

DEVELOPMENT AND TEST OF INDUCTION MOTOR CONTROL DRIVE FOR A VERSATILE POWER ELECTRONICS CONTROLLER

Master Thesis - presented on 18.09.2015
Department of Industrial Engineering
University of the Study of Padua (UNIPD)
to obtaining Master degree in Electrical Energy Engineering by

Michele Zendri *matr.1070755*

developed at:
Laboratoire d'électronique industrielle
École Polytechnique Fédérale de Lausanne (EPFL)

Supervisors:
Prof. Silverio Bolognani, thesis supervisor at UNIPD
Prof. Alfred Rufer, thesis supervisor at EPFL
Dr. Simon Delalay, lab. assitant at EPFL

Padova, UNIPD, 2015

Ai miei genitori ...

Acknowledgements

I wish to thank all those who helped me in the realization of this project.

First, I would like to warmly thank Prof. Alfred Rufer, my supervisor from École Polytechnique Fédérale de Lausanne, for his friendly hospitality in the lab, for having been close in times of need, transmitting great knowledge and valuable suggestions.

At the same way, I wish to express my deepest thankfulness to Prof. Silverio Bolognani, my italian supervisor from Università degli studi di Padova, who, despite the geographical distance, has been always available and willing to help me, giving interesting advices.

Also a special thanks to Dr. Simon Delalay, Mr. Tiago Nabais and Dr. Rosa Martel Danoary Tsirinomeny for their support, always present and very helpful.

Another gratitude to Dr. Nicolas Cherix, Mr. Matthias Lambert, Mr. Roberto Zoia and Mr. Federico Uriel for the technical support without which it would not have been possible the realization of this thesis.

My last thanks is to all my family and friends, who make me feel home wherever I am, and a special mention to Laura for having been alongside me all the way.

Lausanne, 20 August 2015

M. Z.

Riassunto

Questo lavoro di tesi vuole essere un esempio di implementazione di una strategia di controllo vettoriale per gli azionamenti di motori ad induzione. Allo scopo si è utilizzata una piattaforma di controllo di nuova generazione chiamata “*BoomBox*”, prodotto recentemente sviluppato dalla compagnia *Imperix S.A.*, composta da *DSP* ad alta prestazione e *FPGA*. La periferica è inoltre dotata di sistemi per l’acquisizione dati, è liberamente programmabile e facilmente interfacciabile con la maggior parte dei sensori e trasduttori presenti nel mercato. Questo strumento progettato allo scopo di facilitare i ricercatori in ambito della *power electronics*, e quindi in ambienti di *R&D*, semplifica molto l’implementazione di una *PWM* e del controllo tramite fibra ottica di switch controllabili in apertura e chiusura.

Le qualità del motore ad induzione ormai note da tempo all’industria, per via della sua robustezza, economicità, vita lunga e ottime prestazioni in termini di rendimenti elettrici, lo rendono un motore estremamente versatile a molte applicazioni in ambito industriale dove viene richiesto un preciso controllo della posizione dell’asse rotorico e/o della velocità di rotazione.

Nella prima parte di questo lavoro vengono mostrate le equazioni di un motore ad induzione, che portano ad un modello elettrico equivalente nella particolare situazione di regime sinusoidale, grazie al quale si possono determinare i principali parametri elettrici del motore. Grazie a questi, le equazioni differenziali possono essere particolarizzate per il motore scelto, e si può quindi riprodurre un modello del motore utilizzando uno strumento informatico quale *Simulink*. Il capitolo 2 è dedicato a questo modello e alla sua verifica, alla modalità di acquisizione dei parametri elettrici statorici e rotorici, al circuito elettrico equivalente e quindi alle equazioni che descrivono potenza, coppia e flussi.

Il capitolo 3 mostra come le prestazioni di un azionamento scalare in situazioni di transitorio siano migliorabili utilizzando della strategie di controllo vettoriale. Per prima cosa vengono riadattate le equazioni in modo da imporre delle tensioni statoriche tali che il flusso statorico rimanga a $1 p.u.$ La strategia di controllo viene simulata e se ne riportano le prestazioni. Nel paragrafo 3.2 viene invece mostrato il funzionamento dell’azionamento *FOC* indiretto, se

Acknowledgements

ne descrivono le equazioni e vengono progettati gli anelli di corrente e di velocità. Si riportano al contempo alcune problematiche implementative che hanno per esempio portato alla scelta di un ciclo di isteresi nei parametri del regolatore di velocità. Infine si riportano le simulazioni del controllo, con la risposta del sistema a gradini di velocità.

Il capitolo 4 è invece dedicato alla descrizione sia hardware che software dell'azionamento sviluppato. La piattaforma di controllo viene descritta e se ne riportano i parametri per la configurazione dei sensori utilizzati, i *LPFs* scelti e le motivazioni. Vengono mostrati i segnali dell'encoder utilizzato per il calcolo della velocità rotorica e come, processandoli e utilizzando un *PLL*, venga poi dedotta la velocità di rotazione dell'albero motore. Si riporta poi lo schema elettrico dell'inverter impiegato, e le verifiche del suo corretto funzionamento. Allo stesso modo, è stato realizzato e verificato il corretto funzionamento del sistema di frenatura, utile in caso i flussi di potenza si dovessero invertire. Infine viene fatta una descrizione della macchina a corrente continua utilizzata come carico meccanico da applicare all'albero motore, in modo da testare le prestazioni dell'azionamento anche sotto carico.

Il capitolo 5 riporta i risultati della strategia implementata, *step-by-step* di ogni singolo blocco implementato. Partendo dalla situazione di assenza di carico, i blocchi di trasformazione vengono implementati per le correnti misurate e le tensioni imposte. Il modello del motore viene verificato in un controllo a catena aperta. Vengono poi verificati i blocchi di disaccoppiamento degli assi e di compensazione della f.e.m. Infine si riportano le risposte del sistema a catena chiusa a gradini di velocità o di corrente, sia in condizioni di vuoto che di carico.

Si conclude il lavoro con delle considerazioni finali, riportando ciò che è stato svolto e ciò che è stato realmente formativo, le difficoltà incontrate e le possibilità future su ulteriori sviluppi di questo lavoro. Viene inoltre riportato il codice scritto per l'implementazione della strategia e i suoi flow chart per una più facile interpretazione.

Abstract

This thesis wants to be a guide for the implementation of a vector control of an induction motor. Will be seen the theoretical basis, leading to the creation of the first scalar controls used in the past. Will be seen some simulations using Simulink and their performances. It will be seen then, as a vector control can lead to higher performances even in transients and in fast dynamics.

It will be shown how we should define the parameters of such a control system, from the knowledge of the motor parameters to the current and speed regulators. It will analyze also the hardware aspect; as the shaft's speed can be derived using a rotary encoder, the remaining part of the sensors and how to use correctly a control platform, called "BoomBox" developed by the company Imperix. Will be also seen the overall performances of the power converter and the braking system.

Finally, it will be analyzed the results and performance of the implemented vector control, reporting measures taken at the workplace.

Key words: Induction motor, Power electronics, Field oriented control, Electrical drives

Contents

Acknowledgements	i
Riassunto	iii
Abstract	v
List of figures	ix
List of tables	xiii
Introduction	1
1 Induction Motor and Electrical Drives	3
1.1 Induction Motor	3
1.2 Electrical drive control	4
2 Induction Motor Modeling	7
2.1 Dynamic behaviour and space vectors	7
2.2 Stator and rotor equations in $d^k q^k$ reference frame	9
2.3 Stator and rotor equations in $d^s q^s$ stationary reference frame	10
2.4 Power and electromagnetic torque	11
2.5 Equivalent electric circuit	12
2.6 Specification of the motor	13
2.7 Simulink model and results	16
2.8 Verification of the Steady-State condition	19
2.8.1 No Load	19
2.8.2 Rated Load	19
3 Induction Motor Control	21
3.1 Imposing the stator flux through the stator voltage	21
3.1.1 Stator and Rotor equations in $d^r q^r$ rotary reference frame at synchronous speed	21
3.1.2 Imposing the stator flux through the stator voltage modulus	23
3.1.3 Imposing the stator flux through the stator voltage components	29

Contents

3.2	Imposing the rotor flux through the components of the stator currents (FOC - <i>Field Oriented Control</i>)	34
3.2.1	Current loops	37
3.2.2	Design of the current regulators	39
3.2.3	Design of the speed regulator	41
3.2.4	Simulation results	44
4	The Plant	47
4.1	Control platform	48
4.2	Sensors and Encoder	49
4.3	Inverter	55
4.4	Braking chopper	57
4.5	DC Machine	59
5	Experimental results	61
5.1	Open-loop	61
5.1.1	Park transformation blocks	61
5.1.2	Motor model	66
5.2	Closed-loop <i>No Load</i>	68
5.2.1	Current regulators	68
5.2.2	Speed regulator	69
5.3	Closed-loop <i>Under Load</i>	70
6	Conclusions and future prospects	73
A	Flow chart of the Algorithm	75
B	Code of the control system	79

List of Figures

1.1	Induction motor scheme	4
1.2	Principle Scheme of an electrical drive for induction motors	5
2.1	Common reference system	9
2.2	Space vector equivalent circuit in an arbitrary reference frame	12
2.3	Space vector equivalent circuit in a stationary reference frame	12
2.4	Sinusoidal steady state monophas electrical circuit	12
2.5	Equivalent monophas electrical circuit of the IM	15
2.6	Structural diagram of the IM in terms of spacial vectors	17
2.7	Simulink blocks represent the differential equation of the IM	17
2.8	Simulation results	18
2.9	Simulation representing stator voltage and current phase shift of a phase in <i>No Load</i> condition	20
2.10	Simulation representing stator voltage and current phase shift of a phase in <i>Rated Load</i> condition	20
3.1	Modified structural diagram of the IM with stator voltage imposition	23
3.2	Steady state characteristic for modulus of stator voltage u_s for $\Psi_s = 1$	24
3.3	Structural diagram of the IM with stator flux imposition through the modulus stator voltage	24
3.4	Transient phenomena during a rotor frequency variation at $n=1$	25
3.5	Spatial vector Ψ_s and i_s during a rotor frequency variation at $n=1$	26
3.6	Transient phenomena during a rotor frequency variation at $n=0$	27
3.7	Spatial vector Ψ_s and i_s during a rotor frequency variation at $n=0$	27
3.8	Control circuit scheme to impose the stator flux through the modulus stator voltage	28
3.9	Steady state characteristic for componenets of stator voltage u_{sd}, u_{sq} for $\Psi_s = 1$	29
3.10	Structural diagram of the IM with stator flux imposition through the components stator voltages	30
3.11	Transient phenomena during a rotor frequency variation at $n=1$	31
3.12	Spatial vector Ψ_s and i_s during a rotor frequency variation at $n=1$	31
3.13	Transient phenomena during a rotor frequency variation at $n=0$	32
3.14	Spatial vector Ψ_s and i_s during a rotor frequency variation at $n=0$	32

List of Figures

3.15 Control circuit scheme to impose the stator flux through the components stator voltage	33
3.16 FOC reference frame	34
3.17 FOC Induction motor model	35
3.18 Control circuit scheme FOC	36
3.19 Bode diagrams of Y_d (blue) and Y_q (red)	38
3.20 FOC currents control scheme with decoupling and compensation	38
3.21 FOC currents control scheme with decoupling and compensation	39
3.22 Bode diagram of GH_d (blue) and GH_q (red)	41
3.23 Bode diagram of W_q (red) and W_q^* (blue)	42
3.24 Speed-Loop scheme	42
3.25 Hysteresis speed regulator parameters	43
3.26 Transient phenomena during a speed reference variation in No Load condition	44
3.27 Transient phenomena during a speed reference variation in Rated Load condition	44
3.28 Example of speed reference and the tracking of the motor	45
4.1 Workplace 1	47
4.2 Workplace 2	48
4.3 Front panel view of the BoomBox	48
4.4 Block diagram of the analog inputs of the BoomBox	49
4.5 Currents and DC bus voltage in short-circuit operation	50
4.6 Logical scheme for the detection of the rotor speed in Sin/Cos encoders	51
4.7 Encoder mounting	52
4.8 Sin/Cos encoder output signals	52
4.9 General structure of a single phase PLL	53
4.10 Step of acceleration/deceleration and rise time of the PLL from 0 to nominal speed and backwards	54
4.11 Different speed steps and response of the PLL	54
4.12 Electrical scheme of the inverter	55
4.13 Inverter	55
4.14 Inverter typical PWM	56
4.15 Typical operation of the Braking Chopper	57
4.16 Braking Chopper	58
4.17 PCB electrical scheme of the converter used for the braking chopper system . .	58
4.18 Independent excitation of a DC machine	59
5.1 Verification of Park block for the stator voltages	62
5.2 Verification of Park block for the stator currents	62
5.3 Verification of Park current block for inverted stator voltages	63
5.4 Verification of Park blocks at $f_{s_ref} = 0.5p.u.$	63
5.5 Verification of Park blocks at $f_{s_ref} = 0.2p.u.$	64
5.6 Speed and stator currents in transient from $f_{s_ref} = 0p.u.$ to $f_{s_ref} = 1p.u.$. . .	64
5.7 dq statoric currents with dq stator voltage imposition in steady state $f_{s_ref} = 1p.u.$	65

5.8	Speed and stator currents in transient from $f_{s_ref} = 1p.u.$ to $f_{s_ref} = 0.5p.u.$. .	65
5.9	dq statoric currents with dq stator voltage imposition in steady state $f_{s_ref} = 0.5p.u.$	66
5.10	First order lag block diagram	67
5.11	Validation motor model	67
5.12	Validation of current regulators R_{id}	68
5.13	Validation of current regulators R_{iq}	68
5.14	Speed step from 0 to half of the nominal speed	69
5.15	Speed step from 0 to the nominal speed	69
5.16	Example of speed reference and the tracking of the motor	70
5.17	Speed step from 0 to half of the nominal speed under load	71
5.18	Speed step from 0 to the nominal speed under load	71
5.19	Example of Load step ($5Nm$)	72
5.20	Example of speed reference and the tracking of the motor under load	72
A.1	State diagram of the USER commands	75
A.2	Flow Chart of RAMPSTART & OPENLOOP mode	76
A.3	Flow Chart of CLOSEDLOOP mode	77

List of Tables

2.1	Nominal Values of the IM	13
2.2	Measured values of <i>No Load</i> and <i>Short Circuit</i> tests	14
3.1	Current Regulators Parameters	40
4.1	Sensors and Encoder	50
4.2	Sensors and Encoder parameters for Bbox configuration	51

Introduction

Nowadays, one of the big problem and discuss topic in our society is the energy. Since the man discovered the electrical energy, has been very fascinated from its properties, manifestations and potentiality. From the early onsets of awareness of its existence, the man did lots of steps to know it and manage it better. Electrical energy has lots of advantages respect to other types of energy. It is easily transportable, convertible and let us to reach high efficiencies in different energy systems. For these and more reasons, electrical energy will be our future, our sureness and our way to improve the quality life of many people.

In the last years, renewable energy has become more and more important in our energy productions, bringing a huge improvement to the environmental impact respect to the fossil fuels. On the other side, not programmable energy placed on the electrical grids, gives lots of troubles in terms of frequency stability and control of the power flows. With this view, it is highly important to know the root of the generation, being able to control it and manage it in the best way.

Induction machines plays an important role on generation, specially for wind turbines or micro hydro installations, and another big part in the electrical consumption in the industry field, as induction motors. Its control is important to reach high performance for various industrial plants.

The motion control area is a field that implement the control theory to the power electronics. This two areas are almost always related but at the same time are increasingly discussed as different, since each one has its separate development and its particular expert engineers.

Four years ago a versatile control platform was developed from the Industrial Electronics Laboratory at EPFL dedicate to the power electronics applications. This system is made of high performance DSP and FPGA and embeds also many peripherals for data conditioning and acquisition as well as communication purposes. The goal is to help R&D engineers to speed up their power electronic developments, in an open-software system that customers are welcome to self-appropriate and edit up to their own needs [8].

In this view, motor drives could be easily implemented by customers; in this thesis, it will be shown how an induction motor can be control with this platform developing and testing a motor control library.

1 Induction Motor and Electrical Drives

In this chapter we will see how an induction motor works and a brief overview about the most common techniques used for its control.

1.1 Induction Motor

The IM (induction motor), or asynchronous motor, is a rotating machine using alternative current, whose speed and network frequency, at which is connected, is not in a constant ratio. It's normally realized collocating, in an isotropic stator, a three-phase induction winding and, in a isotropic rotor, an induced winding (more common is to collocate one or two aluminum cages, linked by a ring at the terminals).

The rotor and the stator are detached by an air gap. The stator winding is made of 3 equal phase in coils and distribution, but equally spaced along the air gap, in order to have a phase shift of 120 degrees relative to one another.

The stator winding is energized and through the interaction between the produced rotating magnetic field and the rotor winding, the power is transferred from the electrical side to the mechanical one.

This elements are described in the Fig. 1.1

The induction motors have many advantages, they are rugged, reliable and economical, and its mainly applications are in the industry field for systems at variable speed service. They don't require period maintenance and they can be plugged directly into the grid.

No sliding contacts and permanent magnets are needed to make an induction motor work, which makes it very simple and cheap to manufacture. However, it's not easy like in DC motors, to control its velocity and torque. For its non-linear behavior during saturation effect and the oscillation of the electrical parameters influenced by the work temperature, is still difficult and complex to control its mathematical model [1].

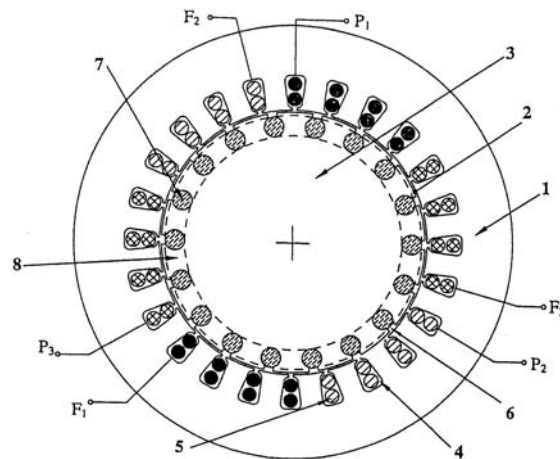


Figure 1.1 – Transversal section of an asynchronous machine three phases with squirrel cage. [1: stator; 2: air gap; 3: rotor; 4: stator caves (4 per pole per phase); 5: stator windings; 6: rotor caves; 7: longitudinal conductive rotoric bars of the cage; 8: ring of the cage; P_1, P_2, P_3 : input terminals for the three statoric phases; F_1, F_2, F_3 : output terminals for the three statoric phases] [1, p.194]

1.2 Electrical drive control

For many years the easiest way to control an induction machine was to regulate the frequency and voltage supply using a scalar control, reaching good performances just in steady-state conditions. This was called V/f ratio constant control. In the first '70s, with the introduction of the spatial vectors, new control techniques have been developed, like FOC (*Field-Oriented Control*) as a vector control with current-loops.

Therefore, the use of induction machines in industrial applications is becoming more practical, thanks to both improved field-oriented control techniques and improvements in the control strategies, power semiconductors, and digital signal processors. The field-oriented control method takes into consideration both successive steady-states and real mathematical equations that describe the motor itself. It reaches better dynamic performance for torque variations in a wide speed range than V/f control, but it requires more computational power, that can be overcome using DSP.

The drawback of FOC is that the rotor speed of the motor must be measured through a speed sensor, *e.g.* an incremental encoder, in order to know the rotor position and be able to calculate other variables, as the rotor flux, otherwise not calculable. Due to the cost of these sensors recent trend is towards the use of sensorless algorithms in FOC. The term sensorless refers to the absence of a speed sensor on the motor shaft, but the motor currents and the voltages must still be measured.

This work is mainly focused on imposing a rotor flux vector in order to decoupling the d and q axes and to control the behavior of the IM in terms of torque or speed. The typical structure of the FOC is reported in Fig. 1.2 [2].

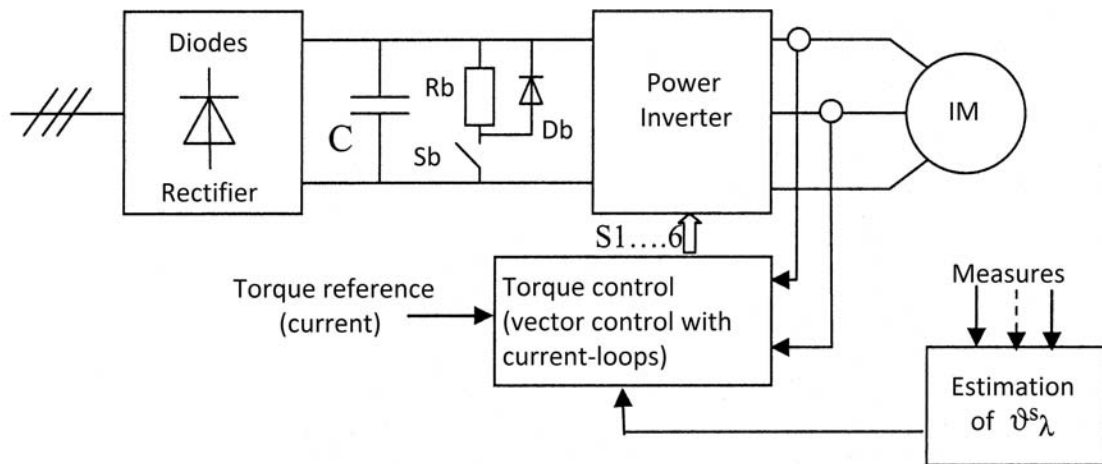


Figure 1.2 – Principle Scheme of an electrical drive for induction motors. [2]

From the three phases electrical grid, a rectifier is used to reach a continue voltage, which supplies a Voltage Source Inverter, which finally allows to have the right voltages or currents shapes for the induction motor supply. The Inverter is controlled by the torque control of the drive.

In the *Field Oriented Control (or vector control)* we will see that using the IM equations we can obtain a decoupling between the electromagnetic torque and the rotor flux, simplifying its dynamic control and bringing back the model to similar DC motor behavior. To reach this simplification, the drive has to work in a reference frame coupled with spatial vector of the rotor flux. This means that in some way the rotor flux has to be measured, estimated or imposed. Using an incremental encoder for the shaft's speed and meters for the currents of the stator phases, the position of the rotor flux can be imposed by a motor model structure. The reference torque usually comes from an external speed-loop which elaborates a speed error between one speed reference and the measured speed on the shaft or on the load. Between the rectifier and the inverter there are always a capacitors bank and an electrical braking system (*chopper*), composed by a switch, a diode and a resistor (the last one is present just in case the load could become a generator).

2 Induction Motor Modeling

In this chapter it will see how an induction motor can be modeled in a stationary frame, using Simulink, and how we can validate as a good model, having the same behavior of the real motor. At first we will see some differential equation that describe the dynamic behavior of IMs, electrical as mechanical ones. After that we will reach the equivalent scheme of one phase, defining the specification of the real machine. At the end there will be a comparison between the Simulink model and the datasheet value of the IM.

2.1 Dynamic behaviour and space vectors

To analyse the dynamic behaviour of the IM we can start from the stator differential equations of the three phases a , b , c :

$$u_{sa} = R_s i_{sa} + \frac{d\Psi_{sa}}{dt} \quad (2.1a)$$

$$u_{sb} = R_s i_{sb} + \frac{d\Psi_{sb}}{dt} \quad (2.1b)$$

$$u_{sc} = R_s i_{sc} + \frac{d\Psi_{sc}}{dt} \quad (2.1c)$$

and for the rotor, since we have a squirrel cage, the three phases are always short circuited, through the rotor equivalent resistors:

$$0 = R_r i_{ra} + \frac{d\Psi_{ra}}{dt} \quad (2.2a)$$

$$0 = R_r i_{rb} + \frac{d\Psi_{rb}}{dt} \quad (2.2b)$$

$$0 = R_r i_{rc} + \frac{d\Psi_{rc}}{dt} \quad (2.2c)$$

Chapter 2. Induction Motor Modeling

With R_s and R_r respectively the statoric and rotoric resistances and Ψ the fluxes. Since every flux is given by all the six currents, we can write assuming a linear magnetic circuit, e.g. with Ψ_{sa} :

$$\Psi_{sa} = \Psi_{stat} + \Psi_{rot} \quad (2.3)$$

with

$$\Psi_{stat} = L_{sa}i_{sa} + L_{Ms}i_{sb} + L_{Ms}i_{sc} \quad (2.4)$$

$$\Psi_{rot} = L_{Mr} \cos(\theta_{me})i_{ra} + L_{Mr} \cos(\theta_{me} + \frac{2\pi}{3})i_{rb} + L_{Mr} \cos(\theta_{me} + \frac{4\pi}{3})i_{rc} \quad (2.5)$$

Defining θ_{me} the rotor angular electrical position expressed in radians $\theta_{me} = p\theta_m$, with p the number of pair of poles and θ_m the rotor angular position expressed in mechanical radians. Since now we will use space vectors to represent a three phase system, mentioning here a generic one:

$$\mathbf{g} = \frac{2}{3} \left[g_a + g_b e^{j\frac{2}{3}} + g_c e^{j\frac{4}{3}} \right] \quad (2.6)$$

So the (2.1) and (2.2) become two complex equation and they are still dependent on time:

$$\mathbf{u}_s^s = R_s \mathbf{i}_s^s + \frac{d\Psi_s^s}{dt} \quad (2.7)$$

$$\mathbf{0} = R_r \mathbf{i}_r^r + \frac{d\Psi_r^r}{dt} \quad (2.8)$$

The subscripts indicate the reference frame of the space vectors and the bold variables represent complex space vectors. All these vectors are referred to stator and rotor fixed reference. In the same way, the (2.4) become, for the three phases, guessing $L_{sa} = L_{sb} = L_{sc} = L_{s\sigma}$, the next equation:

$$\Psi_{stat}^s = L_s \mathbf{i}_s^s \quad \text{with} \quad L_s = L_{s\sigma} + |L_{Ms}| \quad (2.9)$$

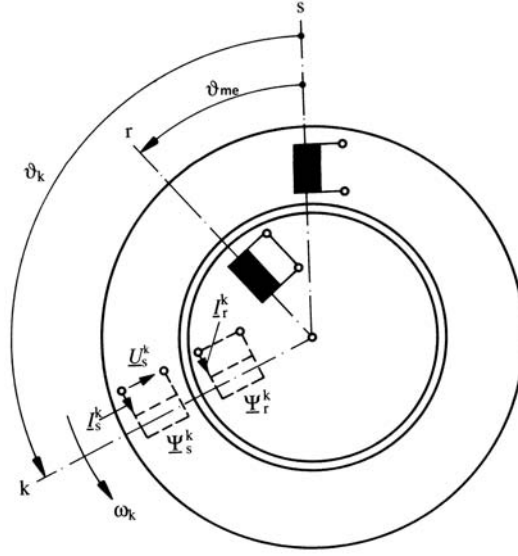


Figure 2.1 – Common reference system [3]

and the (2.5):

$$\Psi_{\text{rot}}^s = \frac{3}{2} L_{Mr} \mathbf{i}_r^r e^{j\theta_{me}} = L_M \mathbf{i}_r^s \quad \text{with } L_M = \frac{3}{2} L_{Mr} \text{ and } \mathbf{i}_r^s = \mathbf{i}_r^r e^{j\theta_{me}} \quad (2.10)$$

So the total stator flux is given by the transformation of (2.3):

$$\Psi_s^s = L_s \mathbf{i}_s^s + L_M \mathbf{i}_r^s \quad (2.11)$$

And in the same way we have obtained the stator flux, the rotor flux depending from the currents is given by:

$$\Psi_r^r = L_r \mathbf{i}_r^r + L_M \mathbf{i}_s^s \quad (2.12)$$

With the same method of *synchronous stator inductance* L_s , we have defined the *synchronous rotor inductance* L_r

2.2 Stator and rotor equations in $d^k q^k$ reference frame

Introducing a generic reference frame k , not fixed but generally rotating in respect to the fixed one, we can ease the mathematical model of the IM. The common reference system is shown in Fig. 2.1.

Multiplying both sides of both statoric equations ((2.7) and (2.11)) by $e^{-j\theta_k}$ we obtain:

$$\mathbf{u}_s^k = R_s \mathbf{i}_s^k + \frac{d\Psi_s^k}{dt} + j\omega_k \Psi_s^k \quad (2.13)$$

$$\Psi_s^k = L_s \mathbf{i}_s^k + L_M \mathbf{i}_r^k \quad (2.14)$$

At the same way we can obtain the rotoric equation in the common reference frame multiplying the (2.8) and (2.12) before by $e^{j\theta_{me}}$, to express it respect to the statoric frame, and later by $e^{-j\theta_k}$:

$$\mathbf{0} = R_r \mathbf{i}_r^k + \frac{d\Psi_r^k}{dt} + j(\omega_k - \omega_{me})\Psi_r^k \quad (2.15)$$

$$\Psi_r^k = L_r \mathbf{i}_r^k + L_M \mathbf{i}_s^k \quad (2.16)$$

2.3 Stator and rotor equations in $d^s q^s$ stationary reference frame

One of the first option is to choose the common reference system in order to keep all the equation in accordance with the stator. We will use this reference to implement the IM equations in *Simulink*.

Imposing $\theta_k = 0$, the k axis overlap the stator axis. Consequently we force that:

$$\frac{d\theta_k}{dt} = \omega_k = 0 \quad (2.17)$$

So (2.13) and (2.15) become:

$$\mathbf{u}_s^s = R_s \mathbf{i}_s^s + \frac{d\Psi_s^s}{dt} \quad (2.18)$$

$$\mathbf{0} = R_r \mathbf{i}_r^s + \frac{d\Psi_r^s}{dt} - j\omega_{me}\Psi_r^s \quad (2.19)$$

And (2.14) and (2.16):

$$\Psi_s^s = L_s \mathbf{i}_s^s + L_M \mathbf{i}_r^s \quad (2.20)$$

$$\Psi_{\mathbf{r}}^{\mathbf{s}} = L_r \mathbf{i}_{\mathbf{r}}^{\mathbf{s}} + L_M \mathbf{i}_{\mathbf{s}}^{\mathbf{s}} \quad (2.21)$$

Writing the last two equations expliciting the currents we obtain:

$$\mathbf{i}_{\mathbf{s}}^{\mathbf{s}} = -\frac{L_r}{L_M^2 - L_s L_r} \Psi_{\mathbf{s}}^{\mathbf{s}} + \frac{L_M}{L_M^2 - L_s L_r} \Psi_{\mathbf{r}}^{\mathbf{s}} \quad (2.22)$$

$$\mathbf{i}_{\mathbf{r}}^{\mathbf{s}} = \frac{L_M}{L_M^2 - L_s L_r} \Psi_{\mathbf{s}}^{\mathbf{s}} - \frac{L_s}{L_M^2 - L_s L_r} \Psi_{\mathbf{r}}^{\mathbf{s}} \quad (2.23)$$

2.4 Power and electromagnetic torque

Remember that (2.18) and (2.19) are complex so they are totally 4 equations. Multiplying (2.18) by i_{sd} and i_{sq} , and (2.19) by i_{rd} and i_{rq} , adding all the 4 equations we obtain the total power:

$$P_e = P_J + P_\mu + P_m \quad (2.24)$$

with P_e - input active power, P_J - joule losses, P_μ - stored magnetic power and P_m mechanical power trasferred (iron and mechanical losses are neglected). We remember that we have to multiplying by $\frac{3}{2}$ when we pass from the synchronous to the stationary reference frame. In this model the iron and mechanical losses are neglected.

$$P_e = \frac{3}{2} (u_{sd} i_{sd} + u_{sq} i_{sq}) \quad (2.25a)$$

$$P_J = \frac{3}{2} [R_s (i_{sd}^2 + i_{sq}^2) + R_r (i_{rd}^2 + i_{rq}^2)] \quad (2.25b)$$

$$P_\mu = \frac{3}{2} \left(i_{sd} \frac{d\Psi_{sd}}{dt} + i_{sq} \frac{d\Psi_{sq}}{dt} + i_{rd} \frac{d\Psi_{rd}}{dt} + i_{rq} \frac{d\Psi_{rq}}{dt} \right) \quad (2.25c)$$

$$P_m = \frac{3}{2} \omega_{me} (\Psi_{rq} i_{rd} - \Psi_{rd} i_{rq}) = \omega_m T_e \quad (2.25d)$$

with T_e the electromagnetic torque.

