ANDREA FAVATO

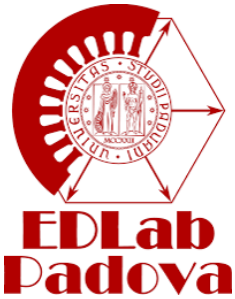
CUTTING EDGE COMPUTATIONAL METHODS FOR THE DEPLOYMENT OF PREDICTIVE CONTROL IN ELECTRIC DRIVES

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

University of Padova
Department of Industrial Engineering

Ph.D. Course in
Electrical Energy Engineering

XXXIV CYCLE



CUTTING EDGE COMPUTATIONAL METHODS FOR THE DEPLOYMENT OF PREDICTIVE CONTROL IN ELECTRIC DRIVES

Candidate
Andrea Favato

Supervisor
Prof. Silverio Bolognani

Ph.D. School Coordinator
Prof. Giulio Rosati

Curriculum Coordinator
Prof. Luigi Aberti

30 September 2021

Dove vanno gli aerei che non prendiamo?
Vanno in posti in cui non sono mai stato.

Se ne vanno altrove.
— The Young Pope

ABSTRACT

This work covers many research aspects of model-based and data-driven control techniques for permanent magnet synchronous motor drives, suitable for both isotropic motor, such as Surface Permanent Magnet (SPM), and anisotropic motors which are interior permanent magnet (IPM) and synchronous reluctance machines (SyRM). On one hand, the research interest on model-based algorithms for electric drive applications is constantly growing. On the other hand, the interest in data analysis is constantly growing and this leads to an increasing attention towards data-enabled methods in all branches of science and engineering. This revolution has a significant impact on the control engineering too. Data-driven control design consists in synthesizing a controller using the data collected on the real system, without defining and identifying a parametric model for the plant. Improvements in the computational power and the development of dedicated hardware solutions are making these complementary methods, that serve the same purpose, suitable for fast-dynamic industrial applications. The thesis is structured as follows. The first part gives the background knowledge needed to develop the topics covered in the following. This comprehends an introduction to the machines and drives, and some basic-concepts of model-based and data-driven control theory. The framework of optimization based algorithms is considered. In particular, the Continuous-Set (CS) approach to Model Predictive Control (MPC) is investigated, which is stated as a quadratic programming (QP) problem type, where feasible system constraints are included directly in the optimization problem. The second part deals with two aspects of MPC: the offset-free formulation in presence of model uncertainties is addressed, and a custom and efficient QP solvers for electric drives applications is presented, where usually limited computational hardware is available. Finally, the data-driven control of electric machines is investigated, with particular attention on the comparison of performance and computational aspects with respect the model-based approach.

SOMMARIO

Il lavoro di tesi copre molti aspetti di ricerca sulle tecniche di controllo basate sul modello e sui dati, applicate agli azionamenti dei motori sincroni a magneti permanenti, adatti sia ai motori isotropi, come il motore a magneti permanenti superficiali, sia ai motori anisotropi che, ovvero a magneti interni e macchine a riluttanza. Da un lato, l'interesse della ricerca sugli algoritmi basati su modelli per applicazioni agli azionamenti elettrici è in costante crescita. Dall'altro lato, l'interesse per l'analisi dei dati è in costante crescita e questo ha portato a una crescente attenzione verso algoritmi basati sui dati, in tutti i rami della scienza e dell'ingegneria. Questa rivoluzione sta avendo un impatto significativo anche sull'ingegneria del controllo. La progettazione di controllori così detti "data-driven" consiste nel sintetizzare un controllore utilizzando i dati raccolti sul sistema reale, senza definire o identificare un modello parametrico per l'impianto. I miglioramenti nella potenza di calcolo e lo sviluppo di soluzioni hardware dedicate stanno rendendo entrambe metodi adatti ad applicazioni industriali con elevate dinamiche.

La tesi è strutturata come segue. La prima parte fornisce alcune conoscenze necessarie per sviluppare gli argomenti trattati nel seguito. Questo comprende un'introduzione alle macchine e agli azionamenti, e alcuni concetti di base della teoria del controllo basata sul modello e dai dati. Vengono considerati poi algoritmi basati sull'ottimizzazione. In particolare, viene studiato l'approccio Continuous-Set (CS) al Model Predictive Control (MPC), che è visto come un tipo di problema di programmazione quadratica (QP), dove i limiti di ingresso e uscita del sistema sono inclusi direttamente nel problema di ottimizzazione. La seconda parte si occupa di due aspetti dell'MPC: viene affrontata la formulazione senza offset in presenza di incertezze del modello, e viene presentato un solutore QP personalizzato ed efficiente per applicazioni agli azionamenti elettrici, dove solitamente è disponibile una limitata potenza di calcolo. Infine, viene studiato il controllo data-driven applicato ai motori elettrici, con particolare attenzione alle prestazioni e agli aspetti computazionali.

PUBLICATIONS

JOURNAL PAPERS

A. Favato, P. G. Carlet, F. Toso, R. Torchio, and S. Bolognani. "Integral Model Predictive Current Control for Synchronous Motor Drives." In: *IEEE Transaction on Power Electronics* (2021), pp. 1–1. DOI: 10.1109/TPEL.2021.3081827.

F. Toso, A. Favato, R. Torchio, P. Alotto, and S. Bolognani. "Continuous Control Set Model Predictive Current Control of a Microgrid-Connected PWM Inverter." In: *IEEE Transactions on Power Systems* 36.1 (2021), pp. 415–425. DOI: 10.1109/TPWRS.2020.3006237.

CONFERENCE PAPERS

F. Toso, R. Torchio, A. Favato, P. G. Carlet, S. Bolognani, and P. Alotto. "Digital Twins as Electric Motor Soft-Sensors in the Automotive Industry." In: *2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*. 2021, pp. 13–18. DOI: 10.1109/MetroAutomotive50197.2021.9502885.

P. G. Carlet, A. Favato, S. Bolognani, and F. Dörfler. "Data-driven predictive current control for synchronous motor drives." In: *2020 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2020, pp. 5148–5154. DOI: 10.1109/ECCE44975.2020.9235958.

F. Toso, P. G. Carlet, A. Favato, and S. Bolognani. "On-line Continuous Control Set MPC for PMSM drives current loops at high sampling rate using qpOASES." In: *IEEE Energy Convers. Congr. and Expo. (ECCE)*. 2019.

P. G. Carlet, F. Toso, A. Favato, and S. Bolognani. "A speed and current cascade Continuous Control Set Model Predictive Control architecture for synchronous motor drives." In: *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2019, pp. 5682–5688. DOI: 10.1109/ECCE.2019.8912277.

P. G. Carlet, A. Favato, F. Toso, and S. Bolognani. "Sensorless control of Interior Permanent Magnet motor using a Moving Horizon Estimator based on a linearized motor model." In: *2019 IEEE 10th International Symposium on Sensorless Control for Electrical Drives (SLED)*. 2019, pp. 1–6. DOI: 10.1109/SLED.2019.8896319.

A. Favato, F. Toso, P. G. Carlet, M. Carbonieri, and S. Bolognani. "Fast Moving Horizon Estimator for Induction Motor Sensorless Control." In: *2019 IEEE 10th International Symposium on Sensorless Control for Electrical Drives (SLED)*. 2019, pp. 1–6. DOI: 10.1109/SLED.2019.8896262.

A. Favato, P. G. Carlet, F. Toso, and S. Bolognani. "A Novel Formulation of Continuous Control Set MPC for Induction Motor Drives." In: *2019 IEEE International Electric Machines Drives Conference (IEMDC)*. 2019, pp. 2196–2202. DOI: 10.1109/IEMDC.2019.8785407.

