UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITA' DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale DII

Corso di Laurea Magistrale in Ingegneria dell'Energia Elettrica

A novel predictive control for AC electric drives with integral action

Relatore: Prof. Silverio Bolognani

Correlatore: Ing. Francesco Toso

Andrea Favato 1154970

Anno Accademico 2017/2018

Abstract

This work deals with a novel robust current control scheme for Synchronous Motor (SM) drives, based on the Model Predictive Control (MPC) theory augmented with an Integral Action. The increased robustness is achieved through the integrator, nevertheless the same effective MPC quadratic formulation is hold. The integral action, within the prediction step of MPC, improves drastically the local accuracy of the nominal model prediction with further advantages in terms steady-state error for reference tracking problem. Holding all the model predictive control advantages, the effectiveness of the proposed control is verified by mean of simulations and test-bed experiments.

Abstract

La tesi è stata svolta presso il Laboratorio di Azionamenti Elettrici EDLab, e si è incentrata sullo studio e l'implementazione di un innovativo controllo predittivo basato sul modello, in inglese indicato come Model Predictive Control (MPC), che include un'azione integrale per ottenere un errore nullo a regime tra riferimento e misura. Tale controllo, denominato I-MPC (Integral MPC) è risultato essere per la prima volta applicato nell'ambito degli azionamenti elettrici. L'algoritmo è stato pensato per un controllo di corrente di un motore sincrono trifase brushless sinusoidale a magneti permanenti. Questa tipologia di motore è di particolare interesse non solo per l'attuale ricerca ma anche per il mercato degli azionamenti industriali. Ciò è dovuto ad una serie di vantaggi rispetto ai motori a rotore avvolto, tra tutti la maggiore efficienza grazie all'assenza di perdite Joule di rotore. L'obiettivo principale dell'elaborato è stato quello di evidenziare il confronto tra il tradizionale approccio del MPC e quello invece innovativo per gli azionamenti proposto. L'intenzione è stata quella di mostrare i vantaggi dell'azione integrale del controllo, specialmente nel caso di variazioni dei parametri del motore. Infatti, con l'utilizzo del MPC si ottiene un'azione di tipo proporzionale di correzione dell'errore, e se i parametri utilizzati dal controllo non coincidono con quelli reali del motore, appare un errore a regime. Si sono quindi effettuate delle prove sperimentali in laboratorio su un motore a riluttanza (SyRM) per ottenere una evidenza dell'azione integrale da parte dell'algoritmo proposto. Al fine di avere una panoramica dello stato dell'arte in questo argomento, è stato condotto preliminarmente una ricerca di cosa attualmente viene proposto in ambito scientifico per gli azionamenti elettrici. Ne è emerso che la tecnica principale per ovviare ai mismatch parametrici consiste nell'affiancare al controllore MPC un osservatore che dalle misurazioni stimi le variazioni dei parametri. Gli obiettivi raggiunti sono stati messi in evidenza sia dalle simulazioni che dalle prove sperimentali. Esse sono state condotte considerando i punti di lavoro ideali in MTPA (Max Torque Per Amps) del motore sincrono a riluttanza. L'obiettivo principale raggiunto è stato quindi quello di presentare un innovativo controllo predittivo per gli azionamenti elettrici dei motori in alternata. I vantaggi principali riscontrati sono che il costo computazionale rispetto ad un tradizionale MPC rimane invariato, ma rispetto alle ricerche proposte al giorno d'oggi in ambito degli azionamenti, l'algoritmo presentato non necessita di alcun osservatore per stimare le variazioni dei parametri o la presenza di disturbi, risultando robusto ai diversi mismatch che si possono presentare durante il funzionamento del motore.

Contents

List of Figures

Chapter 1

Model Predictive Control: principles and electric drives applications

1.1 Introduction to the MPC

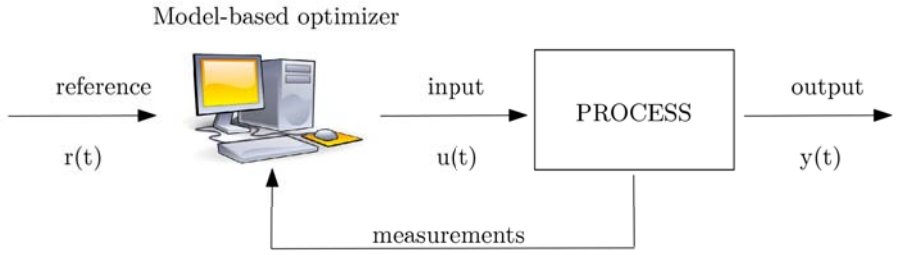


Figure 1.1: Scheme of MPC applied to a process.

Model predictive control (MPC) or receding horizon control (RHC) is a form of control in which the current control action is obtained by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first in this sequence is applied to the plant. This is its main difference from conventional control which uses a pre-computed control law [1]. MPC use a dynamical model of the process to predict its future evolution and optimize the control signal u(t). Due to its superior performances for multi-variable and strongly coupled systems, MPC algorithms are applied since the 1970s, especially within chemical engineering. However, due to the typically high computational demands, the application traditionally was limited to rather slow systems.

In general, two main categories of MPC can be found. One refers to the regulation problem, in which the control input is optimally computed in order to bring the state of the process to the origin. On the other hand it is considered the tracking problem, whose principle is showed in Figure 1.1, and it is the problem that is in the area of interest in electric drives. Basically, it is given a reference signal

and, using measurements from the process, the optimizer predicts over an horizon the optimal control input that permits to the output of the process to track the reference.

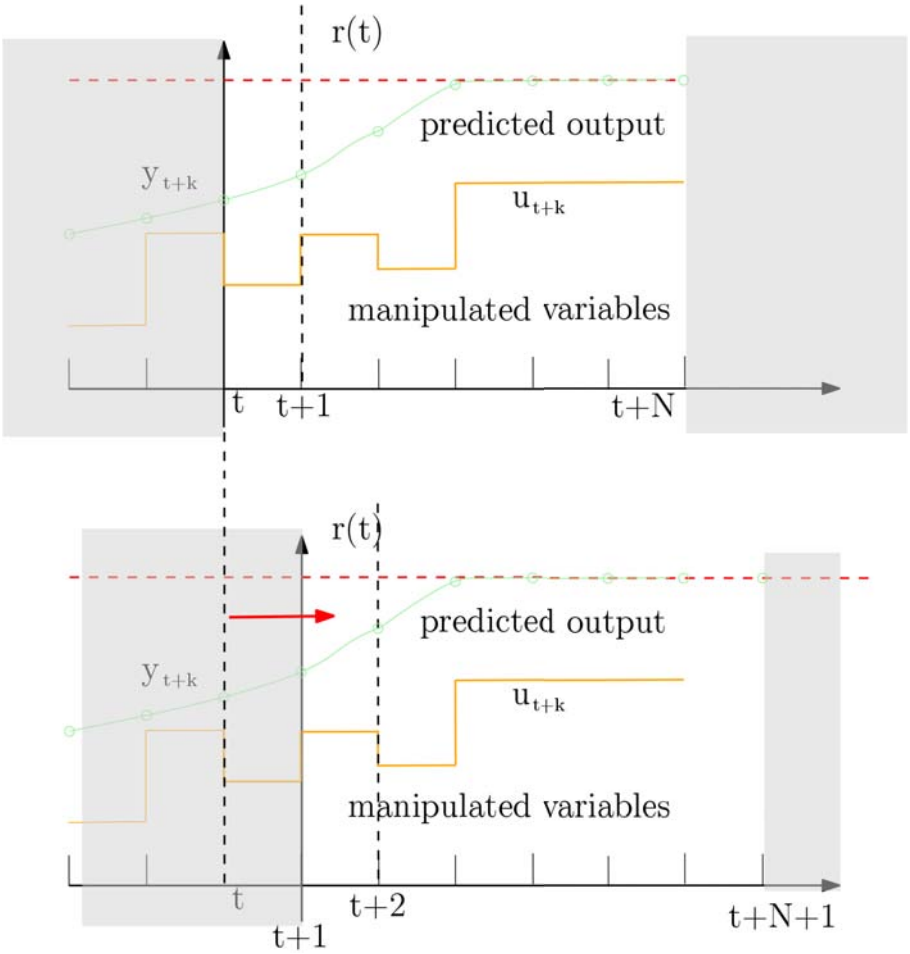


Figure 1.2: Principle of how MPC works

The ideas of how the control works, appearing in greater or lesser degree in the predictive control family, are basically:

- Explicit use of a model to predict the process output at future time instants (horizon);
- Calculation of a control sequence minimizing an objective function;
- Receding strategy, so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step [2].

The methodology of all the controllers belonging to the MPC family is characterized by the following strategy, represented in Figure 1.2.

1. The future outputs for a fixed horizon N, are predicted at each instant using the model equations. These predicted outputs $y(t+k\,t)^1, k = 0...N-1$

[1]Expression $y(t+k\,t)$ means the output prediction y at step t+k, computed at step t

depend on the known values up to instant the t and on the future inputs $u(t+k\,t), k = 0...N-1$, which are those to be deliver to the system and calculated.

2. By optimizing a determined cost function to maintain the system as close as possible to the reference trajectory, the set of future control signals is calculated. This criterion usually takes the form of a quadratic function of the errors between the predicted output signal and the predicted reference trajectory. An explicit solution can be obtained if the criterion is quadratic, the model is linear, and there are no constraints; otherwise an iterative optimization method has to be used.

3. The optimal solution $u(t\,t)$ is sent to the system whilst the next control signals calculated are discarded, because at the next sampling instant y(t+1) will start from the actual measurements as initial condition. Thus, the $u(t+1\,t+1)$ is calculated (which in principle will be different from the $u(t+1\,t)$ because of the new information available) using the *receding horizon concept.*

In the case under analysis the MPC tracking problem is formulated using the state space equations. This mean that the system to be controlled can be well-described by the following linear discrete time model:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \quad x(0) = x_0 \\ y(k) &= Cx(k) \end{aligned}$$

where $x(k)$ $\mathbb{R}^n$ and u(k) $\mathbb{R}^m$ denote the state and control input, respectively and y(k) represents the system output. The receding horizon implementation is typically formulated by introducing the open-loop optimization problem:

$$\begin{aligned} J_{(N)}(x_0) = \min_{u(.)} \sum_{i=1}^{N-1} x^\circ(k+i) - y(k+i) \;{}^2_Q + \sum_{i=0}^{N-1} \Delta u(k+i) \;{}^2_R + \\ + \; x^\circ(k+N) - y(k+N) \;{}^2_S \end{aligned} \tag{1.1}$$

subject to:

$$\begin{aligned} & x(k+i+1) = Ax(k+i) + Bu(k+i) \\ & y(k+i) = Cx(k+i) \\ & u_{min} \leq u(k+i) \leq u_{max} \\ & y_{min} \leq y(k+i) \leq y_{max} \\ & i = 0, ..., N-1 \end{aligned} \tag{1.2}$$

Since N is assumed to be limited, we refer to the problem as finite horizon problem; otherwise, if N was infinite, we would refer to an infinite horizon problem. In this case we indicate (1.1) and (1.2) as a constrained problem. It has to be solved online,

and at each iteration the receding horizon policy proceeds by implementing only the first control $u_N^*(0\ x(k))$ to obtain $x(k+1) = Ax(k) + Bu_N^*(0\ x(k))$. The rest of the control sequence $u_N^*(i\ x(k))$ is discarded and x(k+1) is used to update the optimization problem. Solution of (1.1) subject to (1.2), involves the implementation of a Convex Quadratic Program (QP). Because of the constraints, solution of the problem does not result to be analytically determined. On the contrary, a convex optimization problem has to be solved online. For a QP many algorithms and commercial software exist. The main consequence of using QP related to the MPC implementation results in a high computational time. Usually, as presented in [3], the explicit MPC with inequality constrains requires the solution of the optimization problem off-line, by investigating all the physical cases of the system behavior, leaving only the real-time task searching for the pre-computed solution. Since the work presented focus on a novel strategy of MPC implementation, the solution with inequality constrains is not here considered and other simplified approach will be used. Summing up all the information, mains advantages of using MPC are:

- can handle nonlinear/MIMO (Multiple Inputs Multiple Outputs) dynamics;
- can enforce constraints on inputs and outputs;
- performance is optimized.

While, on the other hand, main disadvantages are:

- it requires a simplified prediction model;
- computation issues more severe than in classical linear methods.

1.2 Overview of MPC application in electric drives

In order to understand how MPC in electric drives is implemented, it results useful a general scheme of an electric drive, which is reported in Figure 1.3.

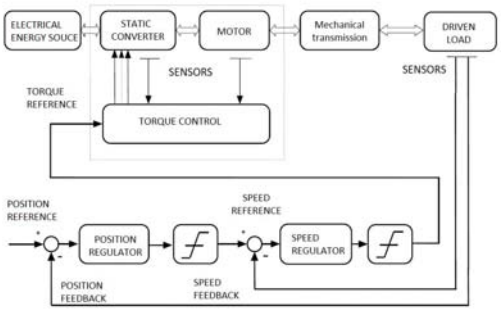


Figure 1.3: Electric drive with closed-loop control of position and speed.

An electric drive is a drive which use an electric actuator, in most cases is an electric machine, usually in motor operation, that performs a conversion from electric to mechanical energy. The motion control is obtained throughout the supply

quantities (such as voltages, currents and frequency). In order to perform the control between the motor and the source of the electric energy it is usually inserted a power device, usually a voltage source inverter, abled to provide the desired quantities. Generally this device will be a static converter, which is composed of power electronics devices. A more detailed description of the instrumentation will be included in next chapters. In Figure 1.3, it is represented a closed-loop control of the position and speed. It is also possible to have open-loop controls, but usually they are cheaper solution but less accurate and with poorer performance with respect to closed-loop solutions. This work focus on the torque control of a brushless motor. Precisely, it will be presented a current control of a synchronous reluctance motor, since the torque depends on currents. The torque reference will be a current reference, and the speed control loop will use a traditional PI (proportional Integrative) control.

The idea of MPC is derived from a rather old approach whose first ideas were published more than 30 years ago [4]. Despite its theoretical validation and consolidation along the decades, the application was usually limited to rather slow system due to the typically high computational demands. Only the recent development of hardware with higher computational powers enables its implementation in systems with fast dynamics, such as electric drives. Electric drives are of particular interest in the application of MPC, because in field as industrial robotics and automotive it is required to have rapid and precise signal tracking, such as the reference torque applied to an electric motor. Since the current dynamic is responsible for the torque dynamic and performance, a fast current-loop is required for satisfying mechanical requirements. Basically (see Figure 1.4), an MPC current control receives as inputs a current reference and measured currents at instant k, and by minimizing the cost function considering the error between reference and measure, it will predict optimal voltages over the horizon $[u*(k)u*(k+1)...u*(k+N-1)]$, but only the first sample is applied by the inverter to the three-phase motor in order to track the current reference,according to the receding horizon policy. Based on the newly measured state x(k+1), the new optimal input $\mathbf{u^*}$(k+1) is then obtained for the shifted horizon and applied, thus combining state feedback and the optimal open-loop input sequences to effectively close the control loop [3].

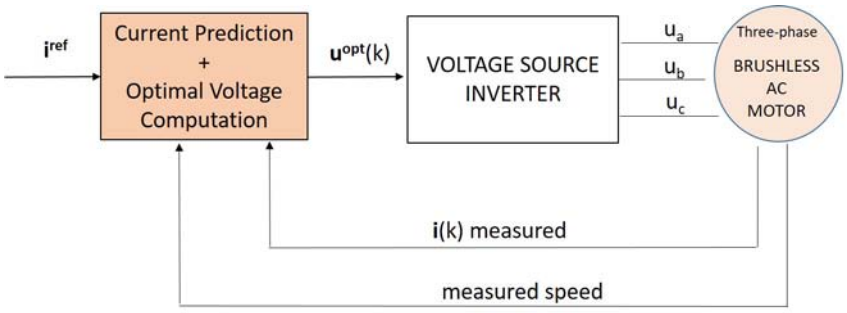


Figure 1.4: Principles of current loop MPC with VS inverter and brushless AC motor.

Model predictive control, in most cases, is based on a discrete-time model of the process. The main issue related to this approach is the knowledge of the parameters. The biggest challenge in MPC implementation is the non-linear behavior

of the drive, since the control strategy needs the full and precise knowledge of the parameters for the prediction. Without these requirements, only the MPC would cause at least a steady-state error. In literature, different approach have been used to overcome the problem of Offset-Free-Tracking. A general differentiation proposed in [5], consider two main categories for MPC applied to electric drives as shown in Figure 1.5.

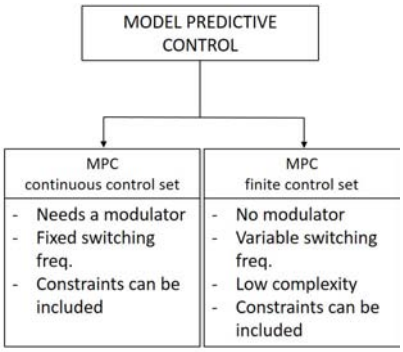


Figure 1.5: Comparison between two main MPC control strategies.

These two different approaches refer to how the inverter is controlled. MPC with finite control set, basically, solve the optimization problem comparing the six voltage vectors that the inverter can be replicated (see Chapter 3) and then it applied the best vector, whose value minimizes the cost function. On the other hand, continuous control set follows a different strategy: the voltage reference is compared with a modulator,following the Pulse-Wodth-Modulation principle that is applied to generate switching states according to the modulation principles and inverter topologies. The outputs of the inverter are voltages that are proportional to the reference one and at the same fundamental frequency. This implies that the structure of the inverter control is more complex with respect to a finite control set but it has the advantage of having a fixed switching frequency. Many applications of MPC in electric drives are performed by using Finite-Control-Set MPC (FCS-MPC) as shown in [6]. In [3], for example, a design of an MPC is applied to the control of a permanent-magnet synchronous motor (PMSM). Instead of implementing the traditional cascade structure of speed and control loops, they are combined together in a single MPC that includes all the state variables of the system. The optimization includes inequality constrains of voltages and currents, but the optimization is performed offline. The proposed solution consisted in adding an integrator of the angular speed error outside the controller, the output of which is used to move the speed reference. Other solutions to overcame model parameter mismatches include a state observer. Observability of a system consider in principle the problem of reconstructing the initial condition x_0 from n output measurements, applying a known input sequence. In [7] is proposed a robust predictive current control for PMSM based on an Extended State Observer (ESO). The ESO is able to estimate general disturbances including parameter uncertainties and sensor noise. This results in an improvement of the steady state performance of the system. However, linear observers require tuning effort and have a limited bandwidth, reducing the overall control drive capabilities.

Despite all the purposes just presented, the main idea of this work is to present a novel approach to the MPC control applied to the current loop of a three-phase brushless AC motor using a voltage source inverter. A reformulation of the state space model of the motor will introduce in order to obtain a offset-free-tracking in presence of a constant current reference, also in presence of parameter mismatches due to non-linearities of the motor, without using any state observer. From now, the new control strategy will be called as I-MPC, where the capital "I" stands for "integral", since the reformulation effect achieves an integral action on the currents. The present work is structured as follows: MPC based scheme current regulator applied to a PMSM is described; I-MPC formulation is written and compared with the standard one; Simulations and Experiments are shown and well commented; Finally the thesis ends with conclusions about the work and future develops are proposed.

Chapter 2

Derivation of the MPC and MPC with integral action

2.1 Introduction

In this chapter are reported the traditional MPC derivation and the novel formulation of the MPC with integral action over the state of the plant, given as assumption an horizon of N = 3 steps. It is important to notice that the higher N, the higher the computational cost is needed to solve the optimization problem. However, a greater value of N corresponds to a longer prediction horizon, which have positive influence on the stability of the control. On the other hand, increasing to much the horizon will not assure valuable effects, resulting in a control action with a heavy computational cost without appreciable increase in performance. Following this criterion, N=3 shows a good compromise between performance and computational cost. In this section the development of the mathematical model of the drive dynamics and the model predictive policy is derived. It will consider a general solution apply to a current control of a brushless AC motor, precisely to a Permanent Magnet Synchronous Motor (PMSM), whose current dynamics in the time domain can be expressed in the dq reference frame. In the following passages the equations are reformulated in order to obtain a linear discrete state-space model representation. Both the traditional derivation of the solution and novel strategy are presented, in order to emphasize the difference and make comparisons. Writing the linear discrete time equations of a PMSM in dq reference, it is considered the unconstrained problem, checking the solution with limits imposed by the DC-bus. The cost function is expressed as function of only the input increment ΔU, and then taking the gradient the explicit optimal solution is derived. It will be shown how the traditional MPC solution due to the minimization of the cost function, considering the receding horizon policy, has the same structure of a proportional state feedback control, while the so called integral version, or offset-free tracking version, added an integral term that is the key element in order to deal with disturbance and mismatch parameters.

The whole general idea behind the MPC is to select control actions over time, which minimize a certain objective function given to the controller, possibly adopting the best feasible pattern. Terms of the cost function are weighted throughout

a positive integer number. In principle, increasing the number weighting a term of the cost function, the algorithm will try to minimize that term more than the others. For our current control of a PMSM, the objective is to track the reference currents, therefore the error can be penalized with a discrete quadratic cost over the prediction horizon as follows:

$$J = \sum_{i=0}^{N-1} \mathbf{x}^{\circ}(k+i+1) - \mathbf{y}(k+i+1) \; {}^{2}_{M}$$

$$M = \begin{matrix} \mathbf{m_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{m_N} \end{matrix}$$

Furthermore, it is added a term to the cost function, the so called regularization term:

$$\Delta u(k+i) \; {}^{2}_{R}$$

that it is necessary to achieve, only under certain conditions, zero-offset between output and tracking reference. Indeed, an equilibrium point is set when $\mathbf{y} = \mathbf{r}$ and $\mathbf{\Delta u} = 0$. However, the main issue is related to mismatch parameters between model and plant. In cases such electric drives, where the control uses a simply and linear model in most cases for computational reasons, mismatch parameters and nonlinearities in fast dynamics appears. In this case the traditional approach doesn't guarantee zero offset tracking errors, while it will be demonstrated the simple and useful idea to add an integral state. The unconstrained problem is solved, by imposing the gradient of J equal to zero, and an explicit optimal solution results. General derivation is before considered for both cases, then the equations are recasting using a prediction horizon N = 3.

2.2 Traditional Model Predictive Control for Tracking Problem

2.3 Model state derivation of a PMSM

Consider the discrete-time, linear, time-invariant system, defined as:

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{h}(k) \\ \mathbf{y}(k) &= C\mathbf{x}(k) \end{aligned} \tag{2.1}$$

where $\mathbf{x} \subseteq \mathbb{X}^n, \mathbf{u} \subseteq \mathbb{U}^m, \mathbf{y} \subseteq \mathbb{R}^p$ are the state, input, and output vectors, while h is a disturbance, just for now it represents as an unknown disturbance. The goal is to manipulate the electrical equations of the motor in the dq reference frame in order to enquire a new set of equations that can be represented as (2.1). Consider general voltage equations in dq reference frame:

$$\begin{aligned} u_d(t) &= Ri_d(t) + L_d\frac{di_d(t)}{dt} - \omega_{me}(t)L_q i_q(t) \\ u_q(t) &= Ri_q(t) + L_q\frac{di_q(t)}{dt} - \omega_{me}(t)L_d i_d(t) + \omega_{me}(t)\Lambda_{mg} \end{aligned} \tag{2.2}$$

where:

- R is the stator phase resistance in Ω;
- L_d and L_q are the inductance of the d-axis and the q-axis respectively, in Henry [H];
- ω_{me} is the electrical angular speed measured in electric radiant per second $[rad_{el}/s]$;
- Λ_{mg} is the flux linkage of the permanent magnets placed in the rotor that are considered as constant value, measured in $V \cdot s$.

First it is necessary to discretize the equations. Forward Eulero discretization is considered. Using a sample time T_s, (2.2) can be written between two discrete time step k and k+1, such that t(k+1)-t(k) = T_s:

$$\begin{aligned} u_d(k) &= Ri_d(k) + \frac{L_d}{T_s}\big[i_d(k+1) - i_d(k)\big] - \omega_{me}(k)L_q i_q(k) \\ u_q(k) &= Ri_q(k) + \frac{L_q}{T_s}\big[i_q(k+1) - i_q(k)\big] + \omega_{me}(k)L_d i_d(k) + \omega_{me}(k)\Lambda_{mg} \end{aligned} \tag{2.3}$$

then it is possible to consider the current as state variable and the voltage as the input variable of the system, obtaining an expression that give the state at the step k+1 as function of magnitudes at the time step k:

$$i_d(k+1) = \left(1 - \frac{R}{L_d}T_s\right) i_d(k) + \omega_{me}\frac{L_q}{L_d}T_s i_q(k) + \frac{T_s}{L_d}u_d(k) \tag{2.4}$$

$$i_q(k+1) = \left(1 - \frac{R}{L_q}T_s\right) i_q(k) + \omega_{me}\frac{L_d}{L_q}T_s i_d(k) + \frac{T_s}{L_q}u_q(k) - \omega_{me}(k)\frac{\Lambda_{mg}}{L_q}T_s \tag{2.5}$$

Equations (2.4) and (2.5) can be easily recast in matrix form and the state linear discrete-time model of a PMSM results:

$$\begin{matrix} i_d(k+1) \\ i_q(k+1) \end{matrix} = A \begin{matrix} i_d(k) \\ i_q(k) \end{matrix} + B \begin{matrix} u_d(k) \\ u_q(k) \end{matrix} + \mathbf{h}(k) \tag{2.6}$$

$$\begin{matrix} y_d(k) \\ y_q(k) \end{matrix} = C \begin{matrix} i_d(k) \\ i_q(k) \end{matrix} \tag{2.7}$$

in which the matrix A,B,C and $\mathbf{h}(k)$ have the following structure:

$$A = \begin{matrix} 1 - T_s \cdot \frac{R}{Ld} & w_{me}(k) \cdot T_s \cdot \frac{Lq}{Ld} \\ -w_{me}(k) \cdot \frac{Ld}{Lq} \cdot T_s & 1 - \cdot T_s \cdot \frac{R}{Lq} \end{matrix} \quad ; \quad B = \begin{matrix} \frac{T_s}{Ld} & 0 \\ 0 & \frac{T_s}{Lq} \end{matrix} \quad ;$$

$$C = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \quad ; \quad \mathbf{h}(k) = \begin{matrix} 0 \\ -w_{me}(k) \cdot \frac{\Lambda_{mg}}{L_q} \cdot T_s \end{matrix} \quad ;$$

2.3.1 Tracking problem and MPC derivation

The objective of the model predictive control is to track a reference signal solving a quadratic optimization problem. The input voltage vector at step k can be indicated as:

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \boldsymbol{\Delta}\mathbf{u}(k)$$

where $\mathbf{u}(k-1)$ is the voltage vector applied at previous step k-1 and $\boldsymbol{\Delta}\mathbf{u}(k)$ is the optimal solution voltage vector computed at step k. In this way we can express the state equation (2.1) as function of $\boldsymbol{\Delta}\mathbf{u}(k)$ at step k+1 as follows:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{h}(k) \tag{2.8}$$

$$but \quad \mathbf{u}(k) = \mathbf{u}(k-1) + \boldsymbol{\Delta}\mathbf{u}(k) \tag{2.9}$$

The equation (2.9) is inserted in (2.8), and we get:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k-1) + B\boldsymbol{\Delta}\mathbf{u}(k) + \mathbf{h}(k) \tag{2.10}$$

Now it is possible to compute the equation (2.1) at step k+2:

$$\mathbf{x}(k+2) = A\mathbf{x}(k+1) + B\mathbf{u}(k+1) + \mathbf{h}(k+1) \tag{2.11}$$

and $\mathbf{u}(k+1)$ can be expressed as:

$$\mathbf{u}(k+1) = \mathbf{u}(k-1) + \boldsymbol{\Delta}\mathbf{u}(k) + \boldsymbol{\Delta}\mathbf{u}(k+1) \tag{2.12}$$

Substituting (2.12) in (2.11) it results:

$$\begin{aligned}\mathbf{x}(k+2) = A^2\mathbf{x}(k) + AB\mathbf{u}(k-1) + AB\boldsymbol{\Delta}\mathbf{u}(k) + \\ + A\mathbf{h}(k) + B\mathbf{u}(k-1) + B\boldsymbol{\Delta}\mathbf{u}(k) + B\boldsymbol{\Delta}\mathbf{u}(k+1) + \mathbf{h}(k+1)\end{aligned} \tag{2.13}$$

We can compute the state equation at step k+1:

$$\mathbf{x}(k+3) = A\mathbf{x}(k+2) + B\mathbf{u}(k+2) + \mathbf{h}(k+2) \tag{2.14}$$

$\mathbf{u}(k+2)$ assumes the following form:

$$\mathbf{u}(k+2) = \mathbf{u}(k-1) + \boldsymbol{\Delta}\mathbf{u}(k) + \boldsymbol{\Delta}\mathbf{u}(k+1) + \boldsymbol{\Delta}\mathbf{u}(k+2) \tag{2.15}$$

Finally, substituting (2.15) in (2.16) it results:

$$\begin{aligned}\mathbf{x}(k+3) = A^3\mathbf{x}(k) + A^2B\mathbf{u}(k-1) + A^2B\boldsymbol{\Delta}\mathbf{u}(k) + A^2\mathbf{h}(k) + AB\mathbf{u}(k-1) + \\ + AB\boldsymbol{\Delta}\mathbf{u}(k) + AB\boldsymbol{\Delta}\mathbf{u}(k+1) + A\mathbf{h}(k+1) + B\mathbf{u}(k-1) + B\boldsymbol{\Delta}\mathbf{u}(k) + \\ + B\boldsymbol{\Delta}\mathbf{u}(k+1) + B\boldsymbol{\Delta}\mathbf{u}(k+2) + \mathbf{h}(k)(k+2)\end{aligned} \tag{2.16}$$

So, it follows at a generic step k+i:

$$\mathbf{x}(k+i) = A^i\mathbf{x}(k) + \sum_{j=0}^{i-1}\sum_{t=0}^{i-1-j} A^tB\boldsymbol{\Delta}\mathbf{u}(k+j) + \sum_{j=0}^{i-1} A^{i-j-1}\Big(B\mathbf{u}(k-1) + \mathbf{h}(k+j)\Big) \tag{2.17}$$

Observing equation (2.17), it is possible to find a generic expression that gives the state $\mathbf{x}(k+i)$ as function of the initial state $\mathbf{x}(k)$, the input at step k-1 and the following input increments. Considering equations (2.8), (2.13) and(2.16),we can rearrange them obtaining a compact matrix form:

$$X = \mathbb{A}\cdot\mathbf{x}(k) + \mathbb{B}\cdot\Delta U + \mathbb{C}\cdot U_{old} + \mathbb{D}\cdot h_1 \tag{2.18}$$

Where:

$$X = \begin{matrix}\mathbf{x}(k+1)\\ \mathbf{x}(k+2)\\ \mathbf{x}(k+3)\end{matrix}, \mathbb{A} = \begin{matrix}A\\ A^2\\ A^3\end{matrix}, \mathbb{B} = \begin{matrix}B & 0 & 0\\ AB+B & B & 0\\ A^2B+AB+B & AB+B & B\end{matrix}$$

$$\Delta U = \begin{matrix}\mathbf{\Delta u}(k)\\ \mathbf{\Delta u}(k+1)\\ \mathbf{\Delta u}(k+2)\end{matrix}, \mathbb{C} = \begin{matrix}B & 0 & 0\\ AB & B & 0\\ A^2B & AB & B\end{matrix}, \mathbb{D} = \begin{matrix}I & 0 & 0\\ A & I & 0\\ A^2 & A & I\end{matrix}$$

$$U_{old} = \begin{matrix}\mathbf{u}(k-1)\\ \mathbf{u}(k-1)\\ \mathbf{u}(k-1)\end{matrix}, h_1 = \begin{matrix}\mathbf{h}(k)\\ \mathbf{h}(k+1)\\ \mathbf{h}(k+2)\end{matrix}$$

Now this expression of the state becomes useful for solving the optimization problem: the cost function will include the state equations, and the goal is to express all the terms as function of ΔU.

