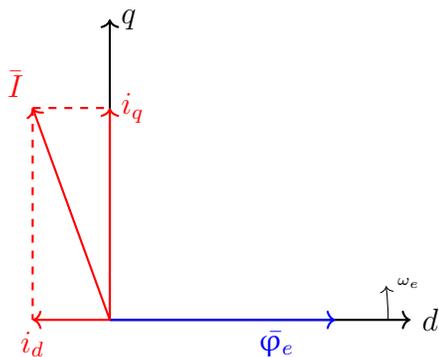


Figure 2.5: Phase currents in rotating reference frame  $dq$ .

The Park Transform can be applied to the  $ABC$  reference frame with the fol-

lowing rotation matrix.

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = K_{park} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta_e) & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin(\theta_e) & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (2.5)$$

The inverse Park transform, from abc reference frame is:

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) & 1 \\ \cos(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e - \frac{2\pi}{3}) & 1 \\ \cos(\theta_e + \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) & 1 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} \quad (2.6)$$

Simplifying the mathematical notation, the park transformation of a vector or matrix  $X$  is denoted as  $\text{park}(X)$ .

$\bar{I}$  is the phasor current on the  $\alpha\beta$  stator reference frame, and the phasor current  $\bar{I}$  in the  $dq$  rotor reference frame is:

$$\bar{I} = i_d + j i_q = \bar{I}' e^{-j\theta_e} \quad (2.7)$$

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (2.8)$$

**Note:** to apply the park transform, the rotating angle  $\theta_e$  is needed.

## 2.2 Mathematic model of a SPM motor

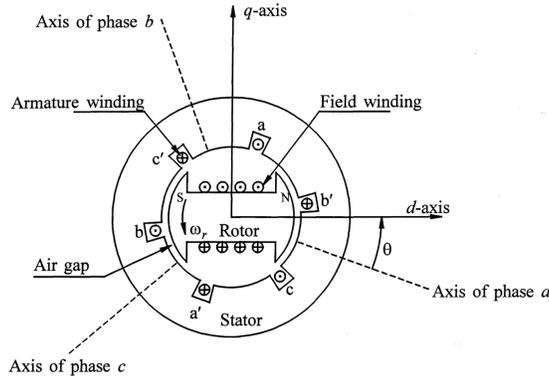


Figure 2.6: Schematic diagram of a three-phase synchronous machine [28].

Before introducing the control scheme implemented in the inverter, it is necessary to properly describe the mathematical model of a general PMSM. A first simplification can be done considering a 3-phase balanced system, that is similar to a SPM motor.

### 2.2.1 Three-phase balanced system model

The equations describing a 3-phase balanced system with inductive and resistive elements in the  $ABC$  reference frame are introduced [29, p. 90]:

$$\begin{cases} \bar{V}_{abc} = \bar{R}\bar{I}_{abc} + \dot{\bar{\Phi}}_{abc} \\ \bar{\Phi}_{abc} = \bar{L}\bar{I}_{abc} \end{cases} \quad (2.9)$$

where:

$$\bar{V}_{abc} = \overbrace{\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}}^{\text{Phase voltages}} ; \bar{R} = \overbrace{\begin{bmatrix} r_a & 0 & 0 \\ 0 & r_b & 0 \\ 0 & 0 & r_c \end{bmatrix}}^{\text{Phase resistance}} ; \bar{I}_{abc} = \overbrace{\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}}^{\text{Phase currents}} ; \bar{\Phi}_{abc} = \overbrace{\begin{bmatrix} \Phi_a \\ \Phi_b \\ \Phi_c \end{bmatrix}}^{\text{Phase linked flux}} ; \quad (2.10)$$

And the inductance matrix  $\bar{L}$ :

$$\bar{L} = \overbrace{\begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix}}^{\text{inductance matrix}} ; \quad (2.11)$$

The 3-phase system is assumed to be symmetrical (as in Fig. 2.7), so it is possible to introduce self inductance  $L_{self}$  and mutual inductance  $M$ . The inductance matrix becomes [29, p. 93]:

$$\bar{L} = \begin{bmatrix} L_{self} & M & M \\ M & L_{self} & M \\ M & M & L_{self} \end{bmatrix} \quad (2.12)$$

Applying the park transform already seen in Eq. (2.5):

$$K_{park} \bar{L} K_{park}^{-1} = \begin{bmatrix} L_{self} - M & 0 & 0 \\ 0 & L_{self} - M & 0 \\ 0 & 0 & L_{self} + 2M \end{bmatrix} \quad (2.13)$$

It is also possible to apply the Clarke transform seen in Eq. (2.2), and the result is the same:

$$K_{clarke} \bar{L} K_{clarke}^{-1} = \begin{bmatrix} L_{self} - M & 0 & 0 \\ 0 & L_{self} - M & 0 \\ 0 & 0 & L_{self} + 2M \end{bmatrix} \quad (2.14)$$

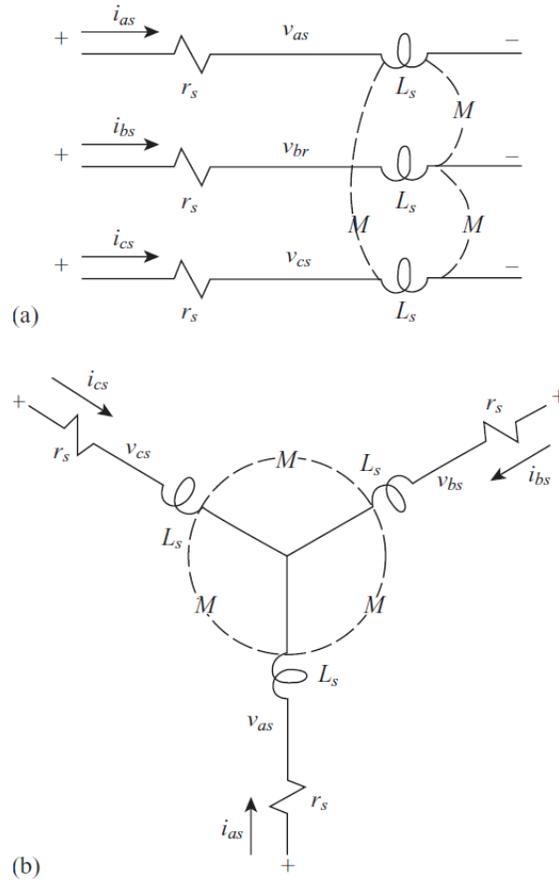


Figure 2.7: Three-phase RL circuit [29].

### 2.2.2 SPM motor model in $ABC$ reference frame

It is possible to model a SPM motor similarly to a 3-phase balanced system in the  $ABC$  reference frame [29, p. 93]:

$$\begin{cases} \bar{V}_{abc} = \bar{\mathbf{R}}\bar{\mathbf{I}}_{abc} + \dot{\bar{\Phi}}_{abc} \\ \bar{\Phi}_{abc} = \bar{\mathbf{L}}\bar{\mathbf{I}}_{abc} + \bar{\Phi}_e \end{cases} \quad (2.15)$$

$\bar{\Phi}_e$  Is the linked flux vector of the permanent magnet (excitation):

$$\bar{\Phi}_e = \begin{bmatrix} \Phi_{e,a} \\ \Phi_{e,b} \\ \Phi_{e,c} \end{bmatrix} = \begin{bmatrix} \Phi_e \cos(\theta_e) \\ \Phi_e \cos(\theta_e - \frac{2\pi}{3}) \\ \Phi_e \cos(\theta_e + \frac{2\pi}{3}) \end{bmatrix} \quad (2.16)$$

The inductance matrix is the same as the 3-phase system (Eq. (2.12)).  $L_{self}$  is split in leakage inductance  $L_{ls}$  and magnetizing inductance  $L_{ms}$ .  $L_{self}$  and  $M$  can

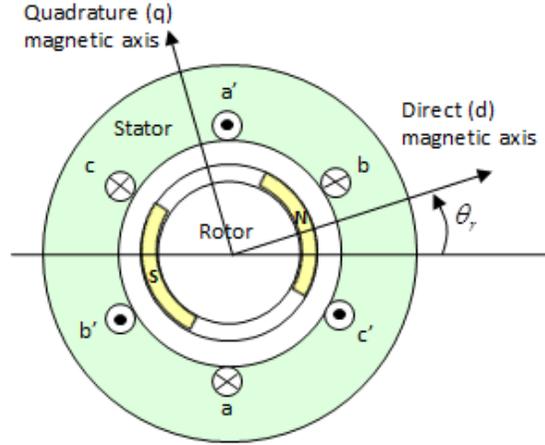


Figure 2.8: Schematic diagram of a three-phase SPM [26].

be defined as follows [28]:

$$\begin{aligned} L_{self} &= L_{ls} + L_{ms} \\ M &= -\frac{1}{2}L_{ms} \end{aligned} \quad (2.17)$$

The inductance matrix  $\bar{\mathbf{L}}$  becomes:

$$\bar{\mathbf{L}} = \begin{bmatrix} L_{self} & M & M \\ M & L_{self} & M \\ M & M & L_{self} \end{bmatrix} = \begin{bmatrix} L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} \end{bmatrix} \quad (2.18)$$

### 2.2.3 SPM motor model in $dq$ reference frame

Similarly to Eq. (2.14), the inductance matrix  $\bar{\mathbf{L}}$  in  $dq$  reference frame becomes:

$$L_{self} - M = L_{ls} + L_{ms} + \frac{1}{2}L_{ms} = L_{ls} + \frac{3}{2}L_{ms} \quad (2.19)$$

$$\text{park}(\bar{\mathbf{L}}) = \begin{bmatrix} L_{self} - M & 0 & 0 \\ 0 & L_{self} - M & 0 \\ 0 & 0 & L_{self} + 2M \end{bmatrix} = \begin{bmatrix} L_{ls} + \frac{3}{2}L_{ms} & 0 & 0 \\ 0 & L_{ls} + \frac{3}{2}L_{ms} & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \quad (2.20)$$

The synchronous inductance of the motor  $L_s$  is introduced:

$$L_s = L_{ls} + \frac{3}{2}L_{ms} \quad (2.21)$$

The mathematical model of the SPM in the dq reference frame is obtained.

$$\begin{cases} \bar{V}_{dq0} = \text{park}(\bar{\mathbf{R}})\bar{I}_{dq0} + \text{park}(\dot{\bar{\boldsymbol{\varphi}}}_{abc}) \\ \bar{\boldsymbol{\varphi}}_{dq0} = \text{park}(\bar{\mathbf{L}})\bar{I}_{dq0} + \text{park}(\bar{\boldsymbol{\varphi}}_e) \end{cases} \quad (2.22)$$

It is possible to demonstrate that:

$$\begin{aligned} \text{park}(\dot{\bar{\boldsymbol{\varphi}}}_{abc}) &= \dot{\bar{\boldsymbol{\varphi}}}_{dq0} + \omega_e[-\boldsymbol{\varphi}_q \ \boldsymbol{\varphi}_d \ 0]^\top \\ \text{park}(\bar{\boldsymbol{\varphi}}_e) &= [\boldsymbol{\varphi}_e \ 0 \ 0]^\top \end{aligned} \quad (2.23)$$

Assuming the current  $I_0 = 0$ , and fixing the  $d$  axis phase with the vector  $\boldsymbol{\varphi}_e$ :

$$\bar{V}_{dq0} = \begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix}; \quad \bar{I}_{dq0} = \begin{bmatrix} i_d \\ i_q \\ 0 \end{bmatrix}; \quad \bar{\boldsymbol{\varphi}}_{dq0} = \begin{bmatrix} \boldsymbol{\varphi}_d \\ \boldsymbol{\varphi}_q \\ \boldsymbol{\varphi}_0 \end{bmatrix}; \quad \bar{\boldsymbol{\varphi}}_{e,dq0} = \begin{bmatrix} \boldsymbol{\varphi}_e \\ 0 \\ 0 \end{bmatrix}; \quad (2.24)$$

$$\text{park}(\bar{\mathbf{R}}) = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix}; \quad \text{park}(\bar{\mathbf{L}}) = \begin{bmatrix} L_s & 0 & 0 \\ 0 & L_s & 0 \\ 0 & 0 & L_{ls} \end{bmatrix}; \quad (2.25)$$

Eq. (2.22) is expanded, and the SPM equations in  $dq$  becomes:

$$\begin{aligned} v_d &= R_s i_d + \frac{d\boldsymbol{\varphi}_d}{dt} - \omega_e \boldsymbol{\varphi}_q \\ v_q &= R_s i_q + \frac{d\boldsymbol{\varphi}_q}{dt} + \omega_e \boldsymbol{\varphi}_d \\ \boldsymbol{\varphi}_d &= L_s i_d + \boldsymbol{\varphi}_e \\ \boldsymbol{\varphi}_q &= L_s i_q \\ T_m &= \frac{3}{2} p \boldsymbol{\varphi}_e i_q \end{aligned} \quad (2.26)$$

#### 2.2.4 SPM motor model in $\alpha\beta$ reference frame

Similarly to subsection 2.2.3, the SPM equations in  $\alpha\beta$  are:

$$\begin{aligned} v_\alpha &= R_s i_\alpha + \frac{d\boldsymbol{\varphi}_\alpha}{dt} \\ v_\beta &= R_s i_\beta + \frac{d\boldsymbol{\varphi}_\beta}{dt} \\ \boldsymbol{\varphi}_\alpha &= L_s i_\alpha + \boldsymbol{\varphi}_e \cos(\theta_e) \\ \boldsymbol{\varphi}_\beta &= L_s i_\beta + \boldsymbol{\varphi}_e \sin(\theta_e) \end{aligned} \quad (2.27)$$

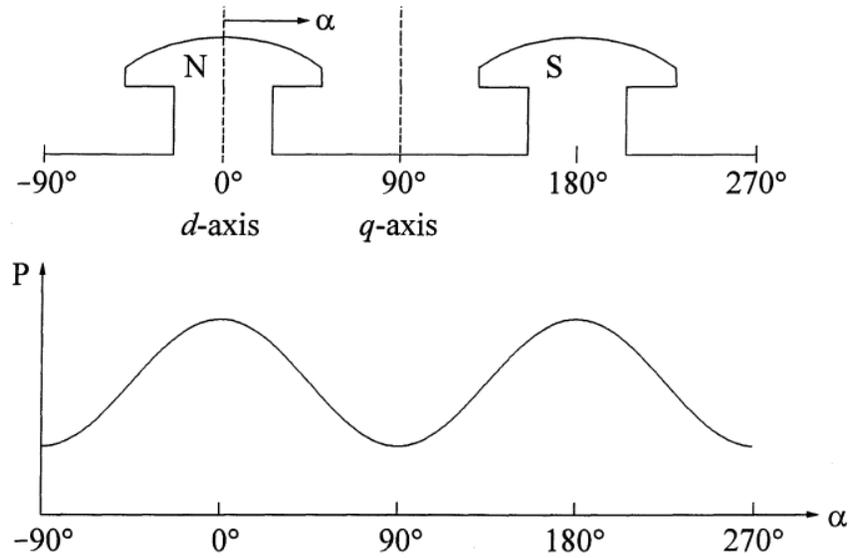


Figure 2.9: Variation of permeance along rotor position [28].

## 2.3 PMSM Model

### 2.3.1 General PMSM In $ABC$ reference frame

Taking in consideration a general PMSM motor, it is not possible to assume all the inductances to be the same, because the rotor is usually salient (permeance varies with the rotor position). It is noticeable that the flux linkage changes according to the rotor position [28].

Starting from the same motor equations seen in Eq. (2.22):

$$\begin{cases} \bar{V}_{abc} = \bar{R}\bar{I}_{abc} + \dot{\bar{\Phi}}_{abc} \\ \bar{\Phi}_{abc} = \bar{L}\bar{I}_{abc} + \bar{\Phi}_e \end{cases} \quad (2.28)$$

$\bar{L}$  matrix will be defined as follows:

$$\bar{L} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix}$$

## 2.3.2 Inductance matrix in $ABC$ reference frame

### 2.3.2.1 Self inductances in $ABC$

Each element on the  $\bar{\mathbf{L}}$  diagonal can be divided into two components: leakage inductance  $L_{ls}$  and Magnetizing inductance  $L_{gaa}$  (as was shown in Eq. (2.17), but with a different notation)

$$L_{aa} = L_{ls} + L_{gaa} \quad (2.29)$$

$L_{gaa}$  could also be named self inductance of phase  $a$  due to the airgap flux [28].

$$L_{gaa} = \frac{N_a \phi_{gaa}}{i_a} \quad (2.30)$$

$\phi_{gaa}$  is the total airgap flux linking phase  $a$ .

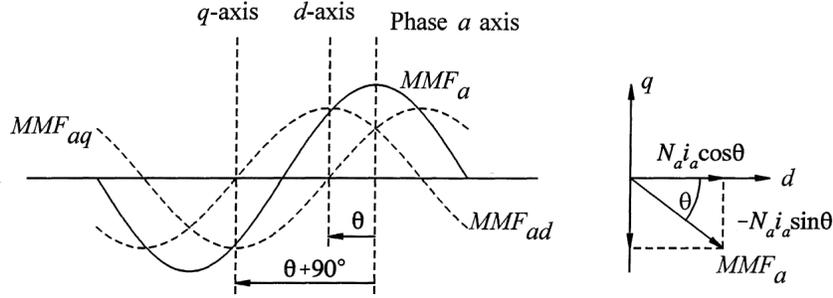


Figure 2.10: Phase  $a$  mmf wave and its components [28].

$\phi_{gaa}$  can be separated in two components ( $\phi_{gad}$  and  $\phi_{gaq}$ ) [28, p. 62]:

Permeance coefficients in  $d$  and  $q$  axis of a motor are introduced ( $P_d$  and  $P_q$ ); as a consequence, the  $MMF_\alpha$  is decomposed in two different sinusoids:  $MMF_{\alpha d}$  and  $MMF_{\alpha q}$  (where  $d$  and  $q$  are axis shifted by  $90^\circ$ , as shown in Fig. 2.10). Furthermore, the airgap fluxes per pole along the  $d$  and  $q$  axis ( $\phi_{gad}$  and  $\phi_{gaq}$ ) are:

$$\begin{aligned} \phi_{gad} &= \overbrace{(N_a i_a \cos(\theta_e))}^{\text{peakMMF}_{ad}} P_d \\ \phi_{gaq} &= \overbrace{(-N_a i_a \sin(\theta_e))}^{\text{peakMMF}_{aq}} P_q \end{aligned} \quad (2.31)$$

The total airgap flux linking phase  $a$

$$\begin{aligned} \phi_{gaa} &= \phi_{gad} \cos(\theta_e) - \phi_{gaq} \sin(\theta_e) = \\ &= N_a i_a (P_d \cos^2(\theta_e) + P_q \sin^2(\theta_e)) = \\ &= N_a i_a \left( \frac{P_d + P_q}{2} + \frac{P_d - P_q}{2} \cos(2\theta_e) \right) \end{aligned} \quad (2.32)$$

$L_{gaa}$  can be arranged as:

$$\begin{aligned}
L_{gaa} &= \frac{N_a \phi_{gaa}}{i_a} = \\
&= N_a^2 \left( \frac{P_d + P_q}{2} + \frac{P_d - P_q}{2} \cos(2\theta_e) \right) = \\
&= N_a^2 \frac{P_d + P_q}{2} + N_a^2 \frac{P_d - P_q}{2} \cos(2\theta_e) = \\
&= L_{ms} + L_m \cos(2\theta_e)
\end{aligned} \tag{2.33}$$

Recalling the notation:

$$\begin{aligned}
L_{ms} &= N_a^2 \frac{P_d + P_q}{2} \rightarrow \text{magnetizing inductance} \\
L_m &= N_a^2 \frac{P_d - P_q}{2} \rightarrow \text{stator inductance fluctuation} \\
L_{ls} &\rightarrow \text{leakage inductance} \\
L_{self} &= L_{ls} + L_{ms} \rightarrow \text{stator self-inductance per phase} \\
M_s &= \frac{1}{2} L_{ms} \rightarrow \text{stator mutual inductance}
\end{aligned} \tag{2.34}$$

It is possible to write  $L_{aa}$  as:

$$\begin{aligned}
L_{aa} &= L_{ls} + L_{gaa} = \\
&= L_{ls} + L_{ms} + L_m \cos(2\theta_e) = \\
&= L_{self} + L_m \cos(2\theta_e)
\end{aligned} \tag{2.35}$$

Finally,  $L_{aa}$  equation is expanded:

$$L_{aa} = L_{ls} + N_a^2 \frac{P_d + P_q}{2} + N_a^2 \frac{P_d - P_q}{2} \cos(2\theta_e) \tag{2.36}$$

### 2.3.2.2 Remaining diagonal inductances

Since the windings of phase  $b$  and  $c$  are identical to phase  $a$ , but displaced by  $-\frac{2\pi}{3}$  and  $\frac{2\pi}{3}$ , it is possible to write:

$$\begin{aligned}
L_{bb} &= L_{self} + L_m \cos\left(2\left(\theta_e - \frac{2\pi}{3}\right)\right) \\
L_{cc} &= L_{self} + L_m \cos\left(2\left(\theta_e + \frac{2\pi}{3}\right)\right)
\end{aligned} \tag{2.37}$$

### 2.3.2.3 particular case: $L_d \approx L_q$

If  $L_d \approx L_q$ , the inductance does not change with the angle ( $L_m = 0$ ), and it is possible to relate this case to the first element of the inductance matrix already seen in Eq. (2.18):

$$\begin{aligned} L_{aa} &= L_{ls} + N_a^2 \frac{P_d + P_q}{2} = L_{self} \\ &= L_{ls} + L_{ms} \end{aligned} \quad (2.38)$$

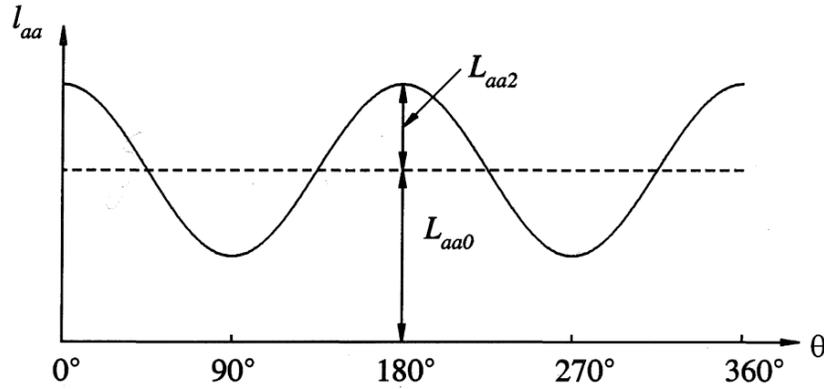


Figure 2.11: Variation of self inductance of a stator phase [28] ( $L_{aa0} = L_{self}$   $L_{aa2} = L_m$ ).

### 2.3.2.4 Mutual inductances in ABC

It is possible to compute the mutual inductance  $L_{ab}$  by evaluating the airgap flux  $\phi_{gba}$  linking phase  $b$  when only  $a$  is excited. The airgap linked flux is generated by  $a$ , so  $\phi_{gba}$  will have the same components seen in Eq. (2.31), but shifted by  $\frac{2\pi}{3}$ :

$$\begin{aligned} \phi_{gba} &= \phi_{gad} \cos\left(\theta_e - \frac{2\pi}{3}\right) - \phi_{gaq} \sin\left(\theta_e - \frac{2\pi}{3}\right) = \\ &= N_a i_a \left[ P_d \cos(\theta_e) \cos\left(\theta_e - \frac{2\pi}{3}\right) + P_q \sin(\theta_e) \sin\left(\theta_e - \frac{2\pi}{3}\right) \right] = \\ &= N_a i_a \left[ -\frac{P_d + P_q}{4} + \frac{P_d - P_q}{2} \cos\left(2\theta_e - \frac{2\pi}{3}\right) \right] \end{aligned} \quad (2.39)$$

It is now possible to introduce the mutual inductance between phases  $a$  and  $b$  due to the airgap flux  $L_{gba}$  (similar to  $L_{gaa}$ ):

$$\begin{aligned}
L_{gba} &= \frac{N_a \phi_{gba}}{i_a} = \\
&= N_a \left[ -\frac{P_d + P_q}{4} + \frac{P_d - P_q}{2} \cos(2\theta_e - \frac{2\pi}{3}) \right] \\
&= N_a \left( -\frac{P_d + P_q}{4} \right) + N_a \frac{P_d - P_q}{2} \cos(2\theta_e - \frac{2\pi}{3}) = \\
&= -\frac{1}{2} N_a \frac{P_d + P_q}{2} + N_a \frac{P_d - P_q}{2} \cos(2\theta_e - \frac{2\pi}{3}) \\
&= -\frac{1}{2} L_{ms} + L_m \cos(2\theta_e - \frac{2\pi}{3}) \\
&= -M_s + L_m \cos(2\theta_e - \frac{2\pi}{3})
\end{aligned} \tag{2.40}$$

The leakage inductance term in  $L_{ab}$  is not present because this is a mutual inductance.

$$\begin{aligned}
L_{ab} = L_{ba} = L_{gba} &= -M_s + L_m \cos(2\theta_e - \frac{2\pi}{3}) = \\
&= -M_s + L_m \cos(2(\theta_e - \frac{\pi}{3})) = \\
&= -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6}))
\end{aligned} \tag{2.41}$$

### 2.3.2.5 Remaining side inductances

Similarly to the self inductances, the remaining self inductances are then obtained:

$$\begin{aligned}
L_{bc} = L_{cb} &= -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6} - \frac{2\pi}{3})) \\
L_{ca} = L_{ac} &= -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6} + \frac{2\pi}{3}))
\end{aligned} \tag{2.42}$$

### 2.3.2.6 particular case $L_d \approx L_q$

Assuming  $L_m = 0$ , the mutual inductances become equal to the ones shown in the SPM equations (Eq. (2.18)):

$$L_{ab} = L_{ba} = \dots = L_{ac} = -\frac{1}{2} L_{ms} \tag{2.43}$$

In conclusion, the inductance matrix can be written with notation from Matlab as:

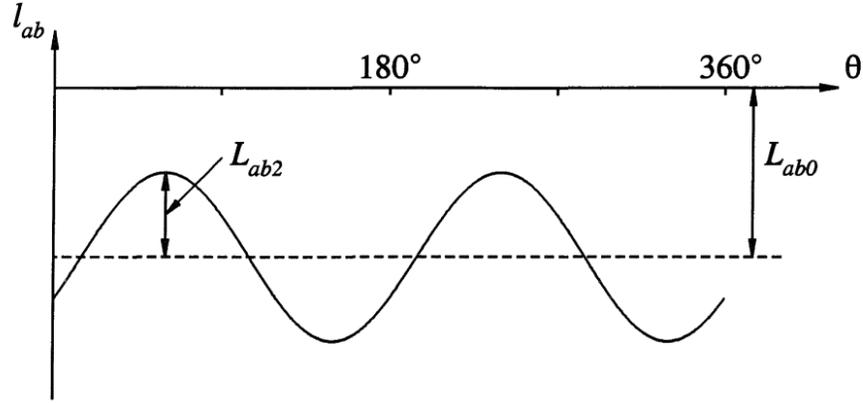


Figure 2.12: Variation of mutual inductance between stator windings [28] ( $L_{ab0} = \frac{L_{ms}}{2}$   $L_{ab2} = L_m$ ).

$$\bar{\mathbf{L}} = \begin{bmatrix} L_{self} + L_m \cos(2\theta_e) & -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6})) & -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6} + \frac{2\pi}{3})) \\ -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6})) & L_{self} + L_m \cos(2(\theta_e - \frac{2\pi}{3})) & -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6} - \frac{2\pi}{3})) \\ -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6} + \frac{2\pi}{3})) & -M_s - L_m \cos(2(\theta_e + \frac{\pi}{6} - \frac{2\pi}{3})) & L_{self} + L_m \cos(2(\theta_e + \frac{2\pi}{3})) \end{bmatrix} \quad (2.44)$$

### 2.3.3 Inductance matrix in $dq$ reference frame

Applying the park Transform to the new inductance matrix  $\bar{\mathbf{L}}$  matrix:

$$\text{park}(\bar{\mathbf{L}}) = \begin{bmatrix} L_{self} + M_s + \frac{3}{2}L_m & 0 & 0 \\ 0 & L_{self} + M_s - \frac{3}{2}L_m & 0 \\ 0 & 0 & L_{self} - 2M_s \end{bmatrix} = \begin{bmatrix} L_d & 0 & 0 \\ 0 & L_q & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \quad (2.45)$$

The equations of a general PMSM motor in  $dq$  are obtained by following the passages seen in Eq. (2.22).