F. Toso, M. De Soricellis, P. G. Carlet, A. Favato, and S. Bolognani. "MHE-MPC Based Control Architecture of an LCL Filter Grid-Connected PWM Inverter." In: *2019 IEEE International Electric Machines Drives Conference (IEMDC)*. 2019, pp. 392–399. DOI: 10.1109/IEMDC.2019.8785172.

A. Favato, P. Carlet, F. Toso, and S. Bolognani. "A Model Predictive Control for Synchronous Motor Drive with Integral Action." In: *Proc. IECON 2018 - 44th Annu. Conf. IEEE Ind. Electron. Soc.* 2018. ISBN: 9781509066841. DOI: 10.1109/IECON.2018.8591769.

SUBMITTED PAPERS

A. Favato, P. G. Carlet, F. Toso, R. Torchio, M. Bruschetta, R. Carli, S. Bolognani, and R. Jose. "Fast Solver for Implicit Continuous Set Model Predictive Control of Electric Drives." In: *IEEE Transaction on Industrial Electronics* ().

P. G. Carlet, A. Favato, S. Bolognani, and F. Dörfler. "Data-Driven Continuous-Set Predictive Current Control for Synchronous Motor Drives." In: *IEEE Transaction on Power Electronics* ().

ACKNOWLEDGEMENTS

First of all, I would like to thank my family, for always being close to me, for encouraging and supporting me in my choices. I will always be grateful to my grandparents, that growed me and teached me "come stare al mondo".

I also thank Alice, who has always supported me with great patience.

I would like to thank Professor Silverio Bolognani, who has guided me during the last years with wisdom. I will always be grateful to him for giving me the opportunity to continue my studies with the Ph.D.

I am also grateful to Professors Florian Dorfler and Saverio Bolognani, for hosting me at ETH Zurich and giving me the opportunity to work in such a stimulating environment.

A special thanks goes to my colleagues at EDLab, for all the experiences we shared together and the discussions and help I received from all of them. Thank you especially for all the evenings we spent together. I will list names here, hoping to remember every single one of them: Giacomo, Grazia, Matteo C., Emanuel, Francesco, Riccardo, Diego, Paolo, Matteo B., Luca, Ludovico, Elia and Giuseppe. A special thank goes to Paolo for sharing this journey together, and for all the bike rides we have done and those that are waiting for us.

Thank to Francesco, for being my support, my roommate and my friend.

Thanks also to Mosè, for the very important support, and for supervising me during my "parallel doctorate" in bicycle maintenance. I will carry with me all the technical (and not) discussions we shared.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AS Active Set

back-EMF back-Electro Motive Force

CCS Continuous Control Set

DeePC Data EnablEd Predictive Control

DSP Digital Signal Processor

DOB Disturbance Observer

EM Electrical Machine

FOC Field Oriented Control

FCS Finite Control Set

GP Gradient Projection

I/O Input/Output

IGBT Insulated-Gate Bipolar Transistor

I-MPC Integral-Model Predictive Control

IP Interior Point

IPM Interior Permanent Magnet

KKT Karush–Kuhn–Tucker

LQR Linear Quadratic Regulator

LTI Linear Time Invariant

MTPA Maximum Torque per Ampere

MOR Model Order Reduction

MPC Model Predictive Control

PMSM Permanent Magnet Synchronous Motor

PEM Prediction Error Method

PID Proportional-Integral-Derivative

pMOR Parametric MOR

POD Proper Orthogonal Decomposition

PWM Pulse-Width-Modulation

QP Quadratic Programming

RH Receding Horizon

ROM Reduced Order Model

SVD Singular Value Decomposition

SPWM Sinusoidal Pulse-Width-Modulation

SyRM Synchronous Reluctance Motor

SVPWM Space Vector Pulse-Width-Modulation

SPC Subsapce Predictive Control

SPM Surface Permanent Magnet

THD Total Harmonic Distortion

Part I

BACKGROUND KNOWLEDGE

1 INTRODUCTION

1.1 THE IMPORTANCE OF RESEARCH

Research and innovation proved to be among the most powerful of European policies to boost the Union's economies and competitiveness at the global scale. With its ability to drive growth, to create up to 320.000 new highly skilled jobs by 2040 and to leverage approximately 11 euro of additional investments for each euro invested at the European level, the Research and Innovation (R&I) policy is an engine of the green and digital transitions on the continent and stairways to 'the future we want'.

These words are included in the Development Goal (DG) Research and innovation strategy plan 2020-2024 [1], which will contribute to all six headline ambitions[1] of the von der Leyen[2] Commission, as well as contributing to a modern, high performing and sustainable European Commission. The DG will deliver important parts of the recovery plan for Europe based on climate, digital, health and resilience.

In this enormous plan, which involves not only the European Community but it strictly requires the collaboration with all the other countries, it emerges that on 9 March 2021, the Commission presented a vision and avenues for Europe's digital transformation by 2030. This vision for the EU's digital decade evolves around four cardinal points: skills, governments, business and infrastructure. Among the key goal, it is highlighted that it is expected that more than 75% of EU companies will make use of Cloud/Artificial Intelligence (AI)/Big Data technologies. Thus, data represent the central point around which new technologies will be developed. This innovation process that is radically changing the way of life, can be pursued only with research, which has the fundamental role of exploring and developing new technologies and pave the way for the future of human society and our planet.

1.2 THE FUTURE OF POWER ELECTRONICS

The technology of power electronics has practically attained maturity after five decades of dynamic evolution. In the future, there will be

1 the European Council's strategic agenda for 2019-2024 set the work of the European institutions around six headline ambitions for Europe for this period: a European Green Deal, a Europe fit for the digital age, an economy that works for people, a stronger Europe in the world, promoting our European way of life, a new push for European democracy.

2 Ursula von der Leyen is the President of the European Commission from 2019 to 2024.

tremendous emphasis on power electronics applications in the areas of industrial, residential, commercial, transportation, aerospace, and electric utility systems [2].

With the development towards Industry 4.0, the requirement for "Smart-Systems" and autonomy in industry drives and automation demands higher levels of intelligence from the power electronic systems used to facilitate correct power flow. With an estimated €36 billions predicted to be spent on inverters and drive systems in 2022, academic and industrial research and development identifies the power electronic inverter as being the critical driver for Industry 4.0 in the coming years [3]. The development of power electronic building blocks has led to the ability to rapidly prototype and develop industry motor drives for a wide range of applications and power levels, but for true autonomy to be possible, the power electronic inverter must now be developed as a "Smart" unit, capable of not only adapting its operation to any electrical supply/load fluctuations, but also be self-configurable to account for changes in operational conditions. Such conditions include the ability to adapt to required circuit topology changes, whether customer-requested (application based) or because of fault conditions/damaged components (protection and reliability based). Improvements in efficiency may also be achieved through software-defined control of the topology and switching states of the semiconductor switches, whilst further modularity improvements may be incorporated with the ability to autonomously add/remove sensors and actuators to increase data collection and functionality of the system.

Two other drivers for the development of power electronics had been clearly identified for the new decade: CO_2 reduction to stop the climate change and digitalization. Thus, as previous said, in the coming decades it is expected to see increasing emphasis on very high-specialized research and developments in system modularization, analysis, modeling, real time simulation, design and experimental evaluations. Power electronics will have increasing impact not only in global industrial automation and high efficiency energy systems, but also on energy conservation, renewable energy systems, and electric/hybrid vehicles. The resulting impact in mitigating climate change problems due to man-made environmental pollution is expected to be considerable [2].

