FAKULTÄT FÜR
Elektrotechnik und Informationstechnik
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Lehrstuhls für Elektrische Antriebssysteme und Leistungselektronik

Quasi-Time-Optimal Controllers for Electrical Drives

Esteban José Fuentes Henríquez

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs
genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Hans-Georg Herzog
Prüfer der Dissertation:
1. Univ.-Prof. Dr.-Ing. Ralph Kennel
2. Univ.-Prof. Dr.-Ing. José Rodríguez
(Technische Universität Federico Santa María, Valparaíso, Chile)

Die Dissertation wurde am 28.01.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 01.04.2015 angenommen.

Ich versichere, dass ich diese Doktorarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den Mai 21, 2015 Esteban José Fuentes Henríquez

Acknowledgments

For their role in my formation, the development of this work and ultimately my sanity, I must thank my family: Rafael, Anibal, Fanny and Nestor, my scientific partners: Dante and Hector, the best mitbewohner: Martin, my neighbours: Felix and Mauricio, my German family: Carlitos, Alejandra, Jethro and Daniela and particularly Signe.

For the opportunity to do and enjoy my doctoral studies I must thank Prof. Kennel, Prof. Rodriguez and the whole EAL team.

Finally, I must thank all honest Chilean taxpayers and the BecasChile/CONYCIT program.

Abstract

This work deals with the design and implementation of feasible quasi-time-optimal control algorithms in the context of drives. A consistent approach, based on ad hoc simplifications and adaptations of more general optimal control methods, was used to produce torque, speed and position controllers for conventional drives, and a torsional torque controller for a drive with a flexible shaft: a two-mass-system. These controllers take the form of non-linear state feedback rules, which require no offline computations and, for the most part, no tunning.

Kurzfassung

Diese Arbeit beschäftigt sich mit dem Design und Implementierung von praktikabel quasizeitoptimalen Regelalgorithmen im Rahmen der Antriebssysteme. Ein gleichartig Konzept, basierend auf Ad-hoc-Vereinfachungen und Anpassungen von mehr allgemeineren Methoden optimaler Regelung wurde verwendet, um Drehmoment-, Drehzahl- und Position-Regler für konventionelle Antriebe und ein Torsionsdrehmoment-Regler für eine Antrieb mit einer flexibel Welle (ein Zweimassen-System) zu schaffen. Diese Regler nehmen die Form von nichtlinearen Zustandsrückfuhrungsregeln an, die keine offline Berechnungen und kaum Tuning erfordern.

Contents

Part I.

Introduction and Background Theory

1. Introduction

The work presented in this thesis deals with the design and implementation of approximate time-optimal controllers in the context of electrical drives.

The broad interest on the particular subject of electrical drives is very easily understood, when one considers that the electrical motor has become the corner stone of industrial production, where its uses range from giant grinding mills for metals extraction and refinement, to miniature laser pointing. Their presence is also fundamental in other aspects of modern societies, such as energy generation, transportation, data management and war. Recent developments in 3D printing, for example, rely on drives and sophisticated control systems. In this kind of applications, the trade-off between speed and precision poses great challenges on the control system design. In this direction, the development of advanced, but feasible control systems might be critical for the descentralization and demonopolization of production means.

All applications have in common the unstopping refinement of their requirements, in particular: efficiency and reliability. Two complementary approaches can be distinguished in this situation, on one hand the improvement of motor designs and on the other, the improvement of the control methods. The work presented in this thesis deals with the later and, although the models used for the drive dynamics are very general and thus, it might be possible to apply the developed control methods to a broad family of motors and motion systems, their performance has been validated using standard industrial equipment (rotating motors and voltage source power converters).

Control theory has been developed hand in hand with the aforementioned applications. In 1867, arguably on the aftermath of the industrial revolution, Maxwell published *On Governors* [1], one of the first analysis on feedback and stability, in the context of centrifugal governors for steam engines. But it was not until after the second world war, that the engineering practices in control, mostly based on trial and error up to that point, began to be formaly defined [2]. The most important results in modern and optimal control theory: the linear quadratic regulator [3], the Kalman filter [4], the dynamic programing principle [5] and the Pontryagin Maximum Principle [6], were developed during the cold war. Before the introduction of fast computers allowed it, however, the application of the later remained constrained to a very narrow scope of problems. Particularly problematic, in the context of these theories, are still the imposition of constraints for the actuation and the states and non-linearities in the system dynamics.

Model predictive control [7] was developed during the eighties for the oil industry, as a simplification of the optimal control problem, in particular, the optimization time frame, or prediction horizon, which in the linear quadratic regulator is assumed to be infinite, is truncated. This allows to approach the optimization problem taking non-linearities and constraints into account, using numerical optimization methods. These schemes usually involve off-line calculations and yield to complex feedback rules, which can be approximated with piecewise affine controllers. Despite the advancements in computing power, however, its application remained still constrained to relatively slow systems, such as chemical plants.

In the context of drives, optimal control and particularly time-optimal control, has been developed for positioning in hard drives [8] and mechatronics, in the form of off-line motion planning [9]. These approaches, generally do not integrate the power converter, and assume very fast torque dynamics.

The application of model predictive control has been rather limited in in the context of drives, mostly due to the high computation requirements imposed by these schemes and the fast dynamics involved, which render them unfeasible. As computers become faster and more reliable, these kind of approaches will be more massively adopted. At the current state of real time computers, however, compromises and ad-hoc adaptations must still be made to, for example, avoid large number of spurious iterations.

The particular area of rotating, variable speed drives has been dominated, practically since its conception, by field oriented control (FOC) [10]. The analysis introduced by Blaschke is nowadays fundamental for the understanding of the internal workings of electrical motors. The core idea in FOC is a change of variables (or rotation), which enables to write the dynamic equations of the motor in a rotating frame of reference, where flux and torque production are proportional to the stator currents. From an historical perspective, this allows to control alternating currents motors as if they were direct current motors, this is, classical proportional-integral controllers can be used at arbitrary speeds and zero steady state error will be obtained for step flux and torque references. In principle, any kind of controller can be used instead of the PI controllers and the system will inherit its properties.

A second very popular control scheme for variable-speed drives is direct torque control (DTC) [11, 12]. In this scheme, the torque and flux magnitudes are directly controlled with hysteresis controllers. The output of the later are combined with the position of the flux in a look-up table, which determines the switching state to be applied by the power converter. In this scheme there is no explicit rotation of variables, but its implicit in the look-up table. This scheme achieves great dynamic behaviour, but generates stator currents with a distributed energy spectrum. This is usually not desirable, since spurious dynamics might be excited.

A more recent development, which achieves similar dynamics as DTC, when applied in drives , is finite-set model predictive control (FS-MPC) [13, 14, 15] The basic assumption behind this method is that the actuation belongs to a finite-set. This assumption has, in the context of power converters, a physical meaning, since the power transistors used in power converters are normally used as switches. On the other hand, in the context of time optimal control, this assumption has a theoretical meaning: the *bang-bang* principle, introduced by Pontryagin, states that for a linear system with real eigenvalues, the minimum time controller will be bang-bang with at most $n - 1$ switchings, where n is the dimension of the state space. The FS-MPC scheme makes a practical use of this assumption: the scheme works as a receding horizon controller, where the optimal control problem is solved explicitly: the system state is predicted for all the feasible actuation sequences and the optimal control is the first component from the sequence that generates the lowest cost. Naturally, this approach suffers from the curse of dimensionality: the number of feasible actuation sequences grows exponentially with the prediction horizon.

Latest research around this method has shown that it can be successfully applied in power converters, particularly multilevel converters [16, 17], where many states must be controlled at the same time. FS-MPC is effective in finding the best trade-off between all the control goals. The dynamics involved are generally of first order and thus, a prediction horizon of one sampling period is enough to account for the desired dynamics. In this way the control decision is reduced to a single dimension and the algorithm can be implemented in conventional hardware, to solve the control problem online, reaching high sampling frequencies.

The control for multilevel converters is an example of a multi-variable and multi-objective control system. Trade off between them is achieved aggregating the performance index for each control goal, for example, through a weighted sum of the squared errors, between the predicted states and their desired states. No closed form for the weights exists and they must be adjusted for each application and set of parameters. In this situation, the imposition of particular closed-loop dynamics is very difficult. For this, numerical schemes have been proposed, to search for weighting factors that optimize more general performance indexes, this is, introducing an optimization problem in an additional level of abstraction, This is further complicated, when one considers systems where there is causal relationships of higher order. We illustrate this with a brief example.

Consider the dynamic equations of the mechanical system in Fig. 1.1

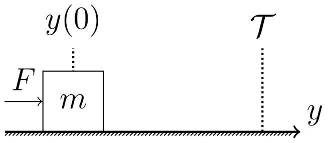

Figure 1.1.: Free body with mass m.

$$\frac{dx}{dt} = f(x,u)\,, \tag{1.1}$$

with

$$x = \begin{bmatrix} v \\ y \end{bmatrix}, \qquad\qquad u = F \tag{1.2}$$

and

$$f(x,u) = \begin{pmatrix} F/m \\ v \end{pmatrix}. \tag{1.3}$$

We can try to apply a receding horizon controller for this system with a prediction horizon N, to steer this system from an arbitrary initial condition towards the origin, using the cost function:

$$J = \sum_{k=1}^{N} \left\{ y[n+k]^2 + \lambda_v v[n+k]^2 + \lambda_u u[n+k]^2 \right\} \tag{1.4}$$

where the λ factors are constant non-zero weights. The state of the system is predicted using a discrete-time model

$$x[n+1] = g(x[n], u[n])\,. \tag{1.5}$$

If we assume that u can only take one of it extreme values or zero

$$u \in \left\{ -\hat{F}, 0, \hat{F} \right\}\,, \tag{1.6}$$

the controller will not react for arbitrarily small initial conditions: the cost of action: increasing u and v, outweighs the cost of steady state error. The effect is aggravated with shorter prediction horizons and more complex dynamics.

This is the starting point of the work developed in this thesis.

Time-optimality is adopted as the control goal and the finite-set algorithm is used to implement quasi-time-optimal controllers for systems with higher order dynamics. The later is understood in this work as the order of the time derivative of the system state being controlled, where the control first appears.

The control systems, or drives considered are:

- torque of the induction motor,
- speed of a synchronous motor,
- speed of the induction motor,
- position of the synchronous motor,
- torsional torque of a two-mass-system.

In each case, the same methodology is applied: a simplified abstract model is introduced to approximate the system transient behaviour, namely, for the first three control systems systems a double integrator and for the later two a triple integrator and a simple oscillator. The time-optimal control problem is solved for these abstract system using numerical methods. The later are then modified (simplified) using the finite-set assumption, to devise feasible control algorithms. One of the driving ideas being the integration of the different abstraction levels involved: stator voltage synthesis, torque production and the higher order dynamics. With this, the proposed control algorithms take the form of state feedback controllers. This approach draws a clear contrast with respect to the traditional approach in drives, where cascaded control structures are used and the inner dynamics are abstracted, imposing bandwidth limits on the outer controllers.

Time-optimal controllers have generally (and particularly, for the aforementioned abstract systems) bang-bang outputs. This generates chattering around the steady state and the switching curve, under real-world conditions (noise, delay, unmodeled dynamics, finite numerical precision). In order to tackle this issue, which is particularly critical for the two last systems, a modification of the finite-set algorithm is introduced, based on space vector modulation, to enable the use of a continuous actuation set.

In each case, the performance of the proposed controllers is validated in an experimental setup and the measured behaviour is checked against the expected behaviour, calculated using general numerical methods to solve the time-optimal control problem for the aforementioned simplified models.

In the reminder of this part, the models of the drives at hand are developed and the relevant control concepts, which make up the context for this work, are introduced. Afterwards, in the second part of this work, the torque controller for the permanent magnet motor is revised and a continuous-set algorithm is introduced.

The third and fourth parts of this work deal with the development of quasi-time optimal controllers for systems with second and third order dynamics, respectively.

1.1. On simulations and experimental results

Throughout this theses, the proposed control methods are validated using simulations and experimental tests.

The simulations were carried out in the Python environment [18], using the Numpy [19] and Scipy [20] packages. The system derivatives and the control algorithms were written in C language, so that the same code could be used in both simulations and experimental tests. For the integration with Python the special purpose library SClib [21] was developed.

All the plots presented in this thesis were produced using the Matplotlib package [22].

The test bench used for the experimental tests consists of a standard 2.2 kW induction motor and a 2 kW PMSM, both driven by a commercial 5 kW two-level voltage source inverters, modified for direct access to the switching state of the power transistors. Currents and dc-link voltage measurements are provided by the inverters. The rotor position is measured using standard incremental encoders with 1024 lines for the induction motor and 5000 lines for the permanent magnet synchronous motor.

The control strategy is run in an industrial PC equipped with a 1.4 GHz Intel processor. The computer runs the Linux operating system, modified with the RTAI package for real-time operation. The measurements and the control signals are managed with special purpose cards connected to the computer through a PC-104 bus.

For studying the two-mass system, the motors are coupled through a specially designed piece, which enables to change the frequency response of the system, by using steel stabs of different diameters as the shaft and by changing the inertias of both, the driving machine and the load, by adding weights at each side.

Technical limitations of the test bench forced the reduction of the voltage range to produce the experimental results presented here: when modulating relatively high voltages, the zero voltage vector is applied for very short time and the noise introduced by the switching elements is very noticeable in the currents measurements, which are synchronized to occur while the zero vector is applied. To tackle this, the value for the maximum actuation was set

$$\hat{u} = 0.9 \cdot \frac{\sqrt{3}}{3} v_{dc} \,. \tag{1.7}$$

The sampling period h used was 46.088 µs for the scheme using space vector modulation and 30.725 µs for the direct methods.

The measurements presented in this work correspond to the measurements available to the controller. Variables not being measured were estimated using a state observer (see Sec. A).

2. System Models

In this chapter we develop models for the systems we wish to control: the permanent magnet synchronous motor drive and the induction motor drive, suitable for control purposes. These models include the actuator: the voltage source inverter, and the relevant dynamics of both motors. The model for the two-mass-system is introduced in chapter 10, since is only relevant for the work developed there.

The models in this chapter are developed in continuous-time, but, since the control methods proposed latter are formulated to be implemented in digital computers, discrete-time models are required. The last section in this chapter presents the method used thought this work to produce them.

From the control perspective, the aim of this chapter is to be able to write an ordinary differential equation of the form:

$$\frac{dx}{dt} = f(x, u, d), \tag{2.1}$$

that effectively describe the relevant dynamics of the systems at hand. In eq. 2.1 x pertains to the system state, u to the actuation and d to a non-measurable disturbance. Notwithstanding this, we would also like to understand the internal workings of the motor, this is why the models are developed starting from the motor's physics; we are interested in the electromagnetic dynamics of the motor: the relationships between voltages, currents and magnetic fields, the mechanical dynamics: the relationship between torque, speed, position of the rotor and in the particular case of the two-mass-system, torsional torque and speed and position of the load, and finally, the energy conversion aspects of the motor, i.e., the mechanism that turns electrical energy into mechanical energy. In this sense, no specific details regarding the construction of the motor are presented, only those necessary to account for these effects.

The models introduced in this chapter are simple representations of complex physical phenomena. Consequently, the parameters introduced to develop this models, which are assumed to be constant, seem to vary, depending on the operation point, for example, and even its past values. The effect produced by the dead-time used to safely switch the transistors in the power converter is an example of the first case. When synthesizing relatively small voltages using pulse-width modulation schemes, the power converter generates short-duration voltage pulses, comparable to the duration of the dead-time. In this scenario, the stator windings seem to have higher resistance. An example of the later case, where past values of the system variables seem to change the parameters of the system, is magnetic hysteresis, which characterizes the iron used to build the motors. In this case, the inductance of the stator windings seem to change.

These and other effects are disregarded in this work, in order to keep the proposed control strategies as simple as possible. In this sense, the control strategies developed in this work show the extent to which this simple models are useful. Nevertheless, special care has to be taken when fitting the parameters of the models, so that their dynamics match the real dynamics, under the conditions imposed by the controller. This is addressed at the end of each section in this chapter.

2.1. Voltage Source Inverter

In its lowest abstraction level, the task of the control algorithm of a drive is to synthesize the voltages to be applied to the motor. From the control perspective this is usually not an issue and the power converter is regarded as a gain or a simple delay. In this work, however, one of the driving ideas is the integration of all the abstraction levels up to the extent where they become relevant to the control goals. In this sense, the actuator: the voltage source inverter, determines the actuation set $\mathscr{U}$.

The control algorithms presented in this work were developed assuming that the motors are driven by a two-level voltage source inverter (VSI), generalization for using other converters should be, however, straight forward. A simplified scheme of this power converter is presented in Fig. 2.1. It consists of a power source and three legs connected to the three terminals of the motor. Different switching configurations produce different voltages in the motor terminals. In each leg only one switch should be connected at a time: the commutation process includes a relatively short dead-time, during which, both switches of the corresponding leg are turned off, this to avoid short circuiting the power source. The switches are implemented using different transistors, depending on the power rate and requirements of the application, however, Insulated-gate bipolar transistors (IGBT) are the most popular Further details regarding the operation of the VSI such as control of the dc-link (v_{dc}), are disregarded in this work. Note, however, that the value of v_{dc} might change drastically, for example, when breaking the motor. In this case energy is taken out of the rotor and transfered to the dc-link. This has a critical effect on the rate of change of the stator currents. To account for this in the experimental tests, v_{dc} is continually monitored and its value is feed to the control algorithm.

The VSI depicted in fig. 2.1, and all the drives we wish to model are triphasic systems, nevertheless, the information required to describe the energy transformation process in the motors can be synthesized in two components. For this, we introduce the Clarke transformation (also known as $\alpha\beta$ transform), which gives the projection of three phase quantities in two orthogonal components $\alpha\beta$ [23]:

$$T_{\alpha\beta} = \frac{2}{3}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}. \tag{2.2}$$

The more complete $\alpha\beta\gamma$ transform includes a third component, which would account for the common mode. We disregard it, since it plays no role in our analysis.

A pseudo-inverse for $T_{\alpha\beta}$ (which will be used later), is given by:

$$T_{\alpha\beta}^{-1} = \begin{bmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix}. \tag{2.3}$$

With this, we can write the voltage applied to the motor in $\alpha\beta$ coordinates, in terms of

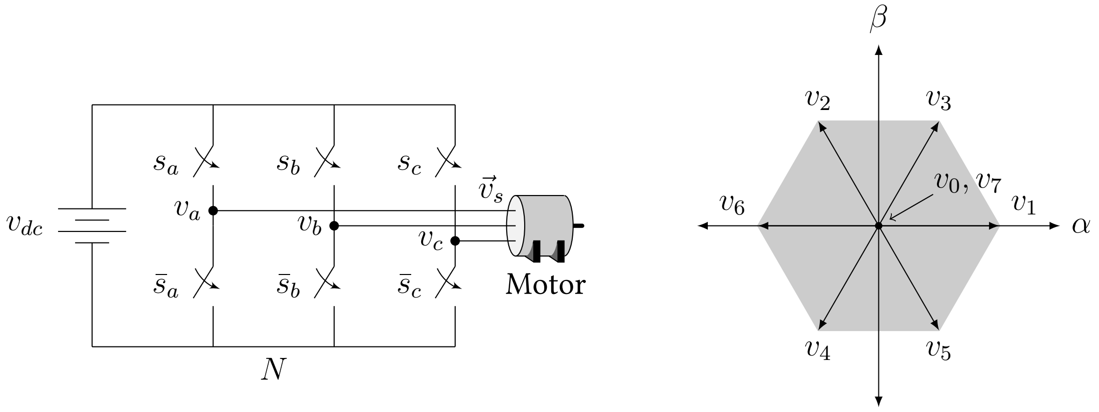

Figure 2.1.: Voltage source inverter (left) and synthesized voltages in the $\alpha\beta$ plane (right). v_a, v_b and v_c are defined with respect to node N. The binary representation of the subscripts in $v_0, \ldots, v_7$ encode the switching state that generates them in the order $s_c s_b s_a$. These voltage vectors constitute the actuation set $\mathscr{U}$ with the definition in eq. (2.4). The gray area constitutes the actuation set with the definition in eq. (2.7).

the state of the switches and the magnitude of the dc-link:

$$\vec{v}_s = v_{dc} T_{\alpha\beta} \begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix}, \qquad \text{with} \qquad s_a,\, s_b,\, s_c \in \{0, 1\} \;. \tag{2.4}$$

The different combinations of switching states generate six active and two zero *voltage-vectors* (see fig. 2.1 for a representation in the $\alpha\beta$ plane). Some of the schemes presented in this work take these voltage vectors as the actuation set:

$$u \in \mathscr{U} = \{v_0, v_1, \ldots, v_7\}\,. \tag{2.5}$$

These are called *direct* or *finite-set* schemes and the underlaying assumption, the *finite-set assumption.*

Another way to define $\vec{v}_s$ is in terms of duty cycles: given the sampling period h, the duty cycle defines the portion of h in which the corresponding switch remains on:

$$s_x(t) = \begin{cases} 1 & \text{if} \quad ({}^{t}\!/\!{}_{h} \bmod 1) \leq d_x \\ 0 & \text{if} \quad ({}^{t}\!/\!{}_{h} \bmod 1) > d_x \end{cases} \;. \tag{2.6}$$

In this way the mean value of the corresponding voltage v_i in one sampling period h can take any value in $[0, v_{dc}]$.

If h is small compared to the system dynamics, we can assume that the voltage feed to the motor is equal to the mean output voltage of the VSI in one sampling period:

$$\vec{v}_s = v_{dc} T_{\alpha\beta} \begin{bmatrix} d_a \\ d_b \\ d_c \end{bmatrix}, \qquad \text{with} \qquad d_a,\, d_b,\, d_c \in [0,1]\,. \tag{2.7}$$

With this, the actuation set is now given by the gray area represented in the $\alpha\beta$ plane in fig. 2.1.

The next section presents a method to obtain the duty cycles in eq. (2.7).

2.1.1. Space vector modulation

Space vector modulation (SVM) [24] is a technique to calculate the duty cycles in eq. (2.7), to approximate a given voltage vector (the reference voltage vector $\vec{v}^*$) inside the gray area in Fig. 2.1.

The basic idea is to regard the problem as a linear algebra problem. The hexagon in figure 2.1 is divided in six triangles or sectors, the first step to calculate the duty cycles is to find in which of these sectors $\vec{v}^*$ lies, this can be done calculating its angle with respect to to the α axis, for example. Then, the following linear problem needs to be solved for d_a and d_b:

$$\begin{bmatrix} v_a & v_b \end{bmatrix} \begin{bmatrix} d_a \\ d_b \end{bmatrix} = \vec{v}^*\,, \tag{2.8}$$

where v_a and v_b represent the two voltage vectors at the sides of the sector. If $\vec{v}^*$ lies inside the hexagon then $d_a + d_b \leq 1$.

At the sides of every sector there is one vector which is is generated by turning one switch on (v_1, v_2 or v_4) and another, which is generated by turning two switches on (v_3, v_5 or v_6), we call them single and combined vectors. At a given sector, the one delimited by v_2 and v_6, for example, we call the single vector at its side v_x (v_2 in the example) and the vector which summed with v_x generates the combined vector v_y (v_4 in the example).

With this, the duty cycle for the single vector is given by

$$d_x = d_1 + d_2 \tag{2.9}$$

and for v_y

$$d_y = \begin{cases} d_1 & \text{if } v_x = v_2 \\ d_2 & \text{if } v_x = v_1 \end{cases}\,, \tag{2.10}$$

During the remaining time

$$d_0 = 1.0 - dx - dy\,, \tag{2.11}$$

a zero vector is applied.

The SVM technique comprehends further details, such as the order for turning the switches on and off, which has a consequence in the harmonic content of the synthesized voltage, and implementations for more complex power converters. [25].

In the work developed later in this thesis, the ideas behind space vector modulation are integrated within the controllers to approximate vectors, not on a voltage plane anymore, but on a plane representing the error in the electrical torque and the satisfaction of the necessary conditions for maximum torque per ampere operation, in the case of the PMSM, and the error in the current necessary to sustain the rotor flux, in the case of of the IM.

2.2. Permanent Magnet Synchronous Motor

The permanent magnet synchronous motor is generally characterized by a high power density and dynamics and is usually employed in servo applications.

Fig. 2.2 presents a simplified scheme of the motor with the relevant variables. Basically, the motor consists of three static windings (the stator) and a rotating magnet (the rotor). A common characteristic, and usually undesired effect, in this kind of motors is that the magnet's field saturates the iron in the stator in the direction where the magnet's poles point [26], this is, in those directions, it lowers the permeability of the iron μ: the ability of the iron to support the formation of magnetic flux density B out of the magnetic field H ($B = \mu H$). As a consequence the inductance of the stator windings, which is proportional to μ diminishes in those directions also. This is relevant to operate the motor efficiently: since in determines the conditions where maximum torque with minimum losses is produced.

The basic workings of the motor can be phrased in this way: by manipulating the input voltages v_a, v_b and v_c, the stator currents i_a, i_b and i_c can be controlled to produce a net magnetic field pointing at an arbitrary direction (orthogonal to the rotation axis). In this way, torque is applied to the rotor, since its own magnetic field tends to align with the magnetic field generated by the stator currents.

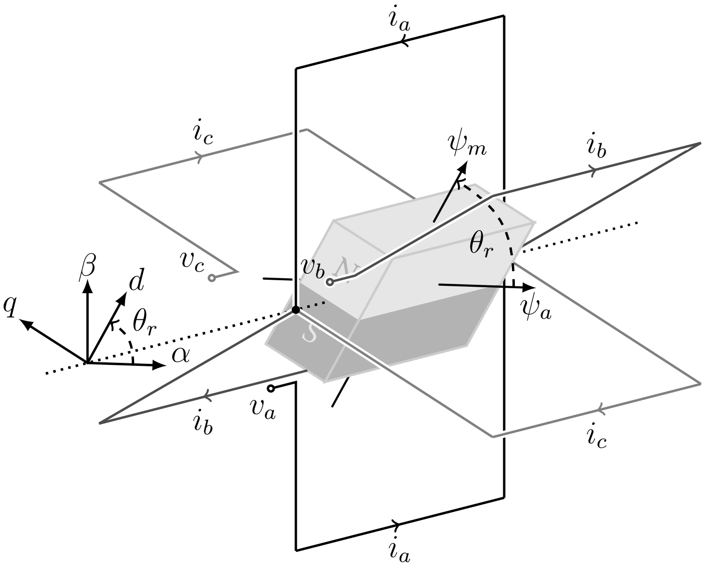

Figure 2.2.: Diagram of the permanent magnet synchronous motor. The rotation axis is depicted with a dotted line. ψ_a represents the magnetic flux passing through winding a and ψ_m the magnetic flux produced by the rotor magnet, θ_r is the angle between them. $\alpha\beta$ and dq represent a stationary and a rotatory frame of reference. The later is aligned with the rotor magnet.

In the following sections the model of the PMSM is developed in two distinct frames of reference or coordinate systems: $\alpha\beta$ and dq. The first one is fixed with respect to the positions of the stator windings and is the most natural to develop. The second frame of reference is a rotatory one and is aligned with the direction of the magnet. In this frame of reference, the currents that generate constant torque are also constant. This makes it specially suitable for control purposes and it is the basis for field oriented control (see Sec. 3.1).

2.2.1. Model of the permanent magnet synchronous motor in a stationary frame

Applying the Kirchhoff's and Faraday's laws to each stator winding, their electrical dynamics can be written in the following form:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = r_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix}. \tag{2.12}$$

ψ represents the flux passing through the respective winding, this is, the surface integral of the normal component of magnetic flux density passing through the winding:

$$\psi = \iint_S BdS \tag{2.13}$$

r_s represents the stator windings resistance and accounts for the Joule losses in the windings.

Assuming that the magnetic field produced by the rotor magnet is distributed sinusoidally, the flux can be expressed as:

$$\begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} = L_s^{\{abc\}} i_s + \psi_m \begin{bmatrix} \cos(\theta_r) \\ \cos(\theta_r - 2\pi/3) \\ \cos(\theta_r - 4\pi/3) \end{bmatrix}, \tag{2.14}$$

ψ_m is the magnitude of the rotor magnet flux, the vector multiplying it gives its projection on the directions normal to the plane of each winding. L_s^{abc} is the inductance matrix of the stator (the superscript denotes that the stator inductance L_s is defined in abc variables) and is given by:

$$\begin{aligned} L_s^{\{abc\}} = & \begin{bmatrix} L_a & M_{ab} & M_{ac} \\ M_{ab} & L_b & M_{bc} \\ M_{ac} & M_{bc} & L_c \end{bmatrix} \\ & + L_\Delta \begin{bmatrix} \cos(2\theta_r) & \cos(2\theta_r + 2\pi/3) & \cos(2\theta_r + 4\pi/3) \\ \cos(2\theta_r + 4\pi/3) & \cos(2\theta_r) & \cos(2\theta_r + 2\pi/3) \\ \cos(2\theta_r + 2\pi/3) & \cos(2\theta_r + 4\pi/3) & \cos(2\theta_r) \end{bmatrix}. \end{aligned} \tag{2.15}$$

In this equation, the first term accounts for the geometry of the windings and the materials used to construct motor. In this term L_i and M_{ij} with $i, j \in \{a, b, c\}$ are called the self and mutual inductances of each winding and between two windings. The second term accounts for the effect produced by the magnetic field of the rotor on the iron in the stator, which is to lower its permeability μ, thus lowering the net inductance of the windings.

Most electrical motors designed to generate constant torque with sinusoidal currents and voltages. One design consideration to achieve this is to distribute the stator windings sinusoidally and to construct them symmetrically. This allows to assume [26]:

$$L_a = L_b = L_c = L_l + L_m \tag{2.16}$$

and

$$M_{ab} = M_{ac} = M_{bc} = -\frac{L_m}{2}. \tag{2.17}$$

L_l and L_m are called leakage and magnetizing inductances. Along this assumptions and the definition:

$$L_\Sigma = L_l + \frac{3}{2} L_m, \tag{2.18}$$

the stator inductance, defined now in the $\alpha\beta$ frame, can be written in the more compact form (for the definition of $T_{\alpha\beta}$ see Sec. 2.1):

$$\begin{aligned} L_s &= T_{\alpha\beta} L_s^{\{abc\}} T_{\alpha\beta}^{-1} && (2.19) \\ &= L_\Sigma \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + L_\Delta \begin{bmatrix} \cos(2\theta_r) & \sin(2\theta_r) \\ -\sin(2\theta_r) & \cos(2\theta_r) \end{bmatrix} && (2.20) \end{aligned}$$

here the effect of the magnet is more clearly understood: the variation in the total inductance peaks at the directions pointed by the north and the sound poles of the magnet.

Furthermore, with the following definitions:

$$\begin{aligned} L_\Sigma &= \frac{L_d + L_q}{2} && (2.21) \\ L_\Delta &= \frac{L_d - L_q}{2} && (2.22) \\ T &= \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{bmatrix} && (2.23) \end{aligned}$$

L_s can be rewritten:

$$L_s = T \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} T^{-1}. \tag{2.24}$$

L_d and L_q represent the equivalent stator inductances in the directions of the north pole of the rotor magnet (the d axis) and in a direction orthogonal to it (the q axis).

Matrix T, known as the inverse Park transform [27], gives the projection of a magnitude defined in the dq axes onto the $\alpha\beta$ axes: the matrix $\left[\begin{smallmatrix} L_d & 0 \\ 0 & L_q \end{smallmatrix}\right]$ corresponds to the stator inductance defined in the dq axes. In the next section the Park transform T^{-1} is used to derive the model of the PMSM in the rotatory frame dq.

With all this we are in place to write the model of the electrical dynamics of the PMSM in the stationary frame $\alpha\beta$ in terms of the relevant variables, namely: the input voltages $\vec{v_s} = [v_\alpha \; v_\beta]^T$, the stator currents $\vec{i_s} = [i_\alpha \; i_\beta]^T$ ($\vec{}$ denotes a vector in the $\alpha\beta$ frame), the rotor position θ_r and rotor speed $\omega_r = \frac{d\theta_r}{dt}$:

$$L_s \frac{d\vec{i_s}}{dt} + r_s \vec{i_s} = \vec{v_s} - \omega_r \left(\frac{\partial L_s}{\partial \theta_r} \vec{i_s} + J \vec{\psi_m} \right). \tag{2.25}$$

In this equation

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad \text{and} \qquad \vec{\psi_m} = T_{\alpha\beta} \psi_m \begin{bmatrix} \cos(\theta_r) \\ \cos(\theta_r - {}^{2\pi}\!/_3) \\ \cos(\theta_r - {}^{4\pi}\!/_3) \end{bmatrix} = T \begin{bmatrix} \psi_m \\ 0 \end{bmatrix}.$$

In eq. (2.25) the first order dynamics between the stator voltages and currents become evident. The second term on the right side of eq. (2.25) account for the interactions with the rotor, this is, the development of torque. An expression for the latter is the only missing part to link the electrical with the mechanic dynamics in the motor. This is further developed in section 2.2.3, after introducing the rotatory frame of reference dq, which makes the necessary manipulations easier, but the form of this term already gives us hints about the workings of the energy conversion process. This term has voltage units and the dot product with the stator currents has power units. On the other hand, the kinetic power transferred to the rotor is proportional to ω_r and the electrical torque T_e, so the latter must be proportional to a product of the stator currents and the terms between the parenthesis. The latter has two components, the second one accounts for the interactions between the stator currents and the rotor magnet, whereas the second is only different from zero if $L_d \neq L_q$ (see eq (2.20)), the torque generated through this process is called reluctance torque. Motors designs based only on the latter process are called reluctance motors. In these motors, the difference between L_d and L_q is not produced by saturation of the iron, but by anisotropies introduced in the design of the rotor [28].

2.2.2. Model of the permanent magnet synchronous motor in a rotatory frame

The Park transform:

$$T^{-1} = \begin{bmatrix} \cos(\theta_r) & \sin(\theta_r) \\ -\sin(\theta_r) & \cos(\theta_r) \end{bmatrix}, \tag{2.26}$$

gives the projection of magnitudes defined in a stationary frame onto the axes of a frame rotated by θ_r. This is the core idea behind field oriented control: the sinusoidal currents that make the motor produce constant torque look constant, when projected on a frame of reference aligned with the magnetic field of the rotor. These *rotated currents* can be then controlled with normal proportional-integral controllers at arbitrary rotor speeds.

To completely understand this we first develop the model of the PMSM in the rotatory dq frame, which is aligned with the rotor magnet (see fig. 2.2). This model can be obtained by first applying the Clarke transform to eq. (2.12) by right multiplying each term with $T_{\alpha\beta}$:

$$\vec{v}_s = r_s \vec{\imath}_s + \frac{d\vec{\psi}}{dt}, \tag{2.27}$$

then, by applying the park transform we obtain ($\circlearrowleft$ and ${}^{\{dq\}}$ denote vectors and matrices defined in the dq frame):

$$L_s^{\{dq\}} \frac{d\overset{\circlearrowleft}{\imath}_s}{dt} + r_s \overset{\circlearrowleft}{\imath}_s = \overset{\circlearrowleft}{v}_s + J\omega_r \left(L_s^{\{dq\}} \overset{\circlearrowleft}{\imath} + \overset{\circlearrowleft}{\psi}_m \right), \tag{2.28}$$

with

$$\overset{\circlearrowleft}{v}_s = \begin{bmatrix} v_d \\ v_q \end{bmatrix} = T^{-1}\vec{v}_s, \quad \overset{\circlearrowleft}{\imath}_s = \begin{bmatrix} i_d \\ i_q \end{bmatrix} = T^{-1}\vec{\imath}_s, \quad L_s^{\{dq\}} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \quad \text{and} \quad \overset{\circlearrowleft}{\psi}_m = \begin{bmatrix} \psi_m \\ 0 \end{bmatrix}.$$

Note that ω_k is the frequency of the stator currents and voltages and is also proportional to the mechanical rotor speed. Hence the *synchronous* in the name of the motor.

2.2.3. Electromechanical conversion

An expression for the torque developed by the PMSM can be derived by a power balance: the last term in eq. (2.28) is known as the *back electro motive force* (back-emf $\mathscr{E}$) and can be understood as a voltage source, if eq. (2.28) is understood as describing the dynamics of an inductive-resistive circuit. The active power transfered to it is transformed to mechanical energy. If P_k is the kinetic energy of the rotor:

$$P_k = T_e \frac{\omega_r}{p} = \frac{3}{2} \mathscr{E} \cdot \overset{\circlearrowleft}{\imath}, \tag{2.29}$$

where p is the magnetic pole pair number, which works as a gear box between the *mechanical* and *electrical* variables, this means $\omega_r^{\{m\}} = \frac{\omega_r}{p}$ and $\theta_r^{\{m\}} = \frac{\theta_r}{p}$ (${}^{\{m\}}$ stands for mechanical), if $\theta_r^{\{m\}}$ and θ_r are aligned at $\theta_r = 0$. The factor $\frac{3}{2}$ on the right side of (2.29) accounts for the scaling introduced by the Clarke transform.

From eq. (2.29) the following expression for the torque developed by the PMSM can be derived:

$$T_e = \frac{3}{2} p (\psi_m i_q + (L_d - L_q) i_d i_q). \tag{2.30}$$

As stated before, the electrical torque has two components: one is due to the interactions between the stator current and the rotor magnet, the second one is produced by the difference between L_d and L_q.

With this, the model for the drive can be completed with the equation for the rotor speed derivative:

$$\frac{d\omega_r}{dt} = \frac{p}{J_m}\left(T_e - T_l\right). \tag{2.31}$$

As stated before, T_l accounts for the torque developed by the load. J_m is the total moment of inertia of the drive.

Maximum torque per ampere

The PMSM is operated with optimal efficiency when for stator currents of a given magnitude, the developed torque is maximum. This is referred to as maximum torque per ampere operation (MTPA).

The conditions required to operate at MTPA can be derived by solving the following problem:

$$\min_{i_d, i_q} ||\overset{\circlearrowleft}{i}|| \tag{2.32}$$

subject to:

$$\begin{aligned} T_e &= \frac{3}{2}p(\psi_m i_q + (L_d - L_q)i_d i_q) && (2.33)\\ &= T_e^*, && (2.34)\end{aligned}$$

where T_e^* is an arbitrary value: the torque reference. The solution of this problem (which can be easily achieved using Lagrange multipliers), leads to the following condition for i_d and i_q:

$$i_d + \frac{L_d - L_q}{\psi_m}\left(i_d^2 - i_q^2\right) = 0. \tag{2.35}$$

With this, for a given reference for T_e, the optimal values for i_d and i_q can be solved from eqs. (2.30) and (2.35).

2.2.4. State space representation

The model developed in the last sections can be synthesized in the following set of ordinary differential equations (ODEs) [29]:

$$\frac{d}{dt}\begin{bmatrix} \vec{i}_s \\ \omega_r \\ \theta_r \end{bmatrix} = \begin{pmatrix} L_s^{-1}\left(\vec{v}_s - r_s\vec{i}_s - \frac{\partial L_s}{\partial \theta}\omega_r\vec{i}_s - J\omega_r\vec{\psi}\right) \\ \frac{p}{J_m}\left(T_e - T_l\right) \\ \omega_r \end{pmatrix}. \tag{2.36}$$

Eq. 2.36 is to be regarded as the state space representation of the PMSM, with its state being $x = [i_\alpha \; i_\beta \; \omega_r \; \theta_r]^T$, the input $u = \vec{v}_s$ and the disturbance $d = T_l$.

2.2.5. Model parameters

The model in eq (2.36) is characterized by a set of parameters. With very few exceptions [30], control systems require certain degree of knowledge on their values controllers. This is true for the controllers developed in this work, since they are all based on system models.

The following are the procedures used in this work to fit the model parameters.

- Stator winding resistance r_s: measured directly using an Ohmmeter on the terminals of the motor.

- Stator windings inductances L_d and L_q: with the rotor braked sinusoidal and balanced voltages with phases shifts 0, ${}^{2\pi}/_{3}$ and ${}^{4\pi}/_{3}$ rad of amplitude vV and frequency $\omega_r$rad s⁻¹ are applied to the motor terminals. This is equivalent to applying ${}^{\pi}/_{2}$rad phase shifted voltages in v_α and v_β and forcing $\omega_r = 0$ in (2.25). The resulting currents should be unbalanced and sinusoidal and, when plotted in the $\alpha\beta$ plane, should generate an ellipse. The major and minor radii of the ellipse correspond to the amplitudes of i_d and i_q respectively. L_d and L_q can be then solved from:

$$\sqrt{(\omega_r L_d)^2 + r_s^2} = \frac{||i_d||}{v} \qquad \text{and} \qquad \sqrt{(\omega_r L_q)^2 + r_s^2} = \frac{||i_q||}{v}$$

  Note that, due to magnetic hysteresis and Eddie currents, the value of the inductances depends strongly on the frequency of the input voltages [31]. The control methods developed in this work are based on predictions of the stator currents for the time-frame of one sampling period, therefore, the value of the stator winding inductance is estimated applying voltages of the highest frequency the control system can produce, this is, half of the sampling frequency.

- Rotor magnet flux ψ_m: with the terminals of the PMSM open, the rotor is forced to rotate at a fix angular frequency $\omega_r$rad^e s⁻¹ (electrical), the voltage measured in the motor terminals is given by eq. (2.25) with $\vec{i}_s = 0$:

$$\vec{v}_s = J\omega_r T \begin{bmatrix} \psi_m \\ 0 \end{bmatrix}$$

  ψ_m can be then solved from this equation.

- Rotor moment of inertia J_m: using a simple torque controller a torque step is applied, the rate of change in the rotor speed is inversely proportional to J_m, as established by (2.31).

The parameters of the motor used for experimental tests are given in table 2.1.

Table 2.1.: Parameters of the Permanent Magnet Synchronous Motor

$p = 3$
$r_s = 2.2\,\Omega$
$\psi_m = 0.226\,\mathrm{Wb}$
$L_d = 8.4\,\mathrm{mH}$
$L_q = 11.1\,\mathrm{mH}$
$J_m = 8.56 \times 10^{-3}\,\mathrm{kg\,m^2}$

2.3. Induction Motor

The induction motor is the by far the most popular electrical motor. This is due to its simplicity of operation: it can be directly connected to the mains, and also of construction, which makes it extremely robust.

Fig. 2.3 presents a simplified diagram of the motor and the relevant variables. The motor consists of three static windings a, b and c, which constitute the stator, and three windings A, B and C attached to the rotor. In practice the rotor windings are constructed as shown in the picture: with conducting bars parallel to the rotation axis connected at their ends with a conducting ring. This kind of rotor is called *squirrel cage*. Despite the simplicity of its contraction, the workings of the induction motor are a little bit more complex than those of the PMSM: by manipulating the stator voltages v_a, v_b and v_c the stator currents i_a, i_b and i_c can be controlled to produce a magnetic field. The currents can then be manipulated to make the magnetic field rotate, as this happens the rotor windings observe a net change in the flux passing through them, inducing currents in the rotor windings, which oppose to the variation of flux (Lenz law). The interaction between the induced rotor currents and the field generated by the stator currents generate torque on the rotor.