### 2.3.4 Inductance matrix in other notations

It is possible to look at the PMSM inductance matrix with a different notation, as seen in other books [29, p. 122]:

$$\begin{aligned} L_{self} &= L_{ls} + L_{ms} \\ L_{ms} &= L_A \\ L_m &= L_B \end{aligned} \quad (2.46)$$

$$\bar{\mathbf{L}} = \begin{bmatrix} L_{ls} + L_A + L_B \cos(2\theta_e) & -\frac{1}{2}L_A + L_B \cos(2(\theta_e - \frac{\pi}{3})) & -\frac{1}{2}L_A + L_B \cos(2(\theta_e + \frac{\pi}{3})) \\ -\frac{1}{2}L_A + L_B \cos(2(\theta_e - \frac{\pi}{3})) & L_{ls} + L_A + L_B \cos(2(\theta_e - \frac{2\pi}{3})) & -\frac{1}{2}L_A + L_B \cos(2(\theta_e + \pi)) \\ -\frac{1}{2}L_A + L_B \cos(2(\theta_e + \frac{\pi}{3})) & -\frac{1}{2}L_A + L_B \cos(2(\theta_e + \pi)) & L_{ls} + L_A + L_B \cos(2(\theta_e + \frac{2\pi}{3})) \end{bmatrix} \quad (2.47)$$

### 2.3.4.1 dq reference frame (different notation)

Applying the Park transformation to the new  $\bar{\mathbf{L}}$  matrix in Eq. (2.47):

$$\text{park}(\bar{\mathbf{L}}) = \begin{bmatrix} L_{ls} + \frac{3}{2}(L_A + L_B) & 0 & 0 \\ 0 & L_{ls} + \frac{3}{2}(L_A - L_B) & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} = \begin{bmatrix} L_d & 0 & 0 \\ 0 & L_q & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \quad (2.48)$$

### 2.3.5 Inductance matrix in $\alpha\beta$ Reference frame

Using the notation from Eq. (2.47), the clarke transform applied to  $\bar{\mathbf{L}}$  becomes [30]:

$$\begin{aligned} \text{clarke}(\bar{\mathbf{L}}) &= \begin{bmatrix} L_{ls} + \frac{3}{2}(L_A + L_B \cos(2\theta_e)) & -\frac{3}{2}L_B \sin(2\theta_e) & 0 \\ -\frac{3}{2}L_B \sin(2\theta_e) & L_{ls} + \frac{3}{2}(L_A - L_B \cos(2\theta_e)) & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} = \\ &= \begin{bmatrix} L_\alpha & L_{\alpha\beta} & 0 \\ L_{\alpha\beta} & L_\beta & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \end{aligned} \quad (2.49)$$

By applying few mathematical considerations [30] [31],  $\text{clarke}(\bar{\mathbf{L}})$  is obtained using  $L_d$  and  $L_q$ :

$$\text{clarke}(\bar{\mathbf{L}}) = \begin{bmatrix} \Sigma L + \Delta L \cos(2\theta_e) & \Delta L \sin(2\theta_e) & 0 \\ \Delta L \sin(2\theta_e) & \Sigma L - \Delta L \cos(2\theta_e) & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \quad (2.50)$$

Where:

$$\begin{aligned} \Sigma L &= \frac{L_q + L_d}{2} \\ \Delta L &= \frac{L_d - L_q}{2} \end{aligned} \quad (2.51)$$

### 2.3.6 General PMSM In $\alpha\beta$ reference frame

The PMSM equation in  $\alpha\beta$  reference frame

$$\begin{cases} \bar{V}_{\alpha\beta 0} = \text{clarke}(\bar{\mathbf{R}})\bar{I}_{\alpha\beta 0} + \text{clarke}(\dot{\bar{\varphi}}_{abc}) \\ \bar{\varphi}_{\alpha\beta 0} = \text{clarke}(\bar{\mathbf{L}})\bar{I}_{\alpha\beta 0} + \text{clarke}(\bar{\varphi}_{e,abc}) \end{cases} \quad (2.52)$$

It is possible to demonstrate that:

$$\begin{aligned} \text{clarke}(\dot{\bar{\varphi}}_{abc}) &= \dot{\bar{\varphi}}_{\alpha\beta 0} \\ \text{clarke}(\bar{\varphi}_{e,abc}) &= \bar{\varphi}_{e,\alpha\beta 0} \end{aligned} \quad (2.53)$$

Then the expanded  $\alpha\beta$  motor equation are:

$$\begin{cases} v_\alpha = R_s i_\alpha + \frac{d\varphi_\alpha}{dt} \\ v_\beta = R_s i_\beta + \frac{d\varphi_\beta}{dt} \\ \varphi_\alpha = (\Sigma L + \Delta L \cos(2\theta_e))i_\alpha + \Delta L \sin(2\theta_e)i_\beta + \varphi_e \cos(\theta_e) \\ \varphi_\beta = (\Sigma L - \Delta L \cos(2\theta_e))i_\beta + \Delta L \sin(2\theta_e)i_\alpha + \varphi_e \sin(\theta_e) \end{cases} \quad (2.54)$$

The equation Eq. (2.54) can be then written as:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} R_s + (\Sigma L + \Delta L \cos(2\theta_e)) \frac{d}{dt} & (\Delta L \sin(2\theta_e)) \frac{d}{dt} \\ (\Delta L \sin(2\theta_e)) \frac{d}{dt} & R_s + (\Sigma L - \Delta L \cos(2\theta_e)) \frac{d}{dt} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \begin{bmatrix} -\sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \omega_e \varphi_e \quad (2.55)$$

$$\begin{cases} v_\alpha = R_s i_\alpha + (\Sigma L + \Delta L \cos(2\theta_e)) \frac{di_\alpha}{dt} + (\Delta L \sin(2\theta_e)) \frac{di_\beta}{dt} - \sin(\theta_e) \omega_e \varphi_e \\ v_\beta = (\Delta L \sin(2\theta_e)) \frac{di_\alpha}{dt} + R_s i_\beta + (\Sigma L - \Delta L \cos(2\theta_e)) \frac{di_\beta}{dt} + \cos(\theta_e) \omega_e \varphi_e \end{cases} \quad (2.56)$$

### 2.3.7 General PMSM in $dq$ reference frame

By using the inductance values found in Eq. (2.47), the equations of a general PMSM motor are:

$$\begin{aligned} v_d &= R_s i_d + \frac{d\varphi_d}{dt} - \omega_e \varphi_q \\ v_q &= R_s i_q + \frac{d\varphi_q}{dt} + \omega_e \varphi_d \\ \varphi_d &= L_d i_d + \varphi_e \\ \varphi_q &= L_q i_q \\ T_m &= \frac{3}{2} p [\varphi_d i_q - \varphi_q i_d] = \frac{3}{2} p [\varphi_e i_q + (L_d - L_q) i_d i_q] \end{aligned} \quad (2.57)$$

### 2.3.8 General SPM in $dq$ reference frame

The SPM motor model is often used for its simplicity: It is known that those type of motors are not very *anisotropic*, so  $L_d \approx L_q \approx L_s$ .

- $L_s$ : synchronous inductance of the motor.

$$L_s = L_{l,ph} + \frac{3}{2}L_{ph} \quad (2.58)$$

- $L_{ph}$ : auto induction coefficient, considering all the coefficients equal:

$$L_{ph} = L_a = L_b = L_c \quad (2.59)$$

- $L_{l,ph}$ : leakage inductance of the rotor phase (assuming again it is the same for each phase)

Then the simplified mathematical model becomes:

$$\begin{aligned} v_d &= R_s i_d + \frac{d\varphi_d}{dt} - \omega_e \varphi_q \\ v_q &= R_s i_q + \frac{d\varphi_q}{dt} + \omega_e \varphi_d \\ \varphi_d &= L_s i_d + \varphi_e \\ \varphi_q &= L_s i_q \\ T_m &= \frac{3}{2} p \varphi_e i_q \end{aligned} \quad (2.60)$$

### 3 Three-phase inverter

The device in charge of controlling the PMSM is the inverter: it transforms the DC voltage of a battery pack, modulating it into a controlled three-phase AC voltage.

As already discussed, the most suitable control technique for EV application is the FOC [24].

In Fig. 3.1, a simplified block scheme of a sensed control algorithm for PMSM.

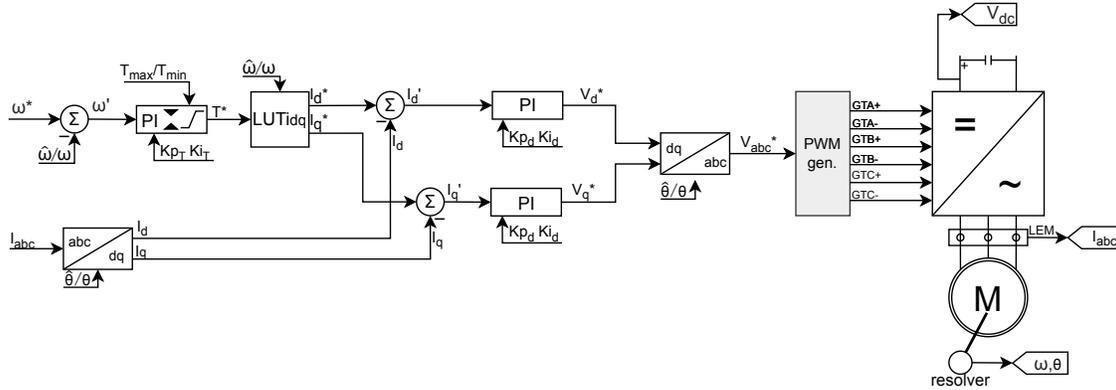


Figure 3.1: Simplified block scheme of sensed control algorithm.

#### 3.1 Control Algorithm

The main steps of the control can be described as follows:

1.  $\bar{I}_{abc}$  currents are sensed; as well as rotor position, angle and speed  $\omega_m \theta_m$ .
2.  $i_d$  and  $i_q$  are obtained from  $\bar{I}_{abc}$  via the Park transformation. The reference angle could be either sensed or estimated.
3. Torque reference  $T^*$  is calculated by means of a PI, tuned considering the desired dynamic performance of the motor.
4.  $i_d^*$  and  $i_q^*$  references are obtained by means of a Lookup Table (LUT), which has been calculated offline through a flux control strategy (such as MTPA and MTPV) [7].
5.  $v_d^*$  and  $v_q^*$  references are obtained by means of a current controller. This PI can be tuned with by using different methods [32].

6.  $\bar{V}_{abc}$  voltages are obtained via the inverse Park transformation, and a Pulse Width Modulation (PWM) generator outputs the right switching signals to be sent to the inverter board.
7. Three-phase sinusoidal voltage is injected to the motor.

### 3.1.0.1 MTPA and MTPV

Maximum Torque Per Ampere is a Flux control algorithm, which adjust the current components  $i_d$  and  $i_q$  to achieve the maximum torque possible with lower current. One simple way to implement that, is to control  $i_q$  accordingly to the torque setpoint, and  $i_d$  will be calculated as a function of  $i_q$  [7]:

$$i_d = \frac{\varphi_e}{2(L_q - L_d)} - \sqrt{\frac{\varphi_e^2}{4(L_q - L_d)^2} + i_q^2} \quad (3.1)$$

Exceeding a given speed, Maximum Torque Per Ampere (MTPA) will not be applicable because of voltage limitation. To increase the speed furthermore, it is possible to use flux weakening techniques [7].

Maximum Torque Per Volt is the most efficient way to implement flux weakening: its  $i_d$  and  $i_q$  references online calculation is not a straight forward task.

It is a common practice to calculate and build a LUT offline, which collects the MTPA and Maximum Torque Per Volt (MTPV) working points of the motor.

### 3.1.0.2 Current control tuning

Current controller PI needs to be tuned carefully, different tuning methods are available in literature [32].

In the thesis, Magnitude Optimum tuning method is implemented, which is a well known method for parametric optimization [33]. This method expects that the controlled system's transfer function has the following closed loop form:

$$G(s) = \frac{1}{2T_\mu^2 s^2 + 2T_\mu s + 1} \approx \frac{1}{2T_\mu s + 1} \quad (3.2)$$

Where  $T_\mu$  is the sum of all small delays in loop (much smaller than the time constant of the system). If the right  $K_p$  and  $K_i$  are used, it is possible to demonstrate that the current's closed loop transfer function has the form of Eq. (3.2) [33].

The proportional and integral gain of both  $i_d$  and  $i_q$  are obtained as [33]:

$$\begin{aligned} K_p &= \frac{L_s}{2K_{inv}K_iT_{inv}} \\ K_i &= \frac{R_s}{2K_{inv}K_iT_{inv}} \end{aligned} \quad (3.3)$$

Where

- $T_{inv} = T_\mu$  delay of the inverter.
- $K_{inv}$  amplification factor of the inverter.
- $K_i$  feedback coefficient.

### 3.1.1 FOC implementation

The control block diagram implemented in the model developed for the thesis is more complex, and can be seen in appendix A.

#### 3.1.1.1 Feed Forward

With Feed Forward (FF) the current control dynamic is improved by decoupling  $d$  and  $q$  axis currents. It is possible to demonstrate that using Eq. (3.4), and respectively  $v_d$  and  $v_q$  feed forward signals, the linearized transfer function of the current controller (Fig. 3.2) becomes as in Eq. (3.5) [7]

$$\begin{aligned} v_{d,ff} &= -\omega_e L_q i_q^* \\ v_{q,ff} &= -\omega_e L_d i_d^* + \omega_e \varphi_e \end{aligned} \quad (3.4)$$

$$\begin{aligned} G_d(s) &= \frac{1}{L_d s + R_s} \\ G_q(s) &= \frac{1}{L_q s + R_s} \end{aligned} \quad (3.5)$$

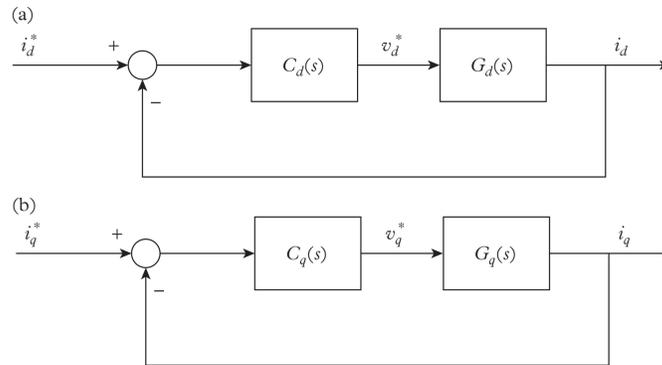


Figure 3.2: Block diagrams of the equivalent current linearized system: (a) d-axis block diagram. (b) q-axis block diagram. [7].

### 3.1.1.2 Flux weakening

Due to misalignment between the mapped and real motor parameters, the MTPV points might be imprecise, so motor control could not be able to run at the desired speed. To overcome this problem, an additional flux-weakening algorithm is implemented.

## 3.2 PWM generator - modulation techniques

PWM generator is used to convert  $\bar{V}_{abc}$  reference into PWM signals to be sent to the inverter switches. Different techniques can be applied, such as Sinusoidal PWM (SPWM) and Space Vector PWM (SVPWM) [34]. Sinusoidal PWM (SPWM) consists in creating three PWM signals approximating the sinusoidal reference  $\bar{V}_{abc}^*$ .

Modulation index is the relative amplitude of the output voltage compared to the maximum one (usually the letter used is  $m$ ). It is possible to increase the modulation index by a factor of  $2/\sqrt{3}$  using a technique called min-max injection.

Space Vector PWM (SVPWM) represents each switching state of the converter by a space vector in the Clarke referential. Then, the desired output voltage is synthesized on average, by alternating between multiple space vectors over each switching period [34].

It is possible to demonstrate that SVPWM and Sinusoidal PWM with min-max injection are equivalent.

### 3.2.0.1 Overmodulation

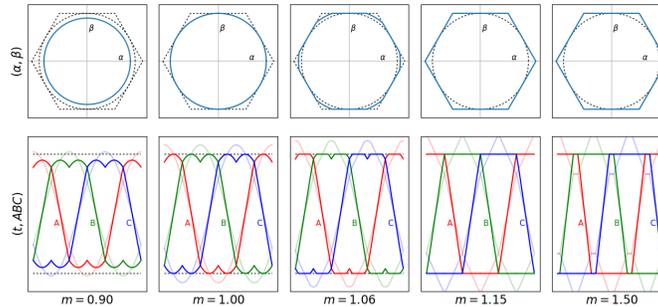


Figure 3.3: Overmodulation example [35].

Overmodulation does occur if the requested phase voltage is higher than the maximum modulation index. The advantage is to use the same motor in higher speed range, as well faster current control, thanks to the increased output voltage range [36] [35]. The drawback of this technique is the distorted output voltage,

deteriorating the performance of the current controller [36], as well as increasing losses on the inverter and motor. If overmodulation is met, a tradeoff needs to be met; for example, a good overmodulation range can be found between 1.05 and 1.25 [35].

### 3.3 Rotor angle observers

The following sections have the aim of clearly explaining the operation of different theta observers.

Every observer can be either used or not in the control loop.

#### 3.3.1 Sliding mode observer method

The Sliding Mode Observer (SMO) is commonly used within system observers for its tuning simplicity. SMOs is also often employed within synchronous motor applications [37].

##### 3.3.1.1 Math model

Assuming a simplified SPM isotropic machine in  $\alpha\beta$  reference frame, its model is expressed as the following equations:

$$\begin{aligned} v_\alpha &= R_s i_\alpha + \frac{d\varphi_\alpha}{dt} \\ v_\beta &= R_s i_\beta + \frac{d\varphi_\beta}{dt} \\ \varphi_\alpha &= L_s i_\alpha + \varphi_e \cos(\theta_e) \\ \varphi_\beta &= L_s i_\beta + \varphi_e \sin(\theta_e) \end{aligned} \quad (3.6)$$

By expanding the equations in Eq. (3.6):

$$\begin{aligned} v_\alpha &= R_s i_\alpha + L_s \frac{di_\alpha}{dt} - \varphi_e \sin(\theta_e) \omega_e \\ v_\beta &= R_s i_\beta + L_s \frac{di_\beta}{dt} + \varphi_e \cos(\theta_e) \omega_e \end{aligned} \quad (3.7)$$

Introducing the back-EMF  $e_{\alpha\beta}$ , it is possible to write Eq. (3.6) as:

$$\begin{aligned} v_\alpha &= R_s i_\alpha + L_s \frac{di_\alpha}{dt} + e_\alpha \\ v_\beta &= R_s i_\beta + L_s \frac{di_\beta}{dt} + e_\beta \\ e_\alpha &= -\varphi_e \omega_e \sin(\theta_e) \\ e_\beta &= \varphi_e \omega_e \cos(\theta_e) \end{aligned} \quad (3.8)$$

$R_s$ ,  $L_s$  and  $\varphi_e$  are assumed constant, then the equation can be rearranged as:

$$\begin{aligned}\frac{di_\alpha}{dt} &= -\frac{R_s}{L_s}i_\alpha + \frac{1}{L_s}v_\alpha - \frac{1}{L_s}e_\alpha \\ \frac{di_\beta}{dt} &= -\frac{R_s}{L_s}i_\beta + \frac{1}{L_s}v_\beta - \frac{1}{L_s}e_\beta \\ e_\alpha &= -\varphi_e\omega_e \sin(\theta_e) \\ e_\beta &= \varphi_e\omega_e \cos(\theta_e)\end{aligned}\quad (3.9)$$

The next assumption is that the two back EMFs are considered as *disturbances*.

$$\begin{aligned}\frac{de_\alpha}{dt} &= 0 \\ \frac{de_\beta}{dt} &= 0\end{aligned}\quad (3.10)$$

Finally, the system can be rewritten in matrix notation.

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{i}_\alpha \\ \dot{i}_\beta \\ \dot{e}_\alpha \\ \dot{e}_\beta \end{bmatrix}}_X = \underbrace{\begin{bmatrix} -\frac{R_s}{L_s} & 0 & -\frac{1}{L_s} & 0 \\ 0 & -\frac{R_s}{L_s} & 0 & -\frac{1}{L_s} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} i_\alpha \\ i_\beta \\ e_\alpha \\ e_\beta \end{bmatrix}}_X + \underbrace{\begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_B \underbrace{\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}}_U \quad (3.11)$$

In Eq. (3.11), the state space model of a IPM motor; from that model, it is possible to build a state observer.

### State observer

A state observer is a system that models a real system in order to provide an estimate of its internal state, given measurements of the input and output of the real system [38]. The observer algorithm works properly if the error between the real and the observed state tends to zero asymptotically  $\hat{X} \xrightarrow[t \rightarrow \infty]{} X$ . If  $e_\alpha$  and  $e_\beta$  are properly observed, it is possible to get the rotor angle from Eq. (3.9) as follows:

$$\theta_e = -\arctan\left(\frac{e_\alpha}{e_\beta}\right) \quad (3.12)$$

Different state observer are used in control theory, in this paragraph, the SMO is implemented.

### 3.3.1.2 SMO algorithm

The objective is to evaluate  $e_{\alpha\beta}$  without actually knowing the angle  $\theta_e$ , so the equations of  $I_{\alpha\beta}$  can be applied in Eq. (3.9): The estimation of  $\hat{I}_{\alpha\beta}$  could be provided by forcing it to "slide" towards the desired value  $I_{\alpha\beta}$ . If the error estimation tends to zero asymptotically,  $\hat{I}_{\alpha\beta} \approx I_{\alpha\beta}$ : it means that the model is approximating the real machine, so the  $\hat{e}_{\alpha\beta}$  obtained from the model are similar to the real ones ( $\hat{e}_{\alpha\beta} \approx e_{\alpha\beta}$ ). As it can be seen from Fig. 3.4, the algorithm acts like a *torque follower*.

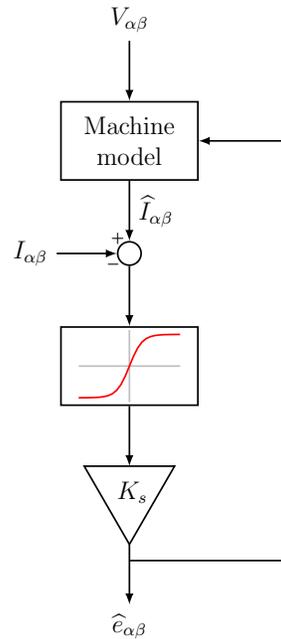


Figure 3.4: SMO algorithm.

1. Compute an estimation of the current  $\hat{I}_{\alpha\beta} = f(V_{\alpha\beta}, \hat{e}_{\alpha\beta})$  from the machine model Eq. (3.9).
2. Compute the estimation error:  $error = \hat{I}_{\alpha\beta} - I_{\alpha\beta}$ .
3. Use the error as the input of a sigmoid function and multiply it by a finite gain, in order to ensure the stability of the observer. The system is stable if the gain of the sigmoid function meets the condition  $K_s > \max(|e_\alpha|, |e_\beta|)$ .
4. The output signal will tend asymptotically to  $\hat{e}_{\alpha\beta}$ .

### 3.3.1.3 Machine Model for current estimation

The first part of the algorithm is needed to estimate the current  $\hat{I}_{\alpha\beta} = f(V_{\alpha\beta}, \hat{e}_{\alpha\beta})$ .

Starting from the matrix notation already discussed in Eq. (3.11), It is possible to apply the Euler method discretization:

$$\begin{cases} \dot{X}(t) = AX(t) + BU(t) \\ Y(t) = CX(t) + DU(t) \end{cases} \xrightarrow{\text{discretization}} \begin{cases} X(k+1) = (I + TA)X(k) + TBU(k) \\ Y(k) = CX(k) + DU(k) \end{cases} \quad (3.13)$$

By applying this discretization to Eq. (3.11)  $\hat{I}_{\alpha\beta}$  is computed:

$$\begin{cases} i_{\alpha}(k+1) = (1 + T(-\frac{R_s}{L_s}))i_{\alpha}(k) - T\frac{1}{L_s}e_{\alpha} + T\frac{1}{L_s}v_{\alpha} \\ i_{\beta}(k+1) = (1 + T(-\frac{R_s}{L_s}))i_{\beta}(k) - T\frac{1}{L_s}e_{\beta} + T\frac{1}{L_s}v_{\beta} \end{cases} \quad (3.14)$$

This computation is made using Simulink blocks Fig. 3.5. To have a stable algorithm,

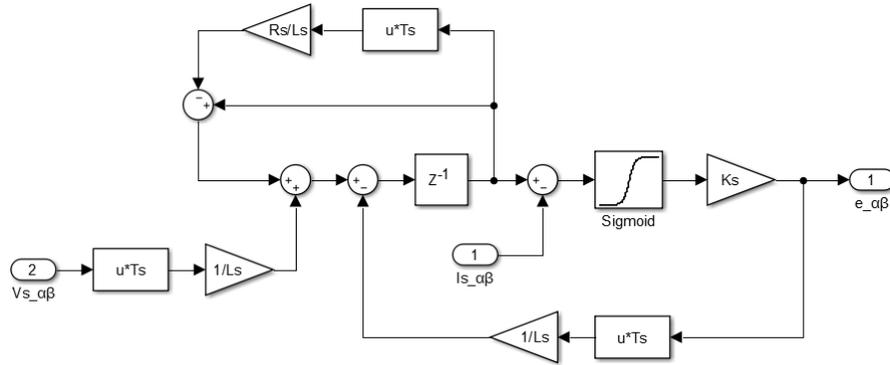


Figure 3.5:  $\hat{I}_{\alpha\beta}$  computation [37].

$K_s$  and the sigmoid constant needs to be fine-tuned.

### 3.3.1.4 Rotor angle estimation

The rotor angle  $\theta_e$  is estimated by using the back-EMF, as seen in Eq. (3.12). The arctan operator can be used, but it is very susceptible to noise (Eq. (3.12)).

A more stable solution is to use a Phase-Locked Loops (PLL). In algorithm implemented in this work [37], a Quadrature-component-based Phase-Locked Loop (Q-PLL) has been implemented.

**How Q-PLL Works** Phase-Locked Loops compares the phase of a reference signal to the phase of an adjustable feedback signal. When the comparison is in

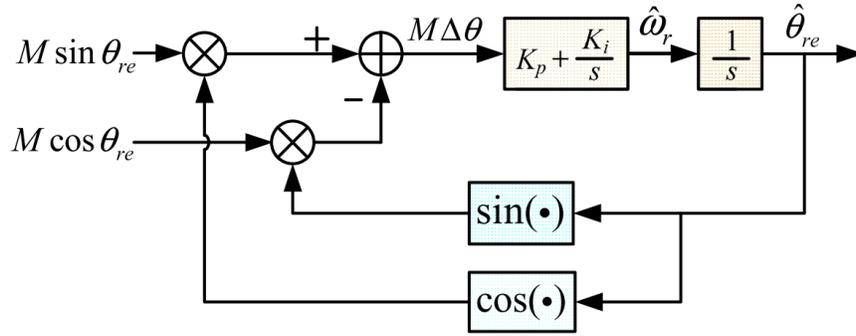


Figure 3.6: Q-PLL basic scheme [41].

steady-state, and the output frequency and phase are matched to the incoming frequency and phase of the error detector, the PLL is locked [39] [40].

In Fig. 3.6, the Q-PLL base structure, used for angle estimation.

A PID regulator estimates the speed from the following signal:

$$M \sin(\theta_e) \cos(\hat{\theta}_e) - M \cos(\theta_e) \sin(\hat{\theta}_e) = M \sin(\theta_e - \hat{\theta}_e) \approx M \Delta \theta_e \quad (3.15)$$

Then, integrating the speed signal  $\omega_e$ , the rotor angle is estimated.

**Q-PLL implemented** In the model implemented in this thesis, the reference speed is added as a disturbance, it can be seen in Fig. 3.7.

Another addition is the compensation network, that will compensate the angle error shift.

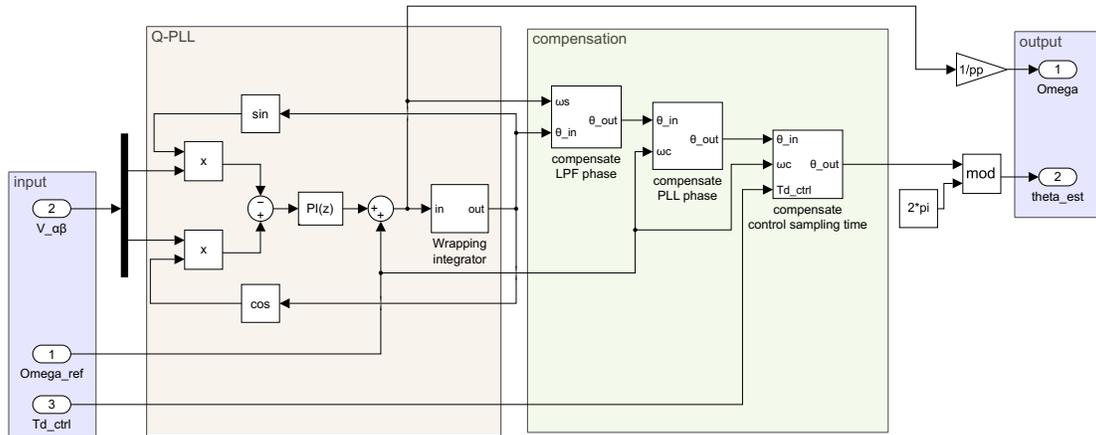


Figure 3.7: Simulink Q-PLL implemented.

### 3.3.1.5 Implementation note

Chattering can affect the output signal, it can be mitigated by using a Low Pass Filter, or tuning the sigmoid function. This second option is preferred because no delay is introduced.

A disadvantage of the SMOs that the model implemented is a simplified isotropic motor ( $L_d = L_q = L_s$ ), so for a general anisotropic motor it is necessary to approximate the inductances as:

$$L_s = \frac{L_d + L_q}{2} \tag{3.16}$$

### Chattering

Chattering occurs when a signal changes abruptly and frequently between two (approximate) values.