1.3 CONTRIBUTION OF THE THESIS

Together with power electronics development, a fundamental factor in industrial automation is the electromechanical conversion, where electric drives, consisting of an electric motor supplied by a frequency converter, play a major role. Efficiency and performance depict the essential and unavoidable targets in up-to-date projects. Both the

goals can be achieved using Permanent Magnet Synchronous Motor (PMSM). They have been adopted in high dynamic applications since the 1980s, when rare–earth magnetic materials and low-cost electronics devices were introduced in the market. The main features of PMSM are the high torque and power density and the high efficiency. Their relatively high production cost is one of the reasons that limited their diffusion. Nevertheless, nowadays their strengths are overcoming their limitations and they are gaining new attentions. In particular, car manufacturers are choosing PMSM for their electric and hybrid cars. In fact, the special features for adjustable-speed operation which distinguish PMSMs from other classes of AC machines were already shown in the very early 1986 [4]. The *Global electric motor market report*, a recent research by Allied Market Research, has revealed that the global electric motor market size was $96,968 million in 2017, and is projected to reach $136,496 million in 2025. [5]. Thus, the vehicle market, which has a great impact in society, even for non-experts or non-engineers, can be the "engine" of the new industrial revolution, allowing electric motors to be a central key point in the digital revolution in all other industries.

The rise in demand for superior and efficient machine in any industrial sector, requires the development of a suitable and high-performance control unit. In this thesis, two complementary approaches are investigated for accomplishing these requirements. On one hand, model-based control is explored. In particular, the paradigm of Model Predictive Control (MPC) is deeply investigated for controlling synchronous motors. MPCs strategies have been investigated for two decades as advanced control method for power electronics applications. The basic concept of MPC combines optimization-based techniques and plant model. The future plant dynamic is estimated by means of models and the actions (inputs) on the plant are computed by optimizing a certain performance expression. The idea of MPC was introduced in chemistry industries, where the higher computational effort with respect to standard controllers was possible thank to the usually relatively slows dynamics of the system [6], [7]. Improvements in the computational power of modern Digital Signal Processor (DSP)s and the development of dedicated hardware solutions are making these algorithms suitable for industrial implementations, representing a promising strategy for electric drives applications [8],[9]. In particular, for the above-mentioned challenges of high penetration of electrical technologies in many industrial sectors, the importance of finding custom solution for the specific applications will require high-level of competences, strong mathematical background and the ability of adapt different existing mathematical tools for satisfying a specific requirement. In this scenario, the deployment of advanced control algorithms will be of paramount of importance. Furthermore, model-based control ap-

proaches such MPC requires much more computational effort with respect standard Proportional-Integral-Derivative (PID) controllers. This fact sounds in contrast with standard controller hardware that is used for power converters, with a limited amount of memory and computational speed (up to some hundreds of MHz). For these reasons, the thesis will focus on computational aspects of MPC algorithms, proposing a fast method that enables the control strategy to be implemented even in low-cost DSP platforms.

On the other hand, a different approach is investigated, spurred by the ongoing digital revolution. The growing importance of data in the digital area is investigated in the field of electric drives. In this direction, data driven control methods are analyzed. This paradigm represents the counterpart of model-based methods, which exploit a model of the system to design a controller. Conversely, The data-driven approach, aims to build a control algorithm using only collected input and output data from the system to be controlled. These strategies go to the direction of the aforementioned smart-inverter, where automatic procedures can be designed to self-build the control of the motor without having knowledge of its parameters, geometry or structure.

These two different approaches, which are deeply investigated, highlighting the computational aspect of the presented strategies.

1.4 OUTLINE OF THE THESIS

In the following, the contents of the each Chapter are briefly described:

CHAPTER 2 In this chapter, fundamentals components governing an electrical drive are presented. The PMSM working principles are reported. The basic transformations from stationary to rotating reference frame and vice versa of a three-phase system of quantities are defined, together with the power converter description and modulations strategies;

CHAPTER 3 This part introduces the basic theory concepts of model predictive control as a quadratic programming optimal control problem, with a brief overview of main strategies adopted in electric drives;

CHAPTER 4 In this chapter it is presented some fundamental concepts of the investigated data-driven approach to control. In particular, the behavioral system theory is briefly described

CHAPTER 5 This chapter deals with a presentation of some model order reduction techniques, which will be useful to join the data-driven control method with a proposed fast quadratic programming solver.

CHAPTER 6 In this chapter, the MPC paradigm is applied to the current control of synchronous motor drives. The intent is to compare the velocity form and the MPC with disturbance observer. A theoretical analysis of the MPC coupled with disturbance observers and the equivalence between these formulations and the velocity form is presented.

CHAPTER 7 This chapter presents an effective method for solving MPC for the current control loop of synchronous motor drives with input constraints. The related quadratic programming problem requires an iterative solver to find the optimal solution. The solver is deeply illustrated, showing its feasibility for real-time applications in the microseconds range by means of experimental tests.

CHAPTER 8 This chapter exploits the concept presented in Chapter 4 with the aim of proposing a potential transition from model-based towards data-driven optimal control strategies. As starting point, the MPC paradigm is considered. The, a complete data-driven approach, named Data EnablEd Predictive Control (DeePC) is presented. The theory behind these techniques is reviewed and design applied for the first time to the design of the current controller of synchronous permanent magnet motor drives.

CHAPTER 9 The aim of this chapter is to propose a method for enabling the real-time implementation of the DeePC control algorithm, exploiting the Quadratic Programming (QP) solver formulation presented in Chapter 7 and the model order reduction method presented in Chapter 5.

2

PERMANENT MAGNET SYNCHRONOUS MOTORS

The fundamental equations of a permanent magnet synchronous motor are presented. Linear current-flux relations are considered. Then, the three-phase power converter is introduced, together with the concept of space vector modulation.

2.1 ELECTRIC DRIVES

An Electrical Machine (EM) is a machine which at least one source of energy (received or delivered) are of electrical nature. In particular, the interest of this work is concentrated on those machines that convert the electrical energy into mechanical energy (and vice versa): they are called *electro-mechanical machines*. In the most cases, these machines use magnetic field to convert the electric energy into mechanical one, exploiting the physical principle described by Faraday's law. This principle is known as *electro-mechanical energy conversion.* An *electric drive* is defined as a drive that uses an electric actuator, i.e. a electric machine, usually in operation from motor, in which a conversion of energy from electric to mechanical takes place. In the past, EMs were usually employed as motor or generator at nearly constant speed, imposed by the grid frequency. The introduction of the concept of electric drives raised from the necessity of controlling the EMs for industrial applications. The control of these devices is performed by manipulating measured quantities, i.e. voltages, currents, speed and rotor position. This opportunity is made possible thank to the presence of a particular equipment that interposes the source of electrical energy (usually the grid) and the drive, which are the *static converter* with a power electronics devices (such as diodes, thyristor, transistor etc..). A principle scheme of the physical system is presented in Figure 2.1, where the scheme of a common control architecture is reported. The double arrow represents the energy direction, which instantaneously can flow from the electric energy source to the load and vice-versa. The saturation block considers the physical limits, avoiding overstressing the system.

