Real-time control system for a permanent magnet synchronous machine

Daniele Squarcio

School of Electrical Engineering
Final Project
Espoo December 2017

Supervisor
Prof. Anouar Belahcen

Advisors
Prof. Francesco Musolino
"Learn from yesterday, live for today, hope for tomorrow.

The important thing is not to stop questioning."

Albert Einstein
Acknowledgments

I would like to say thanks to all the people that support me in the realization of this work, carried out at Aalto University, Department of Electrical Engineering and Automation. Starting from people of Aalto University, in particular the professor Anouar Belahcen that gave me the opportunity to realize this thesis and supported me with availability and competence towards achieving the goals. The M.Sc. Mehnaz Farzam Far and D.Sc. Floran Martin that have been my instructors, they teach me a lot along this path. Thank also M.Sc. Awan Hafiz Asad Ali and D.Sc. Seppo Saarakkala for introducing me the dSPACE environment and supporting me in the laboratory and the group of Electric Drives head by Prof. Marko Hinkkanen for allowing me to use the dSPACE system and the control software. Thanks to all the new colleagues and friends that I met in this interesting and exciting experience, in which I increase my knowledge learning a lot. Thanks also to my supervisor at Politecnico Prof. Francesco Musolino that gave me support and many precious advices to realize my thesis.

I want also to say thanks to whole my family, which gave me the opportunity to do what I felt right for me by giving me unconditional support. Thanks to all my friends, the old ones with which I shared joys and sorrows along this tortuous path that is the life, and the new ones that I met along this last few years in Turin. Each of them was important in my path of personal and professional growth.
Table of contents

Acknowledgments .............................................. II

1 Introduction ................................................. 1

2 The inverter .................................................... 6
   2.1 Single-phase inverter ........................................ 6
       2.1.1 PWM Inverter control .................................... 7
   2.2 Three-phase inverter ......................................... 9
       2.2.1 Space-vector modulation ................................ 10
       2.2.2 Deadtime ................................................. 13

3 Permanent Magnets Motor ................................... 14
   3.1 Electric motors .............................................. 14
       3.1.1 Induction motor ......................................... 15
       3.1.2 Permanent Magnet motors .............................. 16
   3.2 Analysis of Permanent Magnet motors ....................... 17
       3.2.1 PM overview ............................................ 17
       3.2.2 Demagnetization curve ................................ 18
       3.2.3 Permanent magnet materials ............................ 20
       3.2.4 Permanent Magnet Synchronous motor ................. 22
       3.2.5 Different kinds of PM motors ......................... 24
       3.2.6 Dynamic model of PM motors .......................... 26
   3.3 IPM motor used ............................................. 27

4 Control ......................................................... 30
   4.1 Control of the system ..................................... 30
       4.1.1 Controller design ..................................... 33
       4.1.2 Scalar control ......................................... 35
       4.1.3 Vector control ......................................... 36
   4.2 Model Order Reduction (MOR) ................................ 36
       4.2.1 Singular value decomposition (SVD) .................. 37
4.2.2 Using the Orthogonal interpolation method ........................................ 37
4.3 Park-Clark transformation ........................................................................ 39
4.3.1 Clark transformation ........................................................................... 39
4.3.2 Park transformation .............................................................................. 40

5 Proposed solution ......................................................................................... 42
5.1 Introduction to RTI systems ..................................................................... 42
5.2 Simulink model ......................................................................................... 43
  5.2.1 Motor model ......................................................................................... 44
  5.2.2 Control model ....................................................................................... 47
  5.2.3 Zero-order hold (ZOH) ......................................................................... 50
  5.2.4 Simulation results for the basic scheme ................................................ 51
5.3 Simulink model with Model order reduction .............................................. 54
  5.3.1 Simulation results ................................................................................. 58

6 Experimental results ..................................................................................... 60
6.1 Measurement setup ................................................................................... 60
6.2 Simulink model for dSPACE ................................................................... 64
  6.2.1 Blocks analysis ..................................................................................... 64
  6.2.2 Running mode ....................................................................................... 68
  6.2.3 Alignment mode .................................................................................... 68
  6.2.4 Control desktop interface .................................................................... 69
6.3 Results of laboratory setup ...................................................................... 71
  6.3.1 Speed graphs ....................................................................................... 71
  6.3.2 Flux graph ........................................................................................... 73
  6.3.3 Current graphs ..................................................................................... 74
  6.3.4 Execution times comparison ................................................................. 77

7 Conclusions ................................................................................................ 81
7.1 Future developments ................................................................................. 82

A dSPACE environment ............................................................................... 83
A.1 The dSPACE ControlDesk .................................................................... 83
A.2 Working environment description ............................................................. 84
A.3 Layout example ....................................................................................... 86

B Matlab codes ................................................................................................ 88
B.1 Files for simulation model ...................................................................... 89
B.2 Files for laboratory model ...................................................................... 94
List of figures

1.1 System block diagram .............................................. 2
1.2 Single-phase inverter topology [12] ............................ 7
1.2 Switch gate command generation ................................. 8
1.3 Output spectrum for $m_a = 0.8$ and $m_f = 9$ [12] ....... 9
1.4 Standard three-phase inverter topology [12] .......... 10
1.5 State vectors representation and the rotating $V_e$ vector [12] .... 12
1.6 Short circuit condition ............................................. 13
2.1 Motor overview .................................................... 15
2.2 Induction motor .................................................... 16
2.3 Permanent Magnet Synchronous Machine ..................... 17
2.4 Demagnetization curve [13] ..................................... 19
2.5 Demagnetization curves for different PM materials [13] .... 21
3.6 Different rotor configurations for PM synchronous motors: (a) classical configuration, (b) interior-magnet rotor, (c) surface-magnet rotor, (d) inset-magnet rotor, (e) rotor with buried magnet symmetrically distributed, (f) rotor with buried magnet asymmetrically distributed. [13] ................................. 25
3.7 Components of torque [14] ....................................... 28
3.8 Mesh of the IPM motor [15] ...................................... 29
4.1 Dynamic DC equivalent circuit .................................. 31
4.2 DC motor block diagram [16] .................................. 32
4.3 Block diagram ..................................................... 32
4.4 Control loops ..................................................... 33
4.5 Speed control loop ................................................ 35
4.6 Position control loop ............................................. 35
4.7 Three reference frames [20] .................................... 41
5.1 Simulink complete model ........................................ 44
5.2 PMSM and Mechanics internal structure ...................... 45
5.3 Mechanic equation ............................................... 46
5.4 PMSM internal structure ........................................ 47
5.5 Control block details ............................................. 48
List of Symbols

\( B \)  magnetic flux density
\( B_r \)  remanent magnetic flux density
\( B_{sat} \)  saturation magnetic flux density
\( d \)  duty cycle
\( E_f \)  induced voltage
\( f \)  frequency
\( f_c \)  frequency of modulating signal
\( f_{\Delta} \)  frequency of carrier signal
\( H \)  magnetic field intensity
\( H_c \)  coercive field strength
\( i_d \)  d component of current
\( i_o \)  output current
\( i_q \)  q component of current
\( J \)  moment of inertia
\( k_i \)  integral gain
\( k_p \)  proportional gain
\( k_\phi \)  flux factor
\( K_{w1} \)  stator winding coefficient
\( L_d \)  d component of inductance
\( L_q \)  q component of inductance
\( m_a \)  modulation index
\( m_f \)  frequency modulation ratio
\( N_1 \)  number of stator turns per phase
\( n_s \)  measured speed
\( R_s \)  armature winding resistance
\( T_e \)  developed torque
\( T_L \)  load torque
\( T_s \)  sampling period
\( v \)  velocity
\( v_c \)  modulating signal
\( v_d \)  d component of voltage
$v_{\Delta}$ carrier signal
$V_o$ output voltage
$v_q$ q component of voltage
$V_{dc}$ DC bus voltage
$x_f$ final position
$x_i$ initial position
$\mu_r$ magnetic permeability of ferromagnetic material
$\omega_m$ mechanical speed
$\omega_r$ electrical speed
$\phi_f$ magnetic excitation flux
$\psi_d$ d component of flux linkage
$\psi_f$ flux linkage constant
$\psi_q$ q component of flux linkage

List of Acronyms

CPU Central Processing Unit
DEIM Discrete empirical interpolation method
DTC Direct Torque Control
EMF Electromotive force
FEM Finite Element Method
FOC Field Oriented Control
FOH First-order hold
HIL Hardware-In-the-loop
IPM Internal Permanent Magnets
MOR Model Order Reduction
OIM Orthogonal interpolation method
PMSM Permanent Magnet Synchronous Motors
POD Proper orthogonal decomposition
RCP Rapid Control Prototyping
RISC Reduced Instruction Set Computer
RMS Root mean square
RTI Real-time interface
SCU Signal conditioning unit
SVD Singular value decomposition
ZOH Zero-order hold
Chapter 1

Introduction

In the last decades there has been an always increasing demand for reducing the dependence on fossil fuel. This change already involves many sectors among which: home heating that is gradually switching to renewable energy, industry applications and transportations. This last sector in particular is very crucial due to the fact that it produces roughly more than one third of the overall pollution. As a consequence of its importance, the biggest change that is happening in the transportation sector is the increasing number of electric and hybrid vehicles that circulate on the streets. Considering all these aspects is easy to forecast that electric motors will play an even more important role in the future, therefore it is worth to focus research activities in order to improve their behaviour and to exploit them in the best way.

The majority of electronic systems that include electromechanical actuators, have to interface with the mains, which have different advantages as stability, reliability but also some drawbacks, the most critical is the fact that all the parameters are fixed. In case of a motor there is the necessity to vary the speed but connecting directly the motor to the mains this is not possible due to the fixed voltage provided. In this context, a key role is played by the electronic systems devoted to convert the electrical energy in the fixed form that is available in the specific form that is required by the motors. Such power electronic systems are made of reactive elements and transistors that are operated as switches controlled by low power signals generated by specific control circuits. In the specific case of motor field, the controller collects the data from different sensors like currents, speed, torque, position and generates the proper output. With reference to AC motors the schematic block diagram in Fig.1.1 depicts the overall system needed to drive the motor. The control part is where the physical quantities measured on the motor are compared with the reference values and its output is the duty cycle information sent to the power electronic block [1].
Some of these quantities can be easily and directly measured, like current or voltage, some others can require specific equipment, as the case of speed measure that require an encoder on the shaft of motor. Another possibility is the estimation of some quantities, this option usually requires some models. The model allows to predict the value of one quantity using the available measurements of other variables. The control section is the most flexible and also the hardest to develop, because every choice made here can considerably influence the overall system performance. The control must be designed in agreement with the required output performance, which change in relation to the application.

Nowadays, more and more systems are replacing analog control with digital control. The fundamental reason of this change is the increase in flexibility when an analog control is replaced with the digital one. The digital systems are not limited to few basic functions, due to the fact that they are programmable they can perform very complex calculations. Another important advantage is the possibility to modify and improve a digital system also after its implementation, just changing the software memorized in it. This possibility is completely absent in the analog counterpart, where everything is defined in the production step, because fabricated with specific interconnections of physical devices and components.

The problem with digital control is the high level of inefficiency in the computational effort, linked to the fact they are very demanding considering the amount of data to be processed and stored. To solve this critical drawback many techniques have been developed to increase the computational efficiency of this kind of algorithm, which are extremely flexible but not as efficient as desirable. Different digital control strategies try to reduce the number of variables stored, the complexity of the relationships that connect outputs with inputs and other parameters, which change case by case. The most well-known techniques in motor control field are for example, the Field Oriented Control (FOC) [2] that allows to optimize the size and the control of the motor to increase its efficiency. Another technique is the Direct Torque Control (DTC) [3], which is easier and faster than the FOC [4], because does not require any speed sensor. However, this simplified structure also involves
some problems, when the speed is very low: the error starts to increase and makes it impossible to drive motors that reach the zero value for the output frequency.

The efficiency of the computation in the control algorithm is very important because directly affects the efficiency of the whole system. The overall efficiency take into account the component related to the control strategy and the component related to the motor. Furthermore, it is necessary to keep the efficiency as high as possible, like always in power electronic [1] field. In this way less energy is needed at the input, it means a lower cost, and less heat produced, hence it will be also easier cool down the system.

In this optic of increase the overall efficiency, Permanent Magnet Synchronous Motors, also called PMSM[5], are becoming more appealing and they are capturing more markets[6]. This is because with PMSM is possible to reach higher efficiency compared with an induction motor[7] of the same power level.

In this context of digital control constantly evolving to achieve the best compromise between flexibility and efficiency is inserted this thesis work. The thesis, carried out in the laboratories of the Aalto University (Helsinki, Finland), aims to develop a control system for the IPM (Internal Permanent Magnets) motor, which will be used in the different laboratory tests. The goal is to develop a new control strategy for the motor starting from the review of the literature in this field [8], [9], [10] and to integrate in the control a completely new way of estimate the parameters of the motor. This new strategy exploits Model Order Reduction (MOR) techniques, developed in the laboratories of the Aalto University, in the department of Electrical Engineering and Automation, coordinated by the professor Anouar Belahcen. My contribution to this thesis is to integrate into a real-time control system this innovative technique which has so far been tested only by means of theoretical simulations. The actual use of this new model will be analyzed and the performance will be evaluated in a real application and not just by simulations. The aims is also to give concrete feedback regarding an intensive industrial use. In fact, for the first time, thanks to this thesis, will be obtained data on the accuracy of the estimates provided by this new method and especially on the computational time needed to obtain them.

To achieve the goals mentioned above, after the first step of literature review, it will be implemented the model for the controller and the motor in Matlab-Simulink environment. The control is implemented using two independent functional blocks employed to estimate the main parameters. One of these blocks in particular is implemented through MOR. The choice to apply this new technique goes into the direction of increasing the efficiency and it guarantees also an increase in the speed of the control and reduces the number of variables that have to be stored. In this way it is possible to reduce the useless movements of rotor caused by delays on the control path. As a result, losses are reduced and output energy is maximized. The
model of the motor is very important in the simulation step of this thesis, because it should represent the real behaviour of the motor as good as possible. In fact, all the main decisions on the control path will take place during the simulation step. Therefore, if the motor model used in simulations is not very close to the real one, when the solution will be applied to the real motor the output will diverge from the expected one. As a result is very important to adapt the simulation model to the system that have to be analyzed. It will follows a simulation session to investigate if the results obtained by the Matlab-Simulink model are in agreement with the theoretically expected values. The simulations involve the main parameters of the motor, the speed, the current absorbed, the flux produced. When the measured and the estimated values will converge, it will starts the next step. In this phase, the model developed will be adapted to work properly in the dSPACE environment, used to control the motor in real-time. The last step will be the tests in the laboratory to measure all the fundamental variables directly on the motor and compare these measures with the results obtained in the simulation step.

It follows a briefly description of each single chapter present in this thesis to have a quickly overview.

Chapter 1: introduction shows the general context of power electronic sector and digital control techniques in which the thesis is collocate. It is also explained the aims of the thesis and the innovative results achieved.

Chapter 2: The inverter covers one of the main theatrical topics strictly connected to the aim of the thesis. It starts with the explanation of the inverter theory and then enumerates the different possible configurations. Finally, is explained the concept of dead time extremely useful to avoid short circuits.

Chapter 3: Permanent Magnets motor introduces the electric motors with more emphasis on the Permanent Magnet motor due to the fact that IPM motor, the one, which the thesis aims to control, is part of this family. Its structure and working mode will be analyzed in detail.

Chapter 4: Control begins by explaining the main idea of control in the loop and the different configurations, that are more frequently applied. It continues by giving a theoretical explanation of the innovative model adopted in the control system the Model Order Reduction. Finally, is explained the Park-Clark transformation extremely useful to reduce the computational effort.

Chapter 5: Proposed solution describes the specific solution realized to control the motor. In the first part, it analyse in details the Matlab-Simulink model,
explaining the function of each block presents in it. The second part shows the results of simulations.