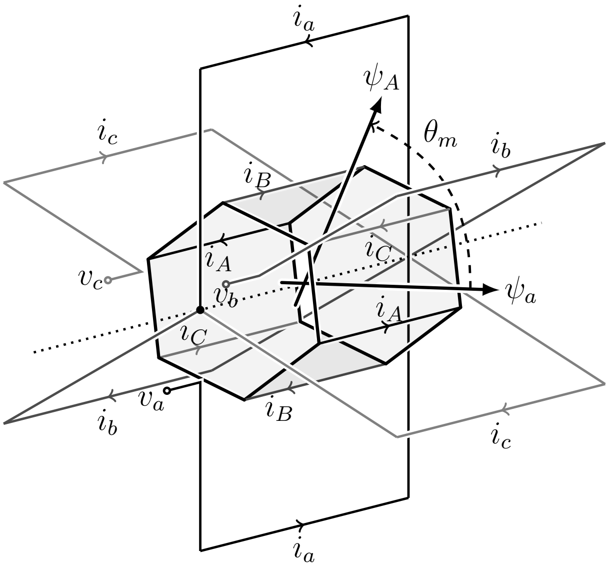

Figure 2.3.: Diagram of the induction motor. The rotation axis is depicted with a dotted line. ψ_a represents the magnetic flux passing through the stator winding a and ψ_A the magnetic flux passing through the rotor winding A.

Just as in the previous section, the model of the induction motor is developed in two distinct frames of reference, $\alpha\beta$ and dq. The first is fixed to the stator windings and the second is aligned with the rotor flux. Note that in this sense the induction and the synchronous motors are different, since the rotor flux is not fixed with to the rotor position: in the induction motor the total rotor flux also rotates with respect to the rotor windings.

The model developed in this section assumes that no saturation of the iron occurs, and that all the rotor quantities are referred to the stator side, through the rotor to stator turns ratio.

2.3.1. Model of the induction motor in a stationary frame

Applying Kirchhoff's and Faraday's laws to each winding in the stator we can write [32]:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = r_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} \tag{2.37}$$

Assuming that the construction of the windings is symmetrical and that they are sinusoidally distributed, the flux passing through them is given by:

$$\begin{aligned} \begin{bmatrix} \psi_a \\ \psi_b \\ \psi_c \end{bmatrix} &= \begin{bmatrix} L_{ls} + L_m & -\frac{L_m}{2} & -\frac{L_m}{2} \\ -\frac{L_m}{2} & L_{ls} + L_m & -\frac{L_m}{2} \\ -\frac{L_m}{2} & -\frac{L_m}{2} & L_{ls} + L_m \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \\ &+ L_m \begin{bmatrix} \cos(\theta_r) & \cos(\theta_r + {}^{2\pi}/_{3}) & \cos(\theta_r + {}^{4\pi}/_{3}) \\ \cos(\theta_r + {}^{4\pi}/_{3}) & \cos(\theta_r) & \cos(\theta_r + {}^{2\pi}/_{3}) \\ \cos(\theta_r + {}^{2\pi}/_{3}) & \cos(\theta_r + {}^{4\pi}/_{3}) & \cos(\theta_r) \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix}, \end{aligned} \tag{2.38}$$

L_{ls} and L_m are the leakage inductance of the stator windings and the mutual inductance between them. Similar equations hold for the rotor windings.

We now define the equivalent rotor and stator inductances L_l and L_r as:

$$L_s = L_{ls} + L_m \qquad\qquad L_r = L_{lr} + L_m$$

where L_{lr} pertains to the rotor leakage inductance, and then apply the Clarke transformations to eqs. (2.37) and (2.38) (and to the corresponding rotor equations) to obtain biphasic versions of them. Up to now each variable is still defined in their own frame of reference: the stator variables in the axes $\alpha\beta$, aligned with the stator windings, and the rotor variables in a biphasic frame of reference aligned with the rotor windings. We use the Park transform (see eq. (2.26)) to get the projections of the rotor variables in the $\alpha\beta$ frame to get:

$$\vec{v}_s = r_s\vec{i}_s + \frac{d\vec{\psi}_s}{dt} \tag{2.39}$$

$$\vec{v}_r = r_r\vec{i}_r + \frac{d\vec{\psi}_r}{dt} - J\omega_r\vec{\psi}_r \tag{2.40}$$

$$\vec{\psi}_s = L_s\vec{i}_s + L_m\vec{i}_r \tag{2.41}$$

$$\vec{\psi}_r = L_r\vec{i}_r + L_m\vec{i}_s, \tag{2.42}$$

These equations can be solved for two variables (out of the four: $\vec{i}_s$, $\vec{i}_r$, $\vec{\psi}_s$ and $\vec{\psi}_r$) to be taken as the motor's electromagnetic state. An usual convention is to choose $\vec{i}_s$ and $\vec{\psi}_r$ [33]. In this way, and noting that $\vec{v}_r = 0$ for the squirrel cage motor, the following equations are obtained:

$$\tau_\sigma \frac{d\vec{i}_s}{dt} + \vec{i}_s = \frac{1}{r_\sigma}\vec{v}_s - \frac{k_r}{r_\sigma \tau_r}(J\omega_r\tau_r - 1)\vec{\psi}_r \tag{2.43}$$

$$\tau_r \frac{d\vec{\psi}_r}{dt} + \vec{\psi}_r = L_m\vec{i}_s + J\omega_r\tau_r\vec{\psi}_r, \tag{2.44}$$

with

$$\tau_\sigma = \sigma {}^{L_s}\!/\!{}_{r_\sigma} \qquad r_\sigma = r_s + k_r^2 r_r$$
$$\tau_r = {}^{L_r}\!/\!{}_{r_r} \qquad k_r = {}^{L_m}\!/\!{}_{L_r}$$
$$\sigma = 1 - {}^{L_m^2}\!/\!{}_{L_s L_r}.$$

Equations (2.43) and (2.44) describe the electromagnetic dynamics of the motor. In this form, the model of the asynchronous motor, should look familiar from a control theory perspective: they describe first order dynamics, plus a couple of exotic non-linear terms, which account for the interactions between stator and rotor windings states, and of these with the relevant mechanical variables: rotor speed and position. To complete the descriptions of this interactions, however, we still need to find an expression for the electrical torque developed by the machine.

2.3.2. Electromechanical conversion

In order to find an expression for the torque developed by the induction motor we make a power balance: considering the effects we took into account for the electromagnetic modeling of the machine, we know that the energy flowing to it will either be burnt as Joule losses (r_s), stored as magnetic fields (L_s, L_r and L_m), or transformed into kinetic energy. Using eqs. (2.39), (2.40), (2.41) and (2.42), we can calculate the total power being consumed by the motor as:

$$P_{\text{total}} = \frac{3}{2}\left(\vec{v}_s \cdot \vec{i}_s + \vec{v}_r \cdot \vec{i}_r\right) \tag{2.45}$$

$$= \frac{3}{2}\left(r_s|\vec{i}_s|^2 + r_r|\vec{i}_r|^2\right) + \frac{3}{2}\frac{d}{dt}\left(\frac{L_{ls}}{2}|\vec{i}_s|^2 + \frac{L_{lr}}{2}|\vec{i}_r|^2 + \frac{L_m}{2}|\vec{i}_s + \vec{i}_r|^2\right)$$
$$+ \frac{3}{2}\omega_r L_m\left(\vec{i}_r \times \vec{i}_s\right) \tag{2.46}$$

The factor $\frac{3}{2}$ accounts for the scaling done by the Clarke transformation.

Just by inspecting (2.46) it becomes clear that the first and second terms of its right side pertain to the Joule losses and the power being stored in the leakage and magnetizing inductances. The third term must pertain to the energy being transformed to kinetic energy. An expression for the torque T_e can be now found by equating the third term in the right side of (2.46) to the rate of change of the kinetic energy in the rotor:

$$P_k = T_e \frac{\omega_r}{p},$$

p is the pole pair number of the stator windings. With this we obtain:

$$\begin{aligned} T_e &= \frac{3}{2} p L_m \left(\vec{i}_r \times \vec{i}_s \right) \tag{2.47} \\ &= \frac{3}{2} p \frac{L_m}{L_r} \left(\vec{\psi}_r \times \vec{i}_s \right) \tag{2.48} \\ &= \frac{3}{2} p \frac{L_m}{\sigma L_s L_r} \left(\vec{\psi}_r \times \vec{\psi}_s \right). \tag{2.49} \end{aligned}$$

The description of the motor dynamics can be now completed with the equation for the rotor speed:

$$\frac{d\omega_r}{dt} = \frac{p}{J_m} \left(T_e - T_l \right), \tag{2.50}$$

where J_m represents the rotor inertia and T_l the torque developed by the load.

2.3.3. Model of the induction motor in a rotatory frame

Equations (2.43) and (2.44) can be defined in an arbitrary rotatory frame of reference, using the Park transformation with an angle θ_k, $T^{-1}(\theta_k) = \left[\begin{smallmatrix} \cos(\theta_k) & \sin(\theta_k) \\ -\sin(\theta_k) & \cos(\theta_k) \end{smallmatrix}\right]$

$$\tau_\sigma \frac{d\overset{\circlearrowleft}{i}_s}{dt} + \overset{\circlearrowleft}{i}_s = \frac{1}{r_\sigma} \overset{\circlearrowleft}{v}_s - J\omega_k \tau_\sigma \overset{\circlearrowleft}{i}_s - \frac{k_r}{r_\sigma \tau_r} (J\omega_r \tau_r - 1) \overset{\circlearrowleft}{\psi}_r \tag{2.51}$$

$$\tau_r \frac{d\overset{\circlearrowleft}{\psi}_r}{dt} + \overset{\circlearrowleft}{\psi}_r = L_m \overset{\circlearrowleft}{i}_s - J(\omega_k - \omega_r) \tau_r \overset{\circlearrowleft}{\psi}_r, \tag{2.52}$$

with

$$\begin{aligned} \overset{\circlearrowleft}{v}_s &= \begin{bmatrix} v_d \\ v_q \end{bmatrix} = T^{-1}(\theta_k) \vec{v}_s, & \overset{\circlearrowleft}{i}_s &= \begin{bmatrix} i_d \\ i_q \end{bmatrix} = T^{-1}(\theta_k) \vec{i}_s, \\ \overset{\circlearrowleft}{\psi}_r &= \begin{bmatrix} \psi_d \\ \psi_{rq} \end{bmatrix} = T^{-1}(\theta_k) \vec{\psi}_r, & \omega_k &= \frac{d\theta_k}{dt}. \end{aligned}$$

A very useful rotatory frame is one aligned with the magnetic field of the rotor, the dq frame. Here, the quadrature component of the rotor flux is null $\psi_{rq} = 0$, which means that the d component of the eq. (2.52) is reduced to:

$$\tau_r \frac{d\psi_d}{dt} + \psi_d = L_m i_d \tag{2.53}$$

and ω_k can be solved from the q component of eq. (2.52) [1]

$$\omega_k = \omega_r + \frac{L_m}{\tau_r} \frac{i_q}{\psi_d}. \tag{2.54}$$

Finally the expression for the torque in (2.48) can be written:

$$T_e = \frac{3pk_r}{2} \psi_d i_q \tag{2.55}$$

Equations (2.51), (2.52), (2.53) and (2.53) lead to the core ideas behind *field oriented control* (FOC), which is the classic control scheme for asynchronous motors: the magnitude of the rotor flux $|\overset{\circlearrowleft}{\psi_r}| = |\psi_d|$ can be controlled manipulating the d component of the stator current i_d and, if it is controlled to be approximately constant, the electrical torque T_e can be controlled independently by manipulating the q component of the stator current i_q. Moreover, both components of the stator currents can be controlled with null stationary state error using simple *proportional-integral* controllers, for constant flux and torque reference values.

2.3.4. State space representation

Incorporating eq. (2.50) to (2.43) and (2.44) the following state space representation of the dynamic behaviour of the induction motor can be written:

$$\frac{d}{dt} \begin{bmatrix} \vec{i}_s \\ \vec{\psi}_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau_\sigma} \left(\frac{1}{r_\sigma} \vec{v}_s - \vec{i}_s - \frac{k_r}{r_\sigma \tau_r} (J\omega_r \tau_r - 1) \vec{\psi}_r \right) \\ \frac{1}{\tau_r} \left(J\omega_r \tau_r \vec{\psi}_r - \vec{\psi}_r + L_m \vec{i}_s \right) \\ \frac{p}{J_m} \left(T_e - T_l \right) \end{bmatrix} \tag{2.56}$$

The state being $x = [i_{s\alpha}\ i_{s\beta}\ \psi_{r\alpha}\ \psi_{r\beta}\ \omega_r]^T$, the input $u = \vec{v}_s = [v_\alpha\ v_\beta]^T$ and the disturbance $d = T_l$. The torque developed by the motor is given by:

$$T_e = \frac{3}{2} p \frac{L_m}{L_r} \left(\vec{\psi}_r \times \vec{i}_s \right) \tag{2.57}$$

2.3.5. Model parameters

As mentioned earlier, the models introduced in this chapter are simple representations of very complex physical phenomena. This is particularly true in the case at hand. The construction of the induction motor, which does not include magnetic materials imposing a magnetic flux, but relies on induction of the rotor flux, makes it extremely reliable, but very sensitive to iron effects: saturation, magnetic hysteresis, Eddie currents.

[1] Note that ω_k is the frequency of the stator currents if the $\alpha\beta$ plane. From here one of the names of the induction motor, asynchronous machine: the speed of the rotor does not match frequency of the stator voltages and currents and the difference is proportional to the load, or the torque being developed by the motor. This becomes clear with eq. (2.55).

Just as in the case of the synchronous motor in last section, the parameters of our model of the induction motor need to fitted, so the dynamics of the model match the real dynamics under the conditions imposed by the control system.

The following is the procedure used in this work to fit the model parameters.

- A first approximation for these parameters is obtained with the well known no-load and locked rotor tests [32]. These test are used to estimate the ratio between L_m and L_s.
- The stator winding resistance r_s is measured directly, using an Ohmmeter on the terminals of the motor.
- The composite parameter σL_s is estimated out of the measured rate of change in the stator currents, applying voltage pulses to the motor terminals, on top of a constant voltage. The latter is used to magnetize the motor and avoid hysteresis effects.
- With this estimates, a rudimentary torque controller, based on field-oriented control is implemented, using the currents model as rotor flux observer (see Appendix B). A step torque reference is feed to it and the value for the composite parameter τ_r is adjusted until the speed is measured to change at a constant rate. When the value of τ_r is not correctly fitted, the observed flux tends to lag behind or overtake the actual induced flux and pulsating torque is produced.

These estimations, together with the assumption $L_s = L_r$ and eq. (2.44) are used to solve the electrical parameters.

The rudimentary torque controller, adjusted with the parameters estimated so far, is used to apply a step torque. The rotor inertia J_m is estimated out of rate of change obtained for the speed (see Sec. 2.2.5).

The fitted parameters, which are used for the experimental tests are given in table 2.2.

Table 2.2.: Parameters of the Induction Motor

$p = 1$
$r_s = 2.5\,\Omega$
$r_r = 1.57\,\Omega$
$L_s = 324.8\,\text{mH}$
$L_r = 324.8\,\text{mH}$
$L_m = 318.0\,\text{mH}$
$J_m = 7.53 \times 10^{-3}\,\text{kg}\,\text{m}^2$

2.4. Discrete-time Models

The control schemes presented in this work were developed to be implemented on digital computers and they all make use, as a central element, of predictions of the states to be controlled. In this sense, discrete-time models are required in each case.

The only particular requirement imposed on these models is that the effect of the input u must be patent in the prediction of all relevant states at the end of the prediction horizon, we call this direct feed-through [34, 35]. The reason for this lies in the fact that the *control decision*: the actuation u to be applied in the next sampling instant, is calculated upon the effects of the actuation on the predictions. In all the applications presented in this work the prediction horizon is one sampling period, Although, this is not completely true for the control systems presented in part IV, where control of higher order dynamics force the use of longer predictions horizons. The techniques used in those cases are presented in the corresponding chapters.

In each case, the discrete model of the system is synthesized by using the Taylor series expansion of the solution of the continuous time state space representation of the system [36]. With this, each component i of the discrete-time state space representation for the system is given by:

$$x_i[n+1] = x_i[n] + \sum_{l=1}^{N_i} \frac{h^l}{l!} \frac{d^l x_i}{dt^l}\bigg|_{t=hn} \tag{2.58}$$

Parameters h and N_i represent the sampling period and the order of the Taylor expansion, respectively. The later is adjusted for each state component to get direct feed-through from the input to every state in one sampling period: it corresponds to the lowest l for which u appears in $\frac{d^l x}{dt^l}$. This is also regarded as the order of the dynamics being controlled.

The discrete-time space representation of the system can be now written as:

$$x[n+1] = f(x[n], u[n], d[n]). \tag{2.59}$$

In this form, the system model is used in the control algorithms developed in this work, to predict the future state of the system.

3. Control

In the last chapter, models for the PMSM and IM drives were developed and put in a compact form, the steady state representation, to describe all the relevant dynamics. In this chapter we develop the three major control concepts, which constitute the context where the main contributions of this theses are located:

Field oriented control, the classical control method for variable speed drives. From the control's perspective, a change of variables that allows to write a simple expression for the torque as a function of the system state, and more importantly, separates the flux and torque control problems in the induction machine.

Finite-set model predictive control, a relatively new control strategy, proposed originally for the control of power electronics. It constitutes an ad-hoc implementation of the model predictive control method in power converters. The method exploits certain properties of these systems to enable vast simplification of a relatively complex control problem: control of a switched, multivariable system with constraints in the inputs and the states, the method solves this problem online and implemented, in commercial digital signal processors, reaches sampling rates in the kHz range.

Time optimal control. In general, the idea behind model predictive control is to solve an optimal control problem, this is, derive control policies out of an optimization problem on the system variables, which takes the system dynamics as constraints. The criteria to optimize might be diverse: steady state error, fuel consumption, etc. The particularity of the model predictive control method is that it solves this problem in a finite time horizon. Time optimal control refers to a specific criteria: the time to reach the control goal.

In the following sections we introduce and develop these subjects in the ways that they are relevant for the control methods proposed in this work.

3.1. Field Oriented Control

Felix Blashcke presented field oriented control (FOC) during the seventies [10] and is still the *de facto* standard control scheme for high performance variable speed drives.

The main idea behind FOC is the description of the electrical dynamics of the motor in a rotating frame of reference, aligned with the rotor flux, hence the name *field oriented control*. With this, in both machines, the PMSM and the IM, the problem of controlling the torque developed by the motor is reduced to the control of a composite variable: the quadrature component of the stator current in the dq frame i_{sq}, which on the one hand is proportional to the torque, and on the other is characterized by first order dynamics and can be controlled by manipulating the corresponding voltage component v_{sq}. Moreover, in the induction motor, the problems of controlling the rotor flux and torque are separated in two practically orthogonal problems: the flux's magnitude is controlled by manipulating the i_{sd} current component, which in turn is controlled by manipulating v_{sd}. Once the magnitude of the flux is stable, the torque can be controlled by manipulating i_{sq}.

In more specific terms and with a little bit of notation abuse: field control reduces the problem of controlling the torque in the PMSM to the control of two plants with the form:

$$i_d \approx \frac{1}{r_s} \frac{1}{\frac{L_d}{r_s} s + 1} v_d \tag{3.1}$$

$$T_e \propto i_q \approx \frac{1}{r_s} \frac{1}{\frac{L_q}{r_s} s + 1} v_q \tag{3.2}$$

and the problem of controlling the torque in the IM to controlling:

$$i_q \approx \frac{1}{r_\sigma} \frac{1}{\tau_\sigma s + 1} v_q \tag{3.3}$$

$$i_d \approx \frac{1}{r_\sigma} \frac{1}{\tau_\sigma s + 1} v_d \tag{3.4}$$

$$\psi_r \approx \frac{L_m}{\tau_r s + 1} i_d \tag{3.5}$$

$$T_e \propto \psi_r i_q. \tag{3.6}$$

If we consider that usually $\tau_r \gg \tau_\sigma$, the electrical torque T_e and rotor flux ψ_r can be controlled independently.

The classic speed control schemes based on FOC are depicted in fig. 3.1 for the PMSM and fig. 3.2 for the IM. In these diagrams, all the variables on the left side of the Park transforms T and T^{-1} are defined in the dq frame. Both schemes clearly define two parallel control structures, one for each component of the stator currents, i.e., one for the torque and another for the flux magnitude. The scheme for the IM also includes a block to estimate $\vec{\psi}_r$, since it is not usual that drives include direct measurements for it. This block is described in the appendix B.

These control schemes crystallize the aforementioned ideas in the simplest way possible and, consequently, suffer from many deficiencies. In the scheme for the PMSM, for

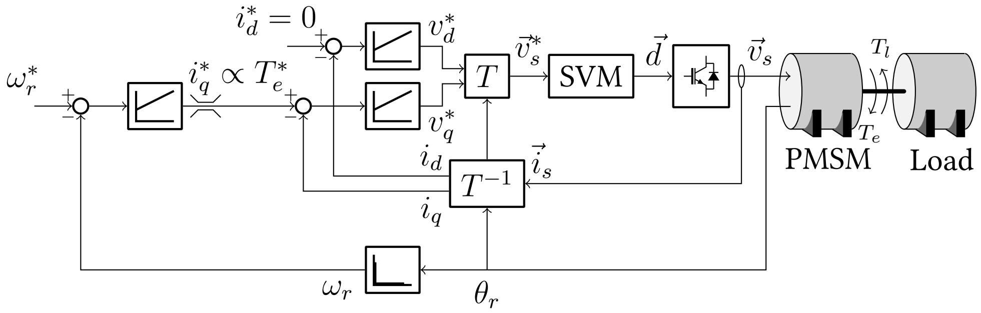

Figure 3.1.: Field oriented control scheme for speed control of the permanent magnet synchronous motor. All the controllers, for i_d, i_q and ω_r are implemented using proportional-integral controllers. The drive is assumed to consider only currents, and rotor angle measurements. The rotor speed is estimated using a derivator. The duty cycles for the transistors in the voltage source inverter are calculated by a space vector modulation block (SVM).

example, the reference for i_d is set to zero $i_d^* = 0$, but, as discussed before in section 2.2.3, maximum efficiency is achieved when i_d and i_q satisfy eq. 2.35. This can be accounted for in the scheme, calculating references for i_d and i_q using eqs. (2.30) and (2.35). Furthermore, the scheme will naturally inherit the deficiencies and strengths of the control structure chosen. A general comparison between different control structures is beyond the scope of this work[1], but still some generalities can be enunciated:

- Proportional integral (PI) controllers, such as those used in the schemes in figures 3.1 and 3.2 will achieve steady state error zero for constant references and load torque, but their efficacy compensating all the terms after the first on the right side of eqs. 2.28, 2.51 and 2.52, regarded as disturbance, is limited by their bandwidth. This can be however improved, by introducing feed-forward structures to compensate for the terms $-L_q i_q \omega_r$ and $(L_d i_d + \psi_m)\omega_r$ in the i_d and the i_q equations respectively.

- The cascaded control structure imposes restrictions on the relationship between the bandwidths of the inner and the outer controllers.

- PI controllers can be tuned using classical frequency domain tools. This is a very important strength of linear controllers in general, since electrical engineers are very familiar with these tools: just as good auto mechanics can diagnose their patients listening to the sounds they make, electrical engineers characterize their systems by *listening* his measurements, i.e., looking at their frequency spectra, and thus tune

[1] For a thorough study on fundamental limitations in classical control theory, please refer to the chapters in part III and chapter 24 in [37]. For a comparison between field oriented control and predictive torque control (to be introduced in the next section) for the induction motor, refer to [38].

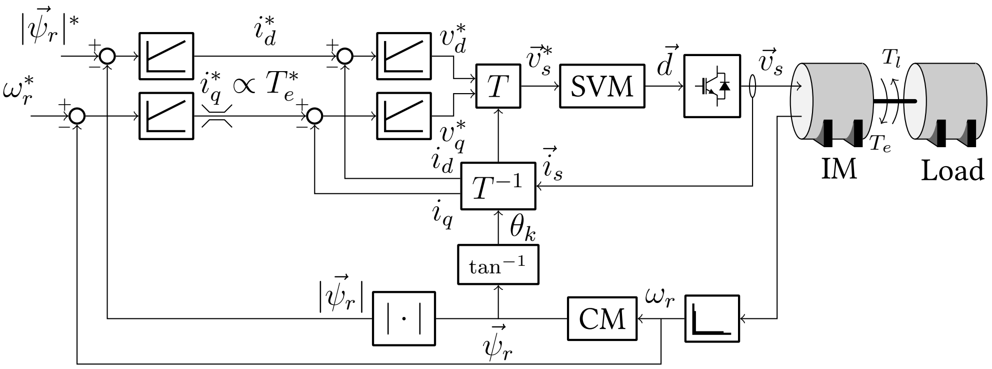

Figure 3.2.: Field oriented control scheme for speed control of the induction motor. All the controllers, for i_d, i_q, $|\vec{\psi}_r|$ and ω_r are implemented using proportional-integral controllers. The drive is assumed to consider only currents, and rotor angle measurements. The rotor speed is estimated using a derivator and the flux is estimated using the currents model (CM). The duty cycles for the transistors in the voltage source inverter are calculated by a space vector modulation block (SVM).

their controllers. Dealing with constraints, however, is not a speciality of linear controllers.

Numerous attempts to circumvent some of these issues and also patent enforcement, have been made.

A very popular alternative to field oriented control is called direct torque control (DTC) [11, 12]. In this scheme the variables rotations, the PI controllers and the space vector modulation are replaced with hysteresis controllers, a look up table and a function that classifies the position of the rotor flux in the $\alpha\beta$ plane. The hysteresis controllers deal with the flux magnitude and torque directly (hence the name), the look up table takes the outputs of these controllers and picks the switching state to be applied based on the position of the rotor flux vector in the $\alpha\beta$ plane: for each point in this plane there is one particular voltage vector that will increase, decrease or maintain the magnitude of the flux and at the same time increase, decrease or maintain the torque. It should be noted that, although implicitly, the principles of field oriented control permeate the workings of the DTC scheme: the position of the flux is fundamental.

The high popularity of this scheme resides in its simplicity (and consequent robustness) and fast dynamics, but it suffers from high torque distortion, and an inherent property of hysteresis controllers: distributed spectrum, which may excite non-modeled resonant modes. Moreover, the frequency domain tunning tools are no longer available. As in many other disciplines, in drives control there is neither such thing as a free lunch [39].

The control scheme presented in the following section: finite set model predictive control, constitutes the main basis for all the work presented in this thesis and bears a lot of resemblance with DTC in the sense that the control policy amounts to chose a single switching state and, as a consequence, their performances are very similar. Nevertheless, the way in which the switching state is chosen is completely different and in the particular method described, which is designed for drive control, its based on the ideas behind field oriented control.

3.2. Finite-Set Model Predictive Control

Finite-set model predictive control (FS-MPC) is an implementation of the model predictive control (MPC) method, in the context of power converters.

The MPC method works by iteratively solving a finite-horizon optimal control problem: let

$$x[n+1] = f(x[n], u[n]) \tag{3.7}$$

be a discrete-time model of the system, describing all the relevant system dynamics (for the sake of simplicity we asume that the system dynamics are time-invariant), where x and u represent the internal state of the system and the actuation, both defined in their particular respective domains: $x \in \mathscr{X}$, $u \in \mathscr{U}$. We call the model represented by eq. (3.7) the *predictive model.*

The receding horizon optimal control problem is formulated by first defining a sequence for the actuation of length N (the prediction horizon):

$$\mathbf{u} = \{u[n], \dots, u[n+N-1]\} \in \mathscr{U}_N. \tag{3.8}$$

which, when applied to the system in eq. (3.7) for a particular initial condition $x[n]$, generates a sequence for the state:

$$\mathbf{x}^{\mathbf{u}} = \{x[n+1], \dots, x[n+N]\}. \tag{3.9}$$

The control goals are crystallized in a scalar cost function:

$$F_c(\mathbf{x}, \mathbf{u}, k). \tag{3.10}$$

The optimal control policy is found by solving:

$$\mathbf{u}^* = \pi(x) = \underset{\pi \in \mathscr{U}}{\arg\min} \{F(\mathbf{x}, \mathbf{u}, k)\} \tag{3.11}$$

constrained to $\mathbf{x}^{\mathbf{u}^*} \in \mathscr{X}^N$. The optimal actuation for the point in the state space x (the initial condition $x[n]$) is then the first component of the sequence $\pi(x)$.

The dependence from $\pi(\cdot)$ of x stresses that the result is a state feedback policy and the fact that the solution must fulfil $\mathbf{x}^{\mathbf{u}^*} \in \mathscr{X}^N$ and $\mathbf{u} \in \mathscr{U}^N$ accounts for state and actuation constraints.

In this context, all the available synthesis methods are based on the dynamic programming principle (DPP) [5]. For instance, for the particular case, when $X = \mathbb{R}^m$, $U = \mathbb{R}^n$, the cost function is quadratic, the systems dynamics are linear and N tends to infinity, this is the well-known linear quadratic regulator problem [3], where the application of the DPP leads to a Riccati equation for the computation of the optimal feedback policy. Different algorithms and approaches exist to solve this problem for different families and parametrizations of the system model, the constraints and cost functions. [7, 40]

In particular the FS-MPC method, which was proposed in the context of power electronics [13], exploits the fact that the actuation set $\mathscr{U}$ is finite and rather small (it corresponds to the valid switching states in the power converter) and that the dynamics to be controlled are simple enough, so that good quality control can be achieved with a short prediction horizon. The algorithm solves the optimal control problem online at each sampling instant, by evaluating the cost function for each member of the $\mathscr{U}$ set; the one producing the lowest cost is the optimal control. This is a major particularity of the FS-MPC method, compared to the linear quadratic regulator, for example, and other solvers that do include state constraints: the solver proposed in [41], for example, solves the optimal control offline, and the solution takes the form of a set of piecewise linear controllers, which are valid for specific regions of the state space. Recent research around FS-MPC has shown that its implementation can be practical, even for complicated power converters, involving many states and bigger actuation sets [14, 42]. The algorithm be implemented in commercial DSPs and even in FPGAs [17]. As hinted before, the systems treated so far involve relatively simple, or low order dynamics, where the actuation has direct impact on the states, hence, short predictions horizons can be used. If this were not the case, the complexity of the control algorithm would render it impractical: the number of possible actuation sequences grows exponentially with the prediction horizon. In the context of dynamic programming this is known as the curse of dimensionality.

The next section shows how FS-MPC is implemented in an example, which is fundamental for the rest of the work developed in this thesis.

3.2.1. Finite-set model predictive current control for the permanent magnet synchronous motor

In a first approach, we can use the FS-MPC algorithm, to replace the inner current control loops in the FOC scheme, described in last section, as depicted in Fig. 3.3

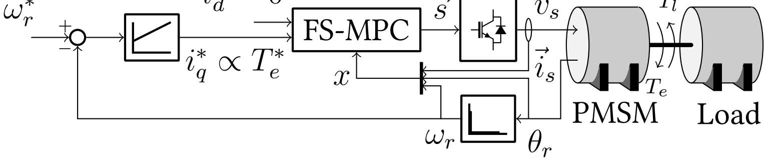

Figure 3.3.: Control scheme: speed control for the permanent magnet synchronous motor using finite-set model predictive current control. The feedback x represents the whole state of the motor: $x = [i_\alpha \; i_\beta \; \omega_r \; \theta_r]^T$: although the task of the FS-MPC is only to control the stator currents, the whole state is required to evaluate the system model.

The dynamics of the system to be controlled are described by the ODE (see Sec. 2.2):

$$\frac{d\vec{i}_s}{dt} = L_s^{-1}\left(\vec{v}_s - r_s\vec{i}_s - \frac{\partial L_s}{\partial \theta}\omega_r\vec{i}_s - J\omega_r\vec{\psi}\right) \tag{3.12}$$

A discrete version of this equation is synthesized with the method described in Sec. 2.4, with $N = 1$ for both components of $\vec{i}_s$ (which amounts to using the forwards Euler method). With this, we get a description for the system dynamics, the predictive model:

$$x[n+1] = f(x[n], u[n]) \tag{3.13}$$

with $x = [i_\alpha \;\; i_\beta]^T$ and $u = [v_\alpha \;\; v_\beta]^T$. When evaluating $f(\cdot)$ in (3.13), ω_r and θ_r are regarded as constants, their value is nevertheless updated each sampling instant with new measurements.

FS-MPC requires the actuation set to be finite and discrete, for this we assume that the motor is driven by a two-level voltage source inverter and that during the whole of a sampling period h, a single switching state is applied (see sec. 2.1):

$$u \in \mathscr{U} = \{\vec{v}_0, \, \dots \, , \vec{v}_7\} \tag{3.14}$$

The control goal for this scheme is to minimize the error of the two current components with respect to their references. In the scheme depicted in Fig. 3.3 the later are given in the dq frame of reference: $i_d^* = 0$ and $i_q^* \propto T_e^*$, and our model in eq. (3.12) is defined in the $\alpha\beta$ frame, however we can still write the cost function for our problem as:

$$F_c(x) = i_d^2 + (i_q - i_q^*)^2, \tag{3.15}$$

with

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta_r) & \sin(\theta_r) \\ -\sin(\theta_r) & \cos(\theta_r) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{3.16}$$

$$\overset{\circlearrowleft}{i}_s = T^{-1}\vec{i}_s. \tag{3.17}$$

A flowchart for the control algorithm is depicted in Fig. 3.4 and simulation results are shown in Fig. 3.5. The latter also include results with the FOC scheme, these are not included to draw a thorough comparison between both methods, but to highlight the general properties of FS-MPC with respect to a more familiar scheme.

The dynamics obtained with the FS-MPC controller is very similar to what would be obtained by using DTC or hysteresis controllers in general: the response is very fast during the transient and in steady state the frequency spectrum of the currents show energy at all frequencies, since the switchings occur with no particular periodicity.

The difference with FOC is evident during the transient, where FS-MPC is much faster, and also in the frequency spectrum of the stator current. In the one obtained with the FOC scheme the high frequency components are concentrated around the switching frequency 10kHz.

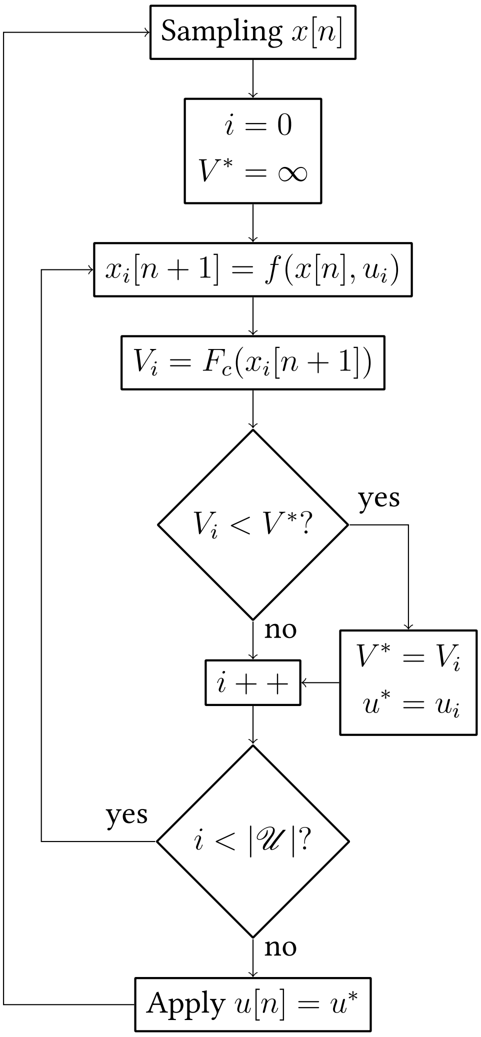

Figure 3.4.: Flow chart for the finite-set model predictive algorithm.

One aspect that renders the comparison between FOC and FS-MPC complex is the switching frequency (times per second a switch is turned on and off). This is critical in solid state power converters, since it determines their thermal design. In FOC the switching frequency is fixed by the modulator (pulse width or space vector modulator), whereas in FS-MPC, switching occurs only at the sampling instants and only when it results in a decrease of the cost, thus, switching patterns in FS-MPC have no fixed periodicity and depend heavily on the operating point. Hence, in general only an bound for the switching frequency in FS-MPC can be given: two times the sampling frequency $f_s = \frac{1}{2h}$. Moreover, when using modulators, how short the time interval between two switching events can be is only limited by the implementation of the modulators (the clock frequency of its internal counters, in digital implementations). This allows for very fine mean value control. In FS-MPC the control is very coarse and finer control can only be achieved with

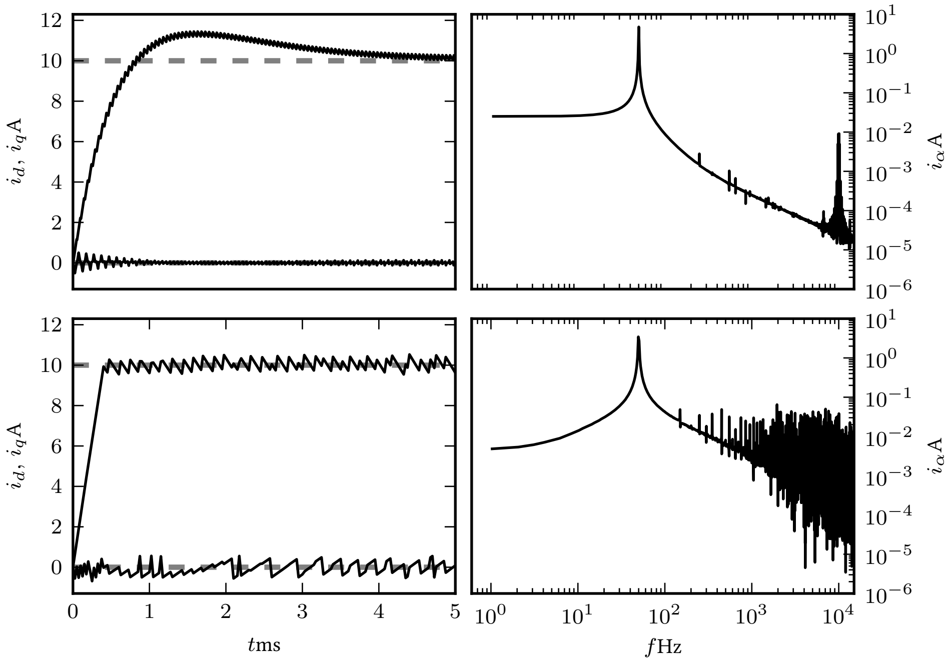

Figure 3.5.: Simulation results for speed control using FOC (top) and FS-MPC (bottom), the later replacing the inner currents controllers in the FOC scheme. In this test, the rotor speed starts at $\omega_r = 2\pi \cdot 30\text{rad}^\text{e}$ and its reference is set at $\omega_r^* = 2\pi \cdot 50\text{rad}^\text{e}$. The motor is the whole time being loaded with $T_l = 5.0\text{N m}$. The sampling periods used are $h = 50\mu\text{s}$ and $h = 25\mu\text{s}$ for the FOC and the FS-MPC schemes respectively. The frequency spectrum for i_α was in both cases calculated in steady state. To obtain these results, the FOC scheme in Fig. 3.1 was modified to compensate for the crossed couplings between i_d and i_q.

shorter sampling periods, which increases the switching frequency.

As stated before, the FS-MPC algorithm solves the optimal control problem online, taking the sampled state as initial condition. This is practical as long as the prediction horizon is short. By looking at the chart in Fig. 3.4 the exponential growth of complexity with an increasing prediction horizon becomes evident: as it is, the algorithm considers a prediction horizon of one sampling period; increasing the prediction horizon by one step would require a nested loop with a second subscript for $u[n]$. Further increases in the prediction horizon add more subscripts: the number of iterations for a prediction horizon N is then given by:

$$n_{it} = |\mathscr{U}|^N . \tag{3.18}$$

Methods to estimate a minimal length for the prediction horizon to ensure stability do exist for specific settings [43], but no general assessments can be done, when state constraints need to be considered or the goal is to fix specific dynamics. It natural to intuit, however, that more complex dynamics, subject to state constraints will require longer prediction horizons.

In FOC the gains of the PI controllers are tuned using frequency domain tools, such as pole placement. This particular FS-MPC scheme does not require tunning: the magnitudes in the cost function have the same units and can be compared without introducing scaling factors. This scheme uses the system model to calculate the actuation, so no gains need to be calculated. This has also as consequence, that all the effects in the model are taken into account: the feed-forward structures included to compensate for the cross couplings between i_d and i_q in the FOC scheme are inherent in the FS-MPC scheme.

As hinted before in section 3.1, the lack of tunning can also be a drawback for FS-MPC: no tunning simplifies the implementation, but puts away a degree of freedom, which is usually useful for tackling problems such as measurement noise and unmodeled dynamics, but then again, these can also be treated as separate problems.

The work developed for this thesis takes the FS-MPC method as starting point and develops it further for more complex dynamics: flux and currents control in the induction motor, speed control for the synchronous and the induction motor, position control for the synchronous motor and torsional torque control in a two-mass-system. In each case, the aforementioned strengths and drawbacks of the FS-MPC method are leveraged and tackled to lead these control systems closer to their physical limits.

3.2.2. Computation delay

One aspect that should be taken into consideration, when implementing the FS-MPC method, is the time required to compute the control. Even if implemented in hardware allowing high parallelization, such as FPGAs, the computation delay can usually not be neglected.

A standard approach to deal with this delay is to shift the control problem one sampling period into the future, this is, if n is the current sample, calculate the control for $n+1$. The initial condition for this problem $x[n+1]$ is estimated using the system model:

$$x[n+1] = f(x[n], u[n]) \tag{3.19}$$

where $x[n]$ is the state measured at instant n and the control $u[n]$ is calculated during the last sampling period: between $t = (n-1)h$ and $t = nh$.

In the following chapters, the proposed control methods are designed assuming that there is no computation delay and the actuation $u[n]$ is instantly available after measuring the state $x[n]$. The implementations used for experimental verification, however, use this scheme to compensate the delay.

3.3. Time-Optimal Control

In this section, we review some classical results concerning the existence and synthesis of minimum time controllers which are relevant for the proposed design.

For motivation let us consider one issue that arises, when implementing model predictive controllers for more complex dynamics and more precisely: in the design of the cost function, is the lack of a consistent method to weight all the involved variables together. The MPC framework is formulated for dimensionless variables or variables with arbitrary units, this is not problematic when all the variables of interest represent the equivalent physical quantities, such as in the current controller introduced in the last section.