2.2 PERMANENT MAGNET SYNCHRONOUS MACHINES

In this section the development of the mathematical model of a Permanent Magnet Synchronous Motor (PMSM) in the stator and rotor reference frame is derived. The permanent magnet synchronous motor is a three phase EM with windings displaced 120° electrical

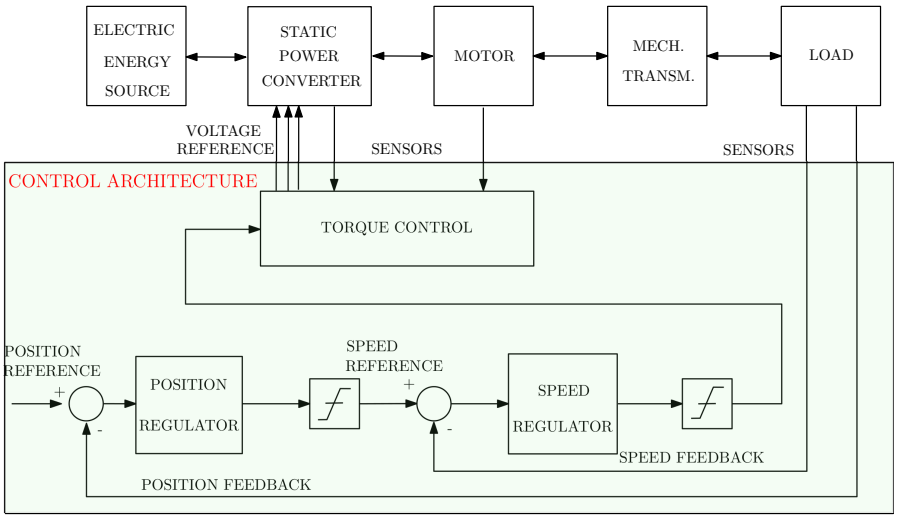


Figure 2.1: Scheme of electric drives with a common control architecture.

degrees in space. Since in general a PMSM is a non-linear, strong coupling system, electromagnetic relations are very complicated. In order to simplify the mathematical model of PMSM, some assumptions are made as follows:

Assumption 1 *Three-phase stator windings are symmetrical and difference of each phase is* 120° *electric angle;*

Assumption 2 *Magnetic saturation, eddy current loss and hysteresis loss are neglected;*

Assumption 3 *No load EMF of stator windings is sine wave;*

Assumption 4 *Stator winding current produces only sine distribution of magnetic potential in the air gap, ignoring the high-order harmonic of magnetic field;*

Assumption 5 *Conductivity of permanent magnet is zero.*

Previous assumptions help to derive an electro-mechanical model of the machine, whose complexity is a trade-off between the accuracy of the model and computational complexity. The coordinate transformations of three-phase magnitudes commonly used in the study and design of control strategies in PMSM is shown in Figure 2.2.
θ_r and ω_r denote respectively the angular mechanical position and velocity of the rotor respect to the reference on the phase a on the stator. The machine sketched in Figure 2.2 has a single pole pair for simplification, but making use of a general number of pole pairs p, the electrical rotor position and speed id computed as follow:

$$\begin{aligned} \theta_e &= p\theta_m \\ \omega_e &= p\omega_m \end{aligned} \tag{2.1}$$

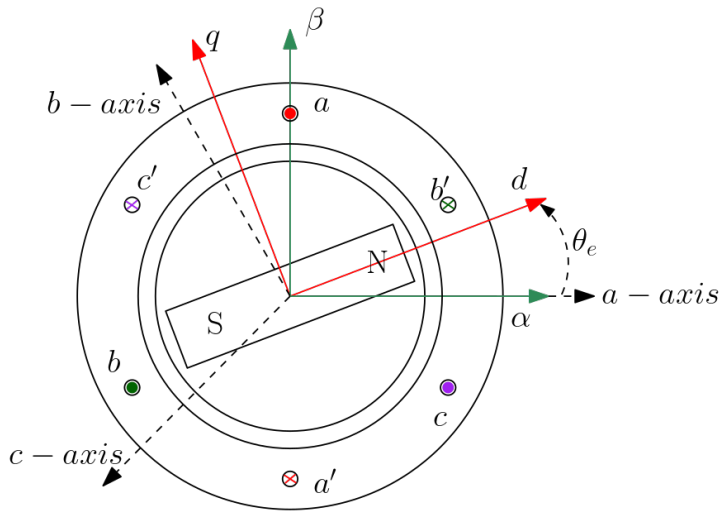


Figure 2.2: Stator (αβ) and rotor (dq) reference frame applied to a PMSM.

2.2.1 *Stator Reference Frame*

In general, a physical three phase winding system distributed with 120° between each phase can be simplified with a linear transformation, leading to a two phase system 90° shifted (see Appendix A). The transformation from three-phase to two-phase quantities can be written in matrix form as[1]:

$$\begin{bmatrix} S_\alpha \\ S_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} \tag{2.2}$$

$$S_{\alpha\beta} = T_{abc/\alpha\beta} S_{abc}$$

where S_a, S_b, S_c denote a general signal in the original 3-phase reference frame, while S_α, S_β are the two-phase orthogonal components after the transformation. The choice of the constant $\frac{2}{3}$ in (2.2) is intended for maintaining unaltered the signal's magnitude across the transformation, although from a power perspective, in order to be consistent, the constant need to be added as a reciprocal factor. The inverse relationship is written as:

$$\begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} S_\alpha \\ S_\beta \end{bmatrix} \tag{2.3}$$

$$S_{abc} = T_{\alpha\beta/abc} S_{\alpha\beta}$$

Transformation (2.2) and (2.3) are also known as Forward and Inverse *Clarke-transformations*.

1 Here it assumed that the windings are star connected and the system is balanced, i.e. the zero sequence component is null.

2.2.2 *Rotor Reference Frame*

The idea of the rotor reference frame transformation is to attach the rotating time-dependent signals in the $\alpha\beta$- coordinate, to a system which forms a rigid body with the machine rotor. In order to obtain this, the variables are transformed into a reference frame rotating at the electrical angular speed ω_e, thus fundamental frequency signals component and inductances will no longer depend on rotor position, becoming constant quantities. Since the transformation arises in an ideal context, where all the original signals are purely sinusoidal, the relationship between rotor and stator reference frames is described as follow:

$$\begin{bmatrix} S_d \\ S_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} S_\alpha \\ S_\beta \end{bmatrix} \tag{2.4}$$
$$S_{dq} = T_{\alpha\beta/dq} S_{\alpha\beta}$$

The elimination of position dependency from the machine variables is the main advantage. The inverse rotation, to transform from the rotating to the stationary reference frame is straightforward:

$$\begin{bmatrix} S_\alpha \\ S_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} S_d \\ S_q \end{bmatrix} \tag{2.5}$$
$$S_{\alpha\beta} = T_{dq/\alpha\beta} S_{dq}$$

Transformation (2.4) and (2.5) are also known as Forward and Inverse *Park-transformations*.

2.2.3 *Voltage Balance*

The stator windings consist of individual coils, which are connected and wounded in different slots. With previous assumptions, the coils are assumed to be sinusoidally distributed along the stator circumference. In Figure 2.2, each phase is depicted as single coil together with their resultant magnetic axes. The electrical dynamic equations of the three-phase system can be written as follow[2]:

$$\underbrace{\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}}_{u_{abc}} = \underbrace{\begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix}}_{R} + \frac{d}{dt} \underbrace{\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix}}_{\lambda_{abc}} \tag{2.6}$$

where u_a, u_b, u_c are the terminal voltages, i_a, i_b, i_c are the phase currents, $\lambda_a, \lambda_b, \lambda_c$ are the flux linkages and R_s is the winding resistance.