In Fig. 3.8 an example of chattering in sliding mode controller; Fig. 3.8a shows the desired behavior of the system: the control signal  $u(t)$  is a continuous signal, therefore the state space the output  $x(t)$  of the plant follows exactly the desired trajectory  $x_d(t)$ . In Fig. 3.8b the control signal of the sliding mode controller  $w(t)$  is discontinuous, and the actuator output  $w(t)$  lags behind: in this case the system trajectory will experience oscillatory behaviors, leading to a chattering phenomenon [42].

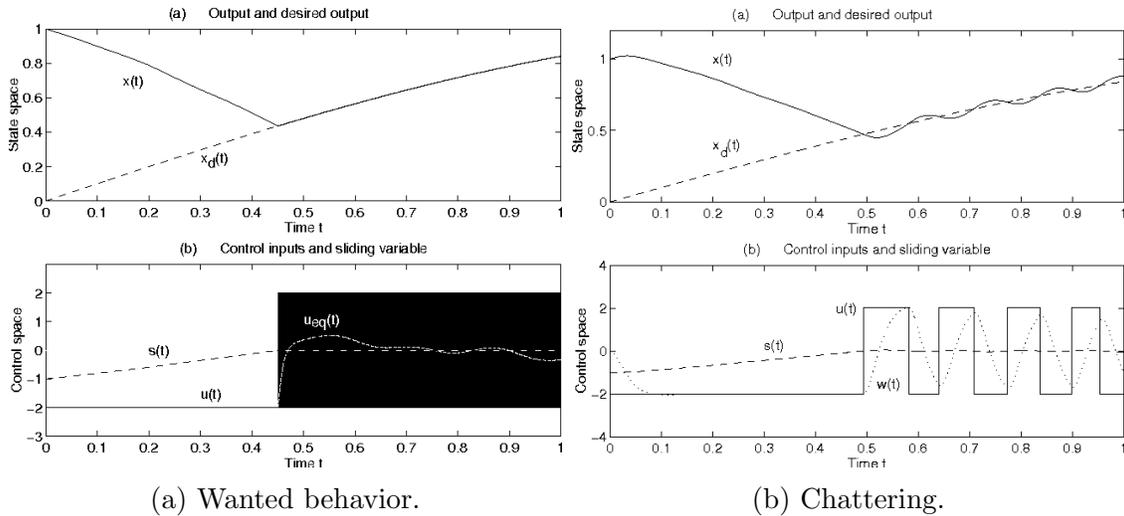


Figure 3.8: Example of chattering in a sliding mode controller [42].

### 3.3.2 Luenberger observer

Another possible solution to estimate the back-EMF is to use a Luenberger observer. An advantage over the SMO is that the algorithm is not affected by chattering, but the implementation of that is not straight forward.

An algorithm introduced from an application note was followed in this section [43].

#### 3.3.2.1 Math model

The first step is to build the state equation of the system to observe. It will have the following form:

$$\begin{cases} \dot{X} &= AX + BU \\ Y &= CX + DU \end{cases} \quad (3.17)$$

The matrix notation of the system from Eq. (3.11) is used as the state equation of the system. The output equation is the following:

$$\underbrace{\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}}_Y = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} i_\alpha \\ i_\beta \\ e_\alpha \\ e_\beta \end{bmatrix}}_X + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_D \underbrace{\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}}_U \quad (3.18)$$

It is now possible to build a Luenberger observer, which estimate the state vector asymptotically ( $\hat{X} \xrightarrow[t \rightarrow \infty]{} X$ ). The luenberger equation are introduced:

$$\begin{cases} \dot{\hat{X}} &= M_e \hat{X} + K_u U + K_y Y \\ \hat{Y} &= C \hat{X} + DU \end{cases} \quad (3.19)$$

Where  $M_e = A - K_y * C$ ,  $K_u = B - K_y * D$  and  $K_y$  is chosen. It is possible to demonstrate that the estimation error dynamic is:

$$e(t) = \dot{X} - \dot{\hat{X}} = M_e e(t) = (A - K_y * C)e(t) \quad (3.20)$$

To make the Luenberger observation converge asymptotically to the real value, it is necessary to find a  $K_y$  that will create  $eig(M_e) < 0$ . If this condition is met, the error dynamics tends to zero asymptotically.

In Fig. 3.9 the system in Simulink blocks, where  $\hat{X}$  is the state observed.  $e_\alpha$  and  $e_\beta$  are obtained from the state observation  $\hat{X}$ , so it is possible to estimate the rotor angle (as already done in subsection 3.3.1.4). A Q-PLL will be used.



Rewriting and expanding Eq. (3.21), the electric dynamic of the motor is obtained:

$$\begin{aligned} v_\alpha \cos(\theta_e) + v_\beta \sin(\theta_e) &= R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q \\ -v_\alpha \sin(\theta_e) + v_\beta \cos(\theta_e) &= R_s i_q + L_q \frac{di_q}{dt} + \omega_e L_d i_d + \overbrace{\omega_e \Phi_e}^{v_p} \end{aligned} \quad (3.23)$$

It is possible to introduce the equation of electrical speed  $\omega_e = p\omega_m$  and electrical angle  $\theta_e = p\theta_m$ , describing the mechanical dynamic of the motor:

$$\begin{aligned} \frac{d\omega_e}{dt} &= \frac{T_m - T_l}{J} p \approx 0 \\ \frac{d\theta_e}{dt} &= \omega_e \end{aligned} \quad (3.24)$$

In Eq. (3.24) it is assumed that the time step of the simulation is short, so  $\omega_e$  is constant along all the timestep ( $\frac{d\omega_e}{dt} = 0$ ).

$i_d$  and  $i_q$  were used as the state variables to be observed, along with the angle  $\theta_e$  and  $\omega_e$ .

$$\begin{cases} \frac{di_d}{dt} = -\frac{R_s}{L_d} i_d + \frac{L_q}{L_d} \omega_e i_q + \frac{\cos(\theta_e)}{L_d} v_\alpha + \frac{\sin(\theta_e)}{L_d} v_\beta \\ \frac{di_q}{dt} = -\frac{L_d}{L_q} \omega_e i_d - \frac{R_s}{L_q} i_q - \frac{\sin(\theta_e)}{L_q} v_\alpha + \frac{\cos(\theta_e)}{L_q} v_\beta - \frac{1}{L_q} v_p \\ \frac{d\omega_e}{dt} = 0 \\ \frac{d\theta_e}{dt} = \omega_e \end{cases} \quad (3.25)$$

Considering the input vector  $U = [v_\alpha v_\beta v_p]^T$ , and output vector  $Y = [i_\alpha i_\beta]^T$ , the obtained model in matrix notations can be written:

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{i}_d \\ \dot{i}_q \\ \dot{\omega}_e \\ \dot{\theta}_e \end{bmatrix}}_{\dot{X}} &= \underbrace{\begin{bmatrix} -\frac{R_s}{L_d} & \frac{L_q}{L_d} \omega_e & 0 & 0 \\ -\frac{L_d}{L_q} \omega_e & -\frac{R_s}{L_q} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} i_d \\ i_q \\ \omega_e \\ \theta_e \end{bmatrix}}_X + \underbrace{\begin{bmatrix} \frac{\cos(\theta_e)}{L_d} & \frac{\sin(\theta_e)}{L_d} & 0 \\ -\frac{\sin(\theta_e)}{L_q} & \frac{\cos(\theta_e)}{L_q} & -\frac{1}{L_q} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_B \underbrace{\begin{bmatrix} v_\alpha \\ v_\beta \\ v_p \end{bmatrix}}_U \end{aligned} \quad (3.26)$$

The output equation of the state space model is the inverse Park transform from  $i_\alpha i_\beta$  to  $i_d i_q$ .

$$\underbrace{\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}}_Y = \underbrace{\begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) & 0 & 0 \\ \sin(\theta_e) & \cos(\theta_e) & 0 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} i_d \\ i_q \\ \omega_e \\ \theta_e \end{bmatrix}}_X \quad (3.27)$$

The state space model of the system to be observed has the following form:

$$\begin{cases} \dot{X} &= A \cdot X + B \cdot U \\ Y &= C \cdot X \end{cases} \quad (3.28)$$

### 3.3.3.2 Discretization

The model is discretized using the *Rectangular method* (Euler's forward) [49]: it uses less computational power, and it is simple to implement. It is possible to increase its approximation precision with higher order Taylor expansions using the *Polynomial method* (Hanselmann) [49].

**First order Taylor expansion implementation** The model in Eq. (3.28) is discretized using the polynomial method:

$$\begin{cases} X(k+1) &= A_d \cdot X(k) + B_d \cdot U(k) \\ Y(k) &= C_d \cdot X(k) \end{cases} \quad (3.29)$$

Where  $A_d$ ,  $B_d$ ,  $C_d$  and  $D_d$  are the discretized matrices computed as follows:

$$\begin{aligned} A_d &= I + \sum_{l=1}^n \frac{T_d^l}{l!} A^l \approx I + T_d A + \left(\frac{T_d^2}{2} A^2\right) \approx I + T_d A \\ B_d &= T_d \left(I + \sum_{l=1}^{n-1} \frac{T_d^l}{l+1!} A^l\right) B \approx B T_d + A B \frac{T_d^2}{2} \approx B T_d \\ C_d &= C \\ D_d &= D \end{aligned} \quad (3.30)$$

$n$  is the desired Taylor order, and  $T_d$  the discrete time step.

The discretization of the electric motor model (seen in Eq. (3.26) and Eq. (3.27))

using first order Taylor expansions will be:

$$\begin{aligned}
\begin{bmatrix} i_d(k+1) \\ i_q(k+1) \\ \omega_e(k+1) \\ \theta_e(k+1) \end{bmatrix} &= \begin{bmatrix} 1 - \frac{R_s}{L_d} T_d & \frac{L_q}{L_d} \omega_e T_d & 0 & 0 \\ -\frac{L_d}{L_q} \omega_e T_d & 1 - \frac{R_s}{L_q} T_d & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & T_d & 1 \end{bmatrix} \begin{bmatrix} i_d(k) \\ i_q(k) \\ \omega_e(k) \\ \theta_e(k) \end{bmatrix} + \\
&+ \begin{bmatrix} \frac{\cos(\theta_e)}{L_d} T_d & \frac{\sin(\theta_e)}{L_d} T_d & 0 \\ -\frac{\sin(\theta_e)}{L_q} T_d & \frac{\cos(\theta_e)}{L_q} T_d & -\frac{1}{L_q} T_d \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_\alpha(k) \\ v_\beta(k) \\ v_p(k) \end{bmatrix} \quad (3.31) \\
\begin{bmatrix} i_\alpha(k) \\ i_\beta(k) \end{bmatrix} &= \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) & 0 & 0 \\ \sin(\theta_e) & \cos(\theta_e) & 0 & 0 \end{bmatrix} \begin{bmatrix} i_d(k) \\ i_q(k) \\ \omega_e(k) \\ \theta_e(k) \end{bmatrix}
\end{aligned}$$

### 3.3.3.3 Extended Kalman Filter

It is not possible to use the classic Kalman observer, because the two matrix  $A$  and  $B$  contain  $\omega_e$  and  $\varphi_e$ , that changes overtime, so an Extended Kalman Filter (EKF) will be needed.

EKF utilizes the state variable equations of the discrete time model of the general PMSM (Eq. (3.31)).

$$\begin{cases} \widehat{X}(k+1) &= A_d(k) \cdot \widehat{X}(k) + B_d(k) \cdot U(k) + v(k) \\ Y(k) &= C_d(k) \cdot \widehat{X}(k) + w(k) \end{cases} \quad (3.32)$$

Where:

- $v(k)$  **system noise vector**: zero mean Gaussian noise, independent with the initial state vector, and has covariance matrix is  $Q$ .
- $w(k)$  **measurement noise vector**: zero mean Gaussian noise, independent with the initial state vector, and has covariance matrix is  $R$ .
- $\widehat{X}$  **estimated state vector**.
- $K$  **Kalman gain matrix**.

### 3.3.3.4 EKF Algorithm

In Fig. 3.11 the main steps of the EKF algorithm. Entering in more details, the algorithm that will work as follows:

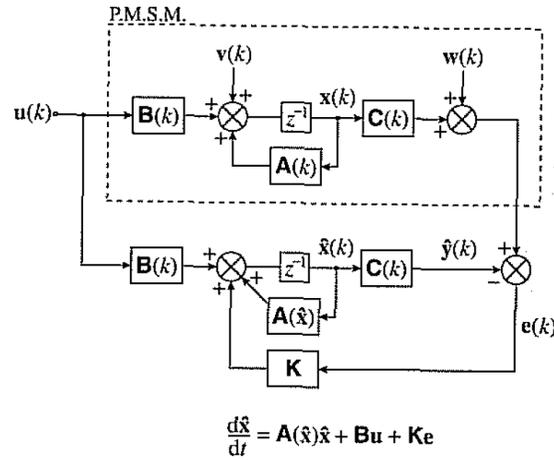


Figure 3.10: Structure of the EKF [47].

**1) Initialization** The first step is the initialization, here the noise and measurement covariance matrix are set:

$$\hat{\mathbf{X}}_0 = \begin{bmatrix} i_{d0} \\ i_{q0} \\ \omega_{e0} \\ \theta_{e0} \end{bmatrix}, P_0 = \text{zeros}(4, 4), Q = \text{diag}([q_1, q_2, q_3, q_4]), \quad (3.33)$$

The choice of  $R$  and  $Q$  will be discussed in subsection 3.3.3.5, but for now it is possible to tune those matrices taking in consideration 2 main rules:

1.  $R$  large  $\rightarrow K$  small  $\rightarrow$  faster transient.
2.  $Q$  large  $\rightarrow K$  large  $\rightarrow$  slower transient.

It is not possible to use  $Q$  too large or  $R$  too small, otherwise instabilities could happen.

**2) State vector prediction**  $A_d(k)$  and  $B_d(k)$  are computed using  $\theta_e$  and  $\omega_e$  estimation from the previous step. The first prediction of the state vector at time  $(k+1)$  is:

$$\hat{\mathbf{X}}(k+1|k) = \hat{\mathbf{X}}(k+1) = A_d(k)\hat{\mathbf{X}}(k) + B_d(k)U(k) \quad (3.34)$$

$\hat{\mathbf{X}}(k+1|k)$  is the predicted value at  $(k+1)$ -th instant, and it is based on measurements up to  $k$ -th instant. To simplify the notation,  $\hat{\mathbf{X}}(k+1)$  will be used instead of  $\hat{\mathbf{X}}(k+1|k)$ . Similarly,  $\hat{\mathbf{X}}(k|k)$  is replaced by  $\hat{\mathbf{X}}(k)$ .

### 3) Covariance estimation of prediction

$$P(k+1|k) = P(k+1) = F(k+1)P(k)F^T(k+1) + Q \quad (3.35)$$

State transition and observation matrices are defined to be the following Jacobians:

$$\begin{aligned} F(k+1) &= \left. \frac{\partial(A_d(k)X + B_d(k)U(k))}{\partial X} \right|_{X=\hat{X}(k+1)} = \\ &= \begin{bmatrix} 1 - \frac{R_s T_d}{L_d} & \frac{L_q T_d \omega_e}{L_d} & \frac{L_q T_d \dot{i}_q}{L_d} & \frac{\cos(\theta_e)}{L_d} T_d v_\beta - \frac{\sin(\theta_e)}{L_d} T_d v_\alpha \\ -\frac{L_d T_d \omega_e}{L_q} & 1 - \frac{R_s T_d}{L_q} & -\frac{T_d}{L_q} (L_d \dot{i}_d + \varphi_e) & -\frac{\cos(\theta_e)}{L_q} T_d v_\alpha - \frac{\sin(\theta_e)}{L_d} T_d v_\beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & T_d & 1 \end{bmatrix} \end{aligned} \quad (3.36)$$

$$\begin{aligned} H(k+1) &= \left. \frac{\partial(C_d(k)X)}{\partial X} \right|_{X=\hat{X}(k+1)} = \\ &= \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) & 0 & -i_q \cos(\theta_e) - i_d \sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) & 0 & i_d \cos(\theta_e) - i_q \sin(\theta_e) \end{bmatrix} \end{aligned} \quad (3.37)$$

$\theta_e$  and  $\omega_e$  are the estimated values at the previous iteration.

### 4) Kalman filter gain computation

$$K(k+1) = P(k+1)H^T(k+1)[H(k+1)P(k+1)H^T(k+1) + R]^{-1} \quad (3.38)$$

#### 5a) Output vector estimation

$$\hat{Y}(k+1) = C_d(k)\hat{X}(k+1) \quad (3.39)$$

#### 5b) State vector estimation

This is the corrected state vector estimation:

$$\hat{X}(k+1|k+1) = \hat{X}(k+1) + K(k+1)[Y(k+1) - \hat{Y}(k+1)] \quad (3.40)$$

**6) Covariance matrix of estimation error** The error covariance matrix can be obtained from:

$$P(k+1|K+1) = P(k+1) - K(k+1)H(k+1)P(k+1) \quad (3.41)$$

**7) Finalization** Save  $\hat{X}(k+1|k+1) = \hat{X}(k)$  and  $\hat{P}(k+1) = P(k)$ , and go back to step 2.

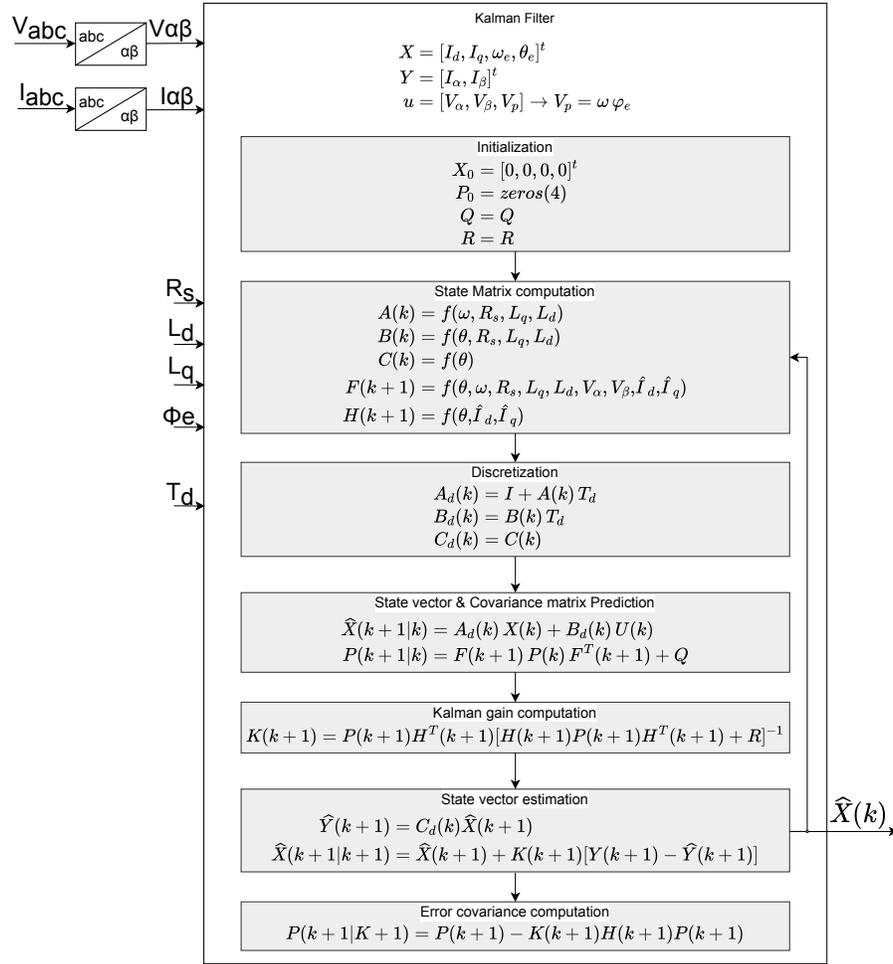


Figure 3.11: EKF block scheme.

### 3.3.3.5 Covariance Matrix $Q$ and $R$

The covariance matrices have an important effect on the filter stability and convergence time.  $Q$  and  $R$  have to be obtained by considering the stochastic properties of the corresponding noises; however, since these are usually not known (like in this case), the covariance matrices are used as weighting factors (tuning parameters) [50].

- System noise covariance  $Q$  accounts for the model inaccuracy, the system disturbances, and the noise introduced by the voltage measurements (sensor noise, A/D converter quantization).
- Measurement noise covariance  $R$  accounts for measurement noise introduced by the current sensors and A/D quantization [47].

In literature [47] it is suggested to choose the covariance matrices empirically via a trial and error approach. It is possible to use diagonal matrices because noise signals are not correlated each other:

- $Q = \text{diag}([q_1, q_2, q_3, q_4])$ .
- $R = \text{diag}([r_1, r_2])$ .

To increase the overall performance of the observer, the matrices can be selected using an optimization algorithm [50] [51]. More on that in subsection 3.3.4.

### 3.3.3.6 Further development

It is possible to enhance this model with load torque observation and better speed evaluation by taking into account additional information such as Rotor inertia [48, p 327].

### 3.3.3.7 EKF in microcontroller

Computational power is not a free resource, this is why it is crucial to understand the limits of real inverter hardware and find the right tradeoff of observer sample time. An effective way is to use the optimization techniques available for microcontroller and compiler [52] [53].

## 3.3.4 Kalman Covariance Matrix optimization

As already mentioned, a crucial task to make the EKF algorithm works is to find the right covariance matrices  $Q$  and  $R$ .

Manual tuning of the EKF using the trial and error method is simple to carry out, but the process is very time-consuming and satisfactory performance can only be obtained with great effort from an experienced operator [51].

In this work, a solution based on optimization algorithm is proposed: the trial and error would be automated by the software, and optimally, obtaining optimized  $Q$  and  $R$  matrices automatically. In academics, this approach was adopted on an EKF observer for a rotor angle observer on induction motors, using Genetic Algorithm [51] and Particle Swarm optimization [50].

This problem can be seen as a Nonlinear Programming Problem.

**Nonlinear Programming Problem [54]** A general optimization problem is to select  $n$  decision variables  $x_1, x_2, \dots, x_n$  from a given feasible region in such a way as to optimize (minimize or maximize) a given objective function  $f(x_1, x_2, \dots, x_n)$  of decision variables. The problem is called a nonlinear programming problem (NLP)

if the objective function is nonlinear and/or the feasible region is determined by nonlinear constraints.

In the thesis, the optimization problem of finding  $Q$  and  $R$  matrices has been solved by means of Sequential Quadratic Programming (SQP) [55] [56] with the help of `fmincon` function [57].

### 3.3.4.1 `fmincon`

The Matlab function `fmincon` implements a nonlinear programming solver [57]. Finds the minimum of a problem specified by:

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ c_{eq}(x) = 0 \\ A \cdot x \leq b \\ A_{eq} \cdot x = b_{eq} \\ l_b \leq 0 \leq u_b \end{cases} \quad (3.42)$$

Where the input of the function are:

- $f(x)$  Function to minimize.
- $x_0$  Initial point.
- $A - A_{eq}$  Linear inequality - equality constraints (matrix).
- $b - b_{eq}$  Linear inequality - equality constraints (vector).
- $c - c_{eq}$  Nonlinear inequality - equality constraints.
- $l_b - u_b$  Upper - lower bounds.

In the thesis, this function was used in "Minimize with Bound Constraints" mode; In this mode, `fmincon` is used to find the minimum of an objective function in the presence of bound constraints. ( $A, A_{eq}, b_{eq}, b, c, c_{eq} = []$ )

```
% Input
% fun      - Cost function
% x0      - Initial point
% Lb      - Lower bound
% Ub      - Upper bound
% options - Options

% Output
% x       - Optimized values
```

```
x = fmincon(fun,x0,[],[],[],[],lb,ub,[],options);
```

### 3.3.4.2 Cost function

The cost function to be minimized is chosen to be sum of the weighed and normalized angle and speed error in module, over time:

$$f_{cost}(q_1, q_2, q_3, q_4, r_1, r_2) = \sum |W_{\omega_e} \cdot e_{\omega_e}(t) + W_{\theta_e} \cdot e_{\theta_e}(t)| \quad (3.43)$$

The error vectors are calculated by arranging multiple test in a simulation environment, which replicates the behavior of an electric motor controlled by a switching inverter subsection 4.2.

The motor in test is dragged from its shaft to increase its speed with a sequence of ramps, reaching its maximum speed. The inverter is asked to output a requested electric torque, and the current control is sensed.

Because the EKF observer angle is not used in control, it is possible to save the algorithm input from the Simulink simulation, and run it offline in matlab environment: this will decrease considerably the time for the EKF to perform. Due to this, it is possible to run the EKF observer algorithm within seconds, instead of simulating the full behavior and control of the motor and the inverter.

1. Simulation i Simulink is run.
2.  $i_d, i_q, \omega_e, \theta_e$  and the motor parameters are saved as a signal in time.
3. The EKF can now be run offline, its output can be used to compute the cost function, and the matrices  $Q$  and  $R$  can be used as the parameters for the optimization.

An example of the output of a single EKF simulation can be found in Fig. 3.12

The input of the algorithm are the diagonal elements of the  $Q$  and  $R$  matrices  $q_1, q_2, q_3, q_4, r_1, r_2$ .

### 3.3.4.3 Results of the optimization

Optimized matrices were very effective using the same motor used in the optimization, but it was possible to use those with different motor by doing a fine-tuning via a quick trial and error approach.

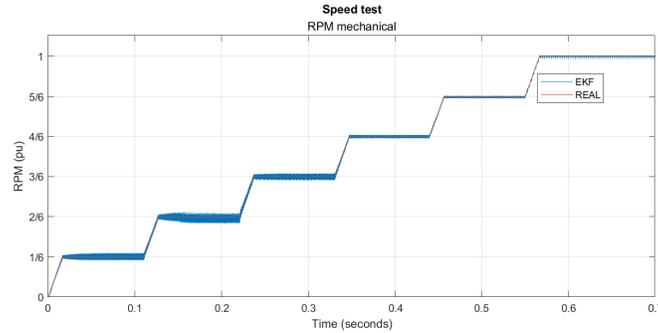


Figure 3.12: Example of test for EKF matrix optimization.

#### 3.3.4.4 Theta compensation

During sensorless control tests, unwanted behaviors were noticed, making the motor uncontrollable. Different solutions were tested, but the most effective was to introduce a small compensation  $C_{omp}$  on the angle output of the algorithm. The compensation factor is directly injected to the observed angle at the beginning of the algorithm, before matrix computation is performed. The electric angle at  $k$ -th instant, that will be used for the state prediction and matrix computation is:

$$\hat{\theta}_e(k|k)_{comp} = \hat{\theta}_e(k|k) + C_{omp} \cdot \hat{\omega}_e(k|k) \cdot T_d \quad (3.44)$$

## 4 Simulation Model

Two different simulation models were used, the first is implemented using a simplified "fixed parameters" motor controlled by an ideal inverter. The second and more complex model is implemented using real motor flux maps and a switching inverter.

This was made to provide a comprehensive benchmark, using a fast and more simple simulation for the initial implementation of the algorithm, and then taking into account more complex behavior with a full-replica model.

### 4.1 Simplified Simulation Model

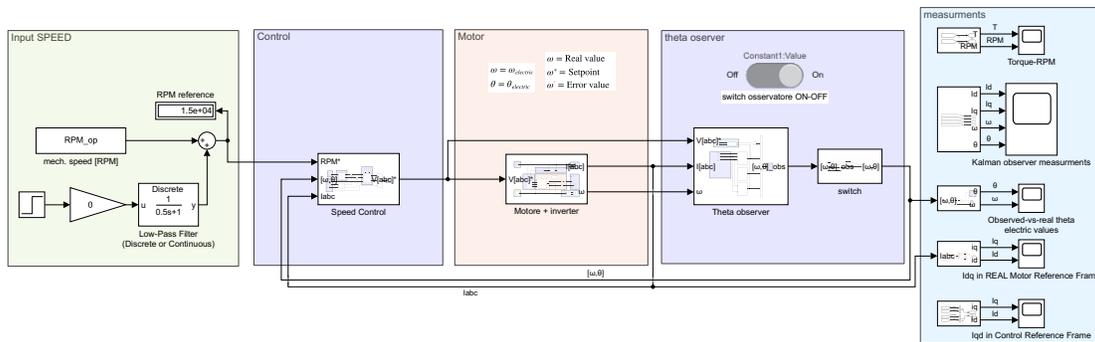


Figure 4.1: Overall Simulink motor model.

Matlab was used to extrapolate the parameters of a real motor tested in HPE COXA, then simulation environment was set up in Simulink.

The chosen simplified motor block is the *Permanent Magnet Synchronous Machine* from *Specialized Power Systems* blockset [58].

#### 4.1.1 Simplified motor parameters

A complete map of a motor simulated with Finite Element Model (FEM) analysis with variable parameters was available, but the chosen block only supports for fixed parameters.

By choosing a suitable value of fixed  $L_d$ ,  $L_q$  and  $\varphi_e$  it was possible to create a new motor map that approximates the real one the best. The best fitting motor model was chosen empirically by tuning the fixed parameters.

#### 4.1.2 Torque-Power maps

A full LUT that follows the MTPA and MTPV region of the simplified motor was obtained using a Matlab script.

### 4.1.3 Motor simulation

The motor implements speed control loop, with the possibility to introduce an external braking torque on the rotor shaft.

The inverter has been modelled with *Universal Bridge* block from Simscape:

- No switching.
- $\bar{V}_{abc}$  Pure sinusoidal.