Input reactive power:

$$Q_e = \frac{3}{2} (u_{sq} i_{sd} - u_{sd} i_{sq}) \quad (2.26)$$

And input apparent power:

$$S = P_e + jQ_e \quad (2.27)$$

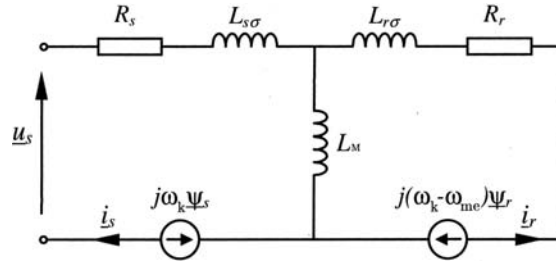


Figure 2.2 – Space vector equivalent circuit in an arbitrary reference frame

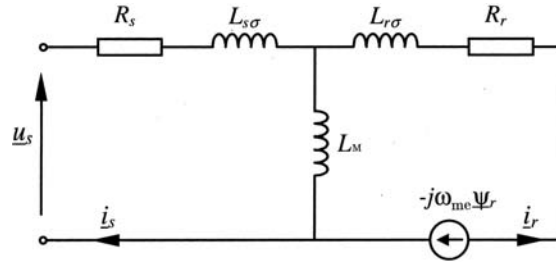


Figure 2.3 – Space vector equivalent circuit in a stationary reference frame ($\omega_k = 0$)

So we can isolate the T_e :

$$\begin{aligned} T_e &= \frac{3}{2} p (\Psi_{rq} i_{rd} - \Psi_{rd} i_{rq}) = \frac{3}{2} p \text{Im}(\Psi_r \mathbf{i}_r^*) = -\frac{3}{2} p \frac{L_M}{L_r} \text{Im}(\Psi_r \mathbf{i}_s^*) \\ &= -\frac{3}{2} p \text{Im}(\Psi_s \mathbf{i}_s^*) = \frac{3}{2} p (\Psi_{sd} i_{sq} - \Psi_{sq} i_{sd}) \end{aligned} \quad (2.28)$$

2.5 Equivalent electric circuit

Every equation in the previous sections can be represent with an equivalent electric circuit. We remember that this electrical schemes report the spatial vectors and characterize the dynamic

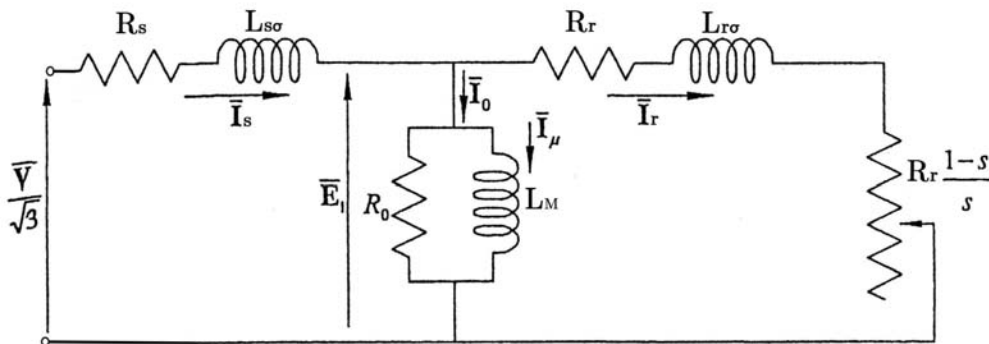


Figure 2.4 – Sinusoidal steady state monophasic electrical circuit [1]

Table 2.1 – Nominal Values of the IM (*ABB MT90L24F165-4 MK119961-S IP55 IEC34 CI F*) [5]

	nominal values
Voltage U_N	400 V
Current I_N	3.5 A
rpm n	1420 rpm
Number of pair of poles p	2
Power P_N	1500 W
Power factor $\cos(\varphi_N)$	0.79
Torque T_N	10 Nm
Inertia moment J	0.0043 Kg m^2
Weight m	16 Kg
Efficiency η	80.3 %

behavior of the model.

In an arbitrary reference frame we have Fig. 2.2.

In the stationary reference frame we obtain Fig. 2.3.

To define the parameters in our real motor, we will use the sinusoidal steady state monophasic electrical circuit, which let us to represent the vectorial generator in Figs. 2.2 and 2.3 as a variable resistance function of the slip s , reported in Fig. 2.4.

The slip is defined as:

$$s = \left(s = \frac{\omega_n - \omega_{me}}{\omega_s} \right) \quad (2.29)$$

We can notice that for $s = 0$ we will not have rotor currents and the power is not transferred; this condition is equivalent to the *No Load* operation (this condition can only be reached if mechanical losses are zero).

With $s = 1$ (locked rotor) the circuit represents the *Short Circuit* operation of the motor.

If the slip is between 0 and 1, the electrical power represented by:

$$\bar{P}_m = 3R_r \frac{(1-s)}{s} \bar{I}_r^2 \quad (2.30)$$

is the mechanical useful power to the motor shaft plus the mechanical losses.

2.6 Specification of the motor

The parameters of the circuit shown in Fig. 2.4 can be defined doing the following tests on the machine:

- *No Load* Test;
- *Short Circuit* Test;

Chapter 2. Induction Motor Modeling

Table 2.2 – Measured values of *No Load* and *Short Circuit* tests

<i>No Load</i>	<i>Short Circuit</i>	Multimeter
$U_0 = 400.3 \text{ V}$	$U_{SC} = 83.4 \text{ V}$	$R_{\text{ohm}} = 9.3\Omega$
$I_0 = 1.88 \text{ A}$	$I_{SC} = 3.48 \text{ A}$	$R_c = 0.12\Omega$
$P_0 = 250 \text{ W}$	$P_{SC} = 360 \text{ W}$	
$Q_0 = 1260 \text{ VAR}$	$Q_{SC} = 340 \text{ VAR}$	
$S_0 = 1290 \text{ VA}$	$S_{SC} = 500 \text{ VA}$	
$\cos(\varphi_0) = 0.79$	$\cos(\varphi_{SC}) = 0.72$	

- Measure resistance R_s of one phase with a multimeter.

The nominal values of the ABB motor are reported in the Table 2.1. For the first two tests, a power quality analyzer helped us to calculate the power factor and to measure the power, the voltage and the currents. The *No Load* test is carried out without applying any load to the motor shaft at the nominal voltage and frequency value (50 Hz).

For the *Short Circuit* test we applied the nominal current when the rotor was locked.

Both derived measurements are reported in the Table 2.2, where we also reported the multimeter measurement for the statoric resistance (R_c is the estimated cable resistance).

So step by step we can derive every electrical parameters of the IM:

$$R_s = \frac{R_{\text{ohm}} - R_c}{2} \simeq 4.6\Omega \quad (2.31)$$

$$R_0 = \left(\frac{U_0}{\sqrt{3}} - R_s I_0 \right)^2 \frac{1}{\left(\frac{P_0}{3} - R_s I_0^2 \right)} = 738\Omega \quad (2.32)$$

$$R_r = \frac{\frac{P_{SC}}{3} - R_s I_{SC}^2}{I_{SC}^2} = 5.3\Omega \quad (2.33)$$

$$X_s + X_r = (R_s + R_r) \tan \varphi_{SC} = 9.54\Omega \quad X_s = X_r = 4.77\Omega \quad (2.34)$$

$$X_M = \left(\frac{Q_0}{3} - X_s I_0^2 \right) \frac{1}{(I_0 \sin \varphi_0)^2} = 118.8\Omega \quad (2.35)$$

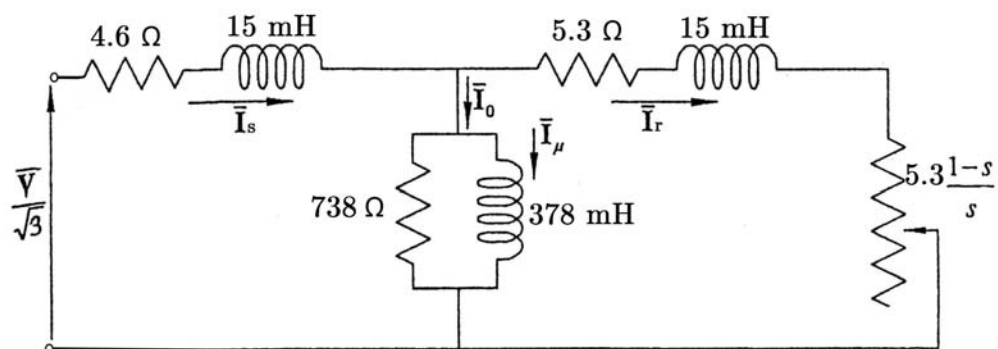


Figure 2.5 – Equivalent monophase electrical circuit of the IM

Obtaining the equivalent monophase electrical circuit of the IM, Fig. 2.5.

2.7 Simulink model and results

The equations 2.18, 2.19, 2.22, 2.23 and 2.28 represented in Fig. 2.7 are not enough to run the model. We have to add the mechanical torque balance:

$$J \frac{d\omega_{me}}{dt} = T_e - T_L \quad (2.36)$$

with T_L the external load torque. The structural diagram is shown in Fig. 2.6.

We will show now how this model can be validated, representing the IM on the real plant, comparing the Simulink model behavior with the datasheet values in Tab. 2.1. We will use an open-loop system simulation. First, we start the simulation applying a zero external torque, to simulate the *No Load* operation, and after 0.2 seconds we will apply the nominal torque $T_L = 10 Nm$. In this case we used an ideal three-phase electrical grid as a power supply, with 230V phase to neutral at the nominal frequency of 50Hz.

Neglecting the first part of the Fig. 2.8 we can notice that:

- The currents increase with the torque: nominal torque, nominal value ($3.5\sqrt{2} \simeq 5A$);
- The rotor passes from the *No Load* speed $\left(\frac{2\pi f_{60}}{p}\right)$ to the nominal one;
- The T_e follows the reference torque (Load) in a converget way.

In the same figure are also reported the stator and rotor fluxes, as the electrical and mechanical power derived from 2.25a and 2.25d. We can notice that in steady state condition, at the nominal torque, the electrical power is equal to 1500W, and the mechanical one is lightly lower, due to the stator and rotor losses (the iron and mechanical losses have been neglected). The model just seen, reflects the characteristics of the ideal motor and its datasheet. It should therefore be considered as good and defined validated.

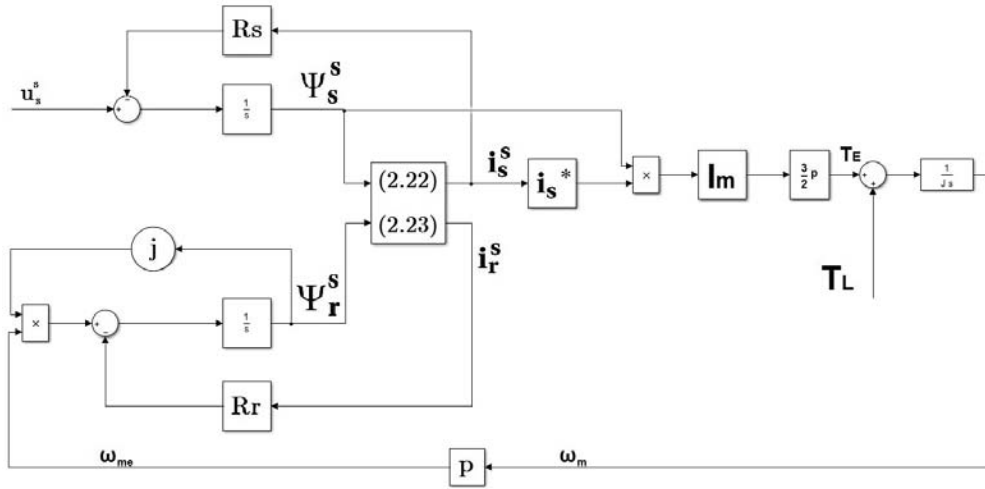


Figure 2.6 – Structural diagram of the IM in terms of spacial vectors

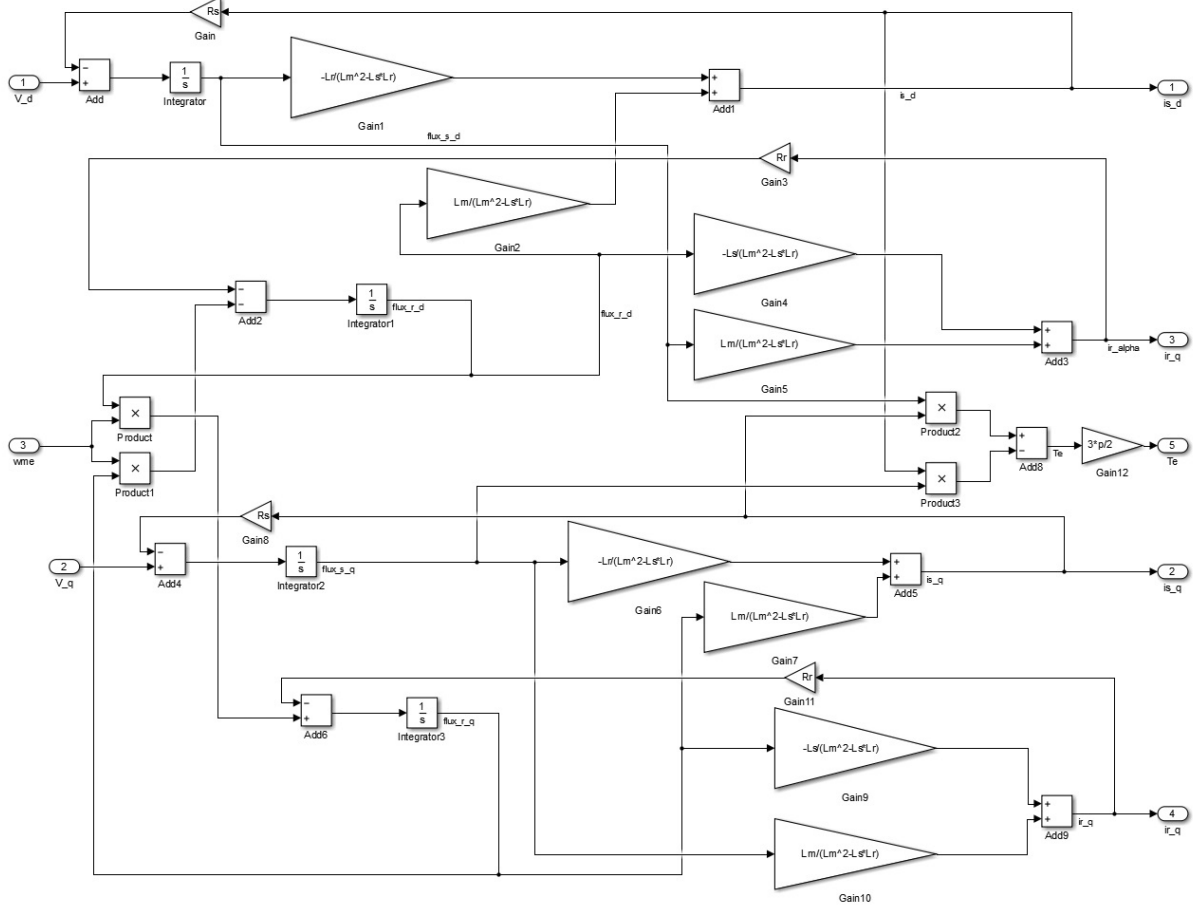


Figure 2.7 – Simulink blocks represent the differential equation of the IM

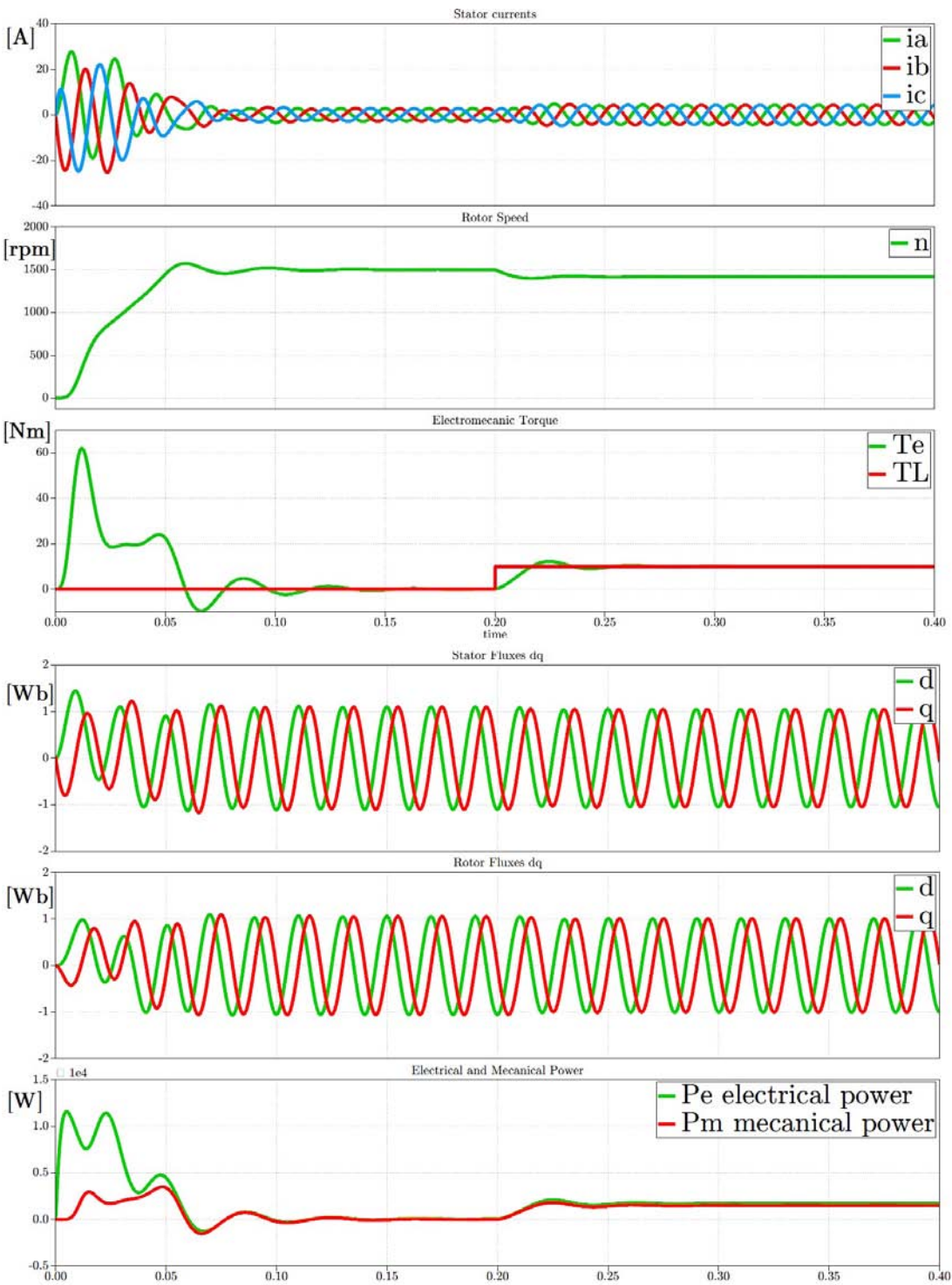


Figure 2.8 – Simulation results

2.8 Verification of the Steady-State condition

2.8.1 No Load

As we can see from the Fig. 2.8 the rotor speed at *No Load* condition is approximately 1500 rpm, so as should be, the slip s is zero. Considering the circuit of Fig. 2.5 we can also calculate \tilde{I}_0 with $s = 0$:

$$\tilde{I}_0 = 1.87 \angle -1.38 A \quad (2.37)$$

The results in the simulation reflects this numbers, as we can see in Fig.2.9 (Note that $1.87\sqrt{2} \simeq 2.7 A$ and $\Delta t = 0.0044 s$ is approximately 1.38 radians).

2.8.2 Rated Load

As we can see from the Fig. 2.8 the rotor speed at *Rated Load* condition is approximately 1420 rpm, so as should be, the slip $s = \frac{4}{75}$. Considering the circuit of Fig. 2.5 we can also calculate \tilde{I}_1 with $s = \frac{4}{75}$:

$$\tilde{I}_1 = 3.038 \angle -0.6746 A \quad (2.38)$$

The results in the simulation reflects this numbers, as we can see in Fig.2.10 (Note that $3.038\sqrt{2} \simeq 4.3 A$ and $\Delta t = 0.0022 s$ is approximately 0.67 radians).

In both load conditions it has been verified that the equivalent monophas electrical circuit actually represents the IM in sinusoidal steady state.

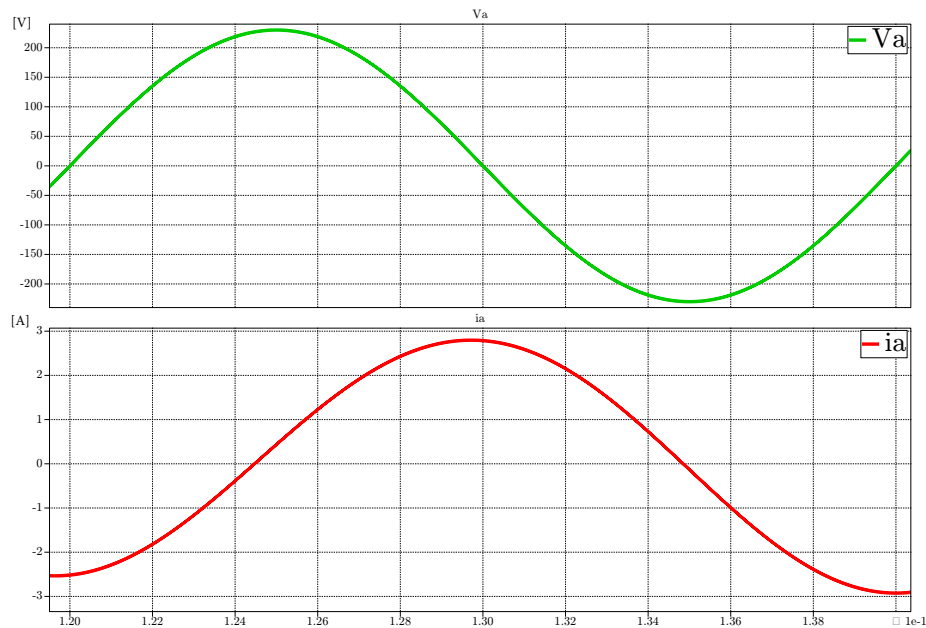


Figure 2.9 – Simulation representing stator voltage and current phase shift of a phase in *No Load* condition

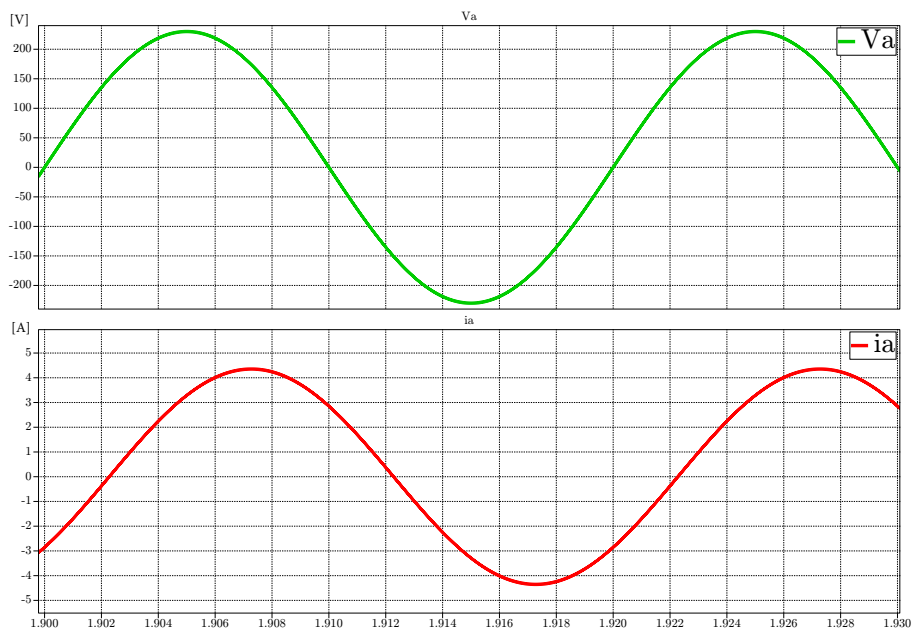


Figure 2.10 – Simulation representing stator voltage and current phase shift of a phase in *Rated Load* condition

3 Induction Motor Control

The IM is one of the most in use motor for variable speed applications. It can reach high performance in terms of speed reference and response (or in terms of torque reference) through the supply of a frequency converter and control circuits, sometimes really complex. This chapter first explain the simplest control techniques and then more and more complicated ones in order to obtain higher performances. The first two strategies impose the stator voltage; the third one impose the stator currents. All these techniques want to maintain the rotor flux close to the nominal value, in order not to exceed the saturation of the motor.

3.1 Imposing the stator flux through the stator voltage

3.1.1 Stator and Rotor equations in $d^r q^r$ rotary reference frame at synchronous speed

Before starting with the control techniques is useful to change the stationary reference frame to a synchronous one, in order to see the stator frequency role. Imposing:

$$\theta_k = \int_0^t \omega_s dt \quad (3.1)$$

$$\frac{d\theta_k}{dt} = \omega_s = f_s \omega_n \quad (3.2)$$

$$\frac{d(\theta_k - \theta)}{dt} = \omega_n(f_s - n) = \omega_n f_r \quad (3.3)$$

with f_s, f_r the relative stator and rotor frequency, and $n = \frac{\omega_{me}}{\omega_n}$. So the 2.13 and 2.15 become:

$$\mathbf{u}_s^r = R_s \mathbf{i}_s^r + \frac{d\Psi_s^r}{dt} + j\omega_n f_s \Psi_s^r \quad (3.4)$$

$$\mathbf{0} = R_r \mathbf{i}_r + \frac{d\Psi_r}{dt} + j\omega_n f_r \Psi_r \quad (3.5)$$

And the 2.14 and 2.16:

$$\Psi_s = L_s \mathbf{i}_s + L_M \mathbf{i}_r \quad (3.6)$$

$$\Psi_r = L_r \mathbf{i}_r + L_M \mathbf{i}_s \quad (3.7)$$

We will now change all the parameters of the motor from absolute values in per unit values so it will be easier to evaluate the dynamic behavior of the motor and the treatment becomes more reading. So introducing $\mathbf{u}_s = \frac{\mathbf{u}_s}{U_n}$, $\mathbf{i}_s = \frac{\mathbf{i}_s}{I_n}$, $\mathbf{i}_r = \frac{\mathbf{i}_r}{I_n}$, $\Psi_s = \frac{\Psi_s \omega_n}{U_n}$, $\Psi_r = \frac{\Psi_r \omega_n}{U_n}$, $r_s = \frac{R_s I_n}{U_n}$, $r_r = \frac{R_r I_n}{U_n}$, $x_s = \frac{L_s I_n \omega_n}{U_n}$, $x_r = \frac{L_r I_n \omega_n}{U_n}$, $x_M = \frac{L_M I_n \omega_n}{U_n}$ we obtain the IM equations in p.u.:

$$\mathbf{u}_s = r_s \mathbf{i}_s + \frac{1}{\omega_n} \frac{d\Psi_s}{dt} + j f_s \Psi_s \quad (3.8)$$

$$\mathbf{0} = r_r \mathbf{i}_r + \frac{1}{\omega_n} \frac{d\Psi_r}{dt} + j f_r \Psi_r \quad (3.9)$$

$$\Psi_s = x_s \mathbf{i}_s + x_M \mathbf{i}_r \quad (3.10)$$

$$\Psi_r = x_r \mathbf{i}_r + x_M \mathbf{i}_s \quad (3.11)$$

And from the last ones, defining $\sigma = 1 - \frac{x_M^2}{x_s x_r}$

$$\mathbf{i}_s = \frac{1}{\sigma x_s} \Psi_s - \frac{1 - \sigma}{\sigma x_M} \Psi_r \quad (3.12)$$

$$\mathbf{i}_r = \frac{1}{\sigma x_r} \Psi_r - \frac{1 - \sigma}{\sigma x_M} \Psi_s \quad (3.13)$$

Introducing the stator and rotor time constants:

$$T_s' = \frac{\sigma x_s}{\omega_n r_s} \quad (3.14)$$

3.1. Imposing the stator flux through the stator voltage

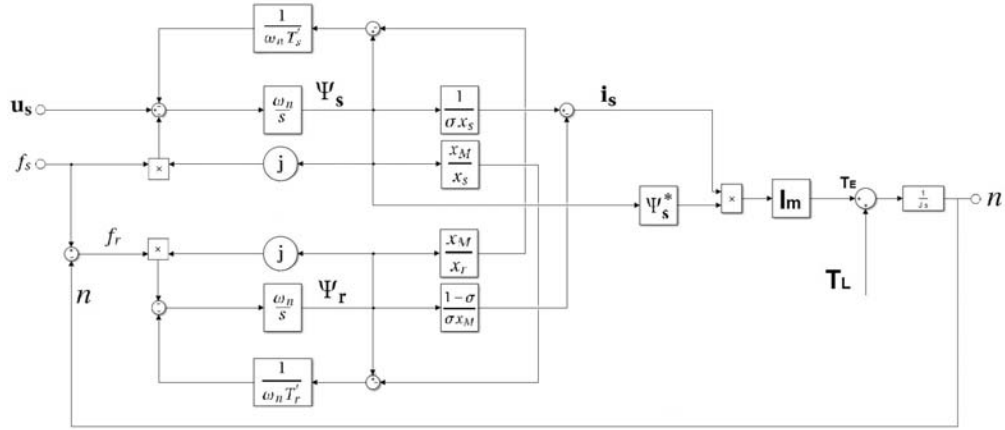


Figure 3.1 – Modified structural diagram of the IM with stator voltage imposition

$$T_r' = \frac{\sigma x_r}{\omega_n r_r} \quad (3.15)$$

and $\frac{1-\sigma}{\sigma x_M} = \frac{x_M}{\sigma x_s x_r}$ finally we obtain the modified structural diagram (Fig. 3.1) and its equations:

$$\frac{d\Psi_s}{dt} = \omega_n \left[\mathbf{u}_s - \left(\frac{1}{\omega_n T_s'} + j f_s \right) \Psi_s + \frac{1}{\omega_n T_s'} \frac{x_M}{x_r} \Psi_r \right] \quad (3.16)$$

$$\frac{d\Psi_r}{dt} = \omega_n \left[- \left(\frac{1}{\omega_n T_r'} + j f_r \right) \Psi_r + \frac{1}{\omega_n T_r'} \frac{x_M}{x_s} \Psi_s \right] \quad (3.17)$$

This structure is really complex, with feedbacks and multiplication blocks making the system strongly non-linear. In addition we have three variable inputs and one output, so the control system is not defined, having more combination giving the same result [4]. It will see how inputs can be controlled to have a defined control system in the next sections.