2 All the quantities are function of time t. The explicit time-dependence has been omitted to simplify the notation

The flux linkage of each phase presents two components: one due to the i_{abc} currents, and one due to the presence of permanent magnets placed in the rotor. Thank to the assumption of no magnetic saturation or eddy current effects, the superimposition principle holds, and the following relation can be written[3]:

$$\begin{bmatrix}\lambda_a\\ \lambda_b\\ \lambda_c\end{bmatrix} = \underbrace{\begin{bmatrix} L_a & M_{ab} & M_{ac}\\ M_{ba} & L_b & M_{bc}\\ M_{ca} & M_{cb} & L_c\end{bmatrix}}_{L_{abc}} \underbrace{\begin{bmatrix} i_a\\ i_b\\ i_c\end{bmatrix}}_{i_{abc}} + \frac{d}{dt}\underbrace{\begin{bmatrix}\lambda_{mg,a}\\ \lambda_{mg,b}\\ \lambda_{mg,c}\end{bmatrix}}_{\lambda_{mg,abc}} \tag{2.7}$$

where L_a, L_b, L_c and $M_{ij}, i = a, b, c, j = a, b, c$ are the self and mutual inductances respectively. $\lambda_{mg,i}, i = a, b, c$ represents flux linkage with each stator phase due to the magnets placed in the rotor. The back-Electro Motive Force (back-EMF) are defined as follow:

$$\begin{aligned} e_a &= \frac{d}{dt}\lambda_{mg,a} = -\Lambda_{mg}\omega_e \cos(\theta_e - \frac{\pi}{2})\\ e_b &= \frac{d}{dt}\lambda_{mg,b} = -\Lambda_{mg}\omega_e \cos(\theta_e - \frac{\pi}{2} - \frac{2\pi}{3})\\ e_c &= \frac{d}{dt}\lambda_{mg,c} = -\Lambda_{mg}\omega_e \cos(\theta_e - \frac{pi}{2} - \frac{4\pi}{3}) \end{aligned} \tag{2.8}$$

Combining (2.7) and (2.8), in matrix form the voltage balance becomes:

$$u_{abc} = Ri_{abc} + L_{abc}\frac{d}{dt}i_{abc} + \frac{d}{dt}L_{abc}\, i_{abc} + e_{abc} \tag{2.9}$$

By applying the transformation (2.2) to (2.9), the voltage equation in the $\alpha\beta$reference frame is:

$$u_{\alpha\beta} = Ri_{\alpha\beta} + \frac{d}{dt}L_{\alpha\beta}\frac{d}{dt}i_{\alpha\beta} + e_{\alpha\beta} \tag{2.10}$$

where $L_{\alpha\beta} = T_{abc/\alpha\beta}L_{abc}T^{-1}_{abc/\alpha\beta}$.

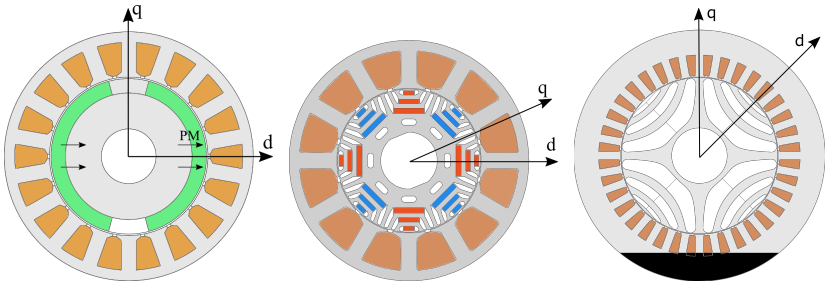
The $\alpha\beta$ transformation can be generalized to the rotating orthogonal reference frame dq with the direct d and quadrature q axes presented in (2.4). The angular position of the dq reference frame is defined θ_e, which is the angle between the d-axis of the rotating reference frame and the a-axis of the three-phase system. By applying (2.4) with $\theta = \theta_e$ to (2.10) the mathematical model of a linear synchronous machine n the dq reference frame is obtained:

$$u_{dq} = Ri_{dq} + L_{dq}\frac{d}{dt}i_{dq} + \omega_e J L_{dq} i_{dq} + e_{dq} \tag{2.11}$$

where:

$$u_{dq} = \begin{bmatrix}u_d\\ u_q\end{bmatrix},\ i_{dq} = \begin{bmatrix}i_d\\ i_q\end{bmatrix},\ e_{dq} = \begin{bmatrix}0\\ \omega_e\Lambda_{mg}\end{bmatrix} L_{dq} = \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} \tag{2.12}$$

3 Further details on flux-current relations and self and mutual inductance expressions can be found in [10].



(a) SPM (2 poles). (b) IPM (4 poles). (c) SyRM (4 poles).

Figure 2.3: Example of cross section of the three different types of synchornous motors.

The inductances matrix becomes constant due to the new reference frame which rotates integral with the rotor. It is implicitly considered that the dq inductances are directly derived transforming opportunely the matrix L_{abc} in (2.7). The inductances matrix loses the position dependency then they become time invariant:

$$L_{dq} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \tag{2.13}$$

Rewriting the (2.12) per components it can be observed the dq flux components:

$$\begin{aligned} u_d &= R_s i_d + L_d \frac{d}{dt} i_d - \omega_e \underbrace{L_q i_q}_{\lambda_q} \\ u_q &= R_s i_q + L_q \frac{d}{dt} i_q + \omega_e (\underbrace{L_d i_d + \Lambda_{mg}}_{\lambda_d}) \end{aligned} \tag{2.14}$$

The rotational reference frame imposed by (2.5) is synchronous with the magnetic flux produced by the rotor and all the flux contribution of the permanent magnet lies on the d-axis as stated in (2.14). This concept is at the basis of the Field Oriented Control (FOC).

In general, if iron saturation effects are considered, the flux linkage becomes a function of the stator currents, and the relations become nonlinear. See for instance [11] for further details.

2.3 MECHANICAL MODEL AND ELECTROMAGNETIC TORQUE

The electromagnetic torque expression of a synchronous motor expressed in the dq rotating reference frame is [12]:

$$\tau = \frac{3}{2} p (\lambda_d i_q - \lambda_q i_d) + \frac{W'_{fld}}{\theta_m} \tag{2.15}$$

W'_{fld} is the magnetic coenergy, which is a function of currents and mechanical angle. It includes all the effects which are not consid-

ered at the fundamental frequency model (i.e., for instance, cogging torque). Its definition is

$$W'_{fld}(i, \theta_m) = i\lambda - W_{fld}(\lambda, \theta_m) \tag{2.16}$$

where W_{fld} is the magnetic energy. The first term of (2.15) returns the average electromechanical torque generated by the motor, assuming an integral–slot winding motor. Finally, it is worth noting that (2.15) describes the torque production even in presence of the magnetic saturation and, in case of a linear motor, it can be simplified using the motor inductances as:

$$\tau = \frac{3}{2} p(\lambda_{mg} i_q + (L_d - L_q) i_d i_q) \tag{2.17}$$

Two different contributions can be recognized in equation (2.17), depending on the magnetic circuit that results from stator and rotor of the machine. The first one represents the torque related to the interaction between the rotor permanent magnets flux and the stator current. The second one is the reluctance torque which implies only stator–generated field. Anisotropic machines present different magnetic paths between d and q axis. In an Interior Permanent Magnet (IPM), magnets are placed inside the rotor, and for d-axis the inductance results in general lower than the inductance of the q-axis, due to the fact that in q-axis flux path pass through iron that has a higher value of permeability. The Synchronous Reluctance Motor (SyRM) is characterized by the lack of permanent magnets then the generated torque is based exclusively on the reluctance principle. In case $L_d = L_q$, the machine reduces to an Surface Permanent Magnet (SPM) and the reluctance contribution to the torque is zero. Here the rotor and the magnet displacement are designed in order to have the same magnetic structure with respect the stator. An example of these three different motor geometries are reported in Figure 2.3 Finally, the mechanical expression of the motor, including the load, is:

$$\tau = \tau_L + B_m \omega_m + J_m \frac{d}{dt} \omega_m \tag{2.18}$$

where B_m and J_m are the viscous friction coefficient of the system and the moment of inertia of the system, respectively. The load torque τ_L is independent from position and motor speed.