**Chapter 6: Experimental results** shows the model adapted to work properly with dSPACE environment. First of all, it explains the main blocks that realize the interface between the desktop control and the real world. Afterwords, analyzes the results obtained from the laboratory tests.

**Chapter 7: Conclusions** summarizes the path followed during the thesis and gives an overview of the possible future developments.
Chapter 2

The inverter

Inverters are electronic switching converters\cite{11}, that transform the DC current at input into AC current at the output. Inverters generate a sinusoidal output voltage with the possibility to control frequency and amplitude. They are widely used in electromechanical conversion of energy, like AC motor driving. Their spread and importance is growing due to the increase of renewable energy produced. In this case they are involved in the conversion of the DC power produced into AC power sent to the grid. Inverters are constructed from power switches that can be controlled in order to produce a modulated voltage at the output. Values of output voltage are therefore discrete and a fraction of the DC voltage source. For the same reason the output waveform is not a sinusoidal voltage. The output of an inverter is usually a three-level waveform. Adding a filter at the output of the inverter makes it possible to have a sinusoidal output voltage, related to the fact that only the fundamental component remains after the filter action. It is possible to change amplitude and frequency of the output signal controlling switches commutations.

2.1 Single-phase inverter

The simplest topology for an inverter is the single phase half-bridge. To generate the required voltage for the neutral point two capacitors are connected in series, and the DC bus voltage is applied to this series. The resulting neutral voltage is equal to $\frac{V_d}{2}$, this neutral point is required for applying a negative voltage to the load. Two switches are used to connect the load alternatively to the positive of the DC input and to ground. Diodes are added in parallel to avoid over-voltages, because switches can usually conduct current in only one direction. Diodes ensure that the current has an alternative path if it cannot flow through the switch.

This configuration has three possible states, one is an undefined state instead the other two are defined. In the defined states one switch is ON and the other one
OFF, the output voltage $v_o$ can be:

$$\begin{cases} 
+\frac{V_{dc}}{2} & \text{S+ On; S- Off} \\
-\frac{V_{dc}}{2} & \text{S+ Off; S- On}
\end{cases} \quad (2.1)$$

When both switches are off the output voltage is determined by the current sign. If the current is positive as a result the diode $D_+$ will turn on, and $D_-$ will turn on if the current is negative. The relation between output voltage and current sign is reported below:

$$\begin{cases} 
+\frac{V_{dc}}{2} & \text{if } i_o > 0 \\
-\frac{V_{dc}}{2} & \text{if } i_o < 0
\end{cases} \quad (2.2)$$

### 2.1.1 PWM Inverter control

To get the desired output switches must to be controlled with a specific pattern. One of the most common techniques to drive switches is the Pulse width modulation (PWM). In this technique the command signal is a square wave and the main information is carried by the duty cycle, that is changed in order to obtain the proper voltage. The output waveform is generated comparing a modulating signal $v_c$ with a triangular waveform $v_{\Delta}$ (carrier signal), while switches are turned on according with the rule reported in Tab. 2.1 and gate signal is shown in Fig.2.2:
The average output value is equal to:

\[
\overline{V_o} = V_{dc} \cdot d - \frac{V_{dc}}{2}
\]  \hspace{1cm} (2.3)

In which \(d\) is the duty cycle of the PWM waveform that controls the switches.

Considering a sinusoidal signal with frequency \(f_c\) and amplitude \(\hat{v}_c\) then the spectrum of the obtained output voltage is similar to Fig. 2.3. It contains the fundamental at the desired frequency plus other harmonics that can be removed by filtering action.

Amplitude of the fundamental is determined by the modulation index \(m_a\) defined as

\[
m_a = \frac{\hat{v}_c}{\hat{v}_\Delta}
\]  \hspace{1cm} (2.4)
with $v_{\Delta}$ that is the amplitude of the carrier. The amplitude of the fundamental $v_{o1}$ is then

$$v_{o1} = v_{aN1} = \frac{V_{dc}}{2} m_a$$

(2.5)

The quality of the output voltage is directly related to the quantity and amplitude of unwanted harmonics it contains. This is correlated to the frequency modulation ratio $m_f$ that is

$$m_f = \frac{f_c}{f_{\Delta}}$$

(2.6)

Furthermore, it is worth pointing out that harmonics appears at specific frequencies $f_h$ with

$$h = lm_f \pm k \quad l = 1,2,3,...$$

(2.7)

where $k = 2,4,6...$ for $l = 1,3,5...$ and $k = 1,3,5...$ for $l = 2,4,6$ [12]; this means that harmonics are centered around $m_f$ and its multiples. The choice of higher $m_f$ moves harmonics at higher frequencies as result the output filtering action will be easier, obtaining also a better output voltage waveform.

### 2.2 Three-phase inverter

Usually in high power applications three-phase loads are preferred. So it is necessary to extend the working mode that regulates the single-phase inverter to a three-phase one. The standard topology of a three-phase inverter is realized with three legs, where each one of the legs can be viewed as a basic single-phase inverter. The three-phase inverter can be controlled, like the single-phase one, with the PWM
technique. There are three modulating signals instead of one, and each signal is phase-delayed by 120° compared to the next one. Moreover, even if it is possible to use this technique, for three-phase inverter frequently it is used an other technique called space-vector modulation that leads to a better control result.

Figure 2.4. Standard three-phase inverter topology [12]

2.2.1 Space-vector modulation

The switches that composes the power module always have two different states: On and Off. Combining all the switches states it is possible to obtain different configurations of the entire system depicted in Fig. 2.4. Not all configurations can be used some of them are forbidden to avoid damage or malfunctioning of the inverter. One of the most critical situation is when two switches belonging to the same leg are On simultaneously causing a short circuit from the positive rail of the DC bus to the negative one, therefore damage the system. To uniquely mark the remaining valid states it is assigned to each leg of the inverter a binary state variable that changes its value according to the leg configuration. The principle used to assign the value of state variable is summarized in the Tab. 2.2: The state

<table>
<thead>
<tr>
<th>State</th>
<th>Switches</th>
<th>$V_o$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S_+$ On and $S_-$ Off</td>
<td>$V_{dc}$</td>
</tr>
<tr>
<td>0</td>
<td>$S_+$ Off and $S_-$ On</td>
<td>0 V</td>
</tr>
</tbody>
</table>

Table 2.2. Rule for assignment of state variable

variables of each phase are assembled together to realize a variable of three bit, one bit for each phase. The number of all possible configurations for the three-phase
The Tab. 2.3 enumerates all the possible configurations for the inverter and the relative output voltage:

<table>
<thead>
<tr>
<th>State</th>
<th>$v_{cb}$</th>
<th>$v_{bc}$</th>
<th>$v_{ca}$</th>
<th>Space vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>$V_{dc}$</td>
<td>0</td>
<td>$-V_{dc}$</td>
<td>$V_1 = 1 + j0.577$</td>
</tr>
<tr>
<td>110</td>
<td>0</td>
<td>$V_{dc}$</td>
<td>$-V_{dc}$</td>
<td>$V_2 = j1.155$</td>
</tr>
<tr>
<td>010</td>
<td>$-V_{dc}$</td>
<td>$V_{dc}$</td>
<td>0</td>
<td>$V_3 = -1 + j0.577$</td>
</tr>
<tr>
<td>011</td>
<td>$-V_{dc}$</td>
<td>0</td>
<td>$V_{dc}$</td>
<td>$V_4 = -1 - j0.577$</td>
</tr>
<tr>
<td>001</td>
<td>0</td>
<td>$-V_{dc}$</td>
<td>$V_{dc}$</td>
<td>$V_5 = -j1.155$</td>
</tr>
<tr>
<td>101</td>
<td>$V_{dc}$</td>
<td>$-V_{dc}$</td>
<td>0</td>
<td>$V_6 = 1 - j0.577$</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$V_7 = 0$</td>
</tr>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$V_8 = 0$</td>
</tr>
</tbody>
</table>

Table 2.3. Inverter configurations in space-vector modulation

The space-vector modulation realizes a transformation of a three-dimensional set of values into a two-dimensional one. This is strictly related to the fact that if the sum of three variables in the abc space is always zero they can be replaced with a vector in the αβ reference frame, thanks to the use of Park transformation [12]:

$$v_{ca} = \frac{2}{3}[v_{ca} - 0.5(v_{cb} + v_{cc})]$$  (2.8)

$$v_{cb} = \frac{\sqrt{3}}{3}(v_{cb} - v_{cc})$$  (2.9)

Considering the case of three variables $V_{c(abc)}$ with identical sinusoidal shape at frequency $f$, the resulting vector $V_c$ in the αβ reference frame will show a constant module $\hat{v}_c$ and will rotate at the same frequency of the three variables. The αβ frame can be divided in 6 identical sectors as displayed in Fig. 2.5.

Every sector is delimited along the edges with vectors that can be generated by the inverter using one of its configurations. Vector $V_c$ representing the modulating signals can lay in any position inside the αβ space. In case the vector is placed between two edges, which delimits one of the six sectors, it cannot be directly represented by the inverter. The basic idea of the space-vector modulation is to approximate the rotating vector using the two adjacent valid states. Analyzing the case in which $V_c$ is laying between the vectors $V_i$ and $V_{i+1}$ only the nearest non-zero states and a zero state should be used to approximate $V_c$ [12]. This technique is used to reduce the switching operations and at the same time to maximize the amplitude of the wanted output. An important requirement to ensure is that in one sampling
period $T_s$ the average output voltage is equal to the wanted vector $V_c$. Another constraint is that the total results from sum of the time intervals of each vector, used to generate $V_c$, must be exactly one sampling period $T_s$. Each time interval is:

$$T_i = T_s \cdot \hat{v}_c \cdot sin\left(\frac{\pi}{3} - \theta\right)$$ \hfill (2.10)

$$T_{i+1} = T_s \cdot \hat{v}_c \cdot sin(\theta)$$ \hfill (2.11)

$$T_s = T_i - T_i - T_{i+1}$$ \hfill (2.12)

The Park transformation is frequently used together with Clarke transformation to obtain the control variables of the system. With the use of Clarke transformation, control variables from $\alpha\beta$ reference frame are reported in a rotating frame.
dq, which rotates at exactly the same speed of the motor. With the use of these two transformations together control variables show constant module, this property considerably simplify the use of these variables.

### 2.2.2 Deadtime

The fact that electronic switches are not ideal gives some problems, the most important is that the variation from the ON state to the OFF state is not instantaneous, due to some parasitic effects it is necessary some time to complete the switching operation. The situation becomes even more critical for the power devices, in which this time can be quite long. Looking at a concrete case, for example one leg of the inverter, if the On→Off commutation of the high side switch and the Off→On of the low side begin at the same instant, as reported in Fig.2.6, the result will be a short circuit between the two different potentials.

![Diagram of Deadtime](image)

**Figure 2.6.** Short circuit condition

This situation is very critical for the whole system and is avoided by using the *deadtime* technique. Which to prevent any overlap in the ON time and consequently any short circuit, changes the command signal introducing a dead time between the turn OFF of one device and the turn ON of the other one.

The length of dead time is decided considering the turn-on and turn-off time shown by the power switches, as a result the signal sent to the gate is modified accordingly to Fig.2.6.
Chapter 3

Permanent Magnets Motor

The chapter topic is the analysis of the motor used in this thesis. First of all, introducing the electric motors and then analyzing in detail the Permanent Magnet motor, its structure and its working mode. The IPM motor used is part of the big family of PMSM and the main characteristics are common to all the motor kinds belonging to this group.

3.1 Electric motors

Electric motors [11] are used in a wide range of applications. This is possible because nowadays they are extremely reliable and also available in any kind of shape, dimension and power range. Considering industry applications, it means something that does not need to be moved, electric motors have been the standard since very long time ago. Moreover, today electric motors are becoming more important also in the transportation sector, which until recently was dominate by thermal motors. This is related to the fact that electric motors are always more efficient and batteries are getting better. As result, performance are increasing, this leads to longer range on a single charge for vehicles that employ this kind of motor. To be more precise there are different types of motors, as possible to see from Fig.3.1, but only some of them can be used in traction field. The first one solution is the induction motor and the second one is the Permanent Magnet Synchronous Machine.
3.1.1 Induction motor

The induction motor [7] always rotates at a speed that is slightly slower with respect of the synchronous speed. The main reason for this small difference is related to the fact that, the synchronous speed is the speed of the magnetic field generated by the stator windings. The latter makes flux into rotor, which in turn starts to rotate. Anyway, due to lagging in the flux current generation in the rotor with respect to flux current in the stator, these two speeds will always show a small difference.

The working principle of induction motor is based on the induced magnetic field. When the stator winding is supplied, the current, which flowing in it generates a magnetic flux. The rotor side is realized in such a way that is a short circuit in this condition, then current can start flowing in it and generates a rotor flux. Thanks to this interaction between the two fluxes, the rotor feel a torque, and it starts to move in the same direction of the magnetic flux. There are different kinds of induction motor, the main division is between single phase motor and three-phase motor. An important difference is that single phase motor is not self-starting instead three phase motor is self-starting. This leads to have more complicated circuit to make single phase motor running. Usually this problem is solved by dividing the stator winding in two windings and putting a capacitor in one of the two windings. In this way are generated different phases for the currents and therefore it is created a torque on the rotor.
3.1.2 Permanent Magnet motors

The PMSM is an electrical machine, which can be easily driven by a microcontroller to get the best performances. This kind of motor is usually implemented in medium-high power applications due to slightly high cost. The most common sectors of application are electric cars and industrial machine. In addition, the history of PMSM started in the first half of the 19th century, thanks a series of inventions inspired by Faraday’s laws of electromagnetic induction. Different inventors tried to use this solution but the result was not the best due to the very poor quality hard magnetic material available at that time. Successively, the introduction of the rare earth materials start to change the scenario and nowadays the permanent materials have better performance and lower price than ever. On the whole, from outside this kind of motor look like exactly similar to an induction motor, and is partially true, because the stator configuration is almost equal, same shape, same way to generate the currents. However the rotor is totally different instead of a squirrel cage, is realized by permanent magnets that can have different shapes, based on what needs to be optimized.
3.2 Analysis of Permanent Magnet motors

3.2.1 PM overview

The use of permanent magnets in construction of electrical machine leads many benefits, the main are listed below:

- No electrical energy is required by the field excitation system, this means also no excitation losses
- Higher power density and higher torque density
- Higher dynamic performance than motors with electromagnetic excitation
- Simplification of construction and maintenance

The use of PM brushless motors is becoming more popular than induction motors. First of all, this popularity is due to the fact that rare earth PMs increase
different aspects as, the motor's steady state performance, the power density and
the dynamic performance. In addition, the improvement achieved in the field of
semiconductors drives made the control of brushless motors easier and cost effective,
giving the possibility to run the motor over a very wide range of speeds ensuring
good efficiency. A PM brushless motor has the magnets mounted on the rotor and
the armature winding that realize the stator. This means, the rotor does not re-
quire current transmitted through slip rings and brushes. These are the major parts,
which require maintenance. Another advantage of PM brushless motors is
that the power losses occur almost only in the stator, where heat can be easily re-
moved with the use of ribbed frame or in case of large machine with a water cooling
system. The permanent magnet motors can be driven in different way mainly de-
pending on which kind of drive is used. All the electromechanical drives are divided
into constant-speed drives, servo drives and variable-speed drives. In the case of a
constant-speed driver it is not need an electronic converter. The synchronous motor
is directly connected to the mains, this configuration it can be a good choice only
when is not required to follow the speed variations with small tolerance. In the
other hand, servo motor drive implies several devices to continuously monitor the
main variables, like speed or position and compare them with some reference value
to get the right outcome. In this solution the precision and accuracy are the main
requirements, for this reason such system can use also multiple feedback loops. Least
but not last, in the variable-speed drive the most important aspect is the possibility
to vary the speed over a wide range, instead the accuracy of the response becomes
less important. Brushless motor drives are divided into two big classes, sinusoidal
excited and square wave. In case of sinusoidally excited motor they are fed with
three-phase sinusoidal waveforms and operate on the principle of rotating magnetic
field.