Quadratic functionals, as the one used in the formulation of the linear quadratic regulator, weight all states together using scaling factors. Physical meaning is given to these functions relating them to measures of energy in the system. This is convenient for stability analysis, since energy tends to deplete in real physical systems, and the cost function can be considered to be a Lyapunov function of the closed loop [43]. This is however not very useful, when the goal is to fix more specific dynamics to achieve a centralized control of variables with clear causal relationships. In the drives context, we could wish to control the speed and torque/currents of the motor together (and not separately using cascaded structures), for instance, to improve performance. It is clear, from the causal relationships between stator currents, torque and speed, that a cost function composed only of a sum of quadratic errors in the states, with respect to their steady state values, will only achieve a good performance with a relatively long prediction horizon: long enough for the controller to be able to *realize* that an early increase in the torque error will cause a decrease in the speed error and of the whole cost in the long run. The situation is further complicated by the introduction of constraints.

In this sense, a valid and appealing approach is to put the drive control problem in terms of a time-optimal control problem, i.e., to set the performance index to the time needed to steer the system from its current state to a prescribed target.

Let us consider a continuous time dynamical system given by

$$\frac{dx(t)}{dt} = f(x(t), u(t)), \quad x(0) = y, \tag{3.20}$$

where $x(\cdot) \in \mathscr{X} \subseteq \mathbb{R}^n$ denotes the state of the system and $u(\cdot) \in \mathscr{U} \subseteq \mathbb{R}^m$ is an admissible control signal. Our control goal is to steer the system to a target $\mathcal{T} \subset \mathscr{X}$ as for instance, to a ball around the origin. If for a control signal $u(t)$ we define the arrival time from y to the target as

$$t_y(u) = \inf_s \{s \in \mathbb{R}^+ : x(s; u(s)) \in \mathcal{T}\}, \tag{3.21}$$

the value function of the minimum time problem is given by

$$T(y) := \inf_{u(\cdot)\in\mathscr{U}} t_y(u). \tag{3.22}$$

To obtain the optimal controller two approaches are possible. On the one hand, it is possible to derive optimality conditions characterizing an open-loop controller via the application of the Pontryagin Maximum Principle [6]. Such an approach is relatively simple to implement, and can yield accurate trajectories at low cost. A different solution method relies on the application of the Dynamic Programming Principle (DPP) [5], and characterizes the value function as the viscosity solution of the following Hamilton-Jacobi equation over $\mathscr{X}$,

$$\sup_{u\in\mathscr{U}}\{-\nabla T(x)\cdot f(x,u)\}=1\,. \tag{3.23}$$

Since the value function is obtained for the whole state space (which can be computationally costly for high-dimensional dynamics), the optimal controller is expressed in feedback form

$$u^{*}(x)=\arg\min_{\pi\in\mathscr{U}}\{\nabla T(x)\cdot f(x,\pi)\}\,, \tag{3.24}$$

which can be implemented for online control once the value function has been computed. However, since an approximation of the state space must be introduced in order to solve eq. (3.23), the value function is defined over a finite grid of points, producing grid-dependent trajectories, which can exhibit an spurious chattering along the switching curves. Therefore, it is relevant to achieve a balance between the robustness of a feedback control and an accurate approximation of the switching structure of the system in order to yield adequate trajectories.

In the research conducted for this work, the minimum time optimal control problem is not explicitly solved for the whole dynamics described in chapter 2, but for simplified systems, which fairly describe the relevant dynamics in each case and, more importantly, the dynamics we wish to fix. This, in an effort to achieve a good trade-off between practicality of the algorithms and performance.

These simplified systems are the simple integrator:

$$\frac{dx}{dt}=\frac{1}{\tau}u\,, \tag{3.25}$$

the double integrator:

$$\frac{d}{dt}\begin{bmatrix}x_0\\x_1\end{bmatrix}=\begin{pmatrix}\frac{1}{\tau_0}u\\\frac{1}{\tau_1}(x_0-d)\end{pmatrix}, \tag{3.26}$$

the triple integrator:

$$\frac{d}{dt}\begin{bmatrix}x_0\\x_1\\x_2\end{bmatrix}=\begin{pmatrix}\frac{1}{\tau_0}u\\\frac{1}{\tau_1}(x_0-d)\\\frac{1}{\tau_2}x_1\end{pmatrix}, \tag{3.27}$$

and an harmonic oscillator:

$$\frac{d}{dt}\begin{bmatrix}x_0\\x_1\\x_2\\x_3\end{bmatrix}=\begin{pmatrix}{}^{u}\!/\!_{\tau_0}\\-\omega_c x_2+k_u u\\\omega_c x_1+k_d d\\{}^{x_2}\!/\!_{\tau_3}\end{pmatrix}, \tag{3.28}$$

Two approaches were used to solve the minimum time optimal control problem for these systems, both based in the well-known fact that the solution of this kind of problems often lead to bang-bang type of controllers, where the limit value of the actuation is always applied, with its sign depending on the position of the state with respect to a curve in the state space: the switching curve.

3.3.1. Backward induction

The first approach, based on the backward induction method, is used to solve the minimal control problem for discrete-time version of the double integrator.

The time optimal control problem for the discrete-time case in the same fashion as for the continuous-time case, this time for a dynamic system with the form:

$$x[n+1] = f(x[n], u[n]) \,, \quad y = x[0] \tag{3.29}$$

The value function in eq. (3.22) is solved in this case using a numerical implementation of the backward induction principle, which consists of thinking backwards in time: we start from the end of the problem, where the state x has already reached its target $\mathcal{T}$, and apply all the feasible values for $u \in \mathcal{U}$ to the system, with the time running backwards. The points reached are stored and the same procedure is repeated starting from each of these point, one step at a time. The steps required to reach a point starting from $\mathcal{T}$ equates to $T(x)$. The development of a particular trajectory stops when it reaches a point, which was already reached before: this means that another trajectory reaches this point in less time.

This approach is applied in section 5.1 find a solution for the time-optimal control problem for the double integrator, leveraging the fact that both the time and the actuation are quantized.

3.3.2. Switching time parametrization method

It is well-known that minimum time optimal control often leads to bang-bang type of controllers. More precisely, the classical results asserts that for a linear system with real eigenvalues, the minimum time controller will be bang-bang with at most $n-1$ switchings, where n is the dimension of the state space; there are different extensions of this result for the nonlinear case, but no general assertion can be made regarding the number of switchings in the presence of constraints in the state. Assuming that this number is finite, the computation of the optimal controller, in the continuous-time case is reduced to the identification of the switching times. A simple solution method is then given in the scalar control case [44], by prescribing a switching sequence

$$\mathbf{u} = \{-\hat{u}, \hat{u}, -\hat{u}, \ldots, \hat{u}\} \,, \tag{3.30}$$

of size $m \geq n$, with corresponding times

$$\boldsymbol{\Delta}\mathbf{t} = \{\Delta t_1, \ldots, \Delta t_m\} \,, \tag{3.31}$$

As a result, the minimum time problem can be cast as

$$\min_{\mathbf{\Delta t}} T := \sum_{i=1}^{m} \Delta t_i \tag{3.32}$$

subject to

$$\frac{dx}{dt} = f(x, \mathbf{u})\,, \tag{3.33}$$

$$x(0) = y\,, \tag{3.34}$$

$$x(T) \in \mathcal{T}\,, \tag{3.35}$$

$$\Delta t_i \geq 0\,, \forall i\,. \tag{3.36}$$

If a bang-bang structure is assumed, this idea can be computationally implemented with a redundant amount of switchings, which shall be shrunk to zero if the optimal solution is achieved.

Such a scheme was implemented using a general purpose numerical optimization algorithm, in this case the *Constrained Optimization BY Linear Approximation* algorithm (COBYLA) [45]. to investigate the properties of the state behaviour under a time-optimal control, for the triple-integrator and the two-mass system, and to hold the solutions as reference. In this form the STPM works as a shooting method: in each iteration the system dynamics are solved (simulated/predicted) using a different actuation signal $\mathbf{u}$ for the i.c. y, the optimization algorithm searches for the solution in the space for $\mathbf{\Delta t}$, $\mathbb{R}^m$, checking (3.35) and evaluating T, defined in (3.32). Constraints are taking into account checking if

$$x(T) \in \mathscr{X}\,. \tag{3.37}$$

The number of iterations required to find a solution with this scheme varies depending on the initial condition and the precision required. The design of controllers described in the last part of this thesis are based on a simplification of this scheme.

Part II.

First Order System

4. Torque control for the permanent magnet synchronous motor

In section 3.2, the current controllers in the FOC scheme for the PMSM, were replaced with finite-set model predictive controllers. In this chapter this scheme is further developed in two directions: first, the scheme is modified to get a torque controller accounting for the maximum torque per ampere operation of the PMSM, and second, voltage modulation is integrated in the scheme, to obtain fixed switching frequency an and allow longer sampling periods.

In section 3.2 a simple property of the MPC scheme was highlighted: regardless of the predictive model being defined in $\alpha\beta$ variables, we could write the cost function using dq variables. We can extend this idea to write the cost functions in terms of the electrical torque, being developed by the motor, and the compliance of the conditions for MTPA operation.

One disputable drawback of the FS-MPC scheme is the reduction of the actuation set to the set of voltage vectors produced by the valid switching states of the power converter. It is clear that this can be leveraged to design a very simple algorithm to solve a relatively complex optimal control problem. The drawback of this is also clear: applying one of these voltage vectors for a whole sampling period results in a very coarse control or a very hard trade-off between sampling time, computation complexity and switching frequency. This issue is addressed here, by integrating the idea of space vector modulation with the FS-MPC scheme. With this, the form taken by the ripple is completely predictable and is given by the pulse width modulation scheme applied. To achieve this, the pursued ideas are first analyzed in the context of a very simple system that, nevertheless, represents the dynamics at hand: a simple integrator. This analysis, although not strictly necessary for the introduced modifications, helps to understand the simplifications introduced in the following chapters, to develop control algorithms for higher-order dynamics.

4.1. Implementation using the finite-set model predictive control algorithm

The first modification for the FS-MPC current control scheme is to transform it into a torque controller with maximum torque per ampere operation and get a control scheme as in Fig. 4.1.

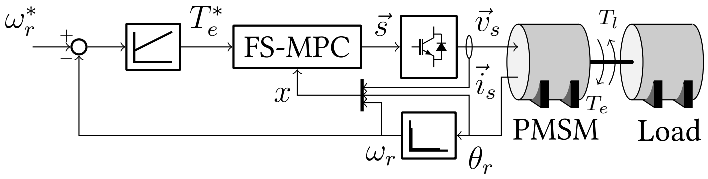

Figure 4.1.: Control scheme: speed control for the permanent magnet synchronous motor with finite-set model predictive torque control considering maximum torque per ampere operation. The feedback x represents the whole state of the motor: $x = [i_\alpha \; i_\beta \; \omega_r \; \theta_r]^T$: although the task of the FS-MPC is only to control the stator currents, the whole state is required to evaluate the system model. Still, only the stator currents are predicted.

In the FOC context this is achieved by taking the MTPA condition into account:

$$i_d + \frac{L_d - L_q}{\psi_m}\left(i_d^2 - i_q^2\right) = 0. \tag{4.1}$$

In [46], for example, this is introduced by adjusting the speed PI controller to give the reference for a magnitude i_n, regarded as the magnitude of the stator current. With this, the references for i_d and i_q are calculated as:

$$i_d^* = -\frac{1}{2}\left(\frac{\psi_m}{2(L_d - L_q)} + \sqrt{\frac{\psi_m^2}{4(L_d - L_q)^2} + 2i_n^{*2}}\right) \tag{4.2}$$

$$i_q^* = \text{sgn}(i_n^*)\sqrt{i_n^{*2} - i_d^{*2}}\,. \tag{4.3}$$

Although, i_n is not exactly proportional to T_e, MTPA operation is achieved and the integral action of the PI controller corrects this discrepancy.

In the FS-MPC scheme, integration of the MTPA curve is straight forward: we adjust the speed controller to give a reference for T_e and then defining:

$$\begin{aligned} e &= g(x) \\ &= \begin{bmatrix} e_{T_e} \\ e_d \end{bmatrix} \\ &= \begin{pmatrix} \frac{3}{2}p(\psi_m i_q + (L_d - L_q)i_d i_q) - T_e^* \\ i_d + \frac{L_d - L_q}{\psi_m}\left(i_d^2 - i_q^2\right) \end{pmatrix}, \end{aligned} \tag{4.4}$$

we can write a cost function:

$$F_c(x) = e_{T_e}^2 + \lambda^2 e_d^2\,, \tag{4.5}$$

With this, the electrical torque developed by the PMSM is controlled directly, ensuring maximum efficiency, with the control algorithm depicted in Fig. 3.4. The scaling factor

λ for the second term is calculated so that the leading terms of $e_{T_e}^2$ and e_d^2, which are proportional to i_q^2 and i_d^2 respectively, weight equivalently in F_c, for equivalent values of i_d and i_q:

$$\lambda = \frac{3p\psi_m}{2} \tag{4.6}$$

Simulations results for the FOC scheme, with the aforementioned modifications, together with simulation results with the torque FS-MPC scheme, are presented in Fig. 4.2.

In the rightmost plot the dashed gray line represents the reference for the electrical torque $T_e^* = 10.235\mathrm{N\,m}$, which is the maximum the motor used for experimental verification can produce with $|\vec{i_s}| \leq 10\mathrm{A}$. In the leftmost plots the dashed and dotted gray lines represent the MTPA curve and the constraint on the magnitude of the currents.

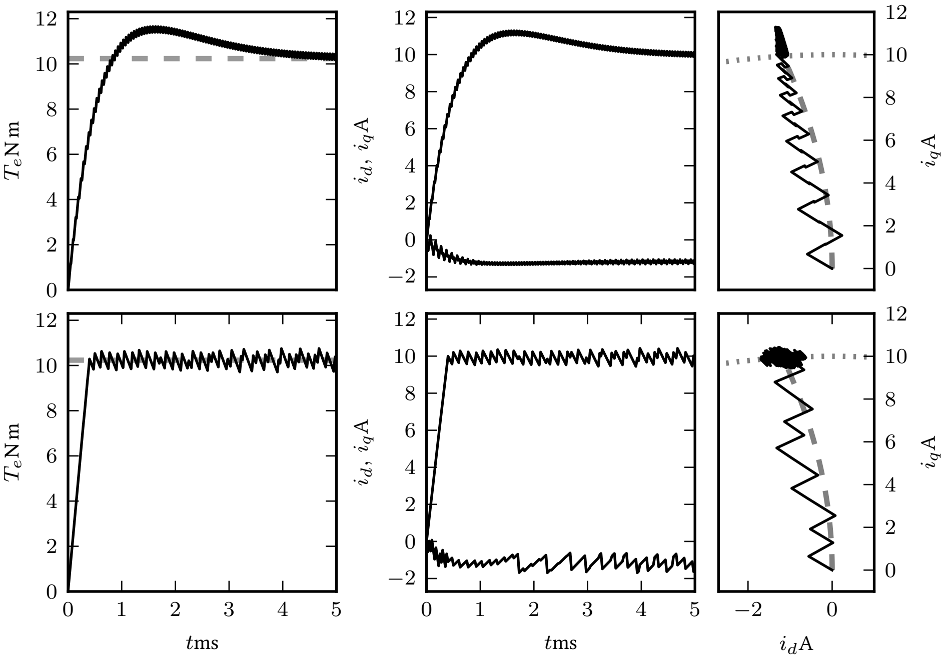

Figure 4.2.: Simulation results: torque control using the FOC (top) and FS-MPC (bottom) schemes, modified to consider MTPA operation. During this test, the rotor speed at $t = 0$s is $\omega_r = 2\pi \cdot 30\mathrm{rad^e}$. The sampling periods used are $h = 50$µs and $h = 25$µs for the FOC and the FS-MPC schemes respectively. The FOC scheme for this test was modified to compensate for the crossed couplings between i_d and i_q and to consider MTPA operation.

Fig. 4.3 presents experimental results using the FS-MPC torque control scheme.

The control system behaves as expected and, in steady state, the torque reference is tracked respecting the current constraint and the MTPA operation.

A noteworthy aspect of the behavior displayed by the FS-MPC algorithm, which is very noticeable in Fig. 4.3, is the irregular switching pattern it generates. Switching occurs

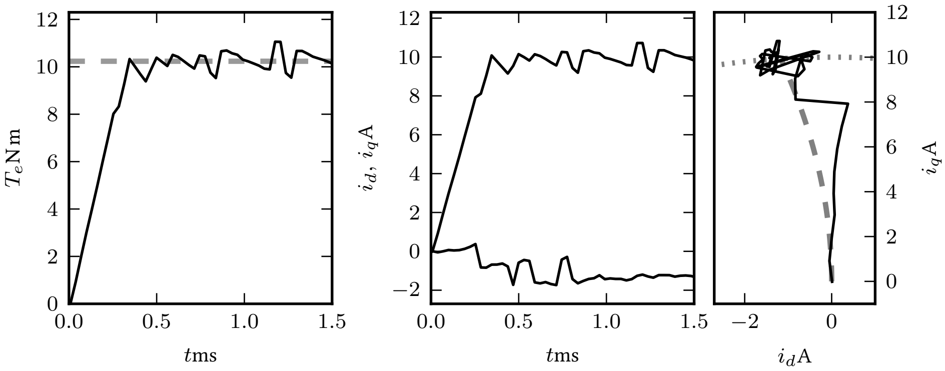

Figure 4.3.: Experimental results: FS-MPC for torque control, considering MTPA operation. During this test, the rotor speed is very near zero and sampling period is $h = 30.725$µs.

when it will result in a decrease in the cost: consider the switch that occurs at $t \approx 0.25$ms, which is very noticeable in the rightmost plot near $i_q \approx 8$A. Before this point, the weight of the torque error dominates the cost. At $t \approx 0.25$ms, the term weighting the distance to the MPTA curve is more important and the controller reacts accordingly. This fact, combined with the constraint that confines actuation switches to the sampling instants, yields to the fact, that switches occur for a quantized subset of the state space: when the state lays near the reference, no switches occur, since that would bring about an increase in the cost. Consequently, the form of the ripple is dependant on the operating point and, more importantly, the stator currents, and consequently all the system states, display a distributed frequency spectrum. In some applications this could be critical, since non-modeled dynamics could be excited, compromising the system performance and stability.

In the following sections, these effects are reduced introducing a modulation scheme based on space vector modulation (see Sec. 2.1.1): note that, in the results presented in Fig. 4.3, if the controller was allowed to use a linear combination of the voltages produced by two switching states, the MTPA curve could be approached from the beginning and, in steady state, the reference could be tracked with arbitrary precision.

4.2. Discrete-time minimum time control for the simple integrator with a continuous actuation set

Consider a simple integrator:

$$\frac{dx}{dt} = \frac{1}{\tau}u \tag{4.7}$$

with $x \in \mathbb{R}$ and $u \in [-\hat{u}, \hat{u}]$. τ is a parameter of the system.

As noted in section 3.3, a continuous-time time-optimal controller for this linear system will take the form of a bang-bang controller. In a regulation problem: u will be either $-\hat{u}$

or $\hat{u}$ depending on the sign of x, or 0 if $x = 0$. With this, we can write this very simple feedback rule for the continuous-time case:

$$\pi(x) = -\mathrm{sgn}(x)\hat{u}\,. \tag{4.8}$$

In this very simple case we can identify $x = 0$ as the switching curve.

Lets consider now the discrete-time version of this problem. A model for the simple integrator in the discrete-time domain is given by:

$$x[n+1] = x[n] + \frac{h}{\tau}u\,, \tag{4.9}$$

where and h is the sampling period. Note that for:

$$x[n] \in \left\{x : |x| < \frac{h}{\tau}\hat{u}\right\}\ , \tag{4.10}$$

the state feedback rule $\pi(\cdot)$ in eq. (4.8) will make the system in eq. (4.9) oscillate (chatter) around $x = 0$. In this very simple case it is easy to understand how, as the authors of [47] remark, time-optimal controllers are not bang-bang in nature over the whole state space for discrete-time systems.

We can correct $\pi(\cdot)$ to eliminate the chattering in the discrete-time case with:

$$\pi(x) = \begin{cases} -\frac{\tau}{h}x & \text{if } |x| \leq \frac{h}{\tau}u \\ -\mathrm{sgn}(x)\hat{u} & \text{if } |x| > \frac{h}{\tau}u \end{cases} \tag{4.11}$$

The new rule in $\pi(\cdot)$ resembles a dead-beat controller [48], a very simple form of a predictive controller. This feedback rule considers a literal inversion of the system dynamics, which in this case is only possible because of the simplicity of the model.

4.3. Implementation using a continuous-set model predictive control algorithm

The model of the PMSM, in particular, can not be inverted as easily, as the discrete-time model for the simple integrator in the last section, mainly due to the non-linearities in its model. If we forget about these non-linearities for a moment and consider that in the FS-MPC scheme the predictions of the state, with $u \in \mathscr{U} = \{v_0, \ldots, v_7\}$ (the voltage vectors generated by the VSI), draw an hexagon in the e_{T_e} , e_d plane (see eq. 4.4 and Fig. 4.4). The vertices of the hexagon correspond to the voltage vectors produced by the switching states the VSI allows (see Fig. 2.1 in section 2.1). The hexagon in the e_{T_e} , e_d plane is, however, rotated by the rotor magnet position and deformed by the system dynamics.

We can assume that any point inside this hexagon can be generated by a linear combination of the predictions and that the same linear combination of the corresponding voltage

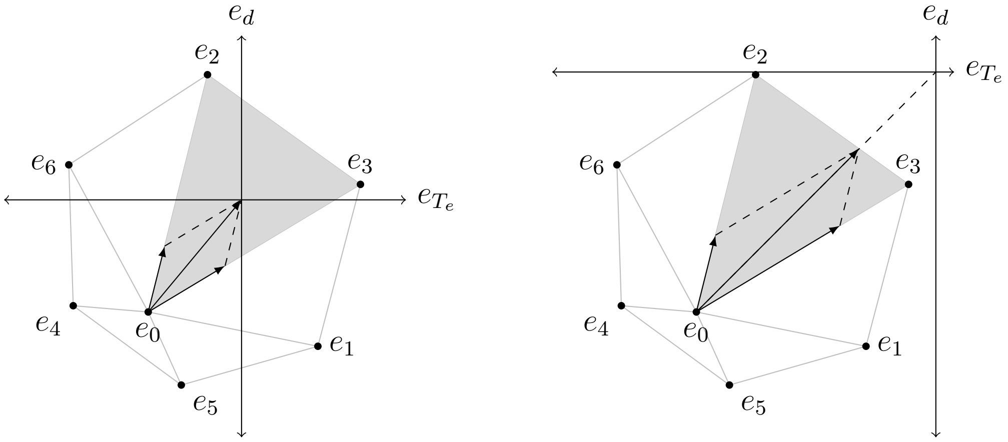

Figure 4.4.: The predictions $e_i = g(x_i[n+1]) = (g \circ f)(x[n], v_i[n])$ (see eq. (4.4)) form an hexagon in the $e_{T_e}\, e_d$ plane. The *target* is the origin of this plane $(e_{T_e}, e_d) = (0,0)$. On the left, the target is achieved by linear combination of predictions e_0, e_2 and e_3. On the right, the target lies outside the hexagon, but it can be approached with a linear combination laying on the intersection between the boundary of the hexagon and the ray between e_0 and the target.

vectors, when applied to the model, will generate this point. The underlying assumption for this, is linearity of the system dynamics during a sampling period. In the case of the drives considered in this work this is not problematic, since the sampling period is small compared to the time constants involved.

The problem of finding the correct linear combination can be solved as in space vector modulation: first we identify the triangle where the point we wish to approximate ($(e_{T_e}, e_d) = (0,0)$) lies, and then we solve a linear algebra problem with the corresponding vectors. The result is an arbitrary voltage vector inside the hexagon described by $\{v_0, \ldots, v_7\}$ in the $\alpha\beta$ plane , which can then be modulated through SVM.

To find the pair of vectors to be combined the following is considered: if $e_a - e_0$ and $e_b - e_0$ are two adjacent vectors, they can be used to approximate the target $-e_0$ if:

$$\begin{aligned} m(e_a, e_b) = ((e_a - e_0) \times (-e_0)) \cdot ((e_b - e_0) \times (-e_0)) &\leq 0 \\ \wedge\, (e_a - e_0) \cdot (-e_0) &> 0 \\ \wedge\, (e_b - e_0) \cdot (-e_0) &> 0\,. \end{aligned} \tag{4.12}$$

Once e_a and e_b are identified, the linear combination can be found solving $d' = [d'_a\ d'_b]^T$ from:

$$[(e_a - e_0)\ (e_b - e_0)]d' = -e_0 \tag{4.13}$$

and the voltage vector to be applied is given by:

$$\vec{v}_s^* = d_a \vec{v}_a + d_b \vec{v}_b \tag{4.14}$$

where $\vec{v}_a$ and $\vec{v}_b$ are the voltage vectors that generated e_a and e_b respectively and:

$$\left[\begin{array}{c} d_a \\ d_b \end{array}\right] = \begin{cases} d' & \text{if } d'_a + d'_b \leq 1 \\ \frac{1}{d'_a + d'_b} d' & \text{if } d'_a + d'_b > 1 \end{cases} . \tag{4.15}$$

The latter accounts for situations where the target lies outside the hexagon.

Fig. 4.5 presents a flow-chart of the algorithm. This is applied in all the drives presented in this work, since they all involve, at their lowest abstraction level, the control of two stator current and stator voltage components.

Fig. 4.6 presents the control scheme for the FS-MPC considering a continuous-actuation set, henceforth referred to as continuous-set model predictive control, applied to control the torque in the PMSM, taking MTPA operation into account.

Fig. 4.7 presents simulation and experimental results using this scheme. The results are in both situations very similar, and the control system behaves as expected. The experimental results do now show the inter-sample current and torque ripple, but their form is very predictable and is given by the SVM technique used.

Fig. 4.8 presents experimental results for the speed control of the PMSM, using the scheme described in this section as inner torque controller. The torque control achieved with this scheme is characterized by a rapid response during the transients, comparable to that produced by FS-MPC, and good quality steady-state control, comparable to that produced by field-oriented control and SVM.

The dynamic for the speed control is very familiar and matches that of an integrator being controlled by a PI controller. In the following part, the schemes presented up to now are further developed to consider second order dynamics to, for example, replace the PI speed controller in 4.6 with a quasi-time-optimal controller.

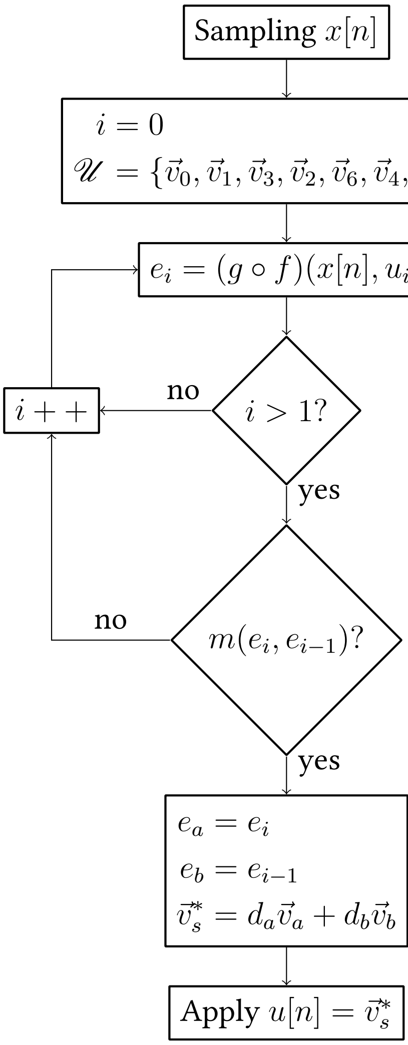
Sampling $x[n]$
$i = 0$
$\mathscr{U} = \{\vec{v}_0, \vec{v}_1, \vec{v}_3, \vec{v}_2, \vec{v}_6, \vec{v}_4, \vec{v}_5, \vec{v}_1\}$
$e_i = (g \circ f)(x[n], u_i)$
$i > 1?$
no
$i++$
yes
$m(e_i, e_{i-1})?$
no
yes
$e_a = e_i$
$e_b = e_{i-1}$
$\vec{v}_s^* = d_a\vec{v}_a + d_b\vec{v}_b$
Apply $u[n] = \vec{v}_s^*$

Figure 4.5.: Flow chart for the finite-set model predictive control algorithm, with continuous-set output. This control algorithm is specified with the *predictive model* $f(\dot{})$, and function $g(\cdot)$, which amounts to a change of variables. The actuation set in this formulation is constituted by the voltage vectors generated by the two-level VSI and $m(\cdot)$ is a boolean function that is true when $e = \vec{0}$ is inside the triangle formed by e_0, e_i and e_{i-1} in the plane of the components of e, in this case, $e_d\, e_{T_e}$.

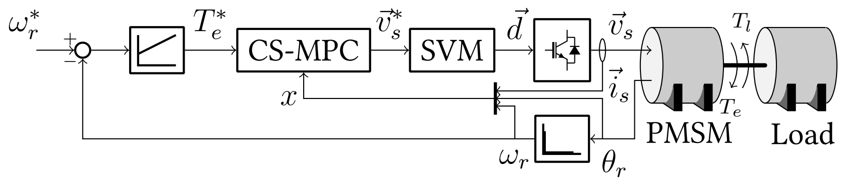

Figure 4.6.: Control scheme: speed control for the permanent magnet synchronous motor with model predictive torque control with continuous actuation set.

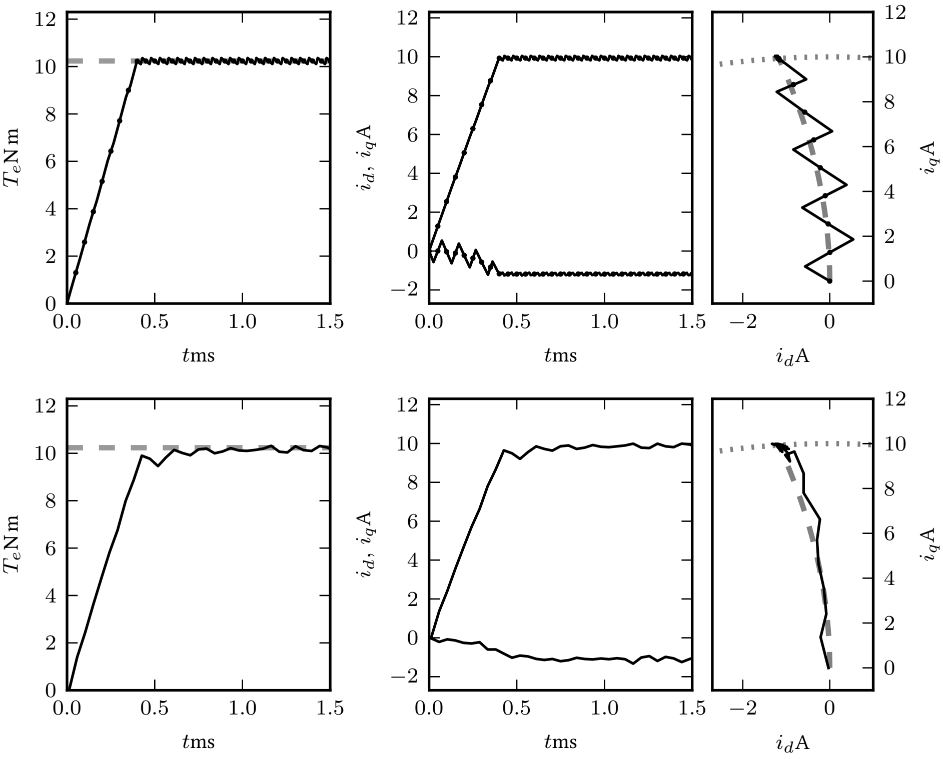

Figure 4.7.: Simulation (top) and experimental (bottom) results for the torque control of the PMSM with CS-MPC scheme and the MTPA operation. For the simulations, the rotor speed and angle at $t = 0$s are $\omega_r = 2\pi \cdot 30$rad^e and $\theta_r = 0$rad^e. The sampling period for the controller is $h = 50$μs. The ripple observed in the current is produced by the space vector modulation: with $\theta_r = 0$rad^e the q axis is aligned with the β axis and the controller alternates between v_3 and v_2 to make the stator current grow along the MTPA curve (dashed gray line). The SVM is implemented using a pulse width modulation scheme with triangular carriers. Sampling occurs at the extreme values of the carrier, so that the current measurements (marked with points) correspond to their mean value. With this, the controller does not *see* the ripple. For the experimental test, the sampling period for the controller is $h = 46.1$μs and at $t = 0$ the rotor is at rest.

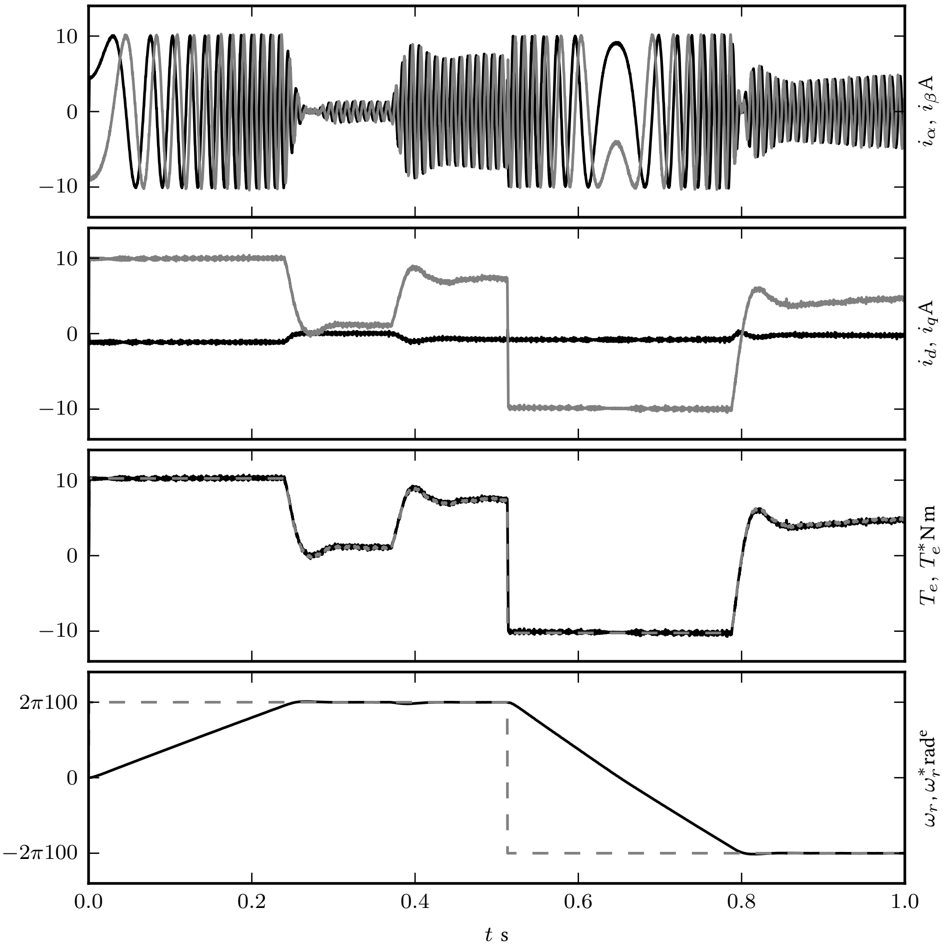

Figure 4.8.: Experimental results: speed control for the PMSM with outer PI speed controller and inner CS-MPC torque controller. The PI speed controller considered anti-windup and was adjusted using pole placement, assuming a plant with the form $G(s) = \frac{1}{J_m s}$, to obtain closed loop pools at $s = \omega_n(-\xi \pm \sqrt{1-\xi^2})$ with $\omega_n = 2\pi \cdot 10$ and $\xi = \frac{1}{\sqrt{2}}$. Slightly before $t = 0.4$s, the motor is loaded with a step shaped load torque.

Part III.

Second Order Systems

5. The double integrator

The control of double integrator is a very well known and extensively studied problem [49]. The interest is justified, despite the apparent simplicity of the problem, because the double integrator model can represent, at least approximately, an immensity of real physical systems (let us think very naively of the very fundamental physical quantities force, speed and position).

The double integrator is used in this work as a very simple approximation to account for the motor dynamics, concerning the speed and flux control in both, the synchronous and the asynchronous motors. The input of the double integrator represents the voltage applied to both the IM and the PMSM. The output of the first integrator represents stator currents and torque. The output of the second integrator represents the flux in the IM and the speed in both motors. These are obviously very rough approximations, but as we will show in the following chapters, they are good enough to account for the desired behavior.

In this chapter we develop a time-optimal control scheme for the double integrator, based in the FS-MPC algorithm presented in section 3.2. The scheme is first developed assuming that the actuation belongs to a finite-set and then it is modified to use a continuous-set.

In the context of optimal control-theory and its application to the double integrator, the methods developed here make little practical sense, we should keep in mind though, that what we actually want is to apply this methods for the control of the synchronous and asynchronous motors.

5.1. Time optimal control with finite actuation set

The double integrator is described by the following dynamics:

$$\frac{d}{dt}\left[\begin{array}{c} x_0 \\ x_1 \end{array}\right] = \left(\begin{array}{c} \frac{1}{\tau_0}u \\ \frac{1}{\tau_1}x_0 \end{array}\right), \tag{5.1}$$

A discrete-time state space model is obtained by using a Taylor series expansion of the solution of this equation. The expansion is truncated conveniently for each component of the state to get direct feed-through from the actuation to both states after one sampling period (see section 2.4 and [35]). The resulting model is given by:

$$x[n+1] = f(x[n], u[n]) \tag{5.2}$$

with

$$f(x,u) = \begin{pmatrix} x_0 + \frac{h}{\tau_0} u \\ x_1 + \frac{h}{\tau_1} x_0 + \frac{h^2}{2\tau_1\tau_0} u \end{pmatrix} \tag{5.3}$$

where h is the sampling period. The actuation u is assumed to belong to the set: $u \in \mathscr{U} = \{-\hat{u},\ 0,\ \hat{u}\}$, and the state x_0 is subjected to the restriction $x_0 \in \mathscr{X} = [-\hat{x}_0, \hat{x}_0]$. This is a representation of the current/torque limit in the motors. The constants τ_0 and τ_1 are the parameters of the system.

We want to synthesize a feedback rule $u = \pi(x)$, that steers the system from a given initial condition to the origin as quickly as possible. This is a very well known and extensively studied problem in optimal control [47, 49]. Here we will adapt these results to use them within the scope of the FS-MPC algorithm for the control of electrical drives. Note that our target, the origin, is fixed: our goal is to develop controllers for relatively slow-changing references.

The time-optimal control problem is solved in this case using the backward induction method (see section 3.3.1), leveraging the fact that both the time and the actuation are quantized.

Fig. 5.1 shows the time optimal trajectories reaching the origin, this is, the trajectories that minimize the time needed for the state to reach the origin, starting from a given initial condition. Note that these trajectories do not cover every point in the state space. This is because both the time and the actuation are quantized; this has as a consequence, that only a subset of the state space can exactly reach the origin. The solution of the optimal control problem calculated using the DPP is actually defined only for this subset.

These trajectories are better understood by studying the step response of the double integrator, given by

$$x_0(t) = \frac{\hat{u}}{\tau_0} t + x_0(0) \tag{5.4}$$

$$x_1(t) = \frac{\hat{u}}{\tau_0\tau_1}\frac{t^2}{2} + x_1(0) \tag{5.5}$$

for

$$u(t) = \hat{u}\mu(t), \tag{5.6}$$

where $\mu(\cdot)$ is the Heaviside step function.

When $u = \pm\hat{u}$ is applied to the system, the state describes parabolas in the state space. Quantization in time and actuation makes that only the points belonging to the parabolas

$$x_1 \pm \tau x_0^2 + \frac{h^2}{\tau_0\tau_1} i = 0 \qquad\qquad i \in \mathbb{Z} \tag{5.7}$$

with $\tau = \frac{\tau_0}{2\tau_1\hat{u}}$, are able to exactly reach the origin. Before reaching it, they must first be lead to the curve

$$x_1 = -\operatorname{sgn}(x_0)\tau x_0^2. \tag{5.8}$$

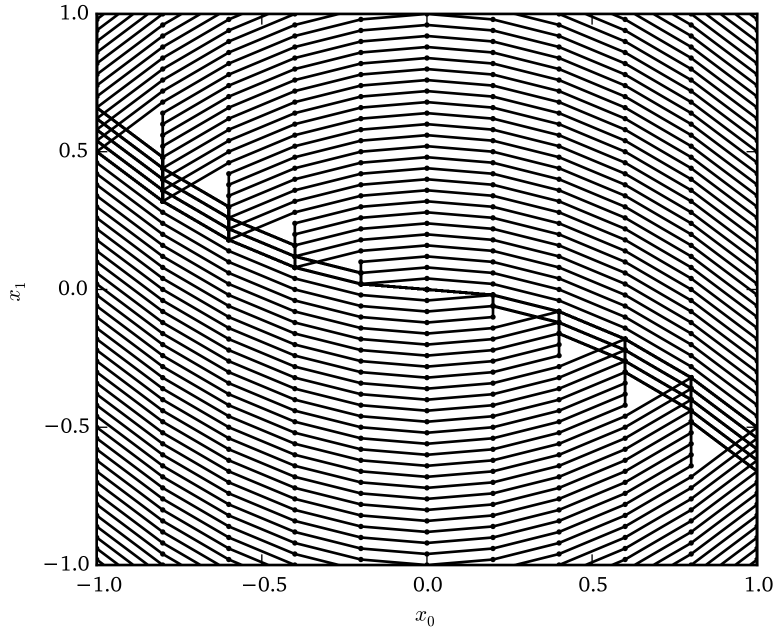

Figure 5.1.: Time optimal trajectories for the double integrator, considering quantized time and actuation. The marked points are the only points in the state space that exactly reach the origin. The trajectories start at one of these points and then develop going first towards the switching curve (see eq. 5.8) and then towards the origin.

When the state reaches this curve, the sign of the actuation changes and the state is driven directly to the origin, following it, thus it is called called the switching curve.

Fig. 5.2 shows the value function (optimal cost or *cost-to-go*), it represents the remaining time before the state reaches the origin, when it follows an optimal trajectory. It is important to note how the cost grows for points away from the origin and the switching curve. This is the key to understand the behaviour that a time-optimal controller should generate.

Fig. 5.3 shows the optimal actuation. This is a very familiar plot in the context of minimum-time control for the double integrator. The classical solution for this problem is a *bang-bang* controller, which applies maximum, minimum and actuation ($\hat{u}$, $-\hat{u}$ and 0), depending on the position of the state, relative to the switching curve. The solution in this case looks slightly different, because of the quantized time: at points near the switching curve it is necessary to apply $u = 0$ so that the state is able to reach it without crossing it, which would result in overshoot, and suboptimal behaviour. Looking at this plot it is clear how the switching curve separates the state space in different regions.