Note: As can be seen from the expression of $\mathbf{h}(k)$, it is a function of the electric angular speed at the same step. Since the predicted speed for following steps is unknown at time step k, it is usually considered the approximation that it is kept constant during the prediction horizon, so we get:

$$\mathbf{h}(k) \quad \mathbf{h}(k+1) \quad \mathbf{h}(k+2)$$

This assumption is justified by the fact that the electric dynamics is faster than the mechanical one, that is related to the mechanic angular speed (and thus to the electric angular speed). This results from comparing the time constant of the two system. The mechanic time constant has expression:

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

while time constant of the electric circuits of the motor are:

$$\tau_{ed} = \frac{L_d}{R}, \quad \tau_{eq} = \frac{L_q}{R}$$

Generally, τ_{ed} and τ_{eq} are of the order of milliseconds, while τ_m is of the order of tenths of a second.

2.3.2 Cost Function and resolution of the unconstrained problem

As already mentioned the cost function to be minimized is:

$$J(\mathbf{\Delta u}) = \sum_{i=1}^{N-1} \mathbf{x}^{\circ}(k+i) - \mathbf{y}(k+i) \ {}^{2}_{Q} + \sum_{i=0}^{N-1} \mathbf{\Delta u}(k+i) \ {}^{2}_{R} + \\ + \ \mathbf{x}^{\circ}(k+N) - \mathbf{y}(k+N) \ {}^{2}_{S} \tag{2.19}$$

$$s.t. \quad \mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{h}(k) \tag{2.20}$$

$$\mathbf{y}(k) = C\mathbf{x}(k) \tag{2.21}$$

where we indicate as $\mathbf{x}^{\circ}(k+i)$ the reference output at step k+i. (2.19) and (2.20) identifies an unconstrained quadratic optimization problem. Precisely, since the state equation will be included, it represents an equality constrained problem. In this way it can be found a closed form of the solution. Each terms of (2.19) are weighted by a term called weight index. Furthermore, it can be noticed that usually the last prediction error between reference and output is weighted with a different index, whose value in general can be different with respect to other term of previous predictions. Also, (2.19) can be expressed in matrix form. Considering the following passages applied with a prediction horizon N = 3, the cost function becomes:

$$(X^{\circ} - Y)^T \mathbb{Q}(X^{\circ} - Y) + \Delta U^T \mathbb{R} \Delta U + (X^{\circ} - Y)^T \mathbb{S}(X^{\circ} - Y) \tag{2.22}$$

where we define:

$$\mathbb{Q} = \begin{bmatrix} \mathbf{q_1} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \mathbf{q_{N-1}} & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \quad \mathbb{R} = \begin{bmatrix} \mathbf{r_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{r_N} \end{bmatrix} \quad \mathbb{S} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{s_N} \end{bmatrix}$$

It is required that Q and R must be semi positive definite, while the regularization matrix R must be positive definite. Now we can substitute the equation (2.18) in (2.22):

$$\begin{gathered} (X^{\circ} - \mathbb{A}\mathbf{x}(k) - \mathbb{B}\Delta U - \mathbb{C}U_{old} - \mathbb{D}h_1)^T \mathbb{Q}(X^{\circ} - \mathbb{A}\mathbf{x}(k) - \mathbb{B}\Delta U - \mathbb{C}U_{old} - \mathbb{D}h_1) \\ + \quad \Delta U^T \mathbb{R} \Delta U \quad + \\ (X^{\circ} - \mathbb{A}\mathbf{x}(k) - \mathbb{B}\Delta U - \mathbb{C}U_{old} - \mathbb{D}h_1)^T \mathbb{S}(X^{\circ} - \mathbb{A}\mathbf{x}(k) - \mathbb{B}\Delta U - \mathbb{C}U_{old} - \mathbb{D}h_1) \end{gathered}$$

By grouping terms that don't depend on ΔU and defining:

$$\underline{\mathbb{Y}} = X^{\circ} - \mathbb{A}\mathbf{x}(k) - \mathbb{C}U_{old} - \mathbb{D}h_1 \tag{2.23}$$

it is possible to get a more compact expression:

$$J(\Delta U) = (\underline{\mathbb{Y}} - \mathbb{B}\Delta U)^T \mathbb{Q}(\underline{\mathbb{Y}} - \mathbb{B}\Delta U) + \Delta U^T \mathbb{R} \Delta U + (\underline{\mathbb{Y}} - \mathbb{B}\Delta U)^T \mathbb{S}(\underline{\mathbb{Y}} - \mathbb{B}\Delta U)$$

Now computing all the calculations we get:

$$\begin{gathered} J(\Delta U) = \underline{\mathbb{Y}}^T \mathbb{Q} \underline{\mathbb{Y}} + (\mathbb{B}\Delta U)^T \mathbb{Q} \mathbb{B} \Delta U - 2\underline{\mathbb{Y}}^T \mathbb{Q} \mathbb{B} \Delta U + \\ + \Delta U^T \mathbb{R} \Delta U + \\ + \underline{\mathbb{Y}}^T \mathbb{S} \underline{\mathbb{Y}} + (\mathbb{B}\Delta U)^T \mathbb{S} \mathbb{B} \Delta U - 2\underline{\mathbb{Y}}^T \mathbb{S} \mathbb{B} \Delta U \end{gathered}$$

Terms that don't depend on ΔU can be neglected, and a compact expression results:

$$\min_{\Delta U} J(\Delta U) = \frac{1}{2}\Delta U^T H \Delta U - d^T \Delta U$$

with:

$$H = 2(\mathbb{B}^T \mathbb{Q} \mathbb{B} + \mathbb{R} + \mathbb{B}^T \mathbb{S} \mathbb{B})$$

$$d^T = 2(\underline{\mathbb{Y}}^T \mathbb{Q} \mathbb{B} + \underline{\mathbb{Y}}^T \mathbb{S} \mathbb{B})$$

We obtain a regular linear-quadratic expression for our penalization function depending only on the increment control input ΔU. Now the minimization of the function J becomes straight-forward by imposing the gradient to zero, in fact:

$$\frac{dJ}{d(\Delta U)} = 0 \qquad \Delta U_{opt} = H^{-1} \cdot d \tag{2.24}$$

So, taking the gradient of J with respect to ΔU we finally find:

$$J = H\Delta U - d = 0 \tag{2.25}$$

$$\Delta U = \Delta U_{opt} = \begin{matrix} \mathbf{\Delta u}(k) \\ \mathbf{\Delta u}(k+1) \\ \mathbf{\Delta u}(k+2) \\ \\ opt \end{matrix} = H^{-1} d \tag{2.26}$$

It is appropriate to recall the receding horizon concept that has to be applied in order to get a closed loop control. At any time k the control solves the optimization problem over the prediction horizon [k,k+N] and applies only the first input $\mathbf{\Delta u}(k)$ of the optimal sequence ΔU. At time k+1 it repeats the optimization over the prediction horizon [k,k+N+1]. In the unconstrained case the optimal increment coincides with the first element of the open-loop solution. The control law at time step k that results is:

$$\mathbf{\Delta u}(k)_{opt} = [I \quad 0 \quad 0] H^{-1} d \tag{2.27}$$

With the aim to show the properties of traditional MPC current control, we can rewrite (2.27) developing d and $\underline{\mathbb{Y}}$ terms recalling the relation (2.23):

$$\begin{aligned} \mathbf{\Delta u}(k)_{opt} = [I \quad 0 \quad 0] H^{-1} \; & (X^\circ - \mathbb{A}\mathbf{x}(k) - \mathbb{C}U_{old} - \mathbb{D}h_1)^T \mathbb{Q}\mathbb{B} + \\ & (X^\circ - \mathbb{A}\mathbf{x}(k) - \mathbb{C}U_{old} - \mathbb{D}h_1)^T \mathbb{S}\mathbb{B} \end{aligned}^T \tag{2.28}$$

The relation (2.28) clearly points out that the receding horizon applied to an unconstrained MPC results in a linear state feedback control. The principle algorithm of the MPC is shown in Figure 2.1. The considered process is a Synchronous Motor, and the control gives to the inverter the optimal voltage reference. The MPC control law has been implemented in Matlab® Simulink and it is reported in Appendix A

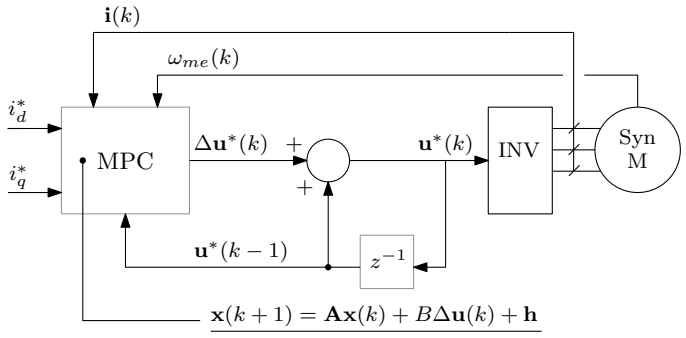


Figure 2.1: MPC control scheme.

2.4 State Model in integral form of a PMSM

Now it is shown the idea to represent the state model with an alternative expression. The proposed formulation is a novelty in electric drives field. According to this approach, a different formulation of motor state equations is required. *It is worth pointing out that the new formulation considers not only input voltage variation but also state increment.* Given N = 3 we start from equations (2.4) which here is reported:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{h}(k)$$
$$\mathbf{y}(k) = C\mathbf{x}(k)$$

with:

$$A = \begin{matrix} 1 - T_s\frac{R}{Ld} & w_{me}T_s\frac{Lq}{Ld} \\ -w_{me}\frac{Ld}{Lq}T_s & 1 - T_s\frac{R}{Lq} \end{matrix} \quad B = \begin{matrix} \frac{T_s}{Ld} & 0 \\ 0 & \frac{T_s}{Lq} \end{matrix} \quad C = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \quad \mathbf{h}(k) = \begin{matrix} 0 \\ -w_{me}\frac{\Lambda_{mg}}{Lq}T_s \end{matrix}$$

Starting from time step k, we can write the state equation at k and k+1 step as:

$$\mathbf{x}(k) = A\mathbf{x}(k-1) + B\mathbf{u} - \mathbf{1}(k) + \mathbf{h}(k-1) \tag{2.29}$$
$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{h}(k) \tag{2.30}$$

Subtracting (2.29) from (2.30):

$$\mathbf{x}(k+1)-\mathbf{x}(k) = A\Big(\mathbf{x}(k)-\mathbf{x}(k-1)\Big)+B\Big(\mathbf{u}(k)-\mathbf{u}(k-1)\Big)+\mathbf{h}(k)-\mathbf{h}(k-1) \tag{2.31}$$

By setting:

$$\begin{aligned} \mathbf{x}(k+1) - \mathbf{x}(k) &= \mathbf{\Delta x}(k+1) \\ \mathbf{x}(k) - \mathbf{x}(k-1) &= \mathbf{\Delta x}(k) \\ \mathbf{u}(k) - \mathbf{u}(k-1) &= \mathbf{\Delta u}(k) \\ \mathbf{h}(k) - \mathbf{h}(k-1) &= \mathbf{\Delta h}(k) \end{aligned} \tag{2.32}$$

we get:

$$\mathbf{\Delta x}(k+1) = A\mathbf{\Delta x}(k) + B\mathbf{\Delta u}(k) + \mathbf{\Delta h}(k) \tag{2.33}$$

Consider now the state equation at k+2 step:

$$\mathbf{x}(k+2) = A\mathbf{x}(k+1) + B\mathbf{u}(k+1) + \mathbf{h}(k+1)$$

We proceed as made before, subtracting (2.30), and we get

$$\mathbf{x}(k+2) - \mathbf{x}(k+1) = A\Big(\mathbf{x}(k+1) - \mathbf{x}(k)\Big) + B\Big(\mathbf{u} + \mathbf{1}(k) - \mathbf{u}(k)\Big) + \mathbf{h}(k+1) - \mathbf{h}(k) \tag{2.34}$$

With the same principle used in (2.32), (2.34) becomes:

$$\mathbf{\Delta x}(k+2) = A\mathbf{\Delta x}(k+1) + B\mathbf{\Delta u}(k+1) + \mathbf{\Delta h}(k+1) \tag{2.35}$$

Now we can substitute the expression of $\mathbf{\Delta x}(k+1)$ (2.33) in (2.35), so we can found the state increment at step k+2 as function of the step increment at the initial step k:

$$\mathbf{\Delta x}(k+2) = A^2\mathbf{\Delta x}(k) + AB\mathbf{\Delta u}(k) + A\mathbf{\Delta h}(k) + B\mathbf{\Delta u}(k+1) + \mathbf{\Delta h}(k+1) \tag{2.36}$$

Repeating the same procedure just presented we can compute the expression of the state increment at step k+3, that becomes:

$$\begin{aligned}\mathbf{\Delta x}(k+2) = A^3\mathbf{\Delta x}(k) + A^2B\mathbf{\Delta u}(k) + A^2\mathbf{\Delta h}(k) + AB\mathbf{\Delta u}(k+1) + \\ + A\mathbf{\Delta h}(k+1) + B\mathbf{\Delta u}(k+2) + \mathbf{\Delta h}(k+2)\end{aligned} \tag{2.37}$$

A general formulation of the state increment at a generic k-step results:

$$\mathbf{\Delta x}(k+i) = A^i\mathbf{\Delta x}(k) + \sum_{j=0}^{i-1}\Big(A^{i-1-j}B\mathbf{\Delta u}(k+j) + A^{i-1-j}\mathbf{\Delta h}(k+j)\Big) \qquad i = 1, ..., N \tag{2.38}$$

Similar to the traditional formulation of the MPC, in this case we can rewrite the state increment at a generic step k+i as function of the initial state increment and input increments. It is important to point out that, looking at the expression of vector $\mathbf{h}(k)$, it contains the term that indicates the back emf of the motor, and the electric angular speed $\omega_{me}(k)$ appears to change at each step of the prediction. However, as already mentioned, the motor speed varies very slowly compared with current dynamics. In other words, the time constant of the mechanical system $\tau_m = \frac{J}{B}$ is much greater than the time constant of the electric circuits of the motor $\tau_{ed} = \frac{L_d}{R}, \tau_{eq} = \frac{L_q}{R}$. This implies that over the entire prediction horizon the vector $\mathbf{h}(k)$ can be assumed as constant. This assumption directly implies that:

$$\mathbf{h}(k) = \mathbf{h}(k+1) = \mathbf{h}(k+2) \qquad \mathbf{\Delta h}(k) = \mathbf{\Delta h}(k+1) = \mathbf{\Delta h}(k+j) = 0$$

Furthermore the generic expression can be reformulated in matrix form as following:

$$\Delta X = \mathbb{A}\boldsymbol{\Delta}\mathbf{x}(k) + \mathbb{B}\Delta U \tag{2.39}$$

where:

$$\Delta X = \begin{matrix} \boldsymbol{\Delta}\mathbf{x}(k+1) \\ \vdots \\ \boldsymbol{\Delta}\mathbf{x}(k+N) \end{matrix} \quad \mathbb{A} = \begin{matrix} A \\ \vdots \\ A^N \end{matrix} \quad \mathbb{B} = \begin{matrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ A^{(N-1)}B & A^{(N-2)}B & \cdots & B \end{matrix} \quad \Delta U = \begin{matrix} \boldsymbol{\Delta}\mathbf{u}(k) \\ \vdots \\ \boldsymbol{\Delta}\mathbf{u}(k+N-1) \end{matrix}$$

In our analysis with a prediction horizon N = 3 the above matrices have the following structure:

$$\Delta X = \begin{matrix} \boldsymbol{\Delta}\mathbf{x}(k+1) \\ \boldsymbol{\Delta}\mathbf{x}(k+2) \\ \boldsymbol{\Delta}\mathbf{x}(k+3) \end{matrix} \quad \mathbb{A} = \begin{matrix} A \\ A^2 \\ A^3 \end{matrix} \quad \mathbb{B} = \begin{matrix} B & 0 & 0 \\ AB & B & 0 \\ A^2B & AB & B \end{matrix} \quad \Delta U = \begin{matrix} \boldsymbol{\Delta}\mathbf{u}(k) \\ \boldsymbol{\Delta}\mathbf{u}(k+1) \\ \boldsymbol{\Delta}\mathbf{u}(k+2) \end{matrix}$$

Now it is useful represents the output $\mathbf{y}(k+i)$ as function of state increments. For the 3 horizon steps we get:

$$\begin{aligned} & \mathbf{y}(k+1) = C\mathbf{x}(k+1) \\ \textit{but} \quad & \mathbf{x}(k+1) = \mathbf{x}(k) + \boldsymbol{\Delta}\mathbf{x}(k+1) \\ & \mathbf{y}(k+1) = C\mathbf{x}(k) + C\boldsymbol{\Delta}\mathbf{x}(k+1) \end{aligned}$$

$$\begin{aligned} & \mathbf{y}(k+2) = C\mathbf{x}(k+2) \\ \textit{but} \quad & \mathbf{x}(k+2) = \mathbf{x}(k) + \boldsymbol{\Delta}\mathbf{x}(k+1) + \boldsymbol{\Delta}\mathbf{x}(k+2) \\ & \mathbf{y}(k+1) = C\mathbf{x}(k) + C\boldsymbol{\Delta}\mathbf{x}(k+1) + C\boldsymbol{\Delta}\mathbf{x}(k+2) \end{aligned}$$

$$\begin{aligned} & \mathbf{y}(k+3) = C\mathbf{x}(k+3) \\ \textit{but} \quad & \mathbf{x}(k+3) = \mathbf{x}(k) + \boldsymbol{\Delta}\mathbf{x}(k+1) + \boldsymbol{\Delta}\mathbf{x}(k+2) + \boldsymbol{\Delta}\mathbf{x}(k+3) \\ & \mathbf{y}(k+1) = C\mathbf{x}(k) + C\boldsymbol{\Delta}\mathbf{x}(k+1) + C\boldsymbol{\Delta}\mathbf{x}(k+2) + C\boldsymbol{\Delta}\mathbf{x}(k+3) \end{aligned}$$

Finally, we get the matrix form:

$$Y = X_k + \mathbb{C}\Delta X \tag{2.40}$$

where:

$$Y = \begin{matrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \\ \mathbf{y}(k+3) \end{matrix} \quad X_k = \begin{matrix} \mathbf{x}(k) \\ \mathbf{x}(k) \\ \mathbf{x}(k) \end{matrix} \quad \mathbb{C} = \begin{matrix} C & 0 & 0 \\ C & C & 0 \\ C & C & C \end{matrix}$$

In this way we are able to reformulate the output vector at each prediction step as function of the initial state at step k and the state increments.

2.4.1 Cost Function and resolution of the unconstrained problem

The cost function, as in the previous analysis, is:

$$\begin{aligned} J(\boldsymbol{\Delta}\mathbf{u}) = \sum_{i=1}^{N-1} \left\| \mathbf{x}^\circ(k+i) - \mathbf{y}(k+i) \right\|_Q^2 + \sum_{i=0}^{N-1} \left\| \boldsymbol{\Delta}\mathbf{u}(k+i) \right\|_R^2 + \\ + \left\| \mathbf{x}^\circ(k+N) - \mathbf{y}(k+N) \right\|_S^2 \end{aligned} \tag{2.41}$$

that can be rewritten in matrix form as:

$$J(\Delta U) = (X^\circ - Y)^T \mathbb{Q}(X^\circ - Y) + \Delta U^T \mathbb{R} \Delta U + (X^\circ - Y)^T \mathbb{S}(X^\circ - Y) \tag{2.42}$$

with:

$$\mathbb{Q} = \begin{matrix} \mathbf{q_1} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \mathbf{q_{N-1}} & \vdots \\ 0 & \cdots & \cdots & 0 \end{matrix} \quad \mathbb{R} = \begin{matrix} \mathbf{r_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{r_N} \end{matrix} \quad \mathbb{S} = \begin{matrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{s_N} \end{matrix}$$

The weight matrices must satisfy the same hypothesis assumed before. Next, we replace (2.40) in (2.42):

$$\begin{aligned} J(\Delta U) = (X^\circ - X_k - \mathbb{C}\Delta X)^T \mathbb{Q}(X^\circ - X_k - \mathbb{C}\Delta X) + \Delta U^T \mathbb{R} \Delta U + \\ + (X^\circ - X_k - \mathbb{C}\Delta X)^T \mathbb{S}(X^\circ - X_k - \mathbb{C}\Delta X) \end{aligned} \tag{2.43}$$

Now, substituting (2.39) in (2.43) we obtain a cost function expression in terms of ΔU:

$$\begin{aligned} J(\Delta U) = (X^\circ - X_k - \mathbb{C}\mathbb{A}\boldsymbol{\Delta}\mathbf{x}(k) - \mathbb{C}\mathbb{B}\Delta U)^T \mathbb{Q}(X^\circ - X_k - \mathbb{C}\mathbb{A}\boldsymbol{\Delta}\mathbf{x}(k) - \mathbb{C}\mathbb{B}\Delta U) + \\ + \Delta U^T \mathbb{R} \Delta U + \\ + (X^\circ - X_k - \mathbb{C}\mathbb{A}\boldsymbol{\Delta}\mathbf{x}(k) - \mathbb{C}\mathbb{B}\Delta U)^T \mathbb{S}(X^\circ - X_k - \mathbb{C}\mathbb{A}\boldsymbol{\Delta}\mathbf{x}(k) - \mathbb{C}\mathbb{B}\Delta U) \end{aligned}$$

Grouping terms that don't depend on ΔU, we define:

$$\mathbb{Y} = X^{\circ} - X_k - \mathbb{CA}\boldsymbol{\Delta}\mathbf{x}(k) \tag{2.44}$$

and therefore:

$$J(\Delta U) = (\mathbb{Y} - \mathbb{CB}\Delta U)^T \mathbb{Q}(\mathbb{Y} - \mathbb{CB}\Delta U) + \Delta U^T \mathbb{R}\Delta U + \\ +(\mathbb{Y} - \mathbb{CB}\Delta U)^T \mathbb{S}(\mathbb{Y} - \mathbb{CB}\Delta U)$$

Developing calculations and ignoring terms that don't depend on ΔU the minimization problem to be solved is:

$$\min_{\Delta U} \frac{1}{2}\Delta U^T H \Delta U - d^T \Delta U \tag{2.45}$$

where:

$$2(\mathbb{B}^T \mathbb{C}^T \mathbb{Q}\mathbb{C}\mathbb{B} + \mathbb{R} + \mathbb{B}^T \mathbb{C}^T \mathbb{S}\mathbb{C}\mathbb{B}) \tag{2.46}$$

$$d^T = 2(\mathbb{Y}^T \mathbb{Q}\mathbb{C}\mathbb{B} + \mathbb{Y}^T \mathbb{S}\mathbb{C}\mathbb{B}) \tag{2.47}$$

Now we take the gradient, and we get the optimal increment input vectors at time step k, k+1 and k+2:

$$J = H\Delta U - d = 0$$

$$\Delta U = \Delta U_{opt} = \begin{matrix} \boldsymbol{\Delta}\mathbf{u}(k) \\ \boldsymbol{\Delta}\mathbf{u}(k+1) \\ \boldsymbol{\Delta}\mathbf{u}(k+2) \end{matrix}_{opt} = H^{-1}d$$

In Figure 2.2 it is shown the scheme implementation of the I-MPC (integral MPC). The optimal input voltage is applied as reference to the inverter.

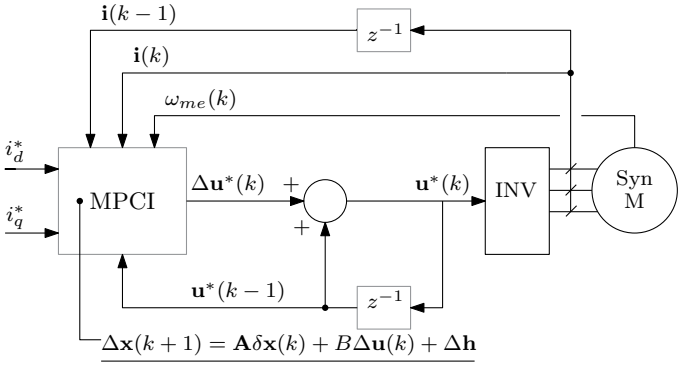


Figure 2.2: I-MPC control scheme.

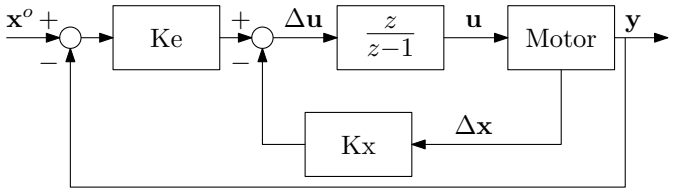


Figure 2.3: Block diagram of MPC control law with receding horizon and integral action.

As made before, we use the Receding Horizon policy using only applying only the first optimal component of the solution. With the aim to show the integral property of the MPC current control, we can rewrite (2.4.1) developing d and $\underline{\mathbb{Y}}$ terms recalling the relation (2.44):

$$\begin{aligned}\mathbf{\Delta u}(k)_{opt} = [I \quad 0 \quad 0]H^{-1}2 \ \big(X^\circ - X_k - \mathbb{CA}\mathbf{\Delta x}(k)\big)^T \mathbb{QCB} + \\ \big(X^\circ - X_k - \mathbb{CA}\mathbf{\Delta x}(k)\big)^T \mathbb{SCB}\ ^T\end{aligned} \tag{2.48}$$

Then letting:

$$\mathbf{e}(k) = X^\circ - X_k$$

the unconstrained control law takes the form:

$$\mathbf{\Delta u}(k)_{opt} = K_e\mathbf{e}(k) - K_x\mathbf{\Delta x}(k)$$

which clearly has an integral action on the error signal e(k). The relation (2.28) clearly points out that the receding horizon applied to an unconstrained MPC results in a linear state feedback control.

In Figure 2.3, it is reported the graphic representation of the MPC reformulation: It is clear that now, in steady-state conditions it is $\mathbf{\Delta x} = 0$, so that $\mathbf{\Delta u} = 0$ only for $\mathbf{e} = 0$. The MPC control law with integral action has been also implemented in Matlab® Simulink and it is reported in Appendix B.

Chapter 3

Test bench presentation

3.1 Introduction to the test bench

The aim of this work is to present a novel strategy of current Model Predictive Control with an integral action. The theory has been derived considering a Permanent Magnet Synchronous Motor using the discretized voltage equation in the d-q reference frame. The control has been implemented on a motor that present a very non-linear behavior. In fact, the control has been tested in a Synchronous Reluctance Motor (SyRM), whose characteristics are explained in this chapter. The main difference with respect a PMSM is the absence of permanent magnets in the rotor, and the electromagnetic torque , named precisely reluctance torque, is due to a particular rotor shape that present a relevant magnetic anisotropy. Tests are carried out using for the first time in EDLab the MicroLab Box, a dSpace hardware application which is a compact development system for the laboratory with high performance and versatility. It lets us to set up the control, test or measurement applications. The Matlab Simulink project of the control can be implemented by using a Real Time Interface of dSpace. Furthermore, another motor called "master" is present. It driven by its inverter, and shafts of the two motors are connected by a mechanical coupling. This allows to test the current (torque) control that we implement.

3.2 General consideration about PMSM

Permanent magnet synchronous motor (PMSM)[8], also called brushless sinusoidal, are widely used in electric drive applications, especially when high performances are required. This because they present several advantages with respect traditional synchronous motors since they don't need an auxiliary circuit that provide rotor excitation because of the presence of the magnets that are placed on the rotor, and they assure a constant magnetization. Furthermore, they don't present rotor Joule losses, and this results in an increase of the efficiency. These machines are formed by a rotor where permanent magnets(PMs) are arranged concentrically creating pole pairs (1 or more), and a stator that contains three-phase windings. Both rotor and stator are cylindrical crown shaped, and they are made of ferroelectric materials, and between them it is present an air gap.

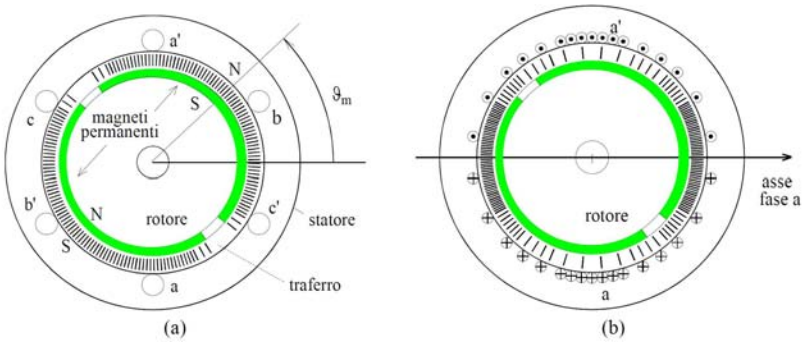


Figure 3.1: Schematic representation of a 2 poles brushless motor.

The electromechanics energy conversion that PMSMs provide are based on the interaction between conductors placed in the stator that are current paths and magnetic fields created by permanent magnets. In Figure 3.1, it is reported a schematic representation of a PMSM. In Figure 3.1a) it is pointed out the induction field at the air gap produced by rotor PMs, similar to a quasi-square wave, while in 3.1b) it is represented the sinusoidal induction field produced by the stator winding of the a phase , thanks to a proper winding arrangement along the stator. Stator contains a tree-phase winding: the three phases have the same number and distribution of conductors, but they present a relative space displacement of $\frac{2}{3}\pi$, and each phase have a couple of clamps that are useful to provide them power supply from an external three-phase source.

In order to describe the behavior of the sinusoidal PMSM we can start from the general voltage balance of the three stator phases a,b,c:

$$\begin{aligned} u_a(t) &= Ri_a(t) + \frac{d\lambda_a(t)}{dt} \\ u_b(t) &= Ri_b(t) + \frac{d\lambda_b(t)}{dt} \\ u_c(t) &= Ri_c(t) + \frac{d\lambda_c(t)}{dt} \end{aligned} \tag{3.1}$$

where i_a, i_b, i_c are the currents that flow through the three phases, $\lambda_a, \lambda_b, \lambda_c$ are linkage magnetics fluxes with each phase and R is the rotor phase resistance.