3.2.2 Demagnetization curve

The main feature of permanent magnet (PM), is that it produce an induced magnetic
field with no dissipation of electric power. The energy from mains is required only
in changing the intensity of the magnetic field, not in maintaining it. PM can be
described by its B—H hysteresis loop, as usual for ferromagnetic materials. PMs are
also called hard magnetic materials, meaning that they have wide hysteresis loop.
In the evaluation of the hysteresis loop an area of particular interest is the upper
left-hand quadrant, also called the demagnetization curve, which is represented in
Fig.3.4. When a reverse magnetic field intensity is applied, to an already magnetized
system, the magnetic flux density drops down to the value determined by the point
K. In this condition, if the reversal magnetic flux is removed the flux density returns
to the point L according to a minor hysteresis loop. As a result, the use of a reverse
field has reduced the remanence, or remanent magnetism. The minor hysteresis
loop can be approximated, with a small error, by a straight line called the recoil line [13]. Another important aspect to underline, if the negative value of applied magnetic field is less than the maximum value corresponding to the K point, the PM can be considered almost permanent. The relationship between the magnetic flux density $B$, that represent the intrinsic magnetization $B_i$ due to the presence of the ferromagnetic material, and magnetic field intensity $H$ may be expressed as reported in [13]:

$$B = \mu_0 H + B_i = \mu_0 (H + M) = \mu_0 (1 + \chi) H = \mu_0 \mu_r H$$  \hspace{1cm} (3.1)$$

In which $M = B_i/\mu_0$, $B$, $H$, $B_i$ are parallel or anti-parallel vectors. Moreover, the relative magnetic permeability of ferromagnetic material is $\mu_r = 1 + \chi >> 1$. The magnetization vector $M$ is directly related to the magnetic susceptibility $\chi$ of the material. The flux density $B_i$ is the contribution given by the ferromagnetic core.
The main parameters that characterize the PMs are enumerated below:

- **Saturation magnetic flux density** $B_{sat}$. This parameter with the corresponding magnetic saturation field $H_{sat}$, define the point in which all the magnetic moment domains are aligned with the external magnetic field.

- **Remanent magnetic flux density** $B_r$, also called remanence. It is the magnetic flux density corresponding to zero magnetic field intensity. There is a direct connection between higher remanence and higher supported magnetic flux density.

- **Coercive field strength** $H_c$. This is the value of demagnetizing field intensity required to bring the magnetic flux density to zero in a material that was previously magnetized.

- **Maximum magnetic energy**. This is the Maximum magnetic energy per unit produced by a PM in the external space.

$$\omega_{max} = \frac{(BH)_{max}}{2} \quad (3.2)$$

In which $(BH)_{max}$ represent the maximum energy density point on the demagnetization curve, Fig.3.4.

### 3.2.3 Permanent magnet materials

There are different Permanent magnet materials used in the actual motor technology, the main ones can be divided into three classes:

- **Alnico** (Al, Ni, Co, Fe)

- **Ceramics**, also called ferrites

- **Rare-earth materials** (SmCo, NdFeb)

To compare these different materials it can be used demagnetization curve, reported in Fig.3.5. It shows the behaviour of the different materials at a fixed temperature of 20°C and for a wide range of magnetic flux density B, reported in the y-axis, and of magnetic field density H along the x-axis.
**Alnico**

The first material that was developed is Alnico. The main advantages of this material are its high magnetic remanent flux density and its low temperature coefficient, it means that can be used also at quite high temperature, like 520° C, without evident changes in its properties. In the other hand, the coercive force is very low and the demagnetization curve is extremely non-linear. Moreover, Alnico is easy to magnetize but as a drawback is also easy to demagnetize. Alnico was the most used material in PM motors in the low-medium power range applications up to the late 1960s, when ferrites start to conquer the PM motors market.

**Ferrites (Ceramics)**

The main ferrite compounds imply the use of Barium and Strontium. Ferrites show an higher coercive force than Alnico, but as counterweight have a lower remanent flux density.

![Demagnetization curves for different PM materials](image)
magnetic flux density. The main benefits of ferrites are the low cost and high electric resistance, that leads to no eddy current losses. These materials are used mostly in the low power range and are realized by powder metallurgy.

**Rare-earth materials**

A great change occurs in the permanent magnet materials market when the rare-earth PMs reached a good level of development. Rare-earth materials are not rare at all, but to produce one of this particular rare-earth compounds it is necessary to combine several elements that singularly do not have any practical application. This is the real limit of such materials. The first compound realized with this methodology was the SmCo5, which has the advantages of high remanent flux density, high coercive force, high energy product, low temperature coefficient and linear demagnetization curve. The only drawback of this material is the cost, because both Sm and Co are quite expensive. Subsequently, a new rare-earth material was realized thanks to the discovery of the inexpensive Neodymium. The NdFeB magnets realized using, as a main component Neodymium, have better magnetic properties with respect to SmCo magnets. The issue is the temperature coefficient, which has an higher impact on the performance of this material reducing its advantages with the increase of temperature. This leads to a low limit for the operating temperature, that is around 250°C. The latest improvements on the NdFeB alloys are reducing the influence of temperature variations achieving a greater thermal stability.

### 3.2.4 Permanent Magnet Synchronous motor

The most important features of the Synchronous motors is that they operate at a constant speed in absolute synchronism with the line frequency. The subdivision between different kinds of Synchronous motors is mostly based on their rotor design and shows four main categories:

- Electromagnetically-exited motors
- PM motors
- Reluctance motors
- Hysteresis motors

In all these motors listed above there are specific values of different parameters to take into account, but in general for the synchronous motor there is a limit to the maximum (pull-out) torque, which can be developed, before the rotor is forced out of synchronism. Typically, this value is around 1.5 times the continuous rated torque, however it can reach also 4 to 6 times in some specific applications. Anyway the two most used configurations for synchronous motors are discussed below.
Electromagnetically-exited motors

In this kind of motor the rotor needs to be fed to excite its self. To achieve this result two solutions are possible. The first solution is to use a pair of slip rings on the shaft. The second one is to use an external, also called auxiliary, brushless exciter on the same shaft of the main motor. The power range for this kind of motors is from few KW up to hundreds of MW.

Permanent magnets motors

In this case it is necessary to supply only the stator, the rotor field is generated by the permanent magnets attached to it. The power range for these motors is from 100 W up to ten MW. The main drawback is that the excitation is inherently fixed by the PMs, in the other hand the advantage is also a robust and reliable rotor construction.

Despite the fact that there are many different kinds of motors, they have some fundamental relationships, which are valid for everyone. The main ones are listed below.

Speed

The speed measured at the rotor is given by the ratio of input frequency over number of pole pairs, considering the steady-state range.

\[ n_s = \frac{f}{p} \quad (3.3) \]

The rotor speed is exactly the same compared to the speed of the rotating magnetic field generated by the stator.

Air gap magnetic flux density

To simplify it is possible to consider only the first harmonic of the air gap magnetic flux:

\[ B_{mg1} = \frac{2}{\pi} \int_{-0.5\alpha_i\pi}^{0.5\alpha_i\pi} B_{mg} \cos(\alpha) d\alpha = \frac{4}{\pi} B_{mg} \sin \frac{\alpha_i\pi}{2} \quad (3.4) \]

In the formula above it is neglected the saturation of magnetic circuit. The value of \( \alpha_i \) is defined as ratio of the average-to–maximum value of the air gap magnetic flux density normal component.

\[ \alpha_i = \frac{B_{av}}{B_{mg}} \quad (3.5) \]
If in the air gap the field distribution is sinusoidal $\alpha_i = \frac{2}{\pi}$

**Voltage induced**

The RMS (Root mean square) induced voltage in one phase of stator winding is evaluated in no-load condition. This induced voltage, also called EMF, is generated by magnetic excitation flux $\phi_f$ of the rotor and is expressed through the eq. (3.6).

$$E_f = \pi \sqrt{2} f N_1 K_{m1} \phi_f$$ (3.6)

In which $N_1$ is the number of stator turns per phase and $K_{m1}$ represent the stator winding coefficient.

### 3.2.5 Different kinds of PM motors

The first example of PM motor was implemented by F.W. Merril, it was a four-pole motor. The leakage flux produced by PM motors can be changed acting on the width of the narrow slot[13]. The PM is attached to the shaft of motor with the use of aluminium or some other alloy. The different configurations that are enumerated below are also reported in Fig.3.6 to have a graphical view of the different possible configurations in the rotor structure. In Fig.3.6 is reported only the rotor view because the stator is exactly the same for all the different kinds of motors, then it does not carry any additional information.

**Interior-magnet motor**

Considering the different configuration for the rotor design, the first for importance is the interior-magnet rotor(Fig. 3.6b), which has radially magnetized and alternately poled magnets. The synchronous reactance in d-axis is smaller than that in q-axis since the q-axis magnetic flux can pass through the steel pole pieces without crossing the PMs. One of the best advantages of this configuration is the fact that the magnets are very well protected against centrifugal forces. Then it is a good choice for high frequency and high speed motors.

**Surface-magnet motor**

Another option is the surface mounted magnets(Fig. 3.6c), in this case the magnets are magnetized radially or also circumferentially. Usually it is used an external non-ferromagnetic high conductivity cylinder, to protect the magnets against the demagnetization and from the centrifugal forces. If the rare-earth PMs are used synchronous reactance in both axes, q and d, are practically the same.
Figure 3.6. Different rotor configurations for PM synchronous motors: (a) classical configuration, (b) interior-magnet rotor, (c) surface-magnet rotor, (d) inset-magnet rotor, (e) rotor with buried magnet symmetrically distributed, (f) rotor with buried magnet asymmetrically distributed. [13]

Inset-magnet motor

The inset-type motors (Fig. 3.6d) is another way of realize the rotor, in this solution PMs are magnetized radially and embedded in shallow slots. The rotor magnetic
circuit can be laminated or made of solid steel. The synchronous reactance of the q-axis is bigger than in the d-axis. In this solution, the EMF induced by PMs is lower than the case of surface PM motors.

**Buried PM motors**

The last alternative in the rotor design is the buried PM motors (Fig. 3.6e), which has circumferentially magnetized PMs embedded in deep slots. The synchronous reactance in q-axis is greater than that in d-axis. In this kind of rotor, an asynchronous starting torque is produced with the aid of both a cage winding, incorporated in slots in the rotor pole shoes, or a solid salient pole shoes made of mild steel. The use of a non-ferromagnetic shaft is fundamental, otherwise a large portion of magnetic flux goes through the shaft and it is wasted.

### 3.2.6 Dynamic model of PM motors

Usually, the strategies of control for the sinusoidally excited synchronous motor use the d-q linear model of electrical machine. The d-q dynamic model is represented in a rotating reference frame that moves at the same synchronous speed \( \omega \) of the magnetic field. With this technique the time varying parameters are eliminated and all the variables in the model are referred to orthogonal or mutual decoupled d-q axes. The following set of general equations is used to describe a synchronous machine:

\[
\begin{align*}
v_q &= R_s i_q + \omega_r \psi_d + p \psi_q \\
v_d &= R_s i_d - \omega_r \psi_q + p \psi_d
\end{align*}
\]  
(3.7)  
(3.8)

The flux linkage is defined as:

\[
\psi_q = L_q i_q
\]

(3.9)

\[
\psi_d = L_d i_d + \psi_f
\]

(3.10)

Substituting the eq.(3.9) and eq.(3.10) in the equations (3.7) (3.8) the result is reported below:

\[
\begin{align*}
v_q &= R_s i_q + \omega_r (L_d i_d + \psi_f) + p L_q i_q \\
v_d &= R_s i_d - \omega_r L_q i_q + p (L_d i_d + \psi_f)
\end{align*}
\]  
(3.11)  
(3.12)
In which \( v_d \) and \( v_q \) are the d and q axes components of the terminal voltage, \( R_s \) is the armature winding resistance, \( i_d \) and \( i_q \) are the d and q axes components of the terminal current and \( \psi_f \) is the maximum value of the flux linkage per phase produced by excitation system.

Moreover, in the equations it is used the rotor electrical speed, \( \omega_r \) that is related to the mechanical speed through the well know formula:

\[
\omega_m = \omega_r \left[ \frac{2}{p} \right]
\]  

(3.13)

The resultant armature inductance are:

\[
L_d = L_{ad} + L_1 \quad L_q = L_{aq} + L_1
\]  

(3.14)

Where \( L_1 \) is the leakage inductance of the armature winding per phase and \( L_{ad} \) and \( L_{aq} \) are the self-inductances in the d and q axes.

The maximum flux excitation linkage is defined as \( \psi_f = L_{fd}I_f \) in which \( L_{fd} \) is the maximum of the mutual inductance between field winding and armature and \( I_f \) is the fictitious current.

The developed torque of a three phase motor is:

\[
T_e = \frac{3}{2}p(\psi_di_q - \psi_qi_d)
\]  

(3.15)

Where \( p \) is the number of pole pairs, substituting the eq.(3.9) and eq.(3.10) in the eq.(3.15), the new expression for the developed torque is:

\[
T_e = \frac{3}{2}p(\psi_f + (L_d - L_q)i_d)i_q
\]  

(3.16)

### 3.3 IPM motor used

The control realized in this thesis is optimized to work with an IPM motor. Certainly, the developed model can be adapted to other kinds of permanent magnets motor just with some small changes in the setting parameters. Anyway, the specific kind of motor used is explained in more details.

As also shown in the previous section the IPM motor is a synchronous motor with permanent magnets embedded in its rotor. Applying a three-phase AC voltage at the stator windings it produces a rotating magnetic field on the rotor. In one frequency cycle, the rotating magnetic field passes from one pole to another one [14]. If the frequency of three-phase voltage is increased, also the speed of rotating
magnetic field is increased. Moreover, the important advantage of IPM motor is the increased torque produced for a specific value of current. This positive aspect is linked to the way in which the torque is obtained that implies two generation mechanisms. In fact, the generated torque in the IPM motor is the sum of magnet torque and reluctance torque.

The magnet torque is directly related to the magnetic flux generated from the current flowing in the stator windings. The attractive or repulsive force between the magnetic flux and the magnets in the rotor produces the magnet torque component. The reluctance component is instead determined by the saliency ratio of the IPM motor.

The vector control of an IPM motor implies the division of current in two components the $I_q$ component used to generate the desired torque and the $I_d$ component used to generate and control the magnetic flux. When the latter is properly used to generate a reluctance torque component, the total torque can be represented as in Fig.3.7.

As is evident from the Fig.3.7 maximum torque delivered is increased with an efficient use of the reluctance component.

To evaluate the flux intensity it is necessary to carefully schematize the internal
structure of the motor. The internal structure of the IPM motor implied in the thesis work is shown in Fig. 3.8.

In the Fig. 3.8 is evident the structure of the motor, the purple area represent the permanent magnets, instead the stator coils are marked with red, green and blue colours for the three different phases. It is reported only one sector due to the symmetry with the others. Another important aspect that can be point out is the structure of the different triangles that realize the mesh used to obtain the finite element model (FEM).
Chapter 4

Control

The chapter deals with control techniques. First of all, introducing the main idea of control and particularly the closed loop control. Then it will be explained the innovative model used to obtain the estimates of the parameters, the Model Order Reduction. Finally, is explained the Park-Clark transformation extremely useful to reduce the computational effort.

4.1 Control of the system

To realize the control it is necessary to pass through a simulation step that makes possible to check if the idea is correct before use it directly. This is due the fact that change the realization in the real system is very difficult, it takes long time and if the solution is wrong it can give problems. Instead, simulations allows to debugging the solution before applying it to the real system, in this way it is possible to save time and money. To have meaningful simulations it is necessary to develop a model of the system under analysis, that is at the same time simple, in this way it does not require long time for simulation, and also accurate, to simulate the real behaviour of the system. Following the idea of getting a simple model for the system, and applying it to the motor field, usually means use a model derived from the DC motor that is the easiest one. With the use of some transformations is possible to exploit the idea behind the DC equivalent model also for other model as AC motors. For this reason, it is useful to know this basic DC model [16] showed in Fig.4.1.