The behaviour generated by the optimal feedback can easily be put in simple terms by observing this plots. The controller applies the minimum (black) or maximum (white) actuation when the error in the state x_1 is significant. The sign is the inverse of the sign

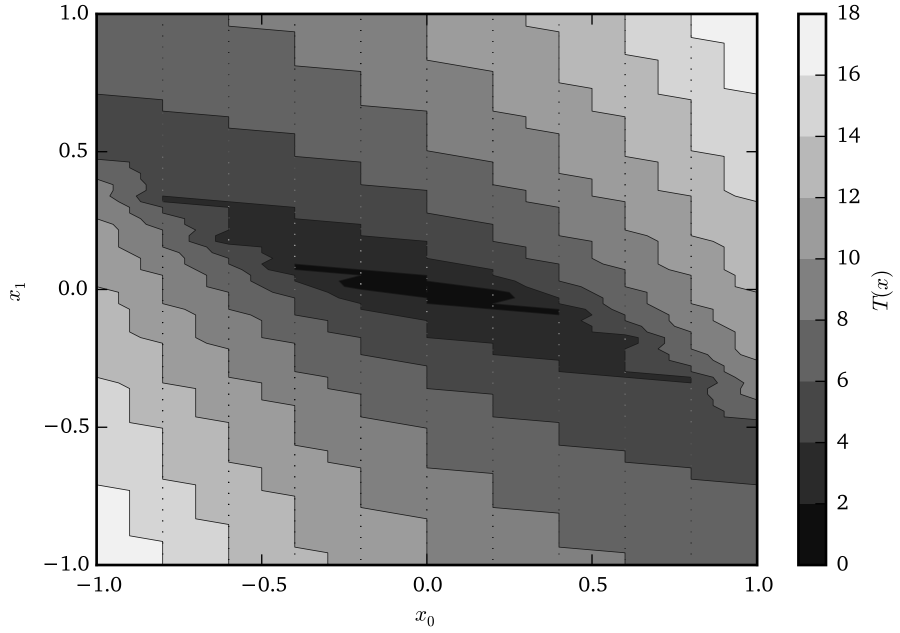

Figure 5.2.: Optimal value function for the time-optimal-control of the double integrator. At each point, understood as the initial condition, the optimal value function represents the minimum time required to steer the state from its initial condition to the origin.

of the error. This behaviour goes on until the state x_0 has reached its maximum value, here the controller applies $u = 0$ (grey) so x_0 stays at its maximum value. Then, when the state is near the switching curve, the controller applies $u = 0$ so the state reaches one point on the switching curve. From then on, the sign of the actuation is the same as the error, so the state trajectory continues to develop on the switching curve until it reaches the origin.

Using these results for controlling electrical drives is not a trivial task, mainly because of the nature of the motor model: the actuation (stator voltage) has two components and how they contribute to build up electrical torque, or magnetizing current, depends on the rotor flux position. The situation is even more complex, when we consider maximum torque per ampere operation and further effects in the stator windings (resistance, back-emf and mutual inductances). The approach proposed in this work consists in using the FS-MPC algorithm: state predictions are mapped into points in the $x_0\,x_1$ plane and the value function, depicted in Fig. 5.2, is used as cost function. In each step the controller will search for the actuation that minimizes this function, so it will naturally follow the trajectories depicted in Fig. 5.1, since for each point in the state space, the optimal trajectory to follow is given by point with the lowest *cost-to-go* out of the points the model can reach.

Still, some issues remain: the *cost-to-go* for the double integrator was calculated and is

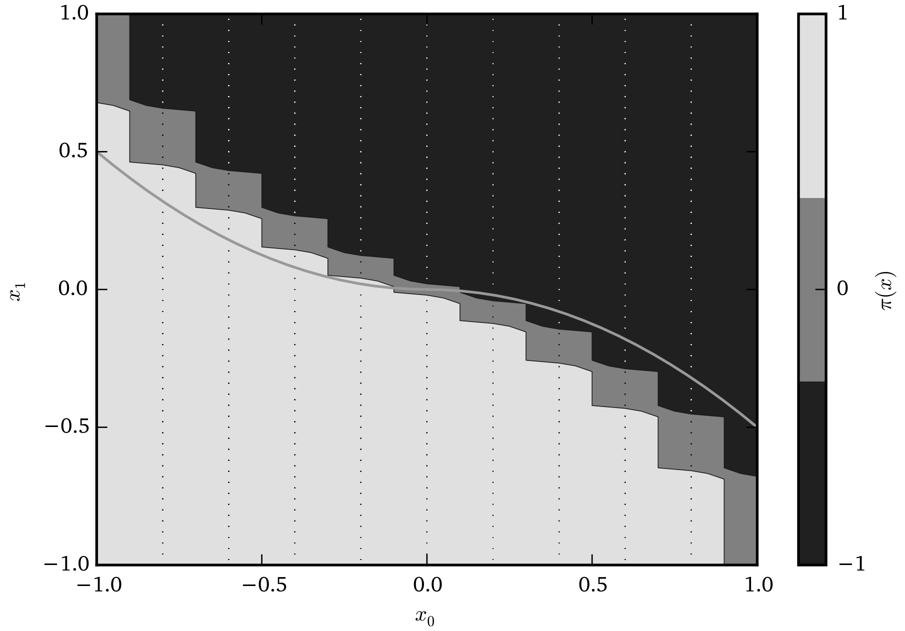

Figure 5.3.: Time-optimal actuation for the double integrator, calculated using the backward induction method. The actuation is only defined for the represented points. Crossing the plot, in gray, is the switching curve in eq. (5.8). Note that the change in the actuation does not exactly match this curve. If it did, the state would not exactly reach it, but cross it, producing overshoot.

defined only for a discrete set of points in the state space, calculation for a more complex system would require harder computations and complete recalculation for different sets of parameters. However, in the FS-MPC algorithm the state predictions are carried out using the whole system model, which includes all its parameters and effects, therefore, the exact *cost-to-go* is not necessary, but only a rough approximation encompassing its main features: location of the switching curve and direction towards it. In this sense the following approximation is introduced:

$$F_c(x) = \left|\tilde{x}_1 + \operatorname{sgn}(x_0)\tau x_0^2\right| \tag{5.9}$$

with

$$\tilde{x}_1 = \begin{cases} x_1 & \text{if } |x_1| \leq \tau\hat{x}_0^2 \\ \tau\hat{x}_0^2 & \text{if } |x_1| > \tau\hat{x}_0^2 \end{cases} \tag{5.10}$$

Function $F_c(\cdot)$ in (5.9), depicted in Fig. 5.4, is zero along the switching curve and grows as the state gets away from it, when the error in x_1 is smaller than the point where the switching curve intercepts the lines describing the constraint for x_0. Otherwise, if the error of x_1 is bigger than this value, the function grows for points away from the limits set for x_0. The latter effect is achieved by introducing $\tilde{x}_1$.

By using this $F_c(\cdot)$ in a FS-MPC scheme with a prediction horizon of one step, exactly the same behaviour as Fig 5.1 is achieved.

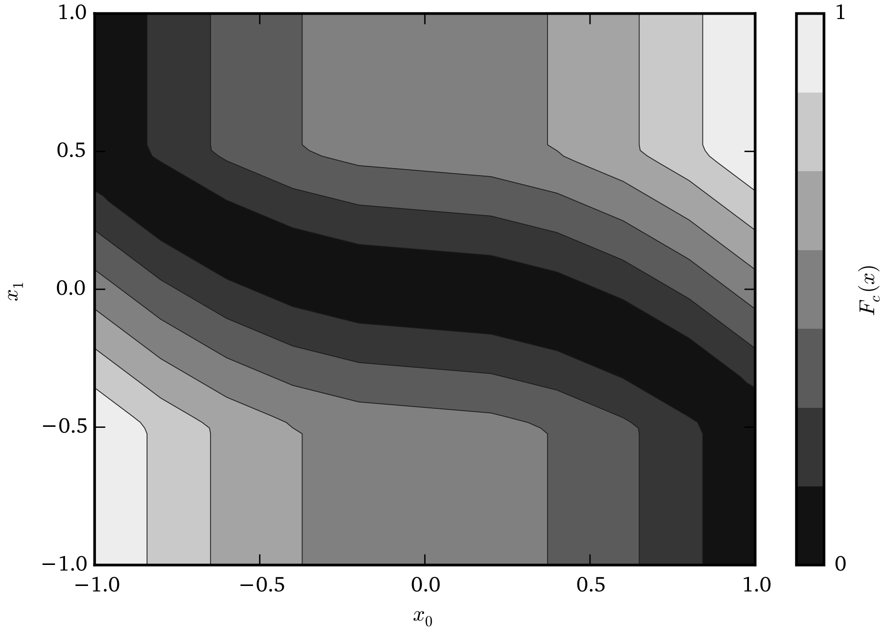

Figure 5.4.: Approximation for the value function: $F_c(x)$.

In drives applications, the electrical torque should usually compensate for a load torque, this can be included in the double integrator model with:

$$\frac{dx_1}{dt} = \frac{1}{\tau_1}\left(x_0 - d\right) \,, \tag{5.11}$$

where the disturbance d represents the load torque. It is also a common requirement for the controller to be able to track references. We can include this in the described scheme, offsetting the state by its steady state value:

$$e = \left[\begin{array}{c} e_0 \\ e_1 \end{array}\right] = \left[\begin{array}{c} x_0 - d \\ x_1 - x_1^* \end{array}\right] \tag{5.12}$$

and fixing the operation limits accordingly:

$$\hat{e}_1 = \tau(\hat{x}_0 - d)^2 \tag{5.13}$$

$$\check{e}_1 = -\tau(\hat{x}_0 + d)^2 \tag{5.14}$$

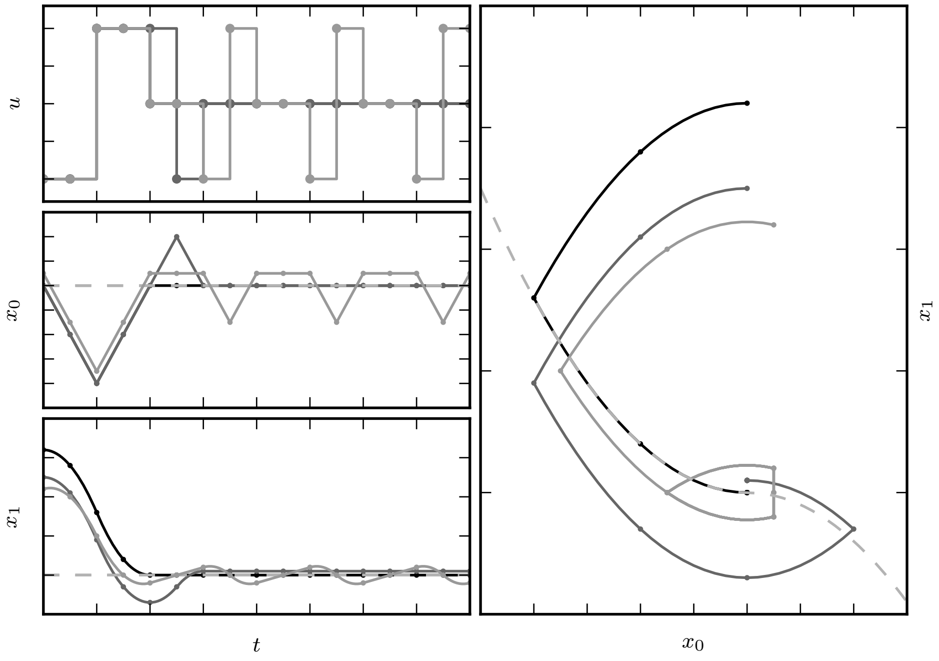

Figure 5.5.: Simulation results for quasi-time-optimal control of the double integrator assuming finite-set actuation. Three different initial conditions are considered. The simulation plotted is black exactly reaches the switching curve (dashed gray line) and is then driven towards the origin. The other two simulations in dark and light gray, can not reach the switching curve and present steady state error or chattering. The dot marks in each plot represent the sampling instants.

where $\hat{e}_1$ and $\check{e}_1$ are the upper and lower values for e_1, where x_0 reaches its limit. With this, the cost function can be then evaluated as $F_c(e)$:

$$F_c(e) = \left|\tilde{e}_1 + \text{sgn}(e_0)\tau e_0^2\right| , \tag{5.15}$$

with

$$\tilde{e}_1 = \begin{cases} \hat{e}_1 & \text{if } \hat{e}_1 < e_1 \\ e_1 & \text{if } \check{e}_1 \le e_1 \le \hat{e}_1 \\ \check{e}_1 & \text{if } \qquad e_1 < \check{e}_1 \end{cases} . \tag{5.16}$$

Figure 5.5 presents simulation results using this scheme. In these plots the problem of chattering and steady state error are evident. Their origin lays on the fact that both time and actuation u are quantized [47, 50]. Extensive simulations were run to quantify the magnitude of the error and the following boundaries were found:

$$\Delta_{x_1} \leq \frac{h}{\tau_0}\hat{u} \tag{5.17}$$

$$\Delta_{x_2} \leq \frac{h^2}{\tau_0 \tau_1}\hat{u} \tag{5.18}$$

If such errors of this magnitude are unacceptable in a given application, the situation can be addressed allowing the use of a continuous set for the actuation. Such an approach is presented in the next section.

5.2. Quasi-time-optimal control with continuous actuation set

Under certain circumstances it might be desirable to address the issues mentioned earlier: steady state error and chattering, caused by the assumption of a finite actuation set, for example, when the error introduced by them (see eq. 5.18) is significantly bigger than the precision of the state measurements or estimations.

Two different approaches are proposed for the vicinities of the switching curve and of the steady state. These situations are depicted if Fig. 5.6.

5.2.1. Vicinity of the switching curve

For this case, we start by acknowledging that, for a fixed sampling period h and an initial state $x[n]$, the predictions $x[n+1]$ with $u \in [-\hat{u}, \hat{u}]$ are mapped to the ray between $f(x[n], -\hat{u})$ and $f(x[n], \hat{u})$ in the $x_0\, x_1$ plane, described by:

$$x_1 = \frac{h}{2\tau_1}x_0 + \left(x_1[n] + \frac{h}{2\tau_1}x_0[n]\right) \quad \text{with} \quad x_0 \in \left[x_0[n] - \frac{h}{\tau_0}\hat{u},\, x_0[n] + \frac{h}{\tau_0}\hat{u}\right] \tag{5.19}$$

This is obtained manipulating (5.3). The ray is depicted in Fig. 5.6 with dotted lines.

When the $x[n]$ lies in the vicinity of the switching curve, the value of the actuation that leads the state exactly to the switching curve can be found by solving the point where the ray and the switching curve (eq. (5.8)) intersect. We are only interested in its x_0 component, given by:

$$x_0^* = \begin{cases} \frac{h\hat{u}}{2\tau_0}\left(1 - \sqrt{1 + \frac{8\tau_0\tau_1}{h^2\hat{u}}\left(x_1[n] + \frac{h}{2\tau_1}x_0[n]\right)}\right) & \text{if } x_1[n] + \frac{h}{2\tau_1}x_0[n] \geq 0 \\ \frac{h\hat{u}}{2\tau_0}\left(-1 + \sqrt{1 - \frac{8\tau_0\tau_1}{h^2\hat{u}}\left(x_1[n] + \frac{h}{2\tau_1}x_0[n]\right)}\right) & \text{if } x_1[n] + \frac{h}{2\tau_1}x_0[n] < 0 \end{cases} \tag{5.20}$$

we can now find u^* replacing $x_0[k+1]$ with x_0^* in (5.3). For the drive controllers presented in the next chapters we feed x_0^* to an internal control loop for T_e (x_0) that considers a continuous actuation set, such as the one introduced in 4 for the torque control of the PMSM.

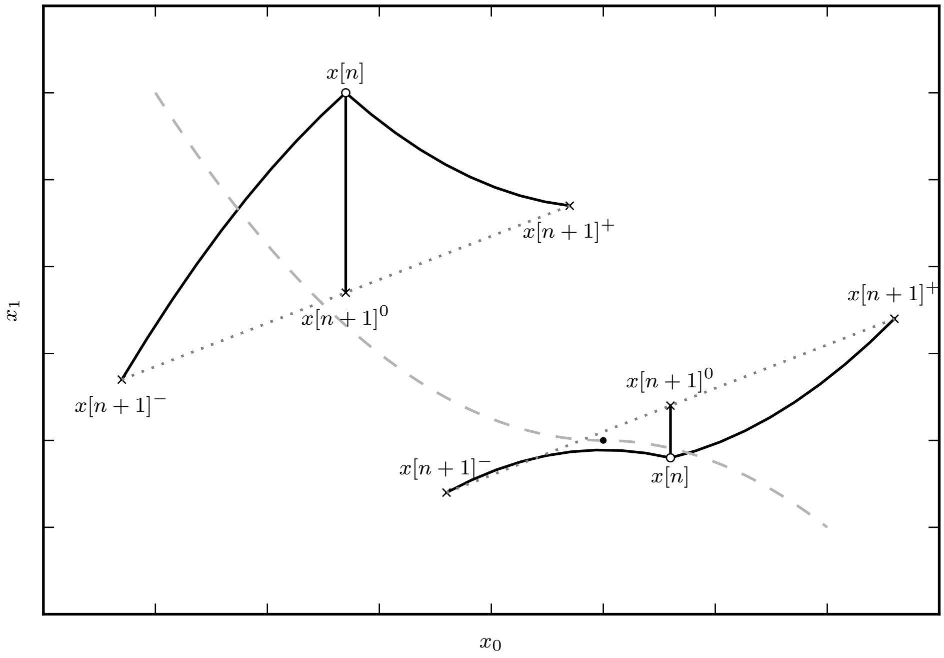

Figure 5.6.: In this plot the points $x[n]$ are the initial conditions and $x[n+1]$ the predicted state with $u = -\hat{u}$, 0 and $+\hat{u}$ respectively. The dashed gray line is the switching curve $x_1 = -\mathrm{sgn}(x_0)\tau x_0^2$ and the dotted lines represent all the points that can be reach with $u = [-\hat{u}, \hat{u}]$. For most of the state space $u \in \{-\hat{u}, 0, \hat{u}\}$ will not lead the state to the origin, nor to points belonging to the switching curve. Note that, when $x[n]$ lays near the target (marked with a black dot), the state will not reach it with any value in $[-\hat{u}, \hat{u}]$. In this situation the aim should not be to reach the switching curve and a smoother controller should take over.

To effectively implement this, first x_0^* is calculated regardless of the position of initial condition and then u^* is calculated from x_0^{**} or, for the drive controllers, x_0^{**} is feed as reference for x_0:

$$x_0^{**} = \begin{cases} x_0[n] + \frac{h}{\tau_0}\hat{u} & \text{if } x_0[n] + \frac{h}{\tau_0}\hat{u} < x_0^* \\ x_0^* & \text{if } |x_0^* - x_0[n]| \leq \frac{h}{\tau_0}\hat{u} \\ x_0[n] - \frac{h}{\tau_0}\hat{u} & \text{if } x_0^* < x_0[n] - \frac{h}{\tau_0}\hat{u} \end{cases}, \tag{5.21}$$

5.2.2. Vicinity of the steady state

In the vicinity of the origin, the later approach would make the state oscillate around it (see Fig. 5.6). A natural solution for this case is use of a linear feedback.

If we assume that near the steady state u can be manipulated to obtain:

$$x_0[n+1] = x_0[n] + \frac{h}{\tau_0}u \tag{5.22}$$

$$= x_0^*[n] \tag{5.23}$$

$$= -k\frac{2\tau_1}{h}\left(x_1[n] - x_1^*[n]\right) , \tag{5.24}$$

where k is a tunning parameter and x_0^* and x_1^* are reference values for the state, we can characterize the closed loop response with the transfer function:

$$\frac{X_1(z)}{X_1^*(z)} = \frac{kz^{-1}(1+z^{-1})}{1+(k-1)z^{-1}+kz^{-2}} \tag{5.25}$$

and use k to fix the dynamics. In the following examples $k = 0.24498$ is used to fix the damping ratio $\xi = \frac{1}{\sqrt{2}}$.

This strategy is imposed when state lays inside the region:

$$|x_0 - d| \leq \frac{h}{\tau_0}\hat{u} \tag{5.26}$$

$$|x_1 - x_1^*| \leq \frac{h^2}{\tau_0\tau_1}\hat{u}\,. \tag{5.27}$$

These limits were identified before as upper bounds for the steady state error, when using a finite actuation set, through extensive simulations, i.e. empirically. In this sense, it would be convenient to have tools to identify this region unequivocally, to restrict its size, to match the behaviour generated by both strategies in the boundary and to guarantee that the controller inside this region will not make the state leave it, which would generate chattering around the boundary.

Figures 5.7 and 5.8 present a plot of the resulting actuation and simulations using this scheme.

What this scheme effectively does is calculating the actuation (or a reference for the state being directly affected by it, in this case x_0) to generate approximately time optimal behaviour, when the state is away from its target or the switching curve and a smooth behaviour, when the state is close to them. In the following chapters this scheme will be referred to as *smoothened quasi-time-optimal control* (SQTOC).

In the following chapters, these schemes are applied for the control of electrical drives, characterized by second order dynamics.

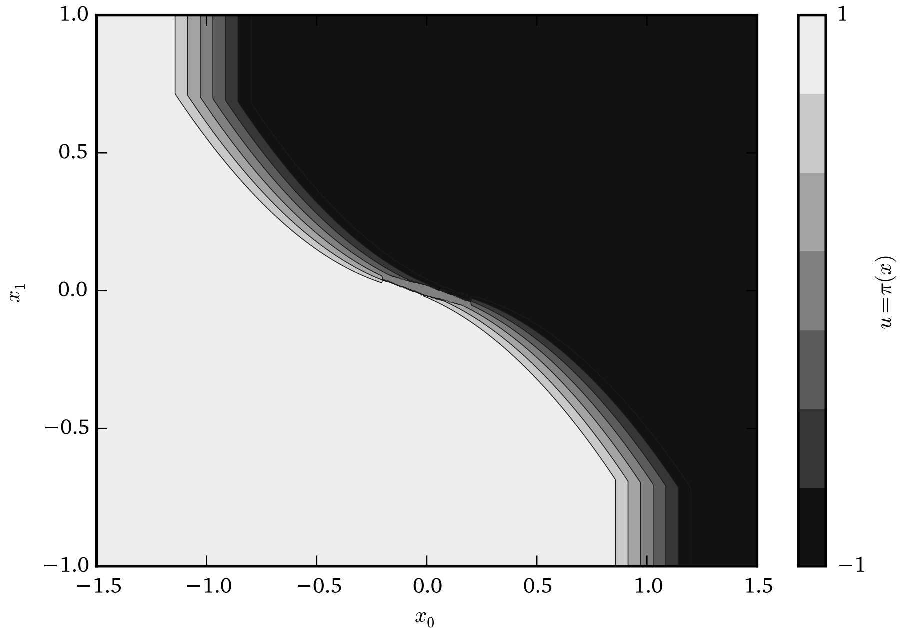

Figure 5.7.: Optimal actuation for the double integrator, with the smoothening scheme. When the initial state is near the switching curve, u adopts exactly the value that would steer the state towards it. Around the steady state the linear feedback controller (5.24) takes over.

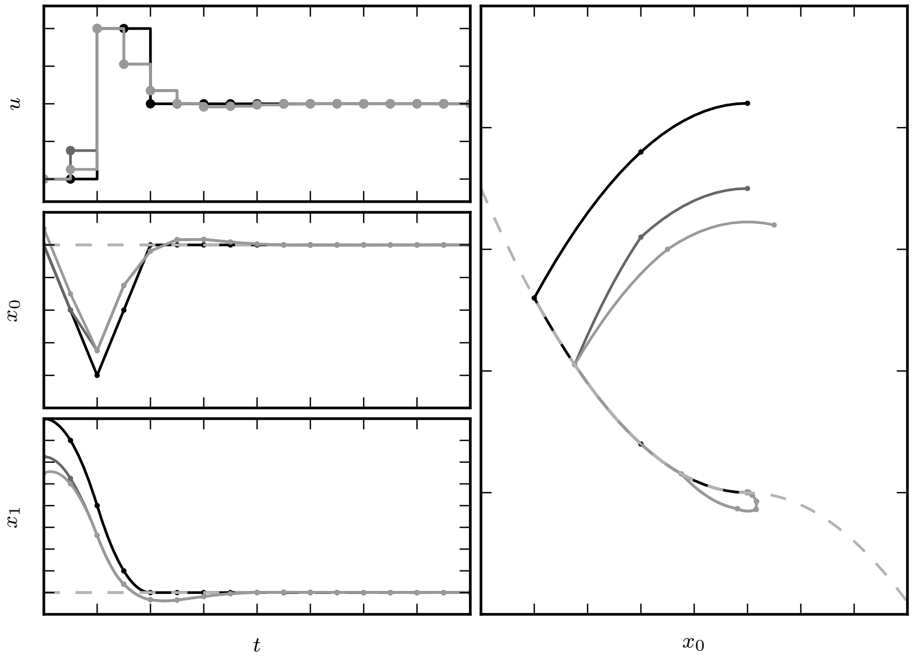

Figure 5.8.: Simulation results for the double integrator using the proposed CS-MPC controller. The initial conditions are the same as in Fig. 5.5, but no chattering or steady state error is observable. The simulation plotted in black reaches the origin in minimum state, as if it did in the previews simulation with FS-MPC scheme.

6. Torque control for the induction motor

The main idea put forward by the field oriented control scheme (see Fig. 3.2) is that the problem of controlling the torque produced by the induction motor can be divided in two subproblems, by means of a transformation that separates the stator currents in two components, in terms of the convention adopted in section 3.1: one related to the rotor flux magnitude, and the second to the electrical torque itself. All these variables are bound by causal relationships: the stator current induces the rotor flux and the electrical torque is proportional to the cross product the flux and currents, however, the time constants characterizing their dynamics are different: the rotor flux being much slower. This suits perfectly to order these variables hierarchically and use a cascaded structure to control them. This is an appropriate and simple solution, but it imposes restrictions on the achievable closed loop dynamics. A way to push them towards their physical limits is to control the whole system dynamics in a centralized fashion.

One approach is to attempt to introduce a finite-horizon model predictive controller for the relevant dynamics, using the well known setting: discrete-time state space model and quadratic cost function, weighting the distance between the actual state and its desired steady state or target. Not very slowly we come to the realisation that first, some kind of scaling will be necessary (input voltage, flux and currents have different dimensions) and second, that in the presence of constraints in the actuation and the states (currents have a top boundary) and given the causal relationships between the states, the optimization should be carried out over a relatively long prediction horizon: long enough for the accumulated error in the flux, for example, to surpass the error in the stator currents, which must be different from zero at some point, if the flux error ought to be minimized.

In this chapter we develop a centralized controller for the flux and torque in the induction motor, based on the framework given by FOC and the model predictive control methods introduced before. The goals are to track a step reference for the flux and an arbitrary reference for the torque and to push the closed loop dynamics towards their physical limits, taking the constraints for the magnitude of the stator currents into account. This is achieved adapting the controllers introduced in the later chapter for the double integrator, to be used in the context of the motor. The controller is formulated to be implemented with both the finite-set and the continuous-set algorithms, depicted in figures 3.4 and 4.5. In both cases a discrete-time version of the state space representation for the IM in eq. (2.56) is used as the *predictive model*. The *cost function* $F_c(\cdot)$ for the finite-set algorithm and the change of variables $g(\cdot)$ for the continuous-set algorithm are defined in terms of *flux oriented* magnitudes: the closed loop dynamics for the flux are fixed using the double integrator as an approximation for the dynamics between v_d, i_d (stator voltage and current in the direction of the flux) and the rotor flux $\vec{\psi}_r$, the electrical torque is assumed to be proportional to the i_q current component.

6.1. Rotor flux dynamics approximation using the double integrator

The dynamic behavior of the rotor flux is given by equations (2.51) and (2.52):

$$\frac{d}{dt}\begin{bmatrix} i_d \\ \psi_d \end{bmatrix} = \begin{pmatrix} \frac{1}{\sigma L_s}v_d - \frac{1}{\tau_\sigma}i_d + \frac{k_r}{\tau_r \sigma L_s}\psi_d + \omega_k i_q \\ \frac{L_m}{\tau_r}i_d - \frac{1}{\tau_r}\psi_d \end{pmatrix} \tag{6.1}$$

and we wish to fix the dynamics of the closed loop using an approximation of these equations, given by the double integrator

$$\frac{d}{dt}\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{pmatrix} \frac{1}{\tau_0}u \\ \frac{1}{\tau_1}x_0 \end{pmatrix}, \tag{6.2}$$

with $u \in [-\hat{u}, \hat{u}]$ for the continuous-set actuation algorithm and $u \in \{-\hat{u}, 0, \hat{u}\}$ for the finite-set actuation algorithm.

Fixing the dynamics in this context means that the controller will steer i_d to go towards its extreme value to correct the error in ψ_d, so that both follow the switching curve

$$x_1 = -\text{sgn}(x_0)\tau x_0^2, \tag{6.3}$$

with $\tau = \frac{\tau_0}{2\tau_1 \hat{u}}$, towards the target.

At a first glimpse, the use of such an approximation seems too rough: the real system dynamics pay little resemblance to the dynamics of the double integrator. First of all, the bounds for v_d are not constant, since v_d is generated by the two-level VSI, their real value depend on the value of the dc-link and the angle of the rotor flux with respect to the stator windings θ_k. Moreover, even if we assume the term $\omega_k i_q$ to be constant over the prediction horizon, it effectively changes the ability of the controller to manipulate i_d. Fig. 6.1 depicts these issues: the plots represent the actual boundaries for $u = v_d$ depending on θ_k and the term $\omega_k i_q$. Despite these issues, and in order to keep the control algorithm simple, the boundaries for u are assumed to be constant. Note that the decision for the value of $\hat{u}$ has an impact on the closed loop dynamics, which is analogue to the effect of the gain in a classical proportional-integral controller: a low value will make the controller fix more conservative dynamics, i.e., the control will be slower and the state will chatter around the switching curve, as the value for $\hat{u}$ is set higher, the controller will take the system closer towards its physical limits and less chatter around the switching curve will occur If the value of $\hat{u}$ is much higher than the real boundaries for u, the desired

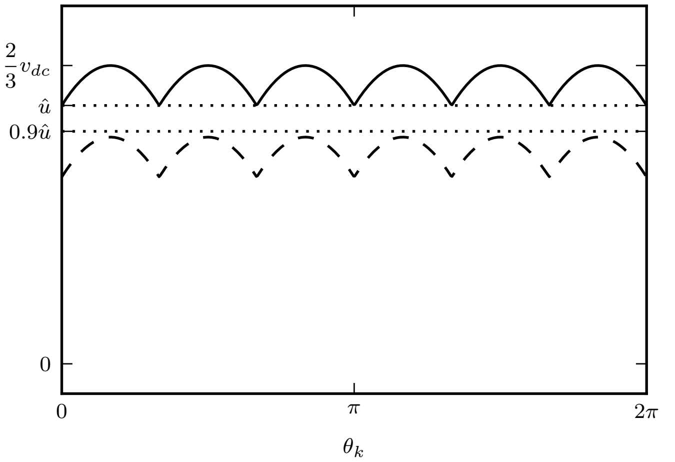

Figure 6.1.: Actuation upper boundaries using the two-level VSI to generate the stator voltage. The solid black line represents the maximum voltage the VSI can synthesize in the d direction, as a function of the flux angle ω_k. The displaced boundary, plotted with a dashed line, represents the maximum voltage the VSI can synthesize with the motor operating at nominal speed $\omega_r = 2\pi 50$ with $T_e = 10\mathrm{N\,m}$. In this situation there is a 24% error between the maximum voltage available at zero speed and load. The dotted lines represent the value assumed for $\hat{u}$ in eq. (6.4) and the value used in the experimental test that make use of space vector modulation (see eq. (6.5)).

dynamics are faster than the system can deliver. In this case overshoot and oscillations around the target will occur. For the rest of this work, the boundary for the actuation is assumed to be

$$\hat{u} = \frac{\sqrt{3}}{3} v_{dc}\,, \tag{6.4}$$

which is the radius of the circle inscribed in the hexagon generated by the voltage vectors the two-level VSI can produce, in the $\alpha\beta$ plane (see Fig. 2.1). This is also the maximum amplitude the two-level VSI can achieve, when synthesizing perfectly sinusoidal output voltages.

For the schemes developed in the following chapters and sections, which use space vector modulation, the boundary for u is assumed to take the value:

$$\hat{u} = 0.9 \cdot \frac{\sqrt{3}}{3} v_{dc}\,. \tag{6.5}$$

This, because of limitations on the test bench used for experimental validation, which only allows measurements at the sampling instants, at the peaks of the triangular carriers of the pulse width modulation: switching introduces noise in the measurements and they occur very near or at the sampling instant, when modulating relatively high voltages.

The controllers developed in the following sections use the whole system model as predictive model, this means that the controllers will have the information required to compensate the effect of the term $\omega_k i_q$, so that i_d and ψ_d follow the fixed dynamics. This also means that the effects relevant in steady state: stator and rotor resistance, will also be compensated setting an appropriate target

$$\begin{bmatrix} i_d \\ \psi_d \end{bmatrix} = \begin{bmatrix} \frac{1}{L_m}\psi_d^* \\ \psi_d^* \end{bmatrix}, \tag{6.6}$$

which will represent the *origin* in the state space of the double integrator.

In this sense, we are interested in the approximation, only insofar as it can model the system dynamics during the transients, which we can characterize using the step response of the system, which assuming $\omega_k i_q = 0$, neglecting the effect of the stator resistance and applying a step voltage v_d of magnitude $\hat{u}$ in eq. (6.1) at $t = 0$, is given by:

$$i_d(t) = \frac{\hat{u}}{\sigma L_s}t + \frac{1}{\sigma^2 L_s^2}\frac{L_m}{\tau_r}\hat{u}\frac{1}{p_1 - p_0}\left(\frac{1}{p_0^2}\left(p_0 - 1 + e^{-p_0 t}\right) - \frac{1}{p_1^2}\left(p_0 - 1 + e^{-p_1 t}\right)\right) \tag{6.7}$$

$$\psi_d(t) = \frac{1}{\sigma L_s}\frac{L_m}{\tau_r}\hat{u}\frac{1}{p_1 - p_0}\left(\frac{1}{p_0}\left(1 - e^{-p_0 t}\right) - \frac{1}{p_1}\left(1 - e^{-p_1 t}\right)\right) \tag{6.8}$$

with

$$p_0,\, p_1 = -\frac{1}{2}\left(\frac{1}{\tau_r} \mp \sqrt{\frac{1}{\tau_r^2} + 4\frac{L_m^2}{\sigma L_r L_s \tau_r^2}}\right). \tag{6.9}$$

The leading terms of the Taylor series expansion of (6.7) and (6.8):

$$i_d(t) = \frac{1}{\sigma L_s}\hat{u}t + \mathcal{O}(t^3) \tag{6.10}$$

$$\psi_d(t) = \frac{1}{\sigma L_s}\frac{L_m}{\tau_r}\frac{\hat{u}t^2}{2} + \mathcal{O}(t^3) \tag{6.11}$$

match the step response of the double integrator:

$$x_0(t) = \frac{1}{\tau_0}\hat{u}t \tag{6.12}$$

$$x_1(t) = \frac{1}{\tau_0 \tau_1}\frac{\hat{u}t^2}{2} \tag{6.13}$$

In this sense, using the double integrator to fix the closed loop dynamics is, in this case, a valid first approximation for the transients. From these step responses we can identify:

$$\begin{aligned} u &= v_d & & \\ x_0 &= i_d & \tau_0 &= \sigma L_s \\ x_1 &= \psi_d & \tau_1 &= \frac{\tau_r}{L_m}. \end{aligned} \tag{6.14}$$

Note that in both equations, (6.7) and (6.8), the initial conditions (not included in the analysis for the sake of simplicity) decay. This does not change the dynamics of the system with a proper translation of the origin (see eq. (6.6)), since the dynamics defined in (6.1) are linear neglecting the effect of $\omega_k i_q = 0$.

The degree of influence of the higher order terms is a matter of scale: the longer the time being accounted for, the bigger it becomes. What a long time is, is defined by the motor parameters.

Figure 6.2 presents simulations of the step response of the double integrator and the system dynamics using the real parameters of the motor used in experimental tests. These simulations were carried out with the time running backwards, this is

$$u(t) = \hat{u}\mu(-t) \tag{6.15}$$

is applied to the system. With this, the simulated step response of the double integrator matches the switching curve described in the last chapter.

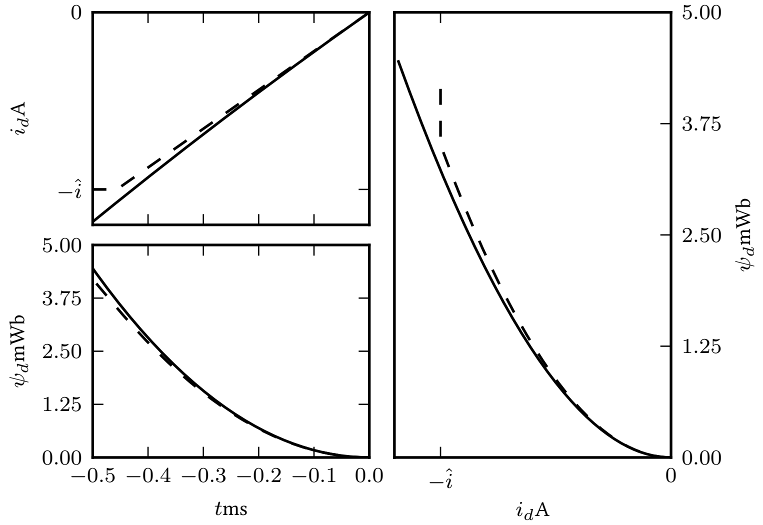

Figure 6.2.: Step response, with time running backwards, of the double integrator (dashed black line) and the *real* rotor flux dynamics, described in eq. (6.1), with $\omega_k i_q = 0$ (solid black line). In the simulation of the double integrator response, $u = 0$ is applied once i_d reaches its constraint $\hat{i} = 10$A.

As expected, the error is very small, when t is still near zero, an tends to null at $t = 0$. Note that the error is not dramatically big when x_0 reaches 10A, which is the constraint for the stator current magnitude, and that the real system is faster than the double integrator when it comes to *get away* from the origin. Consequently, using the double integrator model to fix the dynamics will always lead to conservative closed loop dynamics: slightly slower than the limits put by the real model. This means that, when the system parameters are perfectly known, the approximation will not cause overshoot nor instability.

6.2. Torque control for the induction motor with finite actuation set

In the last section we saw how the dynamics of the rotor flux can be approximated by the dynamics of a double integrator. In this section we use the methods developed in chapter 5 for controlling the double integrator, to achieve quasi time-optimal control of the rotor flux and torque in the induction motor.

A scheme for the control system is presented in Fig. 6.3.

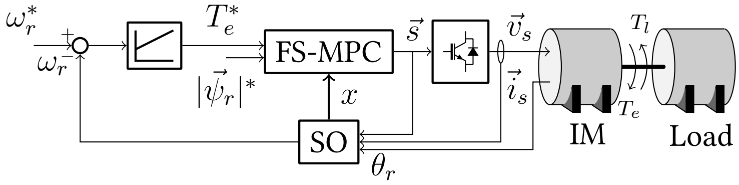

Figure 6.3.: Control scheme: speed control for the induction motor with centralized finite-set model predictive torque and flux control. Block SO represents a state observer and feedback x represents the whole state of the motor: $x = [i_\alpha \; i_\beta \; \psi_{r\alpha} \; \psi_{r\beta} \; \omega_r]^T$. In this scheme the state observer needs only to estimate the rotor speed ω_r and flux $\vec{\psi}_r$. The rotor speed is estimated using a simple derivation of the rotor angle: $\omega_r[k] \approx \frac{\theta_r[k]-\theta_r[k-1]}{h}$ and the rotor flux is estimated integrating the dynamic equations of the flux (see appendix B). Note that the task of the FS-MPC is to control the torque and rotor flux, thus only these are predicted, nevertheless, the whole state of the motor is required to evaluate the system model.

In section 4 we developed a torque controller for the PMSM, which considered two control goals: torque reference tracking and maximum torque per ampere operation. They were put together by means of a cost function (see equation (4.5)), that measured the degree at which a prediction of the state satisfied the control goals, i.e., the distance between the predicted state and the desired state. In the same direction, we can develop a centralized controller for the torque and flux of the induction motor defining a cost function that weights both control goals together.

$$F_c(x) = e_{T_e}^2 + \lambda^2 \left| \tilde{e}_{\psi_d} + \operatorname{sgn}(e_{i_d}) \tau e_{i_d}^2 \right| , \tag{6.16}$$

The first component of $F_c(\cdot)$ pertains to the electrical torque error

$$e_{T_e} = \frac{3pk_r}{2} |\vec{\psi}_r|^* i_q - T_e^* , \tag{6.17}$$

the second component corresponds to the cost function developed for the double integrator with the definitions in (6.14) and (6.4), which lead to

$$e_{\psi_d} = \psi_d - |\vec{\psi}_r|^* \tag{6.18}$$

$$e_{i_d} = i_d - \frac{1}{L_m}|\vec{\psi}_r|^* \tag{6.19}$$

the term $\tilde{e}_{\psi_d}$ is defined as $\tilde{x}_1$ in (5.16) with the constraint for i_d is set

$$\hat{x}_0 = \hat{i}_d = 10\text{A}. \tag{6.20}$$

Parameter τ is given by:

$$\tau = \frac{\tau_0}{2\tau_1 \hat{u}} = \frac{\sigma L_s}{2\frac{\tau_r}{L_m}\frac{\sqrt{3}}{3}v_{dc}}\,. \tag{6.21}$$

The scaling factor λ is calculated so that the weight of i_q in the first term is equivalent to the weight of i_d in the second term

$$\lambda = \frac{3pk_r}{2}|\vec{\psi}_r|^*\frac{1}{\tau}\,. \tag{6.22}$$

The reference for the rotor flux magnitude $|\vec{\psi}_r|^*$ is used in (6.17) and (6.22) instead of the actual flux magnitude to avoid numerical instability when the flux is very small.

The controller is implemented using the finite-set control algorithm, depicted in Fig. 3.4 with the cost function in eq. (6.16) and a discrete version of the system dynamics, described in eq. (2.56).

Fig. 6.4 presents experimental results using this control scheme, considering a step reference for the rotor flux. These results show how the scheme effectively fixes the closed loop dynamics to follow the switching curve and the system behaves as a double integrator being time-optimally controlled: both the stator current i_d and the flux ψ_d reach their references at the same time and as fast as the physical constraints of the system allow. The scheme works as a very high gain controller, however, the steady state performance is not compromised, i.e., measurement noise is not particularly amplified once steady state is achieved.