Consider hypothesis of linearity, the absence of the magnetic circuit saturation and parasitic effects. By using the Clarke and Park transformation it is possible to identifies a new reference frame, called d-q reference frame, that rotates with an angular speed which is assumed to be equal to the electrical angular speed w_{me} of the rotor. Now we can represent the three-phase stator system described by (3.1) with a space vector which has the following components with respect to d and q axis respectively:

$$\begin{aligned} u_d(t) &= Ri_d(t) + L_d\frac{di_d(t)}{dt} - \omega_{me}(t)L_q i_q(t) \\ u_q(t) &= Ri_q(t) + L_q\frac{di_q(t)}{dt} + \omega_{me}(t)L_d i_d(t) + \omega_{me}(t)\Lambda_{mg} \end{aligned} \tag{3.2}$$

that are general voltage equations in d-q reference frame have already been presented and here are reported. The term $\omega_{me}(t)$ refers to the electric angular speed, that is related to the mechanical angular speed by the pole pairs:

$$\omega_{me}(t) = p\omega_m(t)$$

L_d is the synchronous direct inductance crossed by the direct current $i_d(t)$ and L_q is the synchronous quadrature inductance crossed by $i_q(t)$, while the term $\omega_{me}(t)\Lambda_{mg}$ represents the back electromagneticforce (back-emf). From this set of equation it is possible to derive an expression of the electromagnetic torque as result of electromechanics energy conversion principle. If we feed the motor with three sinusoidal currents that are at the same angular speed with respect the back-emf and making a power balance using (3.2) multiplied by the respectively transformed currents, the electromagnetic torque assumes the following expression:

$$m(t) = \frac{3}{2}p\Lambda_{mg}i_q(t) + \frac{3}{2}p(L_d - L_q)i_d(t)i_q(t) \tag{3.3}$$

where p is the number of pole pairs. As can be seen, the torque is composed by two terms: first term due to electrodynamic interaction depends only on q current, while the second term is function of both d and q currents. Second terms is the so called reluctance torque.

Also, the mechanical equation of the motor has to be considered, which as the following expression:

$$m(t) = m_L(t) + B\omega_m(t) + J\frac{d\omega_m(t)}{dt} \tag{3.4}$$

where:

- $m_L(t)$ is a torque disturbance in Nm;
- B represents the viscous inertia coefficient expressed in Ns;
- $\omega_m(t)$ is the mechanical angular speed in rad/sec;
- J is the inertia of the mechanical parts in Kgm^2.

3.3 Synchronous Reluctance Motor

PMSM can be classified in two main categories, considering the magnetic circuit between rotor and stator of the machine:

- Isotropic motor, in which the rotor and the magnet displacement is designed in order to have the same magnetic structure with respect the stator;

- anisotropic motor, in which the rotor structure or the magnet displacement such that it results a relevant magnetic anisotropy.

In an isotropic machine, the magnetic paths along d and q axis present the same reluctance, so it holds the following relation:

$$L_d = L_q$$

so the inductances are the same along the two reference axes. In Figure 3.2, it is reported a representation of a SPM motor. Magnets are equally placed over the rotor and, since their permeability is comparable with air, the reluctance between d and q axes is the same.

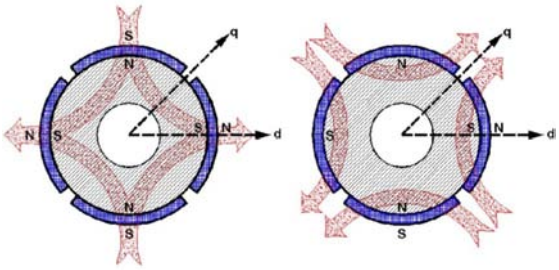


Figure 3.2: Magnetic flux paths in d and q axes of a Surface Permanent Magnet.

On the other hand, anisotropic machine present different magnetic paths between d and q axis, as it can be shown in Figure3.3 where is reported a rotor of an Interior Permanent Magnet (IPM) Motors. Magnets are placed inside the rotor, and for d-axis the inductance results in general lower than the inductance of the q-axis, due to the fact that in q-axis flux path pass through iron that has a higher value of permeability.

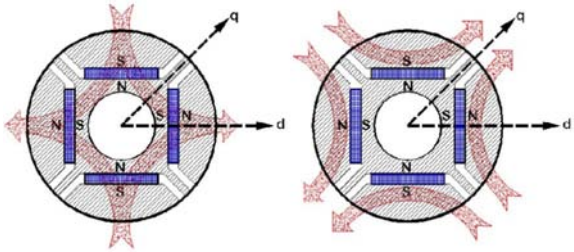


Figure 3.3: Magnetic flux paths in d and q axes of an IPM.

Synchronous Reluctance Motors are a particular anisotropic machine whose rotor doesn't present magnets. The anisotropy is obtained thanks to a proper shape design of the rotor. In Figure 3.4, it is shown an example of rotor of a SyRM with 4 poles.

Conventionally, for a SyRM it is considered as the d-axis where the reluctance is less grater, i.e where the rotor present iron path for the flux linkage. In this way for a SyRM it holds that:

$$L_d > L_q$$

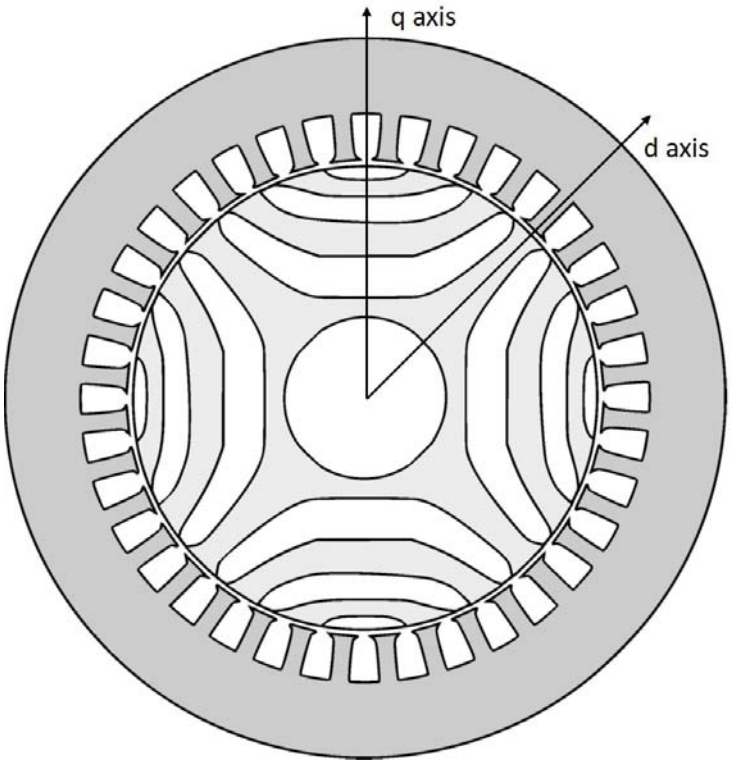


Figure 3.4: Rotor scheme of a SyRM with 4 poles (p=2).

The q axis is shifted 90 electric degree early. Electric and mechanical degrees are linked, such as the angular speed, by the number of pole pairs such that:

$$\theta_{me} = \frac{\theta_m}{p}$$

For a SyRM, due to the absence of rotor magnets, it is $\Lambda_{mg} = 0$, so the term that represents the back emf disappears and equations 3.2 becomes:

$$\begin{aligned} u_d(t) &= Ri_d(t) + L_d\frac{di_d(t)}{dt} - \omega_{me}(t)L_q i_q(t) \\ u_q(t) &= Ri_q(t) + L_q\frac{di_q(t)}{dt} + \omega_{me}(t)L_d i_d(t) \end{aligned} \tag{3.5}$$

and the torque assumes the expression of:

$$m(t) = \frac{3}{2}p(L_d - L_q)i_d(t)i_q(t) \tag{3.6}$$

One of the most important concept concerning electric drives is related to the Max Torque Per Amps (MTPA) operating region. As can be seen from the expression (3.3) in d-q reference frame, the electromagnetic torque depends on currents value. First, it can be notice that there is no torque without currents. In time domain it is possible to demonstrate that this condition can be obtained if sinusoidal currents a,b,c are in phase (considering positive torque in motor operation) or in phase opposition with respect the relative back-electro magnetic forces (considering negative torque).

Referring to the SyRM in order to identify in the voltage-current plane the MTPA conditions, it is necessary to deal with limits and operating regions of an electric drive. Voltages and currents which can be applied to the motor must stay within specified limits, whose value can be represented by nominal values of the machine or similar nominal values of the power supply system. In order to identify operating regions of a SyRM we consider the sinusoidal steady-state. This means that:

- The voltage and current of each phase is sinusoidal in time domain with amplitude and frequency constants;
- angular speed is constant, and equal to Ω;
- also the transformed electric magnitudes in a rotating reference frame synchronous with the rotor result constant, indicated as I_d, I_q, U_d, U_q.

In order to satisfy the current limitation in each stator phases it must have the root-mean-square (RMS) value lower than the nominal current I_{nom} of the motor. This value is related to the temperature of the conductors due to Joule losses above which it could compromise insulating materials.

Also the stator voltage must be limited within its nominal value U_{nom}, which depends on the electromagnetic design of the motor. Using magnitudes in d-q reference frame, two inequality must be satisfied:

$$\begin{aligned} I_d^2 + I_q^2 &\leq I_N^2 \\ U_d^2 + U_q^2 &\leq U_N^2 \end{aligned} \tag{3.7}$$

where:

- I_N is the module of the stator currents space vector corresponding to the RMS nominal phase current I_{nom}. Due to the transformation used, it holds the relation $I_N = \sqrt{2} I_{nom}$;
- U_N is the module of the stator voltages space vector corresponding to the RMS nominal line-to-line voltage U_{nom}. Due to the transformation used, it holds the relation $U_N = \frac{\sqrt{2}}{\sqrt{3}} U_{nom}$.

Consider now a SyRM in steady state operation. Assuming constant values of currents and voltages equations (3.5) becomes:

$$\begin{aligned} U_d &= RI_d - \Omega_{me} L_q I_q \\ U_q &= RI_q + \Omega_{me} L_d I_d \end{aligned} \tag{3.8}$$

Substituting (3.8) in (3.7) and ignoring the voltage drop due to resistance which is usually smaller in steady state condition than other terms, we get following expressions:

$$\begin{aligned} I_d^2 + \left(\frac{L_q}{L_d} I_q\right)^2 &\leq \frac{U_N^2}{\Omega_{me}^2 L_d^2} \\ I_d^2 + I_q^2 &\leq I_N^2 \end{aligned} \tag{3.9}$$

In the I_d-I_q plane the first equation represents family of concentric ellipses, whose axes depend on the reciprocal of the speed. Instead of the second equation describe a circle with radius equal to I_N, centered in the origin. Furthermore, torque equation (3.6) describes a family of hyperboles for different value of torque. The region just described is shown in Figure 3.5, assuming the IPM convention ($L_q > L_d$). If we assume the SyRM convention, ellipses are rotated by an angle of 90°, and the MTPA region remain unchanged because ideally it is represented by a straight line at 45° degrees in the d-q plane.

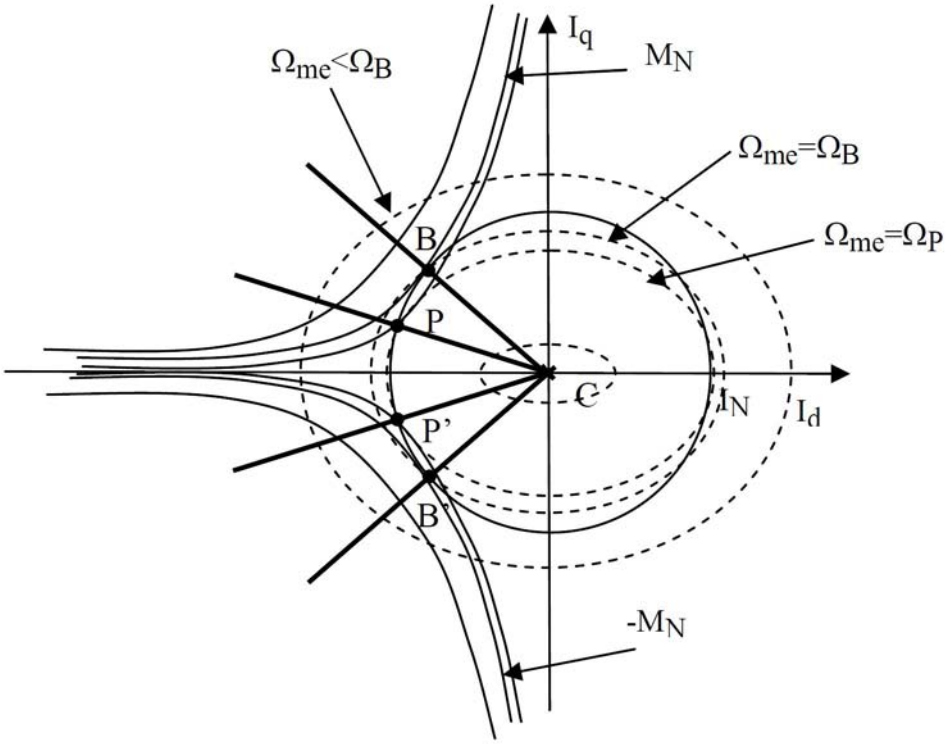


Figure 3.5: Operating regions of a SyRM (IPM convention).

The motor operation, at a given speed, in order to respect voltage and current limits corresponds to an interior point of work of both current and voltage limit. At low speed, voltage limits is very large, so in these conditions it is a good idea to use the motor in a point of the BB' region, depending on the necessary torque, but assuring for that torque the minimum current value. This region is the so called *Max Torque per Amps (MTPA)* and minimum losses are achieved. This approach holds until the BB' region remain within the voltage limit. In other word, referring to 3.5 the so called basis speed Ω_{me_B} is reached, which corresponds a point of work whose ellipse pass through B and B'. Ω_{me_B} represents the basis speed of the drive, under which it is always available the nominal torque. Exceeding that speed the point of work is provided by the intersection between the current limit circle and the ellipse of the voltage limit at that speed. For speed greater than Ω_{me_P} it is convenient to work in the PP' region, that represents the region that assures the *Maximum Torque Per Voltage (MTPV).*

Table 3.1: My caption

Parameter	Value	u.m
R	16	Ω
L_d	1	H
L_q	0.4	H
p	2	-

For a SyRM, the expression of the ideal MTPA region is a straight line with expression:

$$I_q = \pm I_d \tag{3.10}$$

with the plus sign in motor operation, while the MTPV region has the following expression:

$$I_q = \pm \frac{Ld}{Lq} I_d \tag{3.11}$$

3.3.1 SyRM test prototype

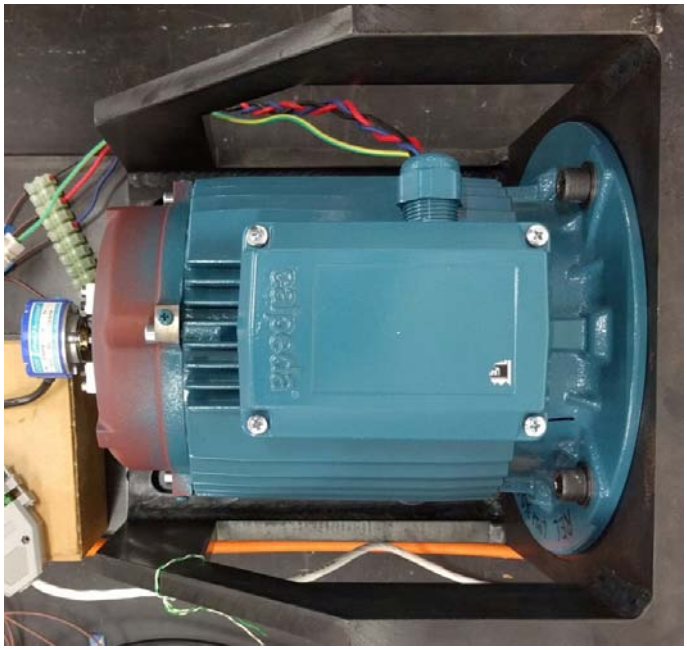


Figure 3.6: The SyRM used for experimental tests.

The SyRM used to test the MPC with integral action and to compare the performance with traditional MPC is a prototype designed in EDLab, and it is shown in Figure 3.6. On the left, it is recognized the incremental encoder, which is used to measure speed and position of the rotor. Nominal parameters are reported in Table 3.1. Since the motor is a prototype, nominal inductances are assumed to be an intermediate value, considering the range of the inductance variation in all current regime, from zero to the limit one (see Figure 3.7).

Despite the previous analysis performed for operating regions, it is important to notice that the effect of iron saturation isn't negligible, especially for a SyRM with very high value of inductances. This means that the iso-torque hyperboles and region of constant voltages differ from those previously calculated. Consequently, Figure 3.5 describes only an ideal case and has only qualitative purpose. In fact in Figure 3.7 are reported L_d and L_q for different operation conditions. Blue lines refer to L_d values, while pink ones refer to L_q values. Dashed lines result from measurements, while continuous ones results from simulation.It can be noticed that the different behaviour at low current values is due to the fact that in the real machine, the iron has a non linear B-H characteristic. So the measurements register an initial non linear trend of inductance values. This graph points out the non-linearity of the motor, because inductances halved with a range of current from zero to the nominal one. Thus, the MTPA region is not well described by the equation (3.10).

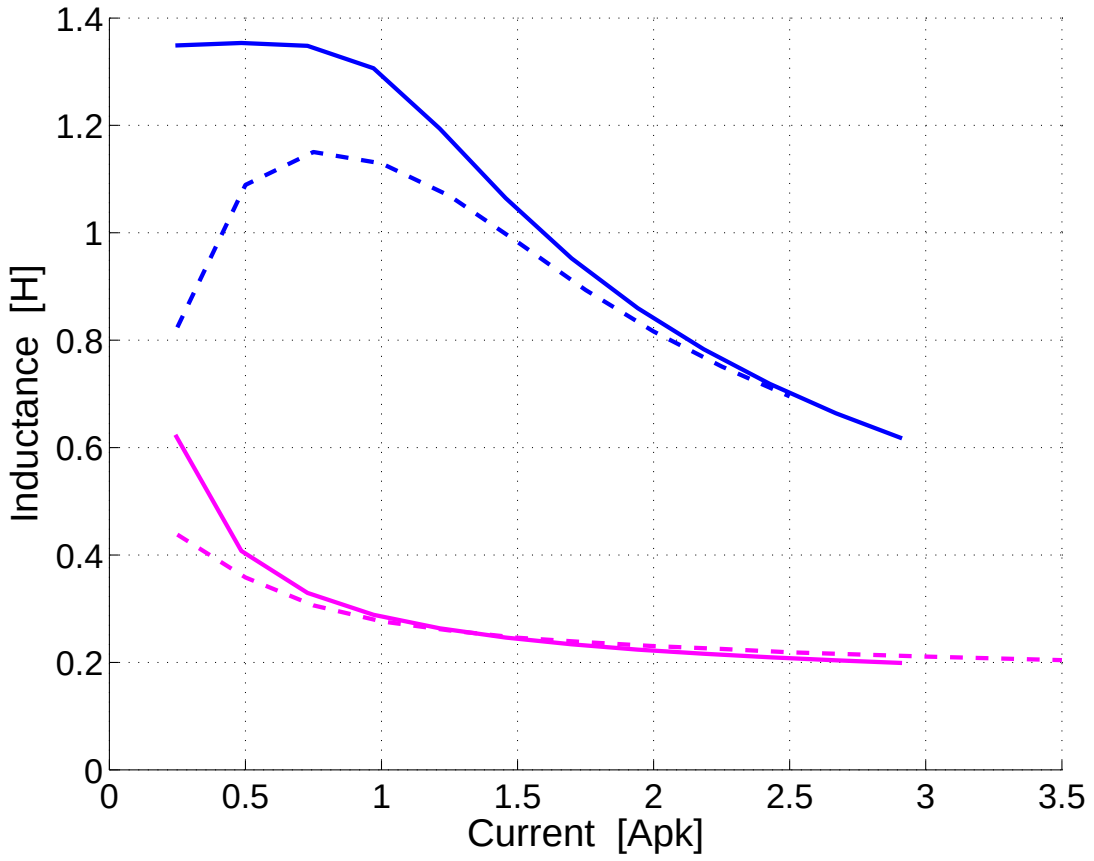


Figure 3.7: Values of inductances for different current values of the SyRM used for control tests.

One of the principal reason we decide to test the MPC with integral action over this motor is represented by its non-linearities. MPC generally uses model parameters to make state prediction and compute optimal inputs, and this is a great opportunity to verified how the control is able to face with so non-linear behaviour of the motor.

3.4 MicroLab Box

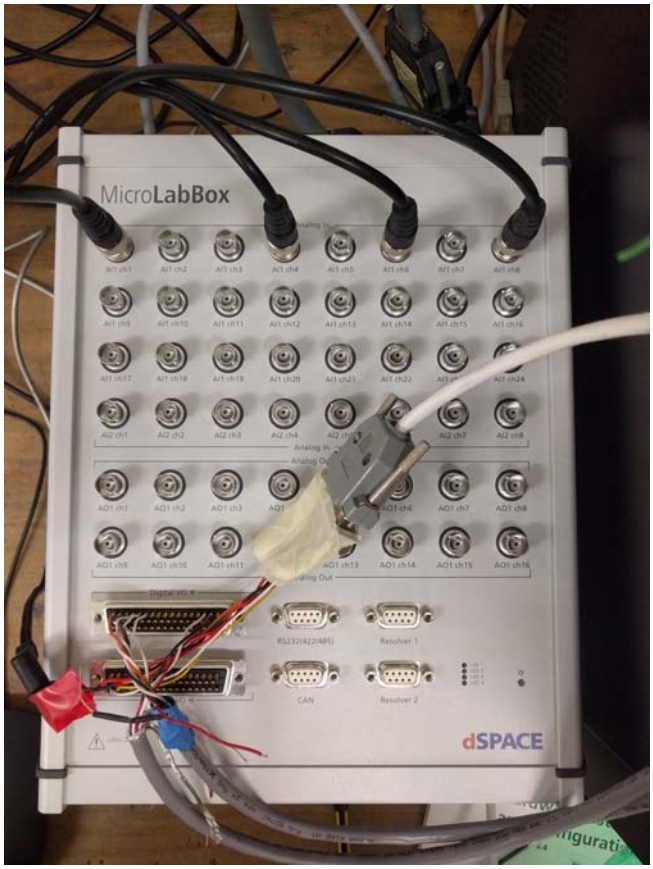


Figure 3.8: The MicroLabBox used for experimental tests.

MicroLabBox[9] is a compact development system for the laboratory with high performance and versatility. More than 100 I/O channels of different types make MicroLabBox a versatile system that can be used in mechatronic research and development areas, such as in our case electric drives control. MicroLabBox is supported by a comprehensive dSPACE software package including, e.g., Real-Time Interface (RTI) for Simulink® for model-based I/O integration and the experiment software ControlDesk, which provides access to the real-time application during run time by means of graphical instruments. Equipped with BNC and Sub-D connectors, the top panel MicroLabBox allows easy access to the analog I/O channels via probes that are typically used in laboratories to offer high analog signal quality. By using BNC connectors it is possible to measure the three-phase currents of the motor, its angular speed by using an encoder and all these signals are then available for the control algorithm. Especially for electric drive application, MicroLabBox includes an easy configuration and implementation of Hall sensor inputs, incremental encoder, Resolver and Synchronous Serial Interface (SSI) interfaces as well as PWM signal generation. In Figure 3.9, it is reported the block diagram that explains how MicroLabBox works.

One of the main advantages of using this type of system is that it allows a comprehensive and user-friendly Electric Motor Control Blockset available on

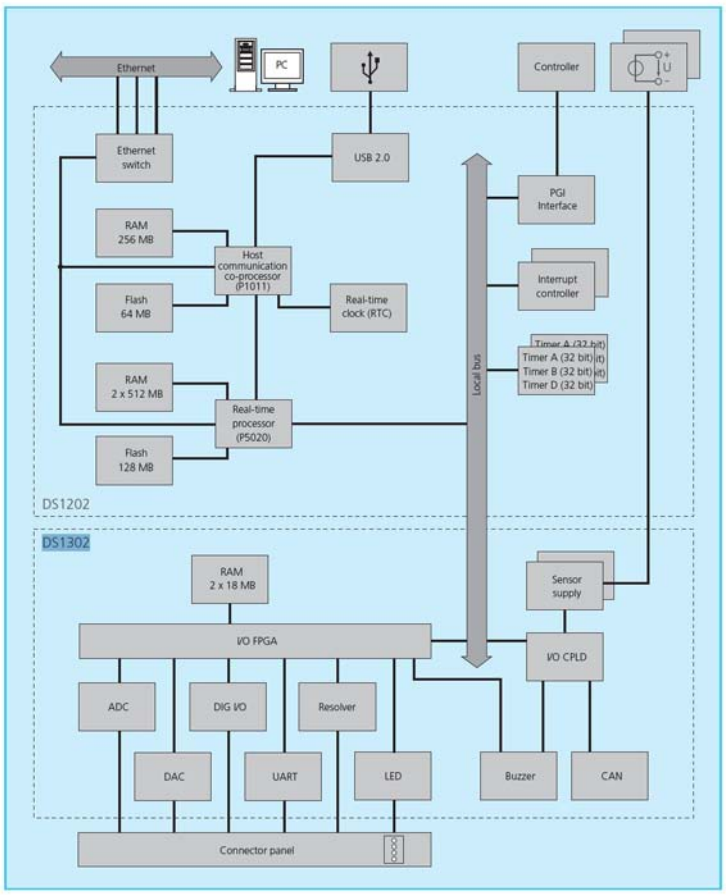


Figure 3.9: MicroLabBox block diagram.

Simulink®. Since the control algorithm is rather simple to implement in Matlab/Simulink code, it results a good strategy to test the electric drive presented in this work. By using this Real Time Interface (RTI) Blockset The current speed, position and angle of the electric motor are automatically calculated. Also, different type of sensors can be used. In order to measure all three-phase stator currents of the SyRM, three LEM are used, and they are reported in Figure 3.10. The three stator phase conductors pass through the LEM sensors and the voltage output signals is sent to the MicroLabBOx using BNC connectors.

In order to load the project on the MicroLabBOx and make possible the execution each sample time T_s (that is the switching period of the inverter), Matlab includes a compiler in C language, that creates an optimizer code ready to be executed by the system. Generally, a Simulink project for an electric drive application that uses a PWM strategy to control the inverter are composed by two main parts, as reported in Figure 3.11. First set of blocks refers to the so called Watchdog, a mandatory set of instructions that gives the enabling of the inverter, and it has a Timer Interrupt that is usually doubled with respect the timer interrupt of the PWM. It plays a key role in real time application because it is able to detect if there are some issue during the execution of the principal interrupt, and it has a priority to stop the execution.

Second set of blocks includes all the process that are performed during a sample

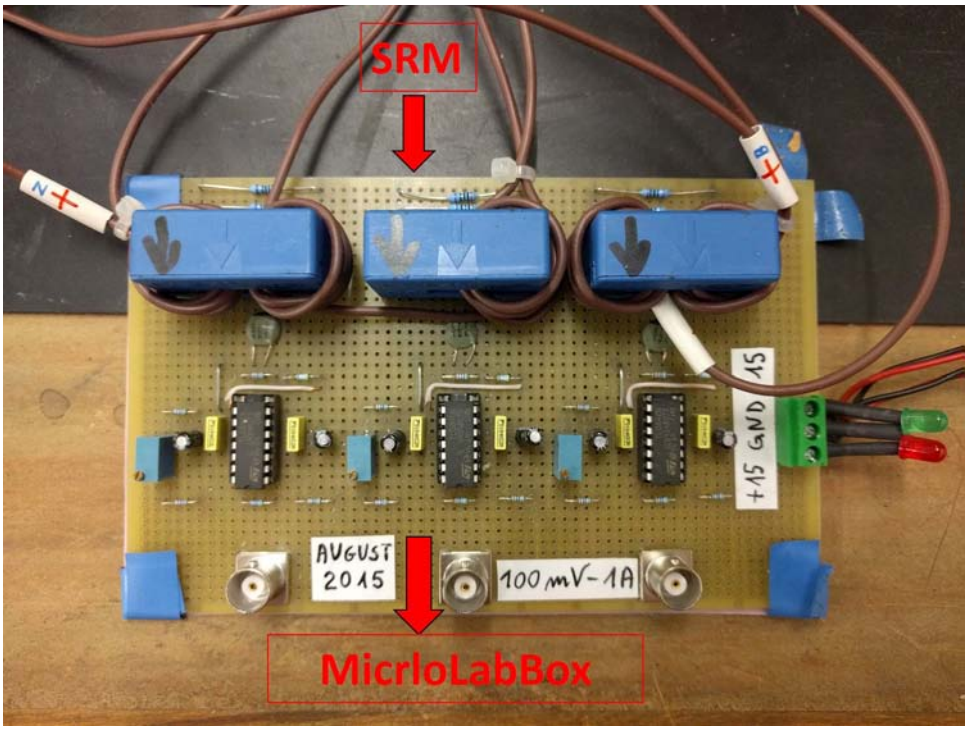


Figure 3.10: Three LEM sensors for stator current measurements.

time T_s of the inverter switching frequency. Principal tasks carried out are:

- Measurement acquisitions (Bus DC,currents, angular speed, mechanical rotor angle);
- made currents and voltages transformations, from a,b,c reference frame to d-q reference frame, using Clark and Park transformation;
- execute the control algorithm and compute the solution;
- generate three PWM signals base on the solution that are sent to the inverter switching control. Signals will be applied during next sample time.

In order to execute experimental tests of the control, a real time Control Desk is available. An example is shown in Figure 3.12.

With this customizable interface it is possible for example to show on line measurements of different magnitudes, give speed and current reference to the control or change parameters to the control as the resistance or inductance values, in order to test the parameter sensitivity.

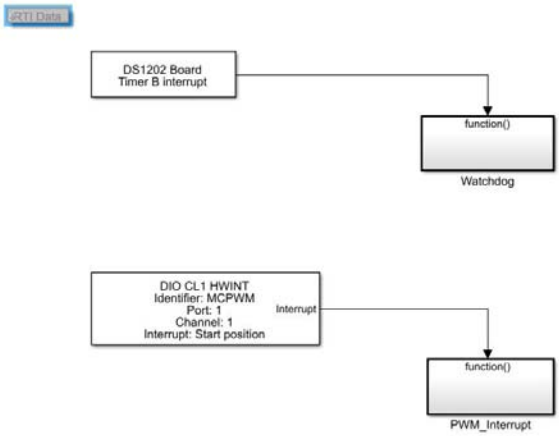


Figure 3.11: Simulink principal control interface.

Figure 3.12: Real Time Control Desk.

3.5 Inverter

In Figure 3.13 is shown the inverter that has been employed in order to drive the SyRM. The MicroLabBox acquires measurements each sample time and compute the optimal input to be applied. When the input is computed, it provides to the inverter the six commands. In fact, the MicroLabBox calculates the voltage references for the Space Vector PWM (SVPWM), which gives the duty cycle signals for the six gates of the inverter.

In order to give some basics about the SVPWM principle, it is reported in Figure 3.14 a general scheme of a three-phase voltage source inverter.

As power supply, the inverter receives a constant voltage value. In our case, this is provided by a Variac, that permits to modify the value of the constant

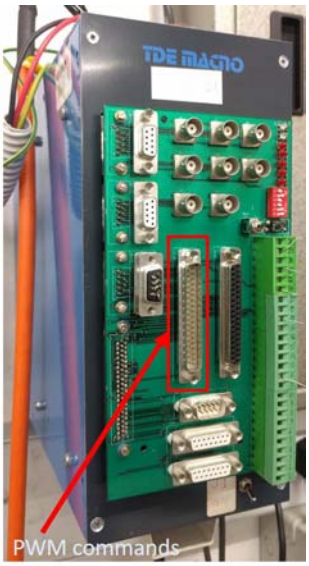


Figure 3.13: The inverter used for the SyRM drive control .