The armature circuit differential equation is [16]:

\[ u_a = R_a i_a + L_a \frac{di_a}{dt} + e_a \]  \hspace{1cm} (4.1)

In this equation, the first term is related to voltage drop across the resistance \( r_a \), the second term takes into account the voltage drop across the inductance \( L_a \) and
the last term represents the induced voltage due to electromagnetic force (back-emf) from the motor, which can be written as $e_a = k_\phi \Omega_m$.
Moreover $k_\phi$ is the flux factor $k_\phi = k_a \phi$ and $\Omega_m$ is the mechanical angular speed of the rotor.
The equation of motion is:

$$T_e = J \frac{d\Omega_m}{dt} + T_L$$ (4.2)

In which $T_L$ is the load torque, $J$ is the moment of inertia, the developed torque is:

$$T_e = k_\phi i_a$$ (4.3)

Considering the flux constant and using the Laplace transformation of the armature equation

$$u_a(s) = R_a i_a(s) + L_a i_a(s) + k_\phi \Omega_m(s)$$ (4.4)

The evaluation of $i_a(s)$ gives:

$$i_a(s) = \frac{1}{1 + \frac{sL_a}{R_a}} u_a(s) - k_\phi \Omega_m(s)i_a(s)$$ (4.5)

Applying Laplace transformation also to the eq.(4.3), the result is:

$$T_e(s) = J \frac{d\Omega_m(s)}{dt} + T_L$$ (4.6)

The resulting $\Omega_m(s)$ is:

$$\Omega_m(s) = \frac{T_e(s) - T_L(s)}{sJ}$$ (4.7)

The last equation needed:

$$T_e(s) = k_\phi i_a(s)$$ (4.8)
Considering all the previous equations it is possible to draw the block diagram as reported in Fig. 4.2. This can be modified to fit exactly the standard form for a block diagram in control engineering, presented in Fig. 4.3.

The last step is to extrapolate the relation that connects this two representations, and the relation between inputs and outputs.

\[
\frac{Y(s)}{R(s)} = \frac{G_1(s)G_2(s)}{1 + G_1(s)G_2(s)H(s)} \quad (4.9)
\]

\[
\frac{Y(s)}{D(s)} = \frac{G_2(s)}{1 + G_1(s)G_2(s)H(s)} \quad (4.10)
\]

In this specific case the values are:

\[
G_1(s) = \frac{\frac{1}{R_a}}{1 + \frac{sL_a}{R_a}} k_\phi \quad (4.11)
\]
\[ G_2(s) = \frac{1}{sJ} \]  
\[ H(s) = k_\phi \]  

The output and input have to be set as follows:

\[ Y(s) = \Omega_m(s) \quad R(s) = U_a(s) \quad D(s) = T_I(s) \]  

4.1.1 Controller design

In the systems that require high precision in the speed curve usually are used two different loops. The inner one that control the torque, and the second loop that control the speed. Furthermore, if it is also necessary to control in a very precise way the position, commonly it is inserted an outermost loop that control the position, and in this case there are three loops. The innermost have to be the fastest, and the outermost have to be the slowest.

The motion control systems have to manage large changes in the imposed, or also reference, values of the torque, speed and position. They also reject large unexpected load changes, called disturbances.

![Control loops](image)

**Figure 4.4. Control loops**

**Torque Control**

The first loop to design is the torque one. For the torque control usually it is used PI controller.

The transfer function of PI is:

\[ \frac{u_c(s)}{e(s)} = k_p + \frac{k_i}{s} = k_i + k_\phi \left( 1 + \frac{k_p}{k_i} \right) \]  

(4.15)
In which $e(s)$ is the error signal that is used as input for the controller, obtained as the difference between the reference signal and the measured signal. Moreover $u_c(s)$ is the output signal of the controller.

To simplify the control the solution is to measure the current instead of measure directly the torque. This is possible because the current is related to torque by the constant $k_\phi$. The simplification leads to the fact that measure a current is easier and more precise than measure directly a torque. To adapt the PI controller to the specific system is necessary to select the gain constants. The zero position is chosen in order to cancel the pole of the system that is at $-\frac{1}{T_a}$. The result is:

$$\frac{k_p}{k_i} = T_a$$  \hspace{1cm} (4.16)

Considering the open loop transfer function of the simplified current control loop:

$$G_{I,OL}(s) = \frac{k_{ii} s}{s} \left( 1 + \frac{s k_p c}{k_i I} \right) k_T \frac{1}{R_a} \frac{1}{1 + s T_a}$$  \hspace{1cm} (4.17)

With the choice of eq.4.16

$$G_{I,OL}(s) = k_{ii} k_T \frac{1}{R_a s}$$  \hspace{1cm} (4.18)

With the requirement $|G_{I,OL}(J \omega_c)| = 1$ it is possible to obtain the angular crossover frequency:

$$\omega_c = \frac{k_{ii} k_T}{R_a}$$  \hspace{1cm} (4.19)

At this point to define all the parameters it is necessary to choose the value of $k_{ii}$. This choice is made in order to get a crossover frequency for the open loop approximately one or two orders of magnitude smaller than the switching frequency of the converter, to avoid undesired interference.

**Speed loop**

During the design of the speed loop it is possible to consider the current closed loop as ideal and represented by a unity block. Indeed, if the control work properly it must be as close as possible to this ideal situation.

The speed controller will be a PI type, the transfer function is:

$$G_{\Omega,OL}(s) = \frac{k_{ii} \Omega}{s} \left( 1 + \frac{s k_p \Omega}{k_i \Omega} \right) k_T \frac{k_\phi}{s J}$$  \hspace{1cm} (4.20)

This function have a double pole in the origin, the magnitude declines of $-40dB$ per decade at low frequencies. The right choice is to have $\omega_c \Omega$ the crossover frequency of the speed loop one order of magnitude smaller than $\omega_c I$ (the crossover frequency of current loop) with also a good value for the phase margin $\phi_{pm,\Omega}$. 

34
Design of position control loop

The position loop it is designed to have only the proportional gain \( k_\theta \), because there is an integration term \( \frac{1}{s} \) already in the open-loop transfer function.

The open loop transfer function is:

\[
G_{\theta,OL}(s) = \frac{k_\theta}{s}
\]  \hspace{1cm} (4.21)

To evaluate the gain \( k_\theta \) it is necessary to choose the value for \( \omega_{c\theta} \), crossover frequency for the position loop. Imposing \( |G_{\theta,OL}(J\omega_{c\theta})| = 1 \) results:

\[
\frac{k_\theta}{\omega_{c\theta}} = 1 \quad k_\theta = \omega_{c\theta}
\]  \hspace{1cm} (4.22)

The strategies that can be used to control the motor are very different one from each other. They can also be divided in different categories, for example it is possible to exploit the division between scalar and vector control [17], or the division between control that uses sensors and sensorless control.

### 4.1.2 Scalar control

The idea on which is based the scalar control is to vary simultaneously two parameters. Increasing or decreasing the supply frequency will change the speed of the
motor and this will also change the impedances. This change in the impedance can leads to some problems, for this reason usually also the frequency is changed in such a way that compensates the variation related to the voltage. The scalar control can be implemented in open loop or closed loop mode. The open loop solution is less expensive and gives a simpler system, but cannot control the torque, that instead can be done with closed loop control together with a more precise speed control.

4.1.3 Vector control

In case of vector control, the most used technique is the Field Oriented Control. This is a mathematical abstraction, which leads to an easier model for the motor in some way similar to the DC machine. The idea behind this technique is related to phasors, also named as rotating vectors represented in a complex coordinate system. This control strategy allows to decouple the field components. The result is to have two independent currents, the torque producing current and the flux producing current, in this way flux and torque are controlled independently.

4.2 Model Order Reduction (MOR)

The first step to optimize the design and the control of an electric device or machine is to analyse the electromagnetic field produced in a deep and precise way. The precise evaluation of the field solution of the machine is quite complicated due to multiple factors as, the complexity of the geometry, the spatial dependency and the non-linear effect due to the material used. To manage this rising complexity the use of numerical method such us finite element method can be very useful. However, if the model under analysis has an higher order, the time and the storage capacity required starts to become a serious problem. These limitations affect directly the accuracy of the modelled solution, which have to be sample enough for the use in real-time control system, which shows very strict limitation in time and data storage capability. Model order reduction (MOR) [15], is a good choice to reduce the complexity of the model and however ensure an acceptable level of accuracy. This method is also applied in a wide range of engineering field. To perform a MOR, a set of precomputed solutions, called snapshot matrix [18], is decomposed into a subset of orthogonal basis with the well-known method of proper orthogonal decomposition (POD). At the end of the process, the system of equation is reduced into a series of orthogonal basis functions. The application of MOR is not complicated if the system under analysis is linear. Contrariwise, the reduced problem of a non-linear system still depends on the non-reduced system, to solve this issue, the POD method is usually combined with the discrete empirical interpolation method, DEIM. With this solution the non-linear part is approximated by projecting it into a subspace,
as a result the reduction efficiency of a non-linear system is enhanced. The method interpolates the non-linear solutions with the right-singular vectors of the snapshot matrix. These vectors are generated by using the singular value decomposition (SVD) while decomposing the snapshot matrix into the subset of orthogonal bases. Therefore, this method does not require any additional projection method and the interpolation functions can be used without changes for any unknown solution. In this way accuracy and computation time are improved.

### 4.2.1 Singular value decomposition (SVD)

The singular value decomposition is a very useful transformation used in wide range of applications. Considering a singular value decomposition of a matrix $A$, this is the factorization of $A$ into the product of three matrices:

$$ A = U \Sigma V^T $$

(4.23)

In which the columns of $U$ and $V$ are orthogonal and the $D$ matrix is diagonal with positive real entries. As mentioned earlier the SVD have many fields of application. The first of those can be when it is necessary to reduce the rank of matrix, finding a low rank matrix which is a good approximation of the original matrix. The singular value decomposition is defined for all kinds of matrices, rectangular or square does not make difference, anyway the more used spectral decomposition is linear algebra.

Another mathematical formulation can be obtained considering $A$ as an $n \times d$ with singular vectors $v_1,v_2,\ldots,v_r$ and corresponding singular values $\sigma_1,\sigma_2,\ldots,\sigma_r$. Keeping in mind $u_i = \frac{1}{\sigma_i}Av_i$ for $i = 1,2,3,\ldots,r$ are the left singular vector. As a result $A$ can be decomposed into a sum of rank one matrices[15]:

$$ A = \sum_{i=1}^{r} \sigma_i u_i v_i^T $$

(4.24)

### 4.2.2 Using the Orthogonal interpolation method

The interpolation-based method is used to reduce the computational time and complexity for solving the system of equations[15]. The method benefits from the orthogonal properties in the computation of the interpolation technique. Instead of developing the interpolation on the full snapshot matrix [19], the interpolation is realized on a smaller orthogonal basis from the SVD. The SVD can also be interpreted by three distinct coordinate transformations. The right-singular vectors in $V^T$ corresponds to rotating the initial coordinate system. It is followed by a stretching transformation with the norm matrix $\Sigma$ and a final rotation $U$. Considering that
the snapshot matrix represent correctly the system the discrete projection operator is determined with \( U \) and its most energetic norm are tracked with the norm matrix \( \Sigma \). These two matrices remain exactly the same for any input variables within the range of the snapshot matrix. This property is fundamental to apply MOR, the prediction of the system output is only dependent on the right-singular vectors in \( V \) [15].

\[
A_m = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1m} \\
a_{21} & a_{22} & \ldots & a_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \ldots & a_{nm}
\end{bmatrix} \quad \rightarrow \quad V^T = \begin{bmatrix}
v_{11} & v_{12} & \ldots & v_{1m} \\
v_{21} & v_{22} & \ldots & v_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
v_{m1} & v_{m2} & \ldots & v_{nm}
\end{bmatrix}
\tag{4.25}
\]

In this context of orthogonal basis, any new input set can be expressed as a vector sum of orthogonal vectors. Each component of this new vector can be independently interpolated with the corresponding components of the right-singular vectors as a function of the original input quantities that can be current, rotor angle or other variables.

\[
V_{new} = \begin{bmatrix}
v_1 \\
v_2 \\
\vdots \\
v_m
\end{bmatrix} = \begin{bmatrix}
f_1(x_1, x_2, \ldots x_m) \\
f_2(x_1, x_2, \ldots x_m) \\
\vdots \\
f_m(x_1, x_2, \ldots x_m)
\end{bmatrix}
\tag{4.26}
\]

The magnetic vector potential for this new input \( V_{new} \) can be estimated with [15]:

\[
A_{new} = U \Sigma V_{new}^T
\tag{4.27}
\]

The orthogonal interpolation method reduces in a significant way the computational effort for both memory allocation and mathematical operations, mainly because \( m \ll n \). In addition to this, since each orthogonal basis vector is independent from the others, interpolating on this orthogonal system does not required any eventual cross-coupling term as in the interpolation between the nodal vector potentials. This is a great advantage of the method because instead in the finite element method the solution of one node depends on its adjacent nodes. Usually, the interpolation method try to neglect this effect, but this simplification can leads to an high inaccuracy. This is especially true when the system under analysis is non-linear and the cross-coupling term could be important.
4.3 Park-Clark transformation

The general behaviour of three-phase machines is described by their voltage and current equations. These equations are differential equations, in which the coefficients are time varying, except when the rotor is stalled. The necessity to take into account these variations leads to a very complex mathematical model. The complexity is mainly due to flux linkages, induced voltages and currents that change continuously as the electric circuit is in relative motion. Accordingly with this analysis, to reduce the complexity of the electrical machine system mathematical transformation are often used. The goal of these transformations is to decouple variables and to solve equations, that involve time varying quantities, by referring all variables to a common frame of reference. The most used transformation is the Park-Clark, mainly used in vector control strategies related to PMSM and asynchronous machines. Park-Clark transformation to be more precise is the combination of two separated transformations, that are:

- Park transformation
- Clark transformation

4.3.1 Clark transformation

This transformation converts balanced three-phase quantities into balanced two-phase quadrature quantities. The Clark transformation is expressed through the equations below:

\[
I_\alpha = \frac{2}{3} I_a - \frac{1}{3} (I_b - I_c) \quad (4.28)
\]

\[
I_\beta = \frac{2}{\sqrt{3}} (I_b - I_c) \quad (4.29)
\]

In which \( I_a, I_b \) and \( I_c \) are the three phase quantities, instead \( I_\alpha \) and \( I_\beta \) are the stationary orthogonal reference frame quantities. Moreover it is also possible to write this transformation in matrix form:

\[
\begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \times \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \quad (4.30)
\]
**Inverse Clark transformation**

The reverse transformation, that from a two-axis orthogonal stationary reference frame gives a three-phase stationary reference frame, it is obviously called inverse Clark transformation. Which is defined by the following equations:

\[
V_a = V_\alpha \\
V_b = \frac{-V_\alpha + \sqrt{3}V_\beta}{2} \\
V_c = \frac{-V_\alpha - \sqrt{3}V_\beta}{2}
\]  

(4.31) \quad (4.32) \quad (4.33)

In which \(V_a\), \(V_b\) and \(V_c\) are the three phase quantities, \(V_\alpha\) and \(V_\beta\) are the stationary orthogonal reference frame quantities.

**4.3.2 Park transformation**

Through this transformation it is possible to convert vectors in balanced two-phase orthogonal stationary system into orthogonal rotating reference frame.

\[
I_d = I_\alpha \cos(\theta) + I_\beta \sin(\theta) \\
I_q = I_\beta \cos(\theta) - I_\alpha \sin(\theta)
\]  

(4.34) \quad (4.35)

In which \(\theta\) is the rotation angle, \(I_d\) and \(I_q\) are the rotating reference frame quantities, \(I_\alpha\) and \(I_\beta\) are the stationary orthogonal reference frame quantities. Furthermore, it is possible to represent the transformation in matrix form:

\[
\begin{bmatrix}
I_d \\
I_q
\end{bmatrix} = 
\begin{bmatrix}
\cos(\theta) & \sin(\theta) \\
-\sin(\theta) & \cos(\theta)
\end{bmatrix} 
\times 
\begin{bmatrix}
I_\alpha \\
I_\beta
\end{bmatrix}
\]

(4.36)

**Inverse Park transformation**

In a dual manner, the variables in rotating reference frame are transformed into two-axis orthogonal stationary reference frame applying the inverse Park transformation, represented by the following equations:

\[
V_\alpha = V_d \cos(\theta) - V_q \sin(\theta) \quad (4.37)
\]

\[
V_\beta = V_q \cos(\theta) + V_d \sin(\theta) \quad (4.38)
\]
In which \(V_d\) and \(V_q\) are the rotating reference frame quantities, \(V_\alpha\) and \(V_\beta\) are the orthogonal stationary reference frame quantities.