2.4 THREE-PHASE INVERTER

The power electronics module represents the fundamental component of an AC drive. In fact, it allows the efficient transformation of DC quantities, from DC power source (e.g. vehicle battery), to AC quantities which are suitable for operating the 3-phase electrical machine connected as a load. The use of a power converter allows the

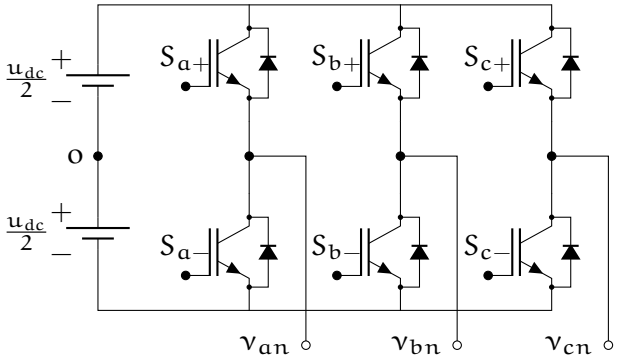


Figure 2.4: Schematic representation of a three-phase voltage source inverter.

motor to be controlled for variable speed applications, with high precision, fast dynamics position and torque control. In Figure 2.4 it is represented a basic scheme of a three-phase voltage source converter. The DC-link is represented by two series-connected constant voltage sources, where large capacitances are often used to maintain a constant valued of the DC voltage. The central point O is conventionally adopted as reference point. The six electronic controlled valves can be either turned on and off via a proper external signal. In particular, in the Figure 2.4 are represented the commonly used Insulated-Gate Bipolar Transistor (IGBT), which are suitable for medium power applications (up to some kW) and relatively high switching frequency (up to 15 kHz) [13], but other power electronics components can be adopted accordingly to the application. When the switch is turned on by the drive circuit, the IGBT allows the current flowing in the arrow direction. The reverse conducting characteristic is achieved by adding an anti-parallel diode to the IGBT. The two switches connected to the same phase, respectively the high side and the low side driver, form the leg of the inverter and allow the phase to be alternatively connected to $v_{iO} = +\frac{V_{dc}}{2}$ or $v_{iO} = -\frac{V_{dc}}{2}$, with $i = a, b, c$. The switching can be open and closed alternatively, and it is avoided the simultaneous conduction of drivers belonging to the same leg, otherwise this would lead to a short circuit of the DC power source. In the two-level configuration, the total number of switch configurations are 8 and they are resumed in Table 1. Applying the definition of a space vector[4], 6 of the 8 states are represented by six state vectors of amplitude $\frac{2}{3}\ u_{dc}$ and offset by $\frac{\pi}{3}$ respectively, describing the vertices of a hexagon centered in the origin of the axes in the stationary reference frame (also known as $\alpha\beta$ plane), see Figure 2.5.

The six-step mode operation of the two-level inverter consists in controlling the three upper switches cyclically, so that their state is 1 for one half-period and then 0 in the remaining half-period, with a phase shift between the states of the inverter branches of $\frac{2\pi}{3}$.

4 See Appendix A.

Table 1: Total number of operation of the proposed QP method in the worst case scenario.

STATE	S_1	S_2	S_3	v_{an}	v_{bn}	v_{cn}
s_0	0	0	0	0	0	0
s_1	1	0	0	$\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$
s_2	1	1	0	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{2}{3}$
s_3	0	1	0	$-\frac{1}{3}$	$\frac{2}{3}$	$-\frac{1}{3}$
s_4	0	1	1	$-\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
s_5	0	0	1	$-\frac{1}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$
s_6	1	0	1	$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{1}{3}$
s_7	1	1	1	0	0	0

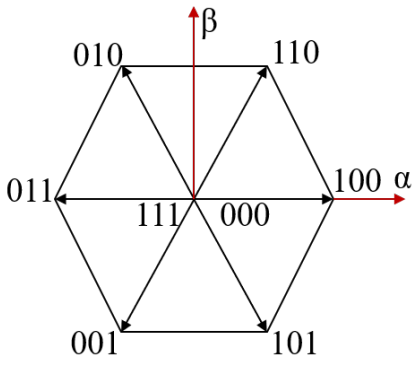
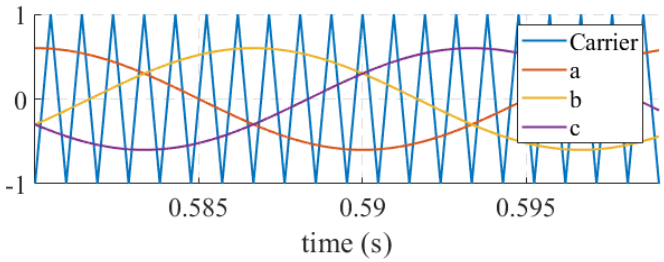


Figure 2.5: Space vectors produced by the 8 possible states of a three-phase two-level inverter.

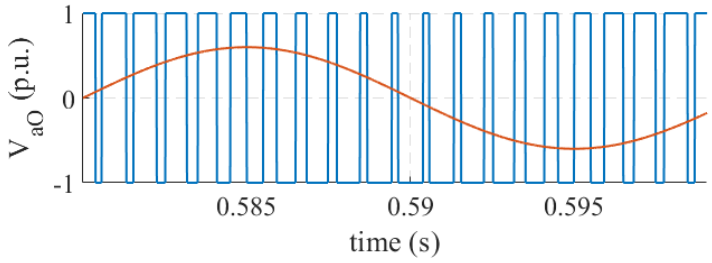
The solution to the problem of adjusting the amplitude of the output voltage of the inverter, together with an improvement of the low-frequency harmonic content, is obtained by introducing a modulation technique, commonly known as Pulse-Width-Modulation (PWM). An extensive description of different modulation techniques can be found in [14].

The simplest modulation method is called Sinusoidal Pulse Width Modulation Sinusoidal Pulse-Width-Modulation (SPWM), shown in Figure 2.6. A duty cycle is generated by the feedback controller, which expresses the desired voltage reference to apply at the machine terminal. The duty cycle of each phase is compared with a carrier, whose frequency is much higher than the frequency of the fundamental voltage component.

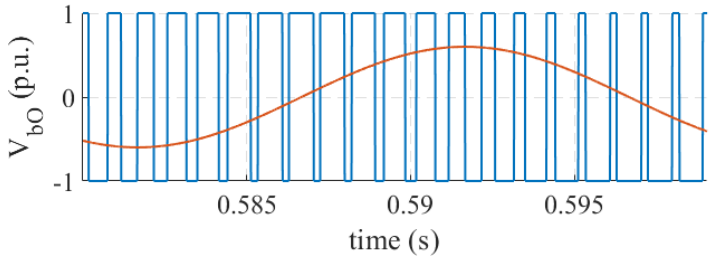
It is possible to generate within each interval $T_s = \frac{1}{f_s}$, with f_s the carrier frequency, a voltage u_{aO} with an average value equal to the one assumed by the voltage reference in the same time interval, making u_{aO} have its positive value $+\frac{u_{dc}}{2}$ for a portion T_{on} of T_s, and its negative value $-\frac{u_{dc}}{2}$ for the remaining portion T_{off}. Considering an



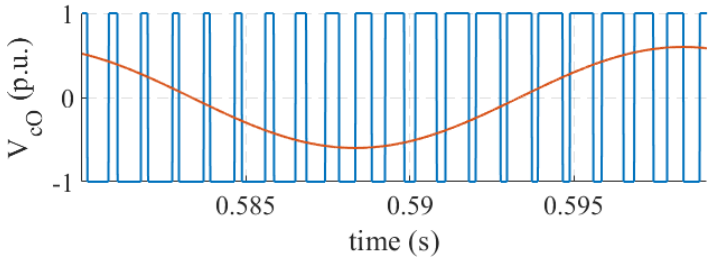
(a) Carrier and reference voltage signals (sinusoids) with aplitude 0.6 p.u.