In Fig. 4.2 the Simulink model of the motor subsystem. The Simscape motor block

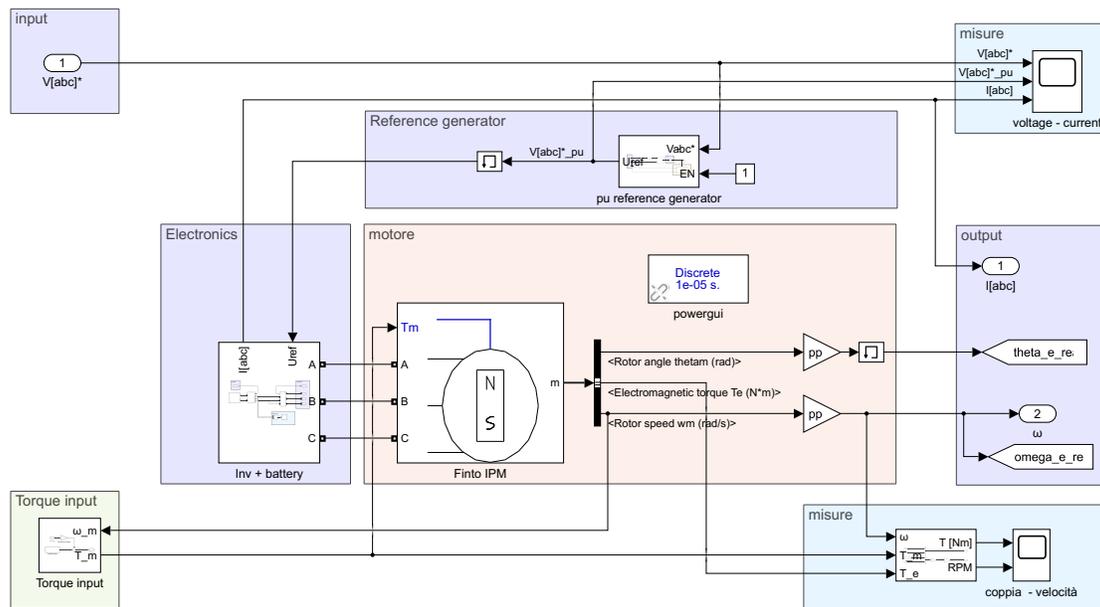


Figure 4.2: Simulink motor model.

is configured as follows:

|                       |                           |
|-----------------------|---------------------------|
| Number of phases      | 3                         |
| Back EMF waveform     | Sinusoidal                |
| Rotor type            | Salient-pole              |
| Mechanical input      | Torque Tm                 |
| Preset model          | No                        |
| Discrete solver model | Trapezoidal non-iterative |
| Sample time           | inherited                 |

Table 4.1: *Permanent Magnet Synchronous Machine* configuration.

4.1.4 Motor Control

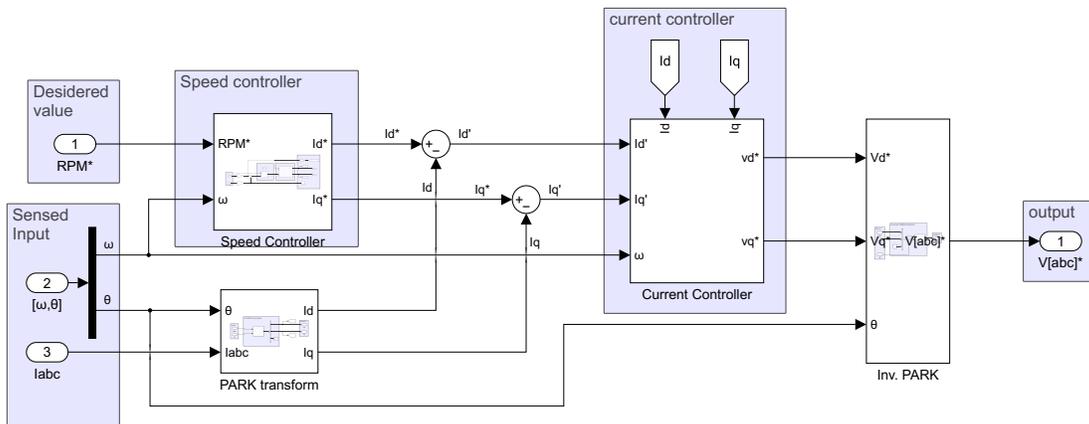


Figure 4.3: Overall Simulink motor control.

The *ABC* voltage signals that feeds the inverter are calculated by the controller subsystem. The overall control scheme can be seen in appendix A

4.1.4.1 Speed control

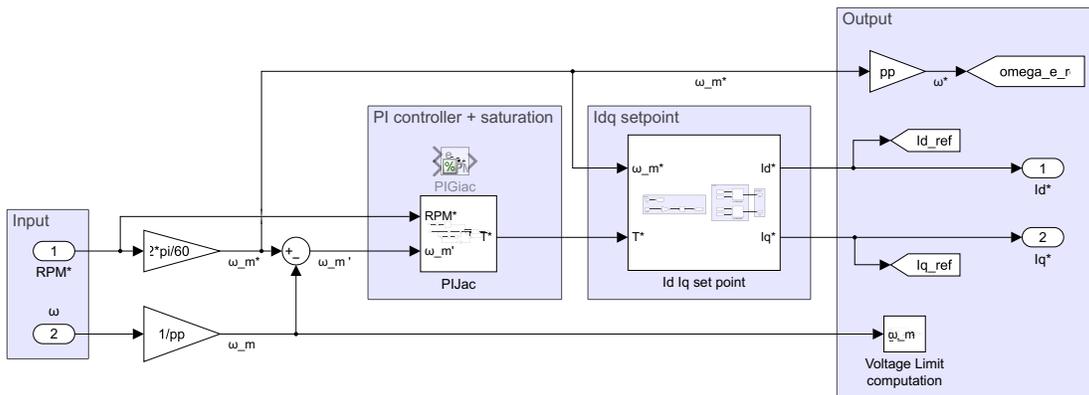


Figure 4.4: Simulink motor speed controller.

The speed control uses a PI regulator, whose output is the saturated torque reference.  $i_d^*$  and  $i_q^*$  reference were computed by means of a Lookup Table implementing the MTPA and MTPV regions.

#### 4.1.4.2 Current control

$v_d$  and  $v_q$  setpoint are obtained from  $i_d$  and  $i_q$  using a saturated PI regulator to not exceed the maximum voltage. It was decided to prioritize  $v_d$  during saturation (if overvoltage is detected,  $v_q$  is clamped, lowering down torque, but still trying to reach the desired speed).

To increase the overall performance of the control, a feed forward was implemented.

#### 4.1.5 Inverter and battery

Inverter and battery have not been modelled: an ideal voltage source was used as the battery, and an average-model based Voltage Source Control (VSC) will feed to the motor with sinusoidal voltage reference (no PWM). The voltage reference is calculated to behave like a SPWM.

An example in Fig. 4.5.

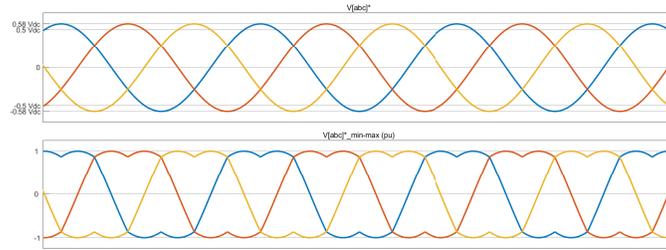


Figure 4.5: Ideal inverter sinusoids with min. max. injection.

Battery is modelled as a constant voltage source.

## 4.2 Three-phase inverter full-replica Switching model

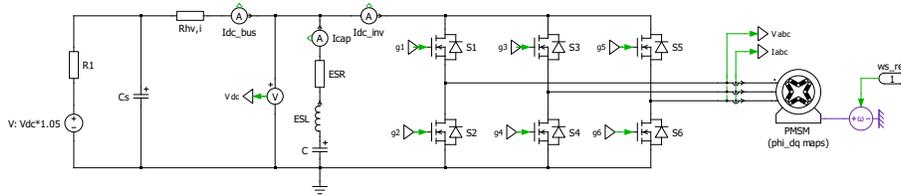


Figure 4.6: Switching motor block scheme in Ples.

After using the simplified model in subsection 4.1, it was possible to use a more realistic Simulink motor model with variable parameters and a switching inverter model.

The motor model was implemented in Plecs [27], using a *Lookup Table-Based Permanent-Magnet Synchronous Machine* [59]. The overall control was implemented using Simulink [26].

#### 4.2.1 Motor parameters

The motor block uses a  $\bar{\varphi}_{dq0}$  LUT obtained from FEM data of a real motor.

As can be seen from Fig. 4.6, the motor is driven by a 3-phase switching MOSFET inverter modelled in Plecs [27].

#### 4.2.2 Motor simulation

The motor shaft is dragged by a rotating reference, and  $I_{dq}$  currents are set to obtain a fixed electric reference torque  $T^*$ . Only the current control loop is implemented, with the possibility to run both in sensorless and sensed mode.

#### 4.2.3 Inverter simulation

A *Three-phase Full-Bridge MOSFET Inverter* is modelled using Plecs [27] (Fig. 4.6). The input of the inverter are the PWM signals generated by the controller, so it will be possible to get an idea of current/torque ripples in output accordingly to the PWM frequency.

The full control strategy was widely covered in subsection 3.1, and the overall scheme can be seen in appendix A. Note that in this model, the speed loop present in appendix A is not present.

##### 4.2.3.1 PWM Generator

The gate signals can be calculated in Simulink [26] using one of the following modulation scheme:

- Sinusoidal PWM (SPWM).
- Sinusoidal PWM (SPWM) with min. max. injection.
- Space vector modulation (SVM).

In the following simulations Sinusoidal PWM with min. max. injection was implemented, with maximum modulation index  $m_{max} = 1$ . Its block scheme implementation can be seen in Fig. 4.7.

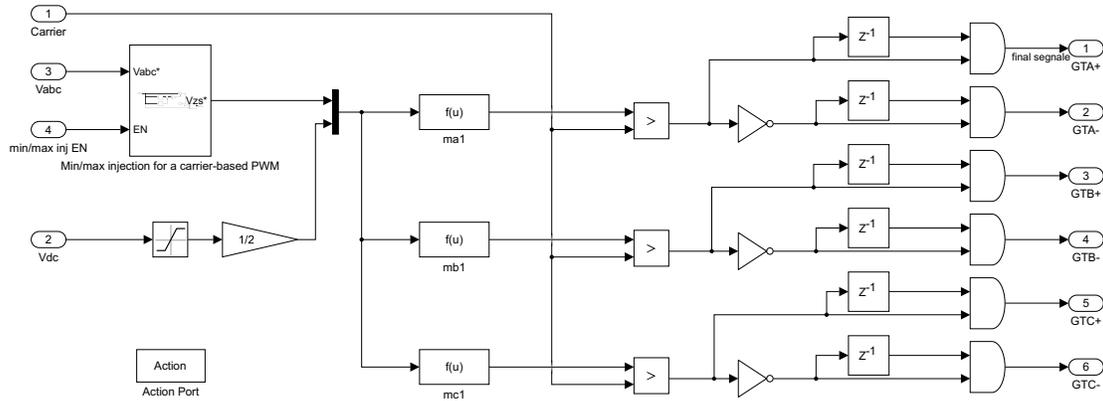


Figure 4.7: Sinusoidal PWM with min. max. injection in Simulink [26].

### 4.2.4 Switching loss

It is possible to implement a thermal circuit of the inverter, and evaluate the temperatures reached during operation, but in the thesis’s simulations this was disabled, speeding up simulation time.

### 4.2.5 Motor Control

The motor controller consists in a speed and current loop, the overall scheme can be seen in appendix A. In Fig. 4.8 the controller scheme of the model implemented Simulink, where only current loop is implemented.

#### 4.2.5.1 MTPA and MTPV strategy

As it was discussed in subsection 4.1.2, it was possible to obtain a full LUT that follows the MTPA and MTPV region of the motor using a Matlab script and FEM data.

#### 4.2.5.2 Speed control

In our simulations the speed is fixed by a rotating reference, so the speed loop is only controlling the current inside the windings to obtain a certain amount of electric torque.  $i_d^*$  and  $i_q^*$  reference were computed by the electric torque reference  $T^*$  and speed reference  $\omega_e^*$  by means of a Lookup Table.

#### 4.2.5.3 Current control

$v_d$  and  $v_q$  setpoint are obtained from  $i_d$  and  $i_q$  using a saturated PI regulator to not exceed the maximum voltage, thus avoiding crossing the overmodulation limit.

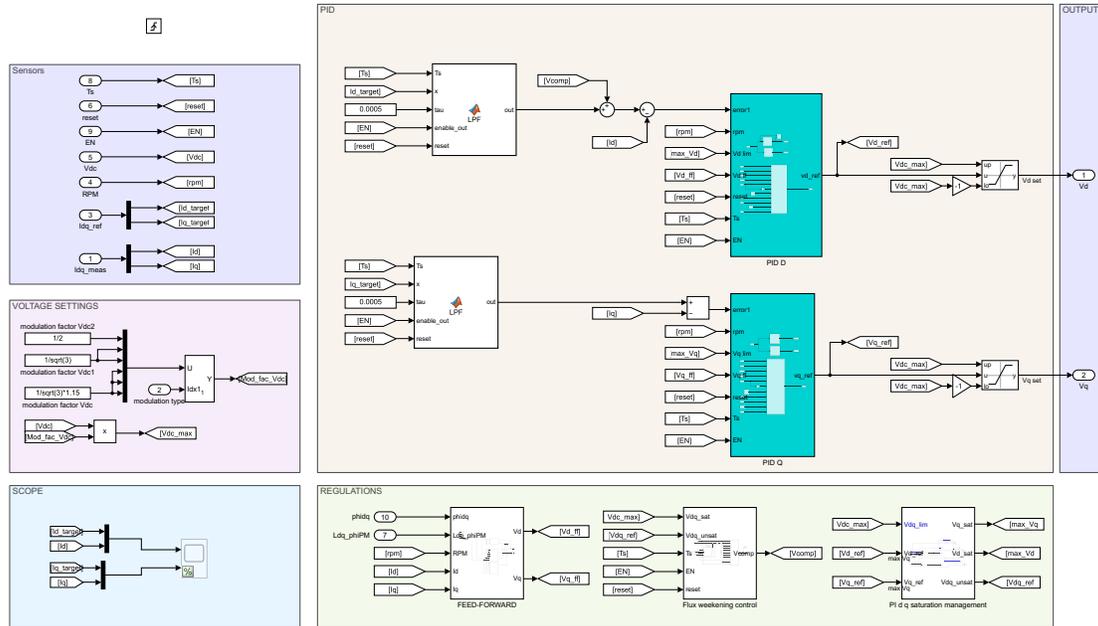


Figure 4.8: Current controller in Matlab.

It was decided to prioritize  $v_d$  during saturation (if overvoltage is detected,  $v_q$  is clamped, lowering down torque, but still trying to reach the desired speed).

To increase the overall performance of the control, a feed forward was implemented, which block scheme can be seen in appendix A.

To increase the robustness of the control in high speed, a flux weakening control was implemented. It will be used whenever the  $i_d$   $i_q$  control maps fails to keep the motor rotating at high speeds.

**Current PI tuning** Current PI are tuned using the Magnitude Optimum technique, as mentioned in subsection 3.1.0.2.

### 4.2.6 Battery

During simulation, the battery is approximated as a constant voltage source. In addition to the battery, the dc link capacitor is implemented.

## 5 Real Hardware test bench

Thanks to the laboratories provided in the department of DEI in UNIBO, I had the opportunity to work on a hardware test bench and verify the observers algorithm in real word scenarios. The objective of the experimental test are the following:

The main questions to be answered during the test are the following:

- Reliability of the algorithm with limited HW resources and computational time.
- Robustness of the algorithm with motor parameter imprecision.

### 5.1 Hardware

The hardware for the test was provided by the DEI laboratory in UNIBO. It consists in six main part:

- **DC power supply** as a DC bus.
- **DSP and MOSFET driver** which controls a three-phase motor.
- **Oscilloscope** to inspect the results of the algorithm.
- **DC motor** as a load connected to the three-phase motor shaft.
- **Three-phase SPM motor** to test the control algorithm and the observer performances.

#### 5.1.1 Three-phase SPM Motor

The three-phase motor implemented is the *TEM BTSS 1524* [60]. It is a SPM form TEM. The motor's main characteristics are summarized in the following table:

|                               |               |                                      |
|-------------------------------|---------------|--------------------------------------|
| Weight                        |               | 1.35 kg                              |
| Continuos stall torque in air | $T$           | 0.60 N m                             |
| Rated speed                   | $\omega_{op}$ | 1500 RPM                             |
| Input Voltage                 | $V_{in}$      | 24 V                                 |
| Rotor inertia                 | $J_s$         | $0.035 \times 10^{-3} \text{ kgm}^2$ |
| Stator phase resistance       | $R_s$         | 0.74 $\Omega$                        |
| Winding inductance            | $L_s$         | 1.4 mH                               |
| Excitation flux               | $\varphi_e$   | 0.0247 Wb                            |

Table 5.1: three-phase motor electromechanical parameters.



Figure 5.1: TEM BTSS 1524.

The Motor shaft is mounted on the DC motor rotor (which acts as a brake), and the three phases are connected to the inverter. This three phase motor will be the main traction machine studied.

This motor has a built-in incremental encoder, which will be used for the FOC

### 5.1.2 DC motor



Figure 5.2: TEM PMB 1524.

The DC motor is used to add a resistant torque to the SPM. The torque generated by this motor is not controlled, and consists in a resistive pack attached on its

windings; therefore, the braking torque will be proportional to the speed. Its main characteristics are summarized as follows:

|                     |               |                        |
|---------------------|---------------|------------------------|
| Weight              |               | 1.35 kg                |
| Nominal Torque      | $T$           | 0.22 N m               |
| Rated speed         | $\omega_{op}$ | 1500 RPM               |
| Rates power         | $P$           | 35 W                   |
| Input Voltage       | $V_{in}$      | 24 V                   |
| Rotor inertia       | $J_s$         | 0.352 kgm <sup>2</sup> |
| Armature resistance | $R_a$         | 3.20 $\Omega$          |
| Armature inductance | $L_a$         | 4.1 mH                 |
| Excitation flux     | $\varphi_e$   | 0.12 Wb                |

Table 5.2: DC motor electromechanical parameters.

### 5.1.3 Load

Different resistive loads can be plugged on the stator windings to change the braking characteristic of the machine.

If the resistive load  $R_{load}$  is known, the braking torque  $T_B$  can be calculated as follows:

$$T_B = \frac{\varphi_{e,dc}^2}{\omega_m(R_a + R_{load})} \quad (5.1)$$

three different resistive load have been utilized in the test:

- No load:  $R_{load} = \text{inf } \Omega$
- Max load:  $R_{load} = 0 \Omega$
- Medium load:  $R_{load} = 20 \Omega$

Medium load have been implemented using a resistive load (Fig. 5.3a), and its load characteristic is in Fig. 5.3b.

For the maximum and minimum load, a relay was connected directly to the DC motor connection, simulating short and open circuits in the motor terminals.(Fig. 5.4a). The maximum load curve can be seen in Fig. 5.4b.

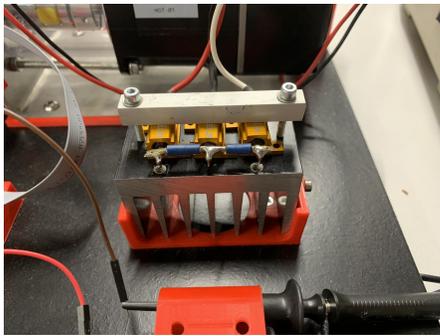
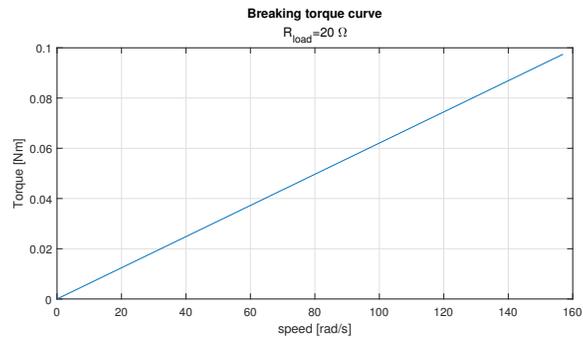
(a)  $20\ \Omega$  resistive load.(b) Resistive load characteristic  $20\ \Omega$ .

Figure 5.3: Motor 1 Sensored speed control dynamic.

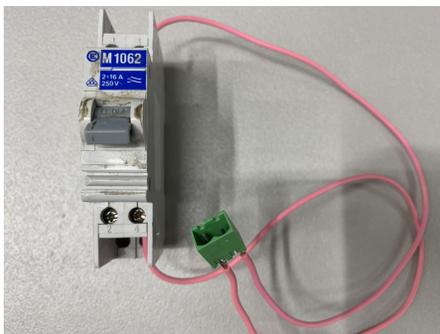
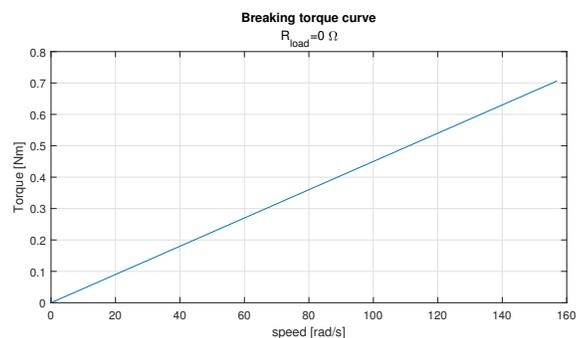
(a) Relay for  $0\ \Omega$  and  $\infty\ \Omega$  resistive.(b) Resistive load characteristic  $0\ \Omega$ .

Figure 5.4: Motor 1 Sensored speed control dynamic.

### 5.1.4 Control HW

The hardware used to control the three-phase DC motor is composed by two main part:

- DSP: LAUNCHXL-F28379D [61].
- Driver : BOOSTXL-DRV8305EVM [62].

#### 5.1.4.1 DSP: LAUNCHXL-F28379D

The DSP operated is an evaluation board based on the TMS320F28379D Microcontroller. Its main characteristics are the following:



(a) LAUNCHXL-F28379D [61].

(b) BOOSTXL-DRV8305EVM [62].

Figure 5.5: Control Hardware.

|                  |                    |
|------------------|--------------------|
| CPU              | 200 MHz dual-core  |
| Flash memory     | 1 MB dual-core     |
| RAM              | 204 kB dual-core   |
| ADC resolution   | 12 bit - 16 bit    |
| Total processing | 800 MIPS           |
| CPU units        | FPU32, TMU, VCU-II |

Table 5.3: TMS320F28379D MCU characteristics.

Many characteristics of the MCU won't be used. Nevertheless, having such high computational power and memory will be very convenient.

#### 5.1.4.2 Driver: BOOSTXL-DRV8305EVM

The driver board is based on CSD18540Q5B Power MOSFET: a Silicon MOSFET built for Motor inverter and DC-DC converters. Its main characteristics are the following:

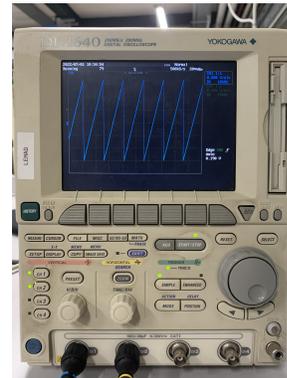
|                          |          |      |
|--------------------------|----------|------|
| Drain-to-source voltage  | $V_{gs}$ | 60 V |
| Continuous Drain Current | $I_D$    | 29 A |
| Body Diode               |          | Yes  |

Table 5.4: CSD18540Q5B Power MOSFET characteristics.

### 5.1.5 Power supply and measurements



(a) EA-PS 2042-20B.



(b) DL1640.

Figure 5.6: Power supply and oscilloscope.

Other instrumentations available in laboratory has been used.

1. DC Power supply: *Elektro-Automatik* EA-PS 2042-20B [63].
2. Oscilloscope: *Yokogawa* DL1640 [64]

The DC voltage supply will be used to feed the DC-bus of the inverter, whereas the oscilloscope will be used to visualize two selected control variables in real time.

#### Note on oscilloscope acquisition

The fastest way to visualize control variables during runs is to get information directly from the DSP board. It was possible by using the DAC peripheral, which outputs the variable on an analog pin. The analog pin is then connected to the oscilloscope which will be used for data acquisition. Therefore, the oscilloscope acquisition presented in the thesis are digital signals.

## 5.2 Software

One advantage of that board is the ability to develop and deploy the control algorithm using Simulink and model-based design. The following software are used to make the DSP board communicating with Simulink:

- Code Composer Studio 8.3.0 [65].
- Control Suite 3.4.9 [66].

- C2000ware 3.03.00 [67].
- Simulink TI C2000 Support from Embedded Coder (TI Delfino F2837xD support) 21.2.2 [68].

The control algorithm, and every board setting (peripherals, ports, clocks etc...), have been set up using Simulink blocks. Code generator and HW deploy was done automatically within Simulink environment.

### 5.2.1 Control algorithm

The motor was controlled in speed, and FOC algorithm was used.

#### 5.2.1.1 Protections

Motor signal are monitored by a safety-function to detect faults. A state machine is implemented to steer the motor into different modes Fig. 5.7.

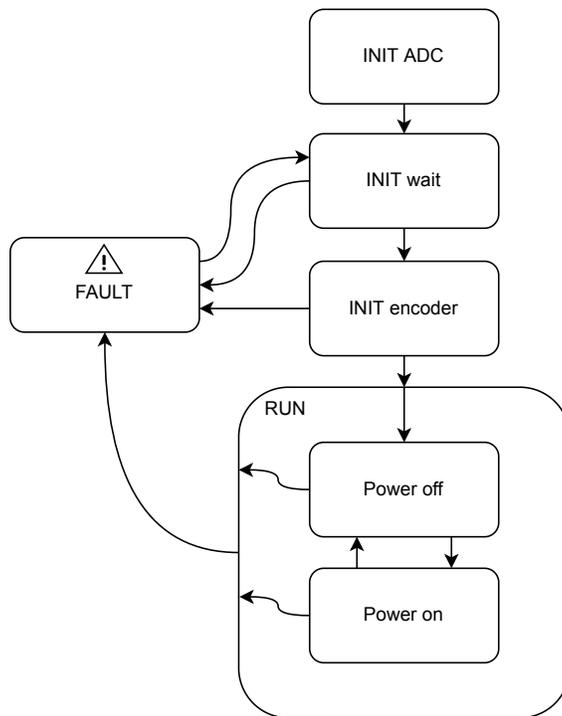


Figure 5.7: Stateflow of the testbench inverter.



## 6 Results and Conclusions

The evaluation of rotor position observer algorithms are discussed in the following chapters. A preliminary evaluation was carried out in simulation using the simplified model, then the algorithms are verified using full-replica model. Lastly, the observer algorithm are verified using a real hardware test bench.

### 6.1 Motor 1: Simplified simulation

The simplified model was implemented for this simulation (subsection 4.1).

In Fig. 6.1 the capability maps of the motor implemented, together with the calculated  $i_d$  and  $i_q$  control map.

The constant motor parameters ( $L_d$ ,  $L_q$ ,  $\varphi_e$ ,  $R_s$ ) are chosen to approximate the FEM curves of a real motor. The motor curves of the real corresponding motor are in Fig. 6.18.

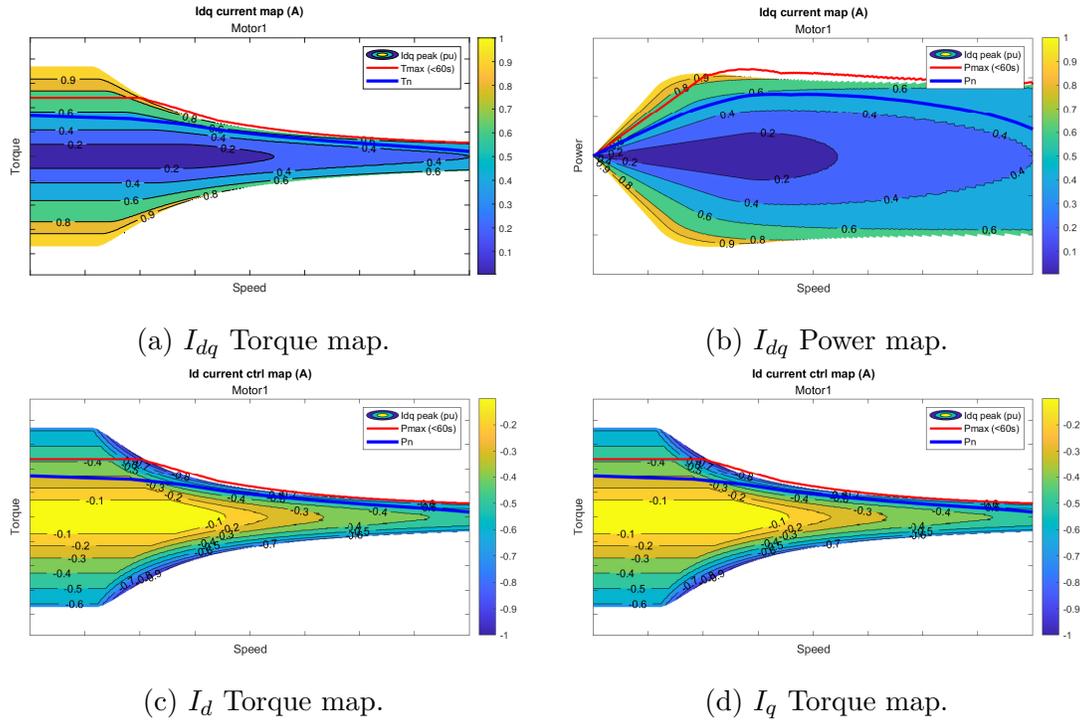


Figure 6.1:  $I_{dq}$  maps of the simplified Motor 1.