Finally, to have a better idea of this transformation is useful to visualize it through the Fig. 4.7, in which are represented the three reference frames used in the transformation.

The three reference frames used are:

- The three-phase reference frame, in which \(I_a\), \(I_b\) and \(I_c\) are co-planar three-phase quantities with a displacement angle of 120° to each other.

- The orthogonal stationary reference frame, in which the two components \(I_\alpha\) and \(I_\beta\) are perpendicular to each other, but in the same plane as the three-phase reference frame.

- The orthogonal rotating reference frame, in which \(I_d\) is at an angle \(\theta\), the rotation angle, to the \(\alpha\) axis and \(I_q\) is perpendicular to \(I_d\) along the \(q\) axis.

To close the explanation of the Park-Clark transformation, it should be stated that this is one of the most useful transformation in the field of electric machine. Such transformation linearises the systems and allows to use all the theorems well-known for the linear systems, that simplify the evaluation of the variables involved in the system.
Chapter 5

Proposed solution

5.1 Introduction to RTI systems

In the engineering field is impossible to develop a control system without simulations. The first step of general procedure for the development of a new controller is to realize a model of the real system and simulate it on a computer. Only after it is added the controller of process and it is optimized to achieve the specifications.

The first step of development is carried out with the use of Simulink, the main advantage of this solution is related to the fact that the software is very fast in evaluate the result for the system. Of course if higher precision is required the computational time increases, therefore is necessary to choose the right balance between precision and computational effort. When the model complies with the specifications the controller is added to the model and the whole system is tested again. Before the controller is implemented at physical level another step of simulation is required. In this case the real system is controlled by a simulated real-time controller implemented through a prototype board, in this thesis work it will be the DS1103. This step is very important because allows to exactly fit the controller to the real system avoiding future problems. This technique it is called Rapid Control Prototyping (RCP).

The last step is to connect the real controller to the real-time simulated model of the plant in order to check if the controller generates some errors that can damage the physical system. This technique is called Hardware-In-the-loop Simulation (HIL).

The real-time simulation is a fundamental aspect, as the automatic generation of the real-time code that will successively implemented into hardware.

In the dSPACE systems the real-time interface (RTI/RTI-MP) give a direct connection between these functions when it is joined with the real-time Workshop of Matlab. Thanks to this cooperation it is possible to generate real-time code to use in dSPACE directly from the Simulink model and let it load straight in the
5.2 Simulink model

As explained above the first step of this thesis work is to develop a Simulink model of the Interior Permanent Magnet (IPM) motor and to build a controller, a vector control in the specific case, to send the right inputs to it.

At the beginning was necessary an important phase of literature review on this topic [8] - [10]. After this phase it was chosen to base the system on a current controller with discrete-time pole-placement [21]. Generally, in this kind of control the stator current is represented in rotor coordinates [22], [23], [24]. This choice comes from the simplification that can be achieved using this coordinates system, which utilize only dc steady state quantities. The most frequent approach in this condition, to realize the control, is the use of Proportional Integral (PI) controller with also implies some decoupling terms to compensate the rotating reference system [25]. Performance improvements are achieved with the use of closed loop control. The plant can be synthesized as a stator admittance from current controller prospective. Therefore, the poles of the closed loop system can be placed in the desired position to obtain the expected output. Anyway, knowing the d-q components can leads to a more accurate estimate. The implementation of a discrete time control can pass through the realization of a continuous-time one. The latter can be discretized with the use of Euler or Tustin approximation [25]. An equivalent discrete model, which include the Zero-order hold (ZOH) and the sampler, is necessary to complete the design of the discrete-time control. Needs to be taken into account also a computational delay of one sampling period, because always present in the real system. A first-order low pass filter can be used in the continuous-time domain to represent the delay, in this way the controller automatically compensates it [26]. While in the discrete-time domain the delay can be represented in an extremely precise way with other techniques [27]. In literature are also present some experiments of current controller for IMP motor [24], [28], [29]. Everyone of them have some negative aspect to be fixed. After obtaining a closed-form hold-equivalent discrete model for the IPM motor [21]. It can be applied to design and implement controllers and observers [30].

Starting from the analysis of the motor model all the main blocks of the Simulink model, reported in Fig.5.1 will be analyzed. The main blocks include the current controller, the Zero-order Hold (ZOH) and the motor. Moreover, looking at the Fig.5.1 it is evident the use of a feedback control. The outputs from the motor, to be more specific the angle and the three phases currents, are measured and sent to the control block. They are used to generate the right input for the motor, passing through the ZOH that is necessary to transform the discrete variables into
continuous signals applied to the real motor.

![Simulink complete model](image)

Figure 5.1. Simulink complete model

### 5.2.1 Motor model

In this section the block marked with the name PMSM and Mechanics will be analyzed in more details to better understand the working mode of the specific block. As the name suggests it contains the equations that describe the behaviour of the equivalent real motor which has to be controlled.

This block has two inputs, as also possible to see in Fig. 5.2. The first input is the voltage in alpha-beta coordinates (uss) generated into the ZOH, the second one is the load torque (TL), this choice allows to have more flexibility in the model, it can simulate different loads just changing this parameter. Of course in the real case this is not a parameter but is a constraint. Therefore, the outputs are the current in alpha-beta coordinates (iss) and the rotor angle (thetam). The current in alpha-beta coordinates is a different way of represent the current in the three phases of the motor, the equivalence between this two forms is obtained with the Clark transformation.

Going one step below, the block shows its internal structure, represented in Fig. 5.2. Analysing in details these blocks, it is clear the structure, composed by two main blocks the PMSM and the Mechanics. The second one has two inputs the
load torque, decided by the user in this case, and the electromagnetic torque \((T_e)\) generated by the motor. The output is the rotor speed \((w_m)\) that is also used as input of the \(PMSM\) block. The formula that is codified inside this block is written below:

\[
W_m = \frac{p}{J_{tot}} \int T_e - TL \ dl
\]

(5.1)

In which \(p\) is the number of pole pairs of the machine and \(J_{tot}\) is the value of total inertia showed. The implementation of this relation through Simulink is reported in the Fig. 5.3, that shows the use of only simple blocks, like gain and integral, listed in the basic components of Simulink libraries.

The other main block is the \(PMSM\), which has two inputs, the voltage in the alpha-beta coordinates reference, generated by the ZOH and the rotor speed evaluated in the \(Mechanics\) block. As output it is possible to enumerate the \(\alpha-\beta\) current, the angle and the electromagnetic torque. To understand how these outputs are calculated is useful go down in the layers and analyse the core of the \(PMSM\) block.
Figure 5.3. Mechanic equation

presented in Fig.5.4. From the figure are evident the different parts: voltage equation, flux equation and electromagnetic torque. The voltage equation contains the codification of the voltage equations discussed in the Chapter 3 of this thesis, where is explained the dynamic model of the permanent magnets motors, and reported also here to simplify the consultation.

\[ v_q = R_s i_q + \omega_r (L_d i_d + \psi_f) + p L_q i_q \]  \hspace{1cm} (5.2)

\[ v_d = R_s i_d - \omega_r L_q i_q + p (L_d i_d + \psi_f) \]  \hspace{1cm} (5.3)

The first step is transform the \( \alpha - \beta \) voltage that arrives as input into the d-q voltage used in the eq.5.2) and eq.5.3. In the Fig.5.4 is directly visible the d-q voltage (us) because the transformation is in the upper level. The other inputs of this block are the rotor speed and the d-q current (is) generated from the flux equation part. Which with the use of the flux information as input generate the correspondent value of d-q current.

Finally, the last part is the Electromagnetic torque block, that has as inputs the flux (psis) and the current (is) both in d-q coordinates and gives as result the electromagnetic torque developed from the motor. The block implements the eq.(5.4) written below:

\[ T_e = \frac{3}{2} p(\psi_d i_q - \psi_q i_d) \]  \hspace{1cm} (5.4)

From eq.(5.4) is evident the necessity of have both the input variables in the d-q reference frame in order to simplify the torque evaluation. However, there are different formulas that can give the electromagnetic torque developed, in this specific
case it was used the eq.(5.4) because all the inputs are already available without transformations or complex equations.

5.2.2 Control model

The control block is marked, in Fig.5.2, with the name Vector Control related to the fact that in this work is implemented a vector control for the motor under analysis. The inputs are, the speed reference (wmref), the current in $\alpha - \beta$ coordinates (iss), the DC voltage linkage bus (Udc) and the measured rotor angle (thetam_meas). The speed reference can be varied by the user also in real-time, the upper limit for this variable is the nominal speed, which for the motor used in the thesis is 1500 rpm. This limit follows the choice of keep the speed under the nominal value since there is no need to go in the weakening region. The output is the duty cycle value (dabc) for the three phases, ’a’, ’b’, ’c’, of the motor. This value is in discrete form, as all
the variables in the control block, from this follows the need of an unit delay and of the ZOH before send it directly to the inverter. Referring to the Fig.5.5, which report the internal structure of the Vector Control block, it can be collected more information about the specific control strategy used.

Figure 5.5. Control block details

From Fig.5.5 it is evident the usage of closed loop feedback with more than one loop. There are two loops one inside the other, as it is well known from the control theory this is realized to increase the accuracy of the control. In this specific case the outermost loop is the speed loop, which involve the Speed controller and Reference Calculation blocks. The inner loop is the current loop, that involve the Current Controller. The inner loop must be faster with respect of the outer one to let the system work properly without errors.

The first block from the left is the Speed controller, which receives three inputs: the reference speed, the measured speed (wm) and the limit torque (Tref_lim). The reference speed is defined by the user as stated before. The measured speed in this case is given by the current controller instead when it will be connected to the real motor this signal will be obtained from the encoder. The limit for the torque is changed dynamically step by step, to better adapt it to motor necessities. Another solution can by a fixed maximum value for the torque decide by the user at the beginning and kept constant for the whole simulation. Inside this block is used a Matlab function to evaluate the output value of the torque reference (Tref). The choice to use a Matlab function goes in the direction of simplify the model, otherwise a complex set of blocks would be needed to realize the same function, that can be written in a few code lines into Matlab. The equation used into the code to evaluate the torque is reported also in the eq.(5.5).
\[ T_{\text{ref}} = k_t \ w_{\text{mref}} + T_{\text{ei}} - k_1 \ w_m - k_2 \ T_{\text{ref, lim}} \]  \hspace{1cm} (5.5)

In which \(k_t\), \(k_1\) and \(k_2\) are constant values defined by the user, dependent on the
parameters of the motor. These values are chosen to define the poles position of the
transfer function. Instead \(T_{\text{ei}}\) is the integral state, whom take also into account the
anti wind-up term. This value is update every cycle as it is possible to understand
from the use of the delay in the loop showed in Fig.5.6.

![Figure 5.6. Speed block internal structure](image)

Analysing the other part named \textit{Reference calculation}, also in this case it is used
a Matlab function. Using the reference torque evaluated in the previous block and
the minimum current value in the d-axis (\(i_{d,\text{min}}\)), the d and q component of the
current are calculated with the eq.(5.6):

\[
\begin{align*}
\text{idref} &= i_{d,\text{min}} \\
\text{iqref} &= \frac{T_{\text{ref}}}{1.5p[(L_d - L_q)\text{idref} + \text{psipm}]} \hspace{1cm} (5.6)
\end{align*}
\]

The current in the d-axis is the magnetization current, which must be different
from zero into an induction motor otherwise the rotor it will not be magnetized.
However, for the motor used in this thesis, that is an IPM motor, the current can
be set to zero without problems, because the magnetization for the rotor is given
by the permanent magnets. The d-q components of the current are directly related
to the flux reference (psisref) output used in the next block and are also used to evaluate the limit torque (Tref_lim) used in the Speed controller.

The current controller is the more complex block to analyze, because receives many different inputs: the DC-linkage voltage, the flux reference, the $\alpha - \beta$ current (iss) and the measured rotor angle. The block also generates the most important outputs. The duty cycle is the fundamental output to control properly the motor, an error in this value dramatically influence the behaviour of the motor. The flux estimation is also very important for the purpose of this thesis to evaluate and compare the different control strategies that it will be implemented to control the motor. The flux is evaluated in two reference frame the d-q reference (psise) and in the three phases system (flux_abc) to have a more complete comparison. Due to the complexity of this block a Matlab function is used. In this function are evaluated the matrices, the placement of the poles of the transfer function is defined, the observer is updated, the gains are evaluated, and the duty cycles for the three phases are defined separately using the technique of the Symmetrical Suboscillation PWM. This specific method is adopted because it allows to use the whole available voltage, which is not possible with the use of only the Suboscillation Method, that reach only 87% of the maximum available voltage. The improvement is achieved with a proper use of the zero-sequence component. The complete Matlab code used into the function is reported in the Appendix B.

5.2.3 Zero-order hold (ZOH)

The use of this block is necessary in any kind of digital control system, as the one used in this work. This is related to the fact that in each digital control system, is necessary to convert the sampled signal, obtained as output of the regulator, into a continuous signal in order to apply it to the physical system under control. The aim is to generate a continuous signal $u_h(t)$ which must be sent to the plant, starting from a discrete $u_k$ signal provided by the controller. Moreover, as evident the realization of the signal $u_h(t)$ from the sequence $u_k$ must be random, because in each instant the signal will not know the future values of the samples generated by the controller. The most employed solution is to keep constant for the whole sampling interval the value of the last sample generated. The device that realize the function described above it is called Zero-order hold (ZOH) and its behaviour is reported in Fig.5.7 This device generate an analogue signal, which in the interval $kT < t < (k+1)T$ has a constant value equal to the value assumed by the input at the instant $t = kT$.

The signal reconstructed can theoretically depends also from more that one sample of the discrete signal, instead of depending just from the last one acquired as in case of ZOH. For example the First-order hold (FOH) gives an output signal $u_h(t)$ in the interval $kT < t < (k+1)T$ that follows the extension of the straight line that
connects the points \((kT, u(kT))\) and \(((k - 1)T, u((k - 1)T))\).

Considering the specific implementation investigated in this work, two inputs are required: the duty cycle \((dabc)\) and the current in \(\alpha - \beta\) coordinates \((iss)\). As main result gives the voltage in \(\alpha - \beta\) coordinates, which after a transformation to get the voltage for the three different phases is directly applied to the motor. The complexity level of this block allows to implement it directly through basic Simulink blocks instead of use a Matlab function. The Simulink implementation is reported in the Fig.5.8.

From Fig.5.8 is evident the structure that imply just a multiplication between the duty cycle and the DC-linkage voltage to get a voltage, one sum, couple of transformations from \(\alpha - \beta\) coordinates to the three phases reference frame and vice versa.

5.2.4 Simulation results for the basic scheme

After the detailed analysis of the different blocks which constitute the basic Simulink model used to drive the IPM motor, are now reported the results of the simulations for the model discussed above. These results are interesting mostly because they give a direct feedback on how the simulated values follow or not the expected values,
that are evaluated with the use of theoretical equations. Moreover, for this specific thesis work, the relevant values to take under control are the speed, in order to check if the output speed is the one imposed, and the flux in the three phases, because later on it will be the value used to compare the MOR with the theoretical estimation of the flux.

However, starting from the analysis of the speed in Fig. 5.9 it is reported the graph that compare the actual speed with the expected one.