3.1.2 Imposing the stator flux through the stator voltage modulus

This is one of the simplest strategy, characterized with a constant stator flux in steady state conditions. The control circuit is simple but the performances, specially in transient state, are low. In steady state the fluxes are constant so 3.16 and 3.17 become:

$$\mathbf{u}_s = \frac{1 + j f_s \omega_n T_s'}{\omega_n T_s'} \Psi_s - \frac{1}{\omega_n T_s'} \frac{x_M}{x_r} \Psi_r \quad (3.18)$$

$$\Psi_r = \frac{1}{1 + j f_r \omega_n T_r'} \frac{x_M}{x_s} \Psi_s \quad (3.19)$$

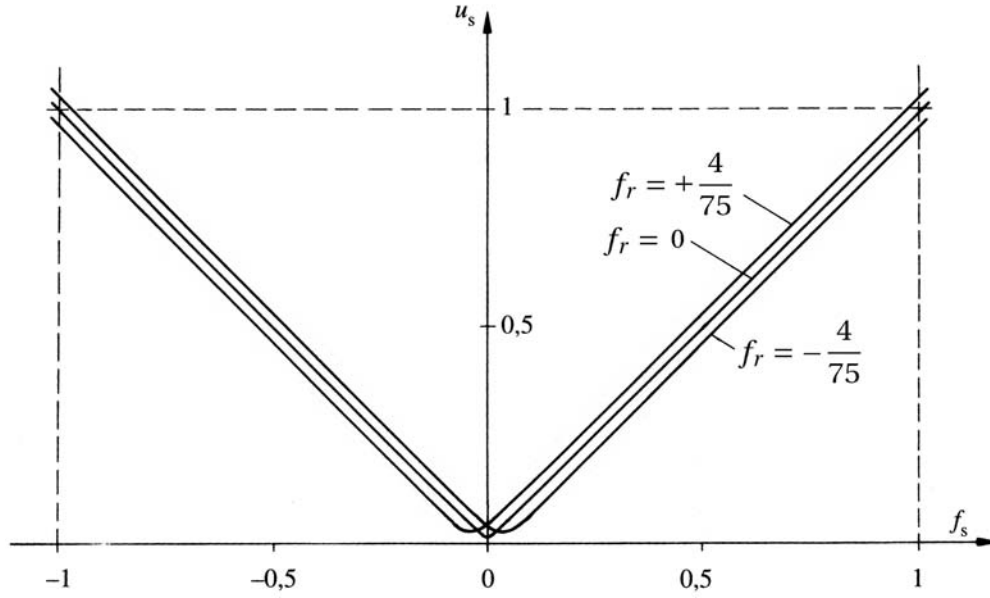


Figure 3.2 – Steady state characteristic for modulus of stator voltage u_s for $\Psi_s = 1$ [4]

In order to eliminate the rotor flux, introducing $T_s = \frac{T'_s}{\sigma}$, $T_r = \frac{T'_r}{\sigma}$ we can obtain the stator voltage modulus:

$$u_s = \frac{\Psi_s}{\omega_n T_s} \sqrt{\frac{(1 - f_s \omega_n T_s f_r \omega_n T'_r)^2 + (f_s \omega_n T_s + f_r \omega_n T_r)^2}{1 + (f_r \omega_n T'_r)^2}} \quad (3.20)$$

For the values in par. 2.6 we can reach the characteristic in steady state of the stator voltage modulus for $\Psi_s = 1$, represented in Fig. 3.2. It is a function of f_s and f_r , where f_s and f_r depending on their sign, represent the 4 operation quadrants of the asynchronous machine.

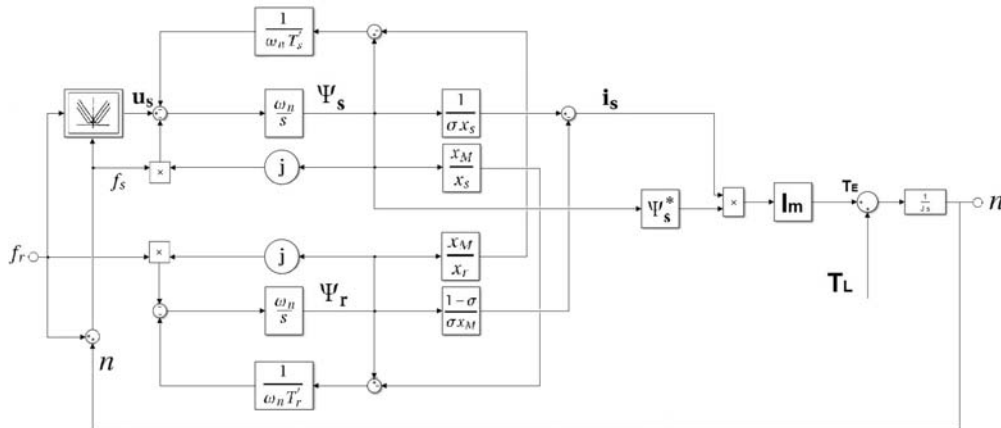


Figure 3.3 – Structural diagram of the IM with stator flux imposition through the modulus stator voltage

3.1. Imposing the stator flux through the stator voltage

The structural diagram is modified in Fig. 3.3 (note that \mathbf{u}_s becomes purely real, in dq rotating reference frame). Using f_r as a nuisance input we can verify the transient phenomena and how the fluxes change.

So we will pass from *No Load* condition to the *Rated Load* condition and backwards, before keeping $n = 1$ constant and then with $n = 0$. At the rated load, $f_{rn} = \pm \frac{4}{75}$, and for $f_r = 0$ represents the no load operation ($i_s = \frac{1}{x_s} = 0.925$).

The Figs. 3.4 and 3.5 show the transient phenomena when the motor is rotating at the no load nominal speed ($n = 1$). We have reached to maintain the stator and rotor fluxes at the nominal value (1 p.u.) during the torque perturbation.

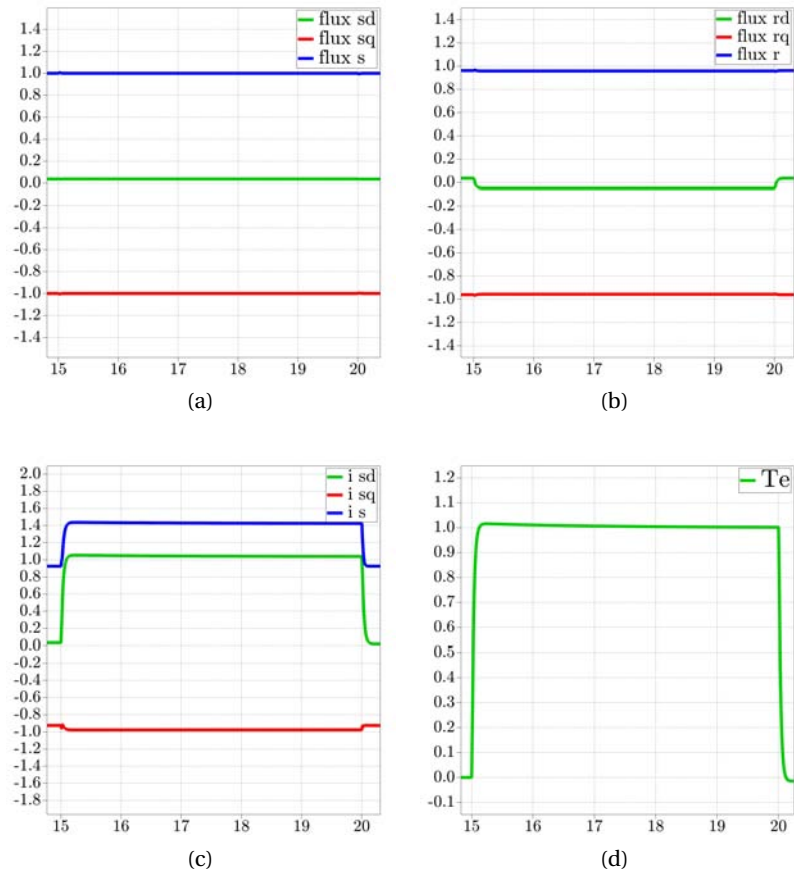


Figure 3.4 – Transient phenomena for a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 1$).

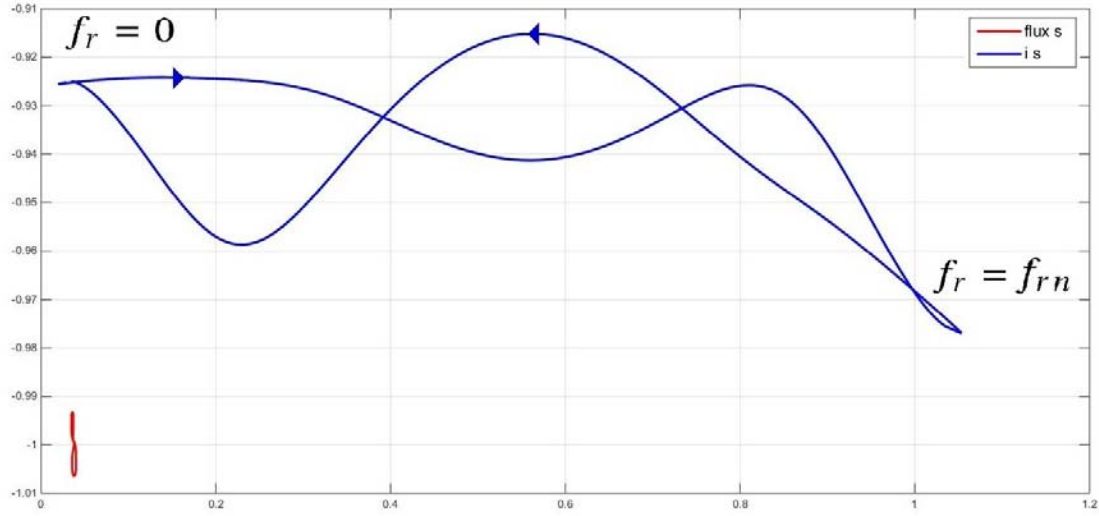


Figure 3.5 – Spatial vector Ψ_s and i_s during a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 1$).

The Figs. 3.6 and 3.7 shows the transient phenomena when the motor shaft is blocked ($n = 0$). In this case the transient phenomena of the fluxes are more clear and evident than before. For $f_r = 0$ the fluxes are purely real as the stator voltage. Both i_s , Ψ_s describe arcs in the transient time from $f_r = 0$ to f_{rn} , and they come back almost as two straight lines.

As it has been showed the transient phenomena depend hardly on the operating point, and this speed regulation is not used when it needs high dynamic performances. However, the control circuit is quite simple and easy to implement (Fig. 3.8). The induction motor is supplied with an inverter (2b), composed by IGBTs, and controlled by a modulator (3) with three voltage references u_{cm1} , u_{cm2} , u_{cm3} generated from the block 4 according to these relations:

$$u_{cm1} = u_{sc} \cos \theta_s \quad (3.21a)$$

$$u_{cm2} = u_{sc} \cos \theta_s - \frac{2\pi}{3} \quad (3.21b)$$

$$u_{cm3} = u_{sc} \cos \theta_s - \frac{4\pi}{3} \quad (3.21c)$$

The block 5 generate the position of the reference frame integrating ω_s derived from a close-loop of n , measured by an encoder (8). The rotor frequency control set (f_{rc}) can be obtained by a regulator PI (R_n), which control the rotational speed of the shaft n on the reference n_c . The block 6 generate the absolute value u_{sc} to apply at the stator voltage according to 3.20.

3.1. Imposing the stator flux through the stator voltage

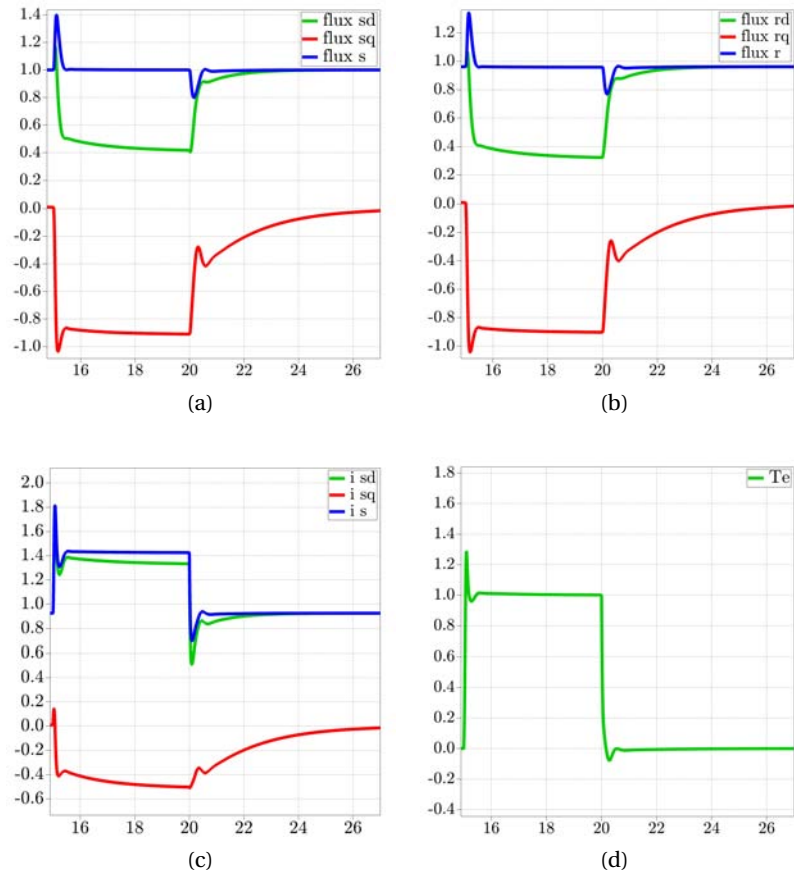


Figure 3.6 – Transient phenomena for a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 0$).

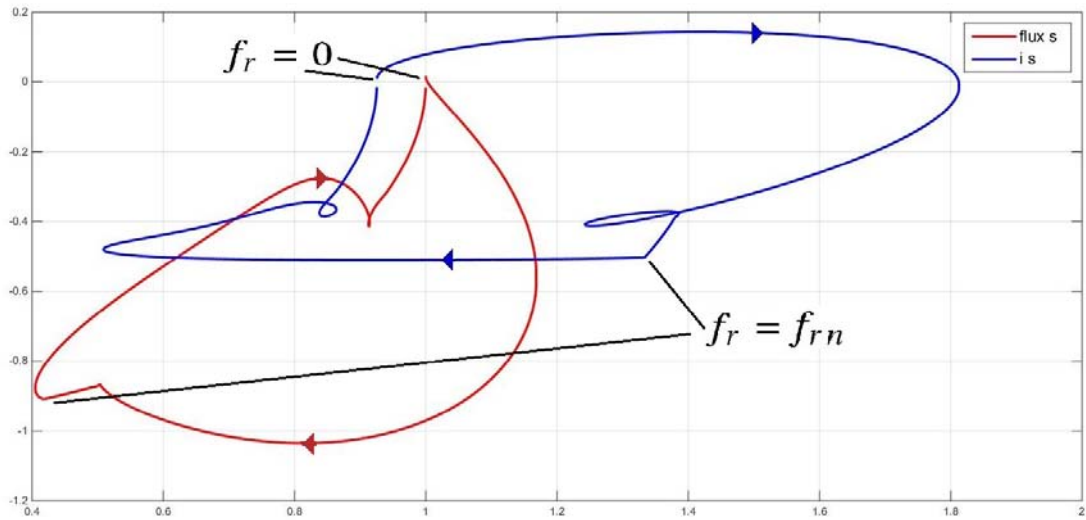


Figure 3.7 – Spatial vector Ψ_s and i_s during a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 0$).

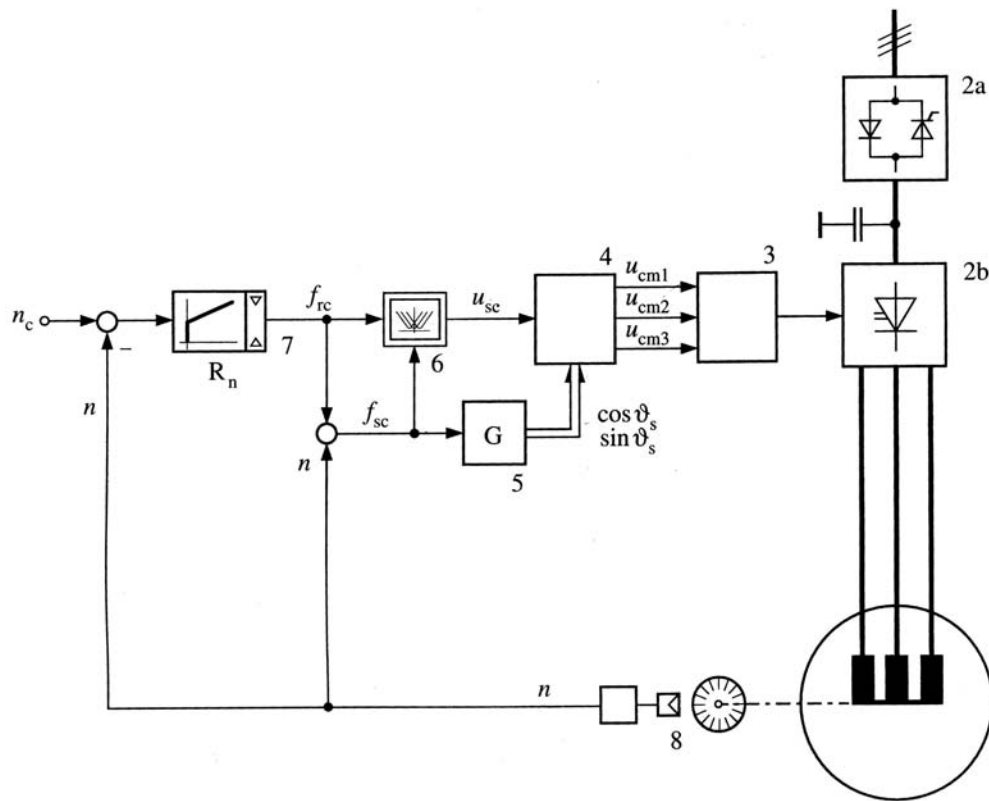


Figure 3.8 – Control circuit scheme to impose the stator flux through the modulus stator voltage [4]

3.1.3 Imposing the stator flux through the stator voltage components

To get an improvement of the dynamic behavior of the previous control, we are going to impose both the components of the stator voltage u_{sd} and u_{sq} . The rotational reference frame will be synchronous with the spacial vector of the stator flux, so $\Psi_s = \Psi_{sd} = \Psi_s$. Using 3.18 and 3.19, at the same way than before, we obtain:

$$u_{sd} = \frac{1 + (f_r \omega_n T_r')^2 / \sigma}{1 + (f_r \omega_n T_r')^2} \frac{\Psi_s}{\omega_n T_s} \quad (3.22)$$

$$u_{sq} = \left[f_s + \frac{f_r (T_r - T_r') / T_s}{1 + (f_r \omega_n T_r')^2} \right] \Psi_s \quad (3.23)$$

The Characteristic in steady state is represented in Fig. 3.9; the u_{sd} componenet is not dependent from f_s , while u_{sq} is proportional to f_s .

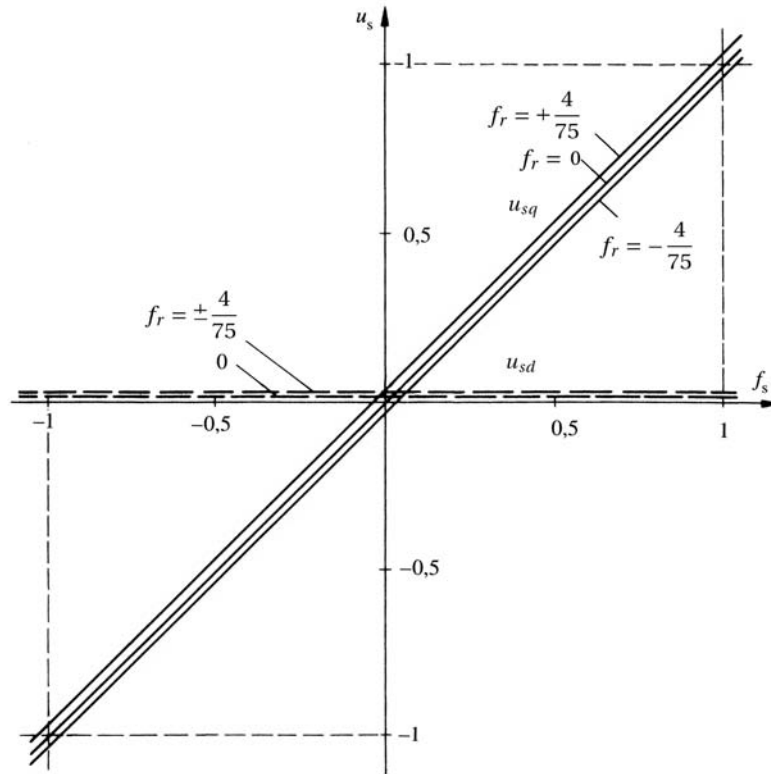


Figure 3.9 – Steady state characteristic for components of stator voltage u_{sd}, u_{sq} for $\Psi_s = 1$ [4]

The structural diagram is modified in Fig. 3.10. Using f_r as a nuisance input we can verify the transient phenomena and how the fluxes change.

So we will pass from *No Load* condition to the *Rated Load* condition and backwards, before keeping $n = 1$ constant and then with $n = 0$. At the rated load, $f_{rn} = \pm \frac{4}{75}$, and for $f_r = 0$

represents the no load operation ($i_s = \frac{1}{x_s} = 0.925$).

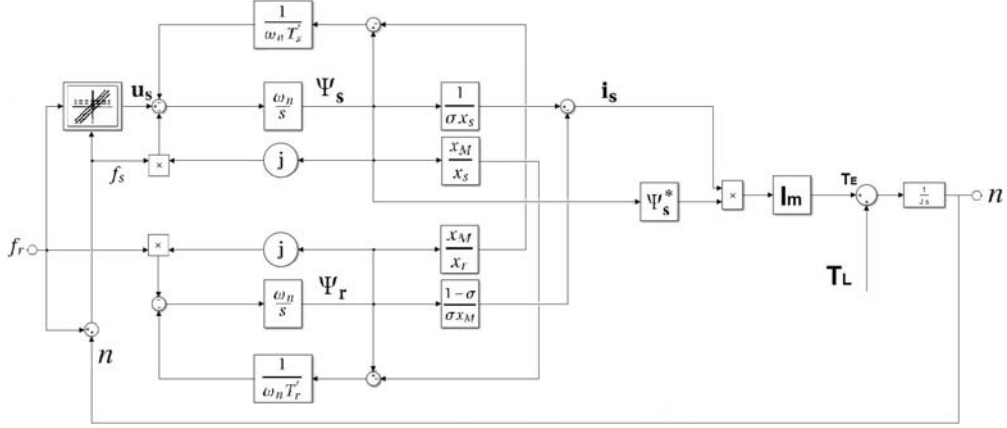


Figure 3.10 – Structural diagram of the IM with stator flux imposition through the components stator voltages

The Figs. 3.11 and 3.12 show the transient phenomena when the motor is rotating at the no load nominal speed ($n = 1$). We have reached to maintain the stator and rotor fluxes at the nominal value (1 p.u.) during the torque perturbation.

The Figs. 3.13 and 3.14 show the transient phenomena when the motor shaft is blocked ($n = 0$). In this case the transient phenomena of the fluxes are more clear and evident than before.

As it has been showed the transient phenomena depend hardly on the operating point, the dynamic behavior is improved than before, but this speed regulation is still not used when it needs high dynamic performances. However, the control circuit is quite simple and easy to implement (Fig. 3.15). As before, the induction motor is supply with an inverter but the three voltage references u_{cm1} , u_{cm2} , u_{cm3} for the modulator are in this case:

$$u_{cm1} = u_{scd} \cos \theta_s - u_{scq} \sin \theta_s \quad (3.24a)$$

$$u_{cm2} = \left(-\frac{1}{2} u_{scd} + \frac{\sqrt{3}}{2} u_{scq} \right) \cos \theta_s + \left(\frac{\sqrt{3}}{2} u_{scd} + \frac{1}{2} u_{scq} \right) \sin \theta_s \quad (3.24b)$$

$$u_{cm3} = \left(-\frac{1}{2} u_{scd} - \frac{\sqrt{3}}{2} u_{scq} \right) \cos \theta_s + \left(-\frac{\sqrt{3}}{2} u_{scd} + \frac{1}{2} u_{scq} \right) \sin \theta_s \quad (3.24c)$$

The block 6 generate the values u_{scd} and u_{scq} to apply at the stator voltage according to 3.22 and 3.23.

3.1. Imposing the stator flux through the stator voltage

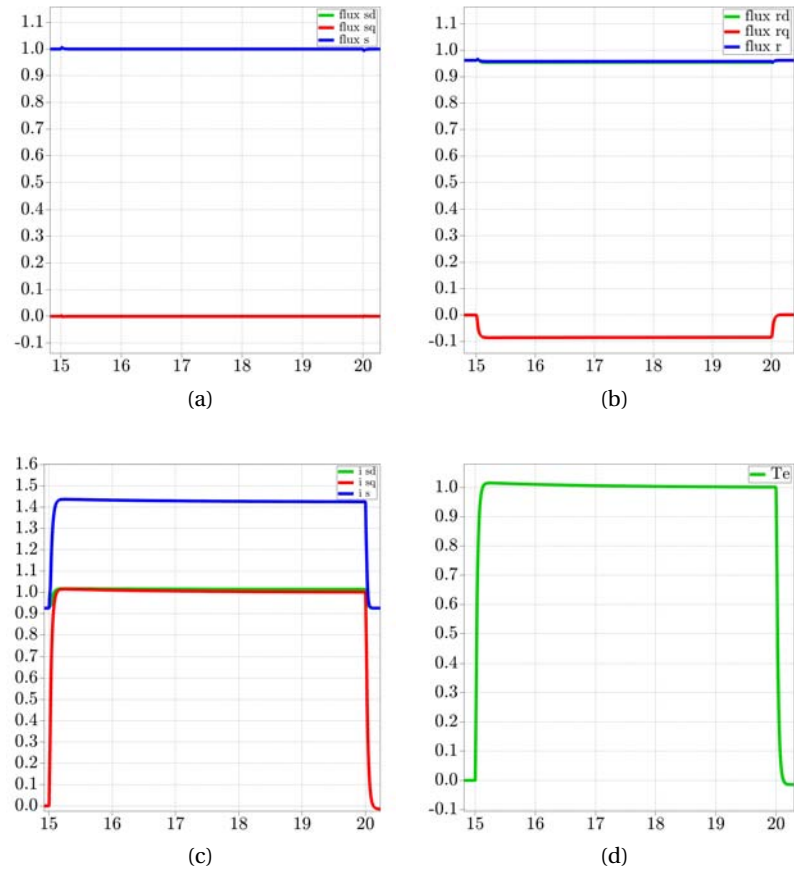


Figure 3.11 – Transient phenomena for a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 1$).

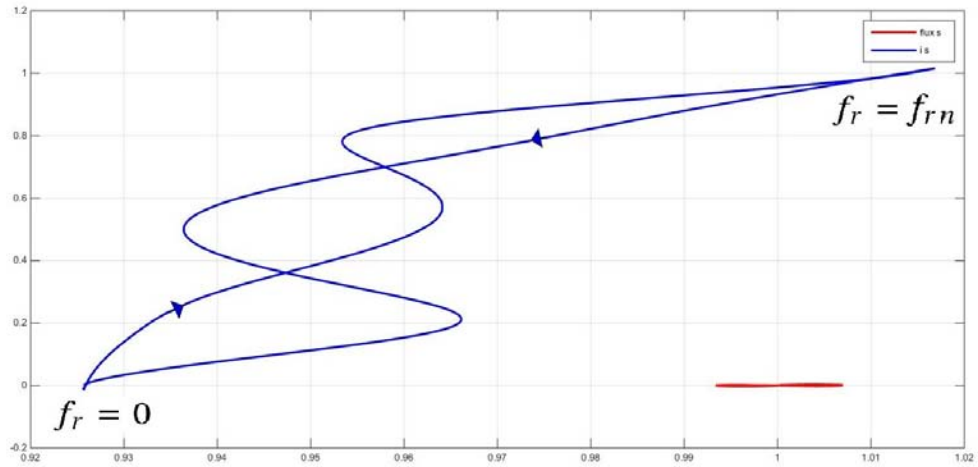


Figure 3.12 – Spatial vector Ψ_s and i_s during a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 1$).

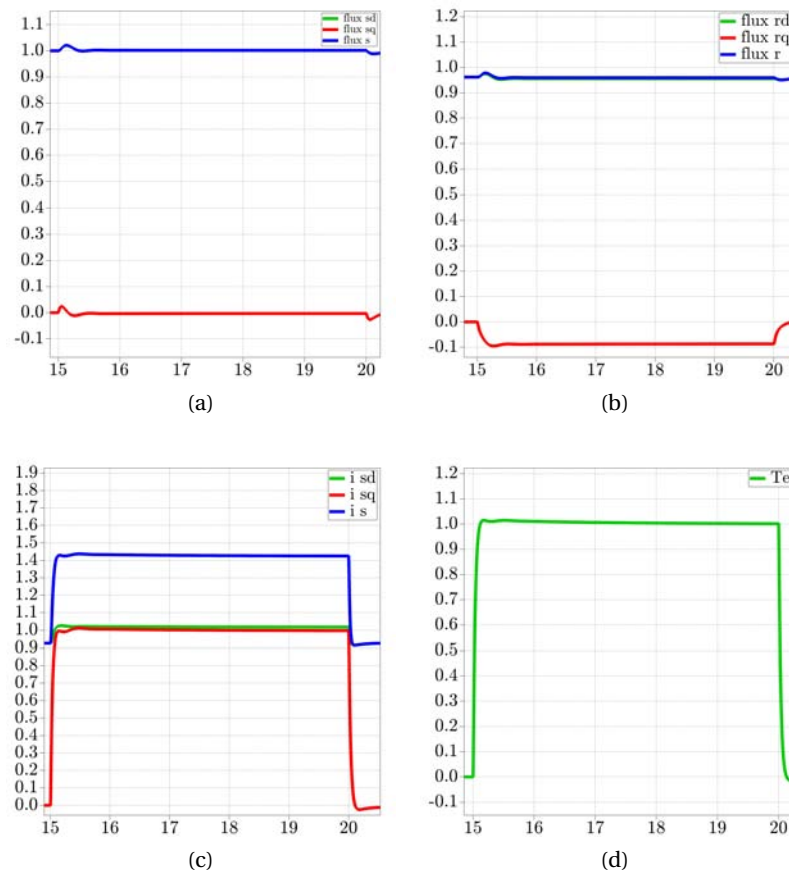


Figure 3.13 – Transient phenomena for a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 0$).