In Fig. 6.2a the dynamic performance of the speed controlled motor, connected to a dynamic torque of  $T_m = 0.1 \cdot \omega_m$ . Sensored control is implemented, and the input speed reference is a dumped step sequence (dumping time constant  $\tau_s = 0.05$  s).

This control input will be utilized to benchmark the observers.

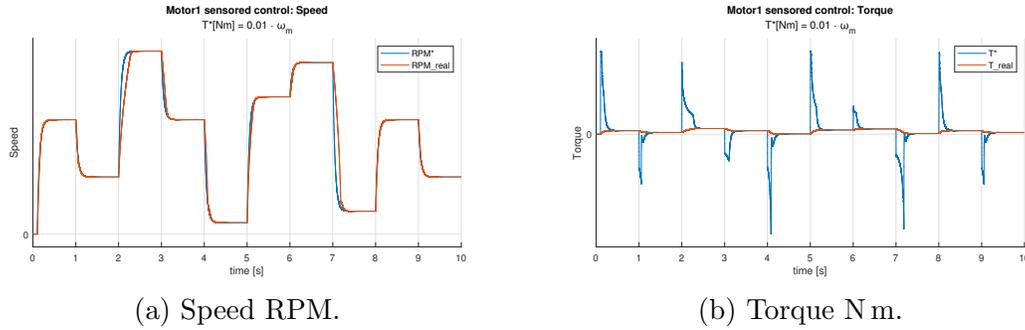


Figure 6.2: Motor 1 Sensored speed control dynamic.

### 6.1.1 Parameters

In Tab. 6.1 the Simulink simulation parameter utilized, and in Tab. 6.2 the observers parameters.

|                 |                      |
|-----------------|----------------------|
| Fixed step size | $1 \times 10^{-6}$ s |
|-----------------|----------------------|

Table 6.1: Motor 1 simulation parameters.

|     |  |       |  |
|-----|--|-------|--|
| SMO | OBS gain                                 | $K_s$ | 1500                                   |
|     | Sigmoid constant                         | $a$   | 0.1                                    |
|     | QPLL proportional gain                   | $K_p$ | 10                                     |
|     | QPLL integral gain                       | $K_i$ | 1000                                   |
|     | Sample time                              | $T_s$ | $2 \times 10^{-7}$ s                   |
| EKF | System noise covariance matrix diag      | $Q$   | $[0.5, 0.5, 50, 1 \times 10^{-6}]$     |
|     | Measurement noise covariance matrix diag | $R$   | $[1 \times 10^{-6}, 1 \times 10^{-6}]$ |
|     | Sample time                              | $T_s$ | $2 \times 10^{-7}$ s                   |

Table 6.2: Motor 1 observer parameters.

### 6.1.2 Observer algorithm steady state evaluation

The two algorithms (EKF and SMO) have been compared in steady state behavior. The motor is speed controlled using sensed control.

The speed and torque setpoint are fixed, and the simulation runs for 3 s to let the dynamics become stable. The output of this test is the percentage error of the electric angle  $\theta_e$  observed VS the real value.

As can be seen from Fig. 6.3, the EKF algorithm seems to be more precise compared to SMO, in every operational point.

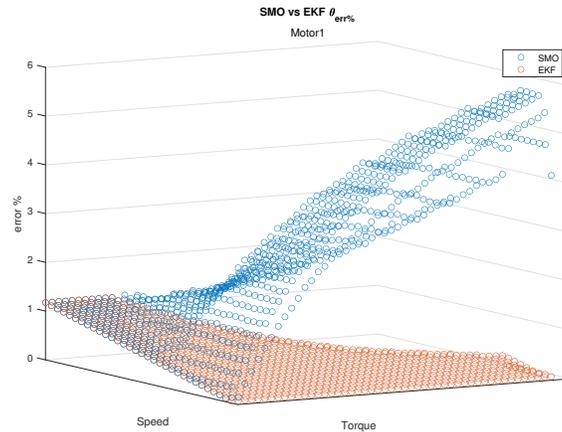
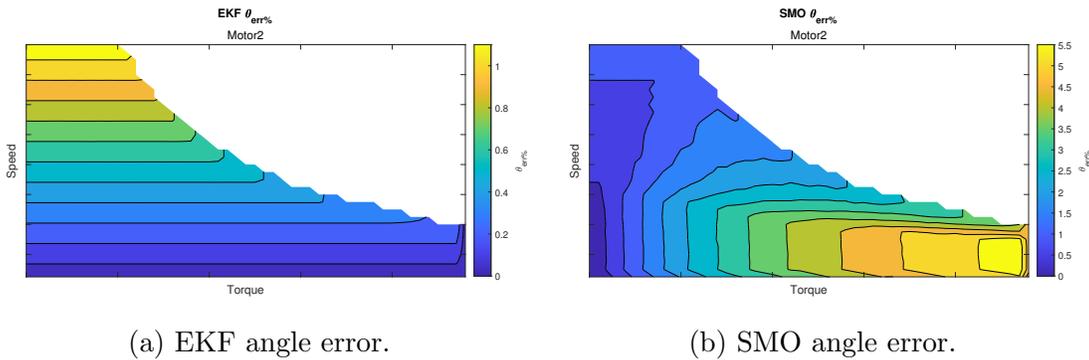


Figure 6.3: Motor 1: SMO VS EKF steady state angle error %.



(a) EKF angle error.

(b) SMO angle error.

Figure 6.4: Motor 1: SMO and EKF steady state angle error %...

### 6.1.3 Observer algorithm dynamic evaluation

To understand the dynamic behavior of the algorithm, another test is arranged. The dynamic test is arranged both with sensed and sensorless control.

During the dynamic tests, the motor is asked to follow the same speed reference already seen in Fig. 6.2a. If control is sensorless, the observed angle and speed will be put in the control loop after a settling time (the algorithms implemented are not designed for low speed).

The following metrics will be evaluated from the results:

- $\theta_{e,err,\%}$ : Electric angle mean error percentage.
- $\theta_{e,err,mean}$ : Electric angle mean error.
- $RPM_{err,std}$ : mechanical speed error standard deviation.

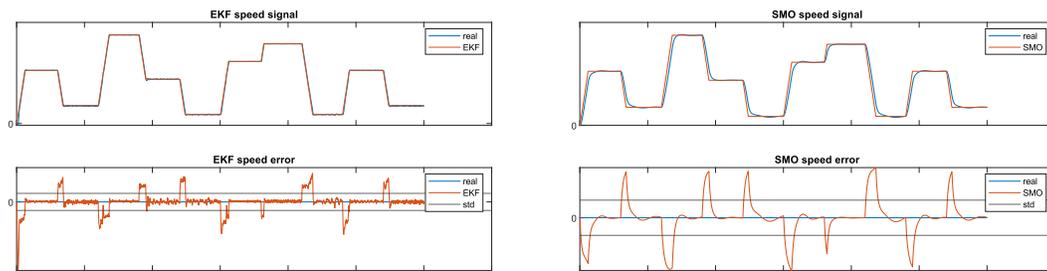
The mechanical speed error is calculated as the difference between the observed speed and the real one:

$$RPM_{err} = \omega_{real} - \omega_{obs} \quad (6.1)$$

Then the standard deviation of this signal is calculated:

$$RPM_{err,std} = std(RPM_{err}) \quad (6.2)$$

An example of speed error standard deviation calculation in Fig. 6.5.



(a) EKF vs Real.

(b) SMO vs Real.

Figure 6.5: Example of speed error standard deviation calculation.

### 6.1.3.1 No load - Sensored

During the no load test, a dynamic torque of  $T_m = 0.1 \cdot \omega_m$  is introduced on the motor shaft.

Both SMO and EKF works correctly, as can be seen on the output plots Fig. 6.6 and the calculated metrics Tab. 6.3

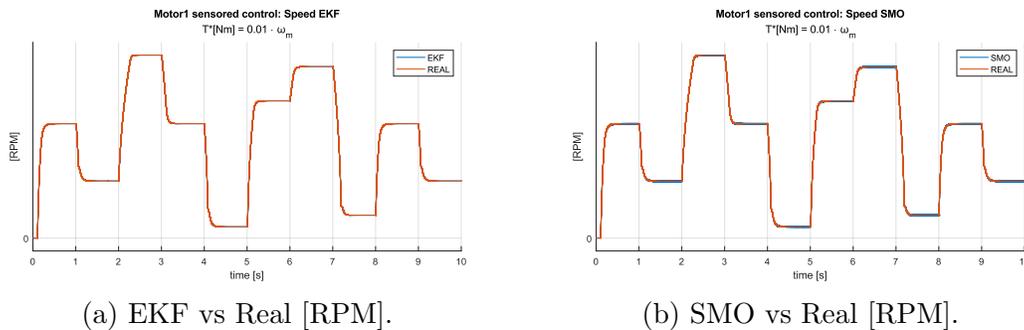


Figure 6.6: Motor 1 - Real speed vs observer - No load - Sensored control.

|   |            | EKF     | SMO     |
|---|------------|---------|---------|
| Electric angle mean error                 | [%]        | 0.0808  | 0.7325  |
| Electric angle mean error                 | [ $\mu$ s] | 1.7163  | 25.2278 |
| Mechanical speed error standard deviation | [RPM]      | 10.3548 | 24.3625 |

Table 6.3: Motor 1 - Real vs observer - No load - Sensored control.

### 6.1.3.2 No load - Sensorless

The no load sensorless test is the same as the no load sensed, but the observer measurements are injected in the control loop with a delay of 0.1 s.

Again, as can be seen from the plots (Fig. 6.7) and metrics (Tab. 6.4), both the algorithms behave correctly.

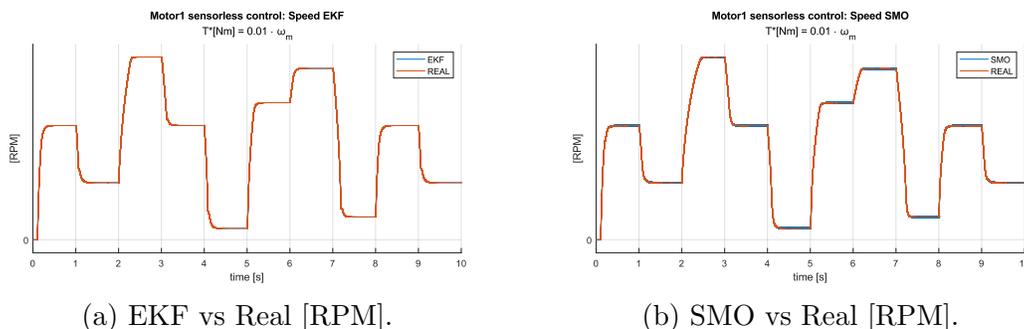


Figure 6.7: Motor 1 - Real speed vs observer - No load - Sensorless control.

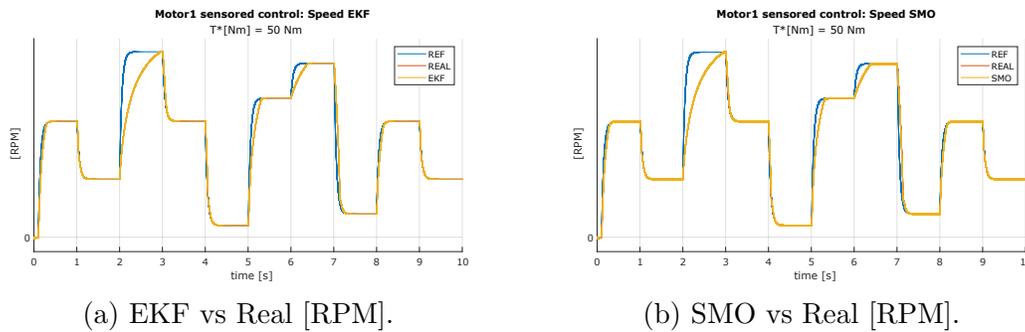
|   |            | EKF     | SMO     |
|---|------------|---------|---------|
| Electric angle mean error                 | [%]        | 0.0799  | 0.7068  |
| Electric angle mean error                 | [ $\mu$ s] | 4.6257  | 25.1242 |
| Mechanical speed error standard deviation | [RPM]      | 10.6058 | 25.2265 |

Table 6.4: Motor 1 - Real vs observer - No load - Sensorless control.

### 6.1.3.3 50 N m - Sensored

During the test, a static counter torque is set at the rotor shaft  $T_m = 50$  N m. The speed reference is the same seen in Fig. 6.2a.

In Fig. 6.8 it is possible to notice that the motor speed does not follow accurately the reference, because of the limitation of the motor performances. Both the observers follows correctly the real speed (yellow and red traces does overlap in Fig. 6.8), and the metrics suggest good behaviors too Tab. 6.5.

Figure 6.8: Motor 1 - Real speed vs observer -  $T_m = 50$  N m - Sensored control.

|   |            | EKF      | SMO     |
|---|------------|----------|---------|
| Electric angle mean error                 | [%]        | 0.608 11 | 1.8444  |
| Electric angle mean error                 | [ $\mu$ s] | 11.9699  | 80.108  |
| Mechanical speed error standard deviation | [RPM]      | 9.8651   | 25.7033 |

Table 6.5: Motor 1 - Real vs observer -  $T_m = 50$  N m - Sensored control.

### 6.1.3.4 50 N m - Sensorless

Observer measurements are injected in control loop after 0.1s, this is necessary because of the poor performances of the sensorless algorithms at low speed.

In Fig. 6.9b it is possible to notice a mismatch between real angle and the observed value in the SMO: the motor controller had difficulties following the reference signal. This behavior may be caused by the mismatch between the real and observed angle.

Despite that, the motor control remains stable, as shown from the speed plot (Fig. 6.9) and the calculated metrics (Tab. 6.6).

As seen in this test, the observer angle error can affect the motor performances.

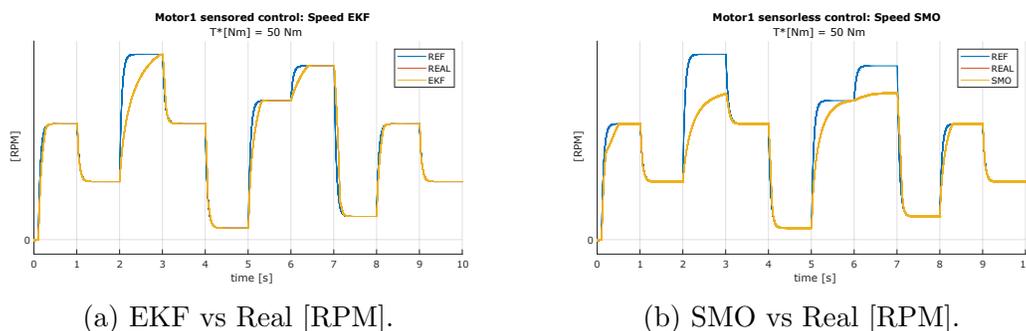


Figure 6.9: Motor 1 - Real speed vs observer -  $T_m = 50$  N m - Sensorless control.

|   |            | EKF      | SMO     |
|---|------------|----------|---------|
| Electric angle mean error                 | [%]        | 0.608 11 | 1.8444  |
| Electric angle mean error                 | [ $\mu$ s] | 11.9699  | 80.108  |
| Mechanical speed error standard deviation | [RPM]      | 9.8651   | 25.7033 |

Table 6.6: Motor 1 - Real vs observer -  $T_m = 50$  N m - Sensorless control.

## 6.2 Motor 2: Simplified simulation

The simplified model was implemented for this simulation (subsection 4.1), but a different and more performing motor was used as reference.

As for motor 1, this simplified motor 2 model is chosen to behave as closely as possible to a real motor simulated via FEM. In Fig. 6.10 the capability maps of the simplified motor 2.

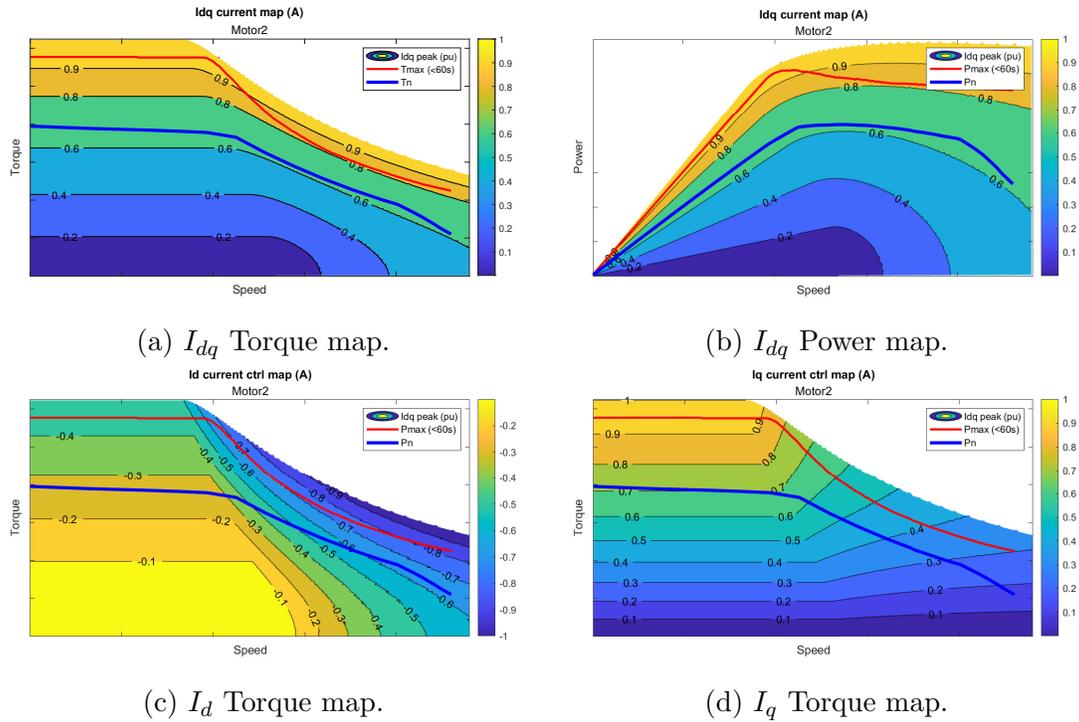


Figure 6.10:  $I_{dq}$  maps of the simplified Motor 2.

Similarly to motor 1, in Fig. 6.11a the dynamic performance of the speed controlled motor, connected to a dynamic torque of  $T_m = 0.1 \cdot \omega_m$ . Sensored control is implemented, and the input speed reference is a dumped step sequence (dumping time constant  $\tau_s = 0.05$  s).

### 6.2.1 Parameters

In Tab. 6.7 the Simulink simulation parameter utilized, and in Tab. 6.8 the observers parameters.

|                 |                      |
|-----------------|----------------------|
| Fixed step size | $1 \times 10^{-6}$ s |
|-----------------|----------------------|

Table 6.7: Motor 2 simulation parameters.

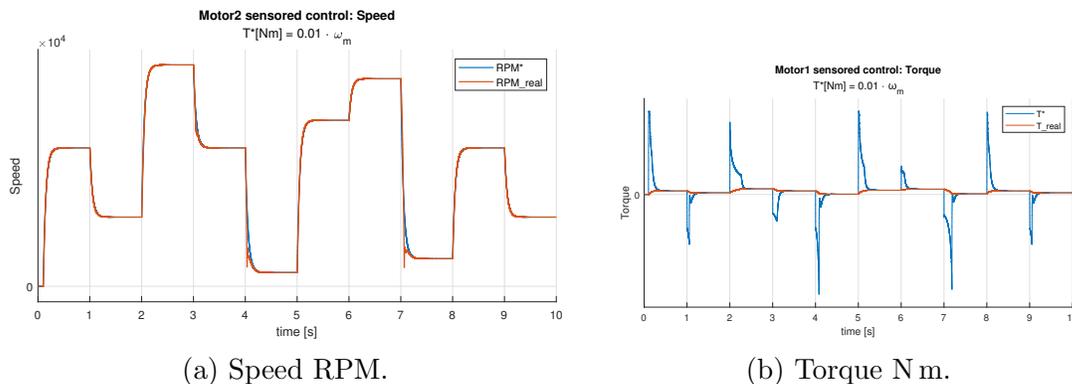


Figure 6.11: Motor 2 Sensored speed control dynamic.

|     |  |       |  |
|-----|--|-------|--|
| SMO | OBS gain                                 | $K_s$ | 1500                                   |
|     | Sigmoid constant                         | $a$   | 0.1                                    |
|     | QPLL proportional gain                   | $K_p$ | 0.025                                  |
|     | QPLL integral gain                       | $K_i$ | 0.5                                    |
|     | Sample time                              | $T_s$ | $1 \times 10^{-6}$ s                   |
| EKF | System noise covariance matrix diag      | $Q$   | $[0.5, 0.5, 50, 1 \times 10^{-6}]$     |
|     | Measurement noise covariance matrix diag | $R$   | $[1 \times 10^{-6}, 1 \times 10^{-6}]$ |
|     | Sample time                              | $T_s$ | $1 \times 10^{-6}$ s                   |

Table 6.8: Motor 2 observer parameters.

### 6.2.2 Observer algorithm steady state evaluation

The two algorithms (EKF and SMO) have been compared in steady state behavior. The motor is speed controlled using sensed control.

The test is the same as what was done to motor 1, except for the higher maximum torque and speed reachable with this motor.

As can be seen from Fig. 6.12, the EKF algorithm does approximate the rotor angle better than SMO in every operational point.

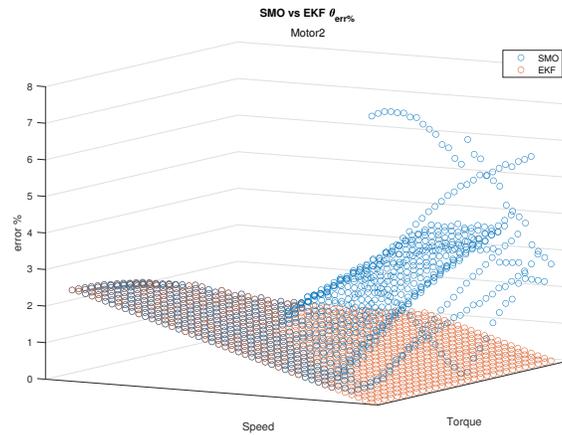
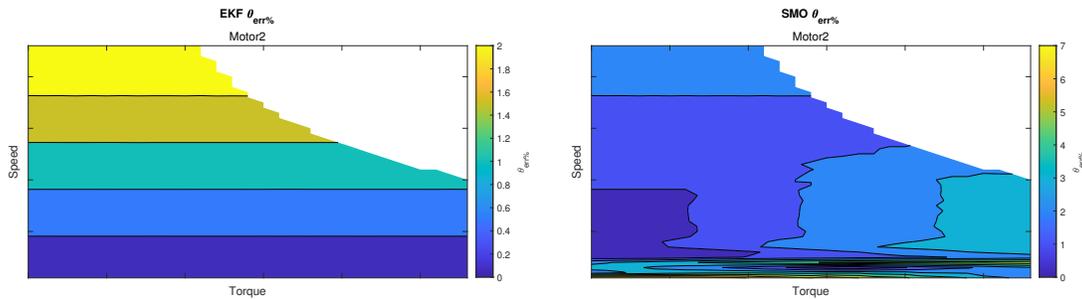


Figure 6.12: Motor 1: SMO VS EKF steady state angle error %.



(a) EKF angle error.

(b) SMO angle error.

Figure 6.13: Motor 1: SMO and EKF steady state angle error %.

### 6.2.3 Observer algorithm dynamic state evaluation

The dynamic tests are the same conducted for motor 1, with the same evaluation metrics.

#### 6.2.3.1 No load - Sensored

Similarly to the no load test performed for motor 1, a dynamic torque of  $T_m = 0.1 \cdot \omega_m$  is added on the motor shaft, and the motor control is sensored.

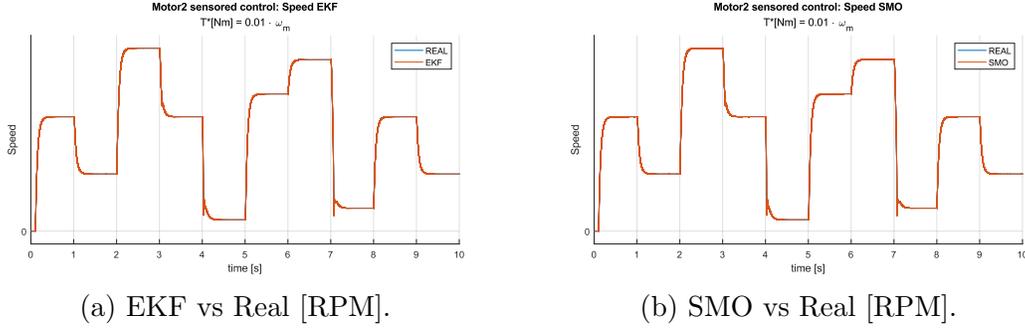


Figure 6.14: Motor 2 - Real speed vs observer - No load - Sensored control.

|   |            | EKF     | SMO     |
|---|------------|---------|---------|
| Electric angle mean error                 | [%]        | 1.2935  | 2.1173  |
| Electric angle mean error                 | [ $\mu$ s] | 10.8777 | 40.9782 |
| Mechanical speed error standard deviation | [RPM]      | 12.239  | 2.9668  |

Table 6.9: Motor 2 - Real vs observer - No load - Sensored control.

### 6.2.3.2 No load - Sensorless

The following test is the same as the no load sensorless test, but the observer measurements are injected in the control loop (after 0.1 s).

Plots (Fig. 6.15) and metrics (Tab. 6.10) shows good approximation of the rotor angle by both the algorithm. It is noticeable that using the EKF, the motor does struggle to follow the speed reference: in Fig. 6.15a the speed does present unwanted spikes during deceleration (at 1 and 9 seconds), and the standard deviation of the speed is higher than the usual value, even if the rotor angle is smaller.

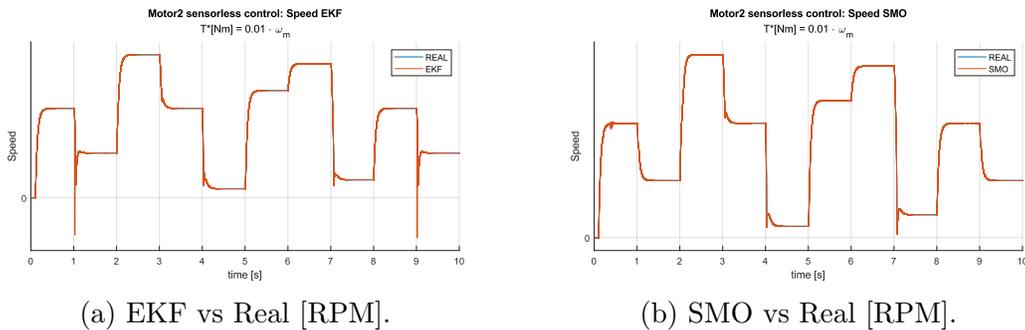


Figure 6.15: Motor 2 - Real speed vs observer - No load - Sensorless control.

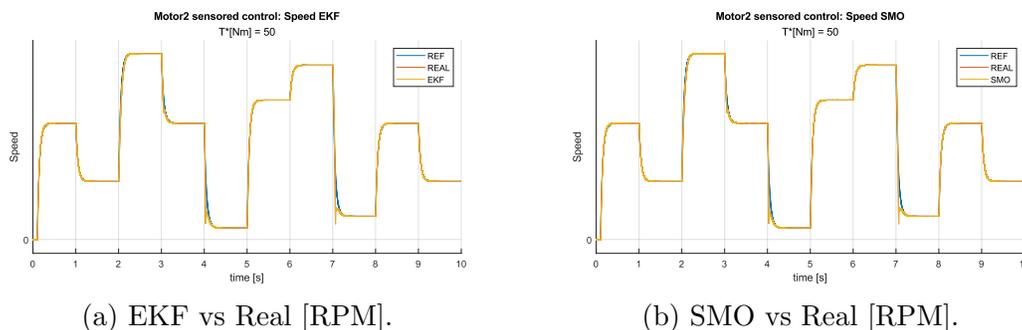
|   |            | EKF      | SMO     |
|---|------------|----------|---------|
| Electric angle mean error                 | [%]        | 1.2875   | 2.0697  |
| Electric angle mean error                 | [ $\mu$ s] | 11.0061  | 40.2785 |
| Mechanical speed error standard deviation | [RPM]      | 183.2388 | 3.0828  |

Table 6.10: Motor 2 - Real vs observer - No load - Sensorless control.

### 6.2.3.3 50 N m - Sensored

This test is the same done for motor 1: the torque is fixed at  $T_m = 50$  N m, the speed reference is the same seen in Fig. 6.11a, and the control is sensored.