Where the value marked with the blue continuous line is the actual value generated by the Simulink model, instead the dotted line is the reference value for the speed evaluated by mathematical equations. Along the y-axis is reported the value of the speed \( \omega_m \) not in absolute value, but normalized to the maximum speed. In the Fig.5.9, for example, it is possible to see that the steady state value is 0.4. It means that the actual running speed for the rotor is the 40% of the nominal speed, which for the motor used is 1500 rpm, then the rotor is running at around 600 rpm. Along the x-axis is reported the time. In fact, still looking at the Fig.5.9, it is possible to see that the system starts at \( t = 0.5s \). Before the reference speed was zero while at exactly 0.5 seconds jumps at the final desired value of 600 rpm, or 40% of the maximum value.

The real speed of course can not follow this sudden change and acting as a first order system reaches the final value after a certain period of time, this period of adjustment depends directly on the system time constant. For the system analysed the time interval is around 0.5 s, because the finale value is reached for x-axis value equal to one second. Looking at the blue continuous line is evident another change with respect of the dotted line occurs at \( t = 1.5s \), the variation is due to a change in the load torque. This variation changes the system parameters and the controller needs a certain time interval to adapt itself to the new condition.
Moreover, another interesting variable is the flux, the Fig. 5.10 reports the two components of the flux in the d-q reference frame.

In Fig. 5.10 are graphed separately the d and q component of the flux. Along the y-axis are reported the values of the flux components not in absolute value but normalized to the maximum value. As reported in the legend the D-component is marked with a blue line and the Q-component is instead marked with red line. Moreover, are also reported the theoretical estimated values of both components marked with a dotted line, because the blue and red lines are for the values evaluated through the Simulink model. The dotted lines are very difficult to see because perfectly superimposed by the coloured lines for the evaluated values, this is positive because means that the two values evaluated and estimated are exactly the same. Analysing the Fig. 5.10 is also clear that the D-component is constant for the whole plotted period, this is related to the fact that the D-component is associated to the magnetization component and in a PM motor is not necessary to use it like into an induction motor. Moreover, in the Q-component are visible huge changes, the highest is at the starting point when the motor require a very high current to start.
moving. After this huge necessity of current, when the speed is kept constant the required current goes down, until the load torque changes and a new peak is reached, smaller than the first but still higher than the steady-state value. The final value depends on the load torque required as is visible in the Fig.5.10.

5.3 Simulink model with Model order reduction

The next step in the thesis work is to introduce the Model order reduction into the Simulink model. After the realization of the basic Simulink model, analyzed above, and the simulation of such model. Is now necessary to introduce the MOR, to obtain the estimation of the flux through this technique and compare it with the one obtained through the current controller in the basic model. The complete Simulink model is reported in Fig.5.11. The new part of the model is in the right part, the other blocks are almost the same with respect of the previous model except for some small changes.
Figure 5.11. Simulink model with MOR
The new part is also reported in the Fig. 5.13 to have a better vision of the blocks inserted to use the Model Order Reduction. The main block is the look-up table, in the specific case is a 3-D look-up table, means that there are three inputs: the magnitude of the current, the measured rotor angle and the ‘z’ variable.

The working mode of the look-up table is fundamental, through the V matrix are loaded all the interpolated functions for some specific values of current and angle defined when is evaluated the so called snapshot matrix in the initialization file reported in Appendix B. After the values of current and angle received as input are used to peak-up the specific value from all the interpolated functions with the aid of the data loaded before.

![Block Parameters: n-D Lookup Table](image)

**Figure 5.12.** Look-up table settings

The settings necessary to use correctly the look-up table are recognizable thanks to the Fig. 5.12.
Figure 5.13. Simulink model with MOR

The variable $V_{\text{new}}$ is the output of the 3-D look-up table. However, the needed output is the flux, to obtain it is necessary to pass through the vector potential, represented with $A_{\text{new}}$, using the eq.(5.7):

$$A_{\text{new}} = SVD(A_s) = U\Sigma V_{\text{new}} = U_{\text{sig}} V_{\text{new}}$$  (5.7)

Where the output of 3-D look-up table is multiplied by $U_{\text{sig}}$. The $U_{\text{sig}}$ is a vector obtained multiplying the right singular vector $U$ by $\Sigma$ the singular values of a diagonal matrix. The last step to get the flux is a multiplication of the vector potential obtained before with the coefficient $C$, only now is obtained the value of the flux for the three phases. The $C$ is a coefficient evaluated through the use of the formula below:

$$\psi = l(A_2 - A_1)$$  (5.8)

The eq.(5.8) take into account $l$ the length of the rotor in the machine and the value of two vector potentials $A_1$ and $A_2$. In the simple structure used for the analysis the flux $\psi$ is directly proportional to the vector potential through the coefficient $C$ as expressed in the formula:

$$\psi = CA_{\text{new}}$$  (5.9)

Other blocks are added to complete this model. There is a block called Magnitude that gives as output the RMS value of the input current. A gain block to transform the value of angle from radians to degrees and the last block that realize the product between matrices.
5.3.1 Simulation results

In this section the results of the simulations for the Simulink model updated with the Model Order Reduction are reported. The graphs for the speed, the current and so on are exactly the same, in fact in Fig. 5.14 are reported the measured speed and the expected one. Comparing Fig. 5.14 with Fig. 5.9 is evident the same behaviour is changed just the nominal speed value. The main difference is the flux estimation and the comparison of the one obtained with the Model order reduction with the one estimated in the current controller. This comparison is shown also in the Fig.5.15. Where along the y-axis is reported the absolute value of the flux, and along the x-axis is reported the time.

![Graph showing simulation results](image)

Figure 5.14. Speed graph with MOR

As is possible to see, in Fig.5.15 is represented a specific time interval in which the curves had reached the steady-state value, that in the specific case is a sinusoidal shape. During the transient interval the comparison would have been not useful for many different reasons which may affect the curves. In the Fig.5.15 the curves
marked with the continuous line are obtained from the flux estimator present in the current controller, instead the dotted lines represent the evaluation obtained from the Model Order Reduction. Analysing the curves in more details, it is possible to see that the two lines, the one evaluated with the MOR and the one evaluated through the Simulink model of the current controller are almost superimposed. This is a very good result because means that the two models gives the same outputs.

Figure 5.15. Flux comparison
Chapter 6

Experimental results

In this chapter is described in details the lab tests of the algorithm analyzed through simulations in the previous chapter. First of all is described the lab setup, with all the hardware used, starting from the dSPACE CPU up to the inverter. In the second part is described the dSPACE software developed, with particular emphasis on the control desktop interface realized to send commands to the motor in real-time. The last part deals with the analysis of the results obtained in the lab tests.

6.1 Measurement setup

The first step is to explain the hardware setup realized in lab to control the motor. The setup used is based in part on a PhD thesis [31] that took place in the same lab a few years ago, this is mainly due to the fact that realize the whole hardware setup could have required a very long time and is not the aim of this thesis work. To have a graphical view, in Fig.6.1 is reported the block scheme of the whole system. The main block is the dSPACE CPU, that control almost everything. It receives the inputs from the desktop control interface about the actual speed required, the functional mode and so on. The control unit receive also the inputs from the motor, to be more precise from the sensors that work around the motor. The inputs measured by the sensors are the actual speed kept under control by the encoder and the current in the three phases, measured before it goes through the windings of the motor. As output, generated by the controller implemented into the dSPACE CPU, it is possible to enumerate the duty cycle, more precisely the three separated duty cycles that control the three different phases of the motor. The inverter receive the information contained on the duty cycle to generate the proper voltage to apply at the three phases of the motor. Moreover, other two outputs are evaluated, the flux with the use of two different blocks, the Model Order Reduction and the flux estimator present in the current controller. However, these outputs are
evaluated in real-time and are sent to the desktop interface to collect these data and plot them in a graph.

![Block diagram](image)

Figure 6.1. Block structure

The permanent magnet motor used for the thesis is an experimental interior permanent magnet motor. The stator of the motor is taken from a commercial induction machine and the magnets are buried in the rotor. The windings of the stator, that are three phases windings, are delta connected. Moreover, all the parameters that describe the PMSM are reported in Tab. 6.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal torque $T_N$</td>
<td>14 Nm</td>
</tr>
<tr>
<td>Number of pole pairs $p$</td>
<td>3</td>
</tr>
<tr>
<td>Permanent magnet flux</td>
<td>0.545 Vs</td>
</tr>
<tr>
<td>Total moment of inertia</td>
<td>0.015 $Kgm^2$</td>
</tr>
<tr>
<td>Stator resistance $R_s$</td>
<td>3.59 Ω</td>
</tr>
<tr>
<td>Direct axis inductance $L_d$</td>
<td>0.036 H</td>
</tr>
<tr>
<td>Quadrature axis inductance $L_q$</td>
<td>0.051 H</td>
</tr>
</tbody>
</table>

Table 6.1. Interior permanent magnet motor parameters
The PMSM is fed in the laboratory tests, by a frequency converter, precisely a Danfoss VLT5004 converter, that has a modified electronic control. The algorithm that control the motor is implemented in the Matlab-Simulink environment. In the Simulink model are used different blocks that realize the interface with dSPACE DS1103 experimental processor board. The first step, to transfer the code in the processor board is to compile the Simulink model for the real-time execution only after the software obtained is uploaded in the board. Inside the DS1103 board work simultaneously two processors. The first one that has the role of master processor and which contains the whole developed software, the PowerPC 604e RISC processor. The second one is instead a Texas Instruments TMS320F240 DSP, which has
the role of slave processor, and for example takes care of the generation for the pulse-width modulation.

<table>
<thead>
<tr>
<th>PMSM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Power rating</td>
<td>2.2 kW</td>
</tr>
<tr>
<td>Nominal voltage</td>
<td>370 V</td>
</tr>
<tr>
<td>Nominal current</td>
<td>4.3 A</td>
</tr>
<tr>
<td>Frequency</td>
<td>75 Hz</td>
</tr>
<tr>
<td>Nominal speed</td>
<td>1500 rpm</td>
</tr>
<tr>
<td>$\cos \phi = 0.9$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current traducers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>LEM LA 55-P/SP1</td>
</tr>
<tr>
<td>Bandwidth $L_d$</td>
<td>from 0 to 200 kHz</td>
</tr>
<tr>
<td>Accuracy $L_q$</td>
<td>±0.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency converter</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Danfoss VLT5004 P T5 B20 EB R3</td>
</tr>
<tr>
<td>Maximum output current</td>
<td>5.6 A</td>
</tr>
<tr>
<td>Output frequency</td>
<td>0 to 1000 Hz</td>
</tr>
<tr>
<td>Supply voltage</td>
<td>380 to 500 V</td>
</tr>
<tr>
<td>Output voltage</td>
<td>0 to 100 % of supply voltage</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Incremental encoder</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>NM701NR3</td>
</tr>
<tr>
<td>Line counts</td>
<td>600 ppr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control board</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>dSPACE DS1103 PPC</td>
</tr>
<tr>
<td>Master processor</td>
<td>PowerPC 604e (400 MHz, 2 MB local SRAM, 128 MB global DRAM)</td>
</tr>
<tr>
<td>Slave processor</td>
<td>Texas Instruments TMS320F240 DSP (20 MHz, 3-phase PWM generation)</td>
</tr>
</tbody>
</table>

**Table 6.2.** Parameters for all system components
The DS1103 board have different input output ports, it is also possible to use digital or analogue signals to interact with this board. To expand the input-output possibilities the board is linked to a signal conditioning unit (SCU). This unit works as an interface between the board and all the other hardware used in the system. The SCU sends four signals to the inverter, three of them are the switching functions necessary to generate the output phases of the inverter. In the switching functions is also considered a dead time of 1.5 \( \mu s \) to avoid any possible problems which can be linked to a short circuit condition. The signal conditioning unit also provides a connection for the measurement of the DC-linkage voltage, used in the inverter, and a connection port for the signal received from an incremental encoder mounted on the shaft of the motor. Moreover, the current of the three phases are measured by LEM’s current transducers and also these informations are sent to the board through the signal conditioning unit. To have a graphical view of the whole system, it is reported in Fig. 6.2, the complete measurement setup mounted on the laboratory, during a test. In addition all the parameters of the different hardware components, used in the tests, are listed in the Tab. 6.2.

### 6.2 Simulink model for dSPACE

First of all, it will be explained the Simulink model, reported also in Fig. 6.3, used to directly control the motor in dSPACE.

#### 6.2.1 Blocks analysis

Looking at Fig. 6.3 it is evident that there are different blocks specifically designed to realize the interface between the dSPACE hardware, the DS1103 board, and the Simulink environment.
Mainly the blocks that realize the interface are the ones highlighted in blue. The one in the upper right is the block predisposed to elaborate the duty cycle informations and generates the PWM signals that have to be sent to the inverter. The generation of PWM in dSPACE is handled directly by the program and it can not be changed by user.

The other interface element is the *Measurement & Control* block. If the previous one generates the outputs to the control board, this block transform the output of the motor into input for the DS1103 board and consequently for the whole real-time software. In fact inside this block is realized the transformation from the analogue signals collected by the sensors to the digital world in which work all the control systems. In the Fig.6.4 is reported only this block to have a better view of the outputs, among which can be enumerated: the current in the three phases, the DC-linkage voltage (ud), the measured angle (wmmeas) and the enable signal. The currents in the three phases are measured by specific transducers that transform the current information into a voltage information. The latter is then sent to an analog-to-digital converter to obtain the value that can be used directly in comparisons.

![Diagram](Image)

**Figure 6.4.** Detail of measure and control block

The enable signal is instead obtained from a complex comparison, which takes into account all the maximum limits for the different variables in the system as currents, voltages, speed. For example, are fixed the maximum and minimum value for the DC-linkage voltage, if this variable is out of the defined range the enable signal is equal to zero, it means that the outputs are forced to zero. This operating mode is related to the presence of a multiplication, between the duty cycle signals and the enable signals, which operates before they reach the interface block. The
limits set in the block are for all the important values that can affect the correct operation of the motor. In fact, there are limits for the maximum current, for the maximum torque, for the maximum speed. If only one of these thresholds is exceeded the motor is stopped, to avoid damages. Moreover, some of the limits derive from the specific PMSM used and other instead are fixed by the inverter.

The main block, visible just in the middle of Fig. 6.3 is called Vector & Control. Inside this block is present the whole Simulink model discussed in Chapter 5 of this thesis. Only a few small changes have been made to adapt the signals received from the real motor to the ones used in the simulation step. There are different control signals among the inputs of this block. Like the signal that enable the filtering action (filter) to get a better estimation from the measured speed. The signal which discriminates between the two operating modes in which the system can run (init). The two operating modes are required due to the use of a IPM motor in which the presence of magnets generate a magnetic field in the rotor that have to be aligned with the one generated through the current, that circulate in the stator. The penalty for any lack of alignment is the impossibility to properly control the functionality of the motor. These two modes are the normal running mode and the alignment mode, which will be explained in detail in the next sections.

The Simulink model used in Vector & Control block is the one without the Model Order Reduction, which is implemented into another separated block called precisely Reduced model. In this block are enclosed all the parts that make up the flux estimation with the use of the MOR. In fact, as inputs can be enumerated the current in $\alpha - \beta$ coordinates (iss), exactly the one used in the current controller, and the measured angle. Instead, as output it gives the flux estimation for the three phases.

An other important part of the Simulink model is the one reported in Fig. 6.5 to highlight the blocks that are involved in that part. The aim of this part is to transform the information got from the incremental encoder, the instantaneous speed of the rotor, into the needed information for the vector controller, namely the measured angle. To realize this task are used different blocks. The first in the queue is the block which allows to discriminate from alignment mode and running mode, because during the alignment the angle information must kept fixed at a specific value, that for a sake of simplicity in this work will be zero degrees. This desired behaviour is achieved using the $y$ output, which is set to 0 in the alignment mode. In this way the product with anything else will be always zero. Instead, the $y$ output is set to 1 during the running mode, in this way the product output will be decided by the other branch. Branch along which the $x$ signal goes through two blocks in succession. The first one is the Discrete-time integrator that gives as output the angle value. After this block is necessary another one that limits the value of the angle between $-\pi$ and $\pi$ otherwise it is possible that at some point the variable will arrive at a very high value that will cause an overflow.
6.2.2 Running mode

The main operating mode, in which the motor is working for most of the time, is the running mode. In this mode it is possible to run the motor at any speed, decided by the user, within the speed maximum limits. Moreover, to enter in this modality it is necessary to pass through the alignment mode otherwise this mode does not work properly. In addiction, while the system staying in this mode is possible to change in real-time, through the control desk interface, the actual speed, the direction of rotation and all the variables are plotted to have a visual control of the whole system.