Fig. 6.5 presents experimental results using a PI controller for the rotor speed and the proposed controller as subordinated torque controller. Note that the flux control is precise and effective, even when the speed in not zero and a step command change command is given to the torque, for example, near $t \approx 0.75$s.

Note that these results display considerable ripple in the stator currents. This is caused by the fact that a single switching state is applied by the voltage source inverter, for the whole of a sampling period, which is the basic assumption for finite-set MPC. This situation is ameliorated in the next section, introducing a controller which makes use of a continuous actuation set, in the same fashion as the controller introduced in section 4.

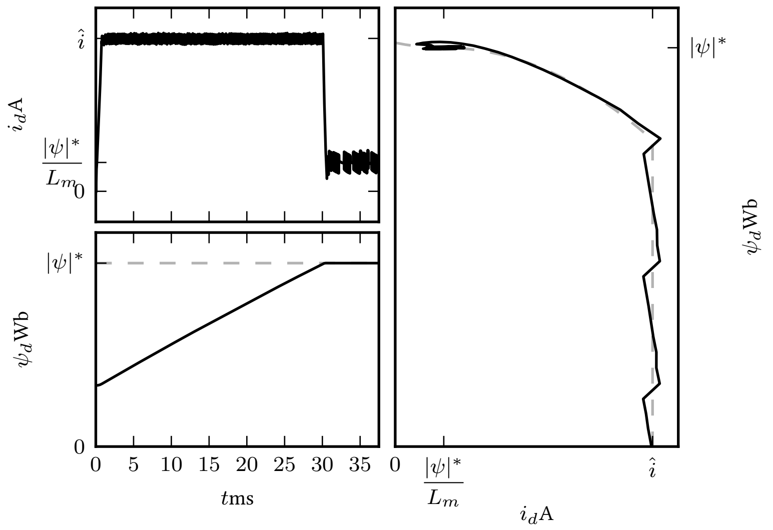

Figure 6.4.: Experimental results: transient for a step flux reference using the FS-MPC algorithm and the cost function defined in eq. (6.16). The plots on the right side present the i_d stator current and the ψ_d rotor flux components in time. The plot on the right presents these states in the $i_d\,\psi_d$ plane. The dashed line in this plot represents the switching curve, defined using the double integrator approximation. The reference for the flux magnitude is $|\psi|^* = 0.6$Wb and for the i_d current $i_d^* = \frac{|\psi|^*}{L_m} = 1.89$A. The constraint for the current is $\hat{i} = 10$A. During this test the rotor is at rest and the sampling time used is $h = 30.725$µs.

6.3. Torque control for the induction motor with continuous actuation set

A model predictive controller, assuming a continuous actuation set, can be implemented for the centralized control of flux and torque in the induction motor, putting together the double integrator approximation for the flux dynamics, the smoothened quasi-time-optimal controller (developed for the double integrator in section 5.2) and the continuous set model predictive control algorithm (developed for the PMSM in section 4.3).

The control scheme is depicted in Fig. 6.6.

The application of the SQTOC is straight forward: the dynamics of the rotor flux are approximated using the double integrator model and the SQTOC is applied, as in section 5.2, using the definition in (6.14).

The CS-MPC algorithm is completely defined to be applied in this situation with the definition of the transformation $g(x)$ (see Fig. 4.5)

$$\begin{aligned} e &= g(x) \\ &= \left[\begin{array}{c} e_{T_e} \\ e_d \end{array} \right] \\ &= \left(\begin{array}{l} \frac{3pk_r}{2} |\vec{\psi}_r|^* i_q - T_e^* \\ i_d - i_d^* \end{array} \right). \end{aligned} \tag{6.23}$$

Figures 6.7 and 6.8 present experimental results using this control scheme. The test presented in these figures are exactly the same as in figures 6.4 and 6.5 in the last section, but the sampling period used is $h = 61.5$µs.

The dynamic behaviour produced by this scheme is equivalent to that generated by the control system presented in the last section, which makes use of the FS-MPC algorithm: both the stator current i_d and the flux ψ_d reach their references at the same time and as fast as the physical constraints of the system allow, without compromising the steady state performance, i.e., measurement noise is not particularly amplified once steady state is achieved. In comparison with the scheme presented in last section, the steady state behaviour produced by this scheme is more stable and, although it is not displayed in these results, the inter-sample ripple is the expected, when using PWM and the stator currents have a concentrated frequency spectrum. The diminishing in the overall ripple also allows to see more clearly the effect of the interactions between both current components. These effects are, nonetheless, effectively compensated.

In the next chapter, the same approach presented here is used to produce a centralized torque and speed controller for the PMSM.

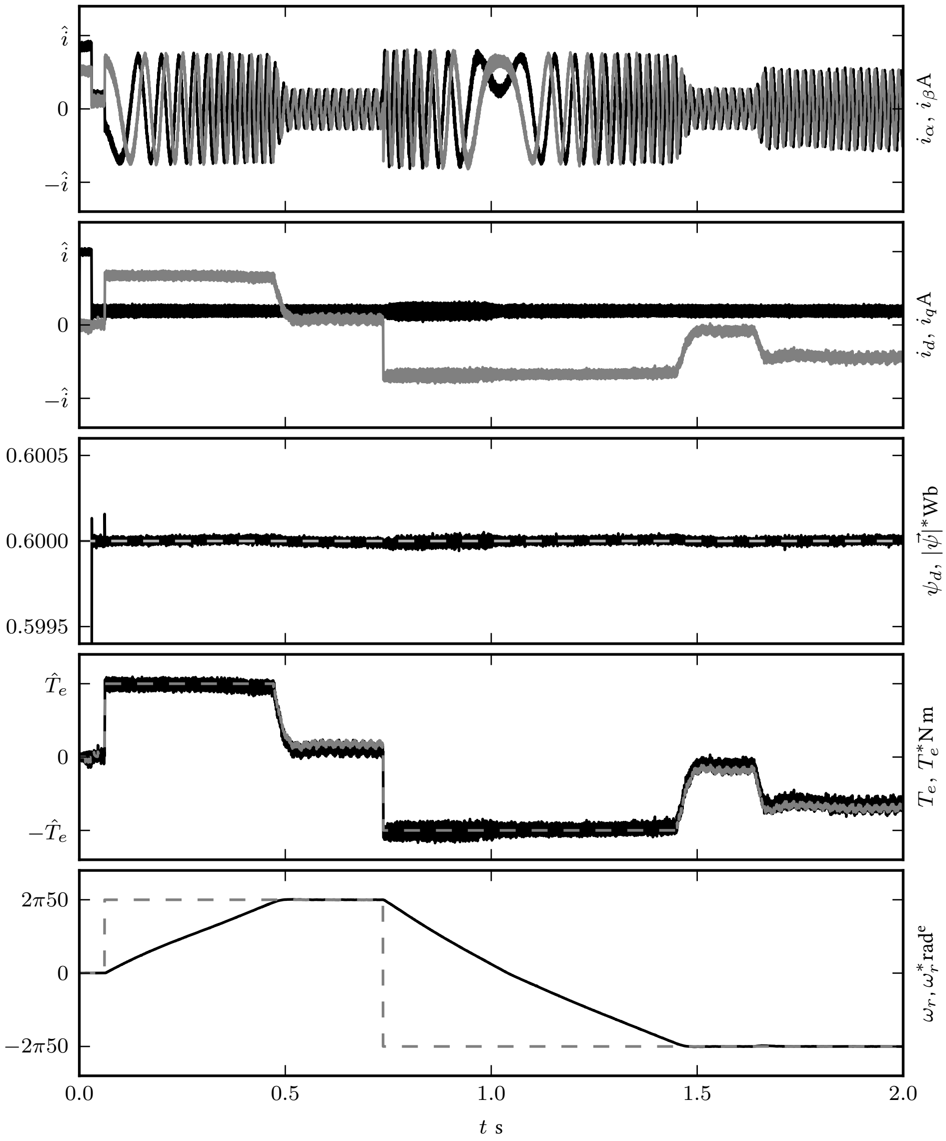

Figure 6.5.: Experimental results: speed control for the IM, with the proposed controller acting as subordinated torque controller. The speed controller is adjusted using pole placement, considering only the dynamics between the electrical torque and the rotor speed, to obtain closed-loop poles with natural frequency $\omega_n = 2\pi 10\text{rad}\,\text{s}^{-1}$ and damping ratio $\xi = \frac{\sqrt{2}}{2}$. The test considers magnetization, startup to nominal speed and speed reversal. At $t \approx 1.7$ a step load torque is applied. The ripple magnitude is slightly bigger, when the rotor is being braked. This effect is caused by the increase of the dc-link voltage, which is storing the kinetic energy being extracted from the rotor. In this test $\hat{\imath} = 10\text{A}$ and $\hat{T}_e = 6\text{N}\,\text{m}$.

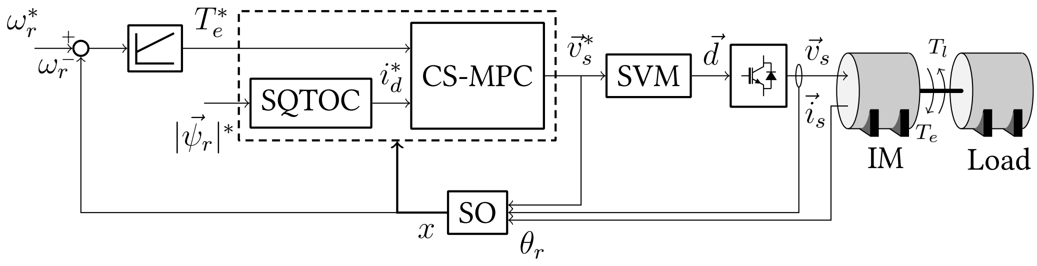

Figure 6.6.: Control scheme: speed control for the induction motor with centralized continuous-set model predictive torque and flux control. Block SO represents a state observer, which estimates the state feedback x. The block SQTOC is the smoothened quasi-time-optimal controller developed for the double integrator in section 5.2.

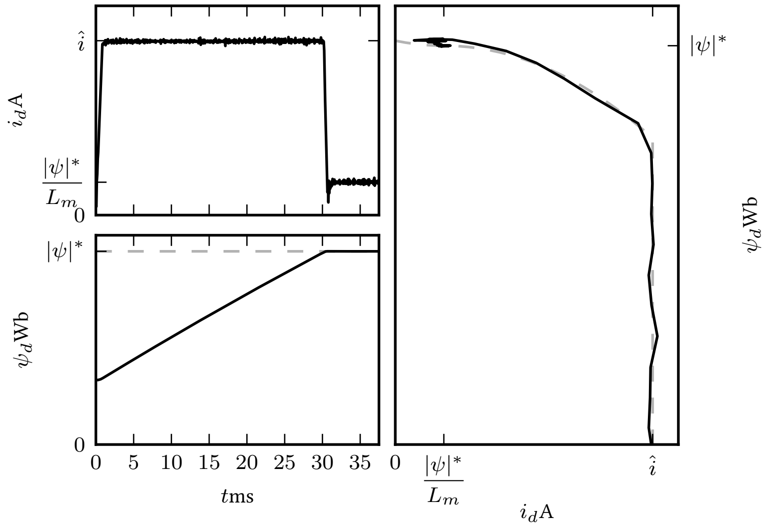

Figure 6.7.: Experimental results: transient for a step flux reference using the CS-MPC algorithm and the SQTOC scheme for centralized control of the torque, stator flux and i_d current in the IM. In this test the reference for the flux is set $|\psi|^* = 0.6$Wb and the constraint for the stator current $\hat{i} = 10$A.

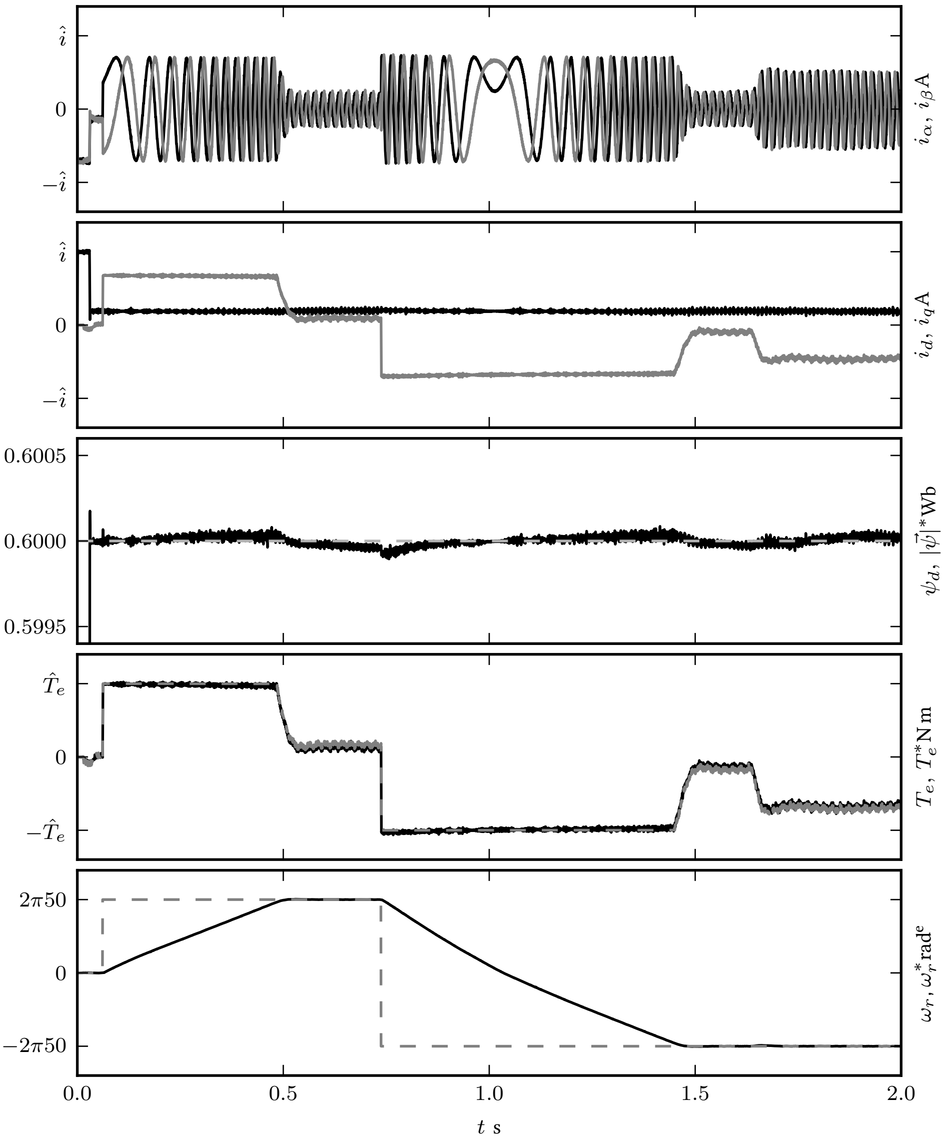

Figure 6.8.: Experimental results: speed control for the IM using the proposed continuous-set model predictive control algorithm for the centralized control of the torque, stator flux and i_d current. In this test $\hat{i} = 10$A and $\hat{T}_e = 6$N m.

7. Speed control for the permanent magnet synchronous motor

In this chapter we follow the procedure introduced in the last chapter: use the double integrator moder to fix the dynamics of the motor, as understood in the framework of field oriented control, to devise centralized controllers, in this case, for the speed of the PMSM.

The dynamics to be approximated using the double integrator are those between the stator voltages, the torque devised by the motor and the speed of the rotor. The stator currents are controlled to obtain maximum torque per ampere operation.

The result are quasi-time-optimal controllers implemented using the finite-set and the continuous-set algorithms described in chapters 3.2 and 4.3.

7.1. Rotor speed dynamics approximation using the double integrator

The dynamics between the stator voltage $\vec{v}_s$ and the electrical torque T_e in the PMSM are non-linear (see eqs. (2.25), (2.28) and (2.30)), specially if MTPA operation is required. Still, some approximations can be done in order to distill them into a form more convenient to our end: approximate them using a simple integrator, so that the dynamics between the stator voltages and the speed can be approximated using a double integrator.

In the PMSM the difference between L_d and L_q should by design be small. In the motor used in this work for experimental tests $L_d - L_q \approx 2.7\,\text{mH}$, whereas $\psi_m = 0.226\,\text{Wb}$. This means that in the electrical torque

$$T_e = \frac{3}{2}p(\psi_m i_q + (L_d - L_q)i_d i_q)\,, \tag{7.1}$$

the most important component is the first: due to the interaction between the rotor magnet and the stator currents. With this, we can assume that the motor is more efficiently operated with $i_d = 0$ and the relevant mechanical dynamics of the PMSM can be written as

$$\frac{d}{dt}\begin{bmatrix} T_e \\ \omega_r \end{bmatrix} = \begin{pmatrix} \frac{3p\psi_m}{2L_q}\left(v_q - \psi_m\omega_r\right) - \frac{r_s}{L_q}T_e \\ \frac{p}{J_m}\left(T_e - T_l\right) \end{pmatrix}. \tag{7.2}$$

We wish to fix the dynamics of the closed loop using double integrator model

$$\frac{d}{dt}\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{pmatrix} \frac{1}{\tau_0}u \\ \frac{1}{\tau_1}x_0 \end{pmatrix}, \tag{7.3}$$

with $u \in [-\hat{u}, \hat{u}]$ for the continuous-set actuation algorithm and $u \in \{-\hat{u}, 0, \hat{u}\}$ for the finite-set actuation algorithm. This is, we want to devise a controller assuming that the dynamics in eq. (7.2) behave as a double integrator, so that it steers x_0 (T_e) to correct errors in x_1 (ω_r) in minimum time, following the switching curve

$$x_1 = -\text{sgn}(x_0)\tau x_0^2, \tag{7.4}$$

where $\tau = \frac{\tau_0}{2\tau_1\hat{u}}$, towards the origin or the target, in this case:

$$\begin{bmatrix} T_e \\ \omega_r \end{bmatrix} = \begin{bmatrix} T_l \\ \omega_r^* \end{bmatrix}. \tag{7.5}$$

Just as in the last chapter, the approximation with the double integrator seems very rough and neglecting the effect of the stator resistance is not enough to make it better: the coupling between both equations, with the term $\psi\omega_r$, can not be easily disregarded. Its effect and the validity of the approximation can be quantified studying the step response of the system (7.2). Neglecting the effect of the stator resistance a with $u(t) = \hat{u}\mu(t)$:

$$T_e(t) = \frac{\hat{u}}{\frac{2L_q}{3p\psi_m}}\left(t - \frac{\psi_m}{\frac{2L_q}{3p\psi_m}\frac{J_m}{p}}\left(\frac{t}{\omega^2} - \frac{\sin(\omega t)}{\omega^3}\right)\right) \tag{7.6}$$

$$\omega_r(t) = \frac{\hat{u}}{\frac{2L_q}{3p\psi_m}\frac{J_m}{p}}\frac{1}{\omega^2}\left(1 - \cos(\omega t)\right) \tag{7.7}$$

with

$$\omega = \sqrt{\frac{3p^2\psi_m^2}{2L_qJ_m}}. \tag{7.8}$$

In the motor used for the experimental tests $\omega = 2\pi \cdot 11.389$rad s⁻¹. One period of this resonance is about 233 times longer than the time needed to steer T_e from zero to its constraint $\hat{T}_e = 9.58$N m, which is the maximum torque this motor can produce with $|\vec{i}_s| \leq 10$A. This means that its influence can be disregarded during the transients. This becomes more clear looking at the leading terms of the Taylor series for eqs. (7.6) and (7.7)

$$T_e(t) = \frac{\hat{u}}{\frac{2L_q}{3p\psi_m}}t + \mathcal{O}(t^3) \tag{7.9}$$

$$\omega_r(t) = \frac{\hat{u}}{\frac{2L_q}{3p\psi_m}\frac{J_m}{p}}\frac{t^2}{2} + \mathcal{O}(t^4), \tag{7.10}$$

which match the step response of the double integrator:

$$x_0(t) = \frac{1}{\tau_0}\hat{u}t \tag{7.11}$$

$$x_1(t) = \frac{1}{\tau_0\tau_1}\frac{\hat{u}t^2}{2} \tag{7.12}$$

In this sense, using the double integrator to approximate the dynamics in eq (7.2) is valid, at least for the transients, with:

$$\begin{aligned} u &= v_q \\ x_0 &= T_e \qquad \tau_0 = \frac{2L_q}{3p\psi_m} \\ x_1 &= \omega_r \qquad \tau_1 = \frac{J_m}{p}\,, \end{aligned} \tag{7.13}$$

Figure 7.1 presents simulations of the step response of the system dynamics and the double integrator using the real parameters of the motor. These simulations were carried out with the time running backwards, this is

$$u(t) = \hat{u}\mu(-t) \tag{7.14}$$

is applied to the system. With this, the simulated step response of the double integrator matches the switching curve described in the chapter 5.1.

These simulations help grasp the scale of the influence the higher order terms in eqs. (7.9) and (7.10) have on the system dynamics. The error introduced by the double integrator approximation, which disregards these higher order terms, is relatively small when x_0 reaches its constraint and tends to null at $t = 0$. The real system is also faster than the double integrator when it comes to *get away* from the origin, consequently, using the double integrator model to fix the dynamics will always lead to conservative closed loop dynamics: slightly slower than the limits put by the real model. This means that, when the system parameters are perfectly known, the approximation will not cause overshoot nor system instability.

The coupling term in eqs. (7.6) and (7.7) pertains to the back-emf and its value does change the ability of u to manipulate the state: the step response of the system with the maximum values for u and, consequently, the optimal switching curve are affected. In the experimental setup, the back-emf is equivalent to 50% of the maximum voltage the VSI can produce, with the rotor running at maximum speed $\omega_r = 2\pi \cdot 150\text{rad}^\text{e}\,\text{s}^{-1}$. Moreover, as hinted in the last chapter, the maximum voltage the VSI can generate in the q axis (or an axis aligned with the MTPA curve) also depends on the rotor angle (see Fig. 6.1) and, when operating with MTPA, also on the working point in the state space.

In an attempt to solve the trade-off between a good approximation of the dynamics at hand and algorithm simplicity, the control methods in the following sections are developed assuming

$$\hat{u} = \frac{\sqrt{3}}{3}v_{dc} \qquad \text{and} \qquad \hat{u} = 0.9 \cdot \frac{\sqrt{3}}{3}v_{dc}\,. \tag{7.15}$$

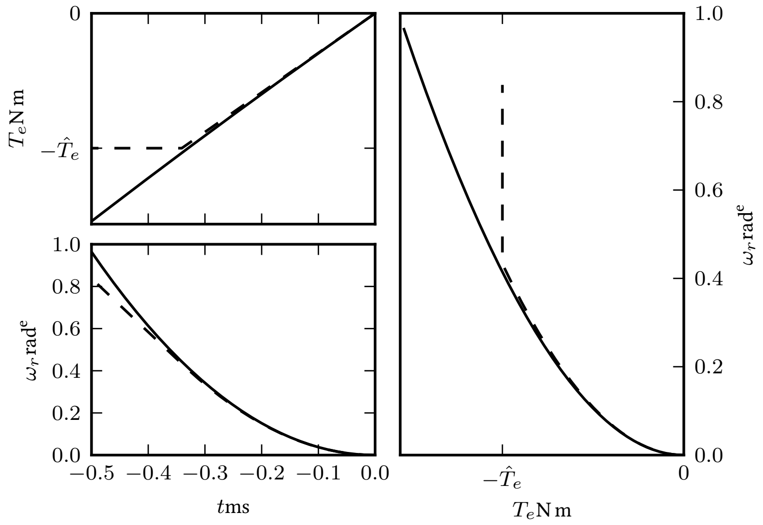

Figure 7.1.: Step response (with time running backwards), of the double integrator (dashed) and the rotor speed dynamics (solid), described in eq. (7.2). In the response for the double integrator, the actuation is set to zero once the electrical torque reaches its constraint $\hat{T}_e = 9.58$N m.

for the finite-set and the continuous-set algorithms respectively.

Effects relevant in steady state: back-emf, stator resistance, load torque, as well as the cross couplings between the stator currents components, are accounted for by these controllers using the whole model of the motor as predictive model.

The effect of neglecting these terms in the closed loop dynamics is very small, as is verified by the experimental results presented in next section.

7.2. Speed control for the synchronous motor with finite actuation set

Using the double integrator approximation, a quasi-time-optimal controller for the rotor speed of the PMSM is devised, in this case, assuming a finite-set actuation: the two-level VSI applies a single switching configuration for the whole of a sampling period. The controller is implemented using the finite-set algorithm introduced in section 3.2 and depicted in Fig. 3.4.

A scheme of the control system is presented in Fig. 7.2.

The two control goals: speed reference tracking and maximum torque per ampere operation are put together using the cost function:

$$F_c(x) = \left|\tilde{e}_{\omega_r} + \operatorname{sgn}(e_{T_e})\tau e_{T_e}^2\right| + \lambda^2 e_d^2\,. \tag{7.16}$$

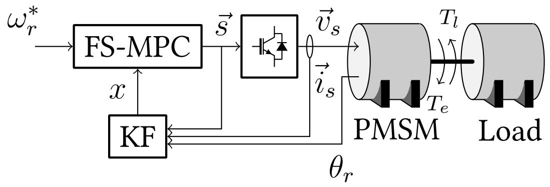

Figure 7.2.: Control scheme: centralized finite-set model predictive speed control for the PMSM. Block KF represents a state observer: a reduced order extended Kalman filter. It is introduced in this scheme to separate the control problem, form the observation problem: the Kalman filter is about the best solution for the later. Its function is to reconstruct the system state x out of the measurements $\vec{i}_s$ and θ_r. For more details on the Kalman filter refer to appendix A. The controller (FS-MPC) takes the state feedback x and the speed reference ω_r^* and calculates the configuration for the VSI switches $\vec{s}$ to obtain speed reference tracking and maximum torque per ampere operation.

The first component of $F_c(\cdot)$ corresponds to the cost function developed for the double integrator with the definitions in (7.13) and (7.15), which lead to

$$e_{\omega_r} = \omega_r - \omega_r * \tag{7.17}$$

$$e_{T_e} = T_e - T_l \,, \tag{7.18}$$

and

$$\tau = \frac{\tau_0}{2\tau_1 \hat{u}} = \frac{L_q}{\psi_m J_m \sqrt{3} v_{dc}} \,. \tag{7.19}$$

Term $\tilde{e}_{\omega_r}$ is defined as $\tilde{x}_1$ in (5.16) with the constraint for T_e set to

$$\hat{x}_0 = \hat{T}_e = 9.58\mathrm{N\,m} \,, \tag{7.20}$$

which is the maximum torque the motor can produce considering MTPA operation and a constraint for the magnitude of the stator currents:

$$\widehat{|\vec{i}_s|} = 10\,\mathrm{A} \,. \tag{7.21}$$

The second term in $F_c(\cdot)$ imposes the MTPA operation

$$e_d = i_d + \frac{L_d - L_q}{\psi_m} \left(i_d^2 - i_q^2 \right) \tag{7.22}$$

and the λ scaling factor is set, so that the weight of the leading i_d^2 term, produced by e_d^2 in $F_c(\cdot)$, equates the weight of the leading i_q^2 term produced by e_{Te}^2 in $F_c(\cdot)$:

$$\lambda = \tau \frac{3}{2} p \psi_m \,. \tag{7.23}$$

The controller is implemented using the finite-set control algorithm, depicted in Fig. 3.4 with the cost function in eq. (7.16) and a discrete version of the system dynamics, described in eq. (2.36).

Figures 7.3 and 7.4 present experimental results using this control scheme. For this test the sampling time is given by $h = 30.725$µs.

These results show how the scheme effectively fixes the closed loop dynamics follow the switching curve: the system behaves as a double integrator being time-optimally controlled and the electrical torque is bounded by its constraint. Note that both variables reach their steady state values virtually at the same time, this is, as fast as the physical constraints of the system allow. This is not detrimental for the steady state performance of the system and measurement noise, once steady state is achieved, not particullarly amplified.

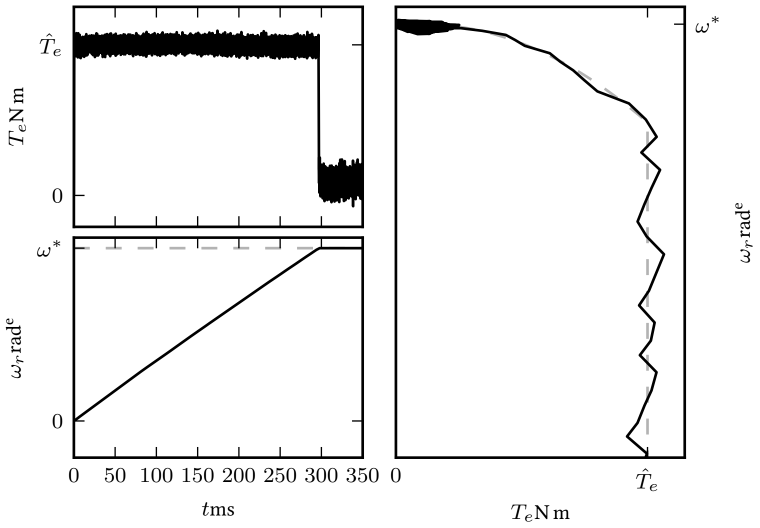

Figure 7.3.: Experimental results: transient for step speed reference using the FS-MPC algorithm and the cost function defined in eq. (7.16). The plots on the right side present the electrical torque T_e and the rotor speed ω_r in time. The plot on the left side presents these variables in the $T_e\omega_r$ plane. In this plot, the dashed grey line represents the switching curve, defined using the double integrator approximation. The reference for the speed is $\omega_r^* = 2\pi \cdot 150$rad^e s^{-1} and the constraint for the torque is $\hat{T}_e = 9.58$N m. In steady state the electrical torque chatters around 0.8N m, to compensate for the friction, which is regarded by the control system as a load torque. The later is estimated by the Kalman filter (see appendix A).

Note that in these tests the electrical torque is steered towards its constraint at the beginning and the end of the speed transients, when the rotor speed reaches its reference, this means that maximum actuation is applied in both situations. The dynamics of the response to the step torque impact, however, is limited by the convergence speed of the

Kalman filter.

In Fig. 7.4 it is possible to notice how the controller commands both current components i_d and i_q to achieve MTPA operation.

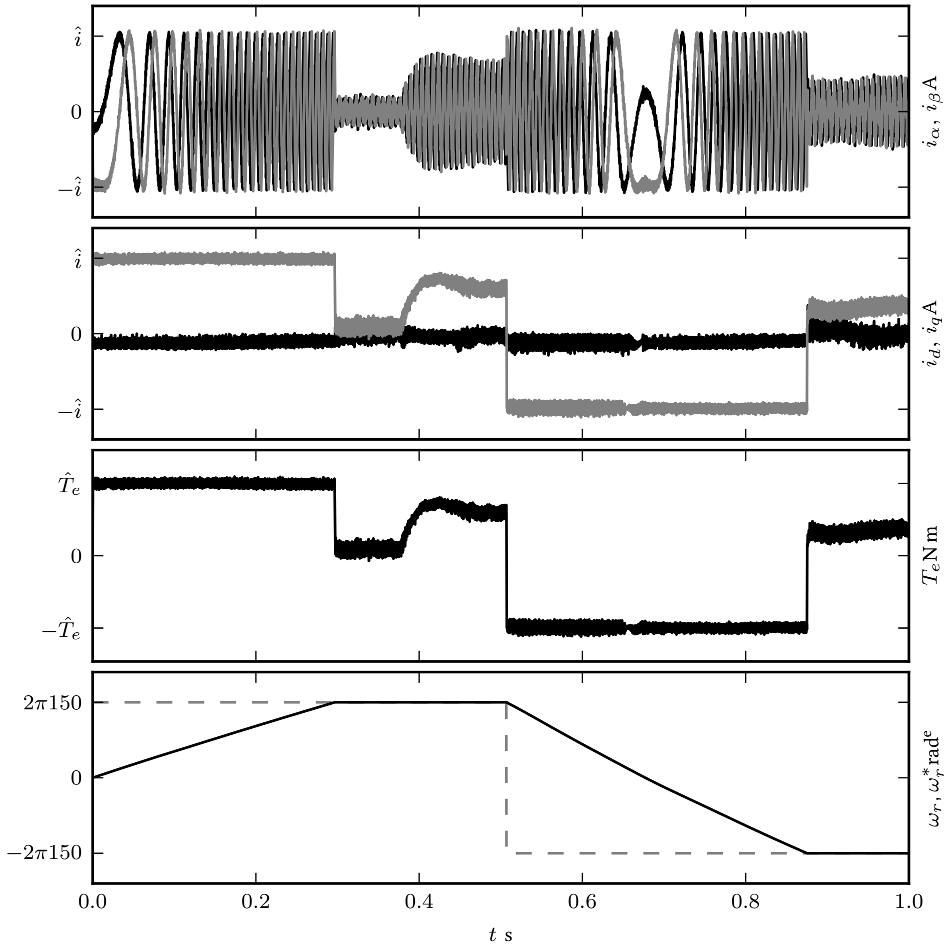

Figure 7.4.: Experimental results: speed control for the PMSM using the FS-MPC algorithm and the cost function defined in eq. (7.16). The test considers startup to nominal speed, step load torque impact and speed reversal. In this test the constraints for the stator currents and electrical torque are given by $\hat{i} = 10$A and $\hat{T}_e = 9.58$N m.

As in the last chapter, the ripple in the stator currents, produced by the finite-set algorithm, is ameliorated introducing a control algorithm, which makes use of a continuous-set algorithm.

7.3. Speed control for the synchronous motor with continuous actuation set

A centralized speed controller for the PMSM, assuming a continuous actuation set, is devised using the double integrator approximation of the rotor speed dynamics, the smoothened-quasi-time-optimal controller, developed for the double integrator in section 5.2 and the continuous-set model predictive torque controller for the PMSM, developed in section 4.2.

Fig. 7.5 presents a scheme of the control system.

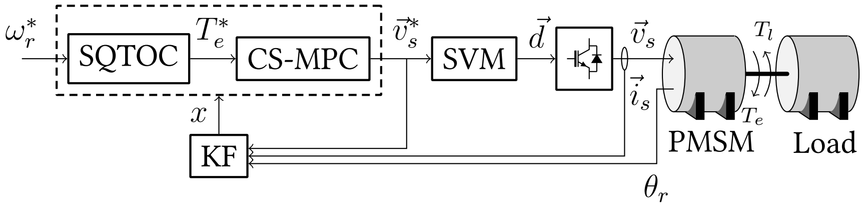

Figure 7.5.: Control scheme: centralized continuous-set model predictive speed control for the PMSM. The controller is divided in two parts, the smoothened-quasi-time-optimal controller takes the reference for the speed ω_r^* and calculates a reference for the torque T_e^*, the continuous-set model predictive torque controller (CS-MPC) takes this reference and calculates a reference for the stator voltage $\vec{v}_s^*$, which is then transformed to duty cycles by the SVM block. The whole controller uses the state feedback x, synthesized by the Kalman filter, out of the measurements of the rotor angle θ_r and the stator currents $\vec{i}_s$.

Figures 7.6 and 7.7 present experimental results using this control scheme. The tests presented in these plots are the same as in figures 7.3 and 7.4 with a sampling time $h = 46.1$µs. The dynamic behaviour produced by it is equivalent to that generated by the scheme presented in the last section, considering the FS-MPC algorithm: both the electrical torque T_e and the rotor speed ω_r reach their references at the same time and as fast as the physical constraints of the system allow, without compromising the steady state performance, i.e., measurement noise is not particularly amplified once steady state is reachedj. In comparison with the scheme presented in last section, less ripple is observed during the steady state and, although it is not displayed in these results, the inter-sample ripple is the expected, when using PWM and the stator currents have a concentrated frequency spectrum.

In the next chapter, the schemes presented in this and the last chapters, are combined to produce a centralized speed, torque and flux controller for the induction motor.

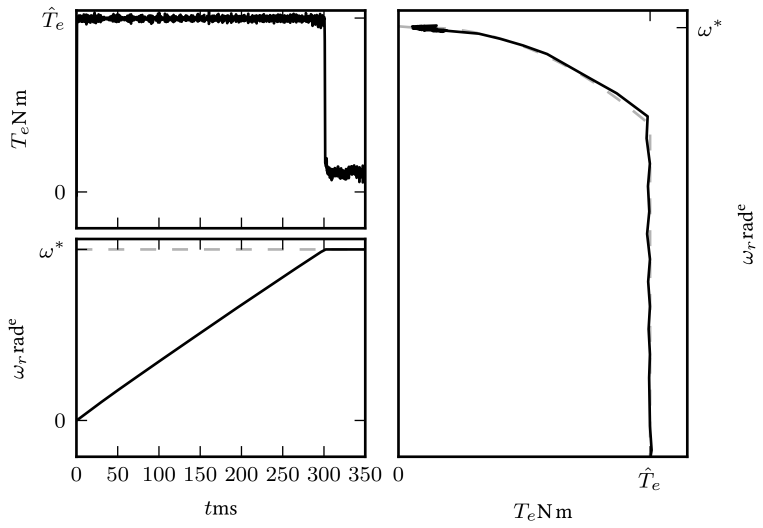

Figure 7.6.: Experimental results: speed step reference using the CS-MPC algorithm and the SQTOC scheme for centralized speed and torque control of the PMSM. In this test the constraint for the electrical torque is given by $\hat{T}_e = 9.58$N m and the reference for the rotor speed is $\omega^* = 2\pi \cdot 150$rad^e s^{-1}.

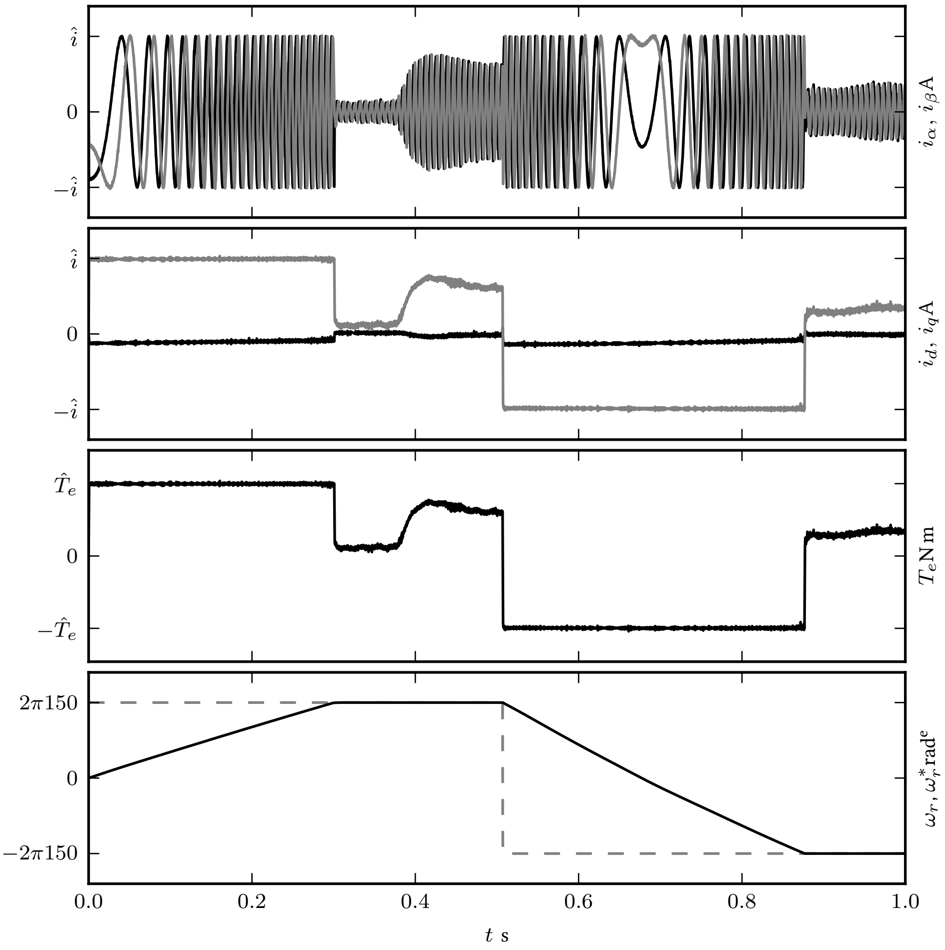

Figure 7.7.: Experimental results: speed control for the PMSM using the CS-MPC and the SQTOC scheme for centralized speed and torque control of the PMSM. In this test the constraints for the stator currents and electrical torque are given by $\hat{i} = 10$A and $\hat{T}_e = 9.58$N m.

8. Speed Control for the Induction Motor

In this chapter the developments presented in the last two chapters: the quasi-time-optimal controllers for the torque and flux of the induction motor and for the speed in the PMSM, are combined to produce a centralized quasi-time-optimal controller for the rotor speed in the induction motor.

The design follows the ideas set by field oriented control and the problem is separated in two, one part relating to the flux control, and the other to the speed and torque control. The closed loop dynamics are fixed using the double integrator approximation, in this case, in two separate or parallel branches: one branch corresponds to the dynamics between stator voltage, the i_d stator current component and the rotor flux magnitude. The second branch pertains to the dynamics between stator voltage, the electrical torque and the rotor speed.

The control system is designed to be implemented using the finite-set and the continuous-set algorithms described in chapters 3.2 and 4.3.

8.1. System dynamics approximation using double integrators

The system dynamics at hand are separated in two branches, the first one related to the rotor flux dynamics

$$\frac{d}{dt}\left[\begin{array}{c} i_d \\ \psi_d \end{array}\right] = \left(\begin{array}{l} \frac{1}{\sigma L_s}v_d - \frac{1}{\tau_\sigma}i_d + \frac{k_r}{\tau_r \sigma L_s}\psi_d + \omega_k i_q \\ \frac{L_m}{\tau_r}i_d - \frac{1}{\tau_r}\psi_d \end{array}\right) \tag{8.1}$$

and the second one to the rotor speed dynamics

$$\frac{d}{dt}\left[\begin{array}{c} T_e \\ \omega_r \end{array}\right] = \left(\begin{array}{l} \frac{3}{2}pk_r\left(\frac{d\psi_d}{dt}i_q + \psi_d\frac{di_q}{dt}\right) \\ \frac{p}{J_m}\left(T_e - T_l\right) \end{array}\right) \tag{8.2}$$

The derivatives in 8.2 are given by

$$\frac{d\psi_d}{dt} = \frac{L_m}{\tau_r}i_d - \frac{1}{\tau_r}\psi_d \tag{8.3}$$

$$\frac{di_q}{dt} = \frac{1}{\sigma L_s}v_q - \frac{1}{\tau_\sigma}i_q - \frac{k_r}{\sigma L_s}\omega_r\psi_d - \omega_k i_d \tag{8.4}$$

The approximation of the flux dynamics using the double integrator, discussed in in section 6.1, is achieved with the definitions

$$\begin{aligned} u &= v_d & & \\ x_0 &= i_d & \tau_0 &= \sigma L_s \\ x_1 &= \psi_d & \tau_1 &= \frac{\tau_r}{L_m}. \end{aligned} \tag{8.5}$$

The approximation of the rotor speed dynamics requires more analysis and assumptions, for the approximation to make sense.