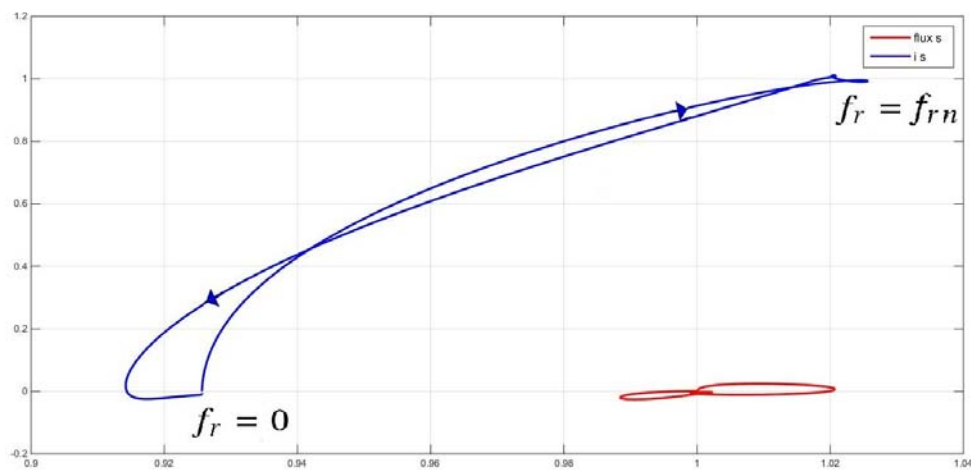


Figure 3.14 – Spatial vector Ψ_s and i_s during a rotor frequency variation from $f_r = 0$ to $f_{rn} = \frac{4}{75}$ and backwards ($n = 0$).

3.1. Imposing the stator flux through the stator voltage

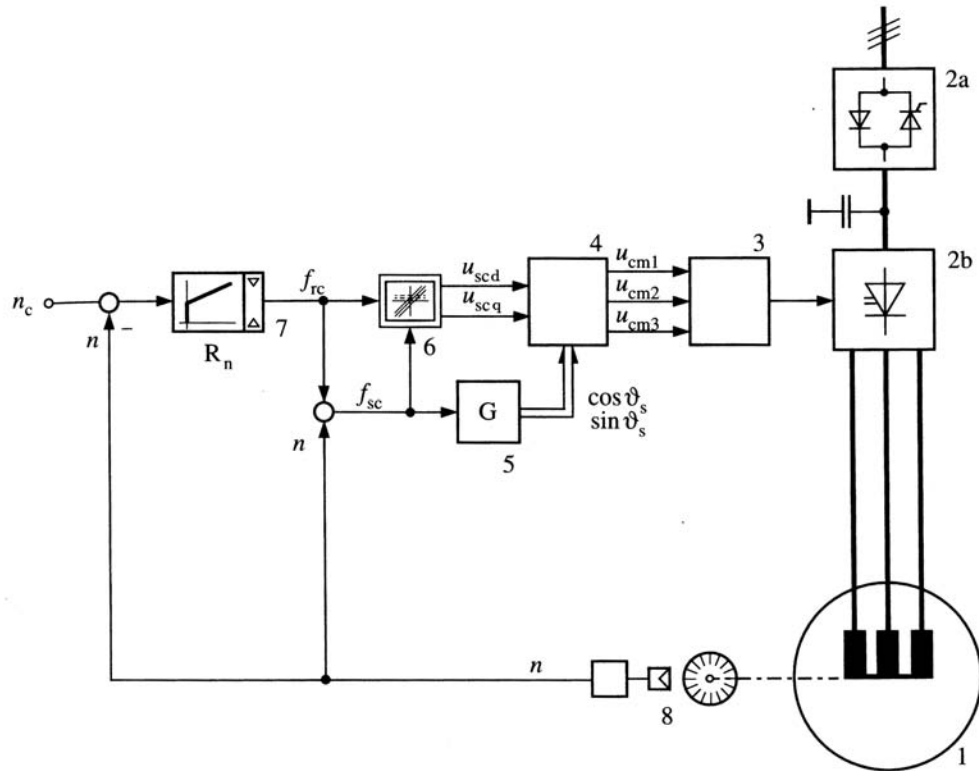


Figure 3.15 – Control circuit scheme to impose the stator flux through the components stator voltage [4]

3.2 Imposing the rotor flux through the components of the stator currents (FOC - *Field Oriented Control*)

Field Oriented control is an high performance technique to control AC motors, decoupling the torque and rotor flux control, as for the DC machine drives. The performance is evaluated with the rising time and overshoots of the control system for a given speed reference step. Vector control for the stator currents is implemented to control the torque (always depending to stator and rotor currents, or currents and fluxes). For this control an synchronous reference frame with the spatial vector of the rotor flux is chosen, in the way that the rotor flux is overlapped to the d axis. In this way $\Psi_{rd} = \Psi_r$ as it's shown in Fig. 3.16. An independent control for the torque and the rotor flux is reached.

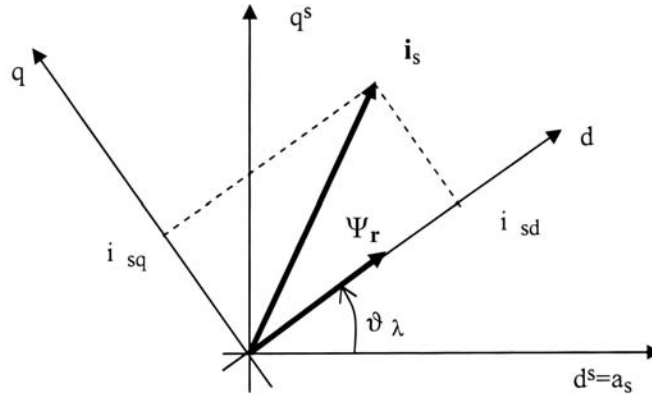


Figure 3.16 – FOC reference frame [2]

From the eq. 3.9 and 3.11 rearranging we can get:

$$\frac{d\Psi_r}{dt} = -\left(\frac{1}{T_r} + j f_r \omega_n\right) \Psi_r + \frac{x_M}{T_r} (i_{sd} + j i_{sq}) \quad (3.25)$$

and decomposed in Real and Imaginary:

$$\frac{d\Psi_r}{dt} = -\frac{\Psi_r}{T_r} + \frac{x_M}{T_r} i_{sd} \quad (3.26)$$

$$0 = -f_r \omega_n \Psi_r + \frac{x_M}{T_r} i_{sq} \quad (3.27)$$

The first equation shows that $\Psi_r = f(i_{sd})$ and the second one that $f_r = f(i_{sq}, \Psi_r)$. Also notice that:

$$T_e = \frac{x_M}{x_r} \text{Im}(\Psi_r^* i_s) = \frac{x_M}{x_r} \Psi_r i_{sq} \quad (3.28)$$

So if the rotor flux is constant, the torque is direct proportional to the i_{sq} , normally happens

3.2. Imposing the rotor flux through the components of the stator currents (FOC - *Field Oriented Control*)

to maintain the stator flux to the nominal one, in case of nominal charge in steady state. The last two equations represent the IM model with currents imposition, and they are represented in Fig. 3.17.

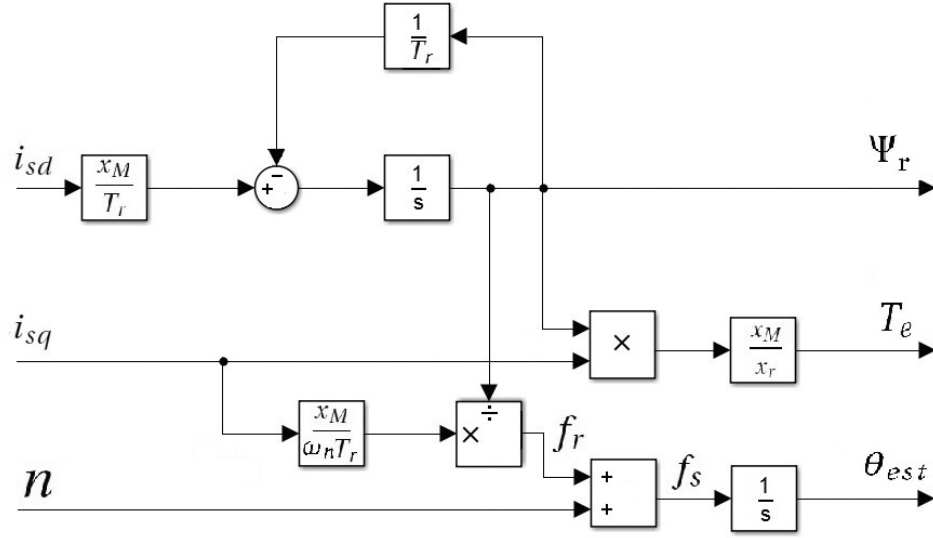


Figure 3.17 – FOC Induction motor model

Instead of rebuilding the space vector of the rotor flux to coincide with the direct axis of the reference frame to be adopted for the vectorial control of the currents, we impose precise conditions of the power supply such that the space vector of the rotor flux is going to be located on the direct axis of the reference system adopted. This is called *Indirect FOC*. Considering the imaginary part of 3.25 and $\Psi_r = \Psi_{rd} + j\Psi_{rq}$, we obtain:

$$\frac{d\Psi_{rq}}{dt} + \frac{\Psi_{rq}}{T_r} = \frac{x_M}{T_r} i_{sq} - f_r \omega_n \Psi_{rd} \quad (3.29)$$

in case the RHS of the equal is zero, the rotor flux becomes purely real, and this happens when:

$$f_r = f_s - n = \frac{x_M}{\omega_n T_r} \frac{i_{sq}}{\Psi_{rd}} \quad (3.30)$$

In the indirect FOC, as we impose where the space vector is going to be located, we can reversely know the stator frequency f_s , so the equation 3.31 will be a part of the motor model:

$$f_s = \frac{x_M}{\omega_n T_r} \frac{i_{sq}}{\Psi_{rd}} + n \quad (3.31)$$

The resulting control circuit scheme for the Indirect FOC is represented in Fig.3.18

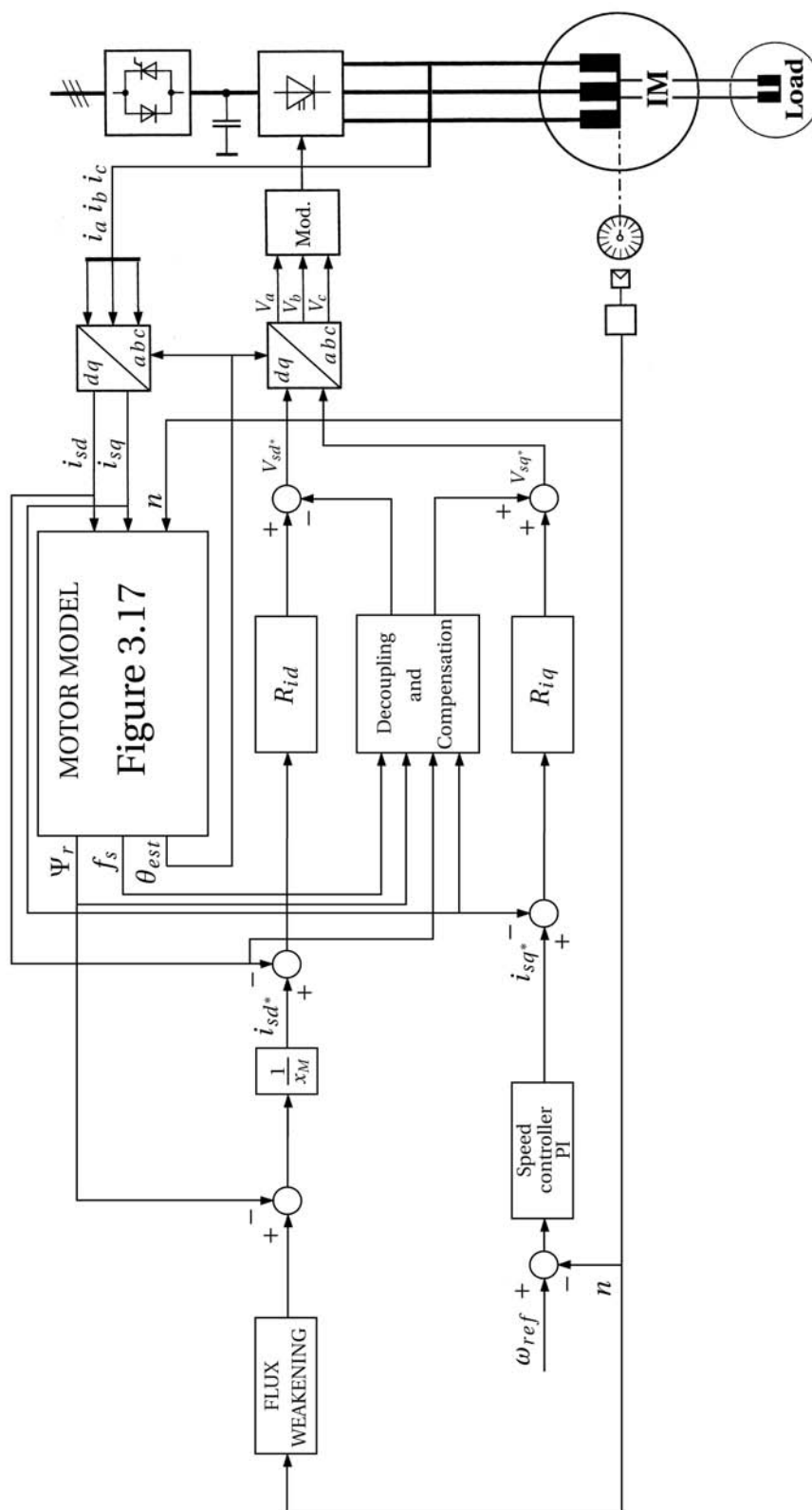


Figure 3.18 – Control circuit scheme FOC

3.2. Imposing the rotor flux through the components of the stator currents (FOC - *Field Oriented Control*)

3.2.1 Current loops

From the eqs. 3.8, 3.10 and 3.11, rearranging we can obtain the stator voltages in function of the stator currents components, the rotor fluxes and the rotational speed of the flux vector (considerign $\Psi_{rd} = \Psi_r$).

$$u_{sd} = r_s i_{sd} + \frac{1}{\omega_n} \frac{x_M}{x_r} \frac{d\Psi_r}{dt} + \frac{\sigma x_s}{\omega_n} \frac{di_{sd}}{dt} - f_s \sigma x_s i_{sq} \quad (3.32)$$

$$u_{sq} = r_s i_{sq} + \frac{\sigma x_s}{\omega_n} \frac{di_{sq}}{dt} + f_s \frac{x_M}{x_r} \Psi_r + f_s \sigma x_s i_{sd} \quad (3.33)$$

So we can define:

$$u'_{sd} = r_s i_{sd} + \frac{\sigma x_s}{\omega_n} \frac{di_{sd}}{dt} + \frac{x_M}{\omega_n x_r} \frac{d\Psi_r}{dt} \quad (3.34)$$

$$u'_{sq} = r_s i_{sq} + \frac{\sigma x_s}{\omega_n} \frac{di_{sq}}{dt} \quad (3.35)$$

which together 3.26 and written in the s domain, express the transfer functions between the currents and voltages of the control.

$$Y_q = \frac{I_{sq}}{U'_{sq}} = \frac{1}{r_s} \frac{1}{1 + sT_s\sigma} \quad (3.36)$$

$$Y_d = \frac{I_{sd}}{U'_{sd}} = \frac{1 + sT_r}{r_s(1 + s(T_s + T_r) + s^2\sigma T_s T_r)} \quad (3.37)$$

Replacing the real values we can obtain Fig. 3.19

This model present a coupling between the two axes and a stator electromotive force in q axis due to the rotor flux. It's necessary to do a decoupling of the axes and compensate the electromotive force. The scheme of the control appears as in Fig. 3.20

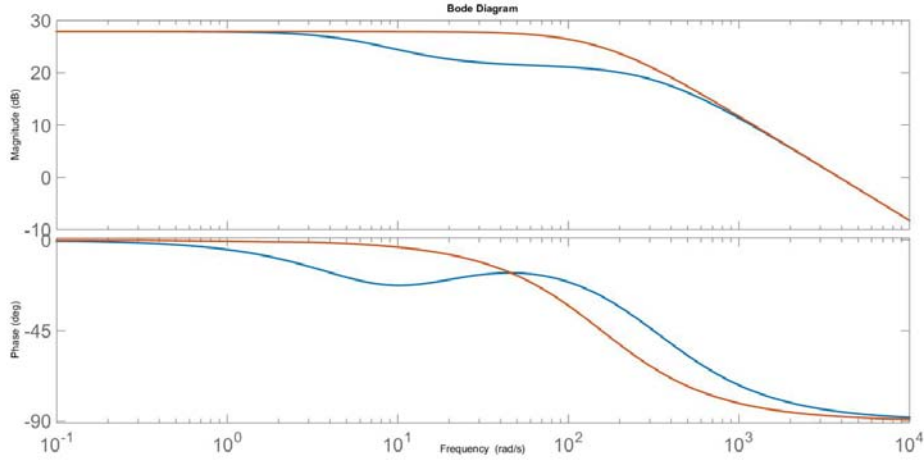


Figure 3.19 – Bode diagrams of Y_d (blue) and Y_q (red)

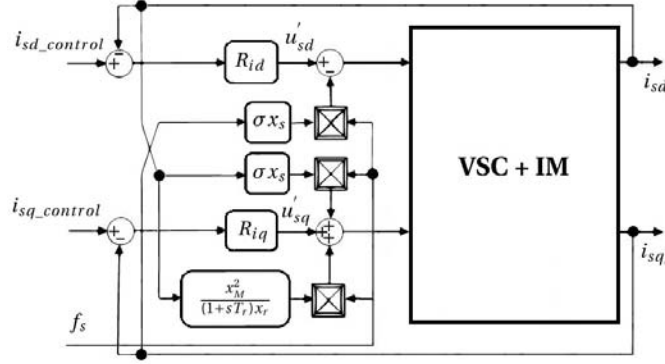


Figure 3.20 – FOC currents control scheme with decoupling and compensation

Thanks to this decoupling, the control scheme becomes really simple and easy to design as it's shown in Fig. 3.21, where G_c is the transfer function of the inverter, commonly taken as:

$$G_c = \frac{1}{1 + sT_{cm}/2} \quad (3.38)$$

with T_{cm} the commutation period, in our case $T_{cm} = \frac{1}{20kHz}$

3.2. Imposing the rotor flux through the components of the stator currents (FOC - *Field Oriented Control*)

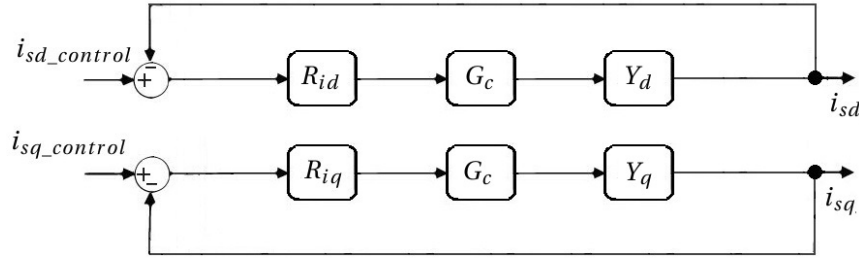


Figure 3.21 – FOC currents control scheme with decoupling and compensation

3.2.2 Design of the current regulators

A common criteria for the regulator design is to choose the breakpoint in order to remove the first pole of the motor transfer function. Since now, we will pass through the absolute values of the motor in order to easily implement later the code and the control.

A first estimation for the k_p and k_i parameters of the digital regulator (example with R_{iq}) can be made following the next formulas [3]:

$$R_{iq} = \frac{1 + sT_n}{T_i} \quad \text{Continuos regulator}$$

Calculation for a digital regulator PI:

$$T_n = \sigma T_s \quad \text{in order to remove the pole of } Y_q$$

$$T_i = 2 \frac{1}{R_s} T_D \quad \left(\frac{1}{R_s} \text{ gain of the motor transfer function} \right) \quad (3.39a)$$

$$T_D = k \frac{T_{cm}}{2} + \frac{T_{cm}}{6} + T_{cm} \quad (3.39b)$$

Where T_D the total delay, is composed in order by a terms corresponding the sampling delay multiplied by a factor depending on the type of regulator (for PI $k = 0.5$), the converter delay and a delay for the adquisition of the measurment and for the retention of the algorithm. We obtain:

$$K_i = \frac{T_{cm}}{2} \frac{1}{T_i} \simeq 1; \quad K_p = \frac{T_n - \frac{T_{cm}}{4}}{T_i} \simeq 200$$

Chapter 3. Induction Motor Control

CALCULATION FOR THE Q AXIS:

As we said, we impose that the zero of the transfer function of the regulator is exactly the pole of the transfer function of the motor:

$$R_{iq} = K_{piq} \frac{1 + s\tau_{iq}}{s\tau_{iq}}; \quad \tau_{iq} = T_s \sigma$$

Imposing a phase margin of 80° , we can get the crossover frequency:

$$\arg[G H_q(j\omega)] = -1.745 \quad (100^\circ)$$

$$\omega = \frac{2}{T_{cm}} \tan(1.745 - \frac{\pi}{2}) \simeq 7050 \frac{rad}{s}$$

With $G H_q(j\omega) = R_{iq} G_c Y_q$, the openloop transfer function of the q axis. Imposing the absolute value at 1 for this frequency, we can get the K_{piq} :

$$K_{piq} = R_s \tau_{iq} \omega \sqrt{1 + \frac{\omega^2 T_s^2}{4}} \simeq 210, \quad \text{and} \quad K_{iiq} = \frac{T_{cm} K_{piq}}{2\tau_{iq}} \simeq 0.8$$

CALCULATION FOR THE D AXIS:

Using the same method for the q axis, and imposing the zero of the regulator to the first pole of the Y_d , we obtain:

$$K_{pid} \simeq 258, \quad \text{and} \quad K_{iid} \simeq 0.6$$

In the real plant, we started from these values, and increasing step by step the K_i while decreasing the K_p we set up the values which give us a better response of the system in terms of rising time and overshoots (*cf.* Cap 5). Adjusting "by hand" we obtained the values in Tab. 3.1

Table 3.1 – Current Regulators Parameters

	K_p	K_i
R_{id}	125	2.0
R_{iq}	110	2.2

In this way the openloop transfer functions ($G H_d$ and $G H_q$) becomes:

3.2. Imposing the rotor flux through the components of the stator currents (FOC - *Field Oriented Control*)

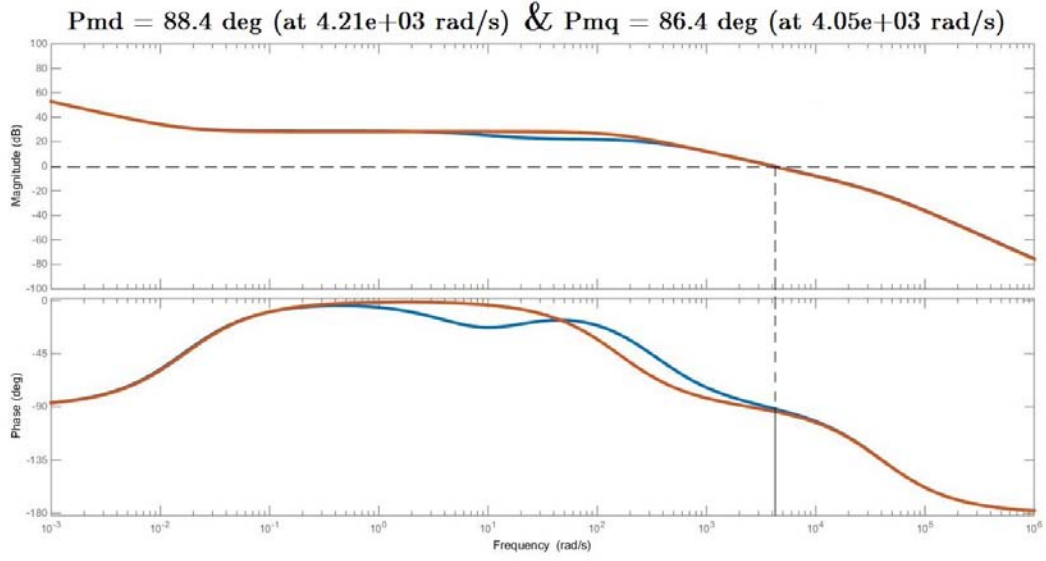


Figure 3.22 – Bode diagram of GH_d (blue) and GH_q (red) open-loop transfer functions

3.2.3 Design of the speed regulator

For the q axis, the closed-loop transfer function would be:

$$W_q = \frac{GH_q}{1 + GH_q}$$

It's a common to approximate it as:

$$W_q^* = \frac{1}{\left(1 + \frac{s}{v_A}\right)(1 + s \frac{T_{cm}}{2})}$$

With v_A the crossover frequency of GH_q . The next bode diagram shows that for frequencies lower than the converter frequency, the approximation can be considered good.

So the speed loop can be represented as in Fig. 3.24.

Considering an inertial load, and R_ω the speed regulator with the transfer function:

$$R_\omega = K_{p\omega} \frac{(1 + s\tau_\omega)}{s\tau_\omega}$$

Using the symmetrical optimum method, we can design the speed regulator in order to have:

$$\frac{1}{\tau_\omega} = \frac{v_A}{10} \simeq 405 \frac{rad}{s} \quad \text{and} \quad v_{A\omega} = \frac{\frac{1}{\tau_\omega} + v_A}{2} \simeq 2230 \frac{rad}{s}$$

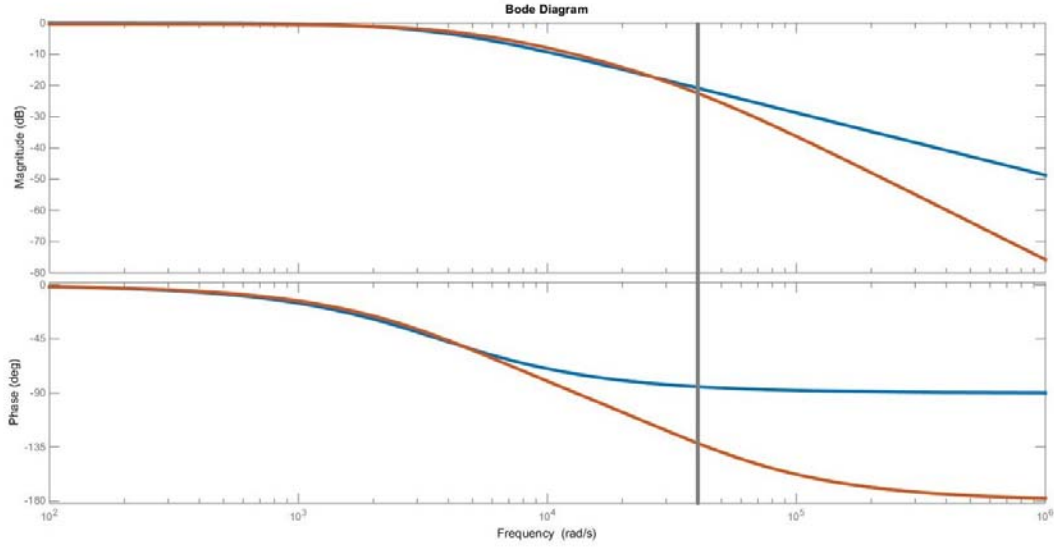


Figure 3.23 – Bode diagram of W_q (red) and W_q^* (blue)

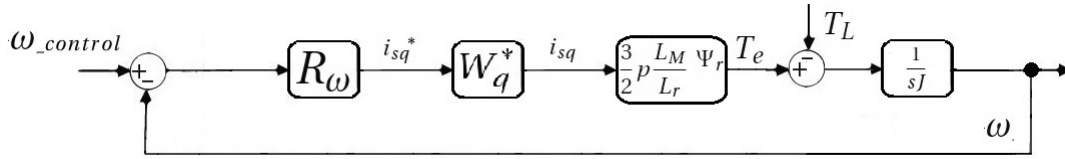


Figure 3.24 – Speed-Loop scheme

Imposing now, the absolute value to 1:

$$\left| GH_\omega(j\omega_{Aw}) \right| = 1$$

we obtain:

$$K_{p\omega} \approx 0.25, \quad \text{and} \quad K_{i\omega} \approx 0.05$$

As before, in the real plant, we started to check out the best behavior for the speed control and its parameters. As it's well known, the IM has a non-linear behavior respect to the speed. To get good performances we had to adjust the parameters of the PI regulator for different speeds, implementing an algorithm of hysteresis in order to not change the parameters with the same threshold. A simple scheme representing what it has been implemented in the algorithm is reported in Fig. 3.25, as well as the implemented values.

Instead choosing the crossover frequency of the open loop transfer function as the arithmetic

3.2. Imposing the rotor flux through the components of the stator currents (FOC - *Field Oriented Control*)

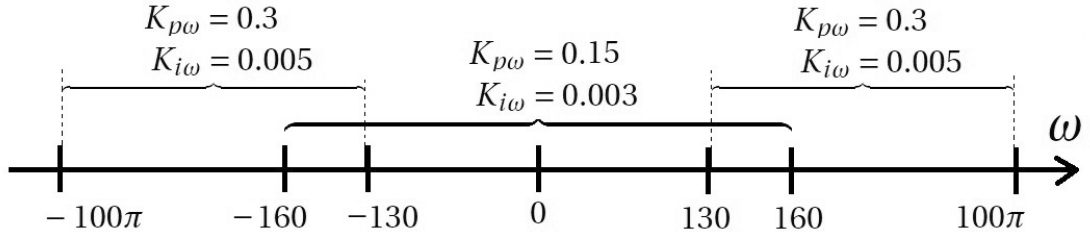


Figure 3.25 – Hysteresis speed regulator parameters

mean, we could decrease it choosing the geometrical mean as:

$$v_{A\omega} = \sqrt{\frac{1}{\tau_{\omega}}} v_A \simeq 1280 \frac{rad}{s}$$

The results in Chapter 5 are not accorded with this frequency. It'd be interesting comparing the drive behavior depending on the $v_{A\omega}$.

3.2.4 Simulation results

Before implementing FOC in the real plant, we simulated the scheme in Fig.3.18 using *Simulink*. First of all, we simulate a speed reference step. In this case, we can see that the rotor accelerate, in function of i_{sq}^* (limited by the anti-windup integral, setted to 25A in order to not exceed the sensor limit and to not stress too much the windings of the motor) reaching the desired speed reference with damped oscillations. We also checked that the rotor flux remains purely real, and we can see the presence of the FluxWeakening intervention. This simulations was made in *No Load* conditions, and are represented in Fig.3.26.