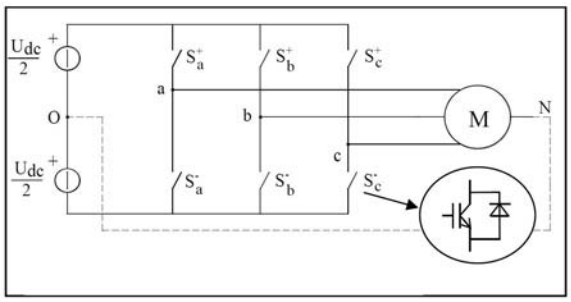


Figure 3.14: Scheme of a three-phase voltage source inverter.

voltage input from 0 V up to 350 V. In Figure 3.13 this is represented by two voltage source series connected in the O central point. The inverter is composed by 6 power switches, and they are connected forming three legs, one for each motor phase. Each switch has an antiparallel diode that enable the current inversion. In fact, usually it is used on and off controlled switches such as IGBT (Integrated Gate Bipolar Transistor), which are abled to conduct the current in one direction only. These devices are suitable for drive application up to 10-15 kHz. Controlling the upper switches, it is possible to apply the positive voltage supply to the respective phase motor. Each couple of switch is driven alternatively, in order to avoid shortcircuits on the supply side. It results eight possible inverter states. The inverter produce three voltages among the a,b,c points and the central point O on the supply side, whose value can be $\frac{+U_{dc}}{2}$, $\frac{-U_{dc}}{2}$. Consider a generic state of the inverter, and apply the following definition to the three voltage magnitudes:

$$\mathbf{u} = \frac{2}{3}\left(u_a + u_b e^{j\frac{2}{3}\pi} + u_c e^{j\frac{4}{3}\pi}\right) \tag{3.12}$$

Equation (3.12) applied to all the eight inverter states, identifies in a complex plane 6 state vectors with amplitude of $\frac{2}{3}U_{dc}$. They identify vertices of a hexagon, which is centered into the origin of axes, as shown in Figure3.15. Two o the eight possible

states represent the null vector.

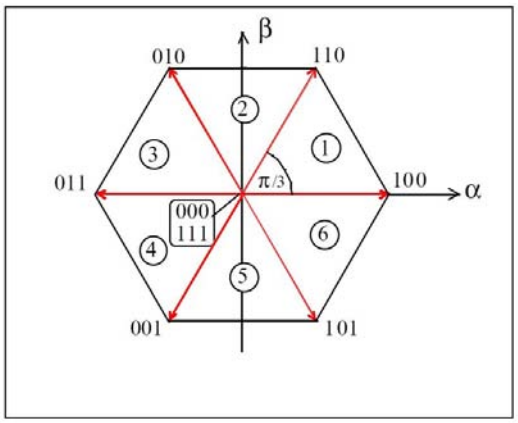
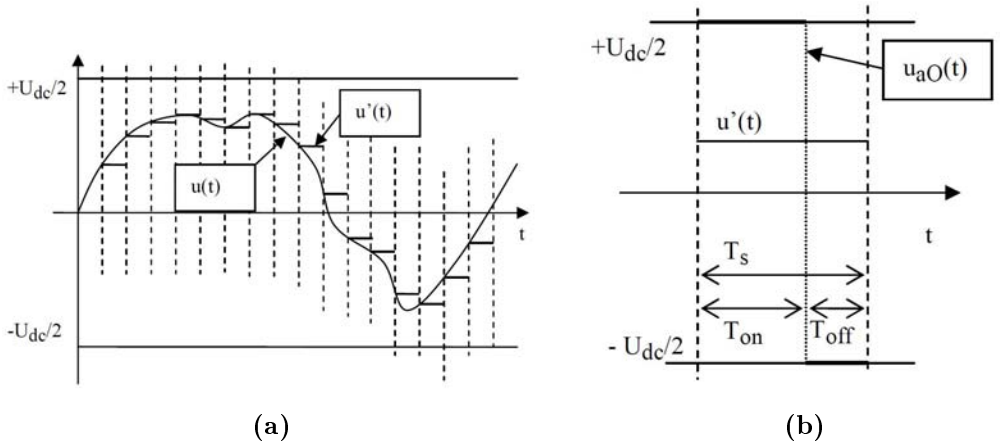


Figure 3.15: Space vector that identify the active state of a three-phase inverter.

Each vector can be also represented by three bits, which refer to the state of the three phases. In general, it is indicated with 1 if it is activated the upper switch of a phase. As can be seen from Figure 3.15, the six vectors divide the hexagon into 6 region, numbered in anti-clockwise order.

General principle of the Pulse Width Modulation (PWM) refers to Figure 3.16. First, consider a generic voltage u(t), whose value remain within $\frac{-U_{dc}}{2}$ and $\frac{+U_{dc}}{2}$. It is considered the subdivision of the time scale in intervals T_s, sufficiently small compared to time intervals in which we register significant variation of u(t). A second voltage u(t)' trend is constructed by using the value of u(t) at each sample time and maintaining its value constant during that T_s (Figure 3.16(a)). The voltages u(t) and u(t)' are obviously different. However, if we apply these voltages to a RL load which has electric time constant sufficiently higher than T_s, the same currents results. Considering a leg of the inverter, we can apply voltages that have value $\frac{+U_{dc}}{2}$ or $\frac{-U_{dc}}{2}$. The inverter can generate in a sample time the voltage which has the mean value equal to u(t), by applying for T_{on} the positive value $\frac{+U_{dc}}{2}$,



(a) **(b)**

Figure 3.16: Principles of the Space Vector PWM.

while during T_{off} it generates the negative value $\frac{-U_{dc}}{2}$, such that $T_{on} + T_{off} = T_s$ (Figure 3.16(b)).

T_{on} can be calculated by imposing the equality between the mean value of u_{aO} (considering for instance the leg that refers to the a-phase of the inverter) in the time interval with the value of u(t)' in the same sample time. It results:

$$u(t) = U_{dc}(\delta - \frac{1}{2})$$

where $\delta = \frac{T_{on}}{T_off}$ is the so called duty-cycle. Regarding a generic k-interval we obtain:

$$\delta_k = \frac{T_{on}}{T_s} = \frac{u_k}{U_{dc}} + 0.5 \tag{3.13}$$

It results that for $u_{aO}(t)$ a succession of couple of positive and negative impulses, with duration of the positive impulse that is width modulated by (3.13). This principle can be extending to a three-phase voltage inverter drive, with the aim to generate the following three sinusoidal voltages:

$$\begin{aligned} u_a(t) &= U_M \cos(\omega t + \theta_0) \\ u_b(t) &= U_M \cos(\omega t + \theta_0 - \frac{2\pi}{3}) \\ u_c(t) &= U_M \cos(\omega t + \theta_0 - \frac{4\pi}{3}) \end{aligned} \tag{3.14}$$

The triad (3.14) can be represented by a space vector using (3.12). This represents a rotating vector in the complex plan. By dividing time in small intervals of duration $T_s \ll \frac{2\pi}{\omega}$, we obtain a sequence of rotating space vectors, separating by a small rotation angle. Finally, it is possible to compute the value of duty cycles at generic instant k as [8]:

$$\begin{aligned} \delta_{a,k} &= \frac{T_{a,on}}{T_s} = \frac{u_{a,k}}{U_{dc}} + 0.5 = \frac{1}{U_{dc}} u_{\alpha,k} + 0.5 \\ \delta_{b,k} &= \frac{T_{b,on}}{T_s} = \frac{u_{b,k}}{U_{dc}} + 0.5 = \frac{1}{U_{dc}} \Big(-\frac{1}{2} u_{\alpha,k} + \frac{\overline{3}}{2} u_{\beta,k} \Big) + 0.5 \\ \delta_{c,k} &= \frac{T_{c,on}}{T_s} = \frac{u_{c,k}}{U_{dc}} + 0.5 = \frac{1}{U_{dc}} \Big(-\frac{1}{2} u_{\alpha,k} - \frac{\overline{3}}{2} u_{\beta,k} \Big) + 0.5 \end{aligned} \tag{3.15}$$

where $u_{\alpha,k}, u_{\beta,k}$ are the real and imaginary part of the space vector at instant k, respectively. The criteria of duty cycle computation described in (3.15) is named Sinusoidal PWM, because the three duty cycles varies as sinusoids. Expression as (3.13) will be implemented to find duty cycles starting from the optimal voltage input computed in the control law of the MPC.

Finally, the test bench includes a Master Interior Permanent Magnet Motor (IPM) that is an anisotropic synchronous motor. It is driven by a commercial inverter. The bench is shown in Figure 3.17. The inverter allows us to perform both the speed and torque master motor control. By directly connected the motor shaft by a mechanical coupling, the master motor can bring in rotation the SyRM. This permits to test our current control scheme, by giving a d-q reference current and bypassing the speed control loop of the SyRM, which make possible to test only the MPC/I-MPC current algorithm.

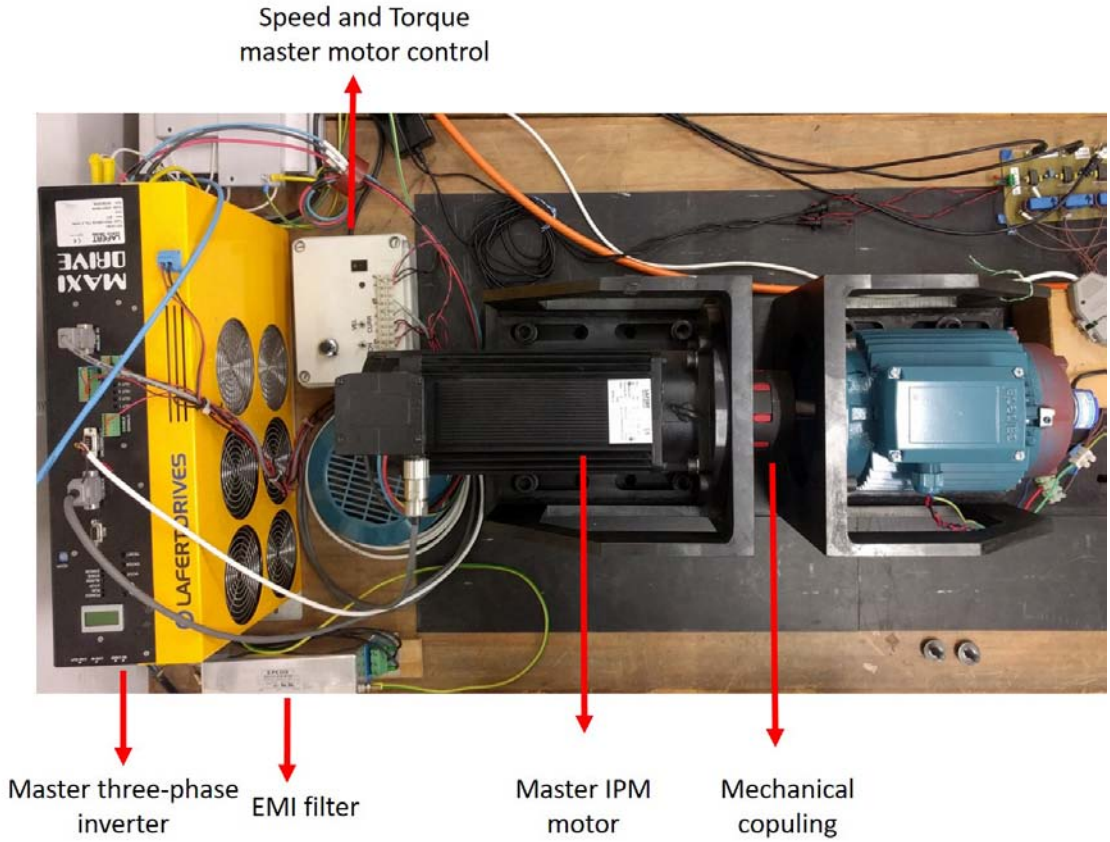


Figure 3.17: Test bench description.

3.5.1 Note on voltage limits

As already mentioned in Chapter 2, a closed solution of the problem is obtainable if we consider the cost function subject to the equality constrain represented by the motor equations. If the equation 3.7 is taken into account, an inequality constrained problem will result. In this case, it is necessary to use iterative methods in order to find the optimal solution. Since the aim of this work it to present a novel strategy belongs to the Predictive Control family, the unconstrained problem is considered and the voltage limits are treated as follows.

The I-MPC algorithm find optimal reference voltages that are applied by the inverter using the SVPWM technique. As previous reported, the maximum phase voltage that the inverter can apply to the motor is $\frac{2}{3}U_{DC}$, that refers to the six hexagon vertices in Figure 3.15. In order to simplify the control, we can assume as limits the circumference that is inscribed in the hexagon, with radius $U_{lim} = \frac{U_{DC}}{3}$. If the control find a voltage vector that exceed the limit, it is truncated while maintaining the same phase. The principle is described in Figure 3.18. Using the notation reported in the figure[1], we can write:

[1]NF stand for Not Feasible, while F means Feasible.

$$\begin{aligned} \frac{U_{d,NF}}{U_{d,F}} &= \frac{U_{q,NF}}{U_{q,F}} \\ U_{d,F}^2 + U_{q,F}^2 &= U_{lim}^2 \end{aligned} \tag{3.16}$$

Substituting the equations we find the expression of the feasible voltage components:

$$\begin{aligned} U_{q,F} &= \pm \frac{U_{q,NF}}{U_{d,NF}} U_{lim} \quad \sqrt{\frac{1}{1 + \frac{U_{q,NF}^2}{U_{d,NF}^2}}} \\ U_{d,F} &= \quad \sqrt{U_{lim}^2 - U_{q,F}^2} \end{aligned} \tag{3.17}$$

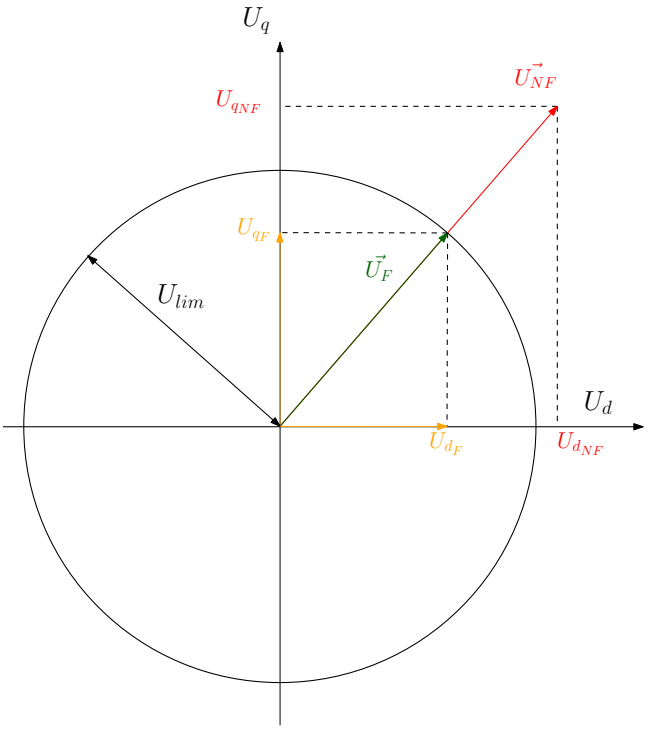


Figure 3.18: Voltage limits.

The code of voltage limit truncations is implemented in both the MPC and the I-MPC algorithms (see Appendix A and Appendix B).

Chapter 4

Simulations

4.1 Simulation procedure

This chapter focuses on simulations of the current MPC algorithm. The main idea is to show differences between the traditional MPC formulation and the novel strategy of the MPC with integral action. Several conditions have been considered with the aim to make comparisons. The control law is applied as current control loop scheme. Simulations are performed considering the SyRM parameters that has been utilized also for the experimental tests. First it is analysed the feasibility of including a speed loop control of the motor. Several tests are performed considering two speed references. Subsequently, the current loop is considered by imposing the motor speed, simulating the steady state condition. It is examined the effect of mismatch parameters on both the control scheme.

In Figure 4.1 it is shown the Simulink set of blocks that are used for simulations. In particular, we can recognize:

- the speed control loop on the left part, that includes a traditional PI control;
- the SyRM block (in orange) in d-q reference frame;
- the inverter (in green) that is modelled as a delay in Laplace domain;
- the Clarke and Park's transformation, which are used to transform the a,b,c current and voltage magnitudes in the d-q reference frame.

In Figure 4.2 it is reported the SyRM block scheme, that represents the implementation of the voltage equations in the d-q reference frame

It can be noticed that the motor presents non-linearities due to the presence of the cross coupling between d-q current axes.

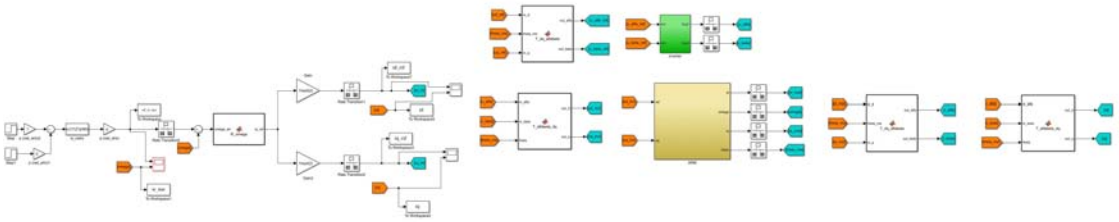


Figure 4.1: Simulink blocks used for simulations.

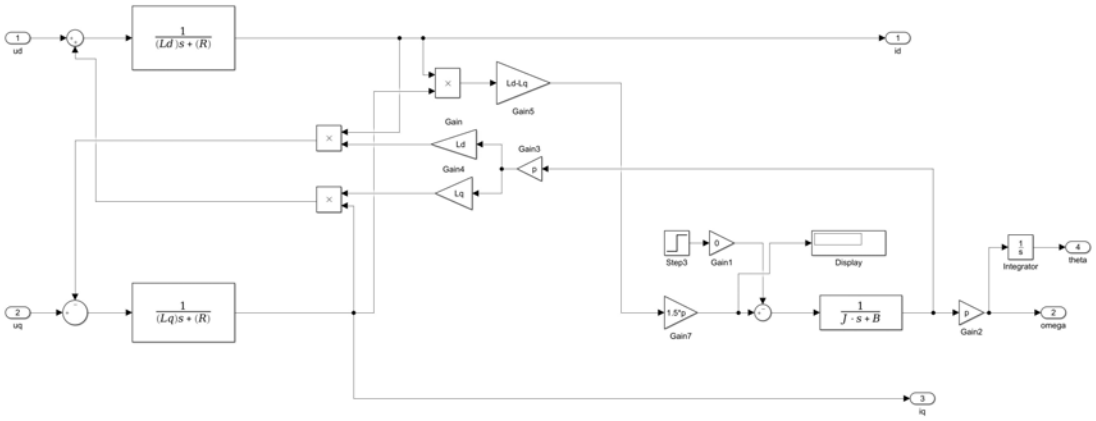


Figure 4.2: SyRM model.

4.2 Speed control loop simulations

Since the motor on which the I-MPC will be tested is a prototype, electric and mechanic nominal parameters are not clearly known. So it is necessary to assume in first analysis relative acceptable parameters, that are reported in Table4.1. Simulations that include the speed control loop consider two speed references, which are 550 and 275 rpm respectively, while for torque step tests is considered the real nominal speed value with the current parameters. It is used to track the speed reference a traditional Proportional-Integral (PI) control. In Figure4.3 it is shown the nested loop current and speed controls. The PI, receives as input the difference between the reference speed and the actual speed of the motor, which identifies the speed error. The control generates a reference current signal that act as a proportional correction (P) of the speed error, weighted by a coefficient k_P. Also, the control integrates the error (I) and weights the integral with a coefficient k_I, in order to achieve zero steady state error after a step speed reference.

Referring to the Laplace domain, a PI control has in general the following transfer function:

$$PI(s) = k_I \frac{1 + s\tau_r}{s}$$

$$\tau_r = \frac{k_P}{k_I}$$

The code implementation of the PI is here reported:

Motor Data			
Pole Pair Number	p	2	
Phase resistance	R	16	Ω
Direct inductance	L_d	1	H
Quadrature inductance	L_q	0.4	H
Nominal current	I_N	3	A(pk)
Rotor Inertia	J	9.5e-4	Nms^2
Viscous friction	B	0.001	Nms
Voltage DC inverter	U_{DC}	300	V

Table 4.1: Motor and inverter nominal parameters.

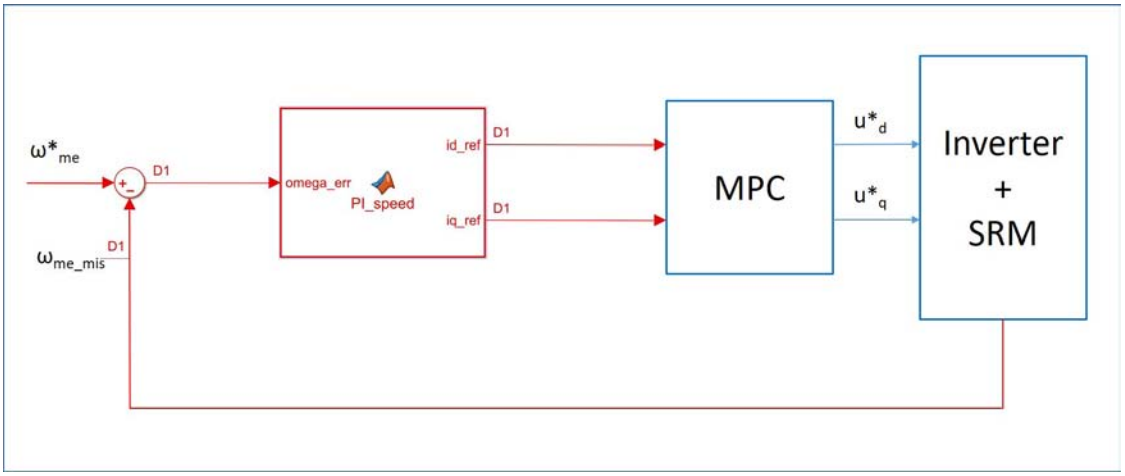


Figure 4.3: Speed control loop of the SyRM.

```
function [id_ref,iq_ref]=PI_speed(omega_err, Ts,I_lim)

%% dati
 K_I=6;
 K_P=0.1;
%% algorithm for i_ref

persistent omega_int

if ( isempty(omega_int) )
    omega_int=0;
end

% discretizzazione di Eulero

omega_int=omega_int + omega_err*Ts;

% PI control

i_ref=K_P*omega_err+K_I*omega_int;
```

```
% anti wind-up statico

if i_ref >= I_lim
    omega_int=(I_lim-K_P*omega_err)/K_I;
    i_ref=I_lim;
end

if i_ref <= -I_lim
    omega_int=(-I_lim-K_P*omega_err)/K_I;
    i_ref=-I_lim;
end

i_ref=K_P*omega_err+K_I*omega_int;

% MTPA SRM
iq_ref = abs(i_ref/sqrt(2));
id_ref = i_ref/sqrt(2);
```

It is pointed out that we suppose to operate in the MTPA region of the SyRM, where $i_d = i_q$. The difficulty to find the nominal speed of the motor due to the relative high variations of the inductances has been overcome by considering as nominal speed a value of 550 rpm. The nominal speed is obtained when the voltage and current is at its limit and the nominal torque is available (point B in Figure 3.5). It can be derived from equations (3.8) considering that in MTPA region it is: $I_d = I_q = \frac{I_{lim}}{2}$. So we give:

$$\Omega_{me,_B} = \pm \frac{\overline{2}U_{lim}}{L_d I_{lim}} \frac{1}{\sqrt{1 + \frac{L_q^2}{L_d^2}}} \tag{4.1}$$

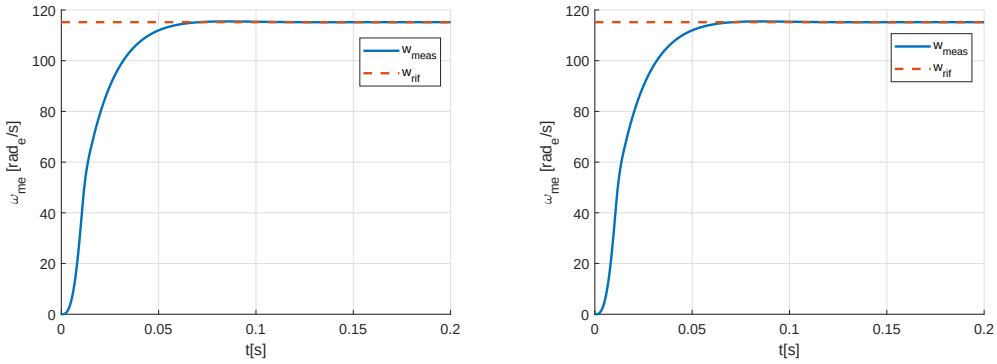
with $U_{lim} = \frac{U_{DC}}{3}$. Referring to Figure 3.7, it is evaluated the inductances at the current of $3A(peak)$ and it results a nominal speed of about 550 rpm. For each speed step reference simulation the currents reference tracking is observed. All the following simulations considered the following weight matrix parameters, both for MPC and I-MPC:

$$\begin{aligned} q_1 &= q_2 = 1 \\ s_1 &= s_2 = 1 \\ r_1 &= r_2 = 10^{-6} \end{aligned} \tag{4.2}$$

Notes about tuning these parameters will be presented in current loop simulations.

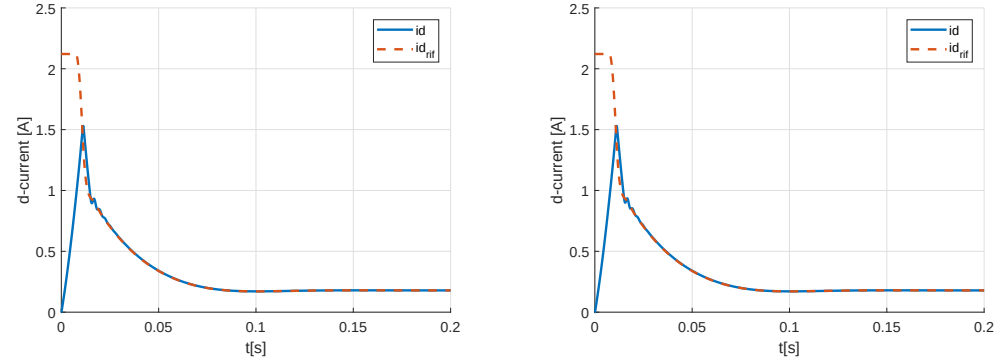
4.2.1 Speed step response

From Figure 4.4 to Figure4.9 it is shown the comparison between the use of MPC and I-MPC considering a reference speed of 550 rpm and 275 rpm. Also, the current dynamics are reported. As can be seen in Figure 4.4 and Figure 4.7, the system behaviour is very similar between the two different speeds. This is quite obvious since the speed control loop is related to the mechanical motor parameters, so the MPC and I-MPC don't point out any substantial difference. In this case the speed dynamics depends also on the mechanical time constant $\tau_M = \frac{J}{B} = 0.95s$. It



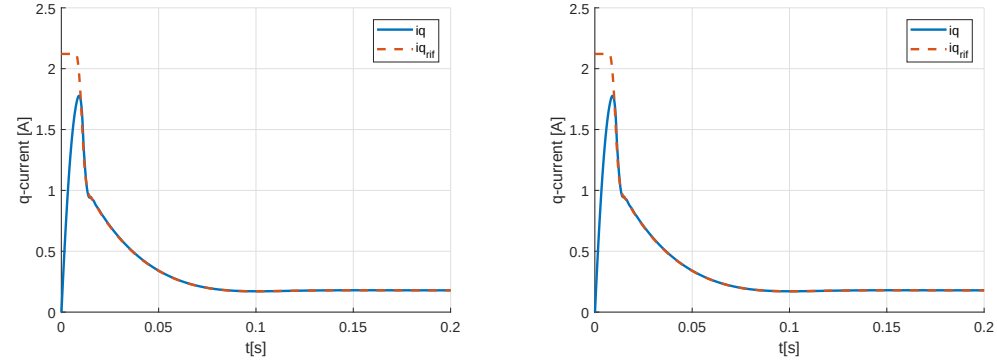
(a) MPC. **(b)** I-MPC.

Figure 4.4: 550 rpm speed step response.



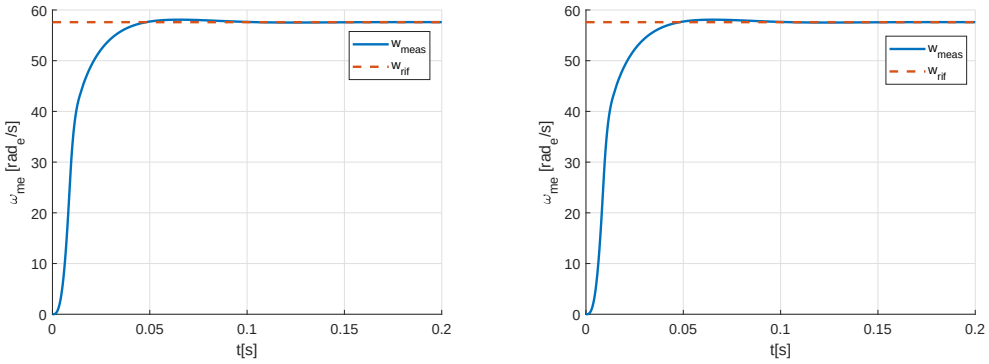
(a) MPC. **(b)** I-MPC.

Figure 4.5: Tracking d-current reference from PI speed control.



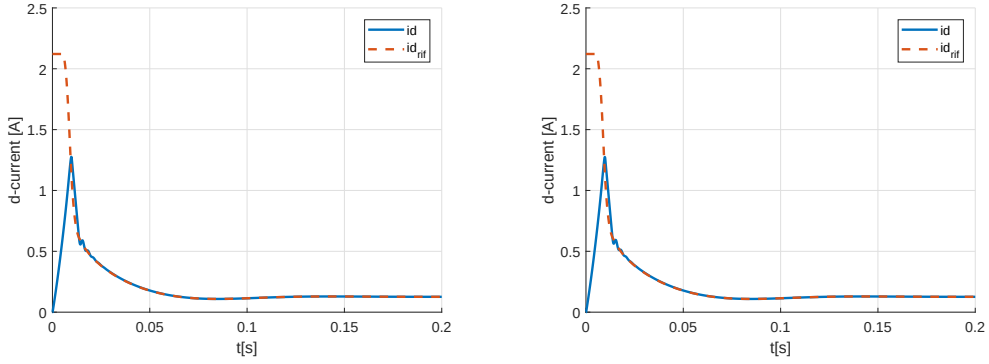
(a) MPC. **(b)** I-MPC.

Figure 4.6: Tracking q-current reference from PI speed control.



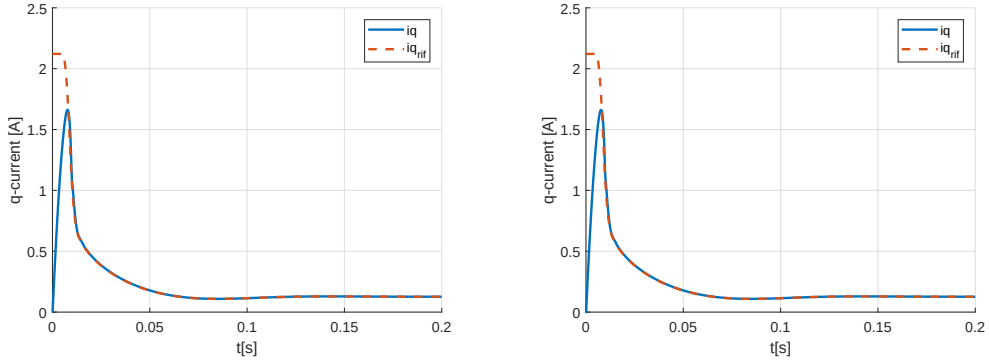
(a) MPC. **(b)** I-MPC.