The first assumption is not rare in drives controlled with FOC: that the motor will be operated with a constant flux reference. We also assume that the controller will be able to sustain this condition and that changes in the speed reference will occur only once the flux had reached its reference. With this $\frac{d\psi_d}{dt} = 0$, $i_d = \frac{1}{L_m}\psi_d$ and the time derivative for the torque is reduced to

$$\frac{dT_e}{dt} = \frac{3pk_r\psi_d}{2\sigma L_s}\left(v_q - \frac{L_s}{L_m}\psi_d\omega_r\right) - \left(\frac{1}{\tau_\sigma} + \frac{1}{\tau_r}\right)T_e \tag{8.6}$$

This equation has the same form as the torque component in eq. (7.2). As shown in section 7.1, if the effect of the stator and rotor resistances are neglected ($\frac{1}{\tau_s} + \frac{1}{\tau_r} = 0$), the step response of this system matches that of the double integrator, right after the application of the input step and then drifts away from it displaying a resonant behaviour, in this case, with natural frequency

$$\omega = \sqrt{\frac{3p^2k_r\psi_d^2}{2\sigma L_s J_m}\frac{L_s}{L_m}}\,. \tag{8.7}$$

In the motor used for the experimental tests $\omega = 2\pi \cdot 7.558\text{rad s}^{-1}$. One period of this resonance is about 187 times longer than the time needed to steer T_e from zero up to its constraint $\hat{T}_e = 6.0\text{N m}$, which is the maximum torque this motor can sustain at $\omega_r = 2\pi 50\text{rad s}^{-1}$ with $|\vec{i}_s| \leq 10\text{A}$ and a reference for the rotor flux $|\vec{\psi}_r|^* = 0.6\text{Wb}$.

With this, the approximation using the double integrator is valid for the transients, with:

$$\begin{aligned} u &= v_q & & \\ x_0 &= T_e & \tau_0 &= \frac{2\sigma L_s}{3pk_r\psi_d} \\ x_1 &= \omega_r & \tau_1 &= \frac{J_m}{p}\,, \end{aligned} \tag{8.8}$$

The couplings between i_d and i_q and effects relevant in steady state are compensated in the controllers developed in the following sections, using the whole model of the motor as predictive model and displacing the origin of the double integrator models with the targets

$$\begin{bmatrix} i_d \\ \psi_d \end{bmatrix} = \begin{bmatrix} \frac{1}{L_m}\psi_d^* \\ \psi_d^* \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} T_e \\ \omega_r \end{bmatrix} = \begin{bmatrix} T_l \\ \omega_r^* \end{bmatrix} \tag{8.9}$$

for the flux and speed models respectively.

A formal proof of the claim that using the whole system model as predictive model will compensate the effects relevant in steady state and the couplings between the dynamics of these two "separate" double integrators is outside of the scope of this work, but we can still grasp how this works by considering standard feed-forward schemes, where knowledge on disturbances: either measurements or models, is used to compensate their effect before they affect the system. In terms of the double integrator approximation, when we use it to fix the closed loop dynamics we consider every effect that makes the behavior of the system drift away from the expected behavior to be a disturbance. By using the whole model of the system as predictive model we incorporate our knowledge on these disturbances and the controller will take it into account, when calculating the actuation. Problems can arise, however, when the intention is to obtain time-optimal behaviour, which makes the control system use only the extreme values of the actuation, leaving no margin to compensate for further effects. Nonetheless, this is only an issue when driving x_0 towards its constraint and back towards the steady state. When the state is on its target or x_0 on its constraint, the controller is supposed to apply $u = 0$, here the controller has enough room to compensate for all other effects.

To illustrate this, lets consider the crossed couplings between i_d and i_q, represented by the term $-J\omega_k\overset{\circlearrowleft}{\imath}$ in (2.51) or the terms $\omega_k i_q$ and $-\omega_k i_d$ in (8.1) and (8.4). In the last section we introduced the assumption that the machine would be operated with a constant flux reference and that changes in the speed reference would only occur once the machine is magnetized (the flux has reached its reference). In this situation $\frac{di_d}{dt} = \frac{d\psi_d}{dt} = 0$, which means $v_d = r_s i_d$. This leaves enough room for v_d to compensate for $\omega_k i_q$, when a change in the speed reference occurs. On the other hand, the term $-\omega_k i_d$ in the equation for i_q undermines the ability of u_q to manipulate it, and makes its dynamic differ from that of the double integrator. We can think of it as changing the real value of $\hat{u}$. However, this is only important when T_e is steered towards its constraint or its target, once they are reached $\frac{di_q}{dt} = 0$ and, since the controller uses the whole system model as predictive model, it has the necessary information to conclude that this is achieved with $v_q = r_\sigma - k_r\omega_r\psi_d + \omega_k i_d$.

Fig. 8.1 presents the simulation of the step response of the system dynamics in eq. 8.6 and the double integrator, with the parameters in 8.8 and the time running backwards. These results show how the approximation using the double integrator is relatively accurate for the transient, up to the point where the electrical torque reaches its constraint.

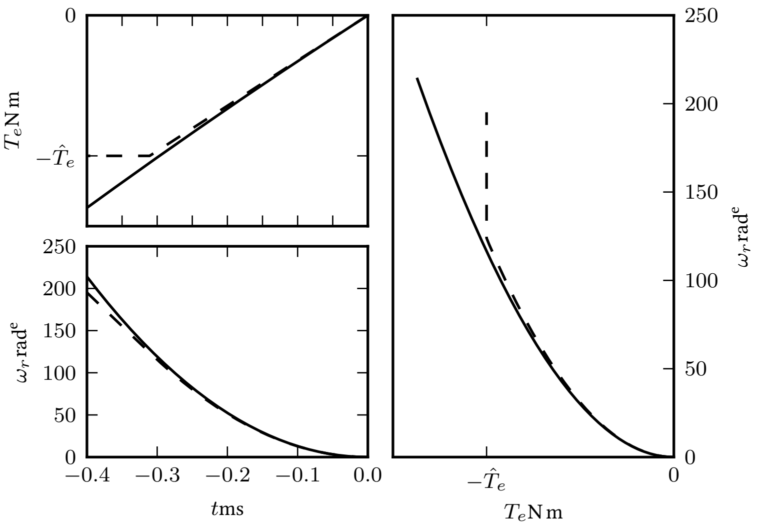

Figure 8.1.: Step response (with time running backwards), of the double integrator (dashed line) and the rotor speed (solid line) dynamics, described in eq. (7.2). In the response for the double integrator, the actuation is set to zero once the electrical torque reaches its constraint $\hat{T}_e = 6\text{N m}$.

8.2. Implementation using the finite-actuation set algorithm

In the later section we established that the dynamics of the induction motor (from stator voltages to rotor speed and flux through electrical torque and stator currents), can be approximated with the dynamics of two parallel double integrators, functioning independently, at least during transients. Of course this is only achieved under a set of assumptions, since the dynamics of all these variables are very tightly intertwined.

The control goal set for this chapter: centralized control of rotor speed, electrical torque, rotor flux and stator currents is first implemented using the finite-set algorithm introduced in section 3.2.

A scheme of the control system is presented in Fig. 8.2.

This implementation requires the introduction of an appropriate cost function, which is devised combining two versions of the costs function developed for the double integrator in section 5.1, one for the speed dynamics and the other for the flux dynamics:

$$F_c(x) = \left(\frac{1}{\tau_\omega}\tilde{e}_{\omega_r} + \text{sgn}(e_{T_e})e_{T_e}^2\right)^2 + \lambda^2\left(\frac{1}{\tau_\psi}\tilde{e}_{\psi_d} + \text{sgn}(e_{i_d})e_{i_d}^2\right)^2 , \tag{8.10}$$

In this case, squares were used instead of absolute values, to better trade-off between both control goals: when the dynamics in each case are very different, the controller tends to control more precisely the component that reduces the cost faster, while the other is let to

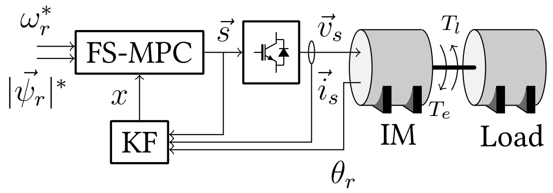

Figure 8.2.: Control scheme: centralized finite-set model predictive speed control for the IM. Block KF represents a state observer: a reduced order extended Kalman filter. It is introduced in this scheme to separate the control problem, form the observation problem: the Kalman filter is about the best solution for the later. Its function is to reconstruct the system state x out of the measurements $\vec{i}_s$ and θ_r. For more details on the Kalman filter refer to appendix A. The controller (FS-MPC) takes the state feedback x and the speed reference ω_r^* and calculates the configuration for the VSI switches $\vec{s}$ to obtain speed and flux reference tracking.

drift away from its reference. This effect is mitigated by using the square function, since it grows faster than the absolute value.

The first component of $F_c(\cdot)$ pertains to the speed control:

$$e_{\omega_r} = \omega_r - \omega_r^* \tag{8.11}$$

$$e_{T_e} = \frac{3pk_r}{2}|\vec{\psi}_r|^* i_q - T_l\,, \tag{8.12}$$

and the second component corresponds to the flux control:

$$e_{\psi_d} = \psi_d - |\vec{\psi}_r|^* \tag{8.13}$$

$$e_{i_d} = i_d - \frac{1}{L_m}|\vec{\psi}_r|^* \tag{8.14}$$

The parameters τ are given by

$$\tau_\psi = \frac{\sigma L_s}{2\frac{\tau_r}{L_m}\frac{\sqrt{3}}{3}v_{dc}} \qquad \text{and} \qquad \tau_\omega = \frac{\frac{2\sigma L_s}{3pk_r|\vec{\psi}_r|^*}}{2\frac{J_m}{p}\frac{\sqrt{3}}{3}v_{dc}} \tag{8.15}$$

and in each case $\tilde{e_{\omega_r}}$ and $\tilde{e}_{\psi_d}$ are defined as $\tilde{x}_1$ in (5.16) with the constraints

$$\hat{T}_e = 6.0\mathrm{N\,m} \qquad \text{and} \qquad \hat{i}_d = 10\mathrm{A}. \tag{8.16}$$

The scaling factor λ is calculated so that the weights of i_q and i_d in the first and the second terms of $F_c(\cdot)$ are equivalent:

$$\lambda = \left(\frac{3pk_r}{2}|\vec{\psi}_r|^*\right)^2 . \tag{8.17}$$

In this way, in steady state i_d and i_q display equivalent ripple magnitudes.

Figures 8.3 and 8.4 present experimental results using the scheme described in this section. These experiments consider a sampling time $h = 30.725$µs and a flux magnitude reference $|\psi|^* = 0.6$Wb.

The results in figure 8.3 show how the scheme manages to fix the system dynamics to those of a double integrator being control in a time-optimal fashion: both the speed and torque reach their steady state values at the same time, making maximum utilization of the constraints of the system and following the switching curve defined by the double-integrator approximation. Note that the dynamics of the response to the load torque impact is constrained by the convergence speed of the Kalman filter.

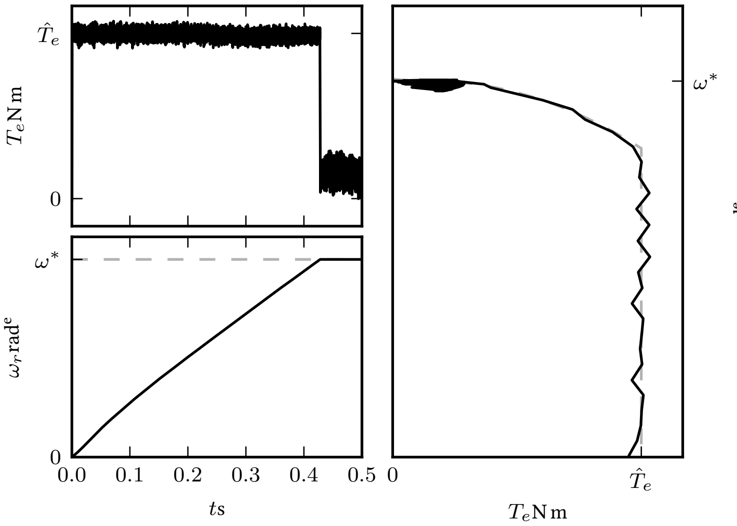

Figure 8.3.: Experimental results: transient for step speed reference using the FS-MPC algorithm and the cost function defined in eq. (8.10). The plots on the right side present the electrical torque T_e and the rotor speed ω_r in time. The plot on the left side presents these variables in the $T_e\,\omega_r$ plane. In this plot the dashed gray line represents the switching curve, defined using the double integrator approximation. The reference for the speed is $\omega_r^* = 2\pi \cdot 50$rad^e s^{-1} and the constraint for the torque is $\hat{T}_e = 6.0$N m. In steady state the electrical torque chatters around 0.8N m, to compensate for the friction, which is regarded by the control system as a load torque. The later is estimated by the Kalman filter (see appendix A).

The results in 8.4 show that the controller effectively controls the speed and flux, however, the flux control is poor, compared to the performance achieved in chapter 6: the ripple in the i_d current component and the rotor flux are considerable. As hinted before, this is produced by the cost function, which can not very effectively solve the trade-off between speed and flux control: the time constants characterizing their dynamics are very different ($\tau_r \approx 60\tau_s$) and the controller takes actuation that reduces the cost faster.

This, together with the issues produced by the FS-MPC algorithm: unstable switching pattern, high ripple and distributed frequency spectrum, are tackled in the next section, incorporating voltage modulation into the scheme.

8.3. Continuous actuation set

The continuous actuation set version of this controller is devised, in the same fashion as the finite-set version, applying the control structure for the double integrator two times, one for the rotor speed dynamics and another for the rotor flux dynamics. The control structure in this case considers smoothened quasi-time-optimal controller, developed for the double integrator in section 5.2, and the continuous set model predictive control algorithm, developed for the PMSM in section 4.3.

A scheme of the control system is presented in Fig. 8.5.

The SQTOC is applied in each case as described in section 5.2 with the definitions in (8.8) for speed control and (8.5) for flux control.

The CS-MPC algorithm (see section 4.3) is applied, as in section 6.3, with:

$$\begin{aligned} e &= g(x) \\ &= \begin{bmatrix} e_{T_e} \\ e_d \end{bmatrix} \\ &= \begin{pmatrix} \frac{3pk_r}{2}|\vec{\psi}_r|^* i_q - T_e^* \\ i_d - i_d^* \end{pmatrix}. \end{aligned} \tag{8.18}$$

Figures 8.6 and 8.7 present experimental results using this control scheme. These figures present the same tests presented in the last section, this time, with a sampling period $h = 46.1$µs. The dynamic behaviour is equivalent to that displayed by the FS-MPC algorithm, but the issues regarding poor flux control, at least for the steady state. Note that when the target lays outside the region defined by the state predictions (see Fig. 4.4), the CS-MPC scheme produces the voltage reference out of the intersection between the voltage constraints and the ray between the prediction for the zero voltage vector and the target, this is, the system state is steered in the direction of the target, even if this means letting one of the variables drift away from its reference. This is noticeable in the plot for the stator flux ψ_d in Fig. 8.7 at the time instants where the electrical torque T_e changes. This could be improved devising a better strategy to chose the voltage to be modulated, when the target lays outside the area reachable in one sampling period.

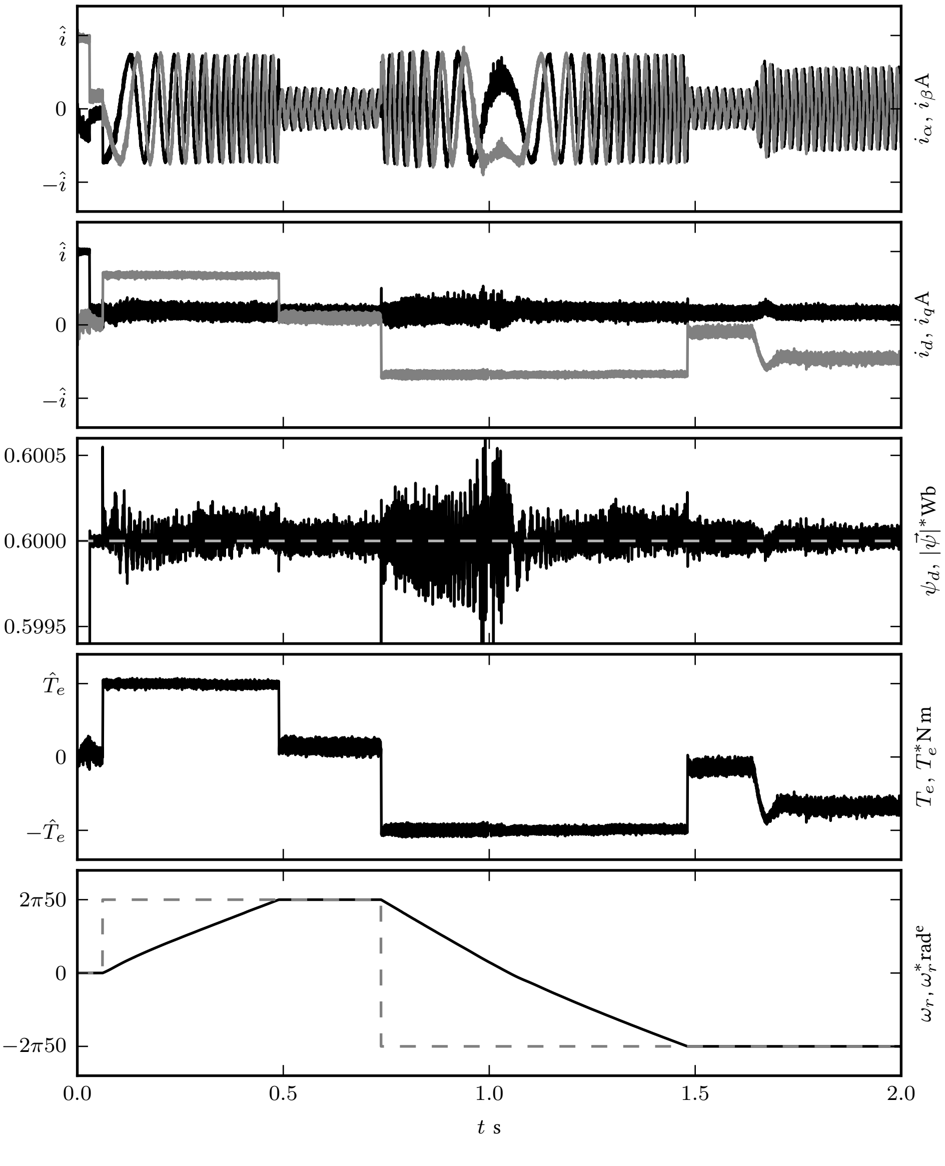

Figure 8.4.: Experimental results: centralized speed and flux control for the IM, using the FS-MPC algorithm and the cost function defined in (8.10). The constraints for the stator currents and electrical torque are given by $\hat{i} = 10\text{A}\ \hat{T}_e = 6.0\text{N m}$. The test considers magnetization, startup up to nominal speed, speed reversal and step load-torque impact.

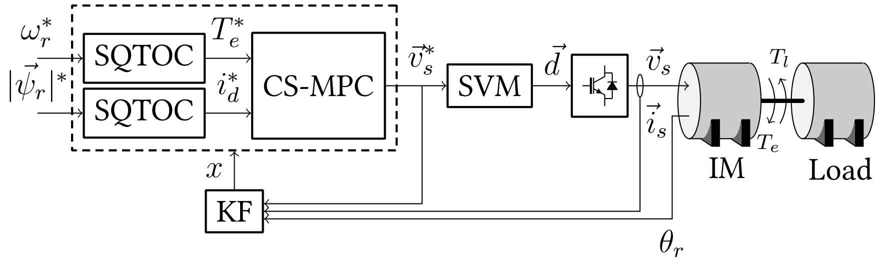

Figure 8.5.: Control scheme: centralized continuous-set model predictive speed and flux control for the IM. The controller consists in two smoothened-quasi-time-optimal controllers (SQTOC), which take the speed and flux references and generate references for the electrical torque and the i_d current component, according to the system state x to generate the time optimal dynamics of a double integrator. The continuous-set model predictive torque controller (CS-MPC) takes these reference and calculates a reference for the stator voltage $\vec{v}_s^*$, which is then transformed to duty cycles by the SVM block. The state feedback x is synthesized by the Kalman filter, out of the measurements of the rotor angle θ_r and the stator currents $\vec{i}_s$.

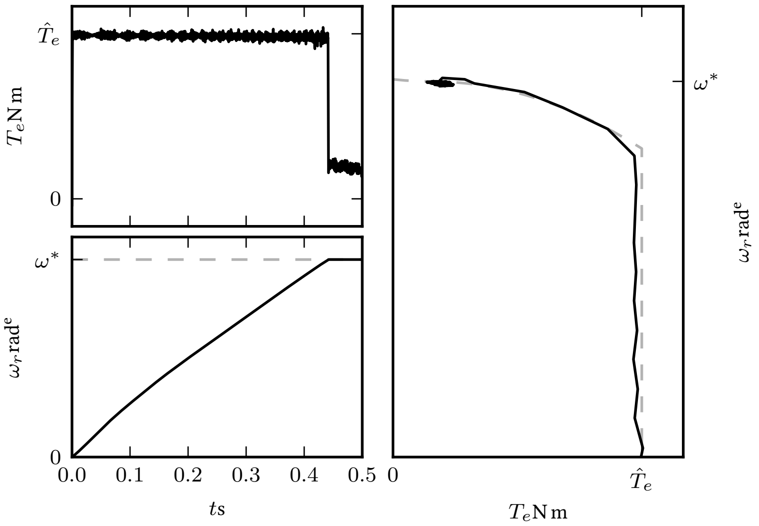

Figure 8.6.: Experimental results: speed step reference using the CS-MPC algorithm and the SQTOC scheme for centralized speed, torque, flux and stator currents in the IM. The constraint for the torque is $\hat{T}_e = 6$N m and the reference for the speed $\omega_r^* = 2\pi \cdot 50$rad^e s^{-1}.

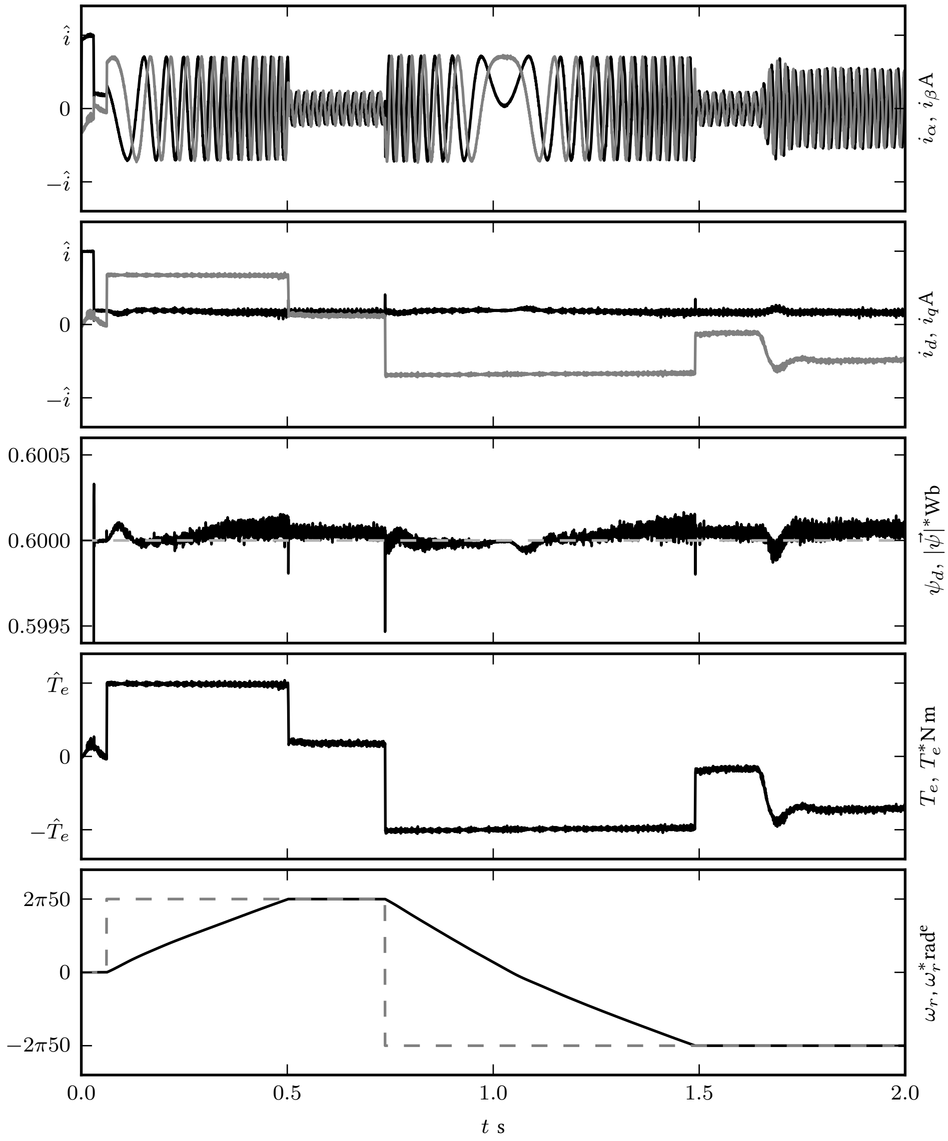

Figure 8.7.: Experimental results: speed step reference using the CS-MPC algorithm and the SQTOC scheme for centralized speed, torque, flux and stator currents in the IM. The constraints for the stator currents and electrical torque are given by $\hat{i} = 10$A $\hat{T}_e = 6.0$N m.

Part IV.

Third Order Systems

9. Position control for the permanent magnet synchronous motor

In this chapter, the control scheme developed in chapter 5: the smoothened quasi-time-optimal control for the double integrator, is extended to devise a quasi-time-optimal controller for the triple integrator (TI), considering actuation and state constraints. The resulting controller is used to devise a quasi-time-optimal control system for position control of the PMSM. A scheme of this system is depicted in Fig. 9.1.

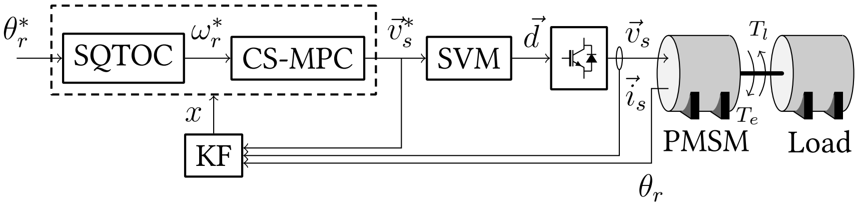

Figure 9.1.: Control scheme for quasi-time-optimal control of the position in a PMSM. The control algorithm consists of a Smoothened Quasi-Time-Optimal Controller (SQTOC), a continuous-set model predictive speed controller, described in section 7.3, and a Kalman Filter (KF). The latter takes applied stator voltage $\vec{v}_s$ and the measurements for the stator currents $\vec{i}_s$ and rotor angle θ_r to reconstruct the state feedback x.

The starting point for the design, as in the previous chapters, is the solution of the time-optimal control problem of a simplified system. Unlike the methods developed before, however, the methodology in this case does not involve the construction of a cost function, to be used with the finite-set predictive control algorithm. The design of the controller is instead based on a numerical implementation of the *switching time parametrization method* (STPM), introduced in section 3.3.2. The proposed algorithm takes the form of a non-linear state feedback controller, its output being a reference for an inner speed controller. The only requirement for the later, is that it generates the same behaviour as the STPM on the inner states. This is achieved using the quasi time-optimal speed controller developed in section 7.

In the proposed method, the value for the speed reference is produced, as in a sliding mode controller, according to the position of the state relative to a given curve and takes at any instant, one of its extreme values. In order to avoid chattering around the steady state, once the state has been steered close enough to it, a classical and smoother linear quadratic regulator (LQR) takes over the control task and generates directly a reference for the electrical torque. The weights in the cost function of the LQR and the parameters defining the region in the state space, where the it should take over, are the only tunning parameters introduced by the scheme.

The STPM can be implemented as an iterative, numerical scheme to solve a time-optimal control problem for a single initial condition, in a continuous-time regime. This scheme is characterized by two basic features. The first one is the imposition of a bang-bang structure on the actuation. This is equivalent to the *finite-set* assumption used in the previous chapters, in the sense that it reduces the search space for the optimal actuation. The second feature is the use of simulations or predictions of the behaviour of the system, using different actuation sequences. In this sense the scheme can be regarded as a shooting method.

The proposed algorithm approximates the behaviour the STPM imposes on the system in a discrete-time regime and in a non-iterative way. Its internals do make use of the shooting concept: it simulates the behaviour of the system, but the iterations are avoided using a particular control sequence, which is given at each sampling instant by the position of the state with respect to a curve in the state space, we will call Γ. The later is a characteristic of the system dynamics and is related to its step response. In terms of computational burden, the algorithm requires four evaluations of the system model, two of the discrete-time model and two of a continuous-time model (its step response), besides the calculation of the position of the actual state, with respect to the Γ curve.

9.1. System dynamics approximation using the triple integrator

The dynamics of the TI are described by

$$\frac{dx}{dt} = f(x,u)\,, \tag{9.1}$$

with

$$f(x,u) = \begin{pmatrix} {}^{u}\!/_{\tau_0} \\ {}^{x_0}\!/_{\tau_1} \\ {}^{x_1}\!/_{\tau_2} \end{pmatrix}. \tag{9.2}$$

with $x = [x_0\ x_1\ x_2]^T$.

In section 7.1 it was shown how the dynamics of a double integrator can approximate the dynamic behaviour of the PMSM, considering stator voltages and currents, electrical torque and speed:

$$\frac{d}{dt}\begin{bmatrix} T_e \\ \omega_r \end{bmatrix} = \begin{pmatrix} \frac{3p\psi_m}{2L_q}\left(v_q - \psi_m\omega_r\right) - \frac{r_s}{L_q}T_e \\ \frac{p}{J_m}\left(T_e - T_l\right) \end{pmatrix}. \tag{9.3}$$

The dynamics up to the rotor position are completed with the equation

$$\frac{d\theta_r}{dt} = \omega_r \tag{9.4}$$

Assuming the approximation with the double integrator for 9.3 is valid, the dynamics of the PMSM can be then approximated with the triple integrator in (9.2) with

$$\begin{aligned} u &= v_q \\ x_0 &= T_e & \tau_0 &= \frac{2L_q}{3p\psi_m} \\ x_1 &= \omega_r & \tau_1 &= \frac{J_m}{p} \\ x_2 &= \theta_r & \tau_2 &= 1\,, \end{aligned} \tag{9.5}$$

In the following sections and for the sake of simplicity, the target for the control system is assumed to be the origin. To implement the controller for the PMSM, the origin is translated to

$$\begin{bmatrix} x_0^* \\ x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} T_l \\ 0 \\ \theta_r^* \end{bmatrix}. \tag{9.6}$$

and the boundaries for x_0 and x_1 are adjusted accordingly. If the constraints for ω_r and T_e are given by

$$|\omega_r| \leq \hat{\omega}_r \qquad\qquad |T_e| \leq \hat{T}_e. \tag{9.7}$$

the constraints for x_0 are then given by:

$$\check{x}_0 \leq x_0 \leq \hat{x}_0\,, \tag{9.8}$$

with

$$\hat{x}_0 = \hat{T}_e - T_l \tag{9.9}$$

$$\check{x}_0 = -\hat{T}_e - T_l \tag{9.10}$$

In the next sections a time-optimal control algorithm for the triple integrator is developed and later, the approximation of the motor dynamics, presented in this section, is used to apply this algorithm to the rotor position control of the PMSM.

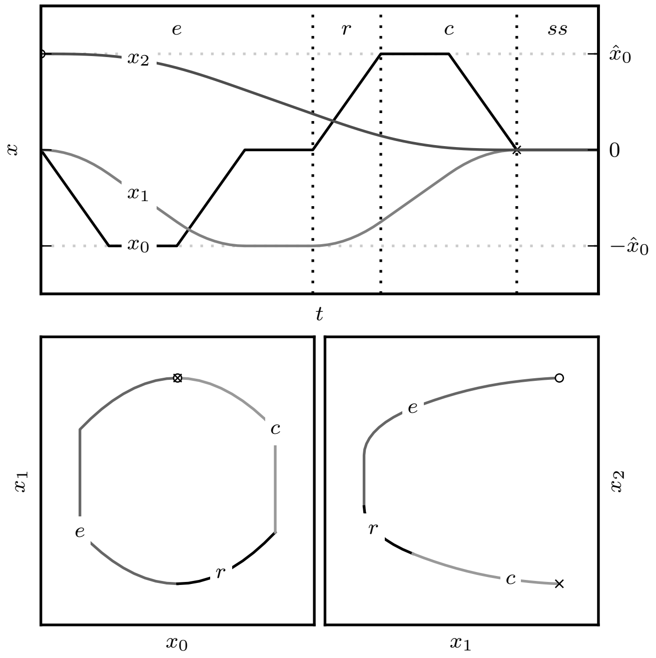

Figure 9.2.: Time optimal trajectory for the triple integrator for an arbitrary i.c.. The plot on the top presents the system state in time. The plots in the bottom present the system state trajectories in the x_0x_1 and x_1x_2 planes, respectively. The control constraints are only represented in the first plot ($\hat{x}_0$). The i.c. and target for state x_2 are marked with the symbols ◯ and ×. In this case the constraint for state x_0 is the same as the constraint for state x_1, i.e. $\hat{x}_0 = \hat{x}_1$. The letters e, r, c and ss stand for *expansion*, *rollback*, *contraction* and steady state. These terms refer to the kinetic energy being transfered to the rotor.

9.2. Time-optimal control for the triple integrator

Our goal is to develop a time-optimal controller for the system dynamics in (9.2), considering the following constraints

$$|x_1| \leq \hat{x}_1 \qquad \check{x}_0 \leq x_0 \,, \leq \hat{x}_0 \qquad \text{and} \qquad |u| \leq \hat{u}. \tag{9.11}$$

The first approach to achieve this consist in applying the STPM, using a general purpose numerical optimization algorithm. Further details on this method and its numerical implementation are covered in section 3.3.2.

Simulations results using this approach for this problem are presented in figure 9.2. Inspecting this results, the behaviour the TI should display when controlled in minimum-time, can be understood in intuitive terms, keeping in mind that the state of the TI relates to physical variables. In order to correct an error in the position (x_2), the system should be accelerated applying maximum voltage ($\hat{u}$), up to the point where, first the torque (x_0)

reaches its constraint and then the speed (x_1) reaches its constraint. During this time, This is called the *expansion stage* (e), to signify that energy is being transfered to the rotor to take the form of kinetic energy. Then, as the error in the position decreases, the system should be decelerated, again applying maximum voltage, so that all the variables reach their steady state values at the same time. This process is divided in two, the *rollback* and *contraction* stages. During the first one, state x_0 is steered towards its contrary boundary and during the second one, the whole state is steered towards its steady state.

The solution of the minimum-time control problem for this system, as the solution for this problem for the double integrator, divides the state space in regions where the magnitude of the actuation ought to be maximum or zero. This time, however, the boundaries between these regions are surfaces. The description of these boundaries can be very cumbersome, specially given the constraints for the magnitude of the torque and speed (x_0 and x_1). In [8] these surfaces are calculated to develop the proximately time optimal servomechanism, but without considering state constraints. The approach used in this work avoids this by means of a shooting scheme.

9.3. Smoothened time-optimal controller for the triple integrator

The design of the proposed control method is based in two main ideas:

- Bang-bang structure for the actuation. This is, we assume that the value the actuation can take at any time belongs to the set $\{-\hat{u}, 0, \hat{u}\}$, and that changes in the actuation can only occur at the sampling instants, which is a common setting for discrete-time control systems. The search space for the optimal control is then drastically reduced and, it can then be solved by direct evaluation of predictions of the system state using sequences with different combinations of the values in this set. The sequences are configured so that the state follows the pattern described in the last section.
- Exploitation of the geometrical properties displayed by the state trajectories, when steered with maximum actuation $\hat{u}$.

The later is understood in terms of the step response of the system, which can be solved analytically assuming

$$u(t) = \hat{u}\mu(t)\,, \tag{9.12}$$

and an i.c. $x(0)$, yielding to

$$x(t) = S(t, x(0), \hat{d}, \hat{u})\,, \tag{9.13}$$

with

$$S(t, x(0), \hat{d}, \hat{u}) = \begin{pmatrix} \frac{t}{\tau_0}u + x_0 \\ \frac{t^2}{2\tau_1\tau_0}u + \frac{t}{\tau_1}x_0 + x_1 \\ \frac{t^3}{3\,2\tau_2\tau_1\tau_0}u + \frac{t^2}{2\tau_2\tau_1}x_0 + \frac{t}{\tau_2}x_1 + x_2 \end{pmatrix} \tag{9.14}$$

The following section describes a key element of the proposed control method: the Γ curve. After this, the control algorithm is explained alongside technical implementation details.

9.3.1. Γ curve

The Γ curve corresponds to the trajectories described by the state of the TI, when, starting from the origin and letting the time run backwards, maximum actuation is applied ($-\hat{u}$ for Γ^+ and $+\hat{u}$ for Γ^-), up to the point where x_0 reaches its constraint (points x^+ and x^-) and after this $u = 0$. The relevance of this curve lays in the fact that, at some point, the time-optimal trajectory from any point in the state space to the origin has to reach the curve Γ and follow it, before reaching the origin (see Fig. 9.2). It should be noted, tough, that since both time and actuation are considered to be discrete, not every point in this curve can be driven exactly to the origin, not even following curve.

The projection of the curve Γ in the x_1x_2 plane is referred to as Γ_{12} and is given by:

$$x_2 = \Gamma_{12}(x_1) = \\ \begin{cases} \kappa_2^+(x_1 - x_1^+)^2 - \kappa_1^+(x_1 - x_1^+) + x_2^+ & \text{if} \qquad x_1 < x_1^+ \\ -\kappa x_1\sqrt{|x_1|} & \text{if } x_0^+ \leq x_1 \leq x_1^- \\ -\kappa_2^-(x_1 - x_1^-)^2 - \kappa_1^-(x_1 - x_1^-) + x_2^- & \text{if } x_1^- < x_1 \end{cases} \tag{9.15}$$

with

$$\begin{aligned} x_0^+ &= \hat{x}_0 & x_0^- &= \check{x}_0 \\ x_1^+ &= -\tau x_0^{+2} & x_1^- &= \tau(x_0^-)^2 \\ x_2^+ &= \frac{\tau\tau_0}{3\tau_2\hat{u}} x_0^{+3} & x_2^- &= \frac{\tau\tau_0}{3\tau_2\hat{u}}(x_0^-)^3 \\ \kappa_2^+ &= \frac{\tau_1}{2\tau_2 x_0^+} & \kappa_2^- &= \frac{\tau_1}{2\tau_2 x_0^-} \\ \kappa_1^+ &= \frac{\tau_1\tau}{\tau_2 x_0^+} & \kappa_1^- &= -\frac{\tau_1\tau}{\tau_2 x_0^-} \\ \kappa &= \frac{1}{3\tau_2}\sqrt{\frac{2\tau_1\tau_0}{\hat{u}}} \end{aligned}$$

The relative position of the initial state, with respect to Γ_{12} gives the sign of the actuation which accelerates the system to correct errors in x_2 during the *expansion* stage.

$$u_e = -\text{sgn}(x_2 - \Gamma_{12}(x_1)) \tag{9.16}$$

This curve can be understood as the result of applying the backward induction principle: it corresponds to the response of the system (with the time running backwards and starting from steady state) to a step excitation of maximum magnitude $u = \pm\hat{u}$, up to the point where x_0 reaches its constraint, and $u = 0$ from then on.

9.3.2. Control algorithm

The proposed method is an *ad-hoc* discrete-time adaption of the STPM, described in section 3.3.2, with a reduced computational overhead by making use of the known properties of the solutions. Similarly as in the STPM, the core of the proposed method is constituted by simulations, or predictions of the state with a given actuation sequence. In the STPM, the control sequence is defined in terms of time intervals during which $u = \pm\hat{u}$ or $u = 0$ is applied, and which are then minimized subject to state and terminal constraints. In the proposed method, the position of the i.c. relative to the Γ curve defines the actuation sequence and it is divided in the three stages: *expansion*, *rollback* and *contraction*. The time for the expansion stage is assumed to be one sampling period and it might or might not occur in the following sampling period. The time for the rollback stage is calculated exploiting the geometrical properties of (9.13). No simulation for the *contraction* stage is carried out, since it is already given by the Γ curve.

The discrete-time model for the triple integrator is produced using the method described in 2.4 and takes the form

$$x[n+1] = f(x[n], u[n]) \tag{9.17}$$

with

$$f(x,u) = \begin{pmatrix} x_0 + \frac{h}{\tau_0}u \\ x_1 + \frac{h}{\tau_1}x_0 + \frac{h^2}{2\tau_1\tau_0}u \\ x_2 + \frac{h}{\tau_2}x_1 + \frac{h^2}{2\tau_2\tau_1}x_0 + \frac{h^3}{6\tau_2\tau_1\tau_0}u \end{pmatrix} \tag{9.18}$$

The expansion stage is simulated with a prediction for the state from the i.c. x_i up to point x_e with $u = u_e$ (see Fig. 9.3).

Starting from x_e, a long term prediction up to point x_r is carried out, to account for the *rollback* stage, using th step response of the system (see eq. (9.13)). The sign of the actuation for this stage is calculated using the position of x_e relative to the Γ curve in the x_0x_1 plane:

$$u_r = \begin{cases} -\mathrm{sgn}(x_0)\hat{u} & \text{if } x_0 > \hat{x}_0 \text{ or } x_0 < \check{x}_0 \\ -\mathrm{sgn}(x_1 + \mathrm{sgn}(x_0)\tau x_0^2)\hat{u} & \text{if } \check{x}_0 \le x_0 \le \hat{x}_0 \end{cases} \tag{9.19}$$

with $\tau = \frac{\tau_0}{2\tau_1\hat{u}}$. The boundary in this case is the switching curve defined for the time-optimal control problem for the double integrator (DI) (see Sec. 5), which is the same curve as the projection of Γ in the x_0x_1 plane, Γ_{01}.