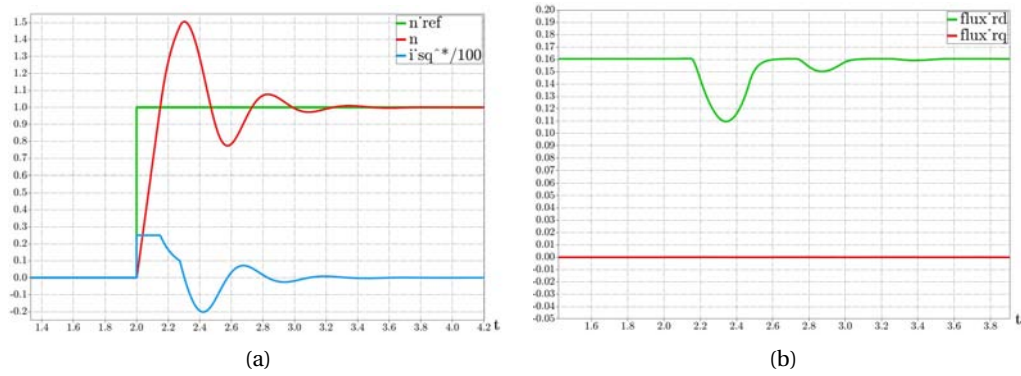


Figure 3.26 – Transient phenomena during a speed reference variation in No Load condition

In the same way, we did the same simulation in *Rated Load* conditions. The difference from before is the value of i_{sq}^* , resulting nonzero in steady state. The results are shown in Fig.3.27.

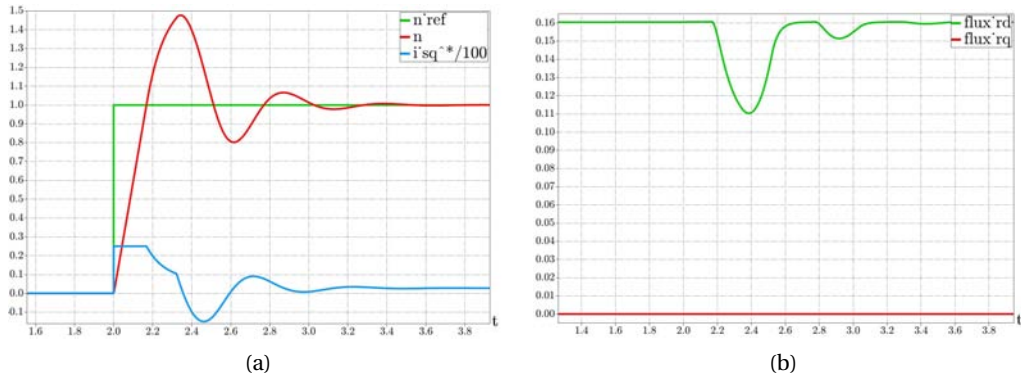


Figure 3.27 – Transient phenomena during a speed reference variation in Rated Load condition

3.2. Imposing the rotor flux through the components of the stator currents (FOC - *Field Oriented Control*)

In the next two figures, we show how the control works for one type of speed reference, with no load (Fig.3.28(a)) and under load (Fig.3.28(b)).

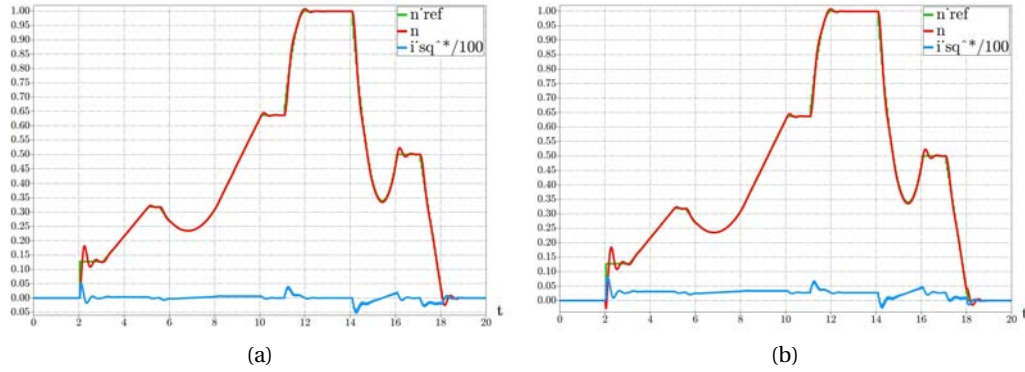


Figure 3.28 – Example of speed reference and the tracking of the motor [(a) : No Load (b) : Rated Load]

4 The Plant

In this chapter the setting-up of the plant and its relative parameters and choices (*e.g.* sensors, converters, control platform, motors etc ...) are described in detail, reporting the datasheets and the related reasons which led to the choice of their usage. This chapter wants to be a good overview of the real plant and its components (*cf.* Figs.1.2 4.1 and 4.2).

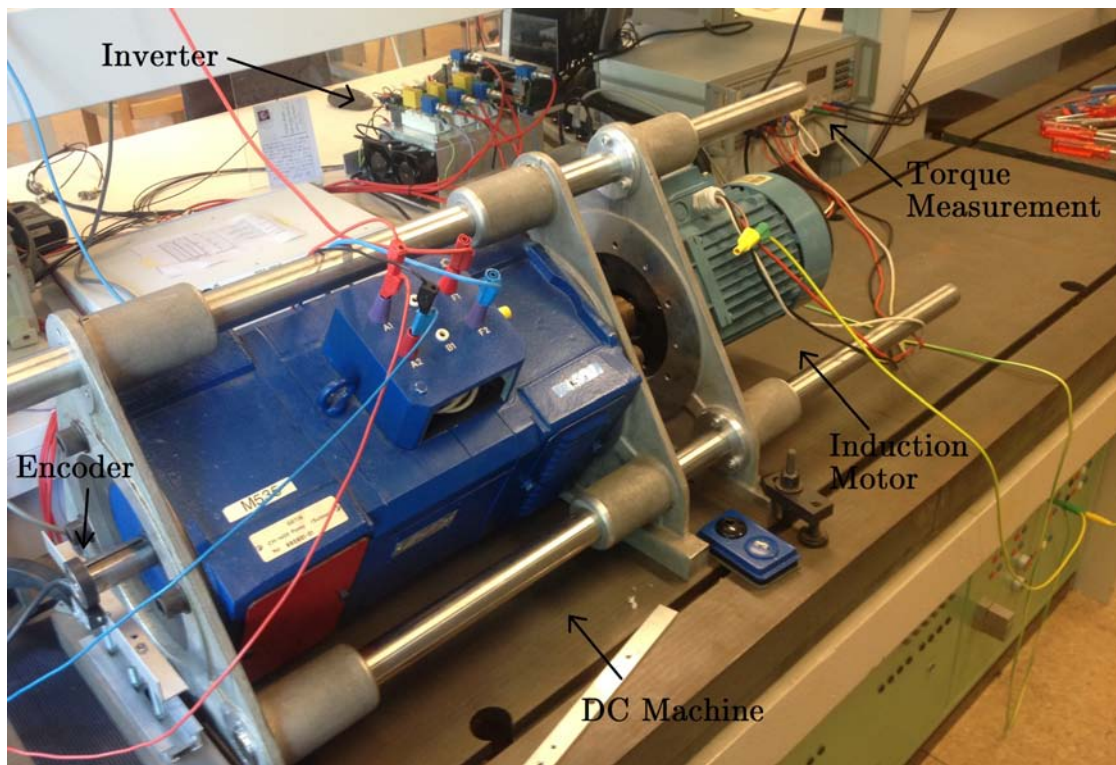


Figure 4.1 – Workplace 1

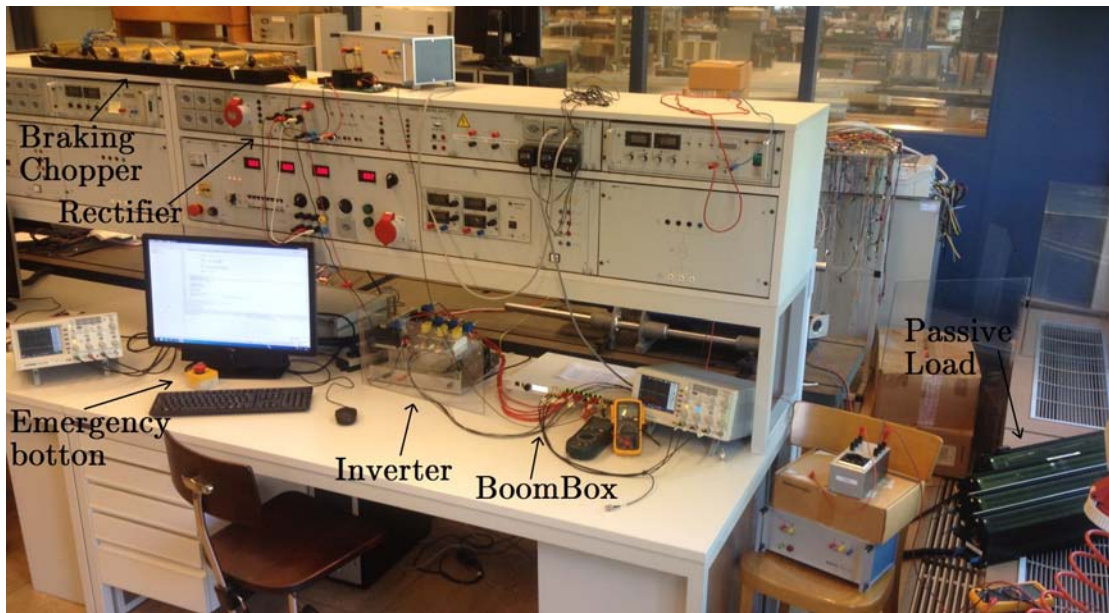


Figure 4.2 – Workplace 2

4.1 Control platform

Imperix SA is a Swiss company that in collaboration with LEI laboratory at EPFL has developed a versatile control platform called BoomBox, specially for power electronics applications in R&D environments. The BoomBox is not only a high-performance platform, but is also, from a software point of view, an open system that customers are welcome to self-appropriate and edit up to their own needs, while guaranteeing a safe and sound control implementation [8]. The BoomBox relies on two key concepts in order to speed up the development for the interfaces without any trade-off on safety : isolated fiber optic PWM outputs and versatile analog frontends that fit in with almost any sensor. The front panel view of the BoomBox is shown in Fig.4.3.

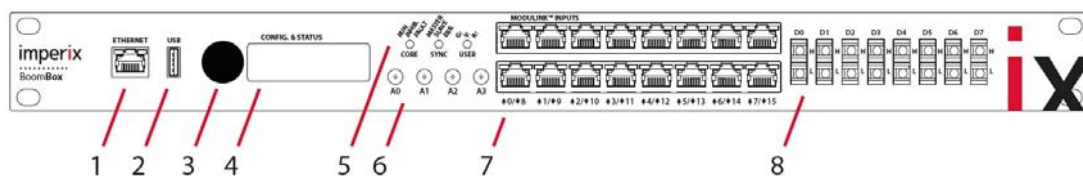


Figure 4.3 – Front panel view of the BoomBox. [1: Ethernet port; 2: USB type B device port; 3: rotary and push button; 4: LCD screen; 5: System and user LEDs; 6: SMA analog outputs; 7: ModuLink analog inputs; 8: Digital fiber-optic PWM outputs] [9]

The first step is to configure the right parameters for each sensor used, in such a way that the values of the magnitudes desirable, have a logical sense within the control code. For this purpose, it gives a schematic diagram of an any analog input of the BoomBox (Fig.4.4).

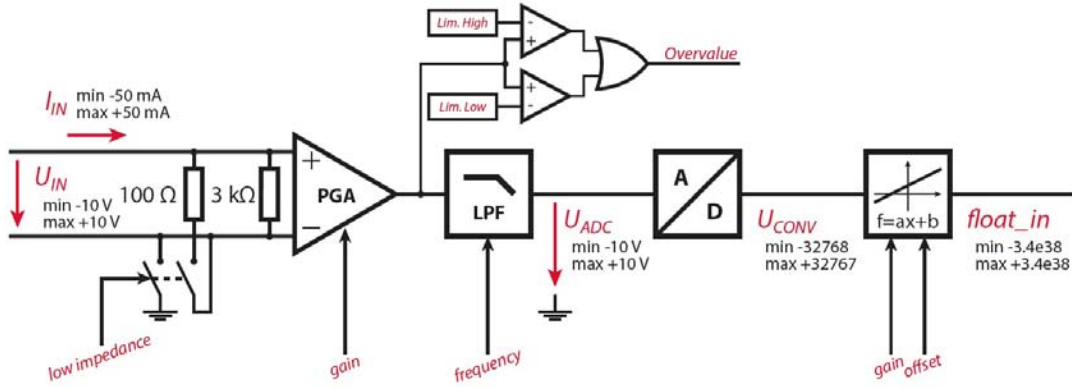


Figure 4.4 – Block diagram of each of the analog inputs [9].

The gain values can be 1, 2, 4 or 8. A fifth-order programmable filter can be activated on each analog input channel. The cut-off frequencies of this filter can be set individually and independently for each input channel (from 500Hz to 40KHz). In case of faults, the user can define two safety limits for each input channel for the input of the A/D converter: a high and a low one. In the last block of the scheme, the user can configure a and b parameters, corresponding to the sensitivity and the offset of the sensor. The sensitivity a of the entire conversion chain can be determined from the parameters of the employed sensors according to:

$$\frac{1}{a} = s_{FE} \frac{32768}{10} \quad (4.1)$$

where s is the sensitivity of the sensor and G_{FE} is the gain programmed on the frontend of the Boombox [10].

4.2 Sensors and Encoder

The implementation of a current control as FOC for Induction Motors need the measurement of the stator currents. For KCL just two currents are required, as the third one can be calculated. However we will use three current sensors, to be able to protect the plant immediatly if some current values exceed the limits. As a matter of safety and check of the state of the plant, two sensor are also used for the measurement of the DC bus Voltage. The LEM sensors was already mounted on the converter, and being suitable for the application, in terms of admissible currents, voltages and compatibility with the control platform, have been selected for this use. In Fig.4.5 is reported the voltage and current sensors behavior when the shaft is blocked, supplying the motor with 380 V in the DC bus. The currents reach $30A_{pp}$.

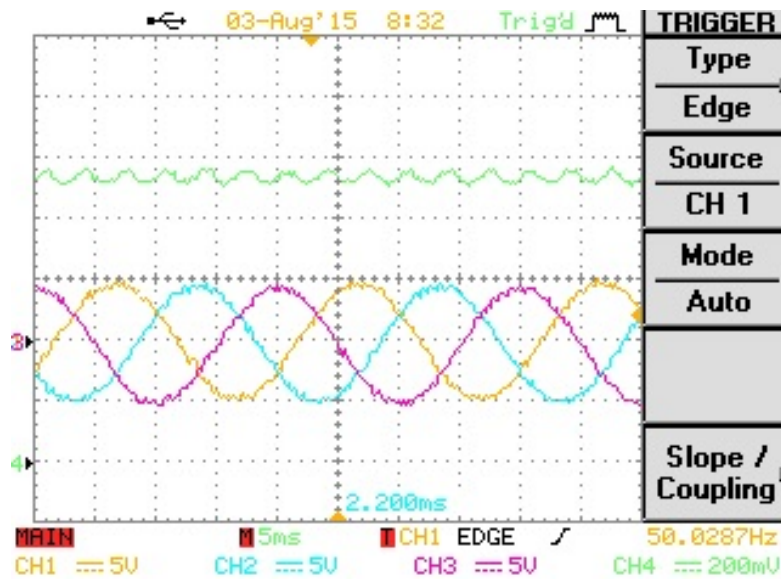


Figure 4.5 – Currents and DC bus voltage in short-circuit operation [CH1/CH2/CH3 : $\frac{\text{phase currents}}{3}$ CH4 : $\frac{V_{DC}}{400}$]

Field oriented control of squirrel cage IM in high performance drives needs the accurate knowledge of the rotor speed, for this reason encoders are highly implemented in electrical drives. The most common encoders are the optical ones which using a Gray disk and photodiodes can generate a two digital output signals (90 degree shifted) at the frequency of the shaft multiplying the PPR (Pulses per Revolution). Concerning the compatibility of the output signals of the encoder and the input signal of the BoomBox, in the case of considering the digital encoder, there would be the need to build an interface for the platform. The digital inputs of the BoomBox (GPI-General Purpose Inputs), have the advantage of an optical isolation, but on the other hand, leads to a slow behavior not suitable for high frequency signals as the digital's ones.

So a magnetic Sin/Cos analog incremental encoder has been chosen for the high resolution of the rotor speed for FOC implementation. A logical scheme for the detection of the rotor speed is shown in Fig.4.6. It uses Hall effect sensors to generate two analog signals in order to implement an external interpolation to obtain the rotor speed. As the current LEM sensors, the encoder has to be configured using the ModuLink analog inputs of the BoomBox. The characteristics of the sensors are reported in Tab.4.1 and all the parameters for the configuration are reported in Tab.4.2, considering the Fig.4.4 and the eq.4.1.

Table 4.1 – Sensors and Encoder [6] [7]

Name	Number of sensors	Purpose	Primary nominal value
LEM LA 25-NP	x3	Stator currents	25 A <i>rms</i>
LEM LV 25-P	x2	DC bus voltages	10 mA <i>rms</i>
Baumer ITD 49H00	x1	Rotor speed	sin/cos 1 V_{pp}

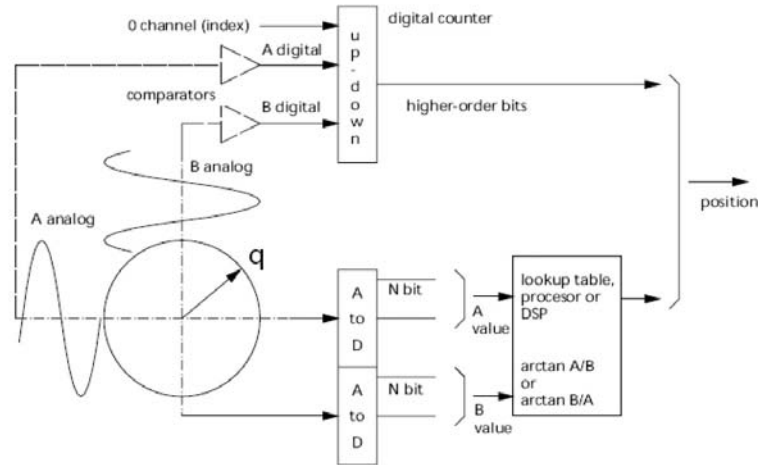


Figure 4.6 – Logical scheme for the detection of the rotor speed in Sin/Cos encoders [11].

Table 4.2 – Sensors and Encoder parameters for Bbox configuration [6] [7]

	$I_a/I_b/I_c$	U_{dc0}/U_{dc1}	ω
Specification	Current phase a, b and c	DC bus	Rotor speed
Sensor	LEM LA 25-NP	LEM LV 25-P	Baumer ITD 49H00
Value	± 36 A	± 700 V	± 1500 rpm
Sensitivity s	100 [mV/A]	$250/R_{IN}$ [V/V]	/
Accuracy	$\pm 0.5\%$	$\pm 1\%$	$\pm 0.3^\circ$
Input impedance	100 Ω	100 Ω	3 K Ω
G_{FE}	2	4	8
LPF	2.5 kHz	bypass	2.5 kHz
Limits	$\pm 10 * I_n$	300/-25 V	/
Bbox Limits	± 7	6.4/-0.5	/
a	$1.52786e^{-3}$	$14.6e^{-3}$	$\approx 3.67e^{-5}$
b	-0.015	-0.45/2.75	0/0.045/0.5

With $R_{IN} = 47$ K Ω .



Figure 4.7 – Encoder mounting [7]

After mounting the encoder on the shaft (Fig. 4.7b), the outputs signals have been checked. As it's shown on Fig.4.8 the first two channels are phase shifted of 90 degree. At the nominal speed of the motor (1500 rpm) we correctly have a reference signals at 25 Hz, and have chosen 64 PPR, the Sin and Cos channels give us a maximum frequency of 1.6 KHz, as it has been calculated from the oscilloscope. At the same time we checked that the amplitude of the Sin Cos signals is correctly $1 V_{pp}$ (we used differential probes with an attenuation ratio 1/100), meanwhile the reference signals is 1 V, instead of 2.5 V declared in the datasheet.

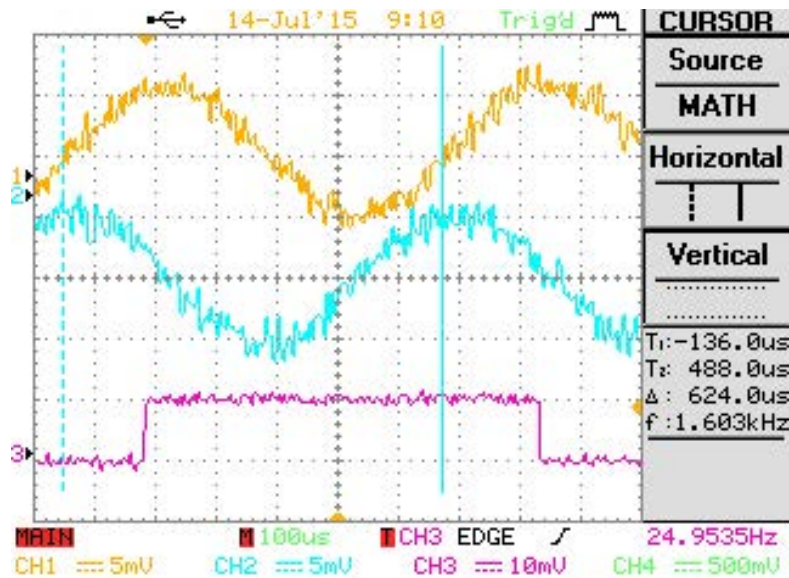


Figure 4.8 – Sin/Cos encoder output signals [CH1 = Channel A; CH2 = Channel B; CH3 = Channel N (*reference*)]

To calculate the rotor speed, we implemented a PLL (*Phase Lock Loop*). The system shown in Fig. 4.9 normally needs a signal, and using the orthogonal system generation (e.g. *SOGI*[12]),

it generates the second one (orthogonal to the other one). In our case, we used directly the sine and cosine signals at the input of the PLL, and we set the V_q generated from the $\alpha\beta/dq$ conversion block to zero, as the feed forward ω_{ff} . Using the symmetric optimum method for the PI controller parameters ([13]), K_p was configured at 3000 and K_I at 30.

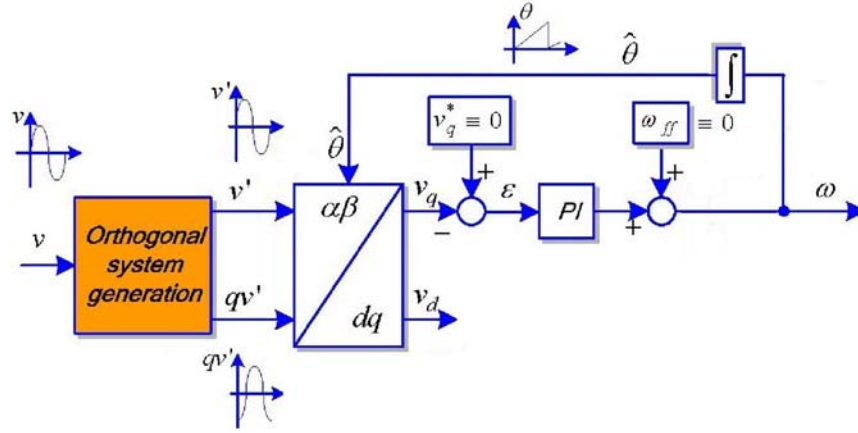


Figure 4.9 – General structure of a single phase PLL [12]

To check how the PLL follow the signal, we supposed that at the *No Load* conditions, the motor has the biggest acceleration, supposing no mechanical losses. The maximum time, in which the machine can reach the *No Load* speed, from zero, it has been calculated in the following way:

$$\Delta t = \frac{\Delta \omega J}{T_e - T_L} = \frac{(2\pi 50 - 0)0.0043}{10 - 0} \approx 0.135s \quad (4.2)$$

We also verified that lower K_p implies lesser crossover frequency and higher phase margin of the openloop transfer function. Hence, for lower K_p , the PLL will better attenuate the effects of unbalance and harmonics [13], but the response will be slower. Having a low pass filter in both sine cosine channels, we don't mind about big oscillations, so we can fix K_p in high values. In fig. 4.10 we can see how the PLL code follows the acceleration and deceleration of the motor, in an almost completely immediate way. As it can be noticed from the picture, we imposed high and low limits for saturation at the output of the PI.

We also verified the behavior for different range steps of speed, observing an acceptable tracking, for the FOC implementation, for almost the entire admissible range of speed (Fig. 4.11).

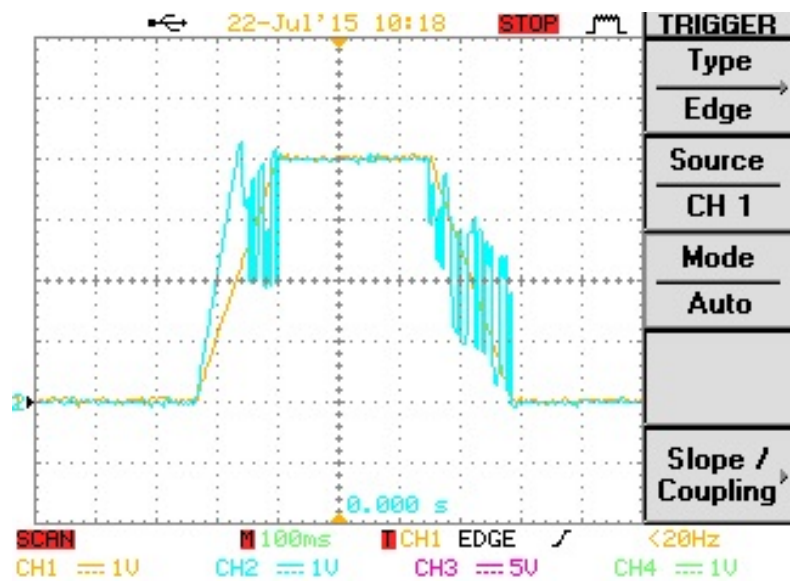


Figure 4.10 – Step of acceleration/deceleration and rise time of the PLL from 0 to nominal speed and backwards [CH1 : simulated shaft speed CH2 : PLL output]

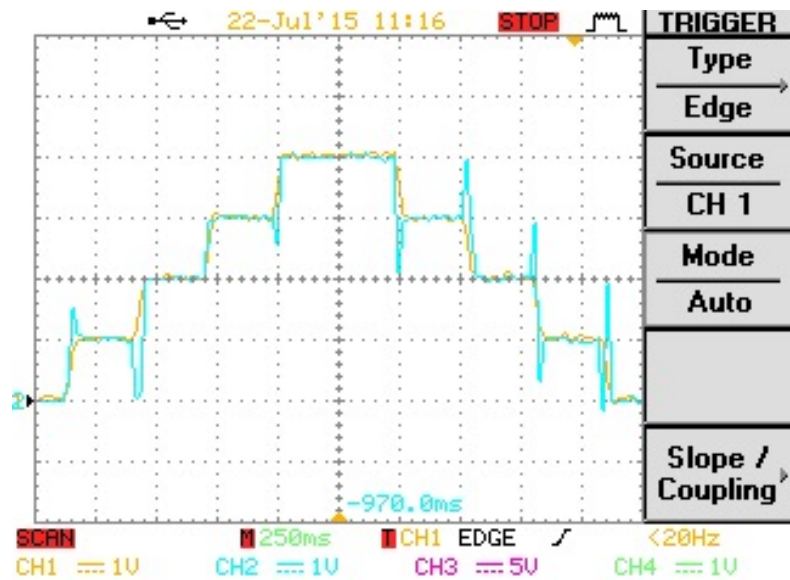


Figure 4.11 – Different speed steps and response of the PLL [CH1 : simulated shaft speed CH2 : PLL output]

4.3 Inverter

A three phase inverter has been used for the IM supply, characterized by 3 IGBTs (datasheet in [14]), a DC bus till 1.2 kV (series of 2 couple of 2 condensators in parallel) and the LEM sensors. Using three references phase shifted of 120 degrees, the inverter produces a set of three voltages PWM (Pulse width modulation) with maximum amplitude of $\frac{V_{DC}}{2}$, phase shifted of 120 degrees. Using the same carrier for three phases (unipolar commutation), the voltages line-to-line fluctuate from V_{DC} , 0 and $-V_{DC}$. Remember the rms value for the voltages line-to-line, working in linear modulation, will be $\frac{\sqrt{3}}{2\sqrt{2}} m_a V_{DC}$, so it's required to increase the DC voltage up to 400 V to reach the nominal value for the line-to-line voltages. At the beginning of testing, due to reasons of technical availability in the laboratory, it was not possible to exceed 400 V in the DC link, verified not be a big deal for the machine, in terms of speed and control efficiency. The electrical scheme of the inverter is shown in Fig. 4.12, and how it really appears in Fig. 4.13.

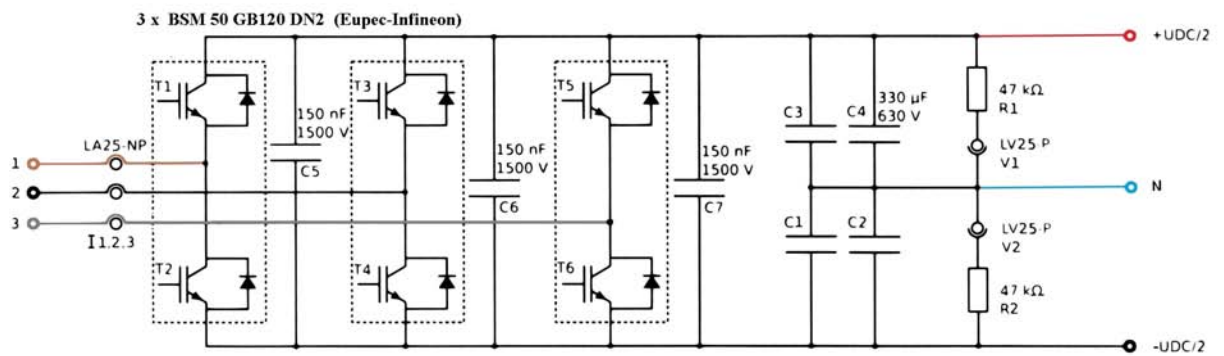


Figure 4.12 – Electrical scheme of the inverter (Internal LEI project)

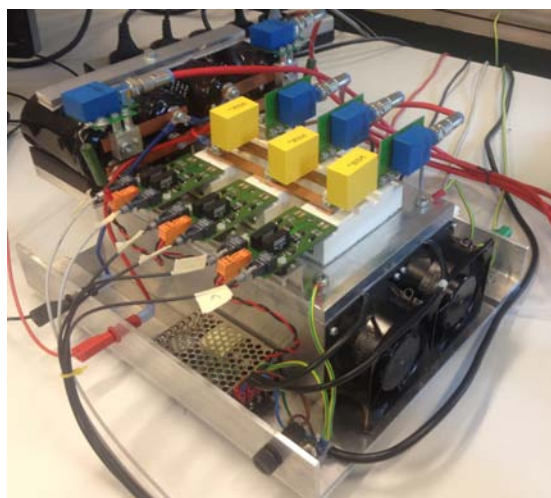


Figure 4.13 – Inverter

Chapter 4. The Plant

We are going to present how the PWM looks like in steady state, *No Load* operation, with a 400 V in the DC bus and nominal speed supplying the motor at 50 Hz (Fig. 4.14). The green line (V_{LLab}) was measured by a differential probe with an attenuation ratio 1/200 while the purple and blue ones with an attenuation ratio 1/100. The yellow line represents the duty cycle of the first leg, from 0 to 1, in order to compare it with the carrier. Instead the red sinusoidal wave represents $V_{a_ref} - V_{b_ref}$ in agreement with the green PWM.