Figure 4.7: 275 rpm speed step response.



(a) MPC. **(b)** I-MPC.

Figure 4.8: Tracking d-current reference from PI speed control.



(a) MPC. **(b)** I-MPC.

Figure 4.9: Tracking q-current reference from PI speed control.

is interesting to observe that the two control scheme developed the same current dynamic, as shown in Figure 4.6 and Figure 4.9 for the i_q currents and in Figure 4.5 and Figure 4.8 for the i_d currents. The current limits is required at initial instants. In steady state condition it can be noticed that both the MPC and the I-MPC are able to track the reference without offsets.

These simulations are useful to show the stability of the nested current and speed control loop, considering two reference speed values.

4.2.2 Torque step response

Another important simulation to be considered is the torque step test, that simulate the insertion of a mechanical load to the electric SyRM drive. It is here reported the expression of the reluctance torque of the SyRM motor:

$$m(t) = \frac{3}{2}p\Big(L_d - L_q\Big)i_d i_q$$

The nominal torque in MTPA region is available at the nominal speed and peak current. However, the nominal speed depends on motor parameters, as can be seen from (4.1), where the voltage drops across the resistance has been neglected. This is acceptable in case of very low value of R. In our case, where the nominal resistance is 16 Ω, its voltage drop should be included in (4.1). Using current parameters, the nominal torque results $N_{nom} = 8\ Nm$. By using (4.1), the actual nominal speed is with $L_d = 1\ H$, $L_q = 0.4\ H$ it results:

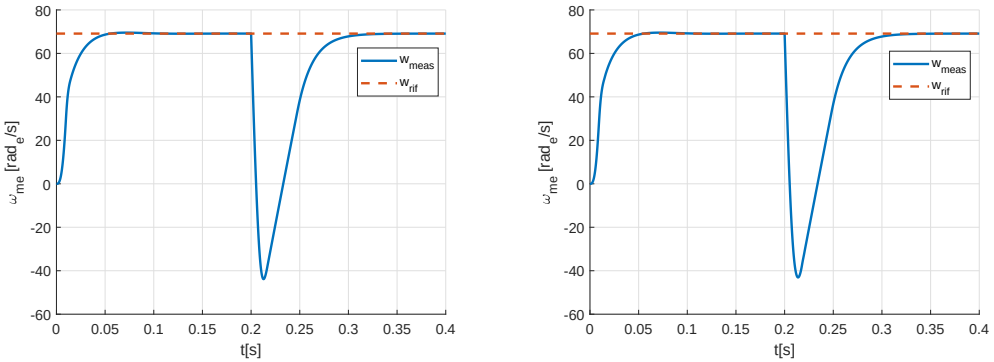
$$\omega_{me_N} \quad 75.8\frac{rad_{el}}{s}$$

that corresponds to 360 rpm. A good compromise to perform the step torque test is to consider e relative lower value of the nominal speed and torque, while maintaining the voltage DC voltage of 300 V. It is chosen a speed reference of 330 rpm and a load torque of 7 Nm. The procedure is based on the following approach: first, we give a speed reference to the motor. When steady state condition is reached, it is applied the load torque

Figure 4.10 shows the comparison between the speed dynamic using the MPC and I-MPC current algorithm respectively. It can be notice that there is no difference between the two approaches, as already anticipated in the previous section. Also, Figure 4.11 and Figure 4.12 point out the d-q current dynamics. This comparison demonstrated that the MPC and I-MPC exhibit the same behaviour after a torque step load.

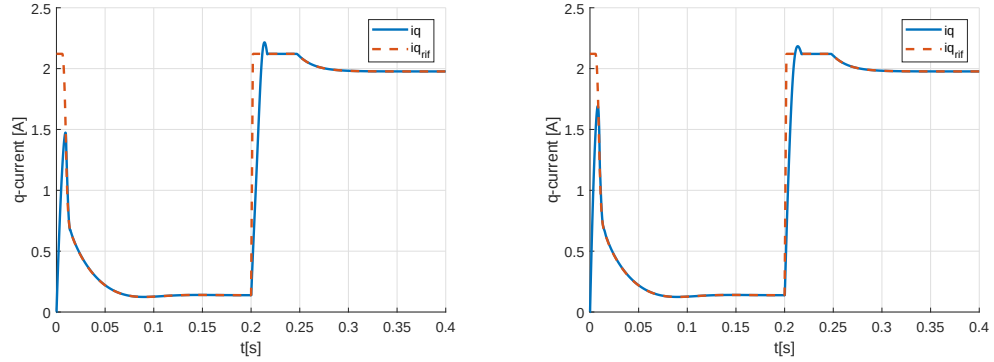
4.2.3 Speed inversion

The most crucial test that involves the speed control loop regards the motion inversion. The simulations consider a step speed reference of 550 rpm. At a certain time, a step reference of -550 is given to the control. The speed dynamic comparison between the MPC and the I-MPC is reported in Figure 4.13, while Figure 4.14 and Figure 4.15 show the d-q current dynamic. Also here, it is confirmed the stability of the nested control loop. In fact, both the MPC and I-MPC behave in



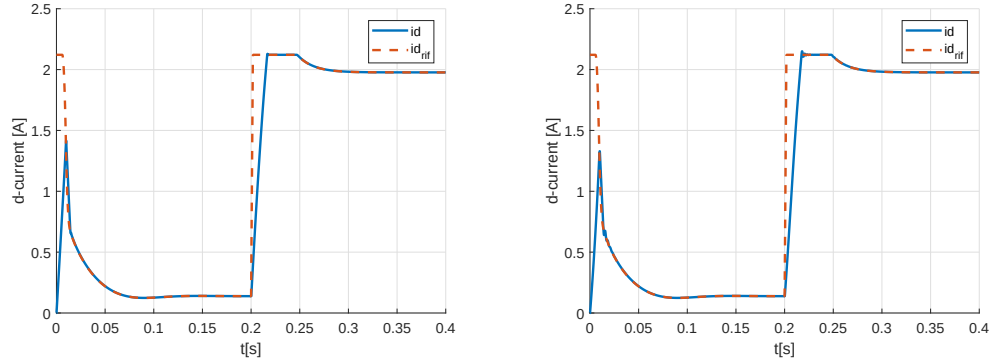
(a) MPC. **(b)** I-MPC.

Figure 4.10: Torque step T = 7 Nm at 330 rpm.



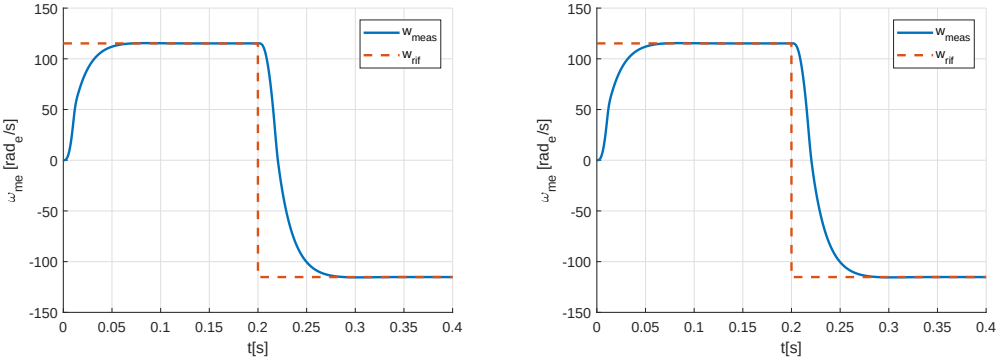
(a) MPC. **(b)** I-MPC.

Figure 4.11: Tracking q-current reference from PI speed control at 330 rpm with a step torque T = 7 Nm.



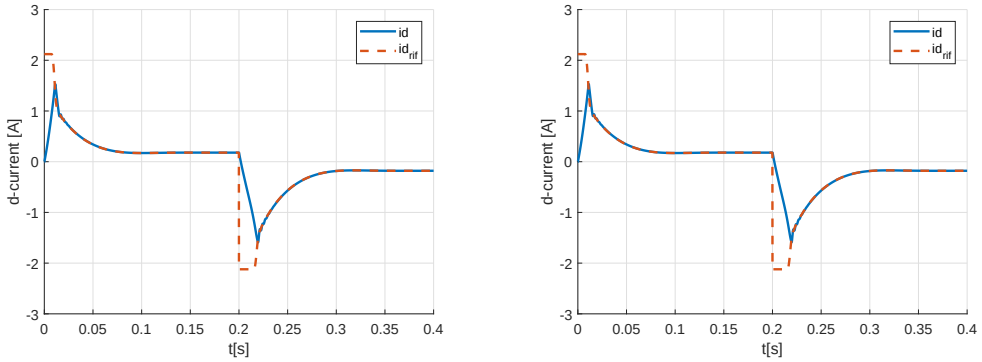
(a) MPC. **(b)** I-MPC.

Figure 4.12: Tracking d-current reference from PI speed control at 330 rpm with a step torque T = 7 Nm.



(a) MPC. **(b)** I-MPC.

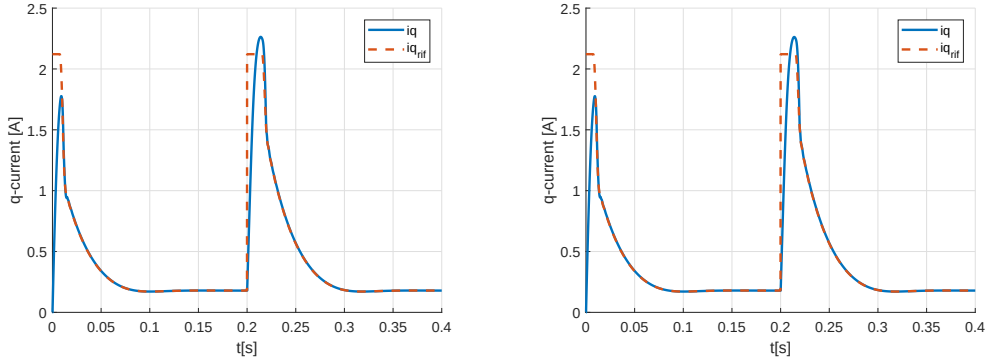
Figure 4.13: Speed inversion from 550 rpm to -550 rpm.



(a) MPC. **(b)** I-MPC.

Figure 4.14: Speed inversion: d-current tracking.

the same way. It is interested to notice that, with a perfect match of the parameters between motor and control scheme, the MPC doesn't evidence any steady state error. Apparently, the control scheme performs the same current action with respect the I-MPC. However, next sections will show benefits of the novel strategy in present of mismatch parameters.



(a) MPC. **(b)** I-MPC.

Figure 4.15: Speed inversion: q-current tracking.

4.3 Current dynamics

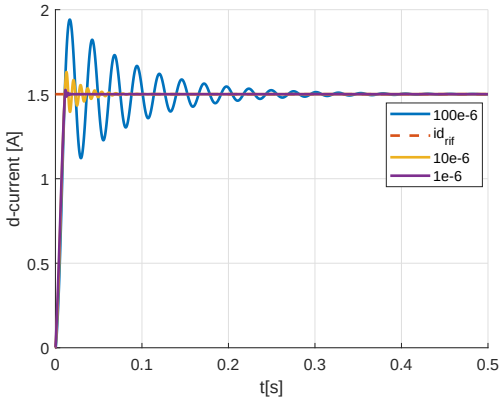
4.3.1 Tuning weight matrices

The chose of weight matrix parameters plays a key role for MPC algorithms. Theory of the MPC has been explained: at each time step the cost function is minimized solving a LQP. Three weight matrices is considered: $\mathbb{Q}$ is related to the minimization of the error between the current reference and the predicted current vector, $\mathbb{R}$ scales the voltage increment $\mathbf{\Delta u}$ and $\mathbb{S}$ refers to the last term of the predicted error. Together, these terms influence the stability of the system; they act on the control as the k_P and k_I coefficients of a PI control. As results from many simulation observations, the most important role is played by the regularization term. First, it is pointed out that it doesn't matter the absolute value of each single weight index, but what it counts is the relative value among the coefficients. Following this principle, it is considered the q indexes equal to one, and other parameters are then tuned. One of the most important remark that has been proved during simulations and experimental tests is the range of value that $\mathbb{R}$ coefficients which can be handled.

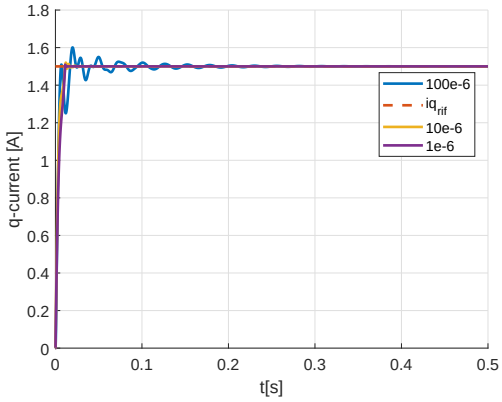
Consider for simplicity the cost function written for the I-MPC d-current with a predicted horizon N = 1. Let the informal annotation we have:

$$\begin{aligned} & i_d^*(k+1) - \big(1 - \frac{RT_s}{L_d}\big)[i_d(k) - id(k-1)] - \\ & \omega_{me}\frac{L_q}{L_d}T_s[i_q(k) - i_q(k-1)] - \frac{Ts}{L_d}\Delta u(k)_{q_1} \\ & \qquad + \Delta u(k)_{r_1} \end{aligned} \tag{4.3}$$

From an intuitive point of view and assuming $q_1 = 1$, the regulation term can have range of values that is in the of magnitudes of $\frac{Ts}{L_d}$. In fact, $\mathbb{R}$ is useful to avoid higher value of voltage variation, both in dynamics and in steady state condition. However, a higher value of r_1 and r_2(for q-axis) compromise the stability because



(a) i_d dynamic.



(b) i_q dynamic.

Figure 4.16: Tune of $\mathbb{R}$ coefficients.

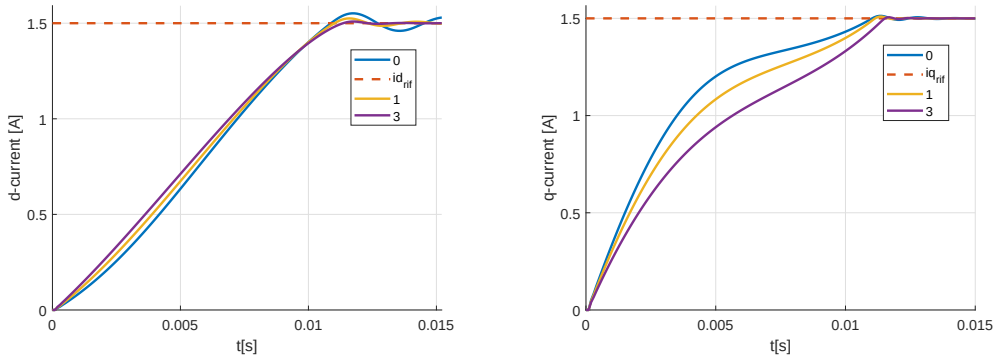
the optimization problem doesn't include the real voltage balance any more. So, for a SyRM the $\mathbb{R}$ coefficients are on the order of or lower than:

$$\frac{T_s}{L_d} = \frac{100e-6}{1} = 100e-6$$

similar value is achieved considering $L-q$. It is interesting to observe the effect of different chose of r_1 and r_2. In order to explain how the regularization term acts on the current control consider Figure 4.16.

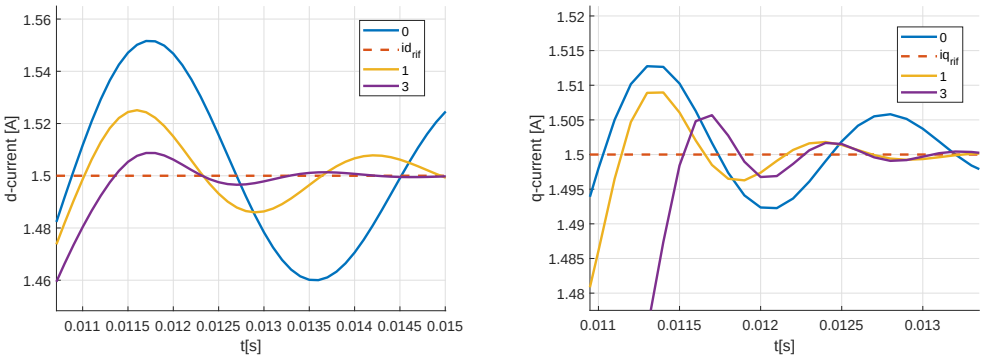
It is given a current step of 1.5 A for both axis currents. The d-q axes are treated with the same value of weight indexes. As can be seen, both for i_d and i_q, with a index comparable to $100e-6$ the dynamic shows oscillations. While, decreasing its value, the overshoot is reduced and also the time rising. In fact, the small the regulation index, the greater the voltage variation is permitted. Indeed, chose the value of r indexes, is a compromize between current fast dynamics and voltage ripple. On the other hand, if we consider r_1, r_2 almost equal to zero, currents exhibit relevant oscillation in steady state.

Another parameter to consider is the matrix $\mathbb{S}$, that weights the last prediction error of the horizon. By considering $\mathbb{Q}$ indexes equal to one, $r_1 = r_2 = 10e-6$, it is interesting to study the effect of the last weight of the predicted error.



(a) i_d dynamic. **(b)** i_q dynamic.

Figure 4.17: Tune of $ coefficients.



(a) i_d dynamic. **(b)** i_q dynamic.

Figure 4.18: Zoom of d-q current dynamics for different value of s_1 and s_2.

It is chosen three values of $ coefficients, and it is compared the d-q current dynamics in Figure 4.17. The weight indexes act in different way: considering i_d, the s_1 coefficient quite improves the rising time, while the dynamics of i_q is slightly slowed increasing its value. However, in both axes the overshoot and the oscillations are reduced , as it is shown in Figure 4.18. Overall, $ coefficients don't strongly affect the performance of the current control, due to the small horizon of N = 3, though they improve the dynamic a bit.

4.3.2 Current step dynamics

In order to evaluate how the current MPC and I-MPC behave in dynamics, the following procedure is used. First, it is necessary to exclude the speed control loop, by not considering the mechanic equation of the motor. Thus, the speed is imposed, simulating what happen in the test bench. In fact, to test the current control of the SyRM, it is dragged by the master motor, which is controlled in speed. Considering sinusoidal steady state condition the motor equation in d-q reference frame is expressed by constant terms, thanks to Clarke and Park's transformations:

$$\begin{aligned} U_d &= RI_d - \Omega_{me} L_q I_q \\ U_q &= RI_q + \Omega_{me} L_d I_d \end{aligned} \tag{4.4}$$

In MTPA regions it holds $I_d = I_q$. It is taken into account also the voltage drop terms. By assuming $I_d = I_q = \frac{I_{lim}}{\bar{2}}$, equations (4.4) can be inserted in the limit voltage equation $U_d^2 + U_q^2 = U_{lim}^2$:

$$(R - \Omega_{me} L_q)^2 \frac{I_{lim}^2}{2} + (R + \Omega_{me} L_d)^2 \frac{I_{lim}^2}{2} = U_{lim}^2$$

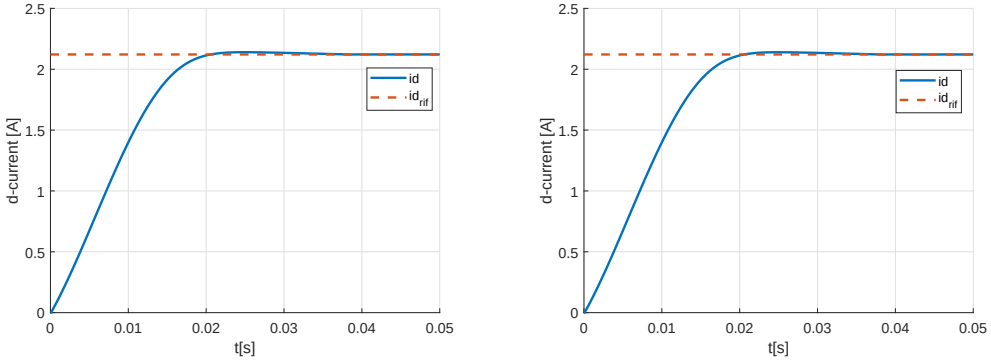
and solving for positive Ω_{me}, it is found:

$$\Omega_{me_N} = 65 \frac{rad_{el}}{s}$$

which corresponds to about $300 rpm$. Giving the limit current as reference, in Figure 4.19 it is shown the comparison between d current dynamics of the MPC and I-MPC, while in Figure 4.19 the q-axis currents is considered. Furthermore, Figure 4.21 and Figure 4.22 report the same simulations, considering a current step of $1.5A$ for both axes. The latter simulations are performed according to the experimental tests that are presented in the next chapter. The following tuning parameters are considered:

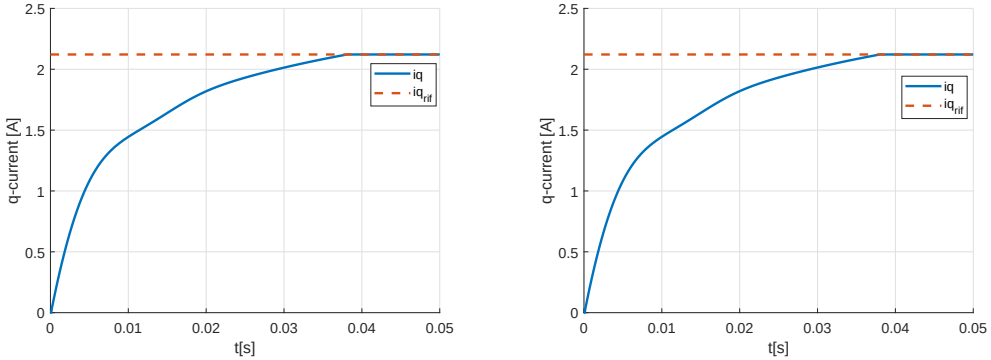
$$\begin{aligned} q_1 &= q_2 = 1 \\ s_1 &= s_2 = 2 \\ r_1 &= r_2 = 1e-6 \end{aligned}$$

It is clear that the performance of the two control scheme in dynamics are almost the same. It is always important to underline the fact that these simulations are only a formal demonstration that the performances of the I-MPC is comparable with those of the MPC. Since the motor parameters, in particular the inductances, varies in a wide range over the operating regions, the magnitudes of L_d and L_q



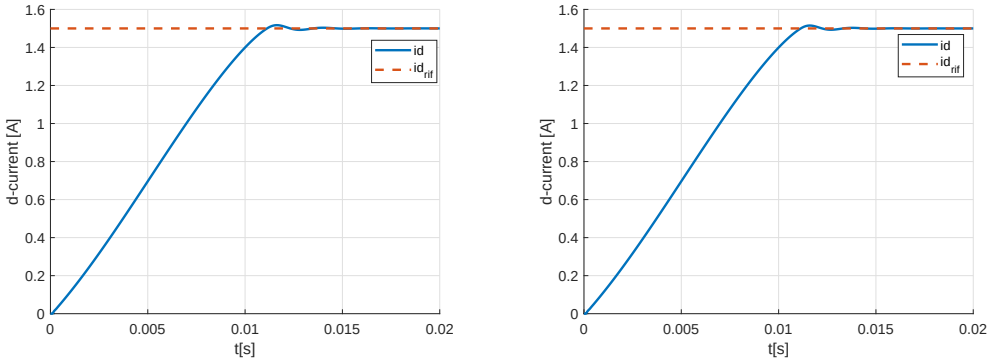
(a) MPC i_d dynamic. **(b)** I-MPC i_d dynamic.

Figure 4.19: Comparison between d-current dynamics using MPC and I-MPC, at 300 rpm and $i^*_{rif} = I_{lim}$.



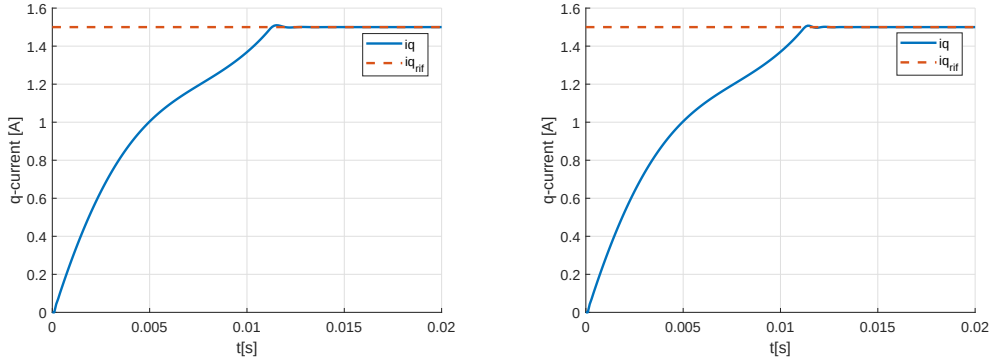
(a) MPC i_q dynamic. **(b)** I-MPC i_q dynamic.

Figure 4.20: Comparison between q-current dynamics using MPC and I-MPC, at 300 rpm and $i^*_{rif} = I_{lim}$.



(a) MPC i_d dynamic. **(b)** I-MPC i_d dynamic.

Figure 4.21: Comparison between d-current dynamics using MPC and I-MPC, at 300 and $i^*_{d_{rif}} = 1.5A$.



(a) MPC i_q dynamic. **(b)** I-MPC i_q dynamic.

Figure 4.22: Comparison between q-current dynamics using MPC and I-MPC, at 300 and $i^*_{q_{rif}} = 1.5A$.

are chosen as indicative. In this case, the i_d current evolves with a time constant $\tau_{ed} = \frac{L_d}{R} = 62.5ms$. The limit current in MTPA is reached in less than a time constant (Figure 4.19 (a),(b)) and also for a reference of $1.5A$ (Figure 4.21 (a),(b)). On the other hand, the time constant of the q-axis $\tau_{eq} = \frac{L_q}{R} = 25ms$. It appears from (Figure 4.20 (a),(b)) that for the q-axis both the control could be improved by change the setting of parameters. Nevertheless, in simulation constant parameters are assumed. In real tests the inductances present a mismatch with respect nominal ones, due to the iron saturation. In this case it has been preferred to assume same parameters for d and q axis in order to limit the variable parameters and so make possible the comparison between the two schemes.

4.3.3 Parameter sensitivity

In this section it is pointed out the most important advantage of using I-MPC. Up to now, it has been shown the comparison of the two schemes assuming that there were perfect parameter correspondence between the motor and the control. A (SyRM) has been chosen for the first implementation due to the fact that the absence of magnets simplifies the motor equation. In particular the PM flux linkage is equal to zero and thus vector **h**. However, it is remembered that the Synchronous Reluctance Motor presents a very non linear behavior in the operating region. In this way, it is interesting to show how the I-MPC can overcome mismatch parameters while robustly keeping the reference tracked. All the possible mismatches are analysed. The simulations are performed in steady state condition, by imposing the speed of the motor and giving a reference current. For the MPC, the simulations consider two motor speeds, the halved value of the nominal and the nominal one. It is quite obvious that refers to an absolute value of the nominal speed is not properly correct. In fact, by changing the motor parameters, the nominal speed of the drive is modified. Overall, in order to make comparison, the value of the nominal speed previously calculated is considered. Only in the case of R mismatches, the value is recomputed. This is due to the fact that, if we reduce the inductance values, the base speed increases, and so the nominal current is available to the drive

also at a lower speed. While,by increasing the resistance value of the motor, the base speed decreases, and the nominal current is not so longer a feasible operation point.

4.3.4 L_d and L_q mismatch

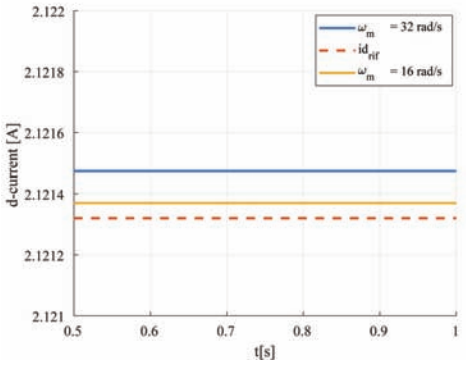
Referring to Figure 3.7, it is known that both the inductances L_d and L_q change of a factor even grater than 2, inside current range space of the machine. Thus, first the d-axis motor inductance is halved, approximating the iron saturation. Simulations are based on the following procedure, considering L_d but it holds also for L_q:

- First, the L_d of the motor equations is halved to the value $L_d = 0.5H$;
- two current references are considered: the current limit of $3A$ and the reference $i_d = 1.5A$, $i_q = 1.5A$; always the ideal MTPA operating region is assumed;
- for the MPC scheme, two speeds are imposed: the nominal one and a half of the nominal speed;
- once the system reach the steady state condition, the mismatch between current reference and measured is reported.

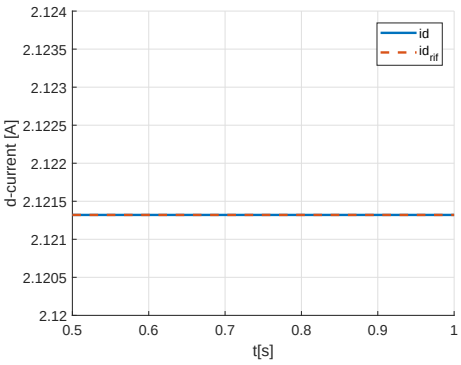
Simulations consider the operation point of $i_d = 1.5A$, $i_q = 1.5A$ and 300 rpm speed because experimental results are performed in this condition. The weight matrix parameters for both the control scheme are set as follows:

$$q_1 = q_2 = 1$$
$$r_1 = r_2 = 1e-6$$
$$s_1 = s_2 = 1$$

Figure 4.23(a) and Figure 4.24(a) report the steady state condition reached giving the limit current as reference. In Table 4.2 it is reported the current error due to a L_d mismatch of a halved of the nominal value. As can be seen, a steady state error using the MPC appears. Furthermore, by simulating iron saturation decreasing the L_d value, it is interesting to observe that the error is more pronounced in the q-axis current than in the d-axis one. This is due to the cross saturation. In fact, each terms of the state equation written for the d-axis motor are dived by the L_d term. So, intuitively the equation is scaled by a factor that depends on the mismatch and the error results less accentuated. On the other hand, the L_d coefficient appears in the q-axis equation only in the cross coupling term $\frac{L_d}{L_q} i_d$, then the MPC solves an optimization problem that presents an offset from the real one. In conclusion, this offset appears in steady state condition, and the reference is not correctly tracked. The error indeed is more pronounced at higher speed (Figure 4.24(a)), and also it varies assuming different reference currents. For instance, considering Figure 4.24(a) and Figure 4.26(a), the steady state error for the q-current increases as many as the current reference is higher.



(a) MPC: i_d current at $\omega_m = 32rad/s$ and $16rad/s$.



(b) I-MPC: i_d current at $\omega_m = 32rad/s$.

Figure 4.23: Comparison between the MPC and I-MPC d-axis inductance sensitivity: i_d steady state current, $i_{ref} = I_{lim}$, $L_d = 0.5H$.

Table 4.2: Steady state current errors using MPC with L_d mismatches.