The time span for the rollback stage t_r is calculated exploiting the geometry of the trajectories described by the state in the x_0x_1: it is given by the intersection the parabola generated by the step response of the system with $u = u_r$ and Γ_{01}, which is described by

$$x_1 = -\mathrm{sgn}(x_0)\tau x_0^2 \tag{9.20}$$

when $\check{x}_0 \le x_0 \le \hat{x}_0$, and the vertical lines $x_0 = \check{x}_0$ and $x_0 = \hat{x}_0$, otherwise. With this

$$t_r = \mathrm{sgn}(u_r)\frac{\tau_0}{\hat{u}}(\tilde{x}_0 - x_0) \tag{9.21}$$

where $\tilde{x}_0$ represents the point where the state trajectory and Γ_{01} cross

$$\tilde{x}_0 = \begin{cases} \hat{x}_0 & \text{if} \quad \operatorname{sgn}(u_r)\sqrt{\frac{|\tilde{x}_1|}{2\tau}} < \hat{x}_0 \\ \operatorname{sgn}(u_r)\sqrt{\frac{|\tilde{x}_1|}{2\tau}} & \text{if } \hat{x}_0 \leq \operatorname{sgn}(u_r)\sqrt{\frac{|\tilde{x}_1|}{2\tau}} \leq \check{x}_0 \\ \check{x}_0 & \text{if } \check{x}_0 < \operatorname{sgn}(u_r)\sqrt{\frac{|\tilde{x}_1|}{2\tau}} \end{cases} \tag{9.22}$$

and $\tilde{x}_1$ is the x_1 component of the point where the state trajectory crosses the line $x_0 = 0$

$$\tilde{x}_1 = x_1 - \operatorname{sgn}(u_r)\tau x_0^2 \,. \tag{9.23}$$

The control decision is taken upon the position of x_r relative to Γ in the x_1x_2 plane, compared to the position of the i.c.: if the state does not cross the Γ_{12} curve, the kinetic energy in accumulated in the rotor can still be expanded before proceeding to the rollback stage. On the other hand, if the state goes beyond the Γ_{12} curve, x_2 will necessarily have to show overshoot before reaching its reference and therefore, in this situation no expansion stage should take place; the state should proceed to the rollback stage and be driven towards Γ.

The control decision takes the form of a reference for an inner controller for x_1: if it is decided to proceed with the *expansion*, the reference for x_1 takes one its boundaries: $\hat{x}_1$ or $-\check{x}_1$. In this way, the constraint for x_1 is taken into account. On the other hand, if the decision is to *rollback*, the reference is set $x_1^* = 0$.

The constraint on the magnitude of x_0 is taken into account considering that both the expansion and rollback stages can only develop up to a point where the constraint in x_0 is upheld. To implement this, whenever the expansion stage prediction violates the constraint with $u = \pm\hat{u}$, the prediction up to point x_e is instead simulated assuming that the actuation takes the exact value that steers the electrical torque (x_0) towards its constraint. This violates the bang-bang structure assumption, but it is necessary to ensure the satisfaction of the constraint and, at the same time, that the whole range of x_0 will be utilized. The constraint is taken into account in the rollback stage in the calculations for t_r.

The proposed controller is implemented as a sampled system and thus is only allowed to switch the actuation at the sampling instants. This is not enforced in the STPM, consequently, the time-optimal behaviour described in the last section requires that switches occur at arbitrary instants. In the SQTOC, when the optimal switching time falls between the current and the next sampling time, the x_r prediction crosses the Γ, but proceeding directly to the *rollback* stage would fall short. To solve this and approximate the behaviour generated by the STPM, two new predictions are introduced, with $u = 0$ and $u = -u_e$ for the *expansion* stage. Then, if the prediction assuming $u = u_e$, x_r^+ crosses Γ_{12}, but the prediction assuming $u = 0$, x_r^0 does not, the actuation is calculated as a linear combination of both u_e and 0: the linear combination between x_r^+ and x_r^0 that exactly hits Γ_{12}. An equivalent procedure is carried out when the current state is near the Γ_{12} curve and the predictions for the expansion stage cross it. In this situation the linear combination is calculated between the prediction that crosses Γ_{12} and the prediction that does not.

Finally, there are regions in the state space where $u_e = u_r$ (see eqs. (9.16) and (9.19)), i.e., no switch of the actuation occurs between the expansion and the rollback stages. This means that x_1 is increasing the error in x_2 and u_e must be applied to revert this situation.

9.3.3. Smoothening near the steady state

Just as in a sliding mode controller [51], the reference for x_1 (ω_r) produced by the algorithm described above depends on the position of the state (in proposed scheme, the predicted state) relative to a switching curve, and it is always one of it extremes values. This naturally results in chattering around the switching curve and poor steady state performance. To avoid this, a classical and smoother LQR is introduced to take over the control the task, when the state is near its desired steady state. This controller is implemented to produce a reference for x_0 (T_e). The tuning of the LQR and the region where it should take over were done empirically through simulations. The main goal for the tuning was to minimize noticeable effects during the transitions from one controller to the other.

9.3.4. Inner Controller

The design of the SQTOC for this system is independent from the inner controllers, but requires them to generate the same behaviour as the STPM imposes on all the states, this is, the behavior produced by a bang-bang controller. This is achieved using the speed and torque controllers developed in sections 7.3 and 4.3.

9.3.5. Experimental results

The proposed controller was validated in the experimental setup described in 1.1, with a sampling period $h = 46.1$µs.

Figures 9.4 and 9.5 present experimental results, where the proposed control system responds to a step change in the angle reference. In both figures it is possible to observe how the proposed controller effectively reproduces the expected time-optimal behaviour, calculated applying the STPM on a triple integrator. Towards the end of the test, however, between $t = 0.05$ and 0.06s, chattering is observed, although it is only noticeable in the plot of the electrical torque T_e and is more evident in the plot at the right side in Fig. 9.5. During this time the SQTOC gives an alternating reference for the rotor speed, in order to track the expected behaviour in the ω_r θ_r plane, this is, the Γ curve, and as a result, the inner controller must decrease the magnitude of the electrical torque, not using the full range for the torque, given by its constraints. This means that the expected time-optimal behaviour calculated using the triple integrator approximation is actually slower than what the real motor can do.

These effects are even more noticeable in Fig. 9.6, which depicts the response of the system to an step load torque impact. In the second plot the load torque T_l, estimated by the Kalman filter, is depicted with thick gray line. Notice that this estimation is relatively slow, compared to the response of the controller.

These effects occur presumably due to neglected dynamics, such as friction and the interaction between the Kalman filter and the controller.

Finally, slightly after $t = 0.06$s if Fig. 9.4 and slightly before $t = 0.05$s in 9.6, a spike can be observed in the electrical torque. At this point the internal LQR takes over at steers the system smoothly towards its desired steady state.

In a first approach these undesired effect could be reduced introducing an smoother LQR controller to act not only when the state is near its target, but also when the state lays near the switching curve. In this sense the control goal would set a hierarchy to make the system, for example, follow the switching curve defined in the $\omega_r\,\theta_r$ plane, regardless of the distance of T_e and ω_r of the switching curve defined in the $T_e\,\omega_r$ plane.

In the next chapter, the same ideas developed in this chapter are used to devise a quasi-time-optimal controller for the torsional torque in a two-mass system.

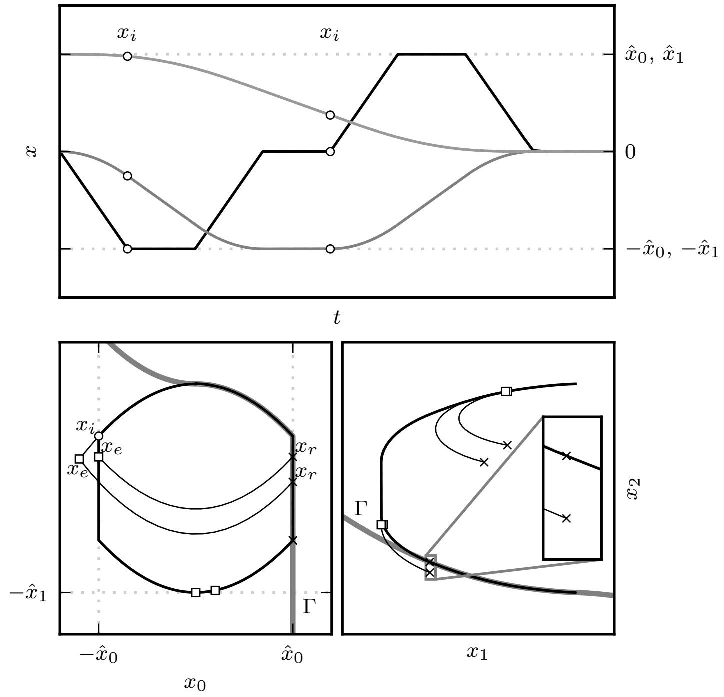

Figure 9.3.: Approximation using the SQTOC method. The i.c. is represented by x_i. x_e (□) is the prediction for the *expansion* stage. Two predictions are shown: using $u = u_e$ (see eq. (9.16)) and $u = 0$. Points x_r (×) pertain to the outcome of the long range predictions, with $u = -u_e$, representing the *rollback* stage. The time frame for this prediction is calculated from θ_r. The rays subtending θ_r intersect at $\left(-\frac{k_d}{\omega_c}d, -\frac{k_u}{\omega_c}u_e\right)$. The sign for u_e is calculated from the position of x_i with respect to the Γ curve (see sec. 9.3.1) in the x_2x_3 plane. The control decision is taken upon the position of x_r, with respect to the Γ curve in the x_2x_3 plane. In first case both predictions, with $u = u_e$ and $u = 0$ stay on the same side as the i.c. with respect to Γ. In this case the controller decides to go on with the expansion stage and applies $u = u_e$. In the second case (in the zoom box), the prediction with $u = u_e$ crosses Γ, whereas the prediction with $u = 0$ does not. In this case $u = 0$ is applied.

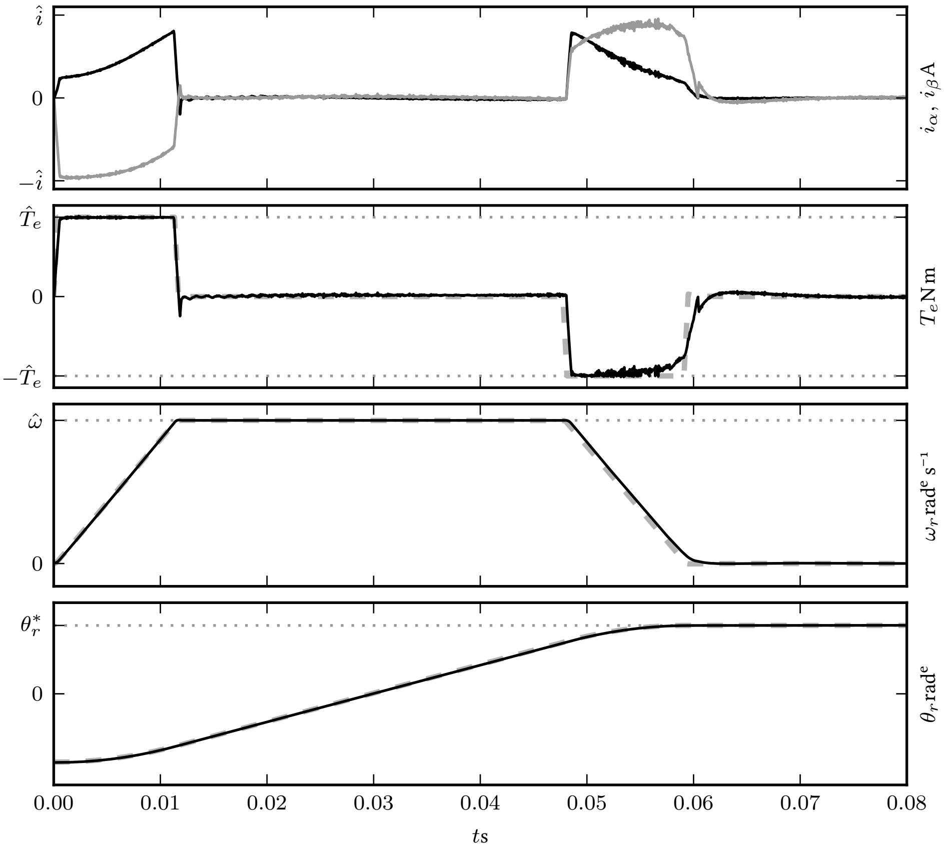

Figure 9.4.: Experimental results: quasi-time-optimal control of the rotor position for the PMSM. In this test the control system reacts to a step change in the rotor angle reference θ_r: at $t = 0$ the rotor is at rest at $\theta_r = -3.0$rad^e, and the reference changes to $\theta_r = 3.0$rad^e. The constraints for the stator currents, electrical torque and rotor speed are set at $\hat{i}_d = 10.0$A, $\hat{T}_e = 9.58$N m and $\hat{\omega}_r = 2\pi 120$rad^e s^{-1}. The light gray dashed line represents the expected time-optimal behaviour in a triple integrator with the parameters in 9.5, for the aforementioned initial conditions, and it was calculated using the STPM method.

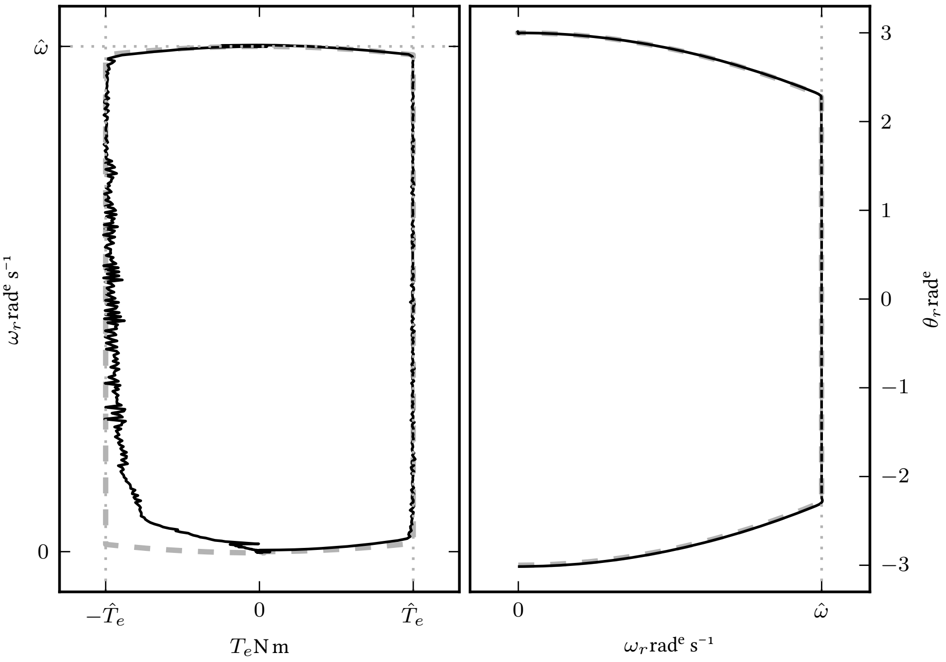

Figure 9.5.: Experimental results: quasi-time-optimal control of the rotor position for the PMSM. The constraints for the stator currents, electrical torque and rotor speed are set at $\hat{i}_d = 10.0\mathrm{A}$, $\hat{T}_e = 9.58\mathrm{N\,m}$ and $\hat{\omega}_r = 2\pi 120 \mathrm{rad^e\,s^{-1}}$. These plots represent the trajectories for the system state presented in time in Fig. 9.4, in the $T_e\,\omega_r$ and $\omega_r\,\theta_r$ planes. Again, the light gray dashed line represents the expected time-optimal behaviour.

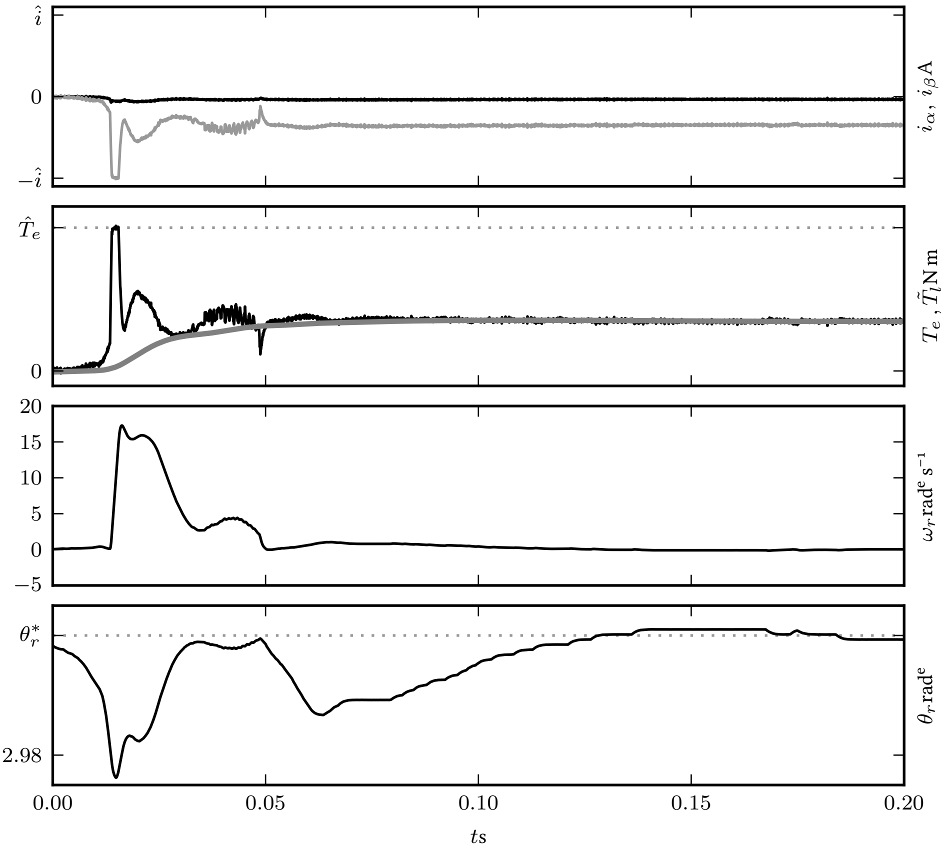

Figure 9.6.: Experimental results: transient response of the proposed control system to a step change in the load torque T_l. In the second plot, the value for the load torque estimated by the Kalman filter, is depicted with a thick gray line.

10. Two-mass system

In this chapter, the control scheme developed in the last chapter is used to devise a feasible algorithm to time-optimally control the two-mass-system (TMS). The goal is to minimize the use of off-line computations and tunning by relying on the system model and speed the closed loop dynamics up to its physical limits.

The system at hand is depicted in Fig. 10.1. It consists of a PMSM, driving a load through a flexible shaft, i.e. a TMS. The voltage for the PMSM is generated by a two-level VSI (VSI).

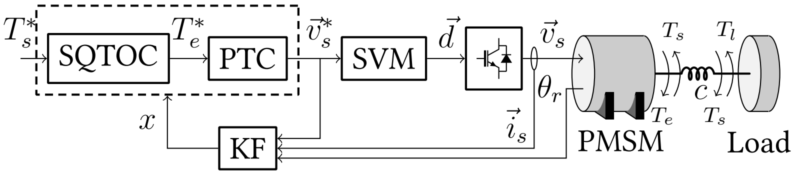

Figure 10.1.: The load is driven by a PMSM through a flexible shaft, with stiffness constant c. The PMSM is driven by a two-level voltage source inverter. The control algorithm consists of the Smoothened Quasi-Time-Optimal Controller (SQTOC), a continuous-set model predictive torque controller, described in section 4.3, and a Kalman Filter (KF). The latter takes applied stator voltage $\vec{v}_s$ and the measurements for the stator currents $\vec{i}_s$ and rotor angle θ_r to reconstruct the state feedback x.

The TMS is a convenient model for many mechanical systems, ranging from wind mills to laser positioning systems. In general, any drive where mechanical design or control specifications weaken the assumption of an infinitely stiff shaft.

The challenge posed by this system is clearly understood by looking at its frequency response (see Fig. 10.2): exciting the resonant mode leads to oscillations that could destroy the shaft. The plot of the experimental data shows clearly, that the actuators and sensors are capable of exciting and identifying the resonant mode.

The control of the two-mass system has been approached from different angles, ranging from classical techniques for non-linear systems [52], sliding mode controllers [53, 54], adaptive control [55, 30] to even fuzzy control [56]. In [57] Thomsen makes a comparison between classic state-feedback controllers and model predictive control and in [58, 59] the author of this theses presented naive approaches using the *finite-set model predictive control* algorithm. In all these works the two main recurring issues are tunning and taking constraints into account. The first one is usually tackled using frequency-domain criteria. The second issue is either ignored or dodged with patches (anti-windup). The approach

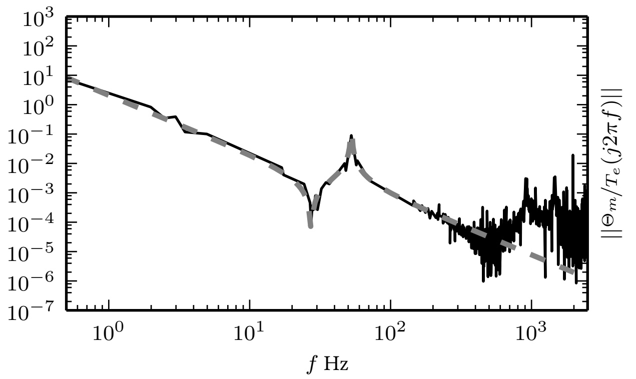

Figure 10.2.: Magnitude of the transfer function between the electrical torque developed by the PMSM T_e and the measured angle θ_r. The black curve was obtained in the laboratory using a torque controller with a pseudo random binary signal as reference. The dashed grey line represents the transfer function of the model with fitted parameters (see Table 10.1). The resonant peak occurs at $f_c = 52.7[Hz]$.

presented in here takes constraints explicitly into account and, for the most part, tunning is not necessary: the directing criteria is time-optimality.

The goal for the scheme developed in this chapter is to control the torque transmitted from the PMSM to the load through torsion of the shaft T_s, to track a reference in minimum time. The controller should take into consideration constraints on the actuation, this is, the voltage synthesized by the VSI, and on the magnitude of the stator currents, which implies a constraint on the magnitude of the torque developed by the PMSM:

$$||\vec{i}_s|| < \hat{i}_s \quad \Rightarrow \quad |T_e| < \hat{T}_e \tag{10.1}$$

The proposed control algorithm follows the same design as the one used in the later chapter: It is based on a simplification of a numerical implementation of the switching time parametrization method (STPM), introduced in section 3.3.2. Time-optimal behaviour is imposed, when the state is away from its target and near the steady state, a smoother LQR takes over to avoid chattering.

The resulting control scheme (see Fig. 10.1) takes the form of a state feedback (it has no internal states) with a cascaded structure. The proposed method generates a reference for the electrical torque, and an inner predictive torque controller generates the stator voltage to be applied to the motor. The internal workings of both controllers are, despite the cascaded structure, closely interwoven: the outer controller does consider the fundamental properties of the whole system, as well as the effect of the inner controller, whereas the latter allows to make assumptions that simplify the way the outer controller takes into account the dynamics of the inner loop.

As is common and desirable in normal drive applications, direct measurements for every component of the system state are not available in the test-bench, which includes only stator currents (LEM) and rotor angle (incremental encoder) measurements. In order to keep the design of the controller independent from the observation problem, a Kalman filter is used to estimate the state.

The performance of the proposed controller is verified with experimental results. They include tests using an outer PI controller for the load speed ω_l, to generate the reference for T_s. The proposed controller enables load speed control with little bandwidth compromise.

10.1. System Model

10.1.1. Drive Model

The dynamics of the drive, including the PMSM drive and the TMS, are described by the following dynamical system [29, 60]:

$$\frac{d}{dt}\begin{bmatrix} \vec{i}_s \\ \omega_m \\ \theta_r \\ T_s \\ \omega_l \end{bmatrix} = \begin{bmatrix} L_s^{-1}\left(\vec{v}_s - r_s\vec{i}_s - \dfrac{\partial L_s}{\partial \theta_r}\omega_m\vec{i}_s - J\omega_m\vec{\psi}\right) \\ \frac{p}{J_m}\left(T_e - T_s\right) \\ \omega_m \\ \frac{c}{p}\left(\omega_m - \omega_l\right) \\ \frac{p}{J_l}\left(T_s - T_l\right) \end{bmatrix} \tag{10.2}$$

Most elements of this model were introduced in section 2.2, particularly, the stator current dynamics and the mechanism behind electrical torque production.

The new states, introduced to model the dynamics of the two-mass system, are the torsional torque T_s and the angular speed of the load ω_l.

For convenience ω_l, is expressed in electrical radians per second.

The torsional torque is produced by torsion of the shaft. A first approximation to model this phenomenon is the use of Hooke's law, which leads to

$$T_s = \frac{c}{p}\left(\theta_r - \theta_l\right) , \tag{10.3}$$

this is, the torsional torque is proportional to the deflection between the rotor of the motor and the load, θ_r and θ_l respectively. In (10.3) both are expressed in electrical radians per second, for convenience. Constant c is the stiffness constant of the shaft. Its value is found with a model fitting scheme (see [60]). The parameter J_l represents the inertia of the load.

The coupling between the mechanical variables gives rise to resonant behaviour (see Fig. 10.2) with resonant angular frequency:

$$\omega_c = \sqrt{c\frac{J_m + J_l}{J_m J_l}}. \tag{10.4}$$

The mechanical parameters of the setup used for experimental verification are presented in table 10.1.

Table 10.1.: Parameters of the two-mass-system

$J_m = 3.265 \times 10^{-3}\,\mathrm{kg\,m^2}$
$J_l = 8.815 \times 10^{-3}\,\mathrm{kg\,m^2}$
$c = 260.657\,\mathrm{N\,m\,rad^{-1}}$

10.1.2. Abstract Description

In order to focus the analysis on the dynamics that are relevant to the desired behaviour, the above presented model is distilled through a set of assumptions, which we discuss in what follows.

The first assumption is that the dynamics between the stator voltage and the electrical torque can be approximated using a simple integrator. In section 7.1 it was shown how the step response of the torque matches the step response of an integrator during the transient. That analysis in that case did not include the dynamics of the TMS, nevertheless, these are of higher order: T_s first appears in the second derivative of T_e. Moreover, one period of the resonant behaviour is about 50 times longer than the time needed to steer T_e from 0 N m to its constraints, in this case also given by $\hat{T}_e = 10.24\,\mathrm{N\,m}$. With this, we can assume this approximation to be precise enough and we can synthesize the relevant information of the first two equations of the system model (10.2) as

$$\frac{dT_e}{dt} = \frac{3p\psi_m}{2L_q} v_q \tag{10.5}$$

or, taking $x_0 = T_e$:

$$\frac{dx_0}{dt} = \frac{u}{\tau_0}. \tag{10.6}$$

The state x_0 is relevant for the control design insofar it stores the information regarding the satisfaction of the constraint on the stator currents, i.e., the constraint on T_e.

The next assumption is that the actuation belongs to a finite set:

$$u \in \{-\hat{u}, 0, \hat{u}\}. \tag{10.7}$$

This assumption is leveraged to reduce the trajectories the state could follow to a manageable set, i.e. shrink the search space for the optimal actuation. This assumption has physical and theoretical meanings: on the one hand the actuator (the VSI) has a constrained nature, and on the other, the actuation synthesized by a time-optimal control adopts a *bang-bang* form. This assumption stops serving its purpose near the switching curve and the steady state: in this situation, restricting the actuation to its limit values will produce, very naturally in a discrete-time regime, chattering. In order to avoid this, when the predicted state falls near the Γ curve, the reference for T_e is synthesized using a linear combination of the values predicted using the extreme values of the actuation and, near the steady state, the control is passed over to a classical LQR, which produces torque references in the continuous range $[-\hat{T}_e, \hat{T}_e]$.

The maximum magnitude of the actuation, as in the latter chapters, is assumed to be

$$\hat{u} = 0.9 \cdot \frac{\sqrt{3}}{3} v_{dc}, \tag{10.8}$$

which is the maximum amplitude the two-level VSI can generate, when synthesising perfectly sinusoidal voltages, multiplied by the factor 0.9, to avoid ringing noise, produced by switching in the VSI, when modulating relatively high voltages.

Finally, considering the following change of variables

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} T_e \\ k_1 \left(T_e - \sigma T_s\right) \\ \omega_m - \omega_l \\ T_s \end{bmatrix}, \qquad d = T_l, \tag{10.9}$$

with

$$\sigma = \left(1 + \frac{J_m}{J_l}\right) \quad \text{and} \quad k_1 = \frac{p}{J_m \omega_c}, \tag{10.10}$$

an abstract model for the drive, including the mechanical dynamics can be written:

$$\frac{dx}{dt} = \begin{pmatrix} {}^{u}\!/\!{}_{\tau_0} \\ -\omega_c x_2 + k_u u \\ \omega_c x_1 + k_d d \\ {}^{x_2}\!/\!{}_{\tau_3} \end{pmatrix}, \tag{10.11}$$

where

$$\tau_0 = \frac{2L_q}{3p\psi_m}, \qquad k_u = \frac{3p^2\psi_m}{2L_q J_m \omega_c}, \tag{10.12}$$

$$\tau_3 = \frac{p}{c}, \qquad k_d = \frac{p}{J_l}. \tag{10.13}$$

Equations for states x_1 and x_2 correspond to the description of an harmonic oscillator with two sources, u and d, and angular frequency ω_c. As it will be discussed in the following section, this model shall convenient to understand, formulate and solve the minimum-time control problem.

In the following we will refer to the system in (10.11) as the abstract two-mass system.

The *total energy* in this harmonic oscillator is related to the distance between the state coordinates in the x_1x_2 plane and the steady state point. Component x_1 pertains to the potential energy stored in the *shaft spring*, whereas component x_2 pertains to the kinetic energy in the motor and load inertiae. In the following sections the words *expansion* and *contraction* will be used in relation to the *kinetic energy* of the oscillator, meaning that the state is driven *away from* or *towards* the steady state point in the x_2 axis, respectively.

Steady State

The steady state and the control target for system (10.11) are reached at point

$$x^{ss} = \begin{bmatrix} \sigma x_3^* - (\sigma - 1)d \\ -\frac{k_d}{\omega_c} d \\ 0 \\ x_3^* \end{bmatrix}, \tag{10.14}$$

where x_3^* is the torsional torque reference. In terms of the drive model this means that in steady state, it holds

$$T_e^{ss} = \left(1 + \frac{J_m}{J_l}\right) T_s^* - \frac{J_m}{J_l} T_l, \tag{10.15}$$

i.e. in steady state T_e should be proportional to the reference T_s^* and also compensate for the load torque T_l.

Continuous time model

The step response of the system described by eq. (10.11) can be solved analytically assuming

$$u(t) = \hat{u}\mu(t) \qquad \text{and} \qquad d(t) = \hat{d}\mu(t) \tag{10.16}$$

and an i.c. $x(0)$, yielding to

$$x(t) = S(t, x(0), \hat{d}, \hat{u}), \tag{10.17}$$

with

$$S(t, x(0), \hat{d}, \hat{u}) = \begin{pmatrix} \frac{\hat{u}}{\tau_0} t + x_0(0) \\ A\cos(\omega_c t) + B\sin(\omega_c t) - \frac{k_d}{\omega_c}\hat{d} \\ A\sin(\omega_c t) - B\cos(\omega_c t) + \frac{k_u}{\omega_c}\hat{u} \\ \frac{1}{\sigma}\left(x_0(t) - \frac{1}{k_1}x_1(t)\right) \end{pmatrix}, \tag{10.18}$$

where

$$A = \frac{k_d}{\omega_c}\hat{d} + x_1(0) \qquad \text{and} \qquad B = \frac{k_u}{\omega_c}\hat{u} - x_2(0). \tag{10.19}$$

In the x_1x_2 plane, the trajectories developed by $S(\cdot)$ follow circular paths with center

$$(x_1, x_2) = \left(-\frac{k_d}{\omega_c}\hat{d}, \frac{k_u}{\omega_c}\hat{u}\right) \tag{10.20}$$

and angular frequency ω_c. This property is later exploited to invert part of the minimum time problem in the x_1x_2 plane: the time required to drive the state between two arbitrary points with $u = \pm\hat{u}$ corresponds to the angle subtended by the arc between the two points from the center of the corresponding circle. This model will also be useful to calculate the Γ curve (see sec. 10.3.1), which is the trajectory that the state should approach and follow in order to reach the target.

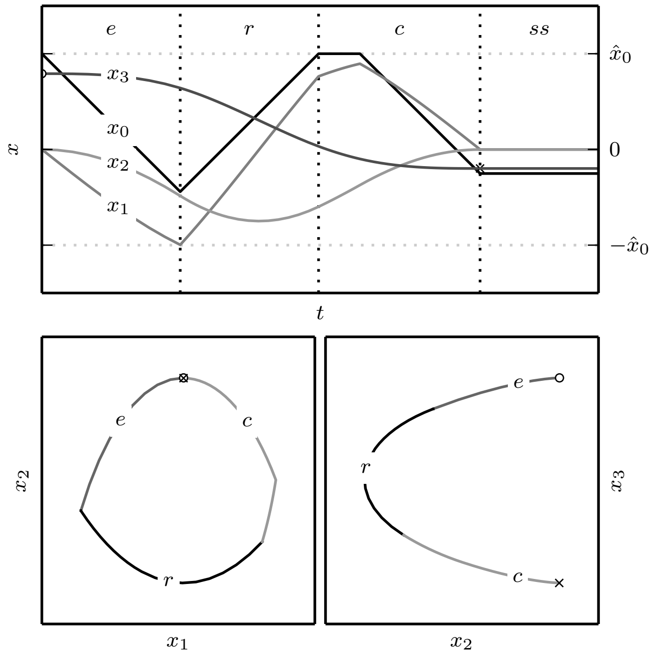

Figure 10.3.: Time optimal trajectory for system (10.11) for an arbitrary i.c.. The plot on the top presents the system state in time. The plots in the bottom present the system state trajectories in the x_1x_2 and x_2x_3 planes, respectively. The control constraints are only represented in the first plot ($\hat{x}_0$). The i.c. and target for state x_3 are marked with the symbols ○ and ×. e, r, c and ss stand for *expansion*, *rollback*, *contraction* and steady state.

10.2. Time-optimal control for the abstract two-mass system

Our goal is to develop a time-optimal controller for the system dynamics in (10.11), considering a constraint on the magnitudes of state x_0 and the actuation:

$$|x_0| \leq \hat{x}_1 \qquad\qquad |u| \leq \hat{u}\,. \tag{10.21}$$

As in the last chapter, the first approach to achieve this goal consist in applying the STPM, using a general purpose numerical optimization algorithm (see Sec. 3.3.2).

Simulations results using this approach for this problem are presented in Fig. 10.3.

In the context of the tow-mass-system, the control goal *to drive T_s towards an arbitrary reference in minimum time*, can be paraphrased as *to store an arbitrary amount of energy in the shaft spring in minimum time*; this perspective can be helpful to understand the systems desired behavior. First, the rotor of the PMSM should be accelerated, so energy is transferred into kinetic energy. As the gap between θ_r and θ_l grows, energy is transferred to the *shaft spring*, where it takes the form of potential energy. The energy is then transformed back to kinetic energy in the load. At some point the kinetic energy in the rotor of the PMSM is more than enough to drive the potential energy in the rotor spring towards its desired level. Then, the surplus should be drawn by decelerating the rotor of the PMSM down to the point where the speeds and accelerations of both the rotor of the PMSM and the load match.

Fig. 10.3 presents plots of the trajectory developed by the state of system (10.11), when controlled to achieve this behavior, against time and in the x_1x_2 and x_2x_3 planes. Labels e, r and c account for three stages characterizing this trajectory. In terms of the model presented in 10.1.2 and the description in the last paragraph, they refer to stages of expansion (e) and contraction (c) of the kinetic energy in the harmonic oscillator x_1x_2, whereas r constitutes a rollback to a point in the state space where the contraction of the kinetic energy can be achieved as fast as the control constraints allow. For the development of the proposed method it is assumed that the system state must follow this pattern. This allows to reduce the optimal control problem to the evaluation of predictions of the system state, using only the set of sequences or configurations of the actuation that make the state follow this pattern.

10.3. Smoothened quasi-time-optimal control for the abstract two-mass system

The control algorithm developed in this section is equivalent to the algorithm developed for the smoothened quasi-time-optimal control of the triple integrator in last chapter. In this sense it is based on the same ideas:

- The actuation sequence has a bang-bang structure and changes can only occur at the sampling instants. With this the search space for the optimal control is reduced. The control sequences are configured so that the state follows the pattern described in the last section.

- Exploitation of the geometrical properties displayed by the state trajectories, when steered with maximum actuation $\hat{u}$.

The following section describes the Γ curve for the case at hand. After this, the control algorithm is explained alongside technical implementation details.

10.3.1. Γ curve

The Γ curve corresponds to the *contraction* stage, described in last section (stage c in Fig. 10.3). If the current state of the system is located at any point along the Γ curve, it will follow it and reach the target in minimum time by applying maximum actuation. Consequently, and according to the DPP, for any i.c. not in this curve, the state should be first taken towards it (the rollback stage: r in Fig. 10.3), before it can be driven towards the target.

The position of the i.c. with respect to Γ determines the sign of the actuation for the expansion stage:

$$u_e = -\text{sgn}\left(\Gamma_{23}(x_2) - x_3\right)\hat{u}, \tag{10.22}$$

where Γ_{23} is the projection in the x_2x_3 plane of the Γ curve.

In order to better understand the form that Γ_{23} takes, some magnitudes relative to it are first introduced, in terms of the parameters of the system model (10.11). The quantity

$$r = \frac{k_u}{\omega_c}\hat{u}\,, \tag{10.23}$$

is the radius of the circular trajectory followed by the state in the x_1x_2 plane, starting from the equilibrium point (10.14), with center at $(x_1, x_2) = \left(-\frac{k_d}{\omega_c}d, \mp\frac{k_u}{\omega_c}\hat{u}\right)$, when maximum actuation ($u = \pm\hat{u}$) is applied, and the time is let to run backwards. Under these conditions and in the same plane, the trajectory will develop an arch of length $r\theta^{\pm}$, before x_0 reaches its constraint $x_0 = \pm\hat{x}_0$, with

$$\theta^{\pm} = 2\pi\frac{\tau_0}{\hat{u}}\left(\pm\hat{x}_0 - \sigma x_3^*\right). \tag{10.24}$$

Once x_0 reaches its constraint, the state would have reached one of these two points:

$$x^{\pm} = \begin{pmatrix} \pm\hat{x}_0 \\ k_1\left(\pm\hat{x}_0 - \sigma x_3^{\pm}\right) \\ \pm r\left(\cos(\theta^{\pm}) - 1\right) \\ \pm\frac{r}{\tau_3\omega_c}\left(\theta^{\pm} - \sin(\theta^{\pm})\right) + x_3^* \end{pmatrix}. \tag{10.25}$$

If, from then on, $u = 0$ is applied, x_0 will remain on its constraint, and the state will follow a circular trajectory in the x_1x_2 plane, centered at $(x_1, x_2) = \left(-\frac{k_d}{\omega_c}d, 0\right)$ with radius given by:

$$(r^{\pm})^2 = (x_1^{\pm})^2 + (x_2^{\pm})^2\,. \tag{10.26}$$

With this, and using an *inverse time* version of (10.17), the projection of the Γ curve in the x_2x_3 plane (see Fig. 10.4) can be written as a function of x_2:

$$\Gamma_{23}(x_2) = \begin{cases} x_3^* & \text{if } x_2 = 0 \\ -\text{sgn}(x_2)\frac{r}{\tau_3\omega_c}\left(\theta_\Gamma - \sin(\theta_\Gamma)\right) + x_3^* & \\ \qquad \text{if } x_2^+ \le x_2 \le x_2^- \wedge x_2 \neq 0 & \\ -\text{sgn}(x_2)\left(\frac{\hat{x}_0}{\sigma} - \frac{1}{\tau_3\omega_c}\sqrt{(r^{\{\text{sgn}(x_2)\}})^2 - x_2^2}\right) + \frac{\sigma-1}{\sigma}d & \\ \qquad \text{if } x_2 < x_2^+ \vee x_2^- > x_2 & \end{cases}\,, \tag{10.27}$$

where

$$\theta_\Gamma = \cos^{-1}\left(1 - \frac{|x_2|}{r}\right). \tag{10.28}$$

10.3.2. Control algorithm

The structure of the control algorithm is the same as in the last chapter. The *expansion* and *rollback* stages are simulated using the discrete-time and the continuous-time models of the system respectively. The sign for the actuation in each stage in given by the position of the state with respect to Γ. The time frame for the expansion stage is assumed to be one sampling period and it might or might not occur in the following sampling period. The time for the rollback stage is calculated exploiting the geometrical properties of (10.17). No prediction is carried out for the *contraction* stage, since it is already given by the Γ curve.

The discrete-time model for the abstract two mass system is produced using the method described in 2.4.

The expansion stage is simulated with a prediction for the state from the i.c. x_i up to point x_e with $u = u_e$ (see Fig. 10.4). Starting from x_e, a long term prediction up to point x_r is carried out, to account for the *rollback* stage. The sign of the actuation for this stage is calculated using the position of x_e relative to the Γ curve in the $x_1 x_2$ plane:

$$u_r = \begin{cases} -\hat{u} & \text{if } R^+ \vee \overline{(R^+ \vee R^-)} \wedge (x_1 < -k_d d) \\ \hat{u} & \text{if } R^- \vee \overline{(R^+ \vee R^-)} \wedge (x_1 > -k_d d) \\ 0 & \text{if } x_1 = x_2 = 0 \end{cases}, \tag{10.29}$$

with

$$R^+ := (x_1 + k_d d)^2 + (x_2 + r)^2 \le r^2 \tag{10.30}$$

$$R^- := (x_1 + k_d d)^2 + (x_2 - r)^2 \le r^2 \,. \tag{10.31}$$

The time span for the rollback stage, t_r, is calculated exploiting the geometry of the trajectories described by the state in the $x_1 x_2$ plane: it is given by the angle subtended by the arc between x_e and x_r in the circumference with center $C = \left(-\frac{k_d}{\omega_c} d, -\frac{k_u}{\omega_c} u_e\right)$ and radius $r_r = ||C - (x_{e1}, x_{e2})||$:

$$t_r = \frac{\theta_r}{\omega_c}. \tag{10.32}$$

The point x_r in the $x_1 x_2$ plane is calculated as the intersection of two circumferences, described by the following equations:

$$\left(x_1 + \frac{k_d}{\omega_c} d\right)^2 + \left(x_2 + \frac{k_u}{\omega_c} u_e\right)^2 = r_r^2 \,, \tag{10.33}$$

$$\left(x_1 + \frac{k_d}{\omega_c} d\right)^2 + \left(x_2 - \frac{k_u}{\omega_c} u_e\right)^2 = \left(\frac{k_u}{\omega_c} \hat{u}\right)^2 \,. \tag{10.34}$$

The first circumference represents the trajectory of the state during the *rollback* stage, the second is the trajectory of the state during the *contraction* stage, or the projection of the Γ curve in the $x_1 x_2$ plane (see Sec. 10.3.1).