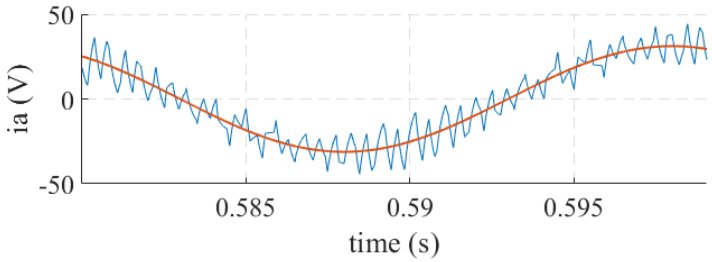
(b)



(c)

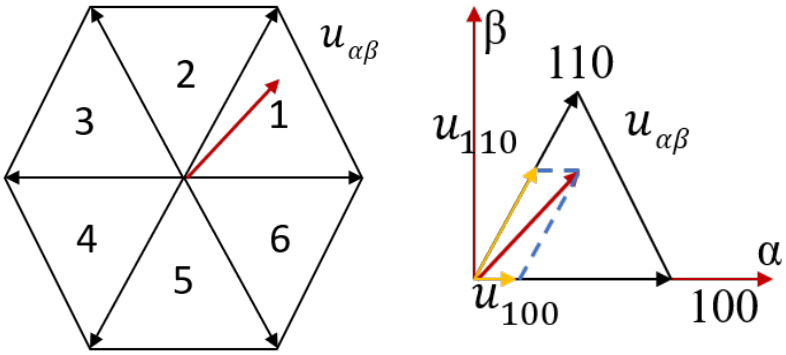


(d)



(e) Current of phase a.

Figure 2.6: SPWM example: the three sinusoids in Figure 2.6(a) are compared with the triangular carrier. Figure 2.6(b),Figure 2.6(c) and Figure 2.6(d) are the voltage between each terminal of the inverter with respect the point O. It also plotted the fundamental voltage component, and in Figure 2.6(e) it is reported the a-phase current, assuming balanced three phase motor.



(a) Sectors. (b) Vector projection in the first sector.

Figure 2.7: SVPWM sectors and example of duty cycles computation.

interval k, the resulting duty cycle for the switch of an inverter leg is computed as:

$$\delta_k = \frac{u_k}{u_{dc}} + 0.5 \tag{2.19}$$

where u_k is the voltage reference sampled at instant k. In case of a three-phase sinusoidal voltage reference[5]

$$\begin{aligned} u_a &= U_m \cos(\omega t + \theta_0) \\ u_b &= U_m \cos(\omega t + \theta_0 - \frac{2\pi}{3}) \\ u_b &= U_m \cos(\omega t + \theta_0 - \frac{4\pi}{3}) \end{aligned} \tag{2.20}$$

the duty cycles (2.19) represented in Figure 2.6 for each phase leg become:

$$\begin{aligned} \delta_{a,k} &= \frac{u_{\alpha,k}}{u_{dc}} + 0.5 \\ \delta_{b,k} &= \frac{1}{u_{dc}}(-0.5u_{\alpha,k} + \frac{\sqrt{3}}{2}u_{\beta,k}) + 0.5 \\ \delta_{c,k} &= \frac{1}{u_{dc}}(-0.5u_{\alpha,k} - \frac{\sqrt{3}}{2}u_{\beta,k}) + 0.5 \end{aligned} \tag{2.21}$$

being u_α, u_β the components related to the space vector generated by (2.20) (see Appendix A). The maximum modulus for the space vector is equal to $\frac{u_{dc}}{2}$ with a duty cycle equal to one, which coincides with the maximum value of the phase voltages.

For having higher values of the fundamental component of the phase voltage, one can in principle add a homopolar component δ_0 to the sinusoidal duty cycles. In this way the maximum voltage delivered by the inverter becomes $\frac{u_{dc}}{\sqrt{3}}$.

5 In this work it is assumed that the load is always represented by a three-phase symmetric RL load with balanced back-EMF.

An alternative way to handle the voltage reference realization is the so called Space Vector Pulse-Width-Modulation (SVPWM). The objective of SVPWM technique is to approximate the reference voltage vector using the eight switching patterns. The reference vector is then synthesized using a combination of the two adjacent active switching vectors and one or both of the zero vectors. The illustrative procedure is shown in Figure 2.6, where the hexagon is divided in 6 sector. The voltage vector components u_α and u_β are the input of the SVPWM algorithm, which firstly identifies the sector of interest. Then, as shown in Figure 2.7(b) for the first sector, the duty cycles are obtained by the combination of the voltage vector 100 and 110 as in the following:

$$\begin{aligned} \|u_{100}\| &= \frac{\|u_{\alpha\beta}\|}{\sin(\frac{2\pi}{3})} \sin(\frac{\pi}{3} - \angle u_{\alpha\beta}) \\ \|u_{110}\| &= \frac{\|u_{\alpha\beta}\|}{\sin(\frac{2\pi}{3})} \sin(\angle u_{\alpha\beta}). \end{aligned} \tag{2.22}$$

Then, it is necessary to calculate the T_{on} time of the selected inverter states within the sampling period T_s, yielding to:

$$\begin{aligned} T_{100} &= T_s \frac{3}{2u_{dc}} \|u_{100}\| \\ T_{110} &= T_s \frac{3}{2u_{dc}} \|u_{110}\| \end{aligned} \tag{2.23}$$

The remaining time of the entire period T_s, the null vector (000 or 111) is applied: $T_{000} = 1 - T_{100} - T_{110}$. If the sum $T_{100} + T_{110}$ (as an example) exceeds 1, the duty cycle saturates and the inverter reaches the over-modulation range. Here it is possible to implement different techniques which allow the transition from the over-modulation region to the six-step operation mode, i.e., where the voltage reference are the 6 active states of the converter. As an example, see [15]. For a complete description of different strategies for implementing the SVPWM, see [14]. Notice that, even if the SPWM with third harmonic injection and the SVPWM could be considered different strategies, it has been shown the equivalence between the two approaches, under certain circumstances [16].

3 MODEL PREDICTIVE CONTROL

This chapter presents a very brief introduction to the principles of Model Predictive Control (MPC). In particular, the problem family of Linear Time Invariant (LTI) systems with linear constraints is considered.

3.1 KEY INGREDIENTS OF MPC

MPC is a form of control in which the current control action is obtained by solving on-line, at each sampling instant, a finite horizon optimal control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence and the first in this sequence is applied to the plant [17]. The MPC provides a way to design state-feedback and tracking control laws, solving an optimization problem. Some basic notions of state-feedback controls are reported in Appendix B. While its idea is relatively old [6], [7], only the recent development of hardware with higher computational powers enables its implementation in systems with fast dynamics, making these algorithms suitable for industrial implementations in applications such as electrical drives. The most important characteristics of MPC, which make it flexible and suitable for a large number of applications, are the possibility to:

- formulate the control as an implementation problem, where different objectives can be considered;
- explicitly include in the control problem formulation state and input constraints;
- synthesize the model-based controller empirically by means of simple experiments.