Current error [mA]	Parameter mismatch	$\Omega_m = 32rad/s$	$\Omega_m = 16rad/s$
e_{i_d}	$Ld = 0.5[H]$	0.153 [mA]	0.05 [mA]
e_{i_q}		40.12 [mA]	20.4 [mA]

On the contrary, observing Figure 4.23(b),Figure 4.24(b),Figure 4.25(b) and Figure 4.26(b) it is clear that the I-MPC is able to face with this mismatch parameter. In fact, in all cases it brings the current to the reference with zero steady-state error. This means that the control is robust with respect L_d parameter mismatches.

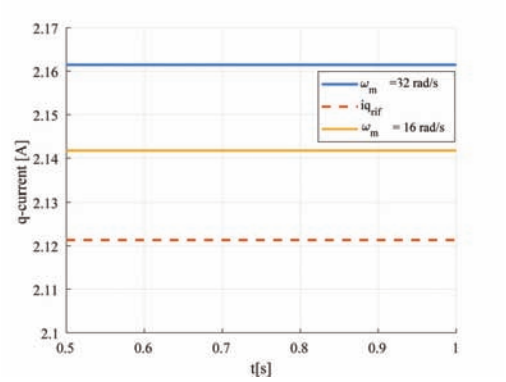
The same analysis can be performed consider the mismatch of the other inductance L_q. A mismatch of 0.2 H is considered with respect the nominal value of 0.4 H. Figure 4.27 and Figure 4.28 report the comparison between the MPC and I-MPC steady state operation. Here it is the dual situation of the L_d mismatch: the error using the MPC registered higher value on the d-axis current due to the cross coupling. The phenomenon is not linear with the motor speed. The I-MPC control scheme is able to deal with L_q mismatch, obtaining zero steady state error. It is said that the I-MPC is robust with the q-axis inductance variations.

4.3.5 R mismatch

The second parameter mismatch regards stator phase resistance. During the operation, its value can significantly grow. It is known that the resistivity of a metal, is a function of the temperature:

$$\rho = \rho_0[1 + \alpha(T - T_0)] \quad [\Omega mm^2/m]$$

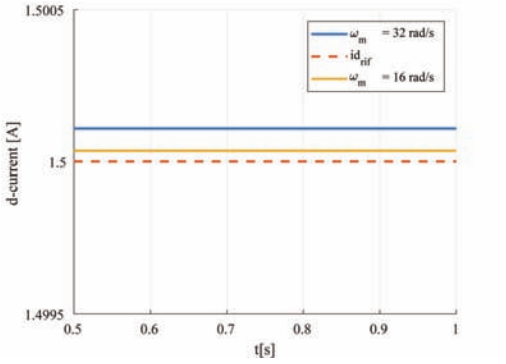
where ρ is the resistivity, T is the temperature and ρ_0 is the metal resistivity at the reference temperature T_0, usually $20°C$. For copper it is $\rho_0 = 0,0175 \quad [\Omega mm^2/m]$ The α value is the thermal coefficient, whose value for the copper is $3.95e-3K^{-1}$.



(a) MPC: i_q current at $\omega_m = 32rad/s$ and $16rad/s$.

(b) I-MPC: i_q current at $\omega_m = 32rad/s$.

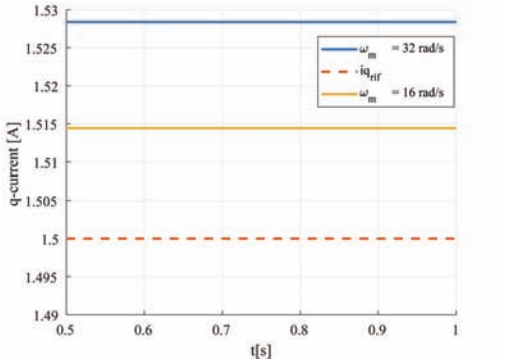
Figure 4.24: Comparison between the MPC and I-MPC d-axis inductance sensitivity: i_q steady state current, $i_{ref} = I_{lim}$, $L_d = 0.5H$.



(a) MPC: i_d current at $\omega_m = 32rad/s$ and $16rad/s$.

(b) I-MPC: i_d current at $\omega_m = 32rad/s$.

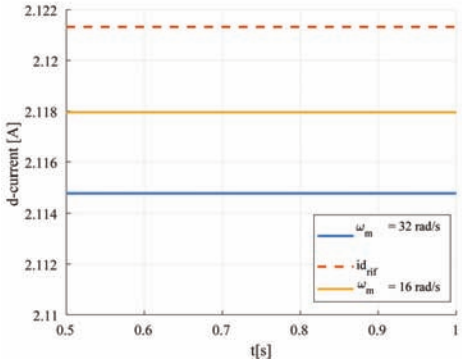
Figure 4.25: Comparison between the MPC and I-MPC d-axis inductance sensitivity: i_d steady state current, $i_{d_{ref}} = 1.5A$, $L_d = 0.5H$.



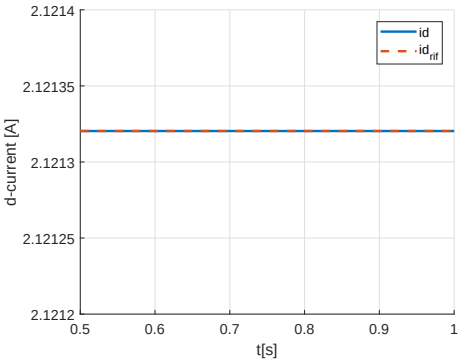
(a) MPC: i_q current at $\omega_m = 32rad/s$ and $16rad/s$.

(b) I-MPC: i_q current at $\omega_m = 32rad/s$.

Figure 4.26: Comparison between the MPC and I-MPC d-axis inductance sensitivity: i_q steady state current, $i_{q_{ref}} = 1.5A$, $L_d = 0.5H$.

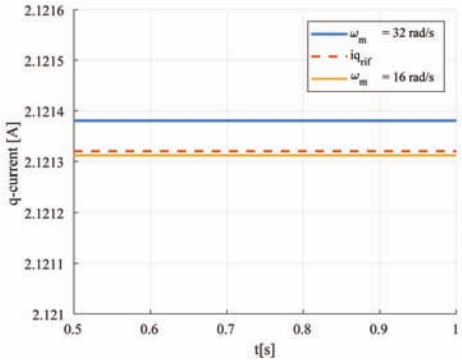


(a) MPC: i_d current at $\omega_m = 32rad/s$ and $16rad/s$

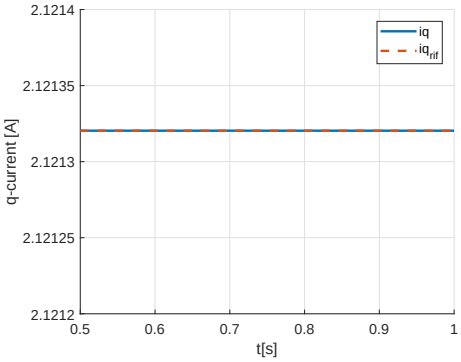


(b) I-MPC: i_d current at $\omega_m = 32rad/s$.

Figure 4.27: Comparison between the MPC and I-MPC q-axis inductance sensitivity: i_d steady state current, $i_{ref} = I_{lim}$, $L_q = 0.2H$.

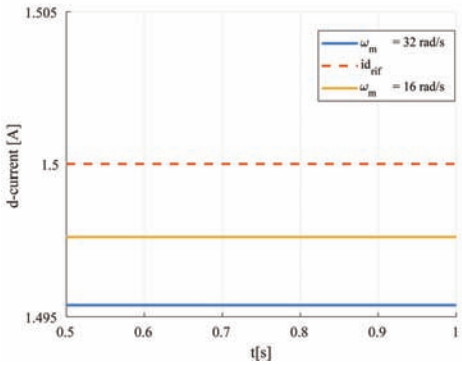


(a) MPC: i_q current at $\omega_m = 32rad/s$ and $16rad/s$.

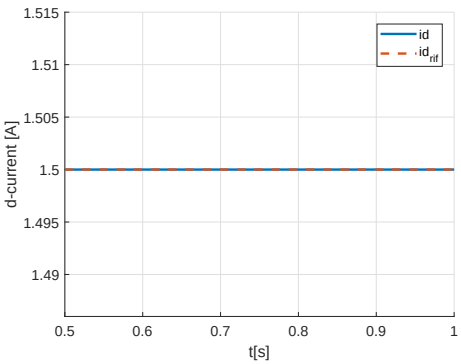


(b) I-MPC: i_q current at $\omega_m = 32rad/s$.

Figure 4.28: Comparison between the MPC and I-MPC q-axis inductance sensitivity: i_q steady state current, $i_{ref} = I_{lim}$, $L_q = 0.2H$.

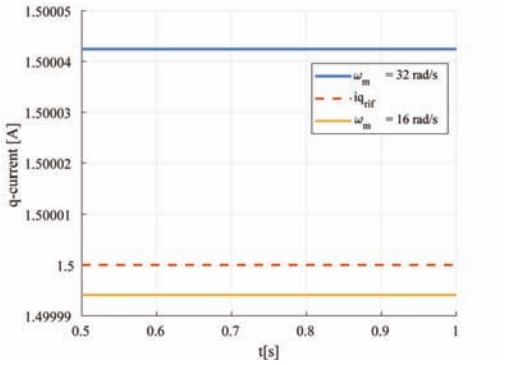


(a) MPC: i_d current at $\omega_m = 32rad/s$ and $16rad/s$.

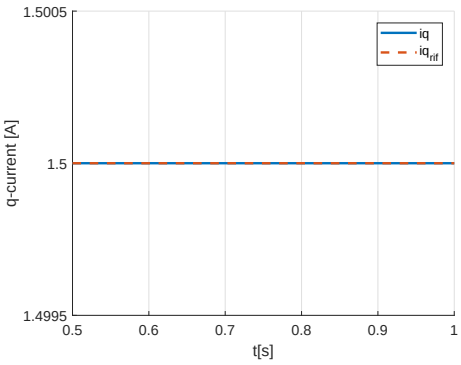


(b) I-MPC: i_d current at $\omega_m = 32rad/s$.

Figure 4.29: Comparison between the MPC and I-MPC q-axis inductance sensitivity: i_d steady state current, $i_{d_{ref}} = 1.5A$, $L_q = 0.2H$.

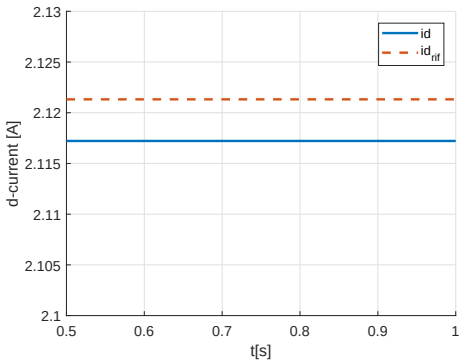


(a) MPC: i_q current at $\omega_m = 32rad/s$ and $16rad/s$.

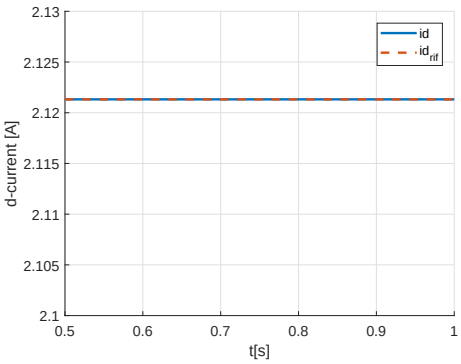


(b) I-MPC: i_q current at $\omega_m = 32rad/s$.

Figure 4.30: Comparison between the MPC and I-MPC q-axis inductance sensitivity: i_q steady state current, $i_{q_{ref}} = 1.5A$, $L_q = 0.2H$.



(a) MPC: i_d current at $\omega_m = 28.8rad/s$.



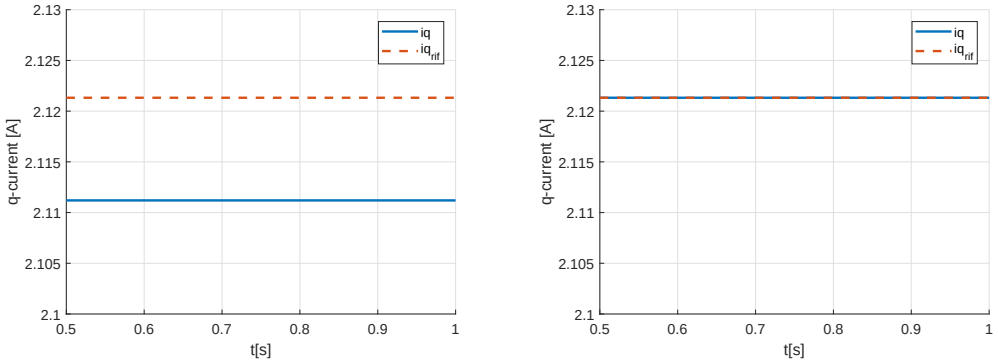
(b) I-MPC: i_d current at $\omega_m = 28.8rad/s$.

Figure 4.31: Comparison between the MPC and I-MPC stator phase resistance sensitivity: i_d steady state current, $i_{ref} = I_{lim}$, $R = 24\Omega$.

It is considered a 40% increase of the nominal resistance value, which corresponds to a temperature rise of almost $100K$.

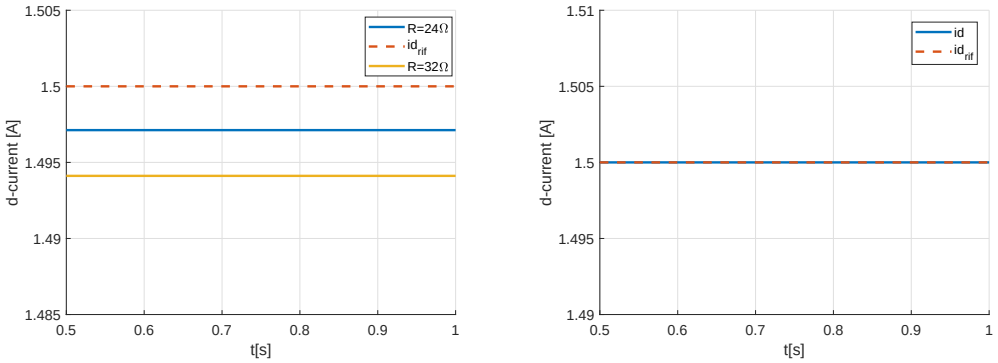
In Figure 4.31 and Figure 4.32 it is compared the steady state operation of the MPC and I-MPC. In this case the motor phase resistance assume value $R = 24\Omega$, simulating the increase of the temperature described above. Besides, it has been considered the limit current as reference, so the nominal speed it is recomputed to the resistance variation. It results a mechanical speed value of $28.28rad/s$. Since both state equations contain the resistive drop, a comparable steady state error affects d-q currents, as shown in the Figure 4.31(a) and Figure 4.32(a). However, Figure 4.31(b) and Figure 4.32(b) pointed out that the I-MPC can deal also with resistive mismatches.

Figure 4.33 and Figure 4.34 replicate the experimental tests that have been carried out ($R = 24\Omega$) with $i_{d,ref} = i_{q,ref} = 1.5A$; also it is reported a simulation that considers the motor phase resistance doubled up to 32Ω. Also in this case, it is possible to notice the advantages of using the I-MPC: it brings the current to



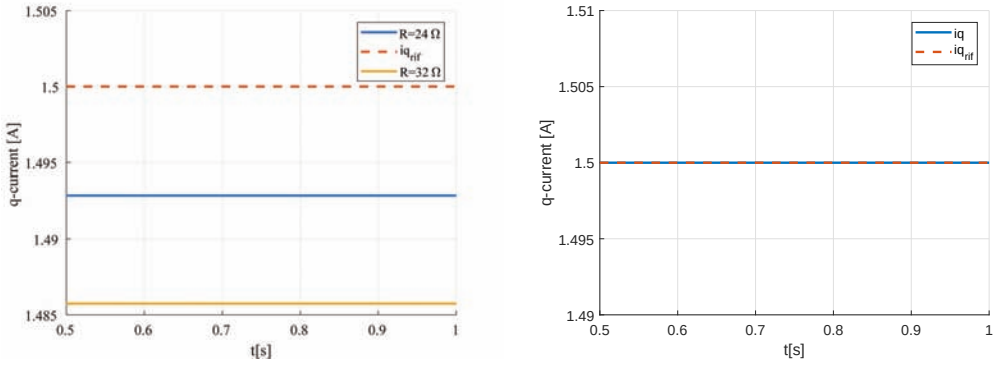
(a) MPC: i_q current at $\omega_m = 28.8 rad/s$. **(b)** I-MPC: i_q current at $\omega_m = 28.8 rad/s$.

Figure 4.32: Comparison between the MPC and I-MPC stator phase resistance sensitivity: i_q steady state current, $i_{ref} = I_{lim}$, $R = 24\Omega$.



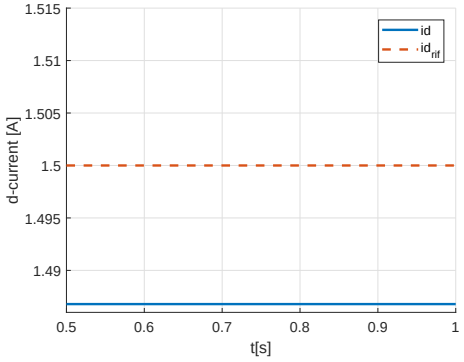
(a) MPC: i_d current at $\omega_m = 32 rad/s$. **(b)** I-MPC: i_d current at $\omega_m = 32 rad/s$.

Figure 4.33: Comparison between the MPC and I-MPC stator phase resistance sensitivity: i_d steady state current, $i_{d_{ref}} = 1.5A$, $R = 24\Omega$ and 32Ω.

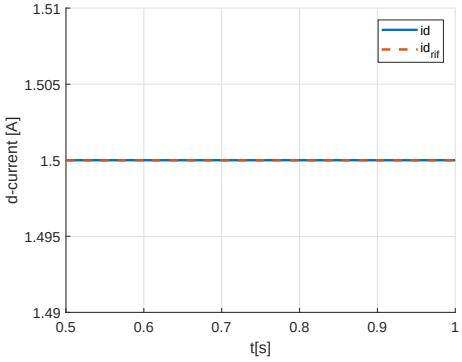


(a) MPC: i_q current at $\omega_m = 32 rad/s$. **(b)** I-MPC: i_q current at $\omega_m = 32 rad/s$. .

Figure 4.34: Comparison between the MPC and I-MPC stator phase resistance sensitivity: i_q steady state current, $i_{q_{ref}} = 1.5A$, $R = 24\Omega$ and 32Ω

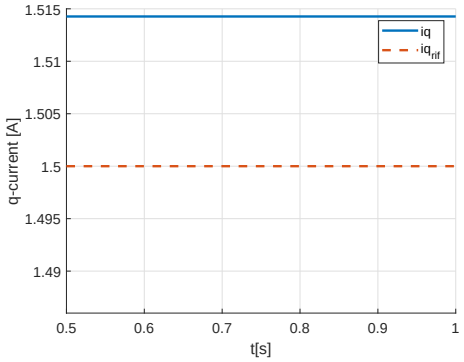


(a) MPC: i_d current at $\omega_m = 32rad/s$.

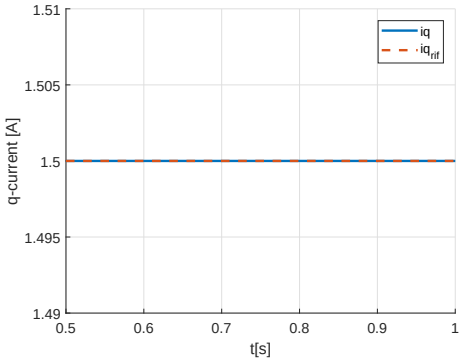


(b) I-MPC: i_d current at $\omega_m = 32rad/s$.

Figure 4.35: Comparison between the MPC and I-MPC sensitivity: i_d steady state current, $i_{d_{ref}} = 1.5A$, $R = 24\Omega$, $L_d = 0.5H$, $L_q = 0.2$.



(a) MPC: i_q current at $\omega_m = 32rad/s$.



(b) I-MPC: i_q current at $\omega_m = 32rad/s$.

Figure 4.36: Comparison between the MPC and I-MPC sensitivity: i_q steady state current, $i_{q_{ref}} = 1.5A$, $R = 24\Omega$, $L_d = 0.5H$, $L_q = 0.2$.

the reference and in steady state condition it reaches zero offset.

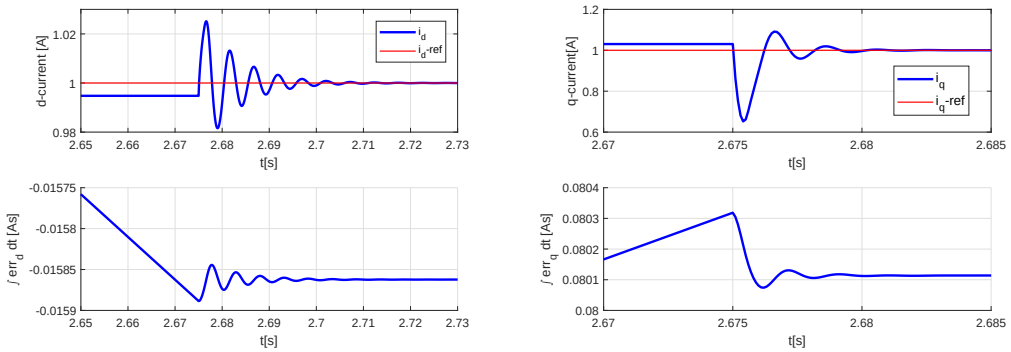
4.3.6 R, L_d and L_q mismatches

Another interesting test is to change all the electric parameters and observe the effect of the two control scheme. In this case, the regulation weight indexes are slightly modified for both the control scheme. In fact, reducing L_d, L_q and increasing R at the same time, the control action need to be adjusted, and the previous value of r_1 r_2 seems to be too small. It is set $r_1 = r_2 = 1e-5$.

In Figure 4.35 and Figure 4.36 it is shown the comparison between the MPC and the I-MPC, assuming that both L_d and L_q are halved, while the stator resistance is increased by 40%. As can be observed, *I-MPC can deal with these mismatches* (Figure 4.35(b),Figure 4.36(b)), while the traditional Model Predictive Control scheme commits a significant steady state error in both axes (Figure 4.35(a),Figure 4.36(a)).

4.3.7 Current integral error

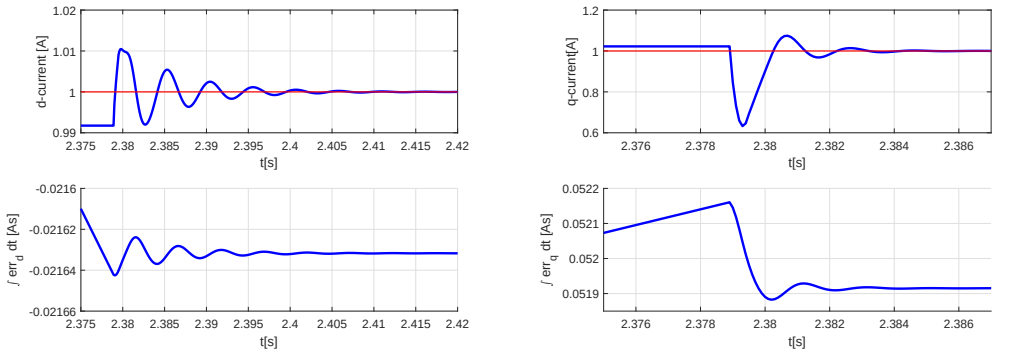
This section aims to show the difference between the MPC and the I-MPC in terms of the integral error. In order to perform significant simulations, the current steady state with $i_{d,ref} = i_{q,ref} = 1$ is considered. The test starts with the MPC algorithm. After a while, a switch from the MPC to the I-MPC strategy is carried out. During simulations, it is computed the integral error of both the d-q axis currents. In Figure 4.37 both the motor d-q axis inductances are halved. As can be observed, *I-MPC can deal with this mismatch*, in fact this case, the integral error caused by the MPC is arrested after the switching to I-MPC strategy, while during the MPC operation the integral error grows linearly, this because the error is constant with respect the reference.



(a) MPC: i_d current at $\omega_m = 32rad/s$. **(b)** I-MPC: i_q current at $\omega_m = 32rad/s$.

Figure 4.37: MPC to I-MPC switch: L_d and L_q mismatch.

Finally, in Figure 4.38 all the mismatches are considered, by halving the d-q axis inductances and doubling the resistive term. It is possible to notice that, after the switch, *I-MPC compensates effectively the steady state error in the reference tracking*. Moreover, the rise of the integral error induced by the MPC is stopped. This means that I-MPC grants an integral action and so an unbiased tracking.



(a) MPC: i_d current at $\omega_m = 32rad/s$. **(b)** I-MPC: i_q current at $\omega_m = 32rad/s$.

Figure 4.38: MPC to I-MPC switch: L_d,L_q and R mismatch.

Chapter 5

Experimental results

5.1 Introduction to experimental tests

The test bench is composed of a master motor (PMSM synchronous electric machine) directly connected to the motor under test (SyRM) with the same nominal machine data used in simulation, as already presented in Chapter 3. The drive has been tested experimentally using dSPACE MicroLabBox which provides the six commands to the inverter. After initialization, the program reads the phase currents and the mechanical speed from the encoder placed on the motor shaft. It proceeds making Clarke and Parks transformations in order to implement the current MPC/I-MPC and calculating the voltage references for the Space Vector PWM (SVPWM), which gives the duty cycle signals for the six gates of the inverter (see Figure 5.1. Both current MPC/I-MPC have been implemented with a horizon length of N = 3. The choice of using this prediction horizon is a compromise between stability and computational cost. It is worth to remember that the longer N, the higher the computational effort, because matrix order grows as long as we extend the prediction horizon. However, the stability of the control loop is related to this parameter. Weight matrices $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{S}$ have been tuned for both the control schemes, achieving a good compromise between dynamics response during transients and steady state stability. Tuning effort has been minimized by acting only on the regulation matrix $\mathbb{R}$ and keeping constant the others. Performances of the two schemes are compared in steady state and dynamics conditions. As represented in Figure 5.1, the master motor maintains the speed at $300rpm$ and a d-q current reference of 1.5 A is given to the SyRM. In order to test the effectiveness

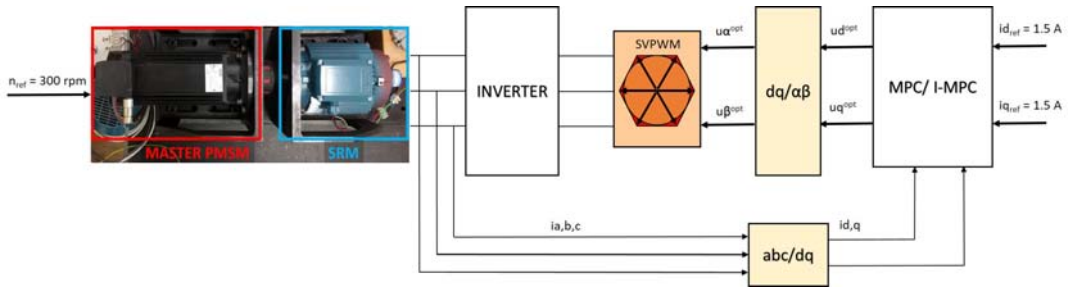


Figure 5.1: Scheme of the experimental test implementation.

of the I-MPC, a change of the motor parameter is forced to the control, and it is observed the difference between the MPC and the I-MPC. Overall, we concentrate the experimental tests on the mismatch parameters; the aim is to demonstrate that the I-MPC can afford these mismatches and keep the reference tracked without offset. For both control schemes, it has been chosen the following weight indexes for d and q axes respectively:

$$q_1 = q_2 = 1$$
$$s_1 = s_2 = 2$$
$$r_1 = 8e-5$$
$$r_2 = 8e-5$$

These parameters differ from those that have been used in simulation tests. In fact, simulations are based on simplify and linear model of the motor, and the inverter is modelled as a simple delay in s-domain. The system behavior is quite different in experiments, and parameters have to be changed to find the optimal configuration.

5.2 Current step dynamics

First, it has been evaluated the current step response along the d and q axes.

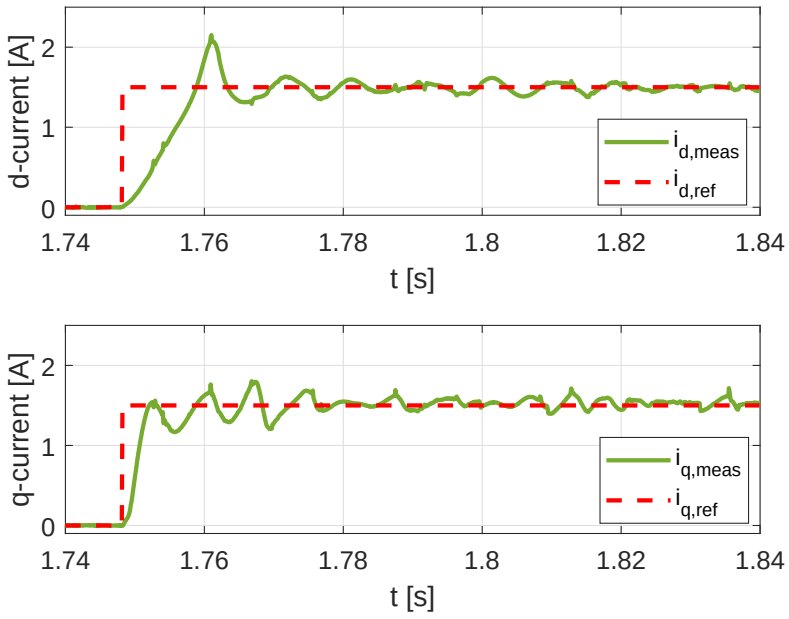


Figure 5.2: Current dynamics: step response using the *MPC* scheme.

For the MPC control scheme, observe the Figure 5.2. It is possible to notice that the best tuning of weight parameters, which is a compromize between low ripple in steady state and overshoot in dynamics, does not eliminate oscillating

step responses. Comparing with Figure 5.3, which reports the I-MPC d-q current step response, it is clear that the MPC can't deal with variable inductances. In fact, at the operation point of 1.5 A, the inductances vary with respect the nominal value. In particular, after dynamics in Figure 5.3 the I-MPC achieves a zero offset in the q-axis current, while in Figure 5.2 a difference between reference and q-axis current measurements appears. Furthermore, by comparing the two scheme dynamics, it results that the I-MPC can be tuned in easier ways, obtaining low ripple and overshoot at the same time.

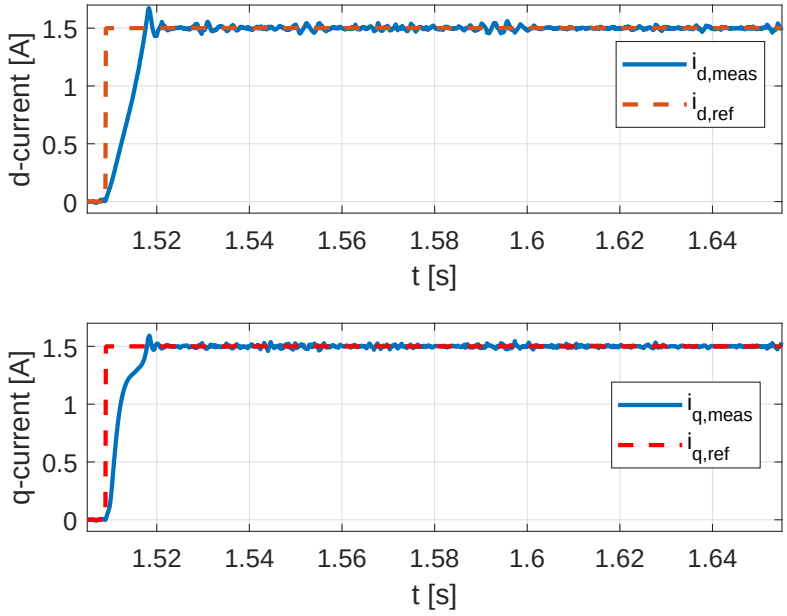


Figure 5.3: Current dynamics: step response using the *I-MPC* scheme.