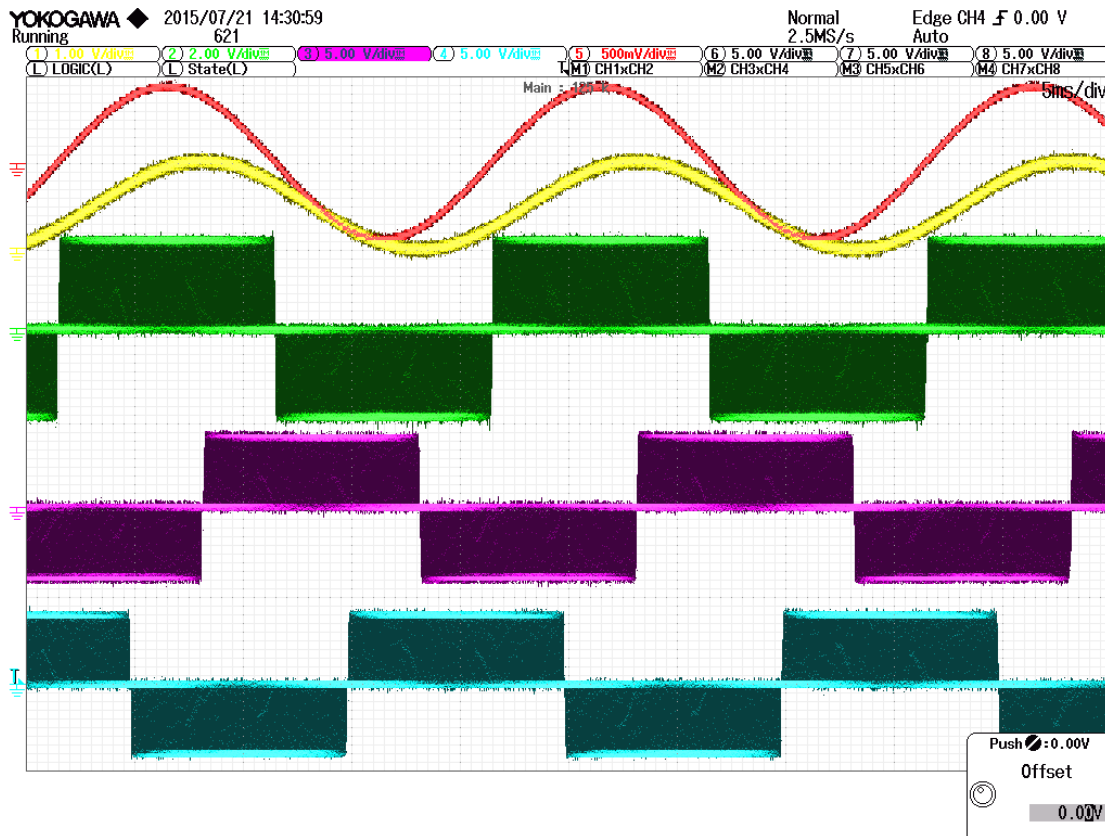


Figure 4.14 – Inverter typical PWM

In addition, the following points have been verified as well:

- Two IGBTs of the same leg, always operate in an alternated way;
- the control platform implement directly a delay for the dead time of the IGBTs, imposed at 800 ns (*cf.* [14]). The switch-on command from the gate, occurs exactly 800 ns later then the switch-off command of the other IGBT;
- the gate signal, to switch off or on, goes from 0 to 15 V;
- the frequency switching is 20 kHz (period of 50 μ s);
- the three PWM are not phase shifted between them.

4.4 Braking chopper

As a matter of safety and check of the state of the plant, a braking chopper was built and tested. It is composed by a resistor, a controlled switch (a MOSFET in this case) and a free wheeling diode, in the case where the inductive part of resistor bears an important role (*cf.* Fig. 1.2). A braking chopper is an electrical switch that limits the DC bus voltage by switching the braking energy to a resistor transforming it in heat. This happens if the power flow tries to change direction, from the machine to the grid. Due to having an uncontrolled converter between the inverter and the grid (the rectifier), this operation would be impossible to execute, increasingly raising the voltage of the DC bus, until the capacitors risk an explosion. So the braking chopper will be automatically activated when the actual measured DC bus voltage exceeds a specified level, 450 in this case (*cr.* Annex B).

It has been chosen a $70\ \Omega$ resistor, in order to have a nominal current of $\frac{400V}{70\Omega} = 5.7A$. The system has been protected by fuses and tested with the code, controlling it at 200 Hz. We can state that the protection system will be able to dissipate a substantial power and that its operation is as expected (*cf.* Fig. 4.15). In the figure it is shown how the evolution and the value of the current flowing through the resistor, are those expected (yellow line). We also see that all the rated voltage falls across the resistor when the MOSFET is placed in ON-state (blue line). It's also shown the trend of the voltage at DC bus (purple line). It's also shown the trend of the voltage at DC bus (purple line).

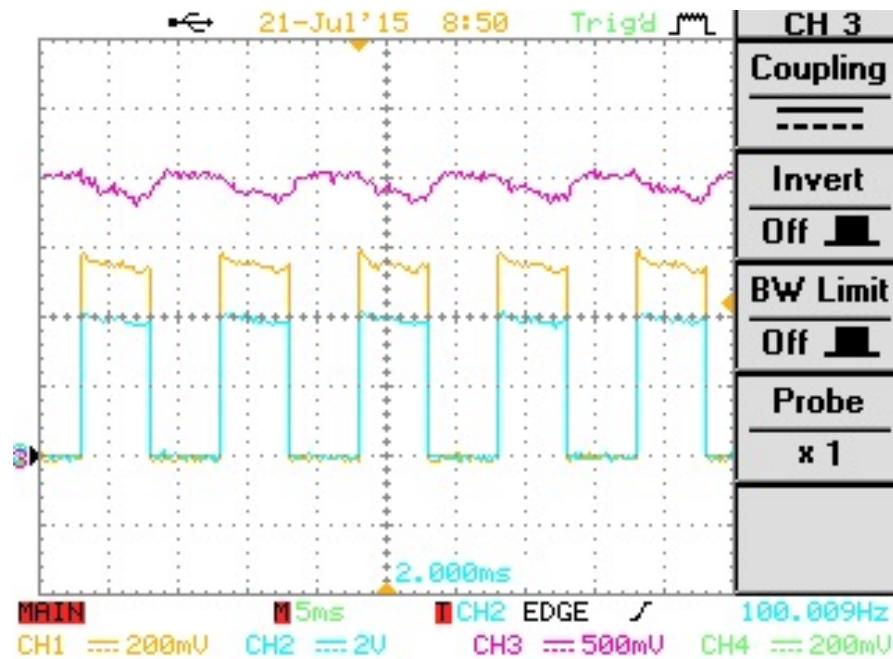


Figure 4.15 – Typical operation of the Braking Chopper [CH1 : $\frac{i_{chopper}}{10}$ CH2 : $\frac{V_{chopper}}{100}$ CH3 : $\frac{V_{DC}}{100}$]

Chapter 4. The Plant

The braking chopper appears as in Fig. 4.16. We can notice the power supply for the gate of the MOSFETs ($5 V_{DC}$), the optical fiber connected to the BoomBox, and the 70Ω resistor composed by seven resistors in series (10Ω each one). In order to switch, it has been recycled an half bridge converter having two MOSFET already mounted on a PCB. This is done using only the MOSFET below, and leave the upper one always opened. It's also needful connect the midpoint P6 to the braking resistor (cf. Fig. 4.17).

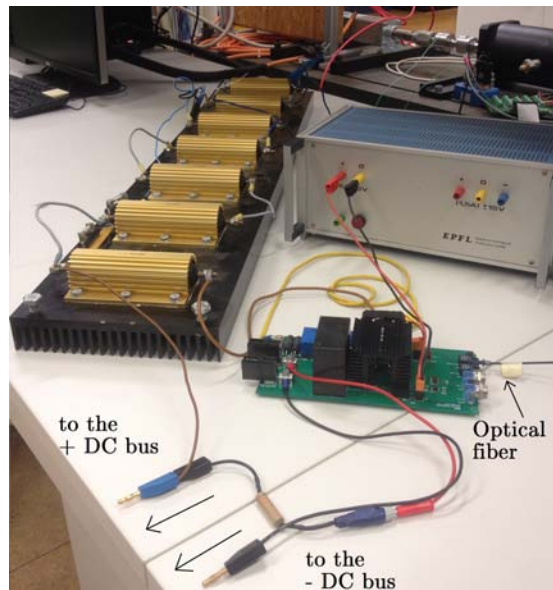


Figure 4.16 – Braking Chopper

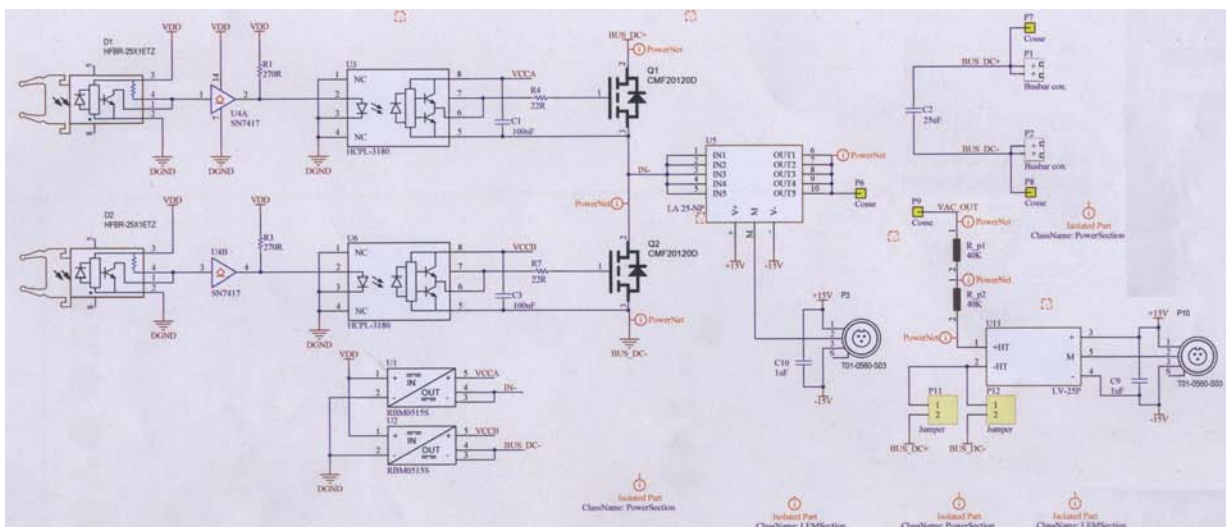


Figure 4.17 – PCB electrical scheme of the converter used for the braking chopper system (Internal LEI project)

4.5 DC Machine

To charge a Load on the shaft of the Induction Motor, a DC machine as a generator has been used. As it is shown in Fig.4.18, it has been supplied with an independent excitation. The DC machine is shown in Fig.4.1.

Applying a constant direct voltage to the inducer windings (U_f), we generate a constant inducer flux ϕ staying down to the saturation limits of the DC machine. As it's known the relations between the torque/current and speed/voltage are:

$$T = k\phi I_a \quad \text{and} \quad U_a = k\phi n$$

with k the DC machine coefficient, I_a the armature current, U_a the internal generated voltage and T the generated torque.

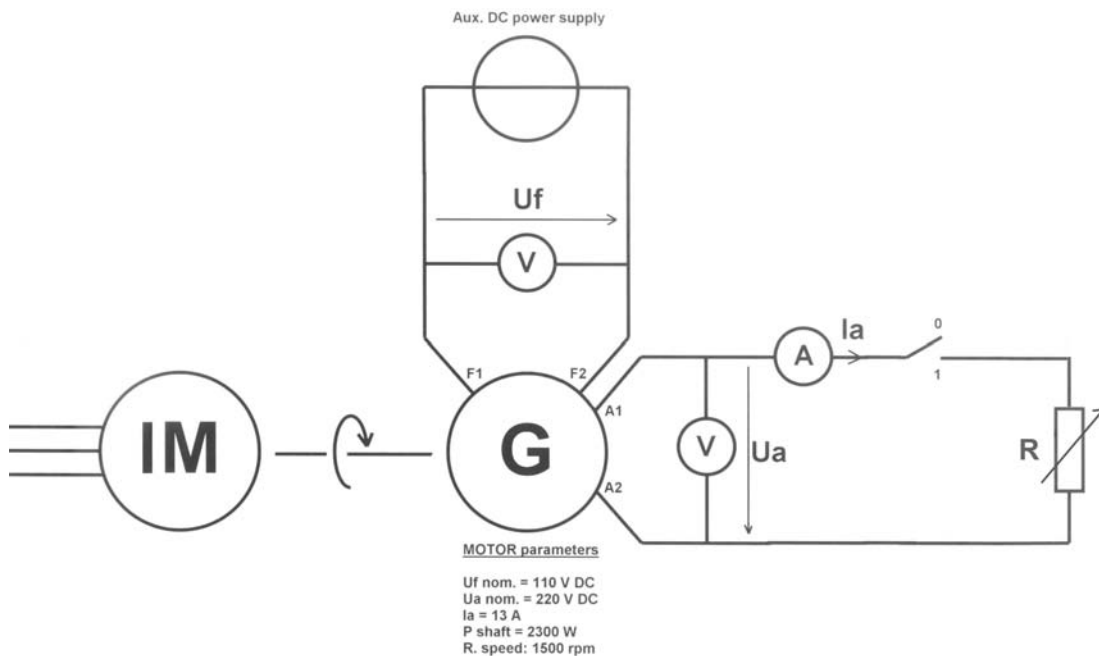


Figure 4.18 – Independent excitation of a DC machine

Notice that the Load is not constant with the shaft's speed. In order to have the same resistance torque on the shaft, increasing the speed (speed reference in FOC), we have to proportional increase the passive load (R), in order to have the same armature current. With this caution, we are now able to change the Load and test the Induction Motor performances.

5 Experimental results

Several experiments and tests of the system were conducted to verify the operation of the hardware both the software. The hardware behavior is mostly explained in the Chapter 4, and in this Chapter will be shown the behavior of the control code, in various operation conditions. Starting with the *No Load* operation, we are going first to verify the correct operation of the tranformation blocks (Park transformation blocks), and the motor model (*cf.* Fig.3.18), in an Openloop control. The decoupling and compensation block is also verified, as the current loops. After that, the control scheme was closed to pass to the Closedloop operation, and verify the correct behavior of the whole FOC strategy. The performance of the system is evaluated by using the oscilloscopes GW INSTEK GDS-2204 and YOKOGAWA DLM4058.

5.1 Open-loop

5.1.1 Park transformation blocks

To verify the correct operation of the Park transformation blocks, an Openloop control was implemented. Imposing a variable f_{s_ref} and two constants values for V_{sd^*} and V_{sq^*} , we verified the duty cycles given to the modulator, the measured currents in Timedomain and in dq . First we imposed a $f_{s_ref} = 1 p.u.$ and $V_{sd^*} = 0$, $V_{sq^*} = 1$. All this experiments were done having 200 V in the DC bus. The ducty cycle of the leg *a* and *b* are shown in the next figure, as well as the value of V_{sd^*} and V_{sq^*} .

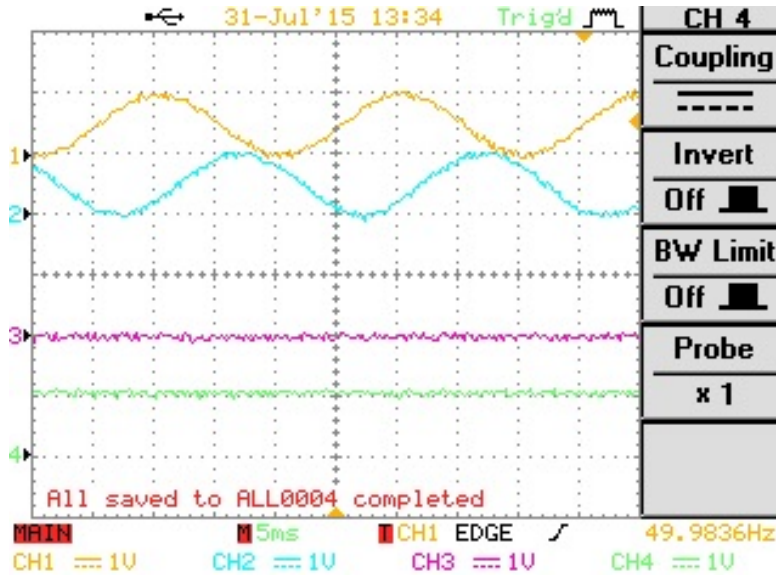


Figure 5.1 – Verification of Park block for the stator voltages $f_{s_ref} = 1 p.u.$ [CH1 : duty cycle a CH2 : duty cycle b CH3 : $V_{sd}^* = 0$ CH4 : $V_{sq}^* = 1$]

In the same operation conditions, in the next figure, the measured currents are shown, in both abc and dq reference frame, first without any LPF, and then with a LPF setted to 2.5 kHz.

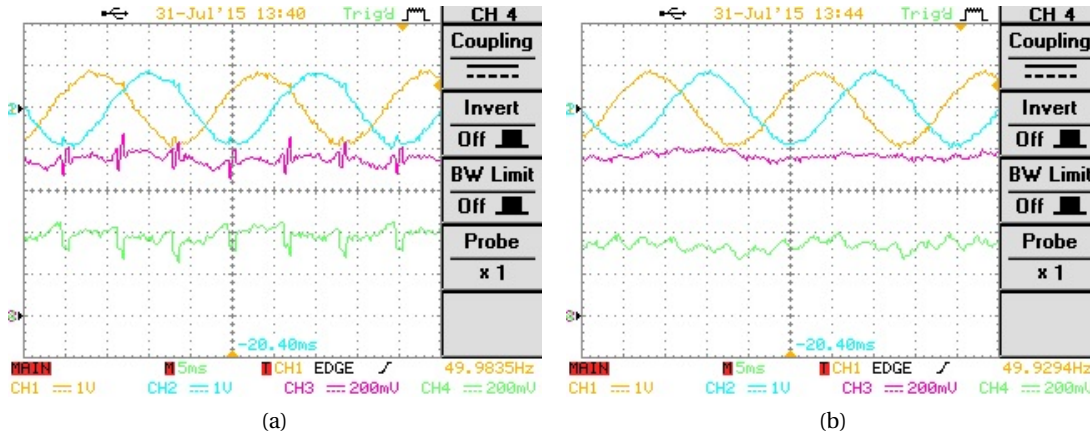


Figure 5.2 – Verification of Park block for the stator currents (a. without LPF b. LPF = 2.5 kHz) $f_{s_ref} = 1 p.u.$ [CH1 : i_a CH2 : i_b CH3 : i_{sd} CH4 : i_{sq}]

As we can notice, using a LPF for the currents sensors, the noises, probably derived from the IGBT switching, are cutted off. This technique is useful to avoid oscillation problems, which could lead to instability of the control. In the next figure is shown how inverting the values of V_{sd}^* and V_{sq}^* , the relatives measured currents in dq are exchanged mutually.

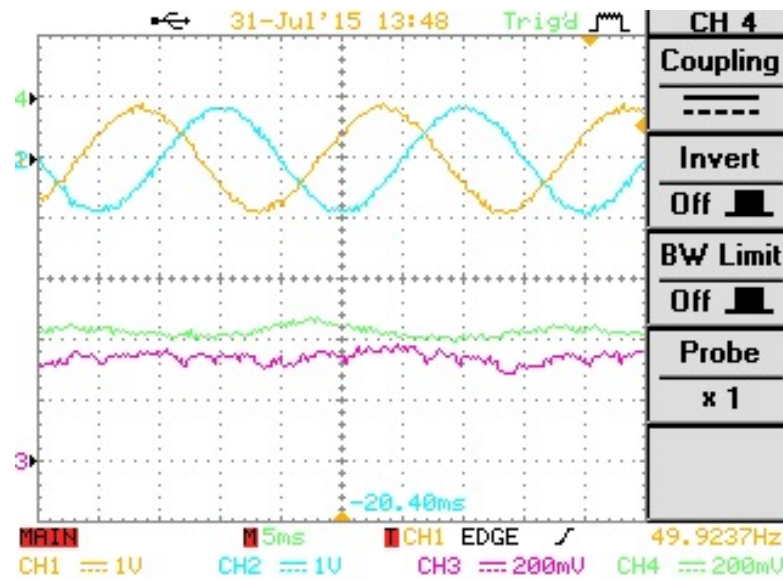


Figure 5.3 – Verification of Park current block for inverted stator voltages $V_{sd}^* = 1$, $V_{sq}^* = 0$, $f_{s_ref} = 1 p.u.$ [CH1 : i_a CH2 : i_b CH3 : i_{sd} CH4 : i_{sq}]

In the next two figures the currents measured are shown as the duty cycle of the first leg, imposing different f_{s_ref} . We can notice that the frequency of the duty cycle decrease according to the imposed stator frequency, as well as the Time-domain measured current. The dq magnitude of the currents are instead depending on the frequency. Normally the q value remains more or less the same, while the d value increase or decrease proportional to the frequency, due to the fluxes variations.

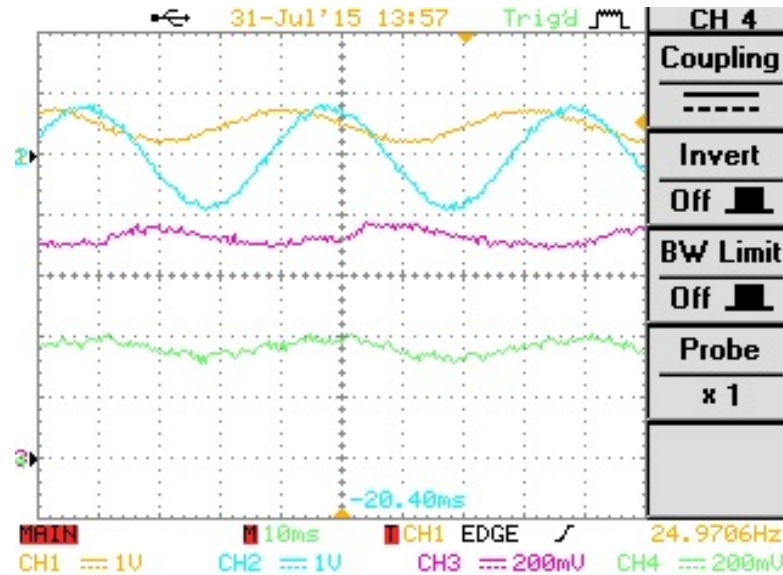


Figure 5.4 – Verification of Park blocks at $f_{s_ref} = 0.5 p.u.$ [CH1 : duty cycle a CH2 : i_a CH3 : i_{sd} CH4 : i_{sq}]

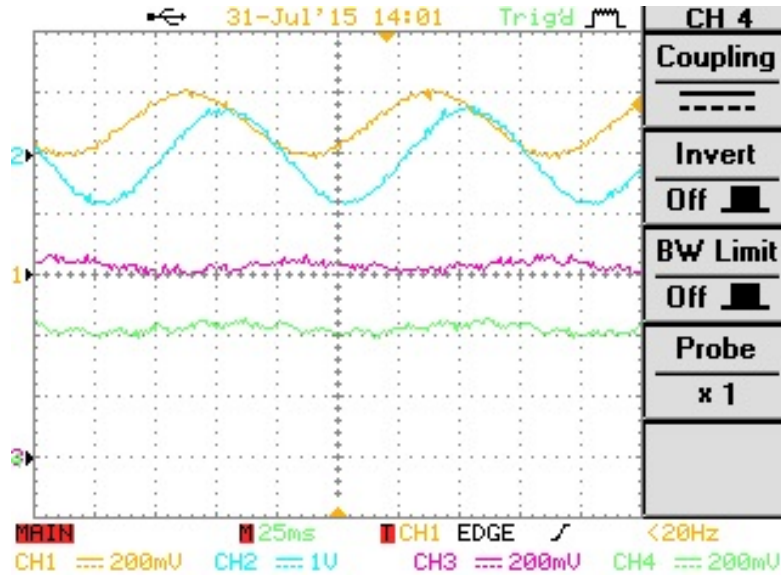


Figure 5.5 – Verification of Park blocks at $f_{s_ref} = 0.5 p.u.$ [CH1 : duty cycle a CH2 : i_a CH3 : i_{sd} CH4 : i_{sq}]

We also checked the speed response of the motor applying step of frequencies. Both the next two cases we have imposed $V_{sd}^* = 0$, $V_{sq}^* = 1$ with the DC bus to the nominal value. The results are here shown:

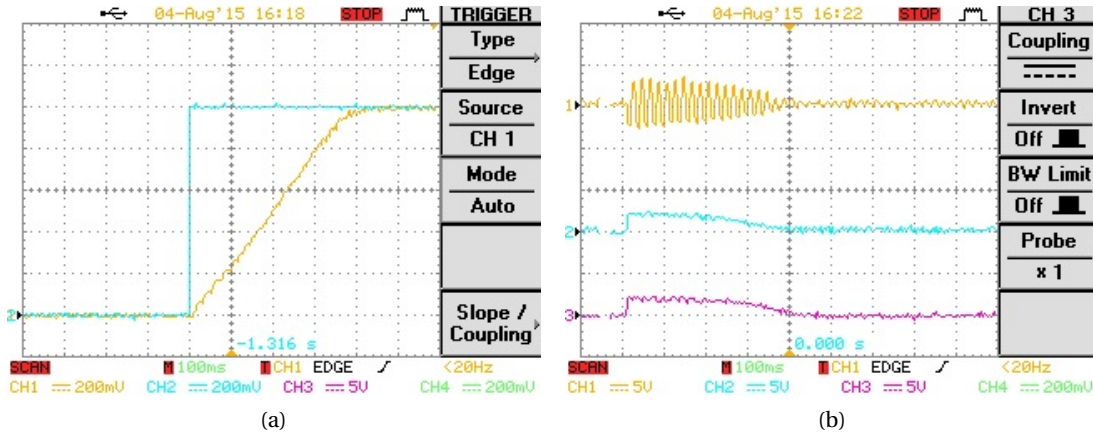


Figure 5.6 – Speed and stator currents in transient from $f_{s_ref} = 0 p.u.$ to $f_{s_ref} = 1 p.u.$ (a) [CH1: $\frac{\omega_{mes}}{\omega_n}$ CH2: f_{s_ref}] (b) [CH1: $\frac{i_a}{5}$ CH2: $\frac{i_{sd}}{5}$ CH3: $\frac{i_{sq}}{5}$]

A detail in steady state operation of the i_{sd} and i_{sq} is shown in the next figure, where we notice that $i_{sd} \approx 1.5 A$ and $i_{sq} \approx 0.25 A$:

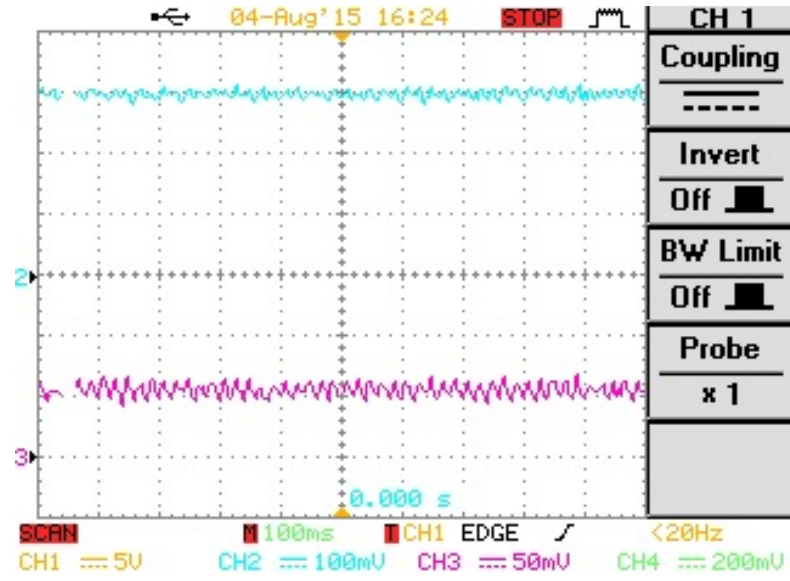


Figure 5.7 – dq statoric currents with dq stator voltage imposition in steady state $f_{s_ref} = 1 p.u.$
[CH2: $\frac{i_{sd}}{5}$ CH3: $\frac{i_{sq}}{5}$]

In the same way we show here the response imposing a step frequency from $f_{s_ref} = 1 p.u.$ to $f_{s_ref} = 0.5 p.u.$

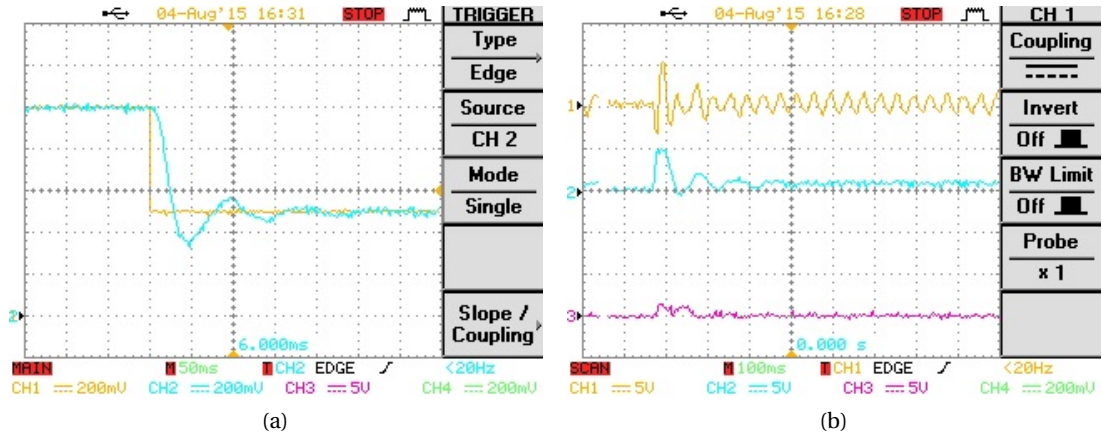


Figure 5.8 – Speed and stator currents in transient from $f_{s_ref} = 1 p.u.$ to $f_{s_ref} = 0.5 p.u.$ (a) [CH1: $\frac{\omega_{mes}}{\omega_n}$ CH2: f_{s_ref}] (b) [CH1: $\frac{i_a}{5}$ CH2: $\frac{i_{sd}}{5}$ CH3: $\frac{i_{sq}}{5}$]

A detail in steady state operation of the i_{sd} and i_{sq} is shown in the next figure, where we notice that now the currents are increased to $i_{sd} \approx 5 A$ and $i_{sq} \approx 0.75 A$:

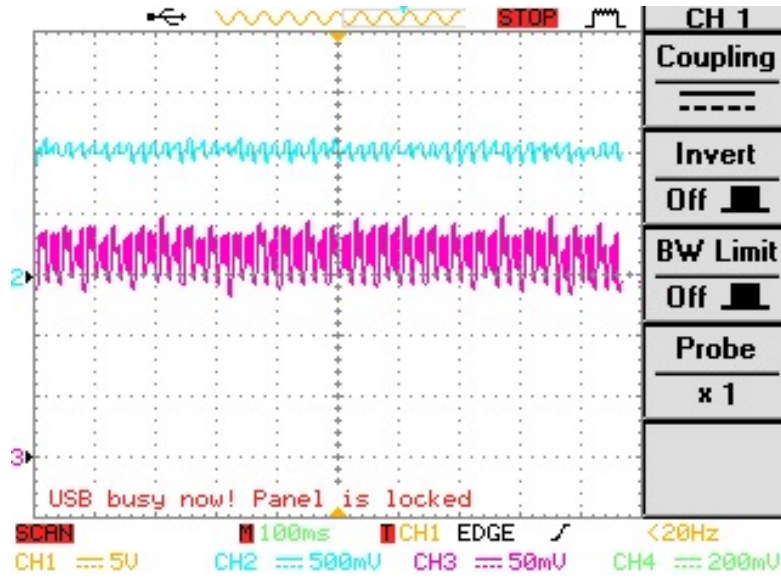


Figure 5.9 – dq statoric currents with dq stator voltage imposition in steady state $f_{s_ref} = 0.5p.u.$ [CH2: $\frac{i_{sd}}{5}$ CH3: $\frac{i_{sq}}{5}$]

5.1.2 Motor model

Before showing the behavior of the Motor model block in the Fig.3.18, it's necessary a precision about how to implement the transfer function to calculate the rotor flux (*cf.* Fig.3.17) in C. Regarding this calculation we can write it in s domain:

$$\frac{\Psi_r}{x_M i_{sd}} = \frac{\frac{1}{T_r}}{\frac{1}{T_r} + s} \quad (5.1)$$

Using the Tustin's method to pass in z domain (discrete-time) with a first-order bilinear approximation we obtain:

$$\frac{\Psi_r}{x_M i_{sd}} = \frac{\frac{T_s}{T_s+2T_r}(1+z^{-1})}{1+z^{-1}(\frac{T_s-2T_r}{T_s+2T_r})} = \frac{n(1+z^{-1})}{1+dz^{-1}} \quad (5.2)$$

With T_s the sampling time. This last equation is easily implementable in C (*cf.* Appendix B - routine *FirstOrderLag*), following the next blocks diagram.