Looking at the plots (Fig. 6.16) and metrics (Tab. 6.11), both algorithm works correctly; It is worth noting that the EKF algorithm does observe the angle better, but the output speed error is higher (standard deviation is higher).

Figure 6.16: Motor 2 - Real speed vs observer -  $T_m = 50$  N m - Sensored control.

|   |            | EKF     | SMO    |
|---|------------|---------|--------|
| Electric angle mean error                 | [%]        | 1.2911  | 2.2525 |
| Electric angle mean error                 | [ $\mu$ s] | 11.245  | 46.789 |
| Mechanical speed error standard deviation | [RPM]      | 12.4846 | 3.0696 |

Table 6.11: Motor 2 - Real vs observer -  $T_m = 50$  N m - Sensored control.

### 6.2.3.4 50 N m - Sensorless

The test is the same as the previous, but the observer measurements are injected in control loop after 0.1 s.

Both the algorithm does not present any particular criticalities (as seen in Fig. 6.17 and Tab. 6.12). As for the sensed test, the rotor angle is better observed by the EKF algorithm; despite this, the standard deviation of the speed error is higher when using the EKF, and more spikes are present in the control (Fig. 6.17a) compared to the SMO control.

Comparing the results of motor 2 versus the results from motor 1 on the same test, motor 2 shows increased performances, resulting to a speed curve correctly followed even at higher speeds.

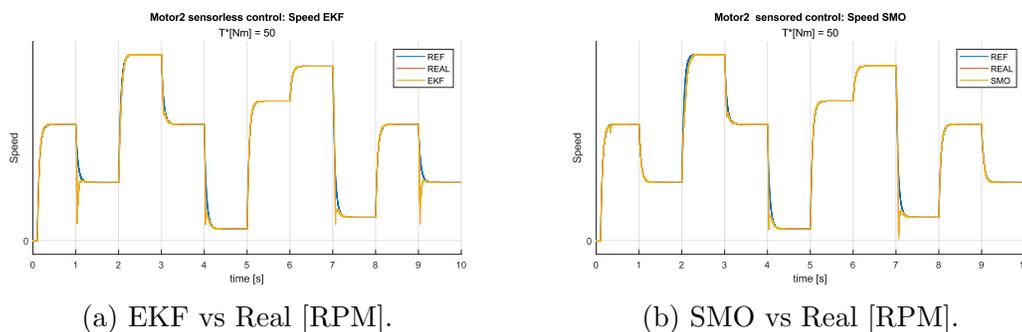


Figure 6.17: Motor 2 - Real speed vs observer -  $T_m = 50$  N m - Sensorless control.

|   |            | EKF      | SMO     |
|---|------------|----------|---------|
| Electric angle mean error                 | [%]        | 1.2868   | 2.2267  |
| Electric angle mean error                 | [ $\mu$ s] | 11.3848  | 47.1677 |
| Mechanical speed error standard deviation | [RPM]      | 109.2954 | 3.1019  |

Table 6.12: Motor 2 - Real vs observer -  $T_m = 50$  N m - Sensorless control.

### 6.3 Motor 1: Full-replica simulation

The LUT utilized for the full-replica model of motor 1 are obtained by means of a FEM simulation of a real motor. The parameters are not constant ( $L_d(i_d, i_q)$ ,  $L_q(i_d, i_q)$ ,  $\varphi_e(i_d, i_q)$ ,  $R_s(i_d, i_q)$ ), and the inverter is fully simulated (more information in subsection 4.2).

In Fig. 6.18 the capability maps.

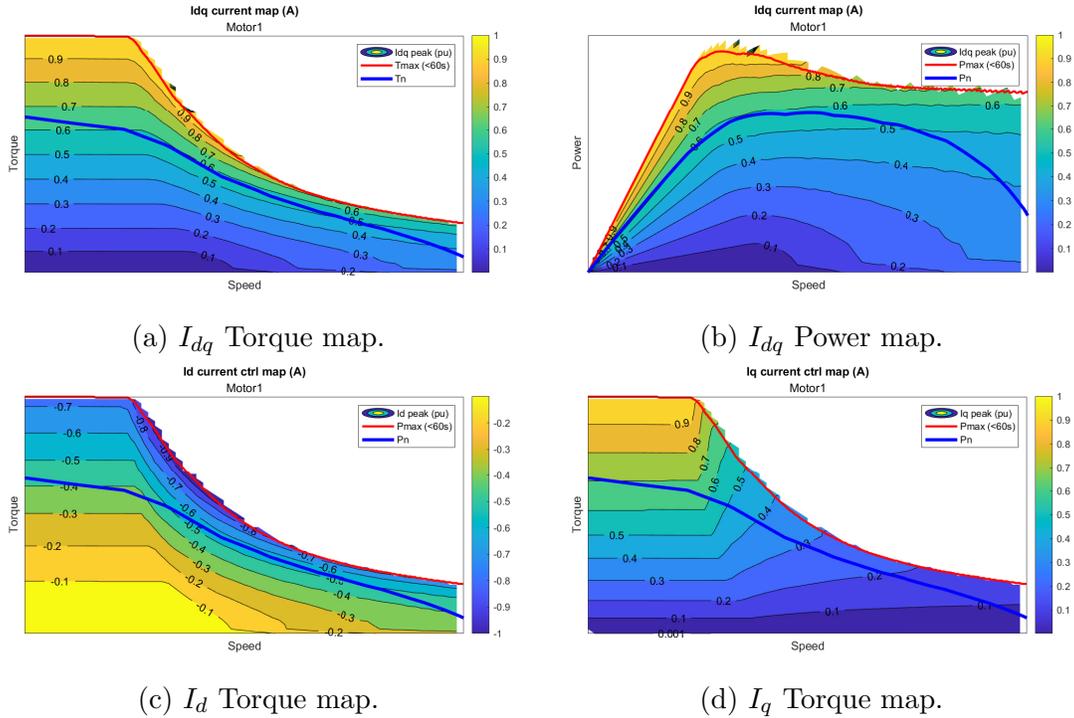


Figure 6.18:  $I_{dq}$  maps of switching motor 1.

#### 6.3.1 Parameters

In Tab. 6.13 the Simulink simulation parameter utilized, and in Tab. 6.14 the observers parameters.

|                    |   |
|--------------------|---|
| Fixed step size    | $1 \times 10^{-7}$ s                            |
| Control frequency  | $1 \times 10^{-6}$ s                            |
| PWM generator      | Sinusoidal PWM (SPWM) with min. max. injection. |
| PWM switching freq | 15 kHz  |

Table 6.13: Motor 1 full-replica simulation parameters.

|     |  |           |                      |
|-----|--|-----------|----------------------|
| SMO | OBS gain                                 | $K_s$     | 1500                 |
|     | Sigmoid constant                         | $a$       | 1.5                  |
|     | QPLL proportional gain                   | $K_p$     | 1                    |
|     | QPLL integral gain                       | $K_i$     | 800                  |
|     | Sample time                              | $T_s$     | $1 \times 10^{-5}$ s |
| EKF | System noise covariance matrix diag      | $Q$       | [11.9, 234, 4078, 0] |
|     | Measurement noise covariance matrix diag | $R$       | [36 313 ,14 348]     |
|     | Compensator                              | $C_{omp}$ | 0.1                  |
|     | Sample time                              | $T_s$     | $1 \times 10^{-5}$ s |

Table 6.14: Motor 1 full-replica observer parameters.

### 6.3.2 Observer algorithm steady state evaluation

Similarly to the simplified model, the two algorithms (EKF and SMO) have been compared in steady state behavior, using sensed control.

The speed and torque setpoint are fixed, and the simulation runs for 0.15s to let the dynamics become stable. The output of this test is the percentage error of the electric angle  $\theta_e$  observed VS the real value, and the standard deviation of the speed error (see subsection 6.1.3).

#### 6.3.2.1 Rotor angle error

The trend of the EKF to be the better performing algorithm is confirmed, as can be seen in Fig. 6.19. In Fig. 6.20, it is shown how the two algorithms have more difficulties near the maximum torque.

#### 6.3.2.2 Speed error standard deviation

As expected, speed observer performances of both algorithms get worse (Fig. 6.21a), due to the introduction of the switching inverter and the increase of sample time. It is possible to notice from the normalized plot in Fig. 6.21b, that lower speeds are the most difficult to observe.

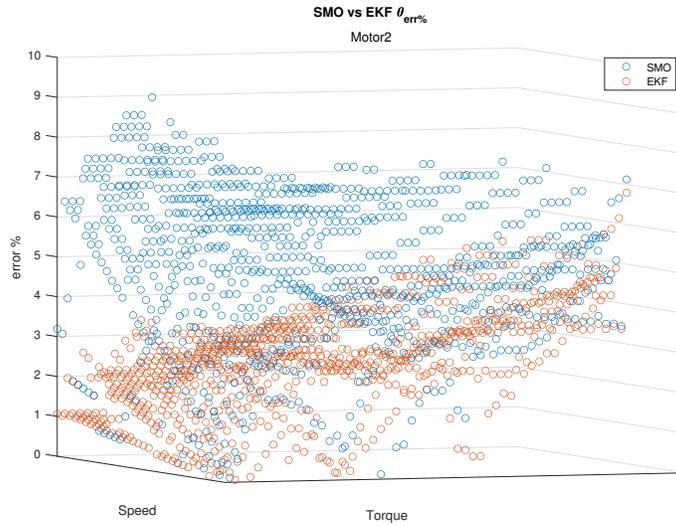
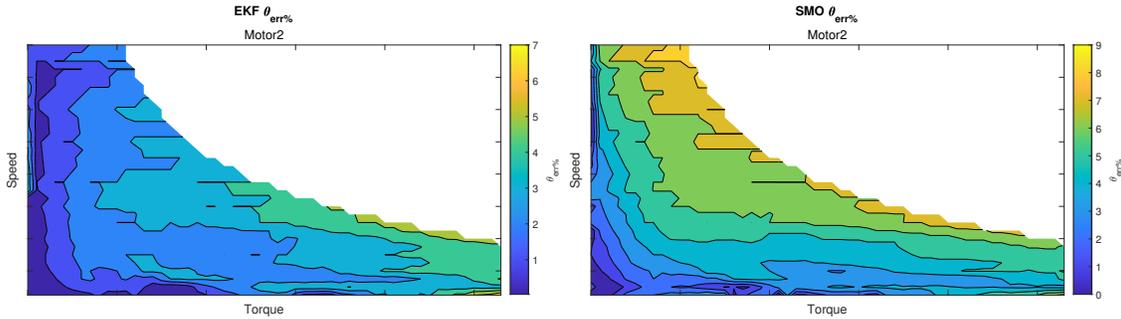


Figure 6.19: Motor 1: SMO VS EKF steady state angle error %.



(a) EKF angle error.

(b) SMO angle error.

Figure 6.20: Motor 1: SMO and EKF steady state angle error %.

### 6.3.3 Observer algorithm dynamic evaluation

In Fig. 6.23 the test that has implemented to benchmark the behavior or the algorithm in dynamic condition.

|   |            | EKF    | SMO    |
|---|------------|--------|--------|
| Electric angle mean error                 | [%]        | 3.5904 | 6.1398 |
| Electric angle mean error                 | [ $\mu$ s] | 108    | 162    |
| Mechanical speed error standard deviation | [RPM]      | 1304   | 2089   |

Table 6.15: Motor 1 full-sim - Real vs observer -  $T_m = 50$  N m - Sensored control.

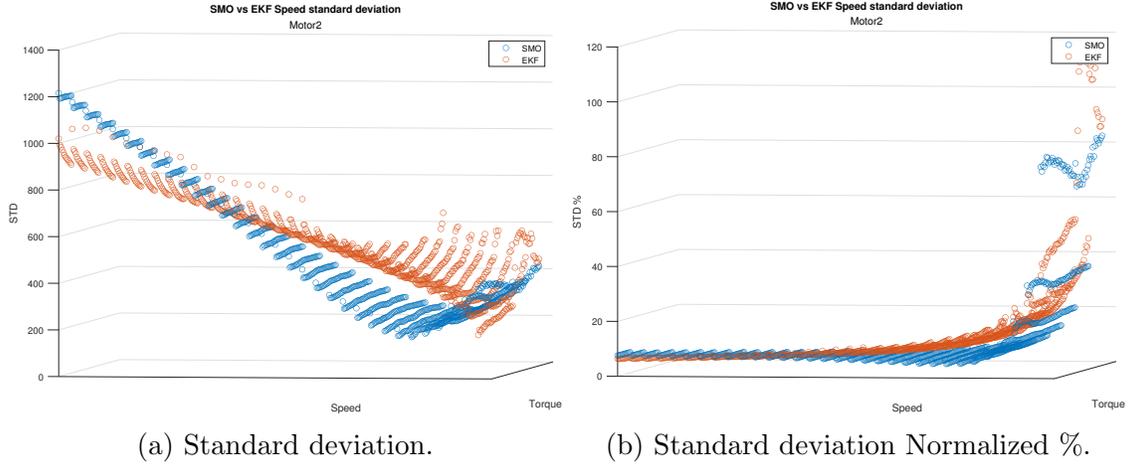


Figure 6.21: Motor 1: SMO VS EKF steady state speed error standard deviation.

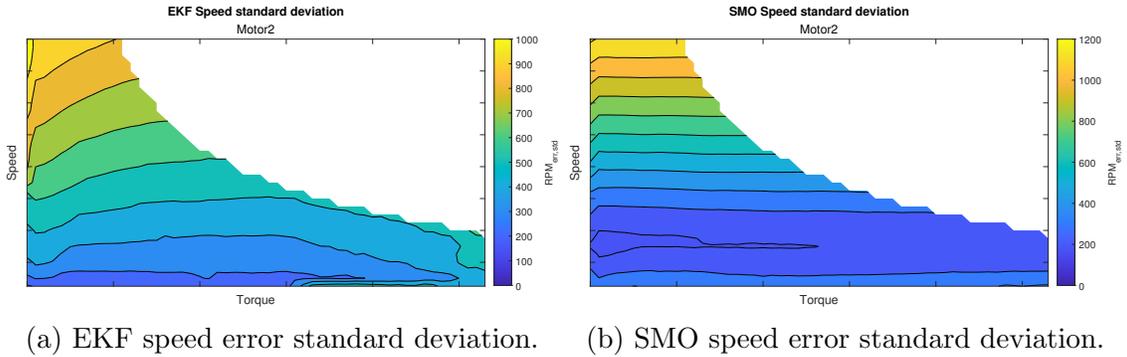


Figure 6.22: Motor 1: SMO and EKF steady state speed error standard deviation.

### 6.3.3.1 Sensorless

A test using sensorless control was arranged using the same speed and torque from the previous test (Fig. 6.23). As can be seen from the results in Tab. 6.16 and Fig. 6.7, both the algorithm worked properly, with an advantage for EKF

|   |            | EKF    | SMO    |
|---|------------|--------|--------|
| Electric angle mean error                 | [%]        | 3.1308 | 4.7648 |
| Electric angle mean error                 | [ $\mu$ s] | 101    | 142    |
| Mechanical speed error standard deviation | [RPM]      | 1305   | 2140   |

Table 6.16: Motor 1 full-sim - Real vs observer -  $T_m = 50$  N m - Sensorless control.

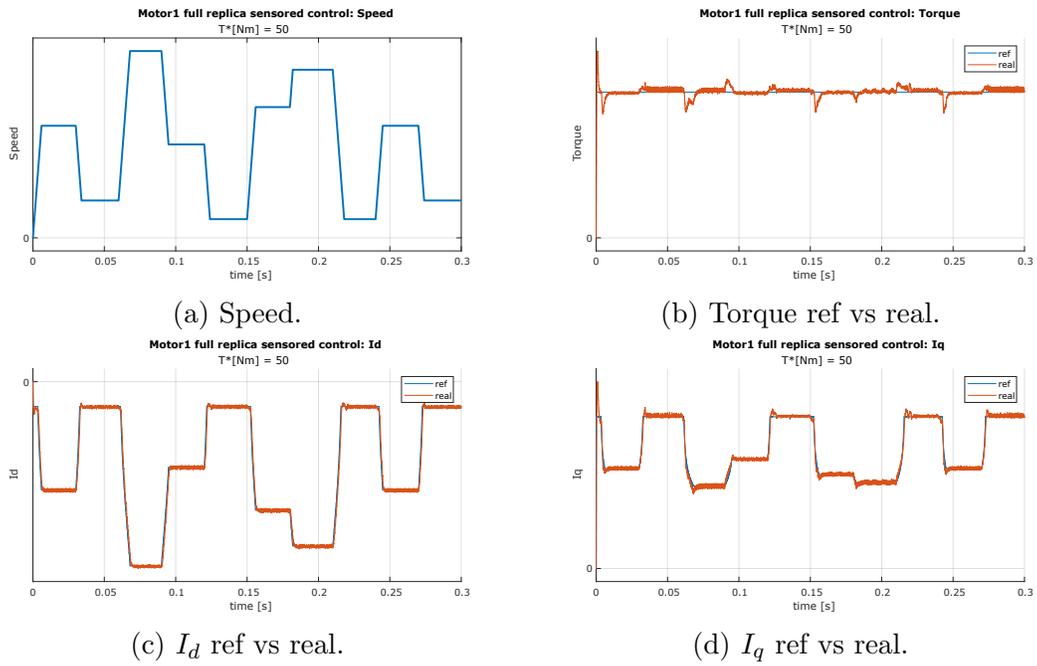


Figure 6.23: Motor 1 sensed dynamic test -  $T_m = 50$  N m.

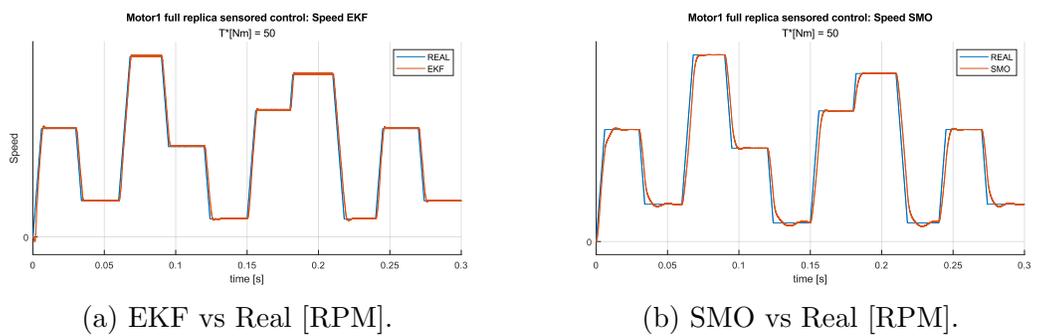


Figure 6.24: Motor 1 - Real speed vs observer -  $T_m = 50$  N m - Sensorless control.

### 6.4 Motor 2: Full-replica simulation

As for motor 1 The LUT implemented for the full-replica model of motor 2 are obtained by means of a FEM simulation of a real motor.

In Fig. 6.25 the implemented maps.

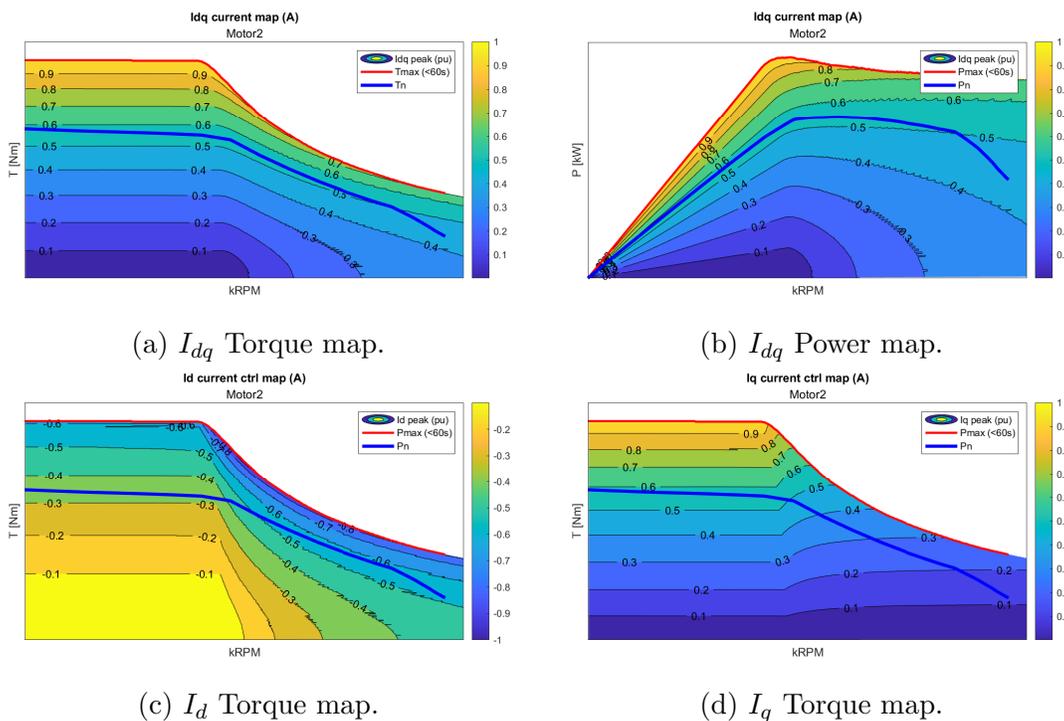


Figure 6.25:  $I_{dq}$  maps of switching motor 2.

#### 6.4.1 Parameters

In Tab. 6.17 the Simulink simulation parameter utilized, and in Tab. 6.18 the observers parameters.

|                    |   |
|--------------------|---|
| Fixed step size    | $1 \times 10^{-7}$ s                            |
| Control frequency  | $1 \times 10^{-6}$ s                            |
| PWM generator      | Sinusoidal PWM (SPWM) with min. max. injection. |
| PWM switching freq | 15 kHz  |

Table 6.17: Motor 2 full-replica simulation parameters.

|     |  |           |                      |
|-----|--|-----------|----------------------|
| SMO | OBS gain                                 | $K_s$     | 1500                 |
|     | Sigmoid constant                         | $a$       | 1.5                  |
|     | QPLL proportional gain                   | $K_p$     | 1                    |
|     | QPLL integrarl gain                      | $K_i$     | 800                  |
|     | Sample time                              | $T_s$     | $1 \times 10^{-5}$ s |
| EKF | System noise covariance matrix diag      | $Q$       | [11.9, 234, 2039, 0] |
|     | Measurement noise covariance matrix diag | $R$       | [7263, 2870]         |
|     | Compensator                              | $C_{omp}$ | 0.1                  |
|     | Sample time                              | $T_s$     | $1 \times 10^{-5}$ s |

Table 6.18: Motor 2 full-replica observer parameters.

### 6.4.2 Observer algorithm steady state evaluation

Similarly to the simplified model, the two algorithms (EKF and SMO) have been compared in steady state behavior, using sensed control.

The speed and torque setpoint are fixed, and the simulation runs for 0.15s to let the dynamics become stable. The output of this test is the percentage error of the electric angle  $\theta_e$  observed VS the real value, and the standard deviation of the speed error  $RPM_{err} = \omega_{obs} - \omega_{real}$ .

#### 6.4.2.1 Rotor angle error

The rotor angle error is better observed in almost any condition, as can be seen in Fig. 6.26. In Fig. 6.27, it is visible how the two algorithm have more difficulties near the maximum torque.

#### 6.4.2.2 Speed error standard deviation

As expected, speed observer performances of both algorithms get worse (Fig. 6.28a), due to the introduction of the switching inverter and the increase of sample time. It is possible to notice from the normalized plot in Fig. 6.28b, that lower speed are the most difficult to observe.

### 6.4.3 Observer algorithm dynamic evaluation

In Fig. 6.30 the test that has implemented to test the behavior or the algorithm in dynamic condition.

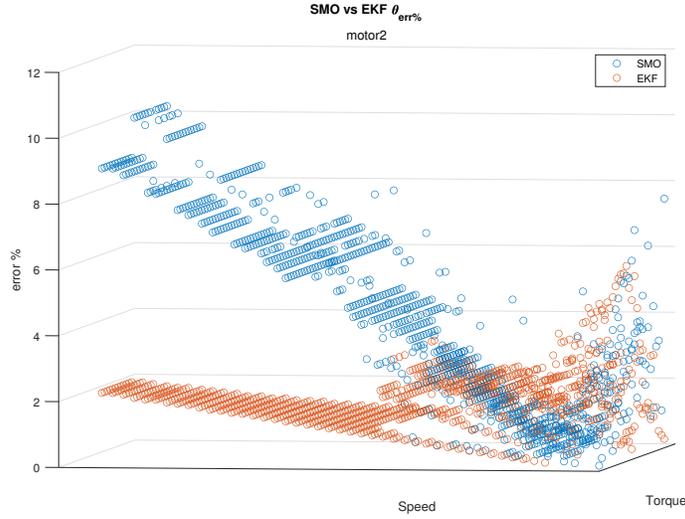
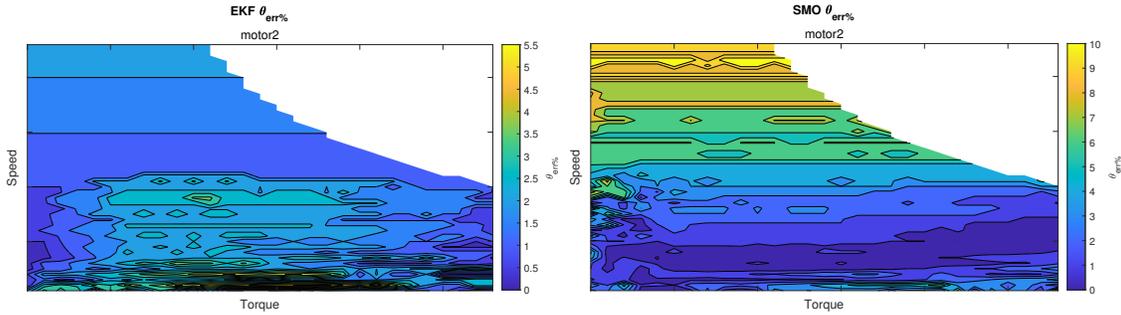


Figure 6.26: Motor 1: SMO VS EKF steady state angle error %.



(a) EKF angle error.

(b) SMO angle error.

Figure 6.27: Motor 1: SMO and EKF steady state angle error %.

|   |            | EKF    | SMO    |
|---|------------|--------|--------|
| Electric angle mean error                 | [%]        | 2.6282 | 5.9413 |
| Electric angle mean error                 | [ $\mu$ s] | 49     | 93     |
| Mechanical speed error standard deviation | [RPM]      | 498    | 1781   |

Table 6.19: Motor 2 full-sim - Real vs observer - No load - Sensored control.

### 6.4.3.1 Sensorless

As was done for motor 1, sensorless control test was arranged using the same speed and torque from the previous test (Fig. 6.30). As can be seen from the results in

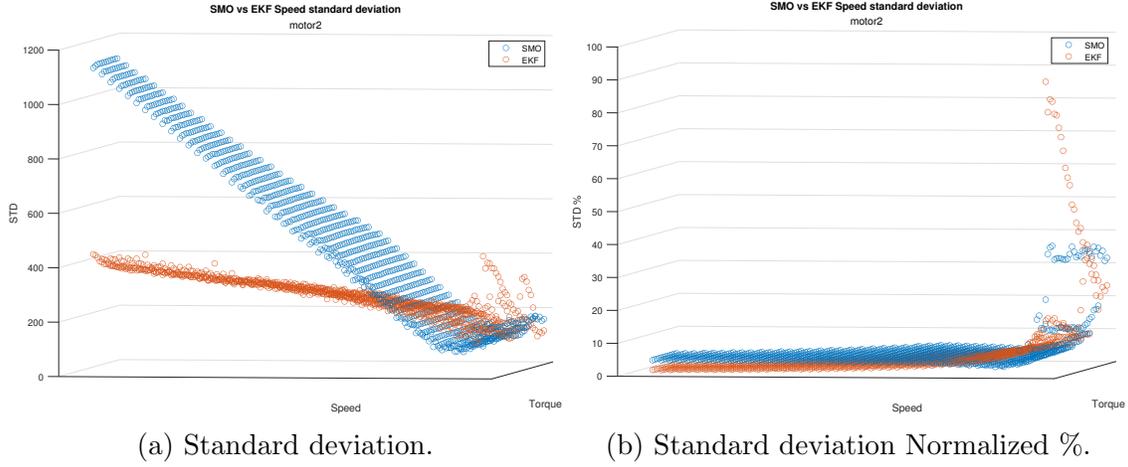


Figure 6.28: Motor 1: SMO VS EKF steady state speed error standard deviation.

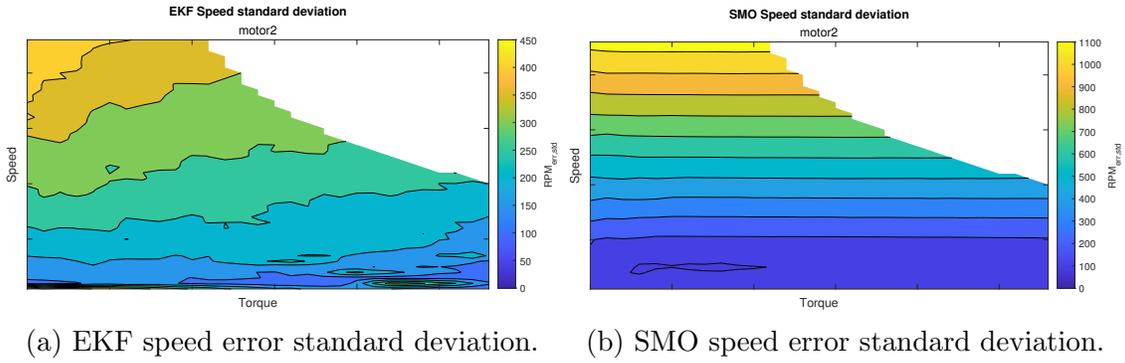


Figure 6.29: Motor 1: SMO and EKF steady state speed error standard deviation.