In Figure 5.4 it is shown a zoom of the current step response using the I-MPC control scheme. Referring to Figure 3.7, the possible real value of inductances at $I_{ref} = \sqrt{i_d^2 + i_q^2} = 2.12A$ are:

$$L_d = 0.8H; \quad L_q = 0.2H$$

so the electric time constant become:

$$\tau_{ed} = 50ms; \tau_{eq} = 12.5ms$$

Observing Figure 5.4 we can derive that the rising time for the d-axis current is 11 ms approximately, about a fifth of τ_{ed}, while the rising time of the q-axis current is 9 ms, less than τ_{eq}. This means that this control scheme can compete with similar predictive control strategy, moreover assuring a correct current tracking. When steady state condition is reached, the motor phase currents in time domain are almost sinusoidal, and they are reported in Figure 5.5 for the MPC and in Figure 5.6 for the I-MPC. Due to the previous comment on the q-axis current using the MPC, it is possible to notice that phase currents of the MPC scheme have a higher peak value, so the reference is not correctly tracked.

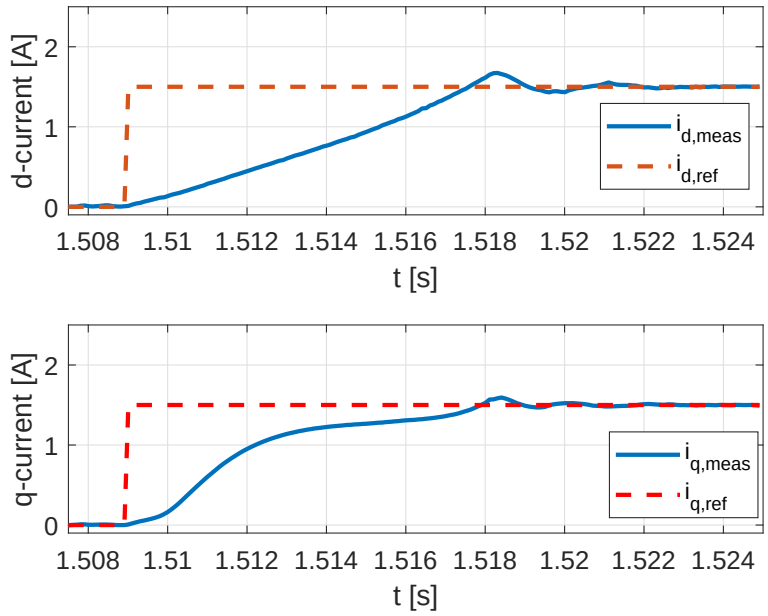


Figure 5.4: I-MPC: Zoom of Figure 5.3.

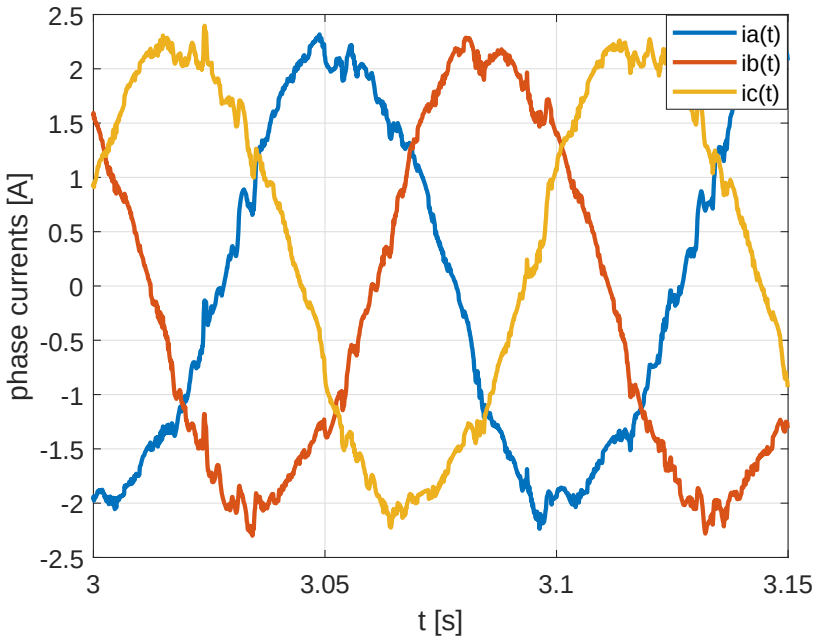


Figure 5.5: Stator phase currents using the MPC.

Another comparison that can be done in dynamics involves the d-q voltages applied by the inverter. The confrontation between the MPC and the I-MPC is pointed out in Figure 5.7 and in Figure 5.8.

The dynamics using the two scheme appears to be very similar, while when steady state is reached the I-MPC shows lower ripple on both axes.

Now it is interesting to compare the two control scheme considering current

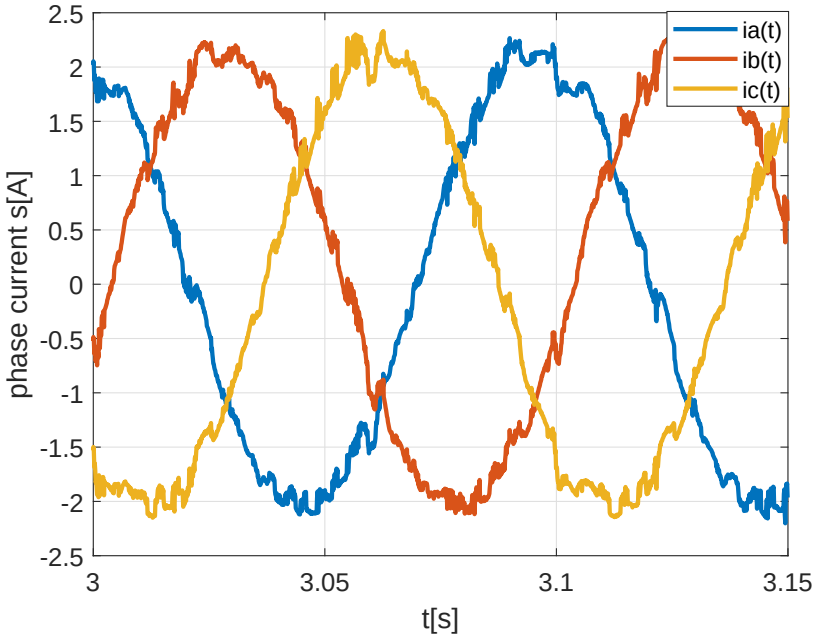


Figure 5.6: Stator phase currents using the I-MPC.

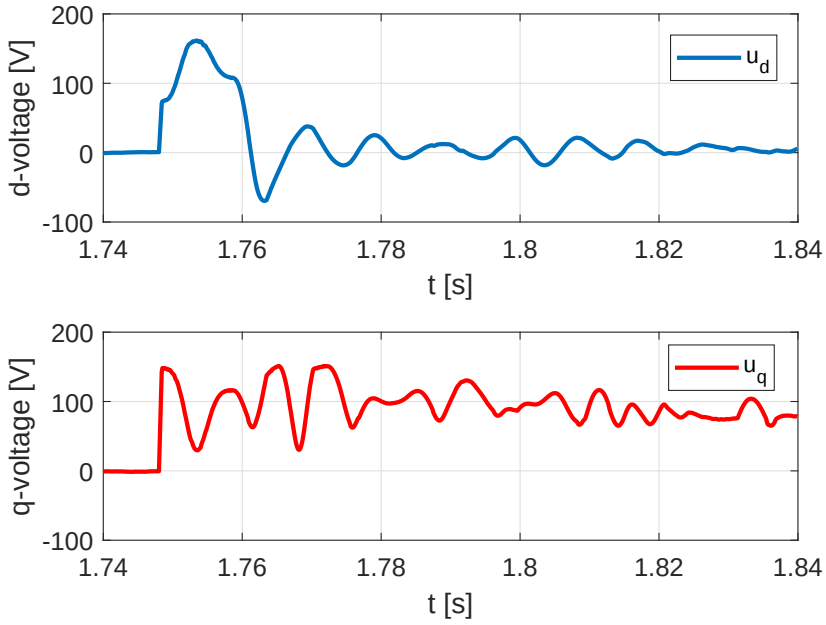


Figure 5.7: Output voltages in d-q reference frame using the MPC.

dynamics and imposing to the controls a mismatch parameter. In order to simulate the iron saturation, the inductances are augmented because the real value will be smaller. In Figure 5.9 and Figure 5.10 are reported the current step response assuming $L_d = 2\ H$. Using the MPC, an error arises in the q-axis current. It is quite obvious that the value of the error depends on the motor speed and also on the parameter mismatch. Here it is considered a d-axis inductance of 2 L_d.

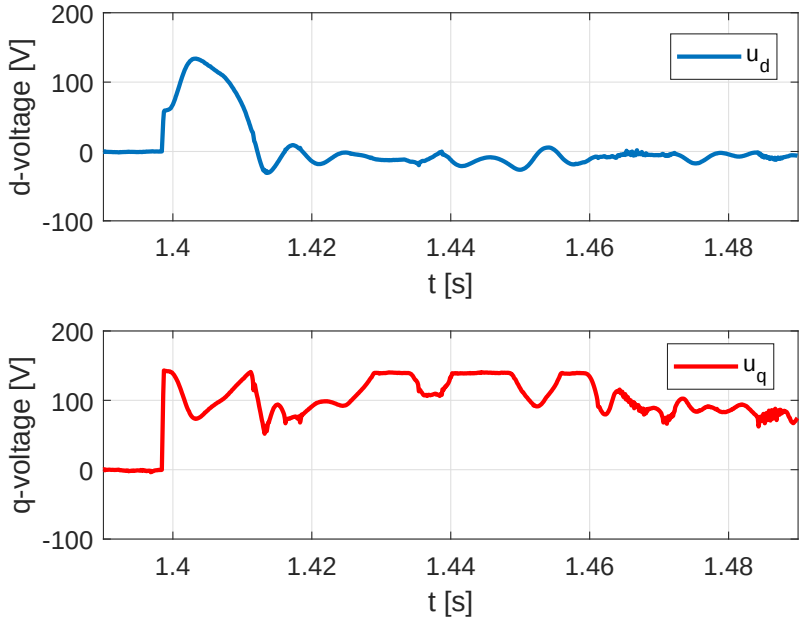


Figure 5.8: Output voltages in d-q reference frame using the I-MPC.

On the other hand, the I-MPC presents a good dynamics compared with the one without mismatch from nominal parameter. It exhibits d-axis current with an increased overshoot and steady state is reached in more time compared to the one with nominal parameters. However, the I-MPC isn't affected by steady state error and it well-tracked the current reference.

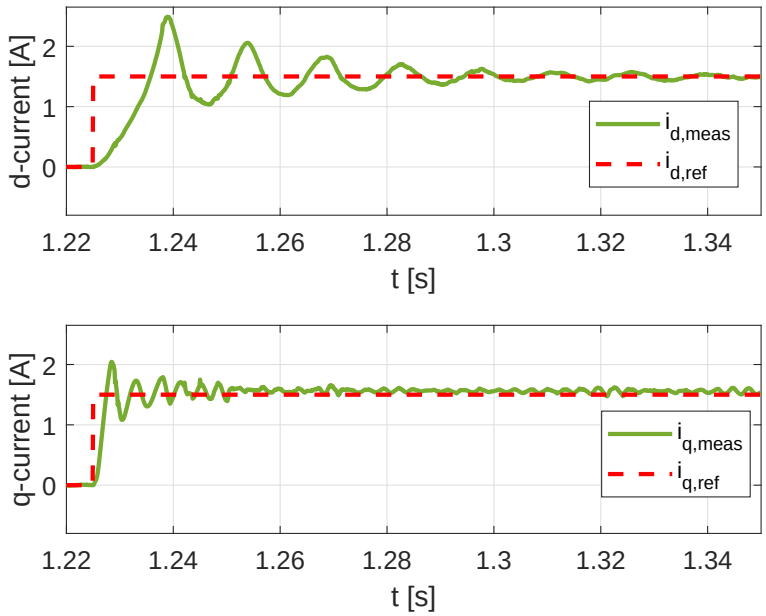


Figure 5.9: Current dynamics: step response using the *MPC* scheme, with control parameter $L_d = 2\ H$.

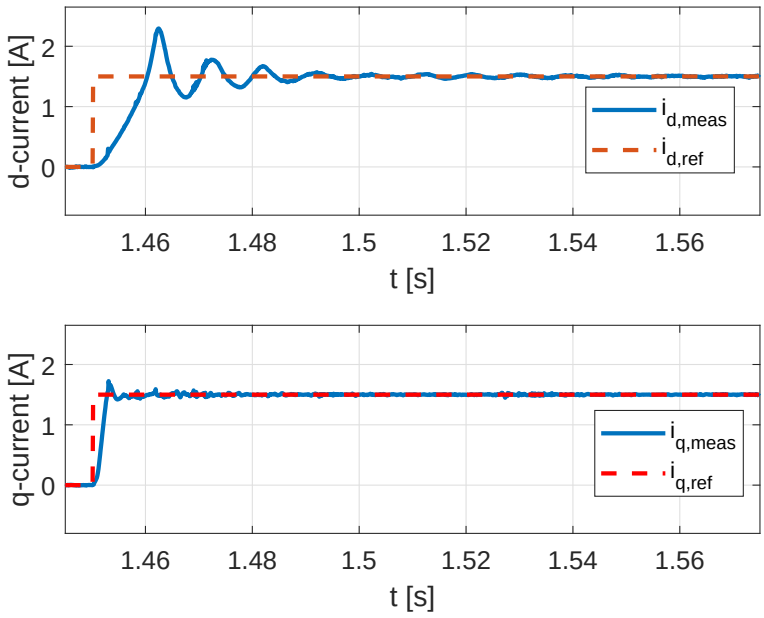


Figure 5.10: Current dynamics: step response using the *I-MPC* scheme, with control parameter $L_d = 2\ H$.

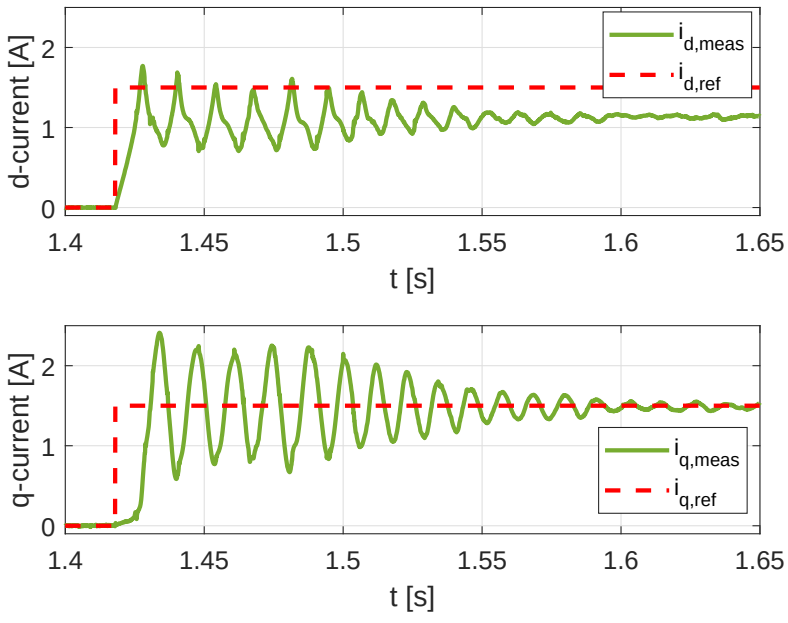


Figure 5.11: Currents dynamics: step response using the *MPC* scheme, with control parameter $L_q = 0.8\ H$.

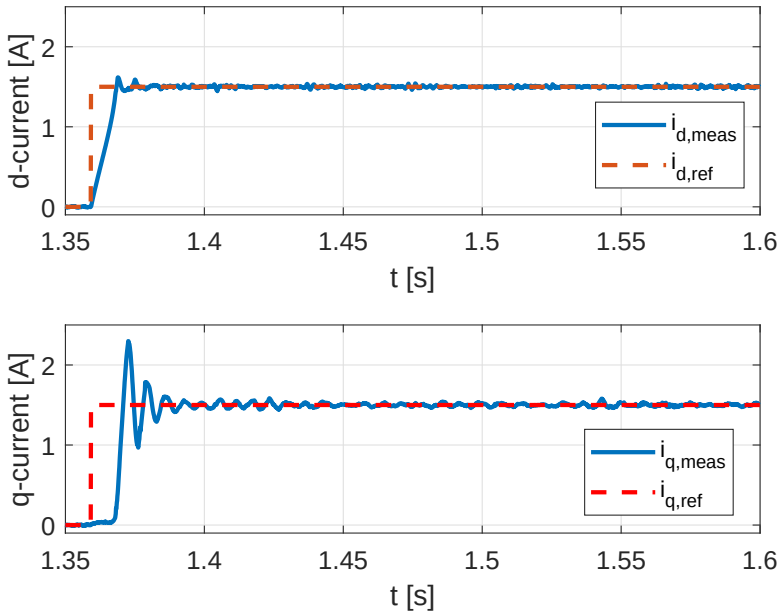


Figure 5.12: Current dynamics: step response using the *I-MPC* scheme, with control parameter $L_q = 0.8\ H$.

Similar considerations can be done in case of q-axis inductance and resistance mismatches. In Figure 5.11 and Figure 5.12 it is given to the control a parameter $L_d = 0.8 \quad H$ of the q-axis inductance. In particular, in Figure 5.11 it is clear that the MPC scheme registered an offset in the d-axis current. Also, the dynamics of the two axes is compromised, showing extensive oscillations. While, the step current responses using I-MPC (Figure 5.12) don't suffer from offset in steady state, and the dynamics is comparable to the one with nominal parameters. It is interesting to notice that, in both control schemes, a so high mismatch of q-axis inductance cause a delay that affects the rising time.

Another forced mismatch parameter analysed is the resistive one. In Chapter 4 it has been shown with simulations that a mismatch of the resistive parameter affect both d and q axes using the MPC. Here, in Figure 5.13, this effect is confirmed. Furthermore, the q-axis current presents continuous oscillations. One more time, the I-MPC confirm the robustness to resistance variations, as reported in Figure 5.14. The current response is comparable with the one that has the nominal parameters.

Finally, putting all together, an experimental test considering all the possible mismatches is considered in Figure 5.15 and in Figure 5.16 for the MPC and I-MPC respectively. Also in this case the I-MPC shows a good step current response and in steady state it tracks the reference, while the MPC scheme presents oscillations and an offset between the reference and the actual currents.

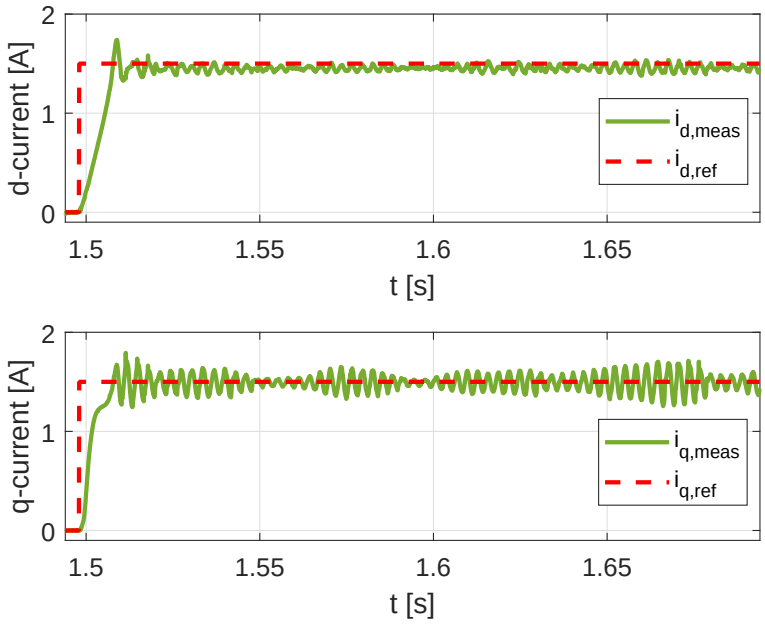


Figure 5.13: Current dynamics: step response using the *MPC* scheme, with control parameter $R = 8\ \Omega$.

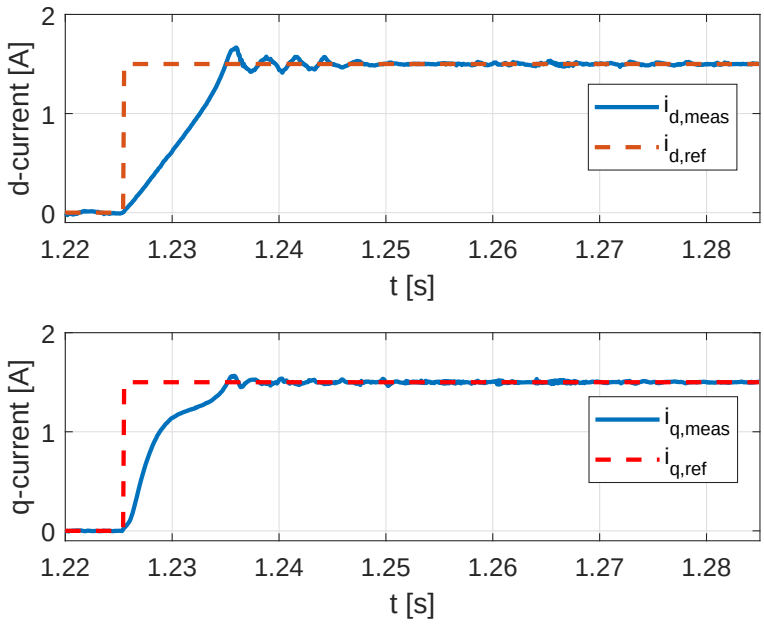


Figure 5.14: Current dynamics: step response using the *I-MPC* scheme, with control parameter $R = 8\ \Omega$.

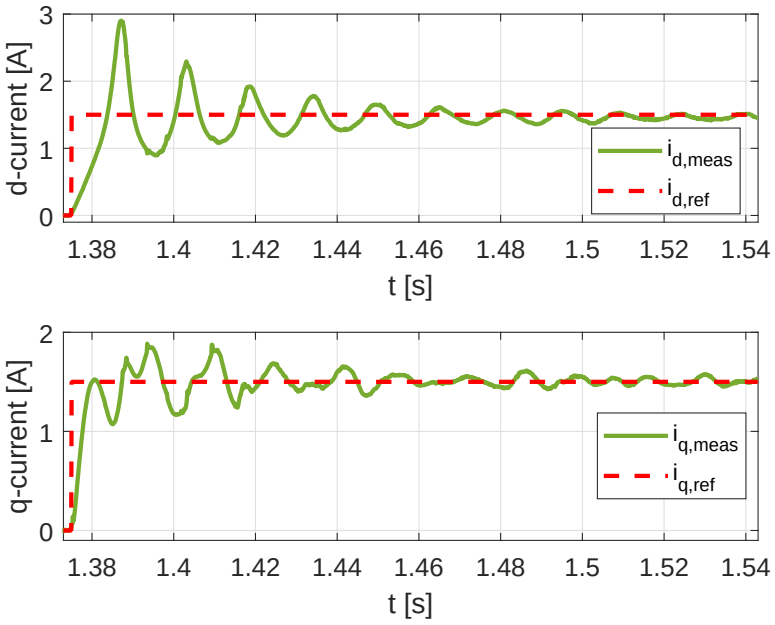


Figure 5.15: Current dynamics: step response using the *MPC* scheme, with control parameter $L_d = 2\ H$, $L_q = 0.8\ H$, $R = 8\ \Omega$.

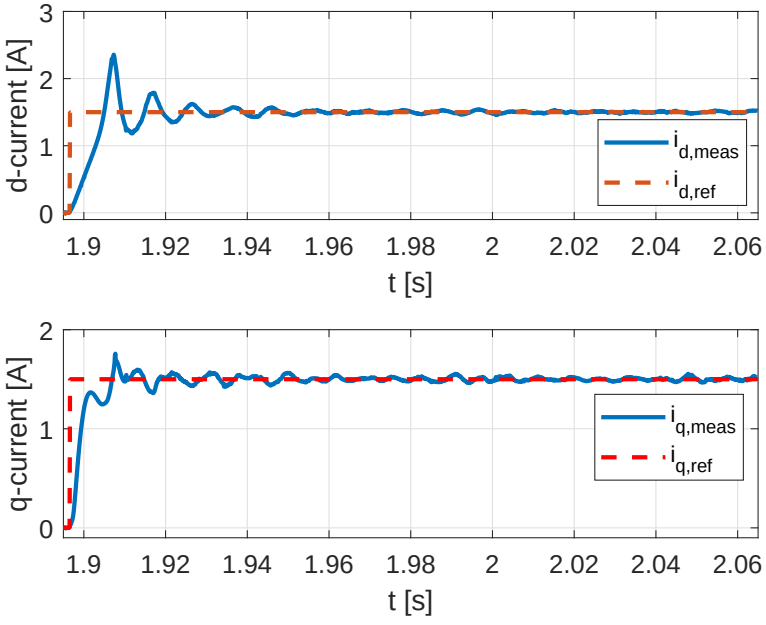


Figure 5.16: Current dynamics: step response using the *I-MPC* scheme, with control parameters $L_d = 2\ H$, $L_q = 0.8\ H$, $R = 8\ \Omega$.

5.3 Steady state tests

It is now presented the crucial advantage of using the I-MPC algorithm. Tests are performed in steady state conditions. It is considered the current reference $I_{d_{ref}}* = I_{q_{ref}}* = 1.5A$ and the motor speed is imposed by the master motor at $300rpm$. The motor parameter are changed online to the control after a while it is reached the steady state operation. The aim is to demonstrated how parameter mismatches can affect the tracking. First, it is considered to switch online the d-axis inductance value which the control uses for solving the optimization problem. The value changes from $L_d = 1\ H$ to $L_d = 2\ H$, simulating the iron saturation.

In Figure 5.17 it is shown the sensitivity on the d-axis inductance using the MPC. In particular, the inductance is doubled in the MPC control at the instant 2.6s.

As can be seen, the nominal parameters already causes an offset, that is conspicuous on the d-axis current. It is also reported the mean value of currents, before and after the d-axis inductance step change. It is possible to compute the error between the real mean value and the reference as:

```
err_id = abs(1.5 - sum(id_meas(t_start:t_end))/length(id_meas(t_start:
    t_end)))*1000
err_iq = abs(1.5 - sum(iq_meas(t_start:t_end))/length(iq_meas(t_start:
    t_end)))*1000
```

where t_{start} and t_{end} is the time interval considered for computing the mean value. So, before the change of L_d it holds:

$$i_{d_{err}} = 44.5 \quad mA$$
$$i_{q_{err}} = 2.6 \quad mA$$

After the online changes it results:

$$i_{d_{err}} = 5.4 \quad mA$$
$$i_{q_{err}} = 74.9 \quad mA$$

It is clear that in presence of d-axis inductance parameter mismatches, the MPC doesn't guarantee zero offset in steady state. The error appears in both axes; in particular, the mismatch affects the q-axis current due to the cross coupling, as confirm of simulations results.

On the other hand, observe the Figure 5.18. It is reported the same procedure just presented, but it is implemented the I-MPC algorithm. At time $t = 1.94\ s$ the d-axis inductance is doubled. This test highlights the effectiveness of the control scheme integral action. The control chases and keep the reference tracked without any offset. This worth for both d and q axes. The only effect is an increasing value of the d-current ripple.

Analogous experimental test has been involved the q-axis inductance. In this case the value that the control uses to compute optimal voltages is augmented online up to $L_q = 0.8\ H$. Using the MPC scheme, the measured currents are

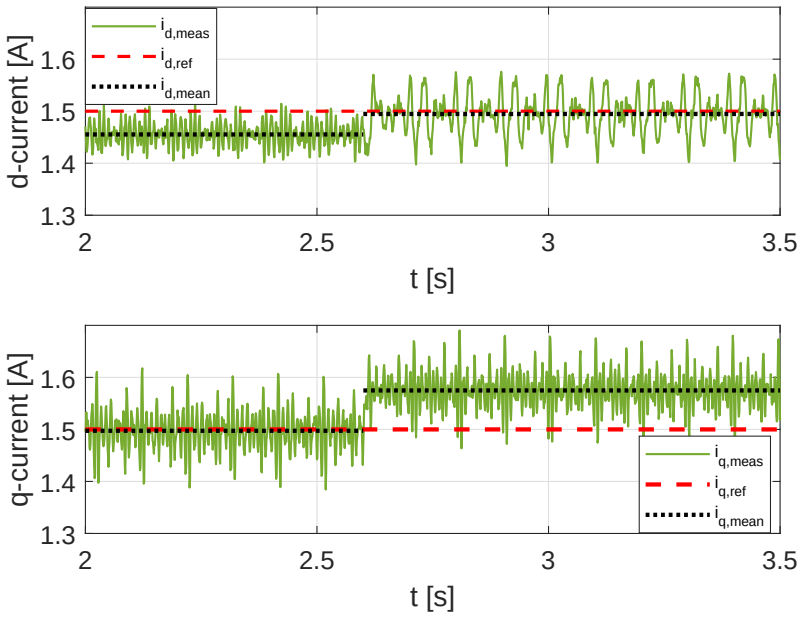


Figure 5.17: MPC steady state: step change of the d-axis inductance.

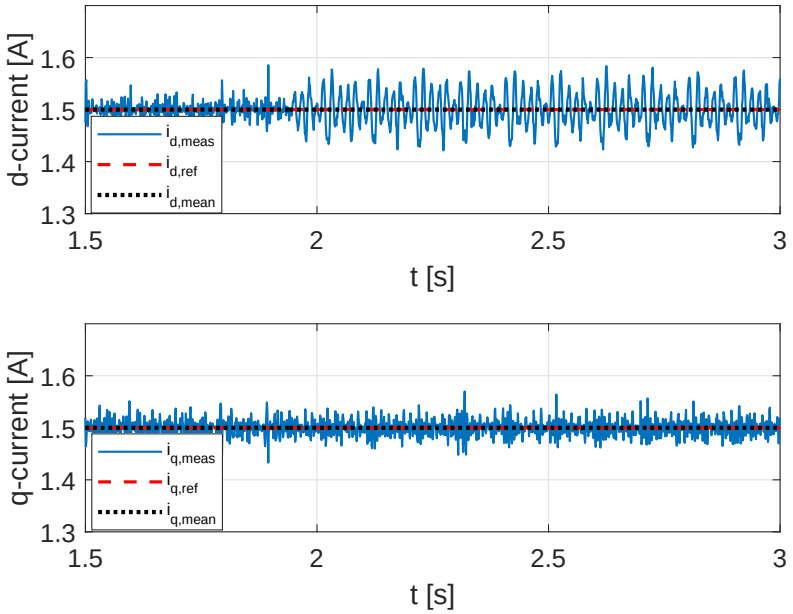


Figure 5.18: I-MPC steady state: step change of the d-axis inductance.