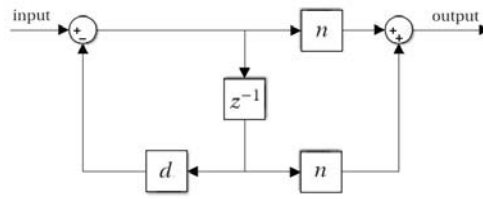


Figure 5.10 – First order lag block diagram

As before we imposed $f_{s_ref} = 1 p.u.$, $V_{sd}^* = 0$ and $V_{sq}^* = 1$. We verified that the angular position estimated by the motor model (blue line) is equal to the angular position imposed by the power supply (orange line). We also checked the $\frac{f_s}{\omega_n}$ reported in CH 3, and the rotor flux estimation.

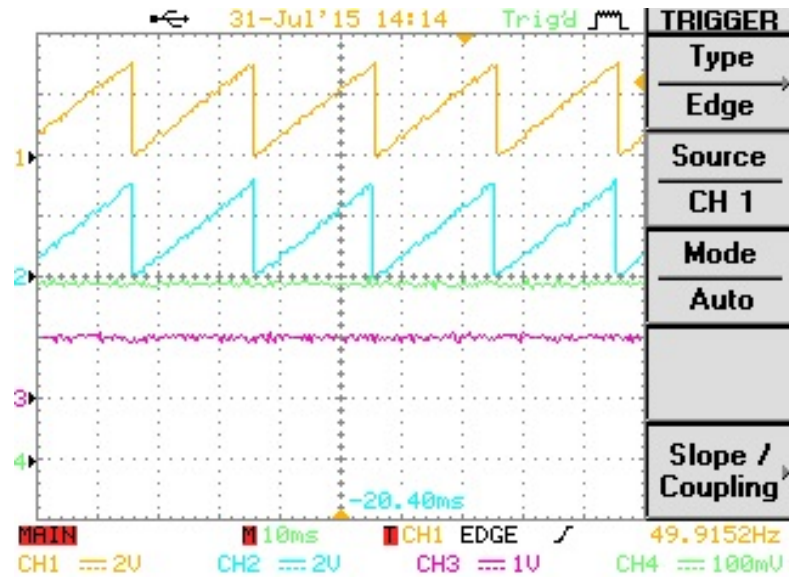


Figure 5.11 – Validation motor model $f_{s_ref} = 1 p.u.$, $V_{sd}^* = 0$, $V_{sq}^* = 1$ [CH1 : $\frac{\theta}{2}$ imposed CH2 : $\frac{\theta}{2}$ estimated CH3 : $\frac{f_s}{\omega_n}$ CH4 : Ψ_r]

5.2 Closed-loop *No Load*

5.2.1 Current regulators

To check the current regulators, we set i_{sd}^* and i_{sq}^* and we implement a step of 1 A. We are going to check which currents the motor produces, looking at i_{sd} and i_{sq} generated from the dq/abc conversion block for the currents (cf. Fig.3.18). We reached a rising time of 2 ms, without any overshoot in both axes.

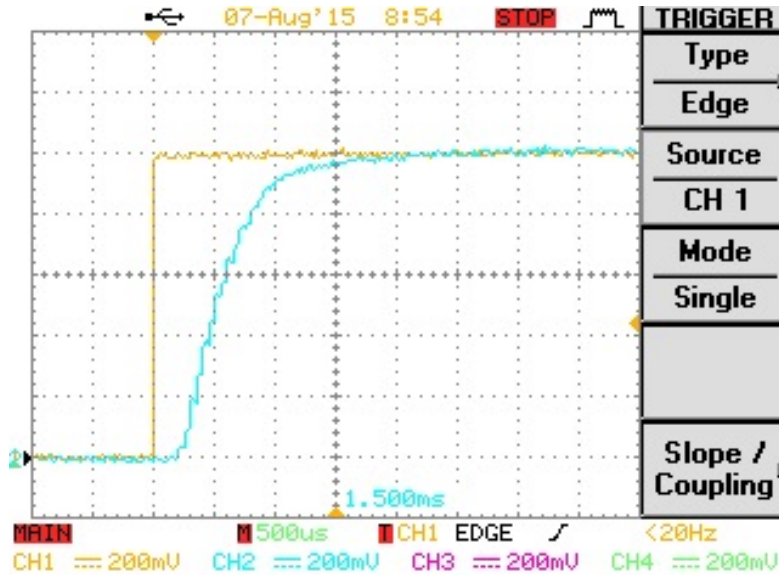


Figure 5.12 – Validation current regulator R_{id} [CH1 : i_{sd}^* CH2 : i_{sd}]

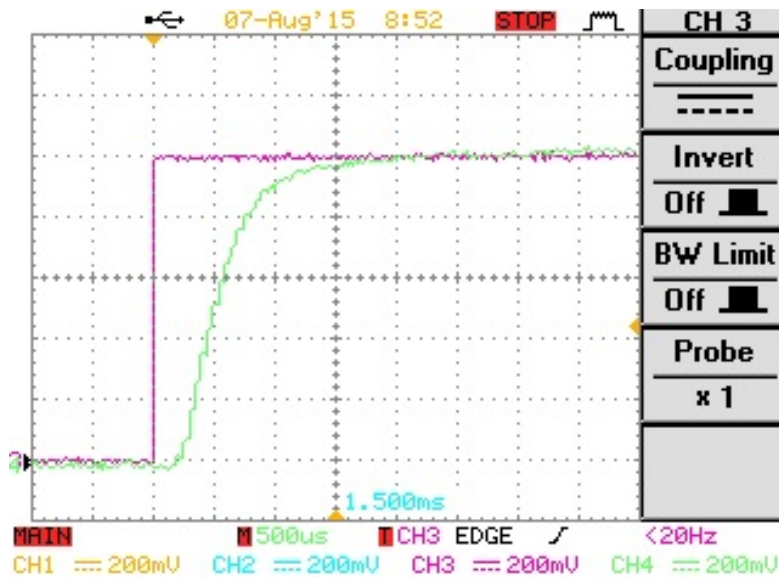


Figure 5.13 – Validation current regulator R_{iq} [CH1 : i_{sq}^* CH2 : i_{sq}]

5.2.2 Speed regulator

Thanks to the speed regulator now we can control the speed of the motor varying the speed reference ω_{ref} . We are going to show the behavior with two steps of speed reference, from 0 to half of the nominal speed (Fig.5.14) and from 0 to the nominal speed (Fig.5.15). We also represent in this figures the shape of i_{sq}^* in order to explain the resulting slopes, caused by the anti-windup method integral used in the speed regulator.

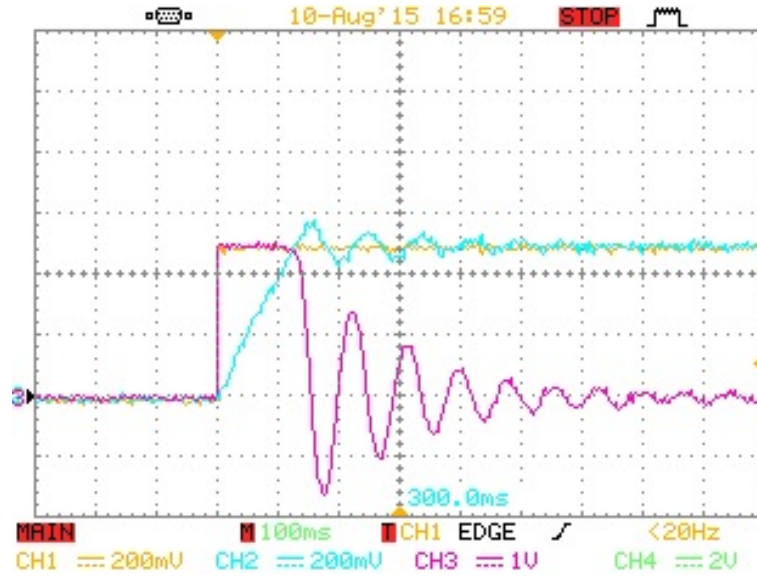


Figure 5.14 – Speed step from 0 to half of the nominal speed [CH1 : ω_{ref} CH2 : n CH3 : i_{sq}^* CH4 : i_{sq}]

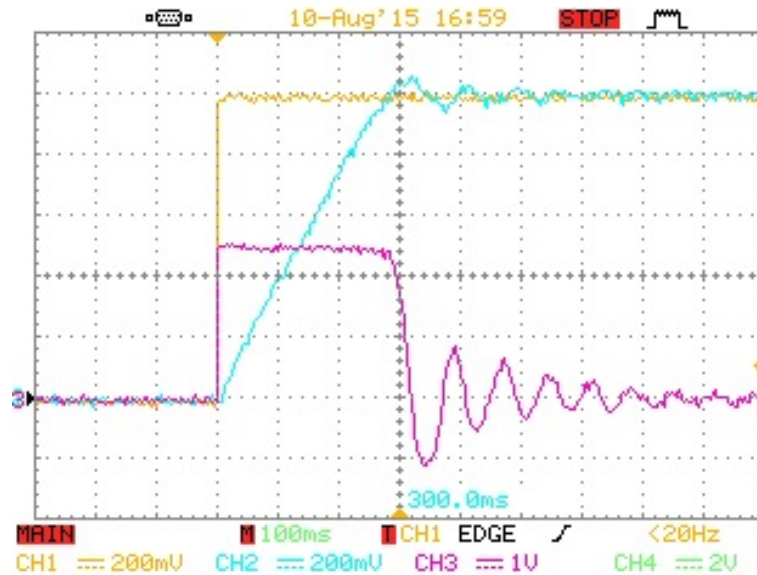


Figure 5.15 – Speed step from 0 to the nominal speed [CH1 : ω_{ref} CH2 : n CH3 : i_{sq}^* CH4 : i_{sq}]

Chapter 5. Experimental results

In Fig.5.16 is shown how the motor can follow any type of speed reference, as this one we just invented and implemented (the values in the figure are the electrical radians per second of the speed reference).

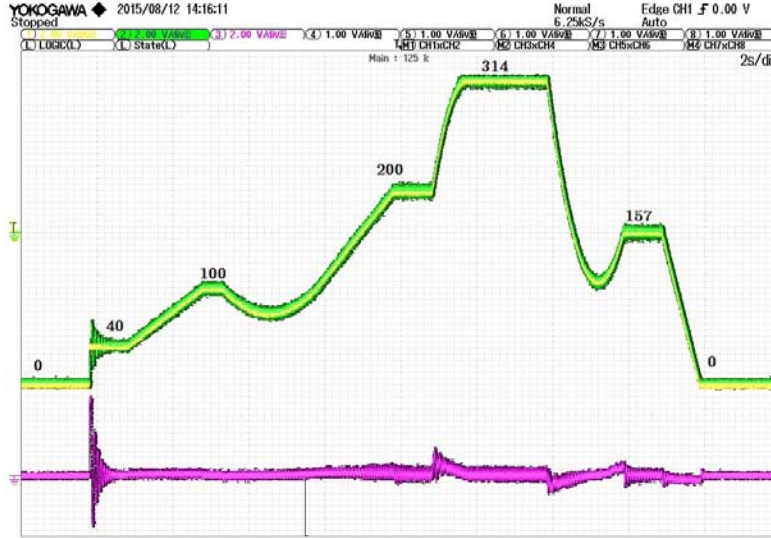


Figure 5.16 – Example of speed reference and the tracking of the motor [CH1 : ω_{ref} CH2 : n CH3 : $\frac{i_{sq}^*}{10}$]

5.3 Closed-loop Under Load

As for the *No Load* condition, we are going to show in the next figures, the behavior of the motor *Under Load*, following the speed references. Due to a hardware limit in our plant to charge the DC bus of the inverter (maximum 400 V), we couldn't supply the motor to the nominal voltage ($400V_{LL_rms}$) but the supplied voltages had an rms value around 240 V. In this operative conditions, it has decided to charge the motor with an aproximatly half nominal Load. The next results show the behavior of the control under speed steps, with a Load of $\simeq 5Nm$ (in case of $\omega_{ref} = \frac{\omega_n}{2}$ the resistance for the passive load was $R = 24.3\Omega$; in case of $\omega_{ref} = \omega_n$, $R = 45.8\Omega$). In this case, we can see that the i_{sq}^* in the steady state is not zero, but proportional to the resistance torque, and the motor needs around $1ms$ more (in comparison without Load) to reach the speed reference but with half overshoot approximately.

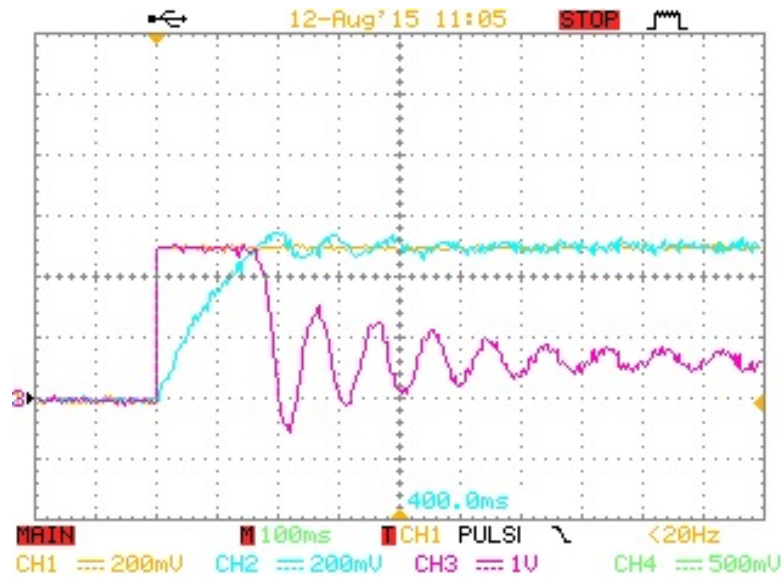


Figure 5.17 – Speed step from 0 to half of the nominal speed under load[CH1 : ω_{ref} CH2 : n CH3 : $\frac{i_{sq}^*}{10}$]

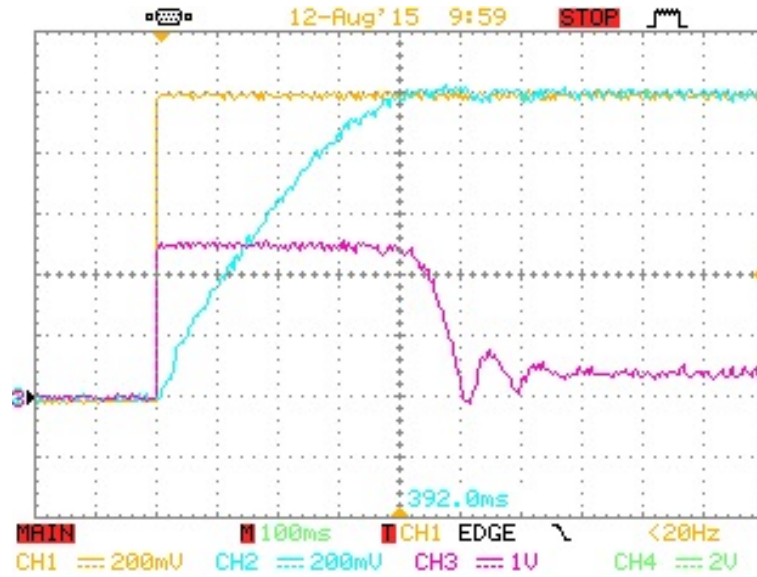


Figure 5.18 – Speed step from 0 to the nominal speed under load[CH1 : ω_{ref} CH2 : n CH3 : $\frac{i_{sq}^*}{10}$]

In the Fig.5.19 we applied a Load step with the rotor at the nominal speed, from $0Nm$ to $5Nm$, in order to see how the speed is perturbed. We can see that the FOC control works fine with Load steps, keeping the speed to the reference without big overshoots.

As in Fig.5.16, we show the behavior of the control with the same speed reference, under load (Fig.5.20).

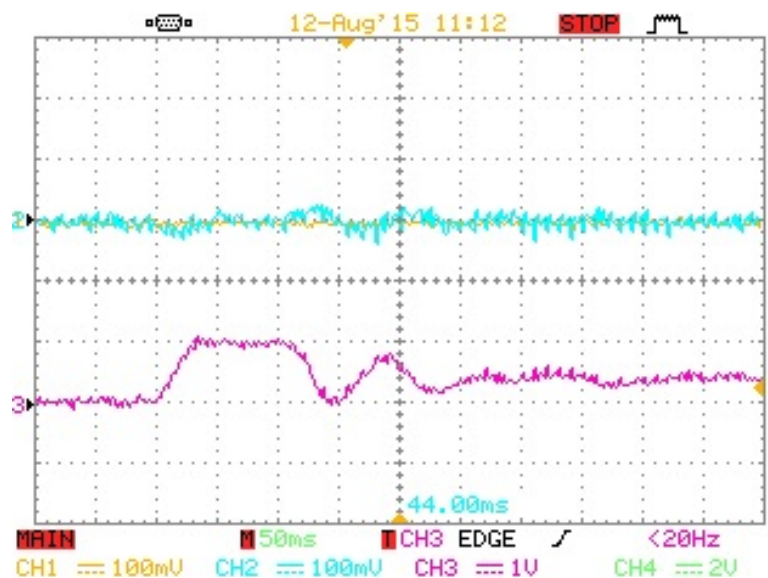


Figure 5.19 – Example of Load step (5Nm) [CH1 : ω_{ref} CH2 : n CH3 : $\frac{i_{sq}^*}{10}$]

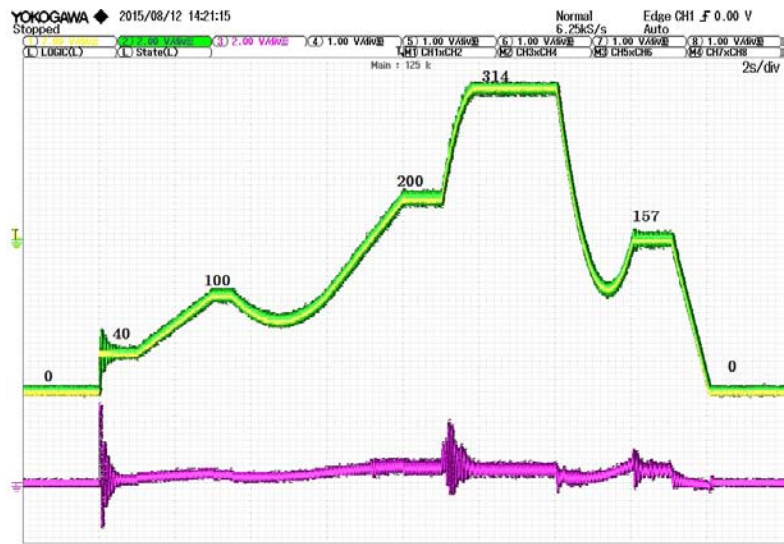


Figure 5.20 – Example of speed reference and the tracking of the motor under load[CH1 : ω_{ref} CH2 : n CH3 : $\frac{i_{sq}^*}{10}$]

6 Conclusions and future prospects

In this thesis a vector drive of an induction motor has been realized using the control platform developed by Imperix SA. The validity of the drive has been verified through simulations and by numerous experimental tests on a prototype built and assembled in the laboratory of industrial electronics (LEI) at the EPFL. Experimental results have been reported as well as simulation tests of the control, during writing of this report.

It has been seen how, thanks to the used control platform, the realization of Field Oriented vector control is not extremely difficult to implement. This allows us to easily deduce important performance considerations of the control system and the hardware; also allow us to consider the platform an excellent peripheral for R&D environments.

Primarily it was verified that the digital implementation of the control system was effective and well-functioning, leading results very similar to those obtained in simulation. It was seen that in the most strenuous condition (under load), the motor reaches the rated speed in times of the order of a few hundred milliseconds, twice the simulation but with an excellent dynamic tracking. From as what could be expected, the tracking motor following the references with damped oscillations.

The parameters of the current and speed regulators were designed with mathematical analysis, tested in simulation and then in the experimental part. For both regulators some adjustments of the parameters were necessary, that deviate it slightly from the theoretical treatment, acceptable thing, as the mathematical models are in fact models that simplify the reality and they don't consider all the involved parameters. In particular regard to the speed loop, having used a motor of small/medium power, the non-linearity behavior as a function of speed was easily observed.

Using the same parameters of the speed controller for high and low speeds, it was seen that, in one or other, some problems of mechanical noise generated by the motor itself were observed. By tuning slightly the values of the regulator we have been obtained good dynamics. However an hysteresis loop has had to implement in order to do not oscillate between two different pairs of values if the engine would operate next to the speed limit.

Chapter 6. Conclusions and future prospects

The choice of a magnetic encoder Sin/Cos has proved as successful, with a very good resolution suitable for the implementation of these types of control, and with an easy deduction of the pulsation of the primitive signal, considering the PLL technology now consolidated in the electrotechnical engineering field. The cost of this transducer is very competitive comparing with the classical incremental optical encoders, having the advantage to not have components, such as photo-transducers, delicate and easily breakable.

The increasingly implementation of sensorless drives (without using speed sensors) would suggest such an implementation for this drive, thus testing more deeply the quality of the BoomBox and the digital processing, evaluating algorithms for the rotor flux estimation. In the same way, we could implement other control techniques (*e.g.* DTC, DFOC or V/f control), test it and make comparisons with the Indirect FOC implemented in terms of performances and stress to the electrical motor.

Another point that could deepen the development of the drive, it would be the energy efficiency of the system. Having taken an inverter previously built and modeled for other applications, we can say that this one was not the best one on the market for this type of application. An advice to increase the system efficiency, in addition to size an inverter *ad hoc* for the application, can be try to find the best sampling and switching frequency, in order to minimize the semiconductor losses and at the same time, increase the injected currents and voltage quality.

A Flow chart of the Algorithm

We report a scheme of the implemented software, to better understand how it has been written. The software consider four USER_STATES in order to operate in different ways. The IDLE mode is just an *off operation*, without active PWMs. In order to pass to the OPENLOOP mode, we implemented a RAMPSTART mode, which starts the motor from 0 to the nominal speed at 50Hz, increasing proportionally the stator voltage. In OPENLOOP mode, we can change the frequency and the stator voltage without steps. For each new control frequency the user inserts, we build a ramp mode to pass to the new frequency. The CLOSEDLOOP mode implement the Field Oriented Control. The next flow chart show the user commands to pass from one to another USER_STATE, using TeraTerm interface. The entire code was written and compiled using TI Code Composer Studio 6.0.1.

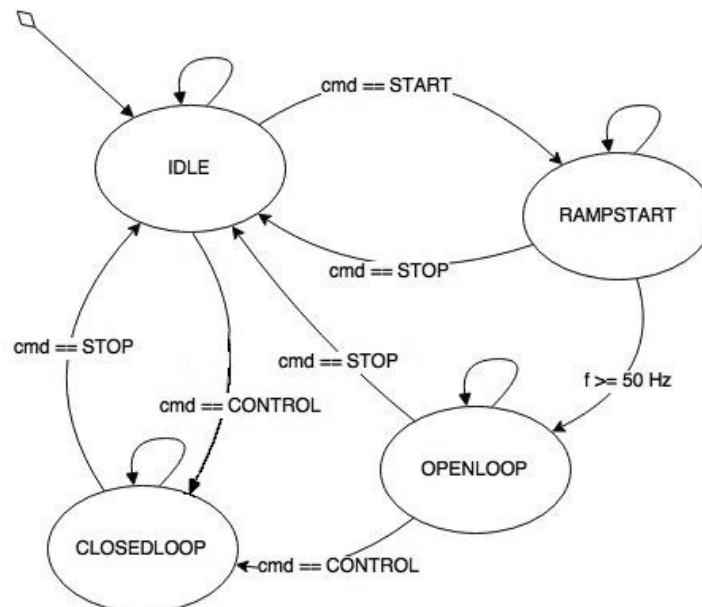


Figure A.1 – State diagram of the USER commands

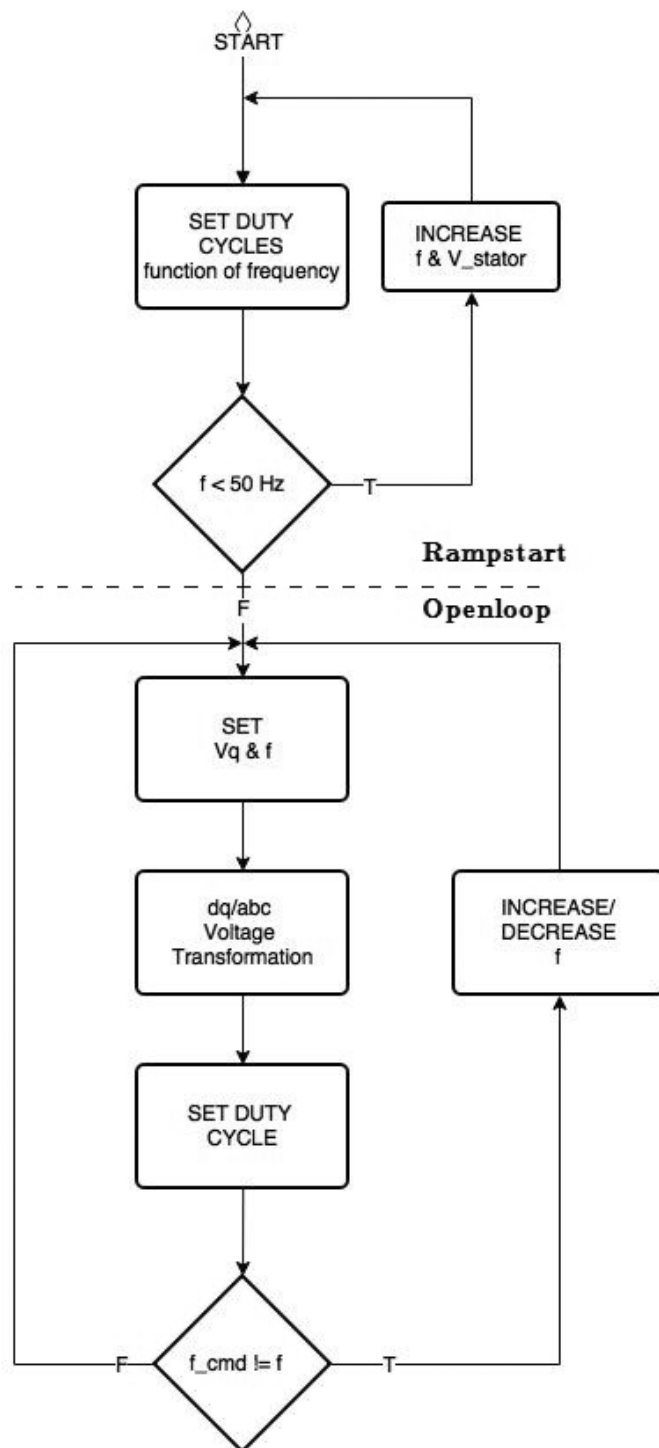


Figure A.2 – Flow Chart of RAMPSTART & OPENLOOP mode

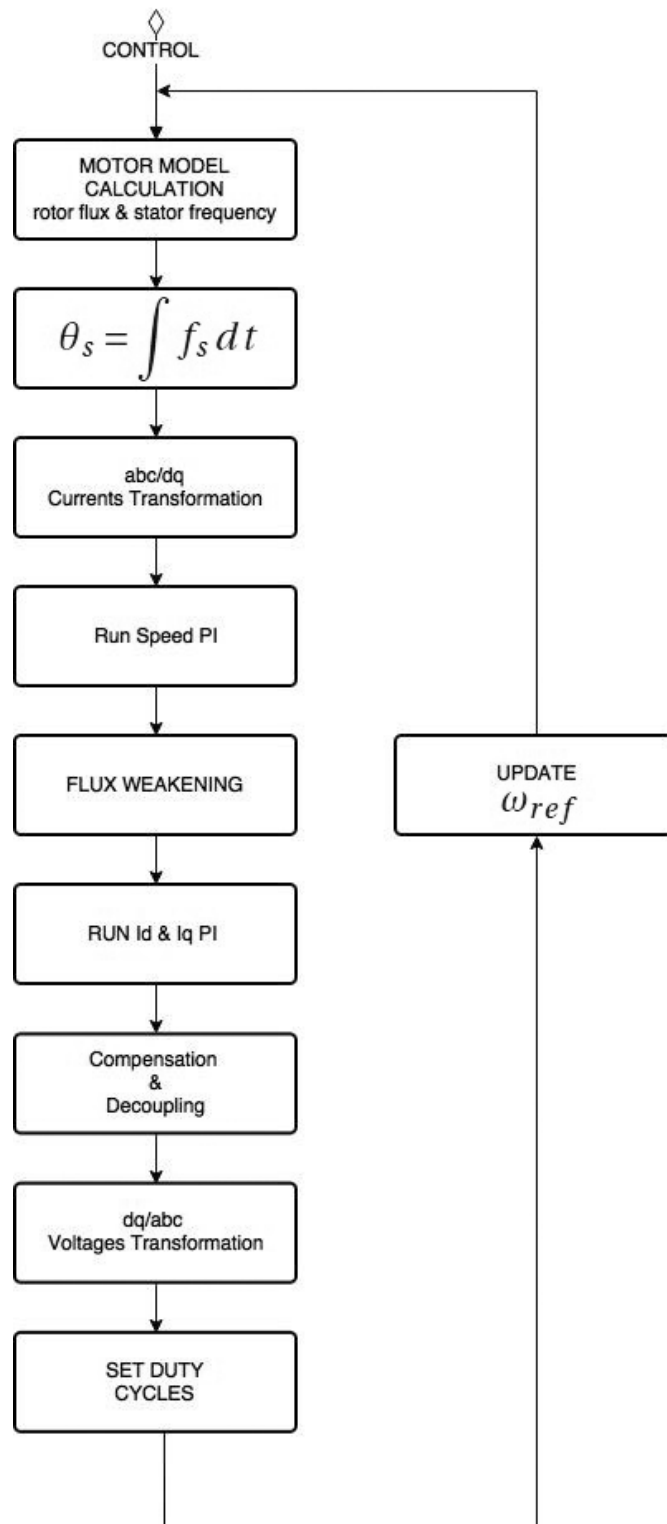


Figure A.3 – Flow Chart of CLOSEDLOOP mode

B Code of the control system

user.c

```
1/*
2 * @title    Closed-loop control of a grid-connected single-phase inverter
3 * @author   imperix SA (dev@imperix.ch)
4 * @file     user.c
5 */
6
7#include "Driver/peripherals.h"           // Available DSP peripherals
8#include "Core/interrupts.h"            // Interrupt-related routines
9#include "user.h"                        // This file's header.
10#include <stdio.h>                       // Standard IO, typ. used for printf()
11#include <math.h>                        // Mathematical functions
12
13#include "../API/transformations.h"      // Coordinate transformations
14#include "../API/controllers.h"         // Discrete-time controllers
15#include "../API/PLLs.h"                // PLLs
16
17
18/**
19 * Global variables
20 * Please pay attention not to define variables that are unnecessarily global !
21 */
22
23USER_STATE opmode, next_opmode = IDLE; // Defines the operation mode of the converter
24
25// Variables for the control implementation
26
27TimeDomain Im_abc;                     // misured phase currents a b c
28TimeDomain V_abc_ref;                  // voltage references for the legs a b c
29SpaceVector V_dq0_ref= {0.0,0.0,0.0}; // voltage references in d q
30SpaceVector Im_dq0 = {0.0,0.0,0.0};    // misured currents in d q
31SpaceVector I_dq0_ref= {0.0,0.0,0.0};  // current references in d q
32SpaceVector V_comp= {0.0,0.0,0.0};     // compensation and decoupling voltages in d q
33SpaceVector Uenc;                      // Sin and Cos values from the encoder
34
35// Variables for the regulators
36PIDController Integration, SpeedControl, IdControl, IqControl;
37SOGIPLL1Parameters SOGIPLL;           // Pseudo-object containing the PLL parameters
38                                     //(Synchronisation with the encoder)
```