Tab. 6.20 and Fig. 6.15, both the algorithm worked properly, with an advantage for EKF

|   | EKF    | SMO    |
|---|--------|--------|
| Electric angle mean error [%]                   | 2.0241 | 6.1059 |
| Electric angle mean error [ $\mu$ s]            | 36     | 93     |
| Mechanical speed error standard deviation [RPM] | 493    | 1769   |

Table 6.20: Motor 2 full-sim - Real vs observer -  $T_m = 100$  N m - Sensorless control.

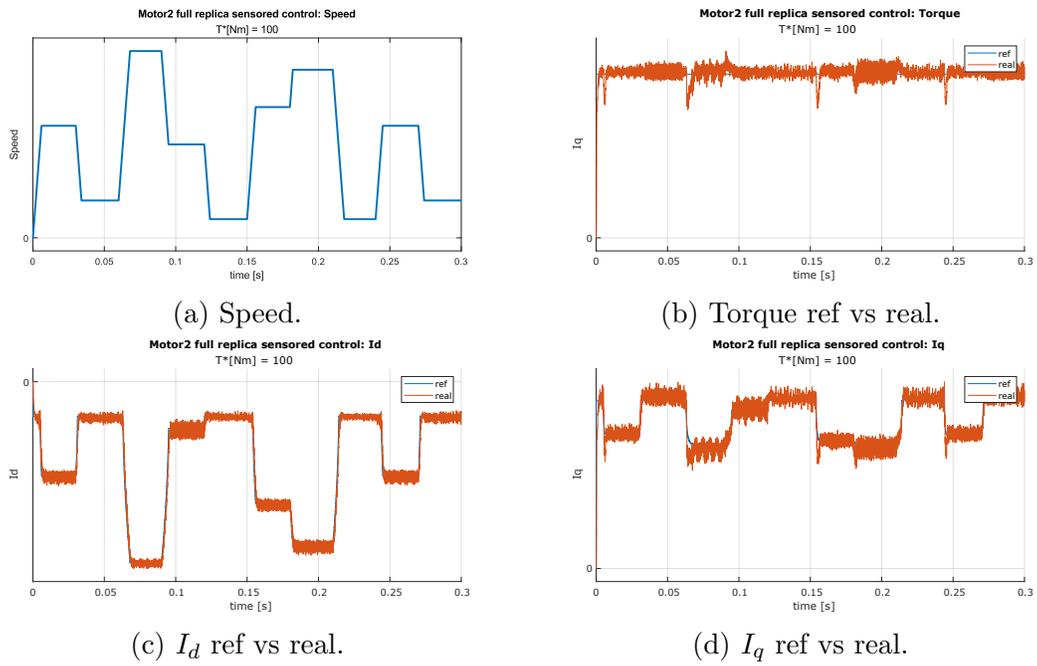


Figure 6.30: Motor 2 sensed dynamic test -  $T_m = 100$  N m.

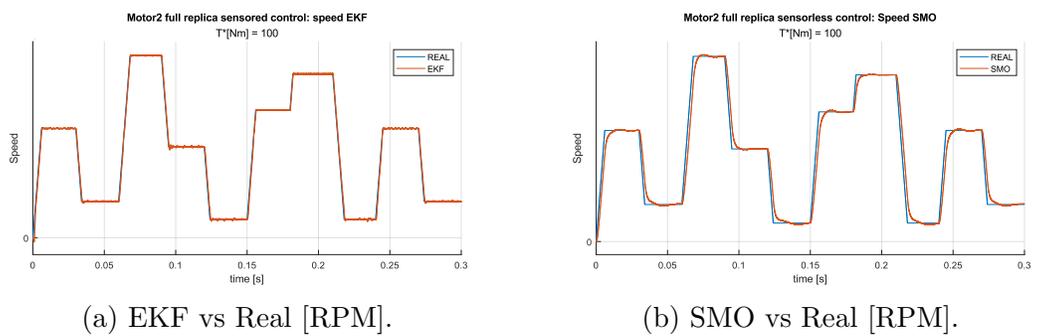


Figure 6.31: Motor 2 - Real speed vs observer -  $T_m = 100$  N m - Sensorless control.

## 6.5 Real Hardware test bench

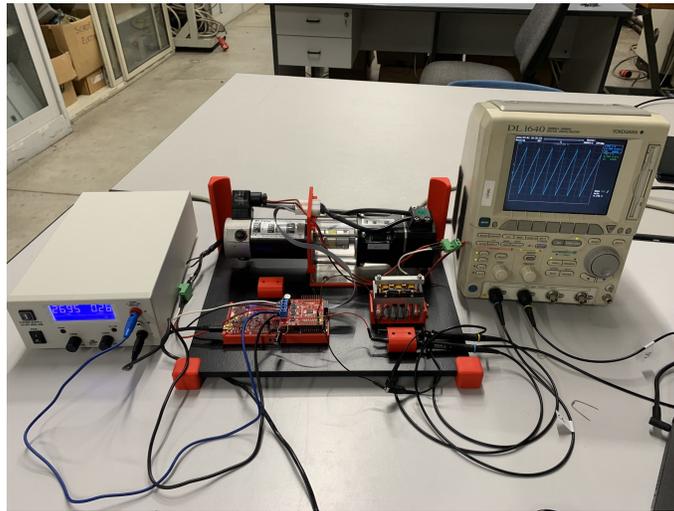


Figure 6.32: Full test bench wired up.

The real Hardware test bench was conducted in the university of Bologna. In Fig. 6.32 a photo of the test bench during a run. Different tests will be done, comparing SMO and EKF in steady state and dynamic conditions.

Results were collected using the oscilloscope.

### 6.5.1 Parameters

The motor parameters have been already presented in Tab. 5.1. Other useful parameters used in this thesis test are present in Tab. 6.21 and Tab. 6.22

|  |                           |
|--|---------------------------|
| MCU core frequency                               | 200 MHz                   |
| Control sample time                              | $1 \times 10^{-6}$ s      |
| Dead time  | $0.7 \times 10^{-6}$ s    |
| Mechanical radian per encoder pulse (resolution) | $7.67 \times 10^{-4}$ rad |

Table 6.21: Motor bench software parameters.

|     |  |       |                       |
|-----|--|-------|-----------------------|
| SMO | OBS gain                                 | $K_s$ | 200                   |
|     | Sigmoid constant                         | $a$   | 0.1                   |
|     | QPLL proportional gain                   | $K_p$ | 100                   |
|     | QPLL integral gain                       | $K_i$ | 1300                  |
|     | Sample time                              | $T_s$ | $1 \times 10^{-4}$ s  |
| EKF | System noise covariance matrix diag      | $Q$   | [11.93, 234, 2038, 0] |
|     | Measurement noise covariance matrix diag | $R$   | [7263, 2870]          |
|     | Sample time                              | $T_s$ | $1 \times 10^{-4}$ s  |

Table 6.22: Motor bench observer parameters.

### 6.5.2 Observer algorithm steady state sensed

Steady state condition is evaluated running the motor for a couple of seconds, and measuring the real angle readings from the encoder and the observer output. The following condition were utilized during this test:

|                |             |
|----------------|-------------|
| Resistive load | 20 $\Omega$ |
| Control        | sensored    |

Table 6.23: Observer algorithm steady state test condition.

Results were collected sensing the digital value of mechanical angle from the DSP by means if an oscilloscope (subsection 5.1.5). Results are available in appendix B; those are screenshots of the oscilloscope, where it is possible to approximately evaluate the difference between the two signals:

| Speed    | SMO  |      | EKF  |      |
|----------|------|------|------|------|
|          | ms   | %    | ms   | %    |
| 200 RPM  | 0.69 | 0.92 | 1.72 | 2.31 |
| 400 RPM  | 0.86 | 2.31 | 0.86 | 2.31 |
| 700 RPM  | 0.31 | 2.08 | 0.41 | 2.8  |
| 1000 RPM | 0.27 | 1.85 | 0.41 | 2.8  |
| 1500 RPM | 0.27 | 1.85 | 0.35 | 2.4  |
| 1800 RPM | 0.22 | 3    | 0.29 | 3.9  |

Table 6.24: Observer algorithm steady state results.

Despite the inaccuracy of the data, it is possible to notice that SMO error is always smaller than EKF.

### 6.5.3 Observer algorithm step change sensed

|                |             |
|----------------|-------------|
| Resistive load | 20 $\Omega$ |
| Control        | sensored    |

Table 6.25: Observer algorithm step test condition.

The second test consists in a rapid step change, from one speed reference to another.

The qualitative result can be seen in appendix C. It is possible to notice that the kalman observer is more reactive and accurate during the speed change.

### 6.5.4 Observer algorithm step change Sensorless

It was possible to arrange the same step change test mentioned previously using sensorless algorithm:

1. Motor speed is set to a fixed value until steady state is reached (init).
2. Sensorless control is activated.
3. Speed reference is changed with a step.

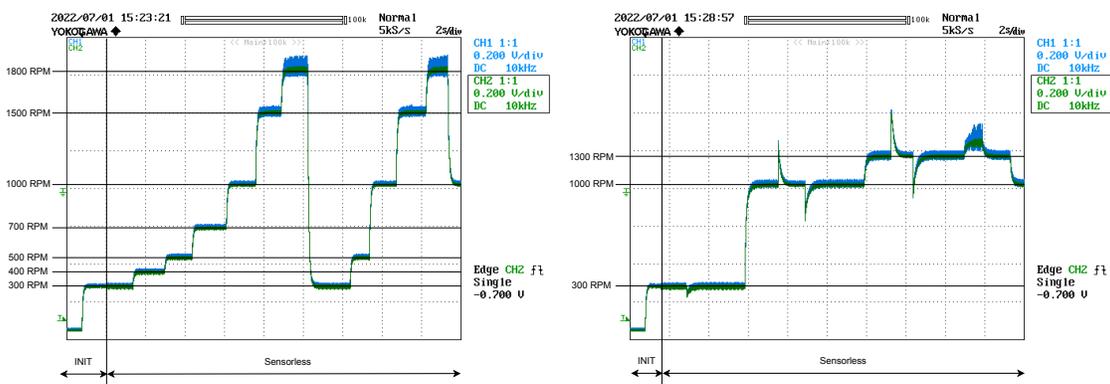
Qualitative results can be seen in appendix D.

Only EKF was able to accomplish this test. When the test was attempted using SMO, the motor control didn't work correctly.

### 6.5.5 Observer algorithm positive speed step change Sensorless

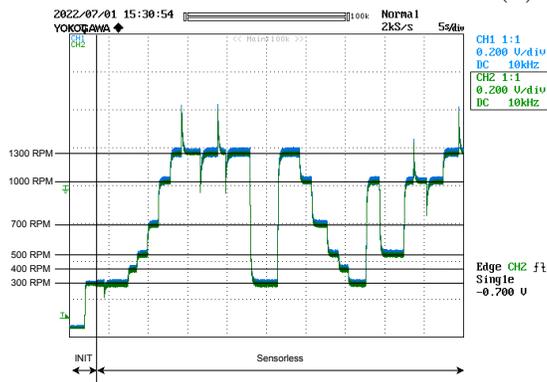
An additional step test was conducted in sensorless condition. During this test only positive speed were investigated. The load in this case is able to change, as it is highlighted in the result images:

1. Motor speed is set to a fixed value until steady state is reached (init).
2. Sensorless control is activated.
3. Speed reference is changed with series of step.
4. Load torque is changed during the test (if relay is connected as load).



(a) Test 1.

(b) Test 2.



(c) Test 3.

Figure 6.33: Positive step sensorless control test SMO (650RPM per 0.1V).

As can be seen from the results in Fig. 6.33, there were no particular problem to be highlighted during the test: the control algorithm was able to perform positive speed step change, and it was robust to the sudden load torque change.

### 6.5.5.1 Note on torque change

In Fig. 6.33c and Fig. 6.33b, it is possible to see some speed spikes during the test: this is due to the load change. The load is changed by switching on and off the relay (Fig. 5.4a), resulting in high or zero load condition (see Fig. 5.4b).

- Spike goes UP: load changed from HIGH LOAD to LOW LOAD.
- Spike goes DOWN: load changed from LOW LOAD to HIGH LOAD.

## 6.6 Conclusion

The first stage of the thesis activity was focused on the implementation of an observer algorithm on an electric motor. All the mathematical equation and models have been deeply explained, with every passage, until two observers algorithms were selected to be implemented and benchmarked.

A simplified model was provided to test, benchmark, and debug the observer algorithm that will be used. Two motors have been utilized to provide a comprehensive benchmark.

During the tests, unexpected complications were observed; different methods were implemented to solve them, such as compensation or the Kalman matrix optimization.

The next step was to implement the properly tuned version of the algorithm on a more complete motor and inverter simulation, using the full-replica model.

Finally, thanks to the UNIBO laboratories, it was possible to deploy the algorithm in real hardware; observer performances were evaluated in real condition, with limited computational resources.

The results of the provided tests shows different characteristics:

- Both SMO and EKF have to be meticulously tuned to work properly.
- In simulation, EKF resulted more effective with both angle and speed error.
- In real life test bench, SMO does observe the angle better. Nevertheless, EKF provided superior results when used in sensorless mode, performing tests that SMO was unable to do.

|            | SIM     | REAL    |
|------------|---------|---------|
| Sensored   | EKF>SMO | SMO>EKF |
| Sensorless | EKF>SMO | EKF>SMO |

Table 6.26: Summary result table.

To conclude, the output of this work shows that both SMO and EKF can be suitable in sensorless PMSM application, with an advantage for EKF. An additional thought is the implementation of such algorithm in high dynamics application: at the current state of the art, observer errors could negatively affect the performance of a sensorless controlled motor, even more if high dynamics are involved; despite that, it can be interesting to use such observers in parallel to a position sensor for two main reasons:

1. If the sensor breaks, the algorithm could still run the motor with lower performances, increasing the reliability of the motor drive.
2. Cheaper sensors can be utilized, which can be helped by the algorithm in tedious operating condition, such as high speed.

The work is open to future work, such as:

- Fine-tuning of the optimization algorithm.
- Fine-tuning of the output speed low pass filter.
- Optimization algorithm for the SMO.
- Observer fault detection, and fault protection.
- Code optimization for real hardware.
- Code testing and validation for safety certification.

## Glossary

**BEV** Battery Electric Vehicles 8

**DSC** Direct Self Control 11

**DTC** Direct Torque Control 11

**EKF** Extended Kalman Filter 45, 46, 48, 49, 51, 52, 67, 70–74, 76–80, 82–85, 87–91, 93, 95, 96

**EV** Electric Vehicles 1, 6, 8, 9, 31

**FCEV** Fuel Cell Electric Vehicles 8

**FEM** Finite Element Model 7, 53, 57, 58, 68, 75, 81, 86

**FF** Feed Forward 33

**FOC** Field Oriented Control 1, 10, 11, 31, 33, 61, 66

**HDV** Heavy Duty Vehicles 1, 8, 9

**HEV** Hybrid Electric Vehicles 8

**ICE** Internal Combustion Engine 6, 8

**IPM** Internal Permanent Magnet 13, 36

**LDV** Light Duty Vehicles 1, 8, 10

**LUT** Lookup Table 31, 32, 53, 55, 57, 58, 81, 86

**MTPA** Maximum Torque Per Ampere 31, 32, 53, 55, 58

**MTPV** Maximum Torque Per Volt 31, 32, 34, 53, 55, 58

**PHEV** Plug-in Hybrid Electric Vehicles 8

**PLL** Phase-Locked Loops 38

**PMSM** Permanent Magnet Synchronous Machine 1, 6, 9–13, 17, 21–31, 42, 45, 96

**PWM** Pulse Width Modulation 1, 32, 34, 56–58, 67

**Q-PLL** Quadrature-component-based Phase-Locked Loop 38, 39, 41

**SMO** Sliding Mode Observer 35–37, 40, 41, 67, 70–74, 76–80, 82–85, 87–91, 93, 95, 96

**SPM** Surface Permanent Magnet 1, 2, 13, 17–21, 26, 30, 35, 60, 61

**SPWM** Sinusoidal PWM 34, 56, 57, 81, 86

**SQP** Sequential Quadratic Programming 50

**SVM** Space vector modulation 57

**SVPWM** Space Vector PWM 34

**VSC** Voltage Source Control 56

## References

- [1] N. Rossi, N. Matteazzi, G. L. Petti, L. Fazzini, S. Nuzzo, D. Barater, and G. Franceschini. Design and thermal assessment of a high performance electric motor for racing applications. In *2021 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)*, pages 52–57, 2021.
- [2] **Council of the EU** fit for 55 package: Council reaches general approaches relating to emissions reductions and their social impacts. <https://www.consilium.europa.eu/en/press/press-releases/2022/06/29/fit-for-55-council-reaches-general-approaches-relating-to-emissions-reductions-and-removals-and-their-social-impacts/>. Last accessed September 26, 2022.
- [3] **The EVC1000 project**. <http://www.evc1000.eu/en>. Last accessed September 26, 2022.
- [4] **TRANSFORM Trusted European SiC Value Chain for a greener Economy**. <https://sic-transform.eu/en>. Last accessed September 26, 2022.
- [5] Mehrdad Ehsani, Krishna Veer Singh, Hari Om Bansal, and Ramin Tafazzoli Mehrjardi. State of the art and trends in electric and hybrid electric vehicles. *Proceedings of the IEEE*, 109(6):967–984, 2021.
- [6] Yasser A Alamoudi, Azzeddine Ferrah, Raghu Panduranga, Ahmed Althobaiti, and F Mulolani. State-of-the art electrical machines for modern electric vehicles. In *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pages 1–8, 2019.
- [7] Sadegh Vaez-Zadeh. *Control of Permanent Magnet Synchronous Motors*, Oxford University Press, 2018. 05 2018.
- [8] Christoph Datlinger and Mario Hirz. Benchmark of rotor position sensor technologies for application in automotive electric drive trains. *Electronics*, 2020, 06 2020.
- [9] Gaolin Wang, Maria Valla, and Jorge Solsona. Position sensorless permanent magnet synchronous machine drives—a review. *IEEE Transactions on Industrial Electronics*, 67(7):5830–5842, 2020.
- [10] R. Bojoi, M. Pastorelli, J. Bottomley, P. Giangrande, and C. Gerada. Sensorless control of pm motor drives — a technology status review. In *2013 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)*, pages 168–182, 2013.

- [11] **HPE COXA** homepage. <https://www.hpe.eu/>. Last accessed September 26, 2022.
- [12] **IEA** global ev outlook 2022, iea, paris. <https://www.iea.org/reports/global-ev-outlook-2022>. Last accessed September 26, 2022.
- [13] Joeri Van Mierlo, Maitane Berecibar, Mohamed El Baghdadi, Cedric De Cauwer, Maarten Messagie, Thierry Coosemans, Valéry Ann Jacobs, and Omar Hegazy. Beyond the state of the art of electric vehicles: A fact-based paper of the current and prospective electric vehicle technologies. *World Electric Vehicle Journal*, 12(1), 2021.
- [14] **Law Insider** light duty vehicle(s) definition. <https://www.lawinsider.com/dictionary/light-duty-vehicles>. Last accessed September 26, 2022.
- [15] **European Commission** questions and answers on the commission strategy for reducing heavy-duty vehicles' (hdvs) fuel consumption and co2 emissions. [https://ec.europa.eu/commission/presscorner/detail/en/MEMO\\_14\\_366](https://ec.europa.eu/commission/presscorner/detail/en/MEMO_14_366). Last accessed September 26, 2022.
- [16] **Governemnt of India** types of electric vehicles. <https://e-amrit.niti.gov.in/types-of-electric-vehicles>. Last accessed September 26, 2022.
- [17] John Reimers, Lea Dorn-Gomba, Christopher Mak, and Ali Emadi. Automotive traction inverters: Current status and future trends. *IEEE Transactions on Vehicular Technology*, 68(4):3337–3350, 2019.
- [18] Wesam Taha, Peter Azer, Alan Dorneles Callegaro, and Ali Emadi. Multiphase traction inverters: State-of-the-art review and future trends. *IEEE Access*, 10:4580–4599, 2022.
- [19] David Ocen. Direct torque control of a permanent magnet synchronous motor. Master's thesis, KTH, School of Electrical Engineering (EES), 2005.
- [20] Daniel Montesinos-Miracle, P. D. Chandana Perera, Samuel Galceran-Arellano, and F. Blaabjerg. *Sensorless V/f Control of Permanent Magnet Synchronous Motors*. 01 2010.
- [21] Cheng Aijun and Jin Xinhai. A stable v/f control method for permanent magnet synchronous motor drives. In *2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, pages 1–5, 2017.
- [22] A. Borisavljevic, H. Polinder, and J. A. Ferreira. Realization of the i/f control method for a high-speed permanent magnet motor. In *The XIX International Conference on Electrical Machines - ICEM 2010*, pages 1–6, 2010.

- [23] **IMPERIX** i-f startup method for sensorless pmsm control. <https://imperix.com/doc/implementation/i-f-startup-method>. Last accessed September 26, 2022.
- [24] Mirza Begh and Hans-Georg Herzog. Comparison of field oriented control and direct torque control, 03 2018.
- [25] **Matlab** homepage. <https://it.mathworks.com/products/matlab.html>. Last accessed September 26, 2022.
- [26] **Simulink** homepage. <https://it.mathworks.com/products/simulink.html>. Last accessed September 26, 2022.
- [27] **Plecs** homepage. <https://www.plexim.com/products/plecs>. Last accessed September 26, 2022.
- [28] Prabha Kundur. *Power System Stability and Control*. McGraw-Hill Education, 1994.
- [29] Paul C. Krause. *Analysis of Electric Machinery and Drive Systems, 3rd Edition*. Wiley-IEEE Press, 2013.
- [30] Zhendong Zhang. "permanent magnet synchronous machine based traction drive design for hybrid scooter considering control nonlinearities and compensations". 2013.
- [31] Maria Laura Bacci, Ferdinando Luigi Mapelli, Stefano Mossina, Davide Tarantino, and Michele Vignati. Wide-speed range sensorless control of an ipm motor for multi-purpose applications. *Inventions*, 5(3), 2020.
- [32] Tore Hägglund Karl Johan Åström. *Advanced PID Control*. Isa, 2006.
- [33] Dmitry Lukichev and Galina Demidova. Features of tuning strategy for field oriented control of pmsm position drive system with two-mass load. 10:88–94, 01 2016.
- [34] **IMPERIX** svpwm vs spwm modulation techniques. <https://imperix.com/doc/implementation/svpwm-vs-spwm-modulation-techniques>. Last accessed September 26, 2022.
- [35] **microchip** overmodulation management for field-oriented control of three-phase motors. <https://microchipdeveloper.com/motor:overmodulation>. Last accessed September 26, 2022.

- [36] Jin-Sik Park, Shin-Myoung Jung, Hag-Wone Kim, and Myoung-Joong Youn. Current control of pmsm in overmodulation region. In *2007 7th International Conference on Power Electronics*, pages 1062–1065, 2007.
- [37] **imperix** sliding mode observer rotor position estimation. <https://imperix.com/doc/implementation/sensorless-motor-control>. Last accessed September 26, 2022.
- [38] **IGI Global** what is state observer. <https://www.igi-global.com/dictionary/state-observer/28181>. Last accessed September 26, 2022.
- [39] Roland E Best. *Phase Locked Loops*. McGraw-Hill Professional Publishing, 2007.
- [40] **analog** phase-locked loop (pll) fundamentals. <https://www.analog.com/en/analog-dialogue/articles/phase-locked-loop-pll-fundamentals.html>. Last accessed September 26, 2022.
- [41] Yue Zhao, Chun Wei, Zhe Zhang, and Wei Qiao. A review on position/speed sensorless control for permanent-magnet synchronous machine-based wind energy conversion systems. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 1(4):203–216, 2013.
- [42] V. I. Utkin J. Guldner. The chattering problem in sliding mode systems, 2021.
- [43] **STMicroelectronics** luenberger state observer rotor position estimation. [https://www.st.com/resource/en/application\\_note/cd00154076-luenberger-state-observer-rotor-position-estimation-simulink-and-software-library-stmicroelectronics.pdf](https://www.st.com/resource/en/application_note/cd00154076-luenberger-state-observer-rotor-position-estimation-simulink-and-software-library-stmicroelectronics.pdf). Last accessed September 26, 2022.
- [44] Ming Qiu. Extended Kalman Filter Application in Permanent Magnet Synchronous Motor Sensorless Control. 5 2021.
- [45] Bolognani, S. and Oboe, R. and Zigliotto, M. Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position. *IEEE Transactions on Industrial Electronics*, 46(1):184–191, 1999.
- [46] Jae-Hoon Kim, Sang soo Lee, and Dong-Seok Hyun. Sensorless control of an ipm synchronous motor with extended kalman filter. In *2012 IEEE Vehicle Power and Propulsion Conference*, pages 13–18, 2012.
- [47] Peter Vas. *Sensorless vector and direct torque control*. Oxford University Press, Oxford, 1998.
- [48] Jean-Paul Louis (Editor). *Control of Synchronous Motors*. Wiley-ISTE, 2013.

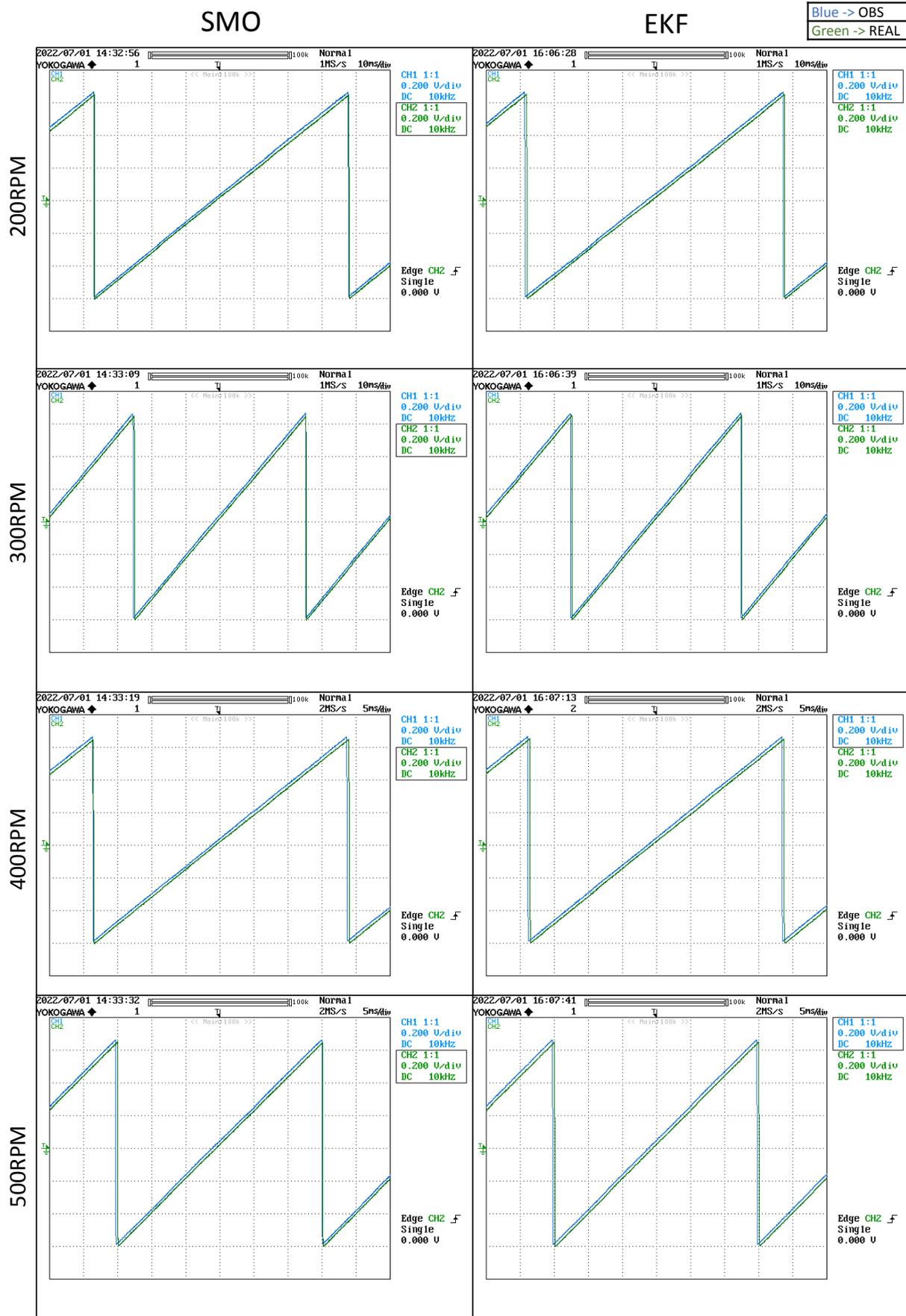
- [49] P. M. J. Van den Hof R. Toth, P. S. C. Heuberger. On the discretization of lpv state-space representations. <https://rolandtoth.eu/wp-content/uploads/2018/07/IET2010.pdf>. Last accessed September 26, 2022.
- [50] Yahia Laamari, Kheireddine Chafaa, and Belkacem Athamena. Particle swarm optimization of an extended kalman filter for speed and rotor flux estimation of an induction motor drive. *Electrical Engineering*, 97:129–138, 06 2015.
- [51] K.L. Shi, T.F. Chan, Y.K. Wong, and S.L. Ho. Speed estimation of an induction motor drive using an optimized extended kalman filter. *IEEE Transactions on Industrial Electronics*, 49(1):124–133, 2002.
- [52] Vito Mario Fico, César Pecharromán Arribas, Álvaro Ricca Soaje, Maria Angeles Martin Prats, Sebastián Ramiro Utrera, Antonio Leopoldo Rodriguez Vazquez, and Luis Miguel Parrilla Casquet. Implementing the unscented kalman filter on an embedded system: A lesson learnt. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 2010–2014, 2015.
- [53] Justas Dilys, Voitech Stankevič, and Krzysztof Łuksza. Implementation of extended kalman filter with optimized execution time for sensorless control of a pmsm using arm cortex-m3 microcontroller. *Energies*, 14(12), 2021.
- [54] MIT nonlinear programming - chapter 13. <https://web.mit.edu/15.053/www/AMP-Chapter-13.pdf>. Last accessed September 26, 2022.
- [55] K. Schittowski. Nlqpl: A fortran-subroutine solving constrained nonlinear programming problems. *Operations Research*, 5:485–500, 01 1985.
- [56] Stephen J. Wright Jorge Nocedal. *Numerical Optimization*. Springer Nature, 2006.
- [57] **mathworks** fmincon (find minimum of constrained nonlinear multivariable function). <https://it.mathworks.com/help/optim/ug/fmincon.html>. Last accessed September 26, 2022.
- [58] **Simscape** permanent magnet synchronous machine. <https://it.mathworks.com/help/physmod/sps/powersys/ref/permanentmagnetsynchronousmachine.html>. Last accessed September 26, 2022.
- [59] **Plecs** lookup table-based permanent-magnet synchronous machine. <https://www.plexim.com/content/lookup-table-based-permanent-magnet-synchronous-machine>. Last accessed September 26, 2022.