6.2.3 Alignment mode

To achieve the alignment is necessary to know exactly the position of the rotor. The measurement of the rotor angle is made by using an incremental encoder. First of all, it is necessary to underline that the encoder provides as a direct output the speed evaluation. To be precise, the encoder gives as output the number of pulses measured and knowing the specs of the encoder it is possible to evaluate the actual speed. This is an instantaneous speed controlled by the feedback system analyzed in details in Chapter 4 dedicated to the control. Therefore, to obtain the desired measurement it is necessary to integrate the speed, because space, and speed are related through the eq. (6.1), as it is well known from the kinematics [32].
\[ x_f = x_i + vt \]  \hspace{1cm} (6.1)

In which \( x_f \) represent the final value of distance, instead \( x_i \) is the initial value and \( v \) is the speed. The eq.(6.1) is correct for the system under analysis only if the speed is considering constant, this is right when the sampling frequency is far higher than the variation of the speed. Moreover, to have a correct estimation of the rotor angle it is important to align one specific position of the rotor with the zero angle of the encoder output. This procedure of alignment is specifically required for the permanent magnet synchronous motor because the rotor flux is generated by the attached magnets. If they are not properly aligned with the stator field, it will not be possible to control the motor because the angle information will be wrong. This operation need to be made before use the running mode in the motor control. For this reason the control system needs two operating modes, one for the alignment and the other to control the motor in normal condition. To exploit the possibility of operate in two different modes it is necessary to have a control variable that keeps the information about which mode have to be used. This variable can be changed in real-time from the control interface to switch from alignment mode to running mode.

In the specific case of this thesis work, the alignment is realized by increasing the magnetization current \( I_d \). This current creates a stator magnetic field, as a result the rotor, that has permanent magnets, will align itself with the stator field. From the experimental measurements, was observed that a current of 0.5 A is enough to align the rotor. However, to have a reasonable margin the \( I_d \) current will be switch from 0 A to 1 A.

In this phase carefully looking at the shaft, it is possible to see the small movement of the rotor that aligns itself with the field generated by the stator. The movement is very small because the machine has 6 poles. Furthermore, another proof of the right alignment is the presence of a holding torque. Before the alignment the rotor can be easily rotated just applying a small force, instead after it can no longer be moved due to the holding torque generated by the fields interaction.

After this phase of alignment, the physical angle and the measured angle are the same. Under this condition the control work properly, the motor follows the required speed decided by user through the interface realized in dSPACE environment.

6.2.4 Control desktop interface

The Control desktop interface is the part that connect the user with the DS1103 board, where is running the real-time software. It gives the possibility to modify in real-time all the variables under control on the motor as speed, running mode
and so on. The use of the control desktop interface is thoroughly explained in the Appendix A.

The Fig. 6.6 shows the desktop interface captured from the desktop computer in the lab during the execution of a test. This is the final version developed to have a simply and efficient way to control the motor, in the upper part are collected all the boxes to interact with the machine and in the bottom are instead placed the graphs to plot the actual value of the variables. In the upper part of the Fig. 6.6 is possible to see the different instruments used to send the informations to the motor. There is the ON/Value check box, when is verified the system is ON otherwise is OFF. Moreover, near by the first one there is the reset_trip/Value, which must be activated to reset the error register before running the machine otherwise the motor will not start. Very important is to reset the register and then remove the signal before running the motor, otherwise the register would continuously reset and at some point the protection of the motor may not be effective anymore. The box below called n_ref/Value has the functionality to define the actual required speed for the motor, in the example is 500 rpm, however it can be changed continuously between -1500 and 1500 rpm the nominal speed in both directions. An other important input box is the init/Value, through which can be defined the operating mode, as also explained in the previous section. In the example is set to zero due to the fact that the system was in running mode, otherwise it will be set to one for the alignment mode. The last interface box that allows to insert some data is the id_alignment/Value, through which the user can precisely define the value of the
id current used during the alignment mode to magnetize the system and then it is forced to zero during the running mode. Moreover, in the upper right is present the interface designed to choose all the parameters necessary to save in Matlab format the data showed in the graphs. As visual feedbacks are present two graphs, one for the rotor speed and the other for the flux in the three phases. Are also present few display to show some value that not require a graph as the value of the enable signal, the value of the DC-linkage voltage (Ud) and the error code (TRIP_IN/Value) to identify which kind of error is occurred.

6.3 Results of laboratory setup

In this section are analyzed the results obtained during the tests in the laboratory, in which was changed the speed reference, the magnetization current and all the main variables are plotted and analysed. The main variable is definitely the speed, because is directly used as output of the motor for any loads that would be attached to its shaft. Nevertheless, in the case of this thesis work is also relevant the flux estimation given as output because is necessary to make the comparison between the two models used for its estimation.

6.3.1 Speed graphs

Starting with the speed results obtained from the tests it is possible to evaluate the accuracy of the controller in following the nominal output. The Fig. 6.7 shows the rotor speed in a fixed time interval. Along the y-axis is reported the actual speed of the motor in rotation per minute (rpm), instead along the x-axis is reported the time. In Fig. 6.7, as listed in the legend, are reported both the measured speed with the blue continuous line and the reference speed numerically evaluated, marked with the red dotted line.

From the Fig. 6.7 it is evident that the control system works in a very good way because the blue and red line are almost always superimposed. The big difference is at the beginning, when the motor starts or changes speed. In this moment, the reference speed changes instantaneously, because is not a physical variable, instead the measured speed changes slower than the reference speed, acting as a first order system. An other difference is also in the constant part, where the measured speed is not exactly flat. The speed has some variation, due to the time interval necessary for the feedback loop to sense the real value and consequently acts if it is not the right one. The speed was changed abruptly at $t = 2s$ and $t = 12s$ through the real-time interface on the desktop computer and sent to the DS1103 board. The jumps in the speed are of 500 rpm at a time, to obtain a first jump from 0 to 500 rpm and another one from 500 to 1000 rpm.
In the Fig. 6.8 is instead represented an opposite variation with the respect of Fig. 6.7. Along the axes are reported exactly the same variables, for uniformity, the speed in rotation per minute along the y-axis and the time along the x-axis. In Fig. 6.8 are graphed two lines, the blue continuous one for the measured speed and the red dotted one for the reference speed, as reported in the legend. The main difference is that in this case is possible to see the behaviour of the system during the turn-off, or when the speed is reduced. The behaviour is still similar to a first order system for the real measured speed as expected. In Fig. 6.8 the speed jumps from 1000 to 500 rpm at 2s and from 500 to 0 rpm around 12s. Also in this graph the two lines, for measured and reference speed, are almost superimposed meaning that the real system follows the required speed properly.

The last figure for the speed is the one to demonstrate that the system can reaches the nominal speed of 1500 rpm and can keep it without problems. In fact, in Fig. 6.9 are reported the same two lines for measured and reference speed, also along the axes the variables are the same. It is evident that the controller works properly also in this condition that represent the limit speed for the aim of this
thesis.

6.3.2 Flux graph

One of the most important variables under analysis in this thesis is the flux, due to the fact that is the variable used to compare the new model, realized to estimate the parameters of the motor, the Model Order Reduction and the classical estimation through equations.

The Fig.6.10 shows a comparison between this two ways to estimate the flux. The results reported in the Fig.6.10 were obtained in the laboratory, whereas to those previously shown in other chapters obtained from computer simulations.

Along the y-axis is reported the absolute value of the flux and along the x-axis is reported the time. In the Fig.6.10 are reported the two values for the flux, the one estimated inside the Simulink block for the current controller and the flux
estimation realized through the Model Order Reduction indicated with (OIM). The continuous lines are used to mark the values related to the estimation with Simulink block, instead the dotted lines mark the estimation obtained with the MOR, as also reported in the legend.

Looking at the Fig.6.10 it is possible to see that the dotted and continuous lines are very similar, the shape is exactly the same, both have a sinusoidal shape and also the minimum and maximum values are perfectly matched. The only difference is a small phase shift between the two lines. This is probably related to a small delay accumulated along the path. Moreover, the shift is so small that it does not cause problems in getting a right estimate for the flux.

6.3.3 Current graphs

Another variable that can give useful information regarding the operation of the motor is the current.

Specifically in Fig.6.11 is graphed the pattern of the three phases currents Ia,
Ib and Ic. Along the y-axis are reported the current values in the three phases and along the x-axis is reported the time. In the legend are marked the different phases with different colours and the same curves are reported in augmented dimensions in Fig. 6.12.

The interesting result that can be viewed from Fig. 6.11 is the behaviour of currents for a wide time interval. Precisely, it is possible to see the shape of currents before the change in the reference speed. Analyzing in details, before the current peaks are less than 0.5 A, then when the speed changes, at $t = 2.8s$, they immediately jump to 1.5 A, this value is limited by the system otherwise it will be far higher. The last part of graph represents the moment in which the reference speed is almost reached and the currents start to decrease and shown again a value around 0.5 A.

Still considering the current graphs is interesting to plot the current in the d-q reference frame. In Fig. 6.13 it is shown the behaviour of the d and q current components. Along the y-axis is reported the value of both currents. Instead, along the x-axis is reported the time. The red dotted line represents the $I_d$ current and the
blue continuous line represents the $I_q$ current, as summarized also in the legend. The shape of the curves in Fig.6.13 is exactly as expected. In fact, the $I_d$ current, that is the magnetization current, is almost zero for the whole interval. This is correct because comes from the decision, made in this thesis, to use only the magnetic field generated by the permanent magnets, embedded in the rotor without influence them during the running mode. In the other hand, the $I_q$ current, which is the one used to generate the torque available at the motor shaft, changes considerably during the running mode accordingly to the required speed and torque. To highlight the variations that occur in the $I_q$ was chosen to plot exactly the interval of time that includes the starting of the motor. Moreover, it is possible to see in Fig.6.10 that at $t = 2.8s$, the $I_q$ current jumps abruptly to reach a plate area for which the value is defined by the torque limit because the current limit is far higher. When the required speed is reached the torque current returns to the value assumed before that was imposed the change in the speed. The evident ripple superimposed to the curves is linked to the closed loop control that measures the rotor speed and consequently
acts on the current. This effect does not give any problem in the specific condition under analysis, anyway in case of necessity the ripple can be reduced incensing the bandwidth allowed for the control feedback or employing an higher quality encoder, that produces a more precise evaluation of the measured speed.

6.3.4 Execution times comparison

Last but not least, it is reported the comparison of time measurements. To be more specific it is measured the execution time of the two fundamental blocks the one that implement the MOR and the current controller block in which is estimated the flux. This comparison is very important to validate the new technique and define the possibility of use it as main technique to control the motor.

The comparison gives a positive result because shows that the MOR is faster than the other flux estimation, as is visible also in Fig.6.14. This result allows to use the developed technique without problems, due to the fact that it will not
produces additional delays. Considering the average values obtained from different measurements, it is possible to state that the average execution time for the MOR is $1.34 \times 10^{-5}$ s instead for the other block the execution time is $1.69 \times 10^{-5}$ s. It is worth to evaluate the speed-up that can be achieved by this new technique with the eq.(6.2) reported below:

$$\text{Speed - up} = \frac{\text{time current contr}}{\text{time MOR}} = \frac{1.69}{1.34} \times 10^{-5} = 1.26$$ \hspace{1cm} (6.2)

This result is extremely important because using this technique that gives the possibility to control in a more accurate way the motor it is also achievable a faster control loop.

The time comparison is shown in Fig.6.14, where along the y-axis are reported the execution times for the two blocks and along the x-axis the time. As explained in the legend the blue curve represents the execution time for the current controller
instead the red curve illustrates the execution time for the MOR identified with OIM. However, looking at Fig. 6.14 seems that the execution times are a straight line. This is due to the fact that the variations present on the time values are quite small. To appreciate them is necessary to consider just one of the two curves present in Fig. 6.14, for example choosing the curve related to MOR. In Fig. 6.15 along the y-axis is reported the execution time for the MOR and along the x-axis the time. Due to the fact that is considered a smaller time interval it is evident the variations superimposed on the execution time as expected.
Figure 6.15. Execution time OIM
Chapter 7

Conclusions

To have a better overview is useful a brief recap of the path that has been followed during the thesis writing. The first point was the analysis of the context in which this thesis is inserted. Particularly looking at the interesting market for electric cars and more in general for electric vehicles. The demand of these type of vehicles is currently growing beyond than the most optimistic forecasts. The only drawback related to this fast growth is the increase in energy demand that have to be supplied. It is in this framework that the increase of efficiency is one of the fundamental aspects to reduce the losses and avoid useless energy waste. From this analysis comes the choice made in the thesis to use an IPM motor, which shows higher overall efficiency than other solution available on the market. Since, the main aim of the thesis has been realize an efficient digital control technique for this kind of motor, the next step was the theoretic analysis of the main components that realize the overall system: the inverter, the motor and the control. After building a solid theoretical basis, it was started an accurate review of the literature in the field of permanent magnets motor control, in order to choose the best control strategy. I described the working principle of each single block, in which is divided the model, and I reported the result obtained in the simulations. The final step was to describe the model used in the dSPACE environment and also the laboratory setup mounted for the tests. Afterwards it was discussed and analyzed the experimental results that I obtained in the laboratory tests for the different variables of interest as currents, voltages, speed and flux.

From the detailed analysis of the collected data, it is possible to confirm the extremely good performance of the new strategy adopted to obtain the flux estimation. This is evident from the results obtained in the simulations and in the laboratory, which underline the almost perfect matching, between the curves obtained from the MOR technique and the curves obtained from the estimator present in the current controller, as evident in Fig. 6.10. Moreover, the computational time comparison points out the possibility to get an improvement in the speed of the
control path due to a reduction in the time necessary for the evaluation with the use of the new strategy analysed in this thesis. Summarizing the main advantage of the tested technique is the possibility to get a very accurate estimate of the variables in real-time conditions. This leads to a faster feedback loop that reduces the useless movements of rotor, related to delays in the feedback path. Furthermore, thanks to this technique is reached a higher efficiency, which also means less heat produced and therefore it makes easier to cool down the system.

7.1 Future developments

Analysing the results obtained during the test phase it is possible to state several starting points for future development and improvement of what has been done in this thesis:

- **Improvement in laboratory setup:** replacing the absolute encoder used by taking one with higher performance, in order to have a better and more precise speed evaluation and as a consequence a more accurate estimation of the rotor position. In this way the ripple on the current curves would be smaller and the overall efficiency higher.

- **Furthermore investigations:** select an other IPM motor in a different power range, for example far higher with the respect of the one used in this work. Testing this new configuration in a variety of condition as different speeds and various loads connected to the shaft of the motor, in order to have a wider range of data to validate the performance of this new technique.

- **Control hierarchy:** use as main and unique control unit the one realized with MOR eliminating the other block used to compare the results. This step is the natural outcome of the positive results obtained in this thesis and allows to increase the speed in the control loop improving the overall performance of the system.
Appendix A

dSPACE environment

The dSPACE system is also known as rapid prototyping tool. This name is due to the fact that the generation of working prototype is rapid and direct. The main reason that leads to this positive property is that the applied system is exactly the same of the tested one, thus is very useful to reduce possible translation errors. Before the introduction of real-time systems, the development of a new product was very long. The reason behind this long developing time was hidden in the differences between the real word and the model used in the simulations. The necessity to reduce the developing time has increase the growth of the real-time systems, also helped by the evolution of real-time processors. In the field of real-time systems, dSPACE is one of the most used system thanks to the ties connection with Matlab-Simulink environment. This close cooperation is related to the very easy integration between dSPACE and Simulink, moreover the generation of the code can be started directly into Matlab. The dSPACE system is an integration between software and hardware. There are a wide range of possible hardware to connect, which is composed from several real-time processors mounted on a single board, that include also a large choice of inputs/outputs.