Appendix B. Code of the control system

```
39
40 float enc[2000];
41
42 float Udc_mp, Udc_mm, Udc_m; // DC bus measured voltage +(mp),-(mm) & dif.(m)
43 float duty_a,duty_b,duty_c,duty_chopper; // duty cycle phases and chopeer
44 float usc = 0.0; // stator voltage reference (value from 0 to 1)
45 float fsc_ref = 1.0; // stator frequency reference (value from 0 to 1)
46 float time = 0.0; // time variable to build some ramps
47 float A_enc, B_enc, N_enc; // three channels of encoder
48 float Omega; // shaft speed
49 double flux_rd = 0.0; // rotor flux (motor model)
50 float flux_ref; // rotor flux reference (weakening)
51 float theta = 0.0; // position (motor model)
52 double u_previous = 0.0; // used as a pointer for FirstOrderLag
53 float fs = 0.0; // stator frequency (motor model)
54 float fr = 0; // rotor frequency (motor model)
55 float w_ref = 0.0; // speed control reference
56 float fsc_control = wn; // stator frequency reference in OPENLOOP
57 float w_ref_old = 0.0, mode = 0.0; // variables hysteresis control speed regulator parameters
58 float thres_up, thres_down; // threshold hysteresis control speed regulator parameters
59
60 unsigned int k= 0; //datalogging
61
62 /**
63  * This function is called only once at startup.
64  */
65 USER_SAFE UserInit(void)
66 {
67     // Configure a 20 kHz switching frequency on clock source #0:
68     SetFreqGenPeriod(0, (int)(SWITCHING_PERIOD/FPGA_CLK_PERIOD));
69
70     // Map PWM channel #0 to clksource #0 with a triangular carrier and a 800 ns deadtime:
71     ConfigPWMChannel(0, 0, TRIANGLE, (int)(800e-9/FPGA_CLK_PERIOD)); // phase a
72     ConfigPWMChannel(1, 0, TRIANGLE, (int)(800e-9/FPGA_CLK_PERIOD)); // phase b
73     ConfigPWMChannel(2, 0, TRIANGLE, (int)(800e-9/FPGA_CLK_PERIOD)); // phase c
74     ConfigPWMChannel(3, 0, TRIANGLE, (int)(400e-9/FPGA_CLK_PERIOD)); // breaking chopper
75
76     // Channels #0 #1 #2 #3 are configured with a phase of zero, respect to the frequency generator:
77     SetPWMPHase(0, 0.0);
78     SetPWMPHase(1, 0.0);
79     SetPWMPHase(2, 0.0);
80     SetPWMPHase(3, 0.0);
81
82     // Analog input #0 to #7 with corrective gain and offsets:
83     SetADCAdjustments(0, 2*7.6393e-4, -0.015); // Ia current phase a
84     SetADCAdjustments(1, 2*7.6393e-4, -0.015); // Ib current phase b
85     SetADCAdjustments(2, 2*7.6393e-4, -0.015); // Ic current phase c
86     SetADCAdjustments(3, 2*7.3e-3, -0.45); // Udc +
87     SetADCAdjustments(4, 2*7.8e-3, 2.75); // Udc -
88     SetADCAdjustments(13, 3.678e-5, 0.0); // A channel encoder
89     SetADCAdjustments(6, 3.66e-5, 0.045); // B channel encoder
90     SetADCAdjustments(7, 3.815e-5, 0.5); // N channel encoder
91
92     // Configure UsrInt1 on the PWM channel #0:
93     RegisterExt1Interrupt(&UserInterrupt1, 0, 0.0, 0);
94
95     // Configure Timer interruption (5000*1e-6) --> 200 Hz frequency control chopper)
96     RegisterTimerInterrupt(&UserInterrupt2, 5000);
97
98     // Configure the sampling at the beginning of the period of the FreqGen #0:
99     ConfigSampling(0, 0.0);
100
101     // Initial value:
102     fr = 0;
103     thres_up = 160;
104     thres_down = 130;
105 }
```

```

106 // Configure and initialize the controllers:
107 #define Ki 1.0
108 #define Kp_speed 0.15
109 #define Ki_speed 0.003
110 #define Kpid 100
111 #define Kiid 2
112 #define Kpiq 100
113 #define Kiiq 2
114
115 ConfigPIDController(&Integration, 0.0, Ki*SAMPLING_PERIOD, 0, 1e10, -1e10, SAMPLING_PERIOD, 0);
116 ConfigPIDController(&SpeedControl, Kp_speed, Ki_speed, 0, 25, -25, SAMPLING_PERIOD, 0);
117 ConfigPIDController(&IdControl, Kpid, Kiid, 0, 450, -450, SAMPLING_PERIOD, 0);
118 ConfigPIDController(&IqControl, Kpiq, Kiiq, 0, 450, -450, SAMPLING_PERIOD, 0);
119
120 // Configure and initialize the PLL:
121 ConfigSOGIPLL1(&SOGIPLL, 3000.0, 30.0, 0.1, 1.0, SAMPLING_PERIOD);
122
123
124 // Initialize the application to its IDLE state:
125 opmode = IDLE;
126 return SAFE; // Return 0 if no error
127 }
128
129 /**
130 * First user-level interrupt service routine.
131 * This routine is synchronous to the PWM #0 period
132 */
133 USER_SAFE UserInterrupt1(void)
134 {
135 // Measure all the necessary quantities and calculate the missing one:
136 Im_abc.A = GetADC(0); // current on a phase
137 Im_abc.B = GetADC(1); // current on b phase
138 Im_abc.C = GetADC(2); // current on c phase
139 Udc_mp = GetADC(3); // DC bus voltage +
140 Udc_mm = GetADC(4); // DC bus voltage -
141 Udc_m = Udc_mp+Udc_mm; // DC bus
142 A_enc = GetADC(13); // Sin encoder
143 B_enc = GetADC(6); // Cos encoder
144 N_enc = GetADC(7); // reference encoder
145
146 // speed calculation:
147 A_enc *= 2;
148 B_enc *= 2;
149 Uenc.real = B_enc;
150 Uenc.imaginary = A_enc;
151 Omega = RunSOGIPLL1(&SOGIPLL, &Uenc, A_enc); // Run the PLL to synchronize with the encoder:
152 Omega = -Omega*p/64; // omega in electrical rad/s (nominal speed 100*pi [rad/s])
153
154 // hysteresis control for the speed regulator parameters (cf. cli_command.c):
155 if (mode == 1.0){
156 if (abs(Omega) > thres_up){
157 SpeedControl.kp = 0.25;
158 SpeedControl.ki = 0.005;
159 }
160 }
161 else if(mode == 0.0){
162 if (abs(Omega) < thres_down){
163 SpeedControl.kp = 0.15;
164 SpeedControl.ki = 0.003;
165 }
166 }
167
168 // Switch depending on the operation mode (state machine)
169 switch (opmode){
170 case IDLE: // Stay in this mode as long as the execution mode is not manually changed:
171 next_opmode = IDLE;
172 break;

```

Appendix B. Code of the control system

```
173
174     case RAMPSTART:
175
176         if (fsc_ref < 1){ // RAMPSTART starts the motor from 0 Hz to 50 Hz gradually
177
178             usc = StatorMagnitude(fsc_ref); //stator voltage magnitude proportional
179                                           //to the frequency
180
181             // Set the voltage references:
182             V_abc_ref.A = 0.5*usc*sin(2*PI*50*fsc_ref*time)+0.5;
183             V_abc_ref.B = 0.5*usc*sin(2*PI*50*fsc_ref*time - 2*PI/3)+0.5;
184             V_abc_ref.C = 0.5*usc*sin(2*PI*50*fsc_ref*time - 4*PI/3)+0.5;
185
186             // Set the duty-cycles:
187             duty_a = V_abc_ref.A;
188             duty_b = V_abc_ref.B;
189             duty_c = V_abc_ref.C;
190
191             // 5 seconds of ramp 1e-5 = (1/(5/5e-5)):
192             fsc_ref = fsc_ref + 1e-5;
193
194             // reset the time variable each period of the references:
195             if(time<1/(50*fsc_ref)){
196                 time += 5e-5;
197             }
198             else{
199                 time = 0.0;
200             }
201             next_opmode = RAMPSTART;
202         }
203
204         // change to openloop if frequency reaches 50 Hz:
205         else{
206             next_opmode = OPENLOOP;
207         }
208         break;
209
210     case OPENLOOP:
211         // if there's a frequency step control, do a ramp from 0 to 1(1 to 0),
212         // increase(decrease) in 10 seconds:
213
214         if(fsc_control/wn != fsc_ref){
215             if(fsc_control/wn < fsc_ref){
216                 fsc_ref -= 5e-6;
217             }
218             else{
219                 fsc_ref += 5e-6;
220             }
221         }
222
223         // integration of the stator frequency:
224         theta = RunIController(&Integration,fsc_ref*wn);
225
226         // stator voltage references in dq:
227         V_dq0_ref.real = 0.0;
228         V_dq0_ref.imaginary = StatorMagnitude(fsc_ref);
229
230         // conversion to abc references:
231         DQ02abc(&V_abc_ref,&V_dq0_ref,theta);
232
233         // Set the duty-cycles:
234         duty_a = V_abc_ref.A/2+0.5;
235         duty_b = V_abc_ref.B/2+0.5;
236         duty_c = V_abc_ref.C/2+0.5;
237
238         // reset the time variable each period of the references:
239         if(time<1/(50*fsc_ref)){
240             time += 5e-5;
241         }
242     }
```



```

242         else{
243             time = 5e-5;
244         }
245
246         next_opmode = OPENLOOP;
247         break;
248
249     case CLOSEDLOOP:           // Implementation of FOC vector control for the IM
250
251     // Motor Model:
252         // firstOrderLag to implement trasf. function as tf = coeff/(s+coeff):
253         flux_rd = 1e-8+Lm*firstOrderLag(Im_dq0.real, 1/Tr, SAMPLING_PERIOD, &u_previous);
254         fr = Im_dq0.imaginary*Lm/(flux_rd*Tr);
255         fs = fr +p*Omega/2;
256
257         // saturation for the maximum stator frequency:
258         if(fs > 110*PI){
259             fs = 110*PI;
260         }
261         else if (fs < -110*PI){
262             fs = -110*PI;
263         }
264         theta = RunIController(&Integration,fs);
265
266         // abc to dq0 of the measured currents:
267         abc2DQ0(&Im_dq0,&Im_abc,theta);
268
269         // Run speed loop:
270         I_dq0_ref.imaginary = RunPIController(&SpeedControl,w_ref-Omega);
271
272         // Data logging:
273         /*
274         if(time < 50){
275             time += 1e-3;
276         }
277         else{
278             if (k < 2000){
279                 enc[k] = I_dq0_ref.imaginary;//I_dq0_ref.imaginary;
280                 k += 1;
281             }
282         }*/
283
284         // Flux Weakening:
285         flux_ref = FluxWeakening(0.75, Omega);
286         I_dq0_ref.real = (flux_ref-flux_rd)/Lm;
287
288         // Id and Iq current loops:
289         V_dq0_ref.real = RunPIController(&IdControl,I_dq0_ref.real - Im_dq0.real);
290         V_dq0_ref.imaginary = RunPIController(&IqControl,I_dq0_ref.imaginary - Im_dq0.imaginary);
291
292         // Compensation and Decoupling:
293         V_comp.real = fs*sigma*Ls*Im_dq0.imaginary;
294         V_comp.imaginary = (Im_dq0.real*sigma*Ls+flux_rd*Lm/Lr)*fs;
295         V_dq0_ref.real -= V_comp.real;
296         V_dq0_ref.imaginary += V_comp.imaginary;
297
298         // Saturation for Vd and Vq voltage references:
299         if(V_dq0_ref.real > 450){
300             V_dq0_ref.real = 450;
301         }
302         else if (V_dq0_ref.real < -450){
303             V_dq0_ref.real = -450;
304         }
305         if(V_dq0_ref.imaginary > 450){
306             V_dq0_ref.imaginary = 450;
307         }
308         else if (V_dq0_ref.imaginary < -450){
309             V_dq0_ref.imaginary = -450;
310         }

```

Appendix B. Code of the control system

```
311         // dq0 to abc voltage references:
312         DQ02abc(&V_abc_ref,&V_dq0_ref,theta);
313
314         //InversePark to abc:
315         V_abc_ref.A = V_abc_ref.A*SQRT_2OVER3;
316         V_abc_ref.B = V_abc_ref.B*SQRT_2OVER3;
317         V_abc_ref.C = V_abc_ref.C*SQRT_2OVER3;
318
319         // Set the duty-cycles:
320         duty_a = V_abc_ref.A/(2*SQRT_3_OVER2*Udc_m)+0.5;
321         duty_b = V_abc_ref.B/(2*SQRT_3_OVER2*Udc_m)+0.5;
322         duty_c = V_abc_ref.C/(2*SQRT_3_OVER2*Udc_m)+0.5;
323
324         next_opmode = CLOSEDLOOP;
325         break;
326     }
327
328     // Display some quantities for debug purposes:
329     SetDACVoltage(0,w_ref/wn);
330     SetDACVoltage(1,0omega/wn);
331     SetDACVoltage(2,I_dq0_ref.imaginary/10);
332     SetDACVoltage(3,I_dq0_ref.imaginary);
333
334     // Set the duty-cycles:
335     SetPWMDutyCycle(0,duty_a);
336     SetPWMDutyCycle(1,duty_b);
337     SetPWMDutyCycle(2,duty_c);
338
339     // Update the modulators:
340     UpdatePWMData();
341
342     // Prepare for the next function call:
343     SetOpMode(next_opmode); // Switch to the next state
344     return SAFE;           // Return a SAFE status (or UNSAFE otherwise)
345 }
346
347 /**
348  * First user-level interrupt service routine.
349  * This routine is used with a relatively long interrupt period and is only
350  * responsible for executing the chopper control
351  */
352 USER_SAFE UserInterrupt2(void)
353 {
354     if (Udc_m > 450.0){
355         duty_chopper = 1;
356     }
357     else{
358         duty_chopper = 0;
359     }
360     ActivatePWMChannel(3);
361     SetPWMDutyCycle(3,duty_chopper);
362     UpdatePWMData();
363
364     return SAFE;
365 }
366
367 /**
368  * This routine aims to handle critical errors by requesting an appropriate answer at the user level.
369  * Typically, the application should be blocked and the core will wait for manual clearing of the error.
370  */
371 void UserError(ERROR_SOURCE source)
372 {
373     printf("\nUserError() has been triggered!"); // Only a warning message is displayed here
374 }
375
376 /**
377  * This routine is responsible for changing the mode of operation, when possible and applicable
378  */
```

```

379 void SetOpMode(USER_STATE newmode)
380 {
381     // Switch depending on the desired execution mode:
382     switch (newmode){
383         case IDLE:      DeactivatePWMChannel(0);
384                        DeactivatePWMChannel(1);
385                        DeactivatePWMChannel(2);
386                        ActivatePWMChannel(3);
387                        // Reset the setpoints:
388                        fsc_ref = 0.0;
389                        usc = 0.0;
390                        break;
391
392         case RAMPSTART:  ActivatePWMChannel(0);
393                        ActivatePWMChannel(1);
394                        ActivatePWMChannel(2);
395                        ActivatePWMChannel(3);
396
397                        break;
398
399         case OPENLOOP:   ActivatePWMChannel(0);
400                        ActivatePWMChannel(1);
401                        ActivatePWMChannel(2);
402                        ActivatePWMChannel(3);
403
404                        break;
405
406         case CLOSEDLOOP: ActivatePWMChannel(0);
407                        ActivatePWMChannel(1);
408                        ActivatePWMChannel(2);
409                        ActivatePWMChannel(3);
410
411                        break;
412
413     }
414
415     // Actually change the execution mode:
416     if (1){ // All transitions are allowed here
417         opmode = newmode;
418     }
419 }
420
421 double StatorMagnitude(float f)
422 {
423     // Function to calculate a stator voltage reference proportional to the
424     // frequency reference in per unit
425     return sqrt((1+pow((f*wn*Ts),2)))/(wn*Ts);
426 }
427
428 double firstOrderLag(double input, double coeff, double dT, double* state){
429     // Function to implement the discretization of a continuous time first
430     // order lag sys = coeff/(s+coeff) using the Tustin (Bilinear) transformation.
431     double num = (1/(1+2/coeff/dT)); // numerator
432     double den = (1-2/coeff/dT)*num; // denominator
433     double temp;
434     double output;
435
436     temp = input - den*(state);
437     output = num*(temp + (state));
438     state = temp;
439
440     return output;
441 }
442
443 double FluxWeakening(float NominalFlux, float Omega){
444     // Function to implement the rotor flux Weakening
445     int R;
446
447     if(Omega > 110*PI){R = 1;}
448     else{R = 0;}
449
450     return NominalFlux*(1-R*(1-100*PI/(Omega + 1e-5)));
451 }
452
453

```

Appendix B. Code of the control system

user.h

```
1 #ifndef USER_H_
2 #define USER_H_
3
4
5 #include "DSP28x_Project.h"           // Headers for C28346
6 #include "extern_user.h"             // Corresponding pre-compiled header
7
8
9 /**
10  * Pre-computed general constants
11  */
12 #define PI (3.141592654)
13 #define TWOPI (2*PI)
14 #define SQRT_2OVER3 (0.8164965809)
15 #define SQRT_3_OVER2 (0.8660254038)
16 /**
17  * Fixed parameters of the motor
18  */
19
20 #define Rs (4.6)           //stator resistance
21 #define Lds (15e-3)        //stator inductance
22
23 #define Rr (5.3)           //rotor resistance
24 #define Ldr (15e-3)        //rotor inductance
25
26 #define Lm (378e-3)        //magnetizing inductance
27
28 #define Pn (1500.0)        //nominal power
29 #define Un (400.0)         //nominal voltage supply
30 #define In (3.5)           //nominal current
31 #define fn (50.0)          //nominal frequency
32 #define p (2.0)            //number of pair of poles
33 #define J (0.0046)         //inertia of the motor
34
35 #define wn (2*PI*fn)
36 #define Ls (Lds+Lm)
37 #define Lr (Ldr+Lm)
38 #define xs (wn*Ls*In/Un)
39 #define xr (wn*Lr*In/Un)
40 #define xm (wn*Lm*In/Un)
41 #define sigma ((xs*xr-xm*xm)/(xs*xr))
42 #define xsigma (sigma*xs)
43 #define rs (Rs*In/Un)
44 #define rr (Rr*In/Un)
45 #define Tsp (sigma*xs/(wn*rs))
46 #define Trp (sigma*xr/(wn*rr))
47 #define Ts (Tsp/sigma)
48 #define Tr (Trp/sigma)
49 #define rt (rs+rr*(xm/xr)^2)
50 #define Ttp (xsigma/(wn*rt))
51
52 /**
53  * Time-related definitions
54  */
55 #define SWITCHING_FREQUENCY (20e3)           // Apparent switching frequency
56 #define SWITCHING_PERIOD (1/SWITCHING_FREQUENCY) // Switching period
57 #define SWITCHING_FREQUENCY_CHOPPER (200)           // Apparent switching frequency
58 #define SWITCHING_PERIOD_CHOPPER (1/SWITCHING_FREQUENCY_CHOPPER) // Switching period
59 #define SAMPLING_PERIOD (1/SWITCHING_FREQUENCY) // Sampling and control period (dsp)
60 #define SAMPLING_PERIOD_US (1/SWITCHING_FREQUENCY*1e6) // Same thing in microseconds...
61 #define FPGA_CLK_PERIOD (33.33e-9) // 30Mhz -> 33.3ns
62
63 /**
64  * Modes of operation for the user-level application
65  * These definitions apply to the user state machine
66  */
```

```

67 typedef enum {
68     IDLE = 0,
69     RAMPSTART = 1,
70     OPENLOOP = 2,
71     CLOSEDLOOP = 3,
72 } USER_STATE ;
73
74 /**
75  * First user-level interrupt service routine.
76  * Remember that it has the highest level of priority when mapped to Ext1, intermediate when
77  * mapped to Ext2 and lowest when mapped to the CPU Timer.
78  * @param void no parameters
79  * @return SAFE or UNSAFE depending on whether the user wants to trigger
80  * the blocking of the application or not
81  */
82 USER_SAFE UserInterrupt1(void);
83
84
85 /**
86  * First user-level interrupt service routine.
87  * Remember that it has the highest level of priority when mapped to Ext1, intermediate when
88  * mapped to Ext2 and lowest when mapped to the CPU Timer.
89  * @param void no parameters
90  * @return SAFE or UNSAFE depending on whether the user wants to trigger the
91  * blocking of the application or not
92  */
93 USER_SAFE UserInterrupt2(void);
94
95
96 /* this function calculate the stator voltage magnitude
97  * to maintain the fluxes at 1p.u. as the theory teaches.
98  *
99  */
100
101 double StatorMagnitude(float f);
102
103 double firstOrderLag(double input, double coeff, double dT, double* state);
104
105 double FluxWeakening(float NominalFlux, float Omega);
106
107 void SetOpMode(USER_STATE newmode);
108
109 #endif /* USER_H_ */

```

cli_commands.c

```
1/*
2 * @title Command Line Interface functions, user interface
3 * @author imperix SA (dev@imperix.ch)
4 * @file cli_commands.c
5 */
6
7#include "DSP28x_Project.h"           // Headers for C28346
8#include "Core/macros.h"             // Global defines and macros
9#include "Driver/cli.h"               // Command line interface interpreter
10#include "../API/transformations.h"   // Coordinate transformations
11#include "user.h"                     // User-level routines
12#include "../API/controllers.h"
13
14#include <string.h>
15#include <stdlib.h>
16#include <stdio.h>
17
18/**
19 * Prototype here each user function first!
20 */
21void DoStart(unsigned int argc, char *argv[]);
22void DoStop(unsigned int argc, char *argv[]);
23void DoControl(unsigned int argc, char *argv[]);
24void DoChangeSpeed(unsigned int argc, char *argv[]);
25
26/**
27 * Global Variables
28 */
29
30extern USER_STATE opmode;
31extern float fsc_control;
32extern float w_ref, w_ref_old, mode;
33extern PIDController SpeedControl, IdControl, IqControl;
34extern float thres_up, thres_down;
35extern float time;
36
37void LoadCLIUserFunctions(void)
38{
39    /* This function is called by the core at system startup.
40     * the user-defined functions have to be registered
41     * by the command line interpreter before they can be used.
42     *
43     * These functions will be mapped in the /user folder in
44     * the command line mode. Use 'cd user' in order to be in this
45     * folder before using these commands.
46     */
47    fs_mkcmd_user("start", DoStart);
48    fs_mkcmd_user("stop", DoStop);
49    fs_mkcmd_user("control", DoControl);
50    fs_mkcmd_user("changespeed", DoChangeSpeed);
51}
52
53
54void DoStart(unsigned int argc, char *argv[])
55{
56    SetOpMode(RAMPSTART);
57    printf("\nUser state successfully changed to CONNECTED, in few seconds at 50Hz.");
58    printf("\nThen write changespeed or control to pass on the CLOSEDLOOP");
59}
60
61
62
63void DoStop(unsigned int argc, char *argv[])
64{
65    SetOpMode(IDLE);
66    printf("\nUser state successfully changed to SHUTDOWN");
67}
68
```

```

69 void DoControl(unsigned int argc, char *argv[])
70 {
71     SetOpMode(CLOSEDLOOP);
72     if (opmode == CLOSEDLOOP){
73         printf("\nUser state successfully changed to CLOSEDLOOP");
74     }
75 }
76
77 }
78 void DoChangeSpeed(unsigned int argc, char *argv[])
79 {
80     float f = 0.0;
81
82     if (argc > 1){
83         sscanf(argv[1], "%f", &f);
84         if (f > 315.0 || f < -315.0){
85             printf("\nYou have selected a speed reference out of the admissible range "
86                 "(-100*pi -- 100*pi)");
87         }
88         else{
89             if (opmode == OPENLOOP){
90                 printf("\nYou have selected a speed reference %s", argv[1]);
91                 sscanf(argv[1], "%f", &fsc_control);}
92             else if (opmode == CLOSEDLOOP){
93                 printf("\nYou have selected a speed reference %s", argv[1]);
94
95                 w_ref_old = w_ref;
96                 sscanf(argv[1], "%f", &w_ref);
97
98                 if (abs(w_ref) >= thres_down && abs(w_ref) <= thres_up &&
99                     (abs(w_ref_old) < thres_down || abs(w_ref_old) > thres_up)){
100                     if (abs(w_ref_old) < thres_down){
101                         mode = 0.0;
102                     }
103                     else{
104                         mode = 1.0;
105                     }
106                 }
107                 else if (abs(w_ref) < thres_down){
108                     mode = 0.0;
109                 }
110                 else if (abs(w_ref) > thres_up){
111                     mode = 1.0;
112                 }
113                 // reset the intergral each time change the speed reference
114                 SpeedControl.ui_prev = 0.0;
115                 IqControl.ui_prev = 0.0;
116                 IdControl.ui_prev = 0.0;
117             }
118         }
119     }
120 }

```


Bibliography

- [1] Andriollo Mauro, Martinelli Giovanni, Morini Augusto *Macchine Elettriche Rotanti*, Cortina, Padova, seconda edizione, 2009.
- [2] Silverio Bolognani *Azionamenti Elettrici* Dipartimento di ingegneria industriale, Padova.
- [3] Hansruedi Bühler *Réglage de systèmes d'électronique de puissance Volume 1* Presses polytechnique et universitaires romandes, Lausanne 1997
- [4] Hansruedi Bühler *Réglage de systèmes d'électronique de puissance Volume 2* Presses polytechnique et universitaires romandes, Lausanne 1997
- [5] General Purpose Motor catalogue (2003) - ABB Group - [Online] URL:
[http://www08.abb.com/global/scot/scot234.nsf/veritydisplay/fb35a46c42805652c1257b130056f574/\\$file/General%20Purpose%20Motor%20catalogue%20092003.pdf](http://www08.abb.com/global/scot/scot234.nsf/veritydisplay/fb35a46c42805652c1257b130056f574/$file/General%20Purpose%20Motor%20catalogue%20092003.pdf) [Last access 19.06.2015]
- [6] lv 25-p and la 25-np (2011/2012) - LEM company - [Online] URL:
<http://www.lem.com/docs/products/lv%2025-p.pdf> and <http://www.lem.com/docs/products/la%2025-np.pdf> [Last access 19.06.2015]
- [7] Baumer_ITD49H00_DS_EN (2015) - Baumer company - [Online] URL:
http://pfinder.baumer.com/pfinder_motion/downloads/Produkte/PDF/Datenblatt/Drehgeber_lagerlos/Baumer_ITD49H00_DS_EN.pdf [Last access 19.06.2015]
- [8] <http://imperix.ch/products/boombox/intro>
- [9] User Manual (2015) - Imperix Company - [Online] URL:
<http://imperix.ch/wp-content/uploads/2015/03/User-Manual.pdf> [Last access 19.06.2015]
- [10] Boost Converter with MPPT for Photovoltaic application (2015) - Imperix Company - [Online] URL:
<http://imperix.ch/wp-content/uploads/2015/06/Boost-for-PV-panel-EN.pdf> [Last access 19.06.2015]

Bibliography

- [11] Sensori e Trasduttori - Cristian Secchi - [Online] URL:
<http://www.automazione.ingre.unimore.it/pages/corsi/materiale didattico/ITSC0809/ITSC09-Sensori.pdf> [Last access 19.06.2015]
- [12] Mihai Ciobotaru, Remus Teodorescu and Frede Blaabjerg *A New Single-Phase PLL Structure Based on Second Order Generalized Integrator* Aalborg University Institute of Energy Technology Pontoppidanstraede 101, 9220 Aalborg DENMARK, DOI: 10.1109/PESC.2006.1711988 Conference: Power Electronics Specialists Conference, 2006. PESC '06. 37th IEEE Source: IEEE Xplore
- [13] Abhijit Kulkarni, Student Member, IEEE, and Vinod John, Senior Member, IEEE *Analysis of Bandwidth–Unit-Vector-Distortion Tradeoff in PLL During Abnormal Grid Conditions* IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 60, NO. 12, DECEMBER 2013
- [14] Infineon-50GB170DN2-DS-v01_01-en_de Infineon - [Online] URL:
<http://www.infineon.com/dgdl/50gb170dn2.pdf?folderId=db3a304412b407950112b4095b0601e3&fileId=db3a304412b407950112b42fce1f4da0> [Last access 01.08.2015]