The control decision is taken upon the position of x_r relative to Γ in the x_2x_3 plane, compared to the position of the i.c.: if the state does not cross the Γ_{23} curve, the kinetic energy in the harmonic oscillator can still be expanded before proceeding to the rollback stage. On the other hand, if the state goes beyond the Γ_{23} curve, x_3 will necessarily have to show overshoot before reaching its reference and therefore, in this situation no expansion stage should take place ;the state should proceed to the rollback stage and be driven towards Γ.

The control decision takes the form of a reference for an inner controller for x_0: if it is decided to proceed with the *expansion*, the reference for x_0 takes the value of its prediction with $u = u_e$. On the other hand, if the decision is to *rollback*, the reference takes the value of the prediction for x_0 with $u = -u_e$.

The constraint on the magnitude of x_0 is taken into account considering that both the expansion and rollback stages can only develop up to a point where the constraint in x_0 is upheld. To implement this, whenever the expansion stage prediction violates the constraint with $u = \pm\hat{u}$, the prediction up to point x_e is instead simulated assuming that the actuation takes the exact value that steers the electrical torque (x_0) towards its constraint. This violates the assumption regarding the bang-bang structure for the actuation, but it is necessary to ensure the satisfaction of the constraint and, at the same time, that the whole range of x_0 will be utilized. On the other hand, whenever the time for the *rollback* stage t_r is longer than the time required to drive x_0 to its constraint, t_r is cut down to the time that takes x_0 exactly to its constraint:

$$t_c = \frac{\tau_0}{u_e}(\text{sgn}(u_e)\hat{x}_0 - x_0) \tag{10.35}$$

The proposed controller is implemented as a sampled system and thus is only allowed to switch the actuation at the sampling instants. This is not enforced in the STPM, consequently, the time-optimal behaviour described in the last section requires that switches occur at arbitrary instants. In the proposed algorithm, when the optimal switching time falls between the current and the next sampling time, the x_r prediction crosses the Γ, but proceeding directly to the *rollback* stage would fall short. To solve this and approximate the behaviour generated by the STPM, two new predictions are introduced, with $u = 0$ and $u = -u_e$ for the *expansion* stage. Then, if the prediction assuming $u = u_e$ (x_r^+) crosses Γ_{23}, but the prediction assuming $u = 0$ (x_r^0) does not, the actuation is calculated as a linear combination of both u_e and 0: the linear combination between x_r^+ and x_r^0 that exactly hits Γ_{23}. An equivalent procedure is carried out when the current state is near the Γ_{23} curve and the predictions for the expansion stage cross it. In this situation the linear combination is calculated between the prediction that crosses Γ_{23} and the prediction that does not.

Finally, there are regions in the state space where $u_e = u_r$ (see eqs. (10.22) and (10.29)), i.e., no switch of the actuation occurs between the expansion and the rollback stages. This means that x_2 is increasing the error in x_3 and u_e must be applied to revert this situation (a reference $x_0^* = \text{sgn}(u_e)\hat{x}_0$ is given to the inner controller).

10.3.3. Smoothening near the steady state

The reference for x_0 (T_e) produced by the algorithm described above depends on the position of the state (in proposed scheme, the predicted state) relative to a switching curve, and it is always the result of applying an extreme value of the control set. This naturally results in chattering around the switching curve and poor steady state performance. To avoid this, a classical and smoother LQR is introduced to take over the control the task, when the state is near its desired steady state. The tuning of the LQR and the region where it should take over were done empirically through simulations. The main goal for the tuning was to minimize noticeable effects during the transitions from one controller to the other.

10.3.4. Inner Controller - Predictive Torque Control

The design of the control algorithm is independent from the inner controller, but requires it to generate the same behaviour as STPM imposes on all the states. This is achieved using the torque controller developed in sections 4.3.

10.3.5. State Estimation

The proposed control method acts as a state feedback and since measurements for all components of the state are not available at the test bench used for experimental verification, a state observer was implemented. For this work, a reduced order extended Kalman filter was used, since it also effectively separates the observation from the control problem (see Appendix A).

10.4. Experimental results

Fig. 10.5 presents a step response of the system using the proposed control method. The reference for T_s goes from approximately 0 N m to 7.0 N m. The controller achieves reference tracking for T_s by forcing T_e towards its constraint for the whole transient. As a result T_s displays a sigmoid behavior, typical of time-optimal controllers.

It is possible to see some overshoot in T_s before it settles at its reference. This can be observed more explicitly in the difference between ω_m and ω_l. This may be the effect of neglecting the *back-emf* to calculate T_e^* in the SQTOC (see eq. (10.5)), since this effect is smaller, when operating at lower speeds. This is more explicit in the first plot in Fig. 10.6, where x_0 represents the electrical torque T_e: when the T_e is steered towards $\hat{T}_e$, its rate of change matches the expected one, on the other hand, when T_e is steered towards $-\hat{T}_e$, its rate of change is slightly slower, since the back-emf opposes to this change. Otherwise, the obtained behaviour is as expected.

Figures 10.6 and 10.7 presents the same results as in Fig. 10.5, in terms of the abstract states introduced in 10.1.2. Experimental measurements are compared with the expected behaviour, calculated using the STPM applied to the abstract two-mass-system. These results show how the controller effectively reproduces the expected behavior. A slight oscillation can observed between t =10.0 ms to 15.0 ms. At this point, the smoother LQR takes over.

Fig. 10.8 presents experimental results using the proposed controller for T_s and an external PI controller for the load speed ω_l. The PI controller was tunned using pole placement for a plant with the form $G(s) = \frac{1}{J_l s}$, to obtain a closed loop natural frequency of approximately 8.67 Hz, this is, only 6 times slower than then resonant frequency of the system. The test consists of a start-up maneuver, where the reference for ω_l goes from 0 to $2\pi 50$rad^{e} s^{-1} (1000rpm mechanical), followed by a speed reversal and a load torque impact.

In Fig. 10.5, at $t \approx$ 16 ms it is also possible to distinguish a small change in T_e. This is caused by the LQR, when it takes over. This effect is more patent in Fig. 10.8 at $t \approx$ 0.4 s and $t \approx$ 0.75 s. Although this behaviour does not cause major problems, it would be desirable to have strategies to match both controllers at the transition boundary, or at least to verify that the LQR action will not take the state out of its region of activity resulting in chattering around the transition frontier.

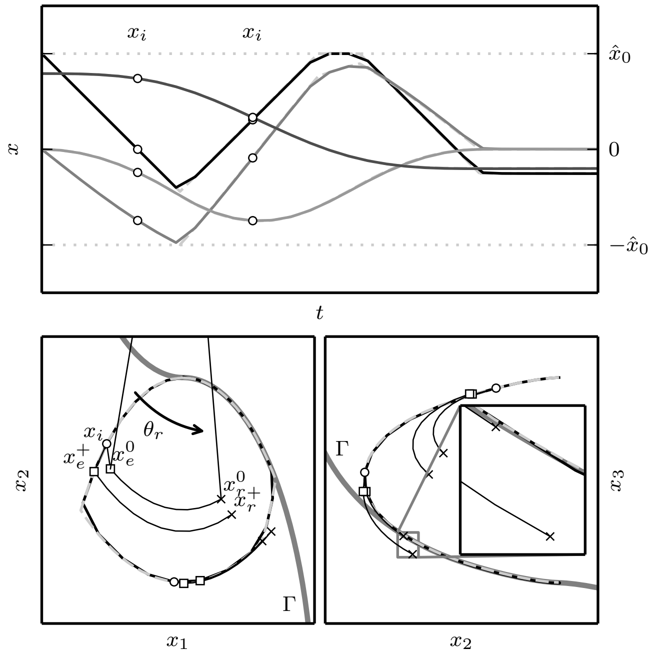

Figure 10.4.: Approximation using the SQTOC method. x_i represents the i.c. for the method. x_e (□) is the prediction for the *expansion* stage. Two predictions are shown: using $u = u_e$ (see eq. (10.22)) and $u = 0$. Points x_r (×) pertain to the outcome of the long range predictions, with $u = -u_e$, representing the *rollback* stage. The time frame for this prediction is calculated from θ_r. The rays subtending θ_r intersect at $\left(-\frac{k_d}{\omega_c}d, -\frac{k_u}{\omega_c}u_e\right)$. The sign for u_e is calculated from the position of x_i with respect to the Γ curve (see sec. 10.3.1) in the x_2x_3 plane. The control decision is taken upon the position of x_r, with respect to the Γ curve in the x_2x_3 plane. In first case both predictions, with $u = u_e$ and $u = 0$ stay on the same side as the i.c. with respect to Γ. In this case the controller decides to go on with the expansion stage and applies $u = u_e$. In the second case (in the zoom box), the prediction with $u = u_e$ crosses Γ, whereas the prediction with $u = 0$ does not. In this case $u = 0$ is applied.

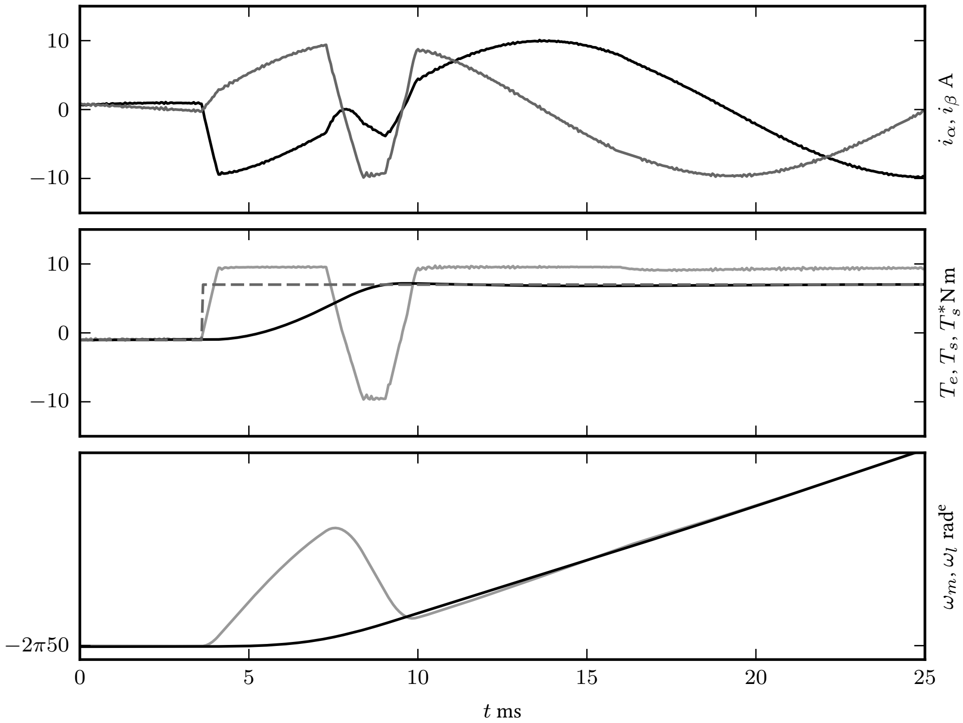

Figure 10.5.: Experimental results: transient for a step torsional torque reference change, using the proposed method. The reference for T_s is represented in the second plot with a gray dashed line. In this plot, the light gray line represents the electrical torque T_e being developed by the PMSM. The third plot represent the speeds of the rotor of the PMSM and the load ω_m and ω_r, respectively. The integral of the difference between them is proportional to the torsional torque.

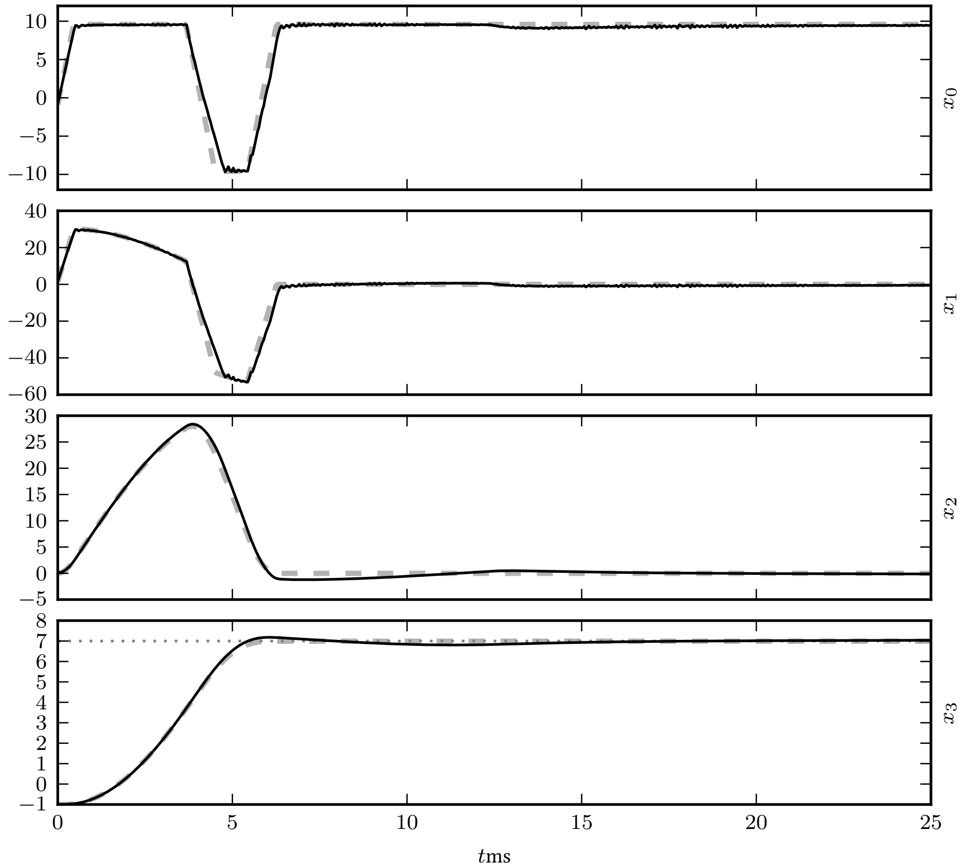

Figure 10.6.: Experimental (solid) and simulation (dashed) results: transient for a step torsional torque reference change, using the proposed method, represented in term of the state of the abstract two-mass-system, introduced in section 10.1.2. Simulation results were obtained using the STPM applied to the abstract two-mass-system, with the parameters of the real drive.

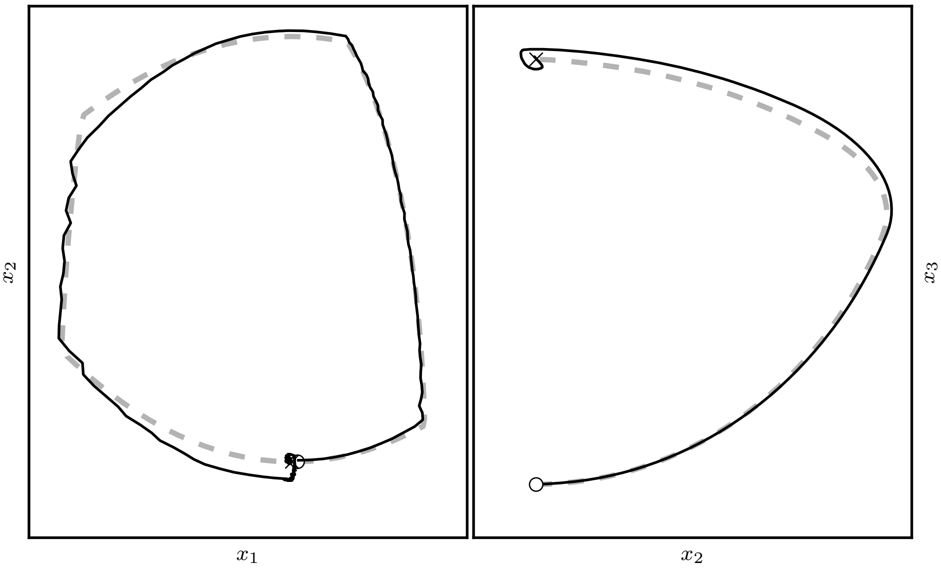

Figure 10.7.: Experimental (solid) and simulation (dashed) results: transient for a step torsional torque reference change, using the proposed method, represented in term of the state of the abstract two-mass-system, in the x_1x_2 and x_2x_3 planes. The initial condition and the target are represented with the ∘ and × symbols.

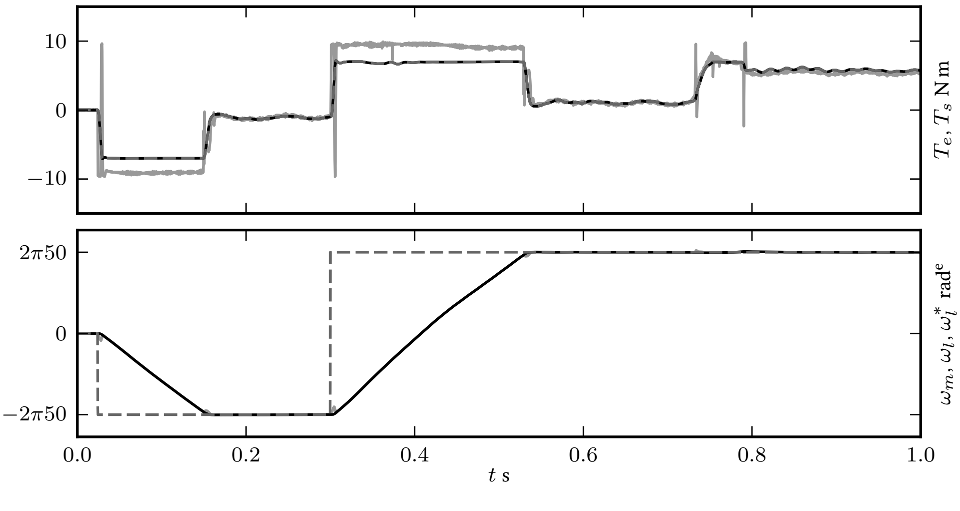

Figure 10.8.: Experimental results: load speed control for the two-mass system. in the first plot the gray line represents T_e and the black line represents T_s. The reference is given by a PI controller. The bandwidth for the speed controller was purposely set beyond the resonance frequency of the system: the inner controller successfully regulates oscillations at that frequency.

11. Conclusions

In this work, the finite-set model predictive control algorithm [13] was taken as starting point and was further developed in two directions.

First, the issues produced by the application of a single switching state by the power converter, during a whole sampling instant, namely:

- irregular switching pattern,
- distributed frequency spectrum
- and the impossibility of achieving control with arbitrary precision,

were tackled in chapter 4, introducing a pulse with modulation scheme. With this, the actuator (the power converter) can be used to approximate voltages in a continuous set. The core ideas in finite-set model predictive control were leveraged to obtain fast dynamics.

The torque controller developed for the PMSM in chapter 4 also introduces a methodology to approach different control goals simultaneously.

The second direction developed in this work was the integration of higher order dynamics in the controller. Chapters 6, 7 and 8 deal with the development of torque and speed controllers for the PMSM and the IM and chapters 9 and 10 with the development of a position controller for a PMSM drive and of the torsional torque in a two-mass system. This was achieved first by introducing specially designed cost functions and second, by adapting existing control methods, based on numerical optimization algorithms, to construct non-linear feedback rules. In all the cases, the performance index used was the time required to steer the state of the system towards a target. With this, a common ground was established to fix the expected dynamic behaviour of the system.

At the core of this methodology is the approximation of the drive dynamics using extremely simplified models. The solution of the time-optimal control problems for these models is not trivial, but is much simpler than solving an optimal control problem considering the whole dynamics of the drive. This thesis verified in each case, that these models were sufficient to roughly account for the desired time-optimal dynamics. The developed control methods are computationally feasible and reproduce the expected behaviour.

In this sense, it is important to remark that, in order to obtain satisfactory results, an appropriate fit of the models was necessary, so that the measured and the expected behaviour match. This led, for example, to use values for the stator inductances, which were smaller than their usual expected values (the *synchronous frequency* inductance), since Eddie currents and iron saturation, which are not considered in the drive models, make the inductance of the motors appear smaller at higher frequencies.

The proposed control methods do not introduce tunning parameters, other than the parameters of smooth controllers used to take over the control task, once the state has approached its desired state behaviour. This was necessary to reduce the undesired steady-state chattering, that the bang-bang part of the proposed control methods produce. For this task, classical controllers were used, adjusted to avoid further chattering around the boundary between the two strategies. Otherwise and for the most part, the proposed control algorithms rely on the system model and its parameters.

11.1. Further work

The work reported in this thesis solved most of the problems laid out at its beginning, namely, fixing the dynamics of higher order systems with feasible control algorithms. It does, however, put more questions forward.

As hinted before, the methods developed in this work do not introduce tunning parameters and the model parameters only change the scale of the problem. This holds primarily, when high switching frequencies are used. In this sense, the pulse width modulation scheme introduced in chapter 4 and the drive model should be improved to account for effects appearing in low switching frequency regimes.

The robustness of the proposed methods against variations in the model parameters needs to be verified. In this sense they could also be extended to use on-line parameter fitting schemes.

From the control perspective, the smoothening property needs to be improved: the results obtained with the position controller in chapter 9 show that chattering near the steady state is effectively eliminated passing the control task to a classical controller. The transition phase, however, is not completely smooth and chattering could still be observed, when the state was near the switching curve. In this sense, analytical tools are required to stablish the region where the classical and smoother controller should take over and to match the behaviour of the classical controller and the bang-bang strategy on the boundary of this region.

The general approach used during the development of this work was to focus on one problem, solve it and move on to the next problem using the lessons learned up to that point. Still, the form the work took displays some kind of pattern: the methods proposed in part III for the second order system consist broadly of a prediction stage and an evaluation stage (the cost function). The methods proposed in part IV for the third order systems consist of two prediction stages and two function evaluation stages: the Γ curve and the cost function: the complexity of the algorithms did not grow exponentially, but rather linearly. A consequence of the dynamic programming principle is the curse of dimensionality: the complexity of an optimal control problem grows exponentially with the order of the system. The algorithm developed in this work seem to suggest that the curse of dimensionality could be dodged using particular knowledge of the system at hand. This is no evidence to make any general statement, but rather opens the following questions

- can the synthesis of the control algorithms proposed in this work be systematized for systems of arbitrary order?

- what is the nature of the information required to devise these algorithms?

Meaningful answers to these questions would enable the development of new control methods, based on the time-optimality performance index, for a broader family of mechatronic systems.

Appendix

A. State Observer: Reduced Order Extended Kalman Filter

In all the control schemes described in this work, except for that in section 6: torque control for the induction motor, the whole state of the system being controlled is required as feedback for the controller.

In order to achieve this and effectively decouple the control problem from the observation one, so as to focus on the control, a Kalman filter was used in each case [61, 4, 62, 63]. The Kalman filter is one of the greatest achievements of control theory in the past century: it is the solution for the optimal observer for linear systems subject to white noise in its measurements and system equations, considering a quadratic criterion. It is to observers, what the Linear Quadratic Regulator is to linear controllers: the gain for the innovations (the error between the estimated output and the actual output of the system: $(c(\hat{x})-y)$), is calculated using the model of the noise and the system to minimize the expected value of square of the error between the actual system state x and the estimated one $\hat{x}$, $E[x-\hat{x}]^2$. The obtained gain is called the Kalman gain.

The Kalman Filter can also be applied to non-linear systems linearizing at each operation point to recalculate the Kalman gain at each step and , although optimality can not be guaranteed, its performance is still reasonable. This kind of filter is called Extended Kalman Filter, and was used in this work [64, 65, 63], given the non-linearities posed by the models of both the induction motor and the synchronous motor in the *back-emf* terms.

It has been repeatedly noted, that one of the difficulties posed by this kind of observer is its computational burden. To cope with this, the order of the implemented observer was reduced by assuming that the noise in the measurements for the stator currents is neglectable. This kind of observers are called Luenberger observers. A complete formulation for such observers can be found in [3].

In this section we will describe the steps necesary to implement a reduced order extended Kalman filter for the systems described in chapter 2.

None of the control schemes described in this work have internal states. This means that they do not have any means to *integrate* to compensate for constant or slow-varying perturbations: they rely on the assumption that an estimate of the perturbation is available. In all cases, where a Kalman filter was used, it was assumed that the only external perturbation is the load torque T_l. In order to estimate its value, the state of the observer was extended to include T_l, with the associated dynamic equation:

$$\frac{dT_l}{dt} = 0, \tag{A.1}$$

which entails that T_l changes very slowly with time.

With this, the state for the PMSM, the IM and the two-mass-system, driven by the PMSM are given respectively by:

$$x = \begin{bmatrix} i_\alpha \\ i_\beta \\ \omega \\ \theta \\ T_l \end{bmatrix} \qquad x = \begin{bmatrix} i_\alpha \\ i_\beta \\ \psi_\alpha \\ \psi_\beta \\ \omega \\ \theta \\ T_l \end{bmatrix} \qquad x = \begin{bmatrix} i_\alpha \\ i_\beta \\ \omega_m \\ \theta_m \\ T_s \\ \omega_l \\ T_l \end{bmatrix} \tag{A.2}$$

PMSM IM PMSM-TMS

The model of the IM, introduced in section 2.3, eq. (2.56), did not include the rotor angle θ_r as a state. It is included here, for the filter to be able to gather information from the angle measurement, which is realized with an incremental encoder. The dynamic equation associated with θ_r is given by:

$$\frac{d\theta_r}{dt} = \omega \tag{A.4}$$

A discrete time version of this equation is obtained as in eq. (2.58), with $N = 3$.

In all cases the formulation of the Kalman filter requires the introduction of a measurements vector as a function of the state:

$$y = \begin{bmatrix} i_\alpha \\ i_\beta \\ \theta \end{bmatrix} \tag{A.5}$$

In each case θ_r corresponds to the rotor angle measurement realized by the incremental encoder and i_α and i_β to the stator currents, measured with LEM sensors.

For each case, the reduced state p, which includes only the states to be filtered, is defined as:

$$p = \begin{bmatrix} \omega \\ \theta \\ T_l \end{bmatrix} \qquad p = \begin{bmatrix} \psi_\alpha \\ \psi_\beta \\ \omega \\ \theta \\ T_l \end{bmatrix} \qquad p = \begin{bmatrix} \omega_m \\ \theta_m \\ T_s \\ \omega_l \\ T_l \end{bmatrix} \tag{A.6}$$

PMSM IM PMSM-TMS

Then, a partition for the output is introduced, which separates the measurements between those with and without noise (y_1 and y_2 respectively):

$$y = \left[\begin{array}{c} y_1 \\ \hline y_2 \end{array}\right] = \left[\begin{array}{c} \theta \\ \hline i_\alpha \\ i_\beta \end{array}\right] \tag{A.8}$$

This introduces a partition in the Kalman gain:

$$H = [H_1 \,|\, H_2] \tag{A.9}$$

and the costate:

$$q[n] = p[n] - H_2[n-1]y_2[n]. \tag{A.10}$$

Using these definitions a state space representation for p is built:

$$p[n+1] = g(p[n], u[n], y_2[n]) + v[n] \tag{A.11}$$

$$z[n] = c(p[n], u[n], y_2[n]) + w[n] \tag{A.12}$$

Here $g(\cdot)$ includes the discrete time version of the dynamic equations for the states included in p: it includes a partition of $f(\cdot)$ in eq. (2.59) and the discrete time versions of (A.4) and (A.1). v and w represent the system and measurement noise respectively, they are assumed to have normal probability distributions, characterized by their covariance matrices Q and R. If a model of w and w were available, their covariance matrices could be used directly, but since the production of such a model is cumbersome and out of the scope of this work, they were tuned empirically by iterative simulation and trial with the experimental setup.

The output function $c(\cdot)$ is given by:

$$c(\cdot) = \begin{bmatrix} y_1[n] \\ y_2[n+1] \end{bmatrix} \tag{A.13}$$

$y_2[n+1]$ corresponds to the discrete time version of the dynamic equations for $\vec{i_s}$: it is a partition of $f(\cdot)$ in eq. (2.59).

The filter algorithm consists of two stages: *correction* and *prediction*. In the correction stage, the state estimate for the current sampling instant k (predicted in the last iteration) is corrected with the new available measurements:

$$\hat{p}[n|n] = q[n] + H_2[n-1]y_2[n]. \tag{A.14}$$

Tanking this estimate for the state as operating point, the system dynamics are linearized and the Kalman gain is calculated:

$$A[n] = \left.\frac{\partial g}{\partial p}\right|_{p=\hat{p}[n|n]} \tag{A.15}$$

$$C[n] = \left.\frac{\partial c}{\partial p}\right|_{p=\hat{p}[n|n]} \tag{A.16}$$

$$H[n] = A[n]P[n]C[n]^T \left(R + C[n]P[n]C[n]^T\right)^{-1} \tag{A.17}$$

$$\tag{A.18}$$

A and C are the Jacobian matrices of $g(\cdot)$ and $c(\cdot)$ respectively.

In the prediction stage, the state and the error covariance matrix for the next sampling instant are predicted using the system model and the estimated state at the current sampling instant k:

$$\hat{p}[n+1|n] = g(\hat{p}[n|n], u[n], y_2[n]) \tag{A.19}$$
$$q[n+1] = \hat{p}[n+1|n] + H_1[n]y_1[n] - H[n]\hat{z}[n] \tag{A.20}$$
$$P[n+1] = A[n]P[n]A[n]^T + Q - H[n]C[n]P[n]A[n]^T \tag{A.21}$$

The system state can be reconstructed after evaluating eq. (A.14) or (A.19):

$$\hat{x}[n] = \begin{bmatrix} y_2[n] \\ \hat{p}[n|n] \end{bmatrix} \tag{A.22}$$

Then $\hat{x}$ is used as the state feedback for the controller. In the control methods described in this work eq. (A.19) was used to compensate for the calculation time delay (see chapter 3.2).

The computation of the filter can be very demanding. Therefore special care was taken for the implementation in the experimental setup, in order to make the computation of the filter as efficient as possible: all the matrix operations were written explicitly, instead of using loops. To evaluate the Jacobian matrices A and C, all terms that are constant are computed offline, and the expressions for the terms that are not constant are grouped so as to end up with expressions consisting only of additions and multiplications between variables and constants. Trigonometric functions are computed only once, since they all take the same argument.

B. Current model for the Induction Motor - Stator Flux Observer

All of the torque control schemes for the induction machine described in this work (see sections 3.1, 3.2 and chapter 6) assume that measurements of at least stator currents and rotor flux are available. Hall sensors for the magnetic fluxes are very complex and, moreover, extremely expensive. Therefore, flux measurements are usually not available and it needs to be estimated. A common technique to achieve this involves the integration of the model equations, using the available measurements. When measurements of the speed (or at least good estimates) are available, the preferred equation is (2.44), written here again:

$$\tau_r \frac{d\vec{\psi}_r}{dt} + \vec{\psi}_r = L_m \vec{i}_s + J\omega_r \tau_r \vec{\psi}_r, \tag{B.1}$$

since the negative feedback of $\vec{\psi}_r$ ensures that the integral for the estimated flux:

$$\underset{\sim}{\vec{\psi}}_r = \frac{1}{\tau_r} \int \left((J\omega_r \tau_r - 1)\underset{\sim}{\vec{\psi}}_r + L_m \vec{i}_s \right) dt \tag{B.2}$$

will not explode, even in the presence of offsets in the current measurements. Eq. (B.2) is implemented in practice using the discrete version of (B.1) (see section 2.4).

With this, the estimated angle of the flux in the $\alpha\beta$ plane can be calculated with:

$$\tilde{\theta}_k = \texttt{atan2}\left(\underset{\sim}{\psi}_{r\beta},\, \underset{\sim}{\psi}_{r\alpha} \right), \tag{B.3}$$

the motor variables in the dq frame are then given by:

$$\overset{\circ}{\underset{\sim}{i}}_s = T^{-1}(\tilde{\theta}_k)\vec{i}_s \tag{B.4}$$

$$\overset{\circ}{\underset{\sim}{v}}_s = T^{-1}(\tilde{\theta}_k)\vec{v}_s \tag{B.5}$$

List of Figures

List of Tables

Bibliography

[1] J. C. Maxwell, "On governors," *Proceedings of the Royal Society of London*, vol. 16, pp. 270–283, 1867.

[2] S. Bergbreiter, "Moving from practice to theory: Automatic control after world war ii," *History of Science, University of California, Berkeley*, 2005.

[3] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems.* John Wiley & Sons, 1972.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[5] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, pp. 503–515, 1954.

[6] L. Pontrëiìagin and L. Neustadt, *The Mathematical Theory of Optimal Processes.* No. Bd. 4 in Classics of Soviet Mathematics, Gordon and Breach Science Publishers, 1962.

[7] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[8] L. Pao and G. Franklin, "Proximate time-optimal control of third-order servomechanisms," *Automatic Control, IEEE Transactions on*, vol. 38, no. 4, pp. 560–580, 1993.

[9] H. Geering, L. Guzzella, S. Hepner, and C. Onder, "Time-optimal motions of robots in assembly tasks," *Automatic Control, IEEE Transactions on*, vol. 31, no. 6, pp. 512–518, 1986.

[10] F. Blaschke, "The principle of field orientation as applied to the new TRANSVECTOR-closed loop control systems for rotating field machines," *Simens Rev.*, vol. 34, 1972.

[11] M. Depenbrock, "Direct self-control (dsc) of inverter-fed induction machine," *Power Electronics, IEEE Transactions on*, vol. 3, no. 4, pp. 420–429, 1988.

[12] I. Takahashi and Y. Ohmori, "High-performance direct torque control of an induction motor," *Industry Applications, IEEE Transactions on*, vol. 25, no. 2, pp. 257–264, 1989.

[13] J. Rodriguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortes, and U. Ammann, "Predictive current control of a voltage source inverter," *Industrial Electronics, IEEE Transactions on*, vol. 54, pp. 495 –503, feb. 2007.

[14] P. Cortes, M. Kazmierkowski, R. Kennel, D. Quevedo, and J. Rodriguez, "Predictive control in power electronics and drives," *Industrial Electronics, IEEE Transactions on*, vol. 55, pp. 4312 –4324, dec. 2008.

[15] H. Miranda, P. Cortes, J. Yuz, and J. Rodriguez, "Predictive torque control of induction machines based on state-space models," *Industrial Electronics, IEEE Transactions on*, vol. 56, pp. 1916 –1924, june 2009.

[16] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, "Model Predictive Control-A Simple and Powerful Method to Control Power Converters," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 6, pp. 1826–1838, 2009.

[17] M. Perez, J. Rodriguez, E. Fuentes, and F. Kammerer, "Predictive control of ac-ac modular multilevel converters," *Industrial Electronics, IEEE Transactions on*, vol. 59, pp. 2832–2839, July 2012.

[18] G. V. Rossum and F. L. Drake, "Python reference manual release 2.2.1," 2002.

[19] D. Ascher, P. F. Dubois, K. Hinsen, J. Hugunin, and T. Oliphant, *Numerical Python*. Lawrence Livermore National Laboratory, Livermore, CA, ucrl-ma-128569 ed., 1999.

[20] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–.

[21] E. Fuentes and H. Martinez, "Sclib, a hack for straightforward embedded c functions in (i)python," 2014–.

[22] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[23] W. Duesterhoeft, M. W. Schulz, and E. Clarke, "Determination of instantaneous currents and voltages by means of alpha, beta, and zero components," *American Institute of Electrical Engineers, Transactions of the*, vol. 70, pp. 1248–1255, July 1951.

[24] H. van der Broeck, H.-C. Skudelny, and G. Stanke, "Analysis and realization of a pulsewidth modulator based on voltage space vectors," *Industry Applications, IEEE Transactions on*, vol. 24, pp. 142–150, Jan 1988.

[25] J. Rodriguez, J.-S. Lai, and F. Z. Peng, "Multilevel inverters: a survey of topologies, controls, and applications," *Industrial Electronics, IEEE Transactions on*, vol. 49, pp. 724–738, Aug 2002.

[26] D. W. Novotny and T. Lipo, *Vector Control and Dynamics of Ac Drives.* Monographs in electrical and electronic engineering, Oxford University Press, Incorporated, 1996.

[27] R. H. Park, "Two-reaction theory of synchronous machines generalized method of analysis-part i," *American Institute of Electrical Engineers, Transactions of the*, vol. 48, no. 3, pp. 716–727, 1929.

[28] T. Miller, *Permanent Magnet and Reluctance Motor Drives.* Oxford, UK: Oxford Science Publications, 1989.

[29] D. Paulus, P. Landsmann, and R. Kennel, "Sensorless field- oriented control for permanent magnet synchronous machines with an arbitrary injection scheme and direct angle calculation," in *Sensorless Control for Electrical Drives (SLED), 2011 Symposium on*, pp. 41–46, 2011.

[30] C. Hackl, H. Schuster, C. Westermaier, and D. Schroder, "Funnel-control with integrating prefilter for nonlinear, time-varying two-mass flexible servo systems," in *Advanced Motion Control, 2006. 9th IEEE International Workshop on*, pp. 456–461, IEEE, 2006.

[31] J. Cordier, P. Landsmann, and R. Kennel, "The influence of magnetic hysteresis on hf injection based inductance calculation," in *Energy Conversion Congress and Exposition (ECCE), 2011 IEEE*, pp. 638–645, Sept 2011.

[32] W. Leonhard, *Control of electrical drives.* Engineering Online Library, Springer-Verlag, 2001.

[33] J. Holtz, "Sensorless control of induction motor drives," *Proceedings of the IEEE*, vol. 90, pp. 1359–1394, Aug. 2002.

[34] E. J. Fuentes, C. A. Silva, and J. I. Yuz, "Predictive Speed Control of a Two-Mass System Driven by a Permanent Magnet Synchronous Motor," *Industrial Electronics, IEEE Transactions on*, vol. 59, pp. 2840–2848, July 2012.

[35] C. A. Silva and J. I. Yuz, "On sampled-data models for model predictive control," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pp. 2966–2971, IEEE, Nov. 2010.

[36] Z. Zhang, D. U. An, H. Kim, and K. T. Chong, "Comparative study of Matrix exponential and Taylor series discretization methods for nonlinear ODEs," *Simulation Modelling Practice and Theory*, vol. 17, no. 2, pp. 471–484, 2009.

[37] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control system design*, vol. 240. Prentice Hall Upper Saddle River, 2001.

[38] J. Rodriguez, R. Kennel, J. Espinoza, M. Trincado, C. Silva, and C. Rojas, "High-performance control strategies for electrical drives: An experimental assessment," *Industrial Electronics, IEEE Transactions on*, vol. 59, pp. 812 –820, feb. 2012.

[39] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.

[40] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[41] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[42] P. Stolze, P. Landsmann, R. Kennel, and T. Mouton, "Finite-set model predictive control of a flying capacitor converter with heuristic voltage vector preselection," in *Power Electronics and ECCE Asia (ICPE ECCE), 2011 IEEE 8th International Conference on*, pp. 210–217, May 2011.

[43] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms* . Springer-Verlag, 2011.

[44] C. Y. Kaya, S. K. Lucas, and S. T. Simakov, "Computations for bang—bang constrained optimal control using a mathematical programming formulation," *Optimal Control Applications and Methods*, vol. 25, no. 6, pp. 295–308, 2004.

[45] M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in optimization and numerical analysis*, pp. 51–67, Springer, 1994.

[46] L. Hongmei, Z. Ya'nan, S. Xiang, B. Zhengjie, and N. Xiaojun, "High-performance control of permanent magnet synchronous motor," in *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, pp. 2738–2741, June 2011.

[47] D. Chen, L. Bako, and S. Lecoeuche, "The minimum-time problem for discrete-time linear systems: A non-smooth optimization approach," in *Control Applications (CCA), 2012 IEEE International Conference on*, pp. 196–201, 2012.

[48] T. Kailath, *Linear systems*, vol. 1. Prentice-Hall Englewood Cliffs, NJ, 1980.

[49] V. Rao and D. Bernstein, "Naive control of the double integrator," *IEEE Control Systems Magazine*, vol. 21, pp. 86–97, Oct. 2001.

[50] R. Aguilera, P. Lezana, and D. Quevedo, "Finite-control-set model predictive control with improved steady-state performance," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 2, pp. 658–667, 2013.

[51] V. Utkin, "Sliding mode control design principles and applications to electric drives," *Industrial Electronics, IEEE Transactions on*, vol. 40, no. 1, pp. 23–36, 1993.

[52] S. Di Gennaro and M. Tursini, "Control techniques for synchronous motor with flexible shaft," in *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pp. 471–476 vol.1, Aug 1994.

[53] P. Koronki, H. Hashimoto, and V. Utkin, "Direct torsion control of flexible shaft in an observer-based discrete-time sliding mode," *Industrial Electronics, IEEE Transactions on*, vol. 45, pp. 291–296, Apr 1998.

[54] P. Koroundi, H. Hashimoto, and V. Utkin, "Discrete sliding mode control of two mass system," in *Industrial Electronics, 1995. ISIE '95., Proceedings of the IEEE International Symposium on*, vol. 1, pp. 338–343 vol.1, Jul 1995.

[55] K. Date, H. Ohmori, A. Sano, Y. Todaka, and H. Nishida, "Speed control of two-mass resonant system by new simple adaptive control scheme," in *Proc. IEEE International Conference on Control Applications*, vol. 2, pp. 1120–1124, 1–4 Sept. 1998.

[56] D. Szabo, S. Kerekes, O. Dranga, and T. Gajdar, "A fuzzy sliding mode approach for the two-mass system," in *Proc. IEEE International Symposium on Industrial Electronics ISIE '99*, vol. 1, pp. 348–352, 12–16 July 1999.

[57] S. Thomsen, N. Hoffmann, and F. W. Fuchs, "Pi control, pi-based state space control, and model-based predictive control for drive systems with elastically coupled loads—a comparative study," *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3647–3657, 2011.

[58] E. Fuentes, C. Silva, and J. Yuz, "Predictive speed control of a two-mass system driven by a permanent magnet synchronous motor," *Industrial Electronics, IEEE Transactions on*, vol. 59, pp. 2840–2848, July 2012.

[59] E. Fuentes and R. Kennel, "Finite-set model predictive control of the two-mass-system," in *Proc. IEEE 1th Workshop on Predictive Control of Electrical Drives and Power Electronics PRECEDE 2011*, pp. 390–395, 17–20 May 2011.

[60] S. Villwock and M. Pacas, "Application of the welch-method for the identification of two- and three-mass-systems," *Industrial Electronics, IEEE Transactions on*, vol. 55, pp. 457–466, Jan 2008.

[61] R. Kalman, "Contributions to the theory of optimal control," 1960.

[62] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Trans. ASME, Ser. D, J. Basic Eng*, p. 109, 1961.

[63] R. Stengel, *Optimal Control and Estimation.* Dover books on advanced mathematics, Dover Publications, 1986.

[64] B. A. McElhoe, “An assessment of the navigation and course corrections for a manned flyby of mars or venus,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-2, pp. 613–623, July 1966.

[65] G. L. Smith, S. F. Schmidt, and L. A. McGee, *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle.* National Aeronautics and Space Administration, 1962.