A.1 The dSPACE ControlDesk

The ControlDesk software is the PC graphic interface between the DS1103 board, on which is running dSPACE, and the user. The software is composed by different modules:

- Experiment management, that allows to control all the data related to an experiment. Which can be loaded directly just with the use of the command "Built" in the Simulink environment.
• Hardware management, that allows to configure the dSPACE hardware and handles the applications directly through the graphic interface.

• Instrumentation kits, that offers a wide range of virtual instrumentations and for the acquisition of new data.

One of the most useful features of the ControlDesk is the possibility to store the behaviour of different variables of interest present in the real-time application. Those variables are saved in Matlab format, then it is very easy to access the data stored through the Matlab Workspace. As a result this function offers the possibility to visualize the variables evolution through an oscilloscope or a digital display and to modify in real-time the parameters under control.

A.2 Working environment description

The working environment of ControlDesk is reported in Fig. A.1 and it is divided in three main parts:

1. Navigator: is the section underlined with the red square in Fig. A.1, and shows three sections:
   • Experiment: describes and reminds all the programs and layouts associated with the experiment.
   • Platform: describes all the simulation platforms with which is possible to interface the ControlDesk.
   • Instrumentation: describes all the opened layout and for each of them keeps trace of the instruments used.

2. Tool window: it is the area underlined in green in Fig. A.1, different function are available:
   • Log view: window in which is possible to view the error and warning messages.
   • File selector: allows to select the application to download on the board in a graphical way.
   • Interpreter: window in which there are the messages of Python interpreter that can be used in ControlDesk.
Figure A.1. The dSPACE ControlDesk environment

• Variable manager: in this window, reported in more details in Fig. A.2 are visualized, in the form of a tree list, all the variables of the program that is running on the board and their properties.

Figure A.2. The dSPACE Tool window

3. Work area: is the graphic area, also called layout, reserved for the program running on the board. Through the layout it is possible to interact with board
in real-time, for example changing the parameters, visualizing the temporal trends for the variables of interest, capturing the data.

- Toolbar: when the program is loaded on the board it is possible to control it through the command on the upper part of the window in which are present the buttons, Run, Stop. From this area it is possible also to change the way in which is used the program, between three options:
  - The **Edit mode**, in which the layout is built in the work area.
  - The **Test mode** used to test the functionality of the model realized.
  - The **Animation mode**, in which the layout is utilized by the application running into the board.

### A.3 Layout example

The layout, initially empty, is a working area in which the different instruments can be placed. To realize this action is necessary to select from the right part of the working area the desired instrument, the second step is to use the mouse to draw the area that will be reserved for the selected instrument. After this the area will be underlined with a red board, because no variable is associated with instrument. To associate the variable is necessary to select it in the Variable Manager window and drag it into the square, which delimits the area of instrument. As a consequence the board of the area it will became grey and the association will be successful. Moreover, an other possibility is to use different layout at the same time, this can be very useful because the graphs reported in any single layout can have the right size to properly look at the behaviour of the variables. The important aspect to underline is to avoid useless graphs, due to the heavy slow-down in the reaction of the real-time interface that they can cause.

An example of this procedure, related to the thesis work, is reported in the Fig. A.3.

In this example the working area is almost completely covered by different kinds of instruments, two plots for speed and flux, the interface to control the recording and few display to visualize the value of important variables, such as the errors, enable, etc.
Figure A.3. Example layout dSPACE
Appendix B

Matlab codes

In this appendix are reported the Matlab files used in the different model developed in the Matlab-Simulink environment. They will be divided in two sections, one section that collects the file employed in the model for the simulations and another section for the model used in the laboratory tests.
B.1 Files for simulation model

**Initialization.m**

clear; clc; close all;

%%% Time parameters for simulations
Ts = 1/5e3; % Sampling period of the flux-control loop
Ts2 = Ts; % Sampling period of the slower subsystem
Tstop = 2.5; % Simulation stop time

%%% 2.2 kW (Motor parameters)

% Parameters
Ld = 36e-3;
Lq = 51e-3;
psipm = 0.545;
Rs = 3.59;
p = 3;

% Rating (nominal values)
PN = 2.2e3; % Power
UN = 370; % Voltage
IN = 4.3; % Current
nN = 1500; % Rotational speed
fN = p*nN/60; % Frequency
TN = p*PN/(2*pi*fN); % Torque

% Other parameters
Udc = 540; % DC-link voltage
Jtot = 0.015; % Moment of inertia

% PWM parameters
uerr = 0; % Nonlinear inverter voltage drop
dmin = 0; % Minimum pulse limitation

%%% Controller parameters
alphac = 2*pi*200; % Flux-control bandwidth
alphas = 2*pi*2; % Speed-control bandwidth

%%% Simulate and plot results
base_vals; % Calculate base values (used in plotting figures)
Tmax = 2*TN; % Maximum torque limit for the speed controller
imax = 2*iB; % Maximum current limit
id_min = 7.5; % Constant current in the d-axis
w0 = 200; % Reference speed step
slm('ipm'); % Simulate the model
fig; % Plotting


B – Matlab codes

Speed controller

```matlab
function [Teref, Tei_new] = fcn(wmref, wm, Teref_lim, Tei, Ts2, Jtot, alphas)

beta = exp(-alphas*Ts2);  % Discrete-time pole

% Gains of the controller
% Direct-discrete pole placement, where the torque controller is assumed
% to be otherwise ideal, but having one sampling-period delay
% (the gain k2 is nonzero due to this).
k1 = Jtot/Ts2*(beta - 1)^2;
k2 = Jtot/Ts2*(3 - 4*beta + beta^2);
k2 = 2*(1 - beta);
k1 = Jtot/Ts2*(1 - beta);

% Control law
% Torque reference without limits
Teref = kt*wmref + Tei - k1*wm - k2*Teref_lim;
```

Reference calculation

```matlab
function [isref, Teref_lim] = refs(Teref, Ld, Lq, psipm, p, id_min, imax)

% Define the reference values for Id and Iq components
idref = id_min;
iqref = Teref/(1.5*p*((Ld - Lq)*idref + psipm));

abs_isref = sqrt(idref^2 + iqref^2);

if abs_isref > imax
    iqref = sign(iqref)*sqrt(imax^2 - idref^2);
end

% Reference current
isref = [idref; iqref];

% Limited torque reference
Teref_lim = 1.5*p*((Ld - Lq)*idref + psipm)*iqref;
```

Current controller

```matlab
function [dabc, us_new, psis_new, thetam_new, wmf_new, usi_new, Te] = fcn(Udc, yref, iss, ths, thtam_meas, us, psis, thetam, wmf, usi,...
    Ts, Rs, Ld, Lq, psipm, alphac, p, dmin, uerr)

% Parameters

% PWM parameters
inv_ithr = 71.4286; %dt = .0036;
% Auxiliary parameters
```

90
J = [0, -1; 1, 0]; I = eye(2); O = zeros(2);

% Position sensor
% Store the old value for the speed calculation
thetam_old = thetam;
% Rotor angle is measured
thetam = thetam_meas;

% Coordinate transformation
is = expm(-thetam*3)*iss;
id = is(1); iq = is(2);
psid = psis(1); psiq = psis(2);

invLd = 1/Ld; invLq = 1/Lq;
C = [invLd, 0; 0, invLq]; % Inverse inductance matrix
L = [Ld, 0; 0, Lq]; % Inductance matrix

abs_psiF = psipm;
psiF = [abs_psiF; 0]; % PM flux
% PM-flux vector

% Position sensor
% Calculate the rotor angular speed
dthetam = thetam_meas - thetam_old;
while(dthetam > pi)
dthetam = dthetam - 2*pi;
end
while(dthetam < -pi)
dthetam = dthetam + 2*pi;
end
wm = dthetam/Ts;

%% Calculate the system matrices A and B
%% Series expansion
Ac = -Rs*C - wm*J;
A = I + Ts*Ac + Ts^2*Ac^2/2;
if(wm == 0)
    comp = .5*Ts*wm/(sin(.5*Ts*wm))*expm(-.5*Ts*wm*J);
else
    comp = expm(-.5*Ts*wm*J);
end
B = Ts*(I + Ts*Ac/2)*comp;
bF = Ts*(I + Ts*Ac/2)*[Rs*invLd; 0];

%% Flux observer
% Position sensor
% 1st-order LPF
wn = 2*pi*100;
b = exp(wn*Td);
wmf_new = b*wmf + (1 - b)*wm;
% Store the rotor angle for the speed calculation in the next sampling period
thetam_new = thetam_meas;

%% Observer gain
% Desired closed-loop matrix Ac
k = 3;
expa = exp(-k*.5*Rs*(invLd + invLq)*Ts);
ac11 = expa*cos(wm*Ts);
ac22 = ac11;
ac21 = -expa*sin(wm*Ts);
Ac = [ac11, -ac21; ac21, ac22];

% Update the observer states
psis_new = A*psis + B*us + bF*abs_psiF + Ko*(psis - psiF - L*is);

% Torque estimate
Te = 1.5*p*(psid*iq - psiq*id);

%% Current control
y = is;
%% Transform system matrices
A = C*A/C; B = C*B;

%% Block-pole placement
betac = exp(-alphac*Ts);

% IMC-type pole locations (Harnefors et al.)
A0 = 0;
A1 = betac^2*I;
A2 = -2*betac*I;
B1 = (1 - betac)*I;

%% Gains
Kt = B\B1;
K2 = I + B\(A2 + A)*B;
K1 = B\(A1 - A) + K2/B*(I + A);
K1 = K1 + B*A0 - K2/B*A;

%% Calculate the control output
usref = Kt*yref + us1 - K1*y - K2*us;

% Reference voltage in stator coordinates
ussref = exp((thetam + Ts*wm)*U)*usref;

%% Symmetrical suboscillation PWM
% Phase-voltage references
ua_ref = ussref(1);
ub_ref = .5*(-ussref(1) + sqrt(3)*ussref(2));
uc_ref = .5*(-ussref(1) - sqrt(3)*ussref(2));

% Vector of phase-voltage references
uabc_ref = [ua_ref; ub_ref; uc_ref];

% Symmetrization of the references by adding the zero-sequence voltage
u0 = .5*(max(uabc_ref) + min(uabc_ref));
uabc_ref = uabc_ref - u0;

% Limiting the reference voltages such that no phase error is caused.
% If all phase voltage reference are realizable (below Udc/2), m = 1 holds.
B – Matlab codes

kpwm_max = 1 - 2*dmin; % Minimum pulse limitation
m = max([2*uabc_ref/(kpwm_max*Udc); 1]);
uabc_ref = uabc_ref/m;

% Duty cycles, which can be directly used for triangle comparison,
% if the triangle wave varies between 0...1
da = .5 + uabc_ref/Udc;
db = da - dabc(2); dc = dabc(3);
% Realized voltage vector (without the deadtime compensation)
% calculated from the limited duty ratios
usslim = [(2*da - db - dc)/3; (db - dc)/sqrt(3)]*Udc;

% Limited voltage in estimated rotor coordinates
uslim = expm(-(thetam + Ts*wm)*J)*usslim;

%% Update the integral state (the last term is the antiwindup)
usi_new = usi + Kt*(yref - y + Kt\(usslim - usref));

%% Deadtime compensation for the observer
% Phase currents
ia = iss(1); ib = .5*(-iss(1) + sqrt(3)*iss(2)); ic = .5*(-iss(1) -
  sqrt(3)*iss(2));
siga = (2/pi)*atan(ia*inv_lthr);
sigb = (2/pi)*atan(ib*inv_lthr);
sigc = (2/pi)*atan(ic*inv_lthr);
uss_comp = userr*[(2*siga - sigb - sigc)/3; (sigb - sigc)/sqrt(3)];
uss_new = usslim - usss_comp;
us_new = expm(-(thetam + Ts*wm)*J)*uss_new;
B.2 Files for laboratory model

Initialization.m

clear;
addpath Utility

% Machine parameters
Brem=1.15; % Remanent induction of the magnets
Is=4; % Rated current amplitude per phase
freq=75; % Frequency
i=1/(4*freq); % Considered instant
current_angle_FEM=0; % Current angle
Display=0;
Tol=1e-6; % Tolerance of iterative solver error

% Definition of parameters needed for MOR technique
alfas = -180:10:180; % angle range
It = linspace(0, 4.143, 4); % current range
Ncur = length(It);
Nalf = length(alfas);\nNtot = Ncur*Nalf;

snap=open('snap_same_currentangle_rotationangle100.mat');
snap=snap.snap;

[A,M,Sig,V] = podbases(snap);
Usig=A*Sig(1:size(A,2),:);
z=1:size(A,2);

% Evaluation of vector potentials
[X,Y,Z] = ndgrid(It, alfas,z);
V_tst=reshape(V(:,:,1:size(z,2)),size(alfas,2),size(It,2),size(z,2));
v=permute(V_tst,[2 1 3]);
A_init=zeros(1379,1);

% Evaluation of the Coe coefficient
load('Coe_m.mat');
I1=find(Coe1(:,1,:));
I2=find(Coe2(:,1,:));
I3=find(Coe3(:,1,:));
Coe1=Coe1(:,I1)*Usig(I1,:);
Coe2=Coe2(:,I2)*Usig(I2,:);
Coe3=Coe3(:,I3)*Usig(I3,:);
C=[Coe1(1:size(z,2));Coe2(1:size(z,2));Coe3(1:size(z,2))];

% Inverter parameters
Udc = 540; % DC-link voltage
Ts = 125e-6; % Sampling period, switching frequency fsw = 1/(2*Ts)
Ts2 = Ts;

% Controller parameters
alphac = 2*pi*150; % Current-controller bandwidth 300 Hz
alphas = 2*pi*2; % Speed-controller bandwidth 10 Hz
tau_max = 20; % Maximum torque
psiR_ref = 0.9;  % Rotor-flux reference

%% 2.2 kW motor
% Parameters
Ld = 36e-3;
Lq = 51e-3;
psipm = 0.545;
Rs = 3.59;
p = 3;
% Rating
FN = 2.2e3;
UN = 370;
IN = 4.3;
nN = 1500;
% Rotational speed
fN = p*nN/60;
% Frequency
TN = p*FN/(2*pi*fN);
% Torque

% Other parameters
Udc = 540;
Jtot = 0.015;

% PWM parameters
uerr = 0;
dmin = 0;

%% Simulate and plot results
base_vals;
Tmax = 2*TN;
% Maximum torque limit for the speed controller
imax = 5;
% Maximum current limit (in the speed controller)
id_min = 0;
% Constant current in the d-axis
id_alignment = 0.1;
% Current used for alignment of rotor
n_ref = 50;
% Reference speed in rpm
wm0 = (pi*n_ref)/30;

% Protections
i_max = 10;
% maximum current [A]
n_max = IN*60/p*2;
% maximum speed 120% synchronous speed [rpm]
Ud_max = 750;
% maximum dc-voltage [V]
Ud_min = 400;
% minimum dc-voltage [V]
tau_e_max = 3;

% inverter parameters
fs = 1/Ts;
% switching frequency [Hz]

% Encoder time constant
TM = 1/fs;
% period of encoder counter reading [s]
INC_LINES = 600;
% number of pulses/revolution

% Scaling factors
% scaling factor for current measurement with LEM-BOX [A/V]
SCALE_ADC_CUR = 25.5;
% scaling factor for DC-voltage for VLT5004 [V/V]
SCALE_ADC_VOL = 991.0;
% scaling factor for actual torque measurement [Nm/V]
SCALE_ADC_TQ = 50.0;
% scaling factor for reference torque DAC for Bivector [Nm/V]
SCALE_DAC_TQ = -1/50.0;
Bibliography