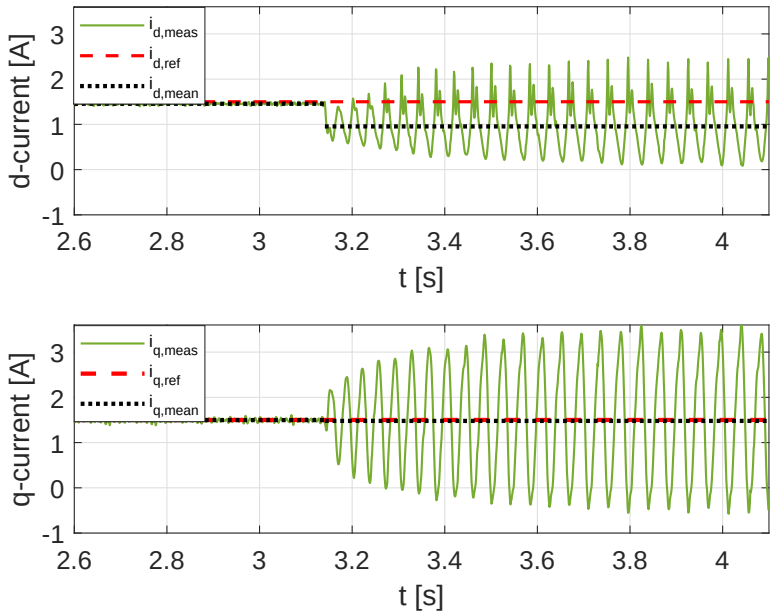


Figure 5.19: MPC steady state: step change of the q-axis inductance.

reported in Figure 5.19. This mismatch affects both d and q axis currents. Errors before the step change are:

$$i_{d_{err}} = 44.5 \quad mA$$
$$i_{q_{err}} = 2.6 \quad mA$$

While, after the instant of $3.14s$, errors between actual currents and reference grow significantly, especially for the d-current:

$$i_{d_{err}} = 545.4 \quad mA$$
$$i_{q_{err}} = 20.9 \quad mA$$

Also here, it is confirmed the simulation test that, due to the cross coupling term in the d-axis voltage equation, the d-current is particularly affected by an offset. Conversely, the Figure 5.19 shows the d and q current trends by implementing the I-MPC. At time instant 2.25 s the same step change of L_q is given to the control. Also in this case there are good evidences of the I-MPC robustness. The mean current coincides at each instant with the reference one. It is demonstrated that this control scheme doesn't involve any offset on the currents. Here, after the switch of the q-inductance value, a greater ripple appears on the q-axis current; this dually happen in the d-axis current considering a mismatch of L_d. But overall the reference is always maintained tracked.

The last mismatch analysed is the one that involves stator phase resistance. To simulate the heating of the motor, a lower value with respect to the nominal one is given to the control. It is considered a halved value $R = 8\ \Omega$ of the nominal

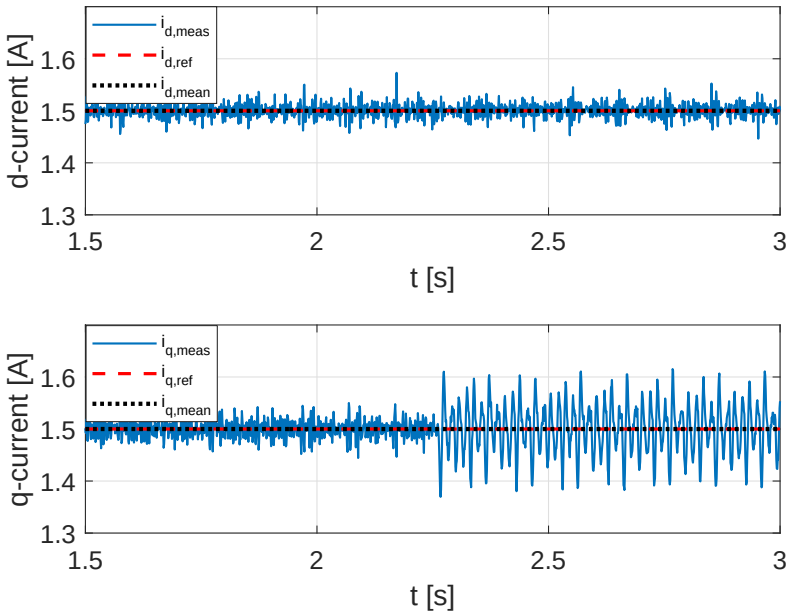


Figure 5.20: I-MPC steady state: step change of the q-axis inductance.

parameter. Using the MPC algorithm, the step change of R occurs at time instant 3 s. The achieved results are reported in Figure 5.21. Also in this case, it is confirmed that the latter control scheme can't afford any motor mismatch parameter. The error before the resistance step change is the same of previous cases. After switching, errors become:

$$i_{d_{err}} = 55.8 \quad mA$$
$$i_{q_{err}} = 7.5 \quad mA$$

This demonstrated that both the currents are affected by an offset, and its value is not equally proportional, other mismatches are already present. In fact, the d-axis nominal inductance previously causes an offset on the d-axis current.

In Figure 5.22 it is shown another benefit of using the I-MPC. The same step change of resistance is here applied, but the tracking isn't affected by offset at all. Furthermore, a halved value of the resistance doesn't change the ripple in both the d and q axis current. So it is confirmed the robustness of the I-MPC against stator phase resistance mismatches.

Finally, both the control are tested considering all the mismatches. The comparison is pointed out observing the Figure 5.23 and Figure 5.24. The I-MPC manifests robustness to all the possible incorrect parameters. While, the MPC pointed out offsets in both the d and q axis current. In conclusion, we can affirm that this MPC with integral action is insensitive to motor parameter mismatches, and it is capable to maintain the correct tracking reference without shows any deviations.

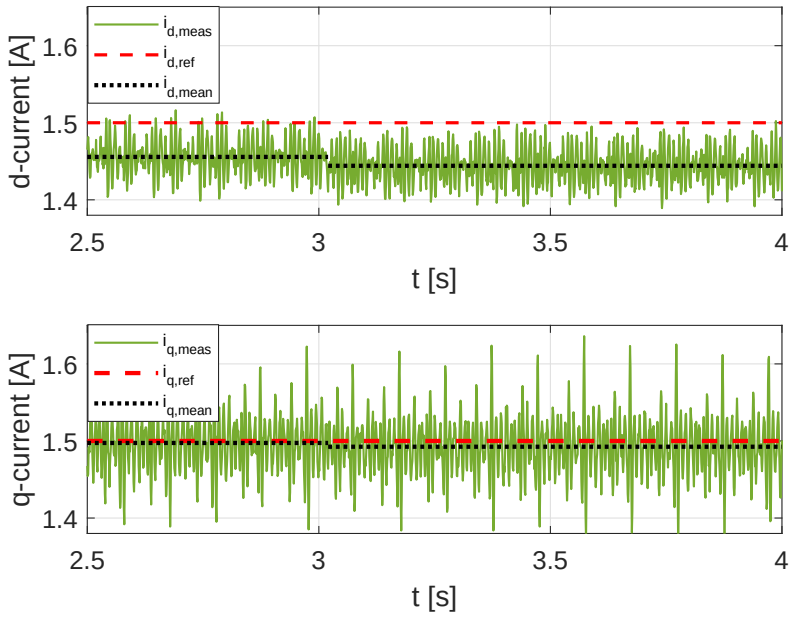


Figure 5.21: MPC steady state: step change of the stator phase resistance.

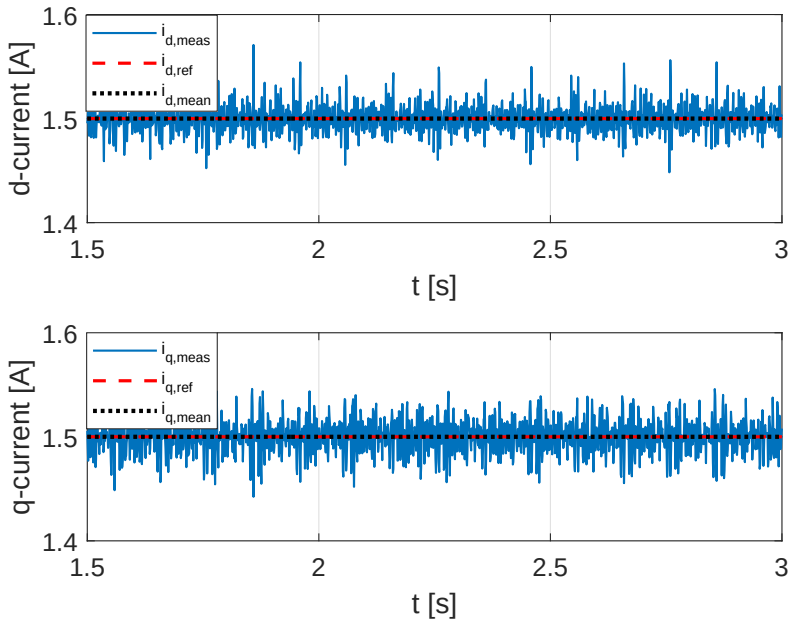


Figure 5.22: I-MPC steady state: step change of the stator phase resistance.

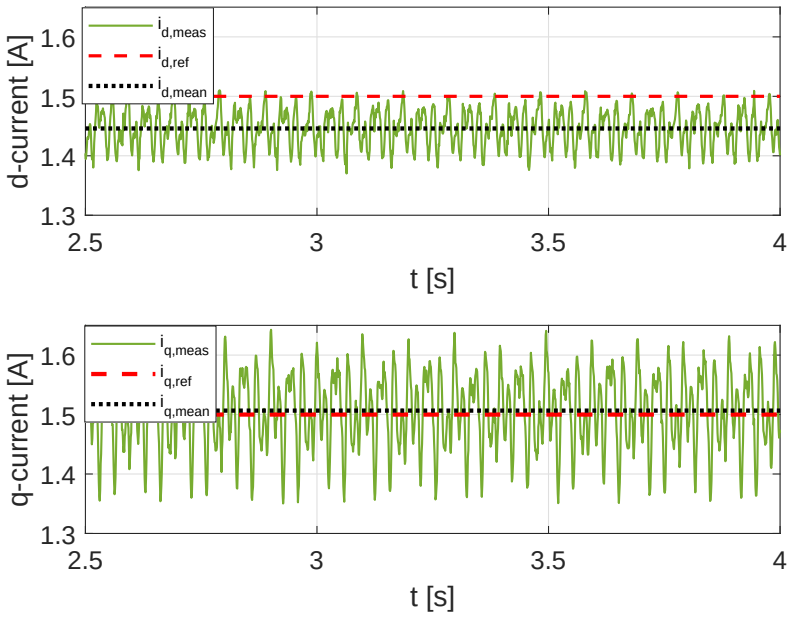


Figure 5.23: MPC steady state with all parameter mismatches.

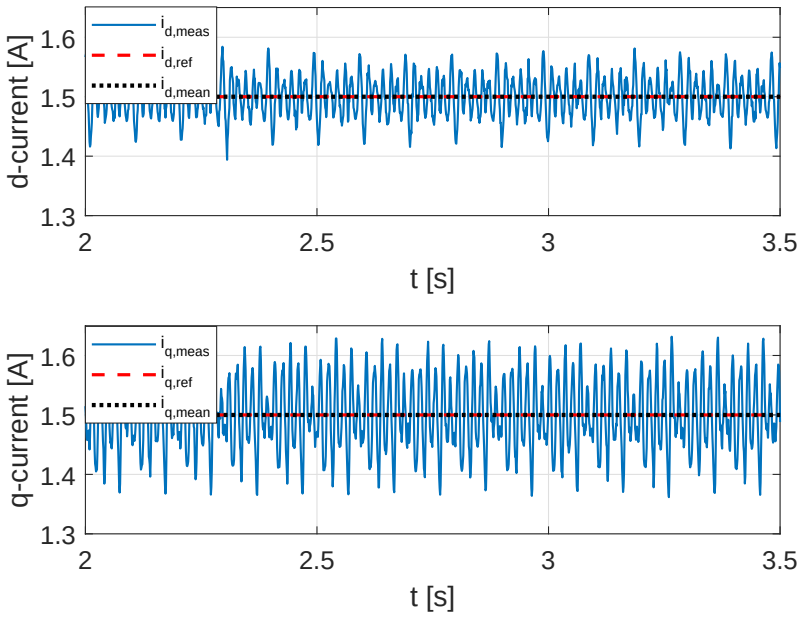


Figure 5.24: I-MPC steady state with all parameter mismatches.

5.4 Note on the prediction horizon

The prediction horizon N plays a key role in matter of the control stability. In general, the larger N, the better the stability. However, as already mentioned, increasing the N value causes a more than linear increase of the computational cost. In this work it has been chosen a prediction horizon length of three steps. This choice entails the inversion of a 6-square matrix in order to solve the optimization problem. As final step of this thesis it has been tested how the I-MPC behaves with a shorter prediction horizon of N =2. It has been considered the same current step dynamics presented in previous section. The comparison between the two prediction horizon lengths is presented in Figure 5.25 and in Figure 5.26. It is interesting to notice that, with N equal to three, the current dynamics exhibit lower oscillations and overshoot. Observing the Figure 5.25, with a prediction horizon equal to two, the control action is maintaned; nevertheless currents present higher overshoot and the steady state is reached after some oscillations.

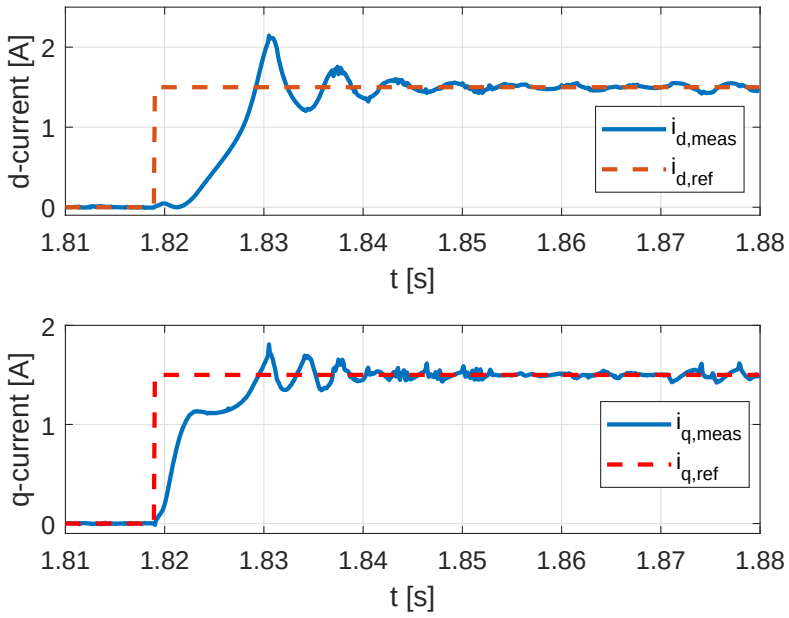


Figure 5.25: I-MPC current step dynamics with a prediction horizon length N = 2.

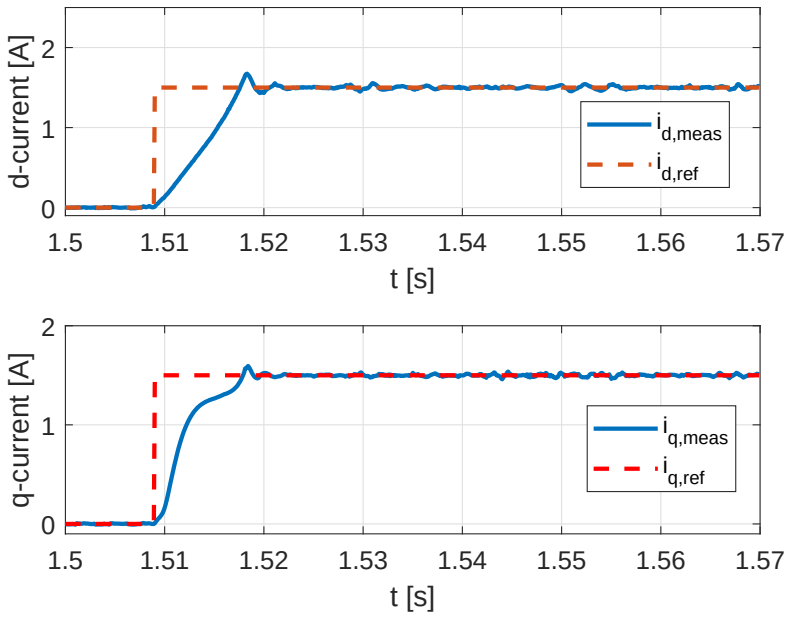


Figure 5.26: I-MPC current step dynamics with a prediction horizon length N = 2.

Conclusions and future developments

In this work a novel MPC strategy with an *Integral Action* has been proposed in the electric drives field *for the first time*. The new formulation of the motor state equation is based on the use of incremental current vectors for both the input voltage and the current state. This makes possible to overcome issues caused by the traditional current MPC approach. In particular, the steady state error is compensated by the integral action on the currents.

Theory validation is achieved by comparing the two scheme simulations in different operating conditions. In particular, it has been shown how the controls behave in case of parameter mismatches. *The proposed current I-MPC is able to compensate the steady state error due to mismatches.* This is proved by the fact that the integral of the tracking error generated by the traditional MPC is stopped by the I-MPC during simulation tests.

Experimental tests performed on a Synchronous Reluctance Motor confirm the advantages of I-MPC implementation. The current dynamics are comparable between the two control schemes in term of rising time. However, the use of I-MPC assures the benefits of a correct current tracking. Furthermore, the experimental tests pointed out the robustness of the control algorithm, because so high step changes of the inductance parameters does not occur in real operations. Nevertheless, the I-MPC is able to maintain the control and keep the reference tracked. It is important to notice that the I-MPC does not need any state observer to estimate disturbances and parameter mismatches. *This permits to simplify the control scheme, while the computational effort is comparable to the MPC.*

Future works will focus on the implementation of the control scheme in different Synchronous Motors, such as the PMSM. The presence of the magnets will give the opportunity to focus the analysis on the model disturbance, represented by the term $\mathbf{h}$. Also, it will be interesting to handle the constrained optimization problem, including the inequality constrains of voltage and current limits. The challenge is to implement this problem in fast dynamics application, such as electrical drives. Moreover, a more precise stability study will be performed; lower values of the prediction horizon length implies lower computational cost, thus the control algorithm results simpler to implement. However, since the MicroLabBox permits very high computational power, an interesting analysis can be consider a higher prediction horizon lenght (for instance equal to four, six ...). Finally, it will be investigated all the high speed operating region, in order to test the robustness and the performance of the control in the flux weakening regime operation.

Bibliography

[1] D.Q. Mayne et al. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789 –814. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(99)00214-9. URL: http://www.sciencedirect.com/science/article/pii/S0005109899002149.

[2] E.F. Camacho and C. Bordons. *Model Predictive Control*. Ed. by Springer. Second Edition. 2007. Chap. 1, p. 3. ISBN: 978-1-85233-694-3.

[3] S. Bolognani et al. "Design and Implementation of Model Predictive Control for Electrical Motor Drives". In: *IEEE Transactions on Industrial Electronics* 56.6 (2009), pp. 1925–1936. ISSN: 0278-0046. DOI: 10.1109/TIE.2008.2007547.

[4] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. "Generalized predictive control—Part I. The basic algorithm". In: *Automatica* 23.2 (1987), pp. 137 –148. ISSN: 0005-1098. DOI: https://doi.org/10.1016/0005-1098(87)90087-2. URL: http://www.sciencedirect.com/science/article/pii/0005109887900872.

[5] P. Cortes et al. "Predictive Control in Power Electronics and Drives". In: *IEEE Transactions on Industrial Electronics* 55.12 (2008), pp. 4312–4324. ISSN: 0278-0046. DOI: 10.1109/TIE.2008.2007480.

[6] F. Wang et al. "Model predictive control for electrical drive systems-an overview". In: *CES Transactions on Electrical Machines and Systems* 1.3 (2017), pp. 219–230. ISSN: 2096-3564. DOI: 10.23919/TEMS.2017.8086100.

[7] H. Chen et al. "A robust predictive current control for PMSM based on extended state observer". In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2015, pp. 1698–1703. DOI: 10.1109/CYBER.2015.7288202.

[8] Silverio Bolognani. *"Electric Drives" Course Notes*. 2009.

[9] dspace digital signal processing and control engineering GmbH. *dSpace MicroLabBox product information*. URL: https://www.dspace.com/shared/data/pdf/2018/dSPACE_MicroLabBox_Product_information_2018_02_E_180209.pdf.

Appendices

Appendix A

MPC Matlab code

The Matlab code of the MPC with prediction horizon of N = 3 is reported below. The computation of the inverse matrix $H^{-}1$ is here indicated with the command $inv(H)$. However, the implemented code include the parametric inversion of the matrix, in order to avoid high computational cost.

```
function [Uk,duty_a,duty_b,duty_c,id_p,iq_p] = mpc(w_me,theta_me, id,
    iq,id_rif,iq_rif, U_DC,Ld, Lq,R,q1,q2,r1,r2,s1,s2)

% MPC equality constrained, predictive horizon N =3
%current control of a SyRM

flux_mg = 0;
Ts = 100e-6;

%x(k+1) = A*x(k) + B*u(k) + h
% C = [1 0; 0 1];
a11 = 1 - Ts*R/Ld;
a12 = w_me*Ts*Lq/Ld;
a21 = -w_me*Ld/Lq*Ts;
a22 = (1- Ts*R/Lq);
A = [a11 a12; a21 a22];

b11 =  Ts/Ld;
b12 = 0;
b21 = 0;
b22 = Ts/Lq;
B = [b11 b12; b21 b22];

h = [0; -w_me*flux_mg*Ts/Lq];

persistent U_old;
% set original U_old to zero the
% first time this function is invoked
if isempty(U_old)
  U_old=[0;0;0;0;0;0];
end

%A1 = [ A; A^2; A^3];
%A
 aa11 = a11;
 aa12 = a12;
```

```
 aa21 = a21;
 aa22 = a22;
%A^2
aa31 = a12*a21 + a11*a11;
aa32 = a11*a12 + a12*a22;
aa41 = a11*a21 + a21*a22;
aa42 = a12*a21 + a22*a22;
%A^3
 ax1 = a12*a21 + a11*a11;
 ax2 = a11*a21 + a21*a22;
 ax3 = a22*a22 + a12*a21;
 ax4 = a11*a12 + a12*a22;

 aa51 = a11*ax1  + a12*ax2;
 aa52 = a12*ax3 + a11*ax4;
 aa61 = a21*ax1 + a22*ax2;
 aa62 = a22*ax3 + a21*ax4;

A1 = [aa11 aa12; aa21 aa22; aa31 aa32; aa41 aa42; aa51 aa52; aa61 aa62
     ];
B1 = [B zeros(2,4); (A*B +B) B zeros(2,2); (A^2*B +A*B +B) (A*B +B) B
     ];
C1 = [B zeros(2,4); A*B B zeros(2,2); A^2*B A*B B];
D = [eye(2,2) zeros(2,4); A eye(2,2) zeros(2,2); A^2 A eye(2,2)];
h1 = [h; h; h];

%weight matrices
Q1 = [q1 0 0 0 0 0;
      0 q2 0 0 0 0;
      0 0  q1 0 0 0;
      0 0  0  q2 0 0;
      0 0 0 0 0 0;
      0 0 0 0 0 0];

S1 = [0 0 0 0 0 0;
      0 0 0 0 0 0;
      0 0  0 0 0 0;
      0 0  0  0 0 0;
      0 0 0 0 s1 0;
      0 0 0 0 0 s2];

R1 = [r1 0 0 0 0 0;
      0 r2 0 0 0 0;
      0 0  r1 0 0 0;
      0 0  0 r2 0 0;
      0 0 0 0 r1 0;
      0 0 0 0 0 r2];

%current acquisistion
xk = [id;iq];

%reference vector
rif = [id_rif;iq_rif;id_rif;iq_rif;id_rif;iq_rif];

Y = rif- A1*xk - C1*U_old - D*h1;
```

```
%Hessian matrix
H = 2*(B1'*Q1*B1 +R1 +B1'*S1*B1);
d = 2*((Y'*Q1*B1) + (Y'*S1*B1))';

%Solution of the unconstrained problem
deltaU = inv(H)*d;

%RH
deltaUk = deltaU(1:2);
U_prec = U_old(1:2);
Uk1 = U_prec + deltaUk;

%Voltage limits
Ulim_star = U_DC/sqrt(3);
if sqrt(Uk1(1)^2 + Uk1(2)^2)>= Ulim_star && abs(Uk1(1)) >= 0.00001

      if      Uk1(1) >= 0 && Uk1(2) >=0

              Uk1(2) = Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1))
                  ^2));
              Uk1(1) = sqrt(Ulim_star^2 - Uk1(2)^2);
      end

         if Uk1(1) <= 0 && Uk1(2) >=0

              Uk1(2) = -Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1)
                  )^2));
              Uk1(1) = -sqrt(Ulim_star^2 - Uk1(2)^2);

         end

         if Uk1(1) <= 0 && Uk1(2) <=0

              Uk1(2) = -Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1)
                  )^2));
              Uk1(1) = -sqrt(Ulim_star^2 - Uk1(2)^2);
         end

         if Uk1(1) >= 0 && Uk1(2) <=0

            Uk1(2) = Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1))
                ^2));
              Uk1(1) = sqrt(Ulim_star^2 - Uk1(2)^2);

         end



else
    if sqrt(Uk1(1)^2 + Uk1(2)^2)>= Ulim_star && abs(Uk1(1)) < 0.00001
          if Uk1(2) > 0
              Uk1(2) = Ulim_star;
          else
              Uk1(2) = -Ulim_star;
          end
    end
end
```

```
Uk = Uk1;
U_old = [Uk;Uk;Uk];

  %Current Prediction

i_pred = A*xk + B*Uk + h;

id_p = i_pred(1);
iq_p = i_pred(2);

 % Duty cycle for the inverter
co = cos(theta_me);
sen = sin(theta_me);
T = [co -sen; sen co];
u_alfabeta=T*Uk;

u_alfa = u_alfabeta(1);
u_beta = u_alfabeta(2);

duty_a = 0.5 + u_alfa/Ulim_star;
duty_b = 0.5 + (-0.5*u_alfa + sqrt(3)/2*u_beta)/Ulim_star;
duty_c = 0.5 + (-0.5*u_alfa - sqrt(3)/2*u_beta)/Ulim_star;
```

Appendix B

I-MPC Matlab code

The Matlab code of the I-MPC with prediction horizon of N = 3 is reported below. The computation of the inverse matrix $H^{-}1$ is here indicated with the command $inv(H)$. However, the implemented code include the parametric inversion of the matrix, in order to avoid high computational cost.

```
function [Uk,duty_a,duty_b,duty_c,id_p,iq_p]  = IMPC(w_me,theta_me, id
    ,iq,id_rif,iq_rif, U_DC,Ld, Lq, R,q1,q2,r1,r2,s1,s2)

% I-MPC, equality constrain,  prediction horizon N =3
 %current control of a SRM
 %parameters give as input (possible online change)

Ts = 100e-6;   %sample time

% model state equation (discrete-time)
%x(k+1) = A*x(k) + B*u(k)

A = [ (1 - Ts*R/Ld) w_me*Ts*Lq/Ld; -w_me*Ld/Lq*Ts (1- Ts*R/Lq)];
B = [ Ts/Ld 0; 0 Ts/Lq];
C = [1 0; 0 1];

persistent U_old;
% set original U_old to zero the
% first time this function is invoked
if isempty(U_old)
  U_old=[0;0];
end

%deltax(k+1) = A1*deltax(k) + B1*DeltaU

%deltax(k) = x(k) - x(k-1)
A1 = [ A; A^2; A^3];
B1 = [B zeros(2,4); A*B  B zeros(2,2); A^2*B  A*B  B];
C1 = [1 0 0 0 0 0;
      0 1 0 0 0 0;
      1 0 1 0 0 0;
      0 1 0 1 0 0;
      1 0 1 0 1 0;
      0 1 0 1 0 1];

%current acquisitions
```

```
xk = [ id; iq];

persistent x_old;
% set original U_old to zero the
% first time this function is invoked
if isempty(x_old)
  x_old=[0;0];
end

Xk = [id;iq;id;iq;id;iq];

rif = [ id_rif; iq_rif;id_rif; iq_rif;id_rif; iq_rif];

Y = rif - Xk - C1*A1*(xk - x_old) ;

% Weigth matrices

Q1 = [q1 0 0  0  0 0;
      0 q2 0  0  0 0;
      0 0  q1 0  0 0;
      0 0  0  q2 0 0;
      0 0  0  0  0 0;
      0 0  0  0  0 0];

S1 = [0 0 0 0 0  0;
      0 0 0 0 0  0;
      0 0 0 0 0  0;
      0 0 0 0 0  0;
      0 0 0 0 s1 0;
      0 0 0 0 0  s2];

R1 = [r1 0  0  0  0  0;
      0  r2 0  0  0  0;
      0  0  r1 0  0  0;
      0  0  0  r2 0  0;
      0  0  0  0  r1 0;
      0  0  0  0  0 r2];

  % Hessian matrix

H = 2*(B1'*C1'*Q1*C1*B1 + R1 +B1'*C1'*S1*C1*B1);

 d = 2*((Y'*Q1*C1*B1) + (Y'*S1*C1*B1))';

 % Optimal solution

  deltaU = H\d;

  %Receding Horizon

 deltaUk = deltaU(1:2);

 % Check voltage limits

Uk1 = U_old + deltaUk;
```

```
Ulim_star = U_DC/sqrt(3);


if  sqrt(Uk1(1)^2 + Uk1(2)^2)>= Ulim_star && abs(Uk1(1)) >= 0.00001

      if      Uk1(1) >= 0 && Uk1(2) >=0 % I dial

              Uk1(2) = Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1))
                   ^2));
              Uk1(1) = sqrt(Ulim_star^2 - Uk1(2)^2);
      end

         if Uk1(1) <= 0 && Uk1(2) >=0 % II dial

              Uk1(2) = -Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1)
                   )^2));
              Uk1(1) = -sqrt(Ulim_star^2 - Uk1(2)^2);

         end

         if Uk1(1) <= 0 && Uk1(2) <=0 % III dial

              Uk1(2) = -Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1)
                   )^2));
              Uk1(1) = -sqrt(Ulim_star^2 - Uk1(2)^2);
         end

         if Uk1(1) >= 0 && Uk1(2) <=0 %IV dial

            Uk1(2) = Uk1(2)/Uk1(1)*Ulim_star*sqrt(1/(1+(Uk1(2)/Uk1(1))
                 ^2));
            Uk1(1) = sqrt(Ulim_star^2 - Uk1(2)^2);

         end


else
     if  sqrt(Uk1(1)^2 + Uk1(2)^2)>= Ulim_star && abs(Uk1(1)) < 0.00001
           if Uk1(2) > 0
               Uk1(2) = Ulim_star;
           else
               Uk1(2) = -Ulim_star;
           end
     end
end


% Voltage vector Uk = [Ud; Uq]

Uk = Uk1;

U_old = Uk;
x_old = xk;

%compute the current prediction
```

```
    i_pred = A*xk + B*Uk;

  id_p = i_pred(1);
  iq_p = i_pred(2);

% duty cycle for the inverter

co = cos(theta_me);
sen = sin(theta_me);
T = [co -sen; sen co];
u_alfabeta=T*Uk;

u_alfa = u_alfabeta(1);
u_beta = u_alfabeta(2);

duty_a = 0.5 + u_alfa/Ulim_star;
duty_b = 0.5 + (-0.5*u_alfa + sqrt(3)/2*u_beta)/Ulim_star;
duty_c = 0.5 + (-0.5*u_alfa - sqrt(3)/2*u_beta)/Ulim_star;
```