- [60] **TEM** btss 1524 three-phase motor. [https://www.tem-electric-motors.com/en/products/servomotors/servo.php?category=bts&model=BTSS&productid=25&dataFlag=0&productco=0&constructionco=0&optionsco=0#!#q\\_series\\_full\\_spec](https://www.tem-electric-motors.com/en/products/servomotors/servo.php?category=bts&model=BTSS&productid=25&dataFlag=0&productco=0&constructionco=0&optionsco=0#!#q_series_full_spec). Last accessed September 26, 2022.
- [61] **Texas Instruments** c2000 delfino mcu f28379d launchpad development kit. <https://www.ti.com/tool/LAUNCHXL-F28379D>. Last accessed September 26, 2022.
- [62] **Texas Instruments** drv8305n 3-phase motor drive boosterpack evaluation module. <https://www.ti.com/tool/BOOSTXL-DRV8305EVM>. Last accessed September 26, 2022.
- [63] **Elektro-Automatik** ea-ps 2042-20b dc laboratory power supply. <https://elektroautomatik.com/shop/en/products/programmable-dc-laboratory-power-supplies/dc-laboratory-power-supplies/series-ps-2000b-br-100-up-to-332-w/727/power-supply-0...42v/0..20a/320w?c=623>. Last accessed September 26, 2022.
- [64] **yokogawa** dl1640/dl1640l digital oscilloscopes. <https://www.yokogawa.com/library/resources/yokogawa-technical-reports/dl1640-dl1640l-digital-oscilloscopes/>. Last accessed September 26, 2022.
- [65] **Texas Instruments** code composer studio. <https://www.ti.com/tool/CCSTUDIO>. Last accessed September 26, 2022.
- [66] **Texas Instruments** control suite. <https://www.ti.com/tool/CONTROLSUITE>. Last accessed September 26, 2022.
- [67] **Texas Instruments** c2000ware. <https://www.ti.com/tool/C2000WARE>. Last accessed September 26, 2022.
- [68] **Mathworks** simulink ti c2000 support from embedded coder. <https://it.mathworks.com/hardware-support/ti-c2000-embedded-coder.html>. Last accessed September 26, 2022.

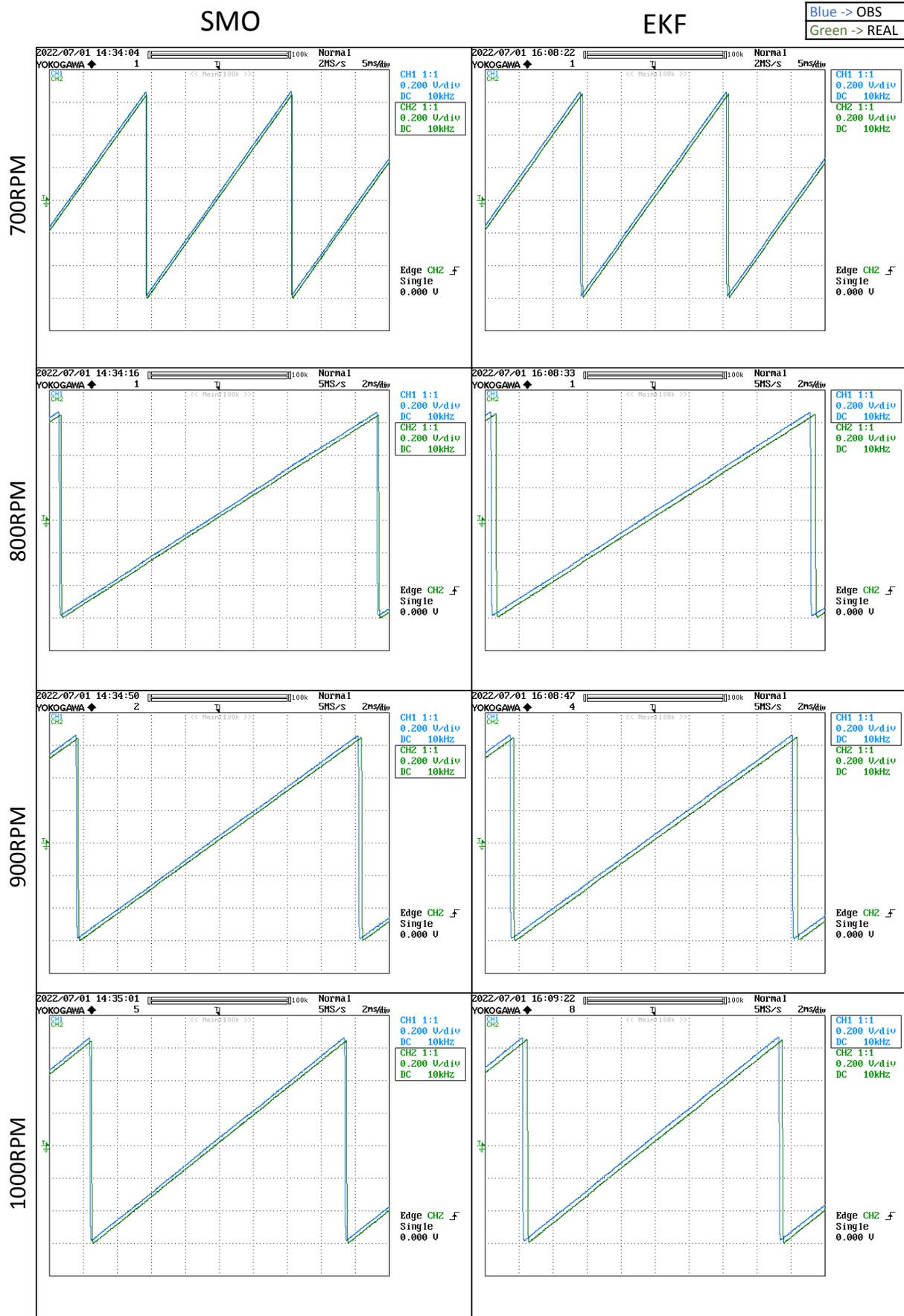


# Appendix B

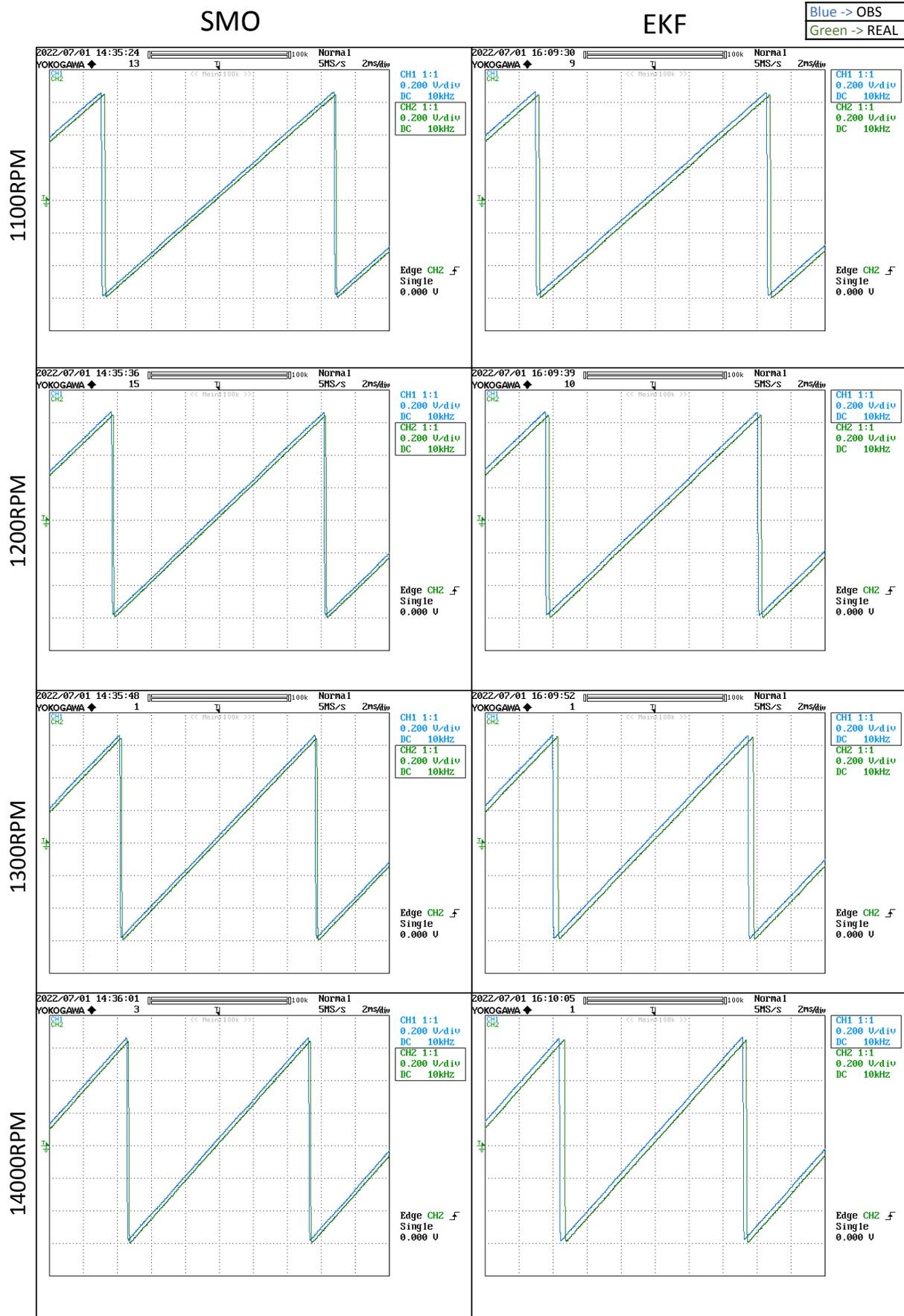
## Real test bench steady state result (1)



# Real test bench steady state result (2)

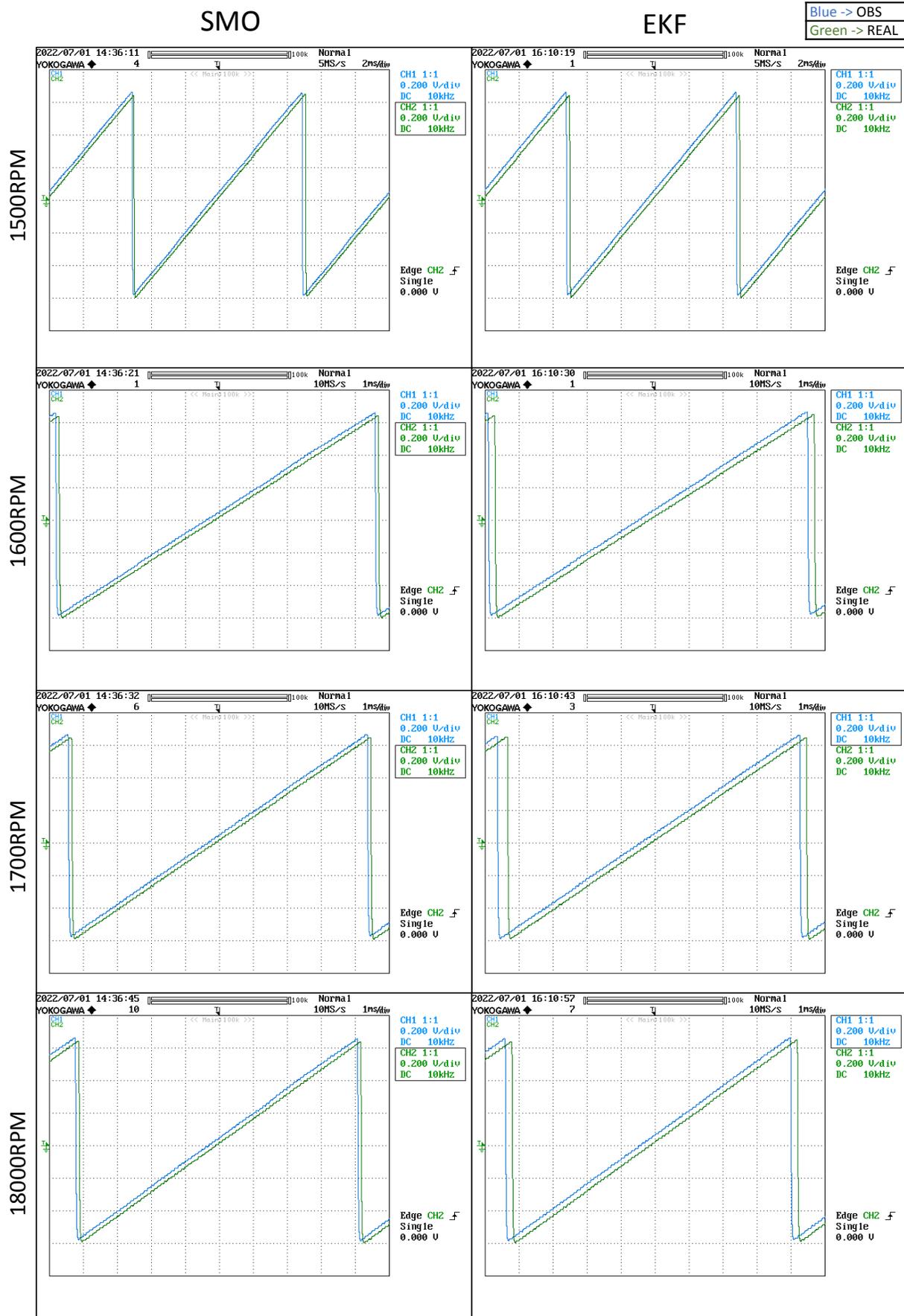


# Real test bench steady state result (3)



Blue -> OBS  
Green -> REAL

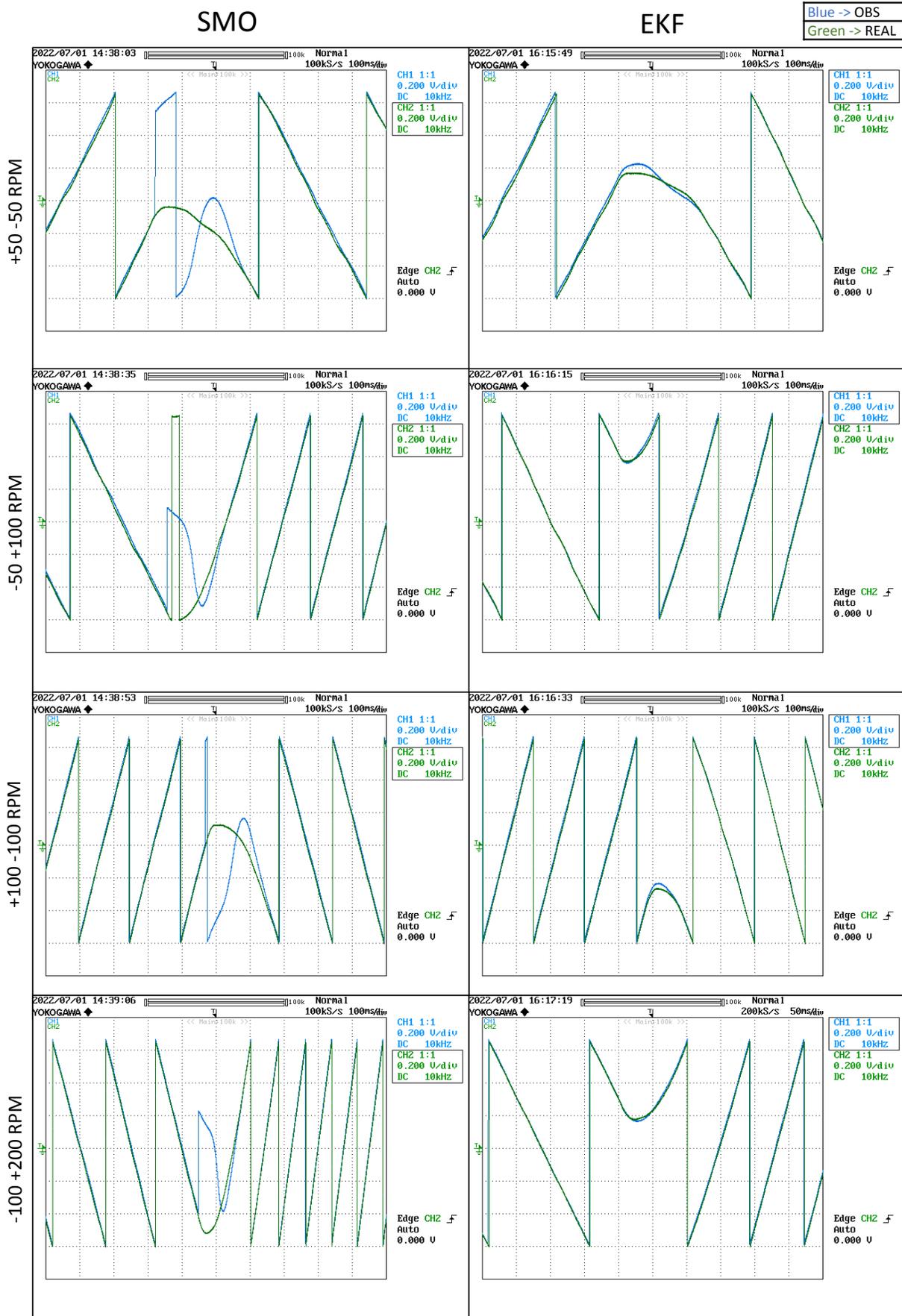
# Real test bench steady state result (4)



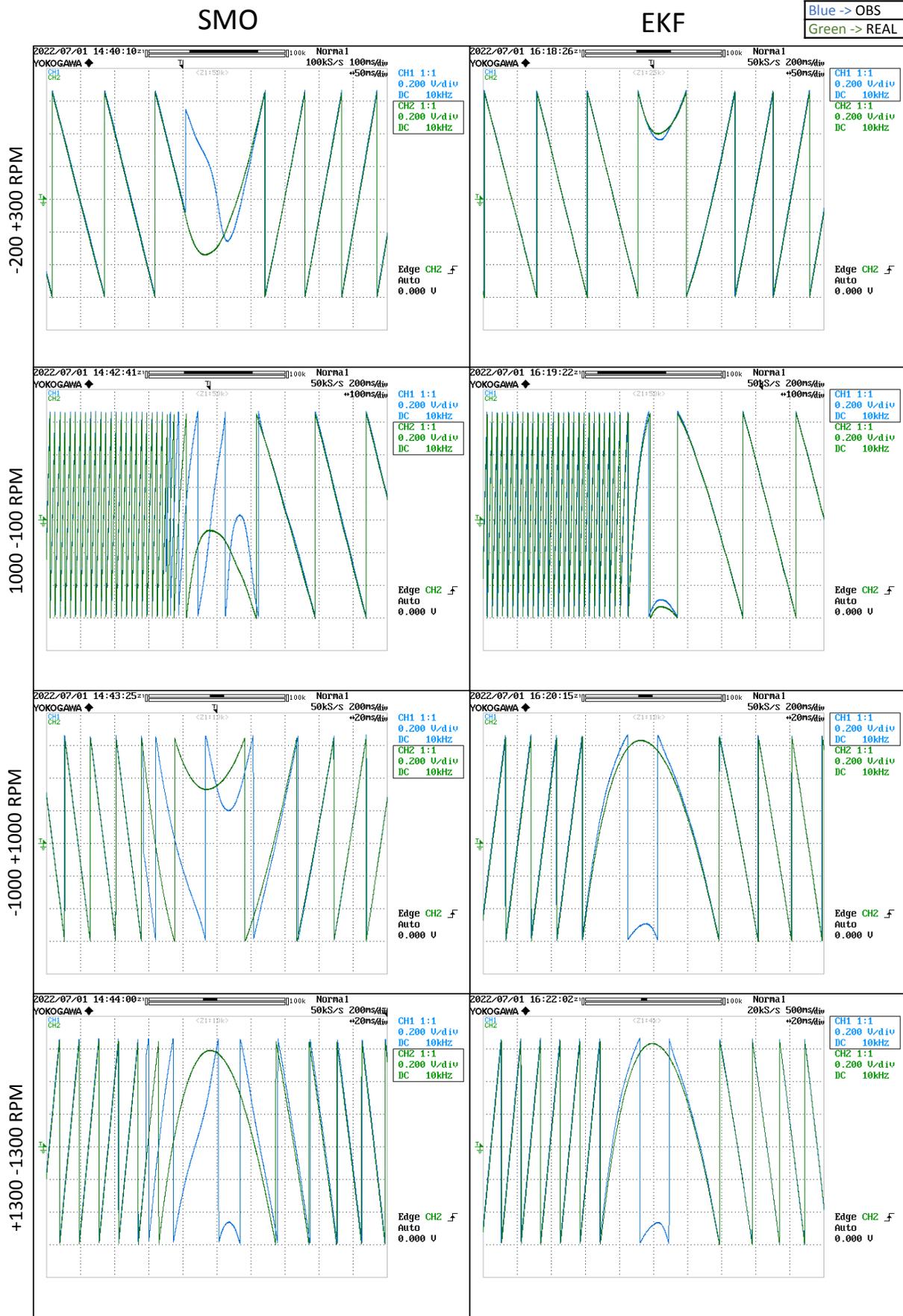
Blue -> OBS  
Green -> REAL

# Appendix C

## Real test bench step result (1)



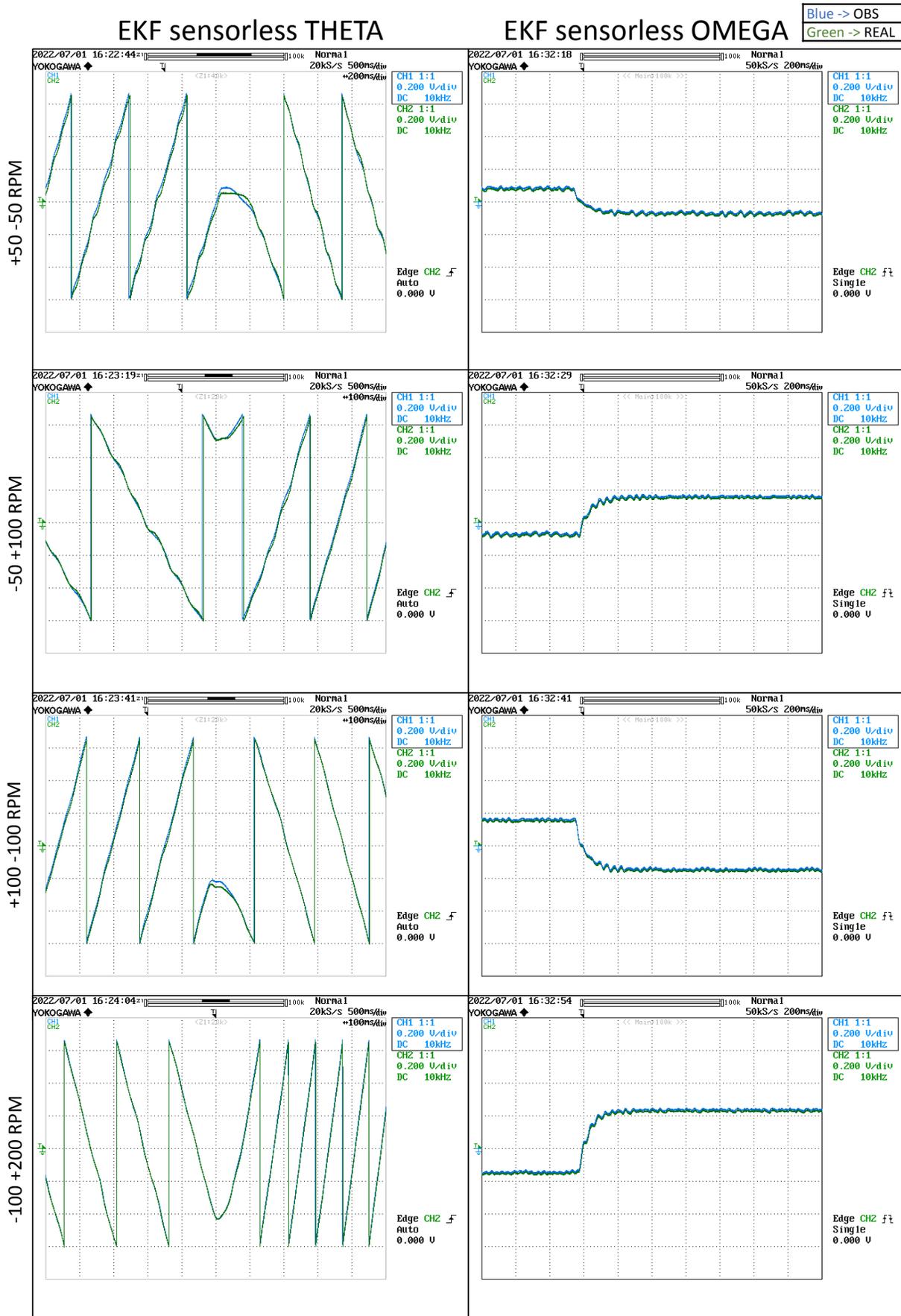
# Real test bench step result (2)



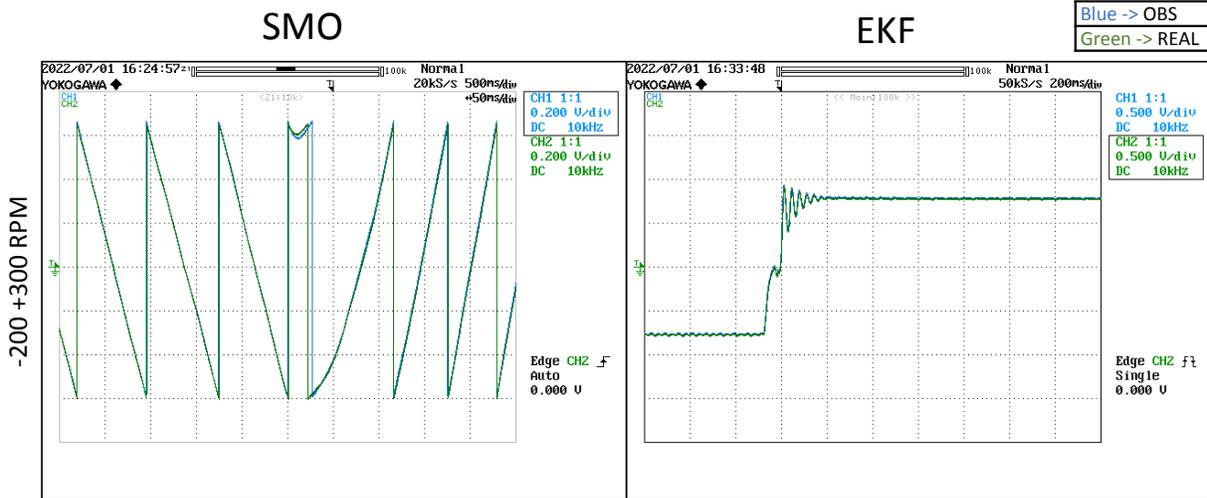
Blue -> OBS  
Green -> REAL

# Appendix D

## Real test bench step sensorless EKF result (1)



# Real test bench step sensorless EKF result (2)



## EKF sensorless OMEGA