The main ingredients of an MPC algorithm are:

- a model of the process, usually in discrete-time domain;
- input, output, and state constraints;
- a cost function to be minimized;
- an optimization solver;
- the so called Receding Horizon (RH) principle (see Figure 3.1), which can be explained as follows. At any time instant, based on

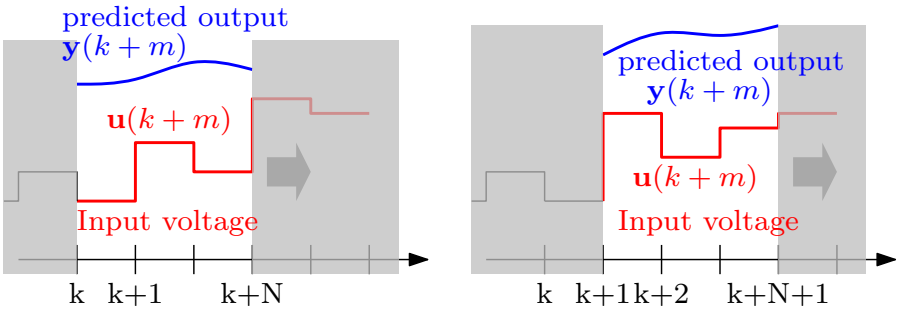


Figure 3.1: Receding horizon principle.

the available process information, solve the optimization problem is solved with respect to the future control sequence, but only the first input is effectively applied to the system. Then, at the next time instant, a new optimization problem is solved. It is worth noting that, by means of RH, a time invariant feedback control law is obtained even if a finite horizon optimization problem is solved at any time instant.

3.2 QUADRATIC OPTIMIZATION PROBLEM

The MPC, in general, belongs to the family of quadratic optimizations problem. In the following, some basic concepts of constrained optimization are introduced and the family of problems that will be considered is described. In particular, *convex* optimization problems are considered.

A set C is convex if the line segment between any two points in C lies in C, i.e., if for any $x_1, x_2 \in C$, and any θ with $0 \leqslant \theta \leqslant 1$, it holds that

$$\theta x_1 + (1-\theta)x_2 \in C.$$

In Figure 3.2(a) it is reported an example of a convex and non-convex domain. A convex optimization problem is one of the forms

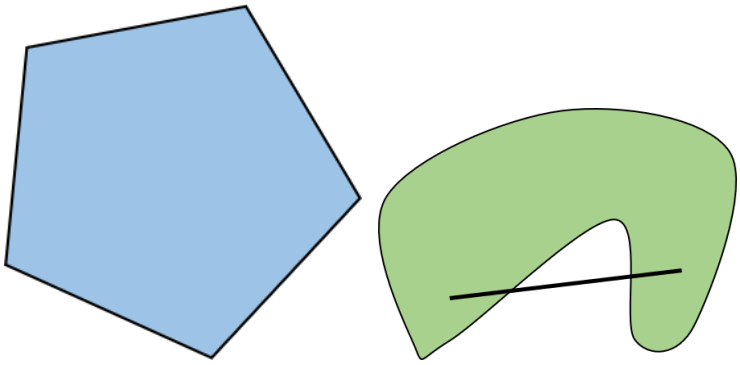
$$\begin{aligned} &\text{minimize} \;\; f_0(x) \\ &\text{subject to } f_i(x) \leqslant 0, \; i = 1, \dots, j \\ &\qquad\qquad a_i^T x = b_i, \; i = 1, \dots, k \end{aligned} \tag{3.1}$$

where $f_0, \dots, f_j : \mathbb{R}^n \to \mathbb{R}$ are convex, i.e. satisfy

$$f_i(\alpha x + \beta y) \leqslant \alpha f_i(x) + \beta f_i(y)$$

for all $x, y \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}^n$ with $\alpha + \beta = 1, \alpha \geqslant 0, \beta \geqslant 0$. The set of points for which the objective and all constraint functions are defined,

$$D = \bigcap_{i=0}^{j} \operatorname{dom} f_i \cap \bigcap_{i=1}^{k} \operatorname{dom} h_i$$



(a) Convex set. (b) Non convex set.

Figure 3.2: An example of simple convex and nonconvex set. Figure 3.2(a) the hexagon, which includes its boundary, is convex. Figure 3.2(b) the kidney shaped set is not convex, since the line segment between the two points is not contained in the set.

and it is called the domain of the optimization problem. A point $x \in D$ is said to be feasible if it satisfies the constraints. The problem (3.1) is said to be feasible if there exists at least one feasible point, and infeasible otherwise. The set of all feasible points is called the feasible set or the constraint set. The optimal value p^* of the problem (3.1) is defined as

$$p^* = \inf\{f_0(x) | f_i(x) \leqslant 0,\ i = 1, .., j,\ h_i(x) = 0,\ i = 1, ..., k\}.$$

The convex optimization problem (3.1) is called a Quadratic Programming (QP) if the objective function is (convex) quadratic, and the constraint functions are affine [1]. In a quadratic program, a convex quadratic function is minimized over a polyhedron. This constrained optimization problem does not admit an explicit solution, but it can be solved by means of QP solvers based on iterative methods.

3.3 MPC FORMULATION

Firstly, consider the LTI system:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \quad x_0 = x(0) \end{aligned} \tag{3.2}$$

where the state $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ is the control variable, and $y \in \mathbb{R}^p$ is the output variable. The aim is to find the sequence of control inputs that steers the system from its initial state x_0 at time sample k (it can be assume $k = 0$ as starting point without loss of generality) to the origin. This is known as *regulation problem*. The optimization problem

1 A set $C \subseteq \mathbb{R}^n$ is affine if the line through any two distinct points in C lies in C, i.e., if for anu $x_1, x_2 \in C$ nd $\theta \in \mathbb{R}$, $\theta x_1 + (1-\theta)x_2 \in C$. In other words, C contains the linear combination of any two points in C

consists of computing, at each time instant k, the control sequence $u(k), u(k+1), ..., u(k+N)$ that minimizes the finite horizon quadratic cost function:

$$\begin{aligned}
\underset{u(k+i),x(k+i)}{\operatorname{argmin}} \quad J(x,u,k) &= \sum_{i=0}^{N-1} \|x(k+i)\|_Q^2 + u(k+i)\|_R^2 + \|x(k+N)\|_S^2 \\
\text{subject to } x(k+i+1) &= Ax(k+i) + Bu(k+i)\ i = 0, ..., N-1 \\
y(k+i) &= Cx(k+i)\ i = 0, ..., N-1 \\
x_0 &= x(k) \\
u(k+i) &\in \mathbb{U}\ i = 0, ..., N-1 \\
x(k+i) &\in \mathbb{X}\ i = 0, ..., N-1 \\
y(k+i) &\in \mathbb{Y}\ i = 0, ..., N-1
\end{aligned} \tag{3.3}$$

where $Q \geqslant 0$, $R > 0$, $S \geqslant 0$ are matrices of suitable dimensions.[2] In MPC, the positive integer N is usually called the prediction horizon length.

3.3.1 *Unconstrained Solution*

The cost function is based on the predicted effects of the control sequence, i.e. on the predicted future states. A closed-form expression of the control law can be obtained if limitations of x, u, y are neglected. Assuming this case, one way to find the solution of problem (3.3) can be derived recalling the expression of free and forced response of system (3.2):

$$x(k+i) = A^i x(k) + \sum_{j=0}^{i-1} A^{i-j-1} Bu(k+j), \qquad i > 0. \tag{3.4}$$

2 The relations $\geqslant$ and $>$ indicates a semi-positive definite and positive definite matrix, respectively.

where the first term of (3.4) is the free response due to non-null initial state condition, while the second one is the forced response when the input u is applied to the system. Letting

$$X(k) = \begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N-1) \\ x(k+N) \end{bmatrix}, \mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^N \end{bmatrix}, U(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-2) \\ u(k+N-1) \end{bmatrix},$$

$$\mathcal{B} = \begin{bmatrix} B & 0 & 0 & \cdots & 0 & 0 \\ AB & B & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ A^{N-2}B & A^{N-3}B & A^{N-4}B & \cdots & B & 0 \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \cdots & AB & B \end{bmatrix}$$

it follows that

$$X(k) = \mathcal{A}x(k) + \mathcal{B}U(k). \tag{3.5}$$

Defining the matrices, with N block on the diagonal,

$$\mathcal{Q} = \begin{bmatrix} Q & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & 0 & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & 0 & \cdots & Q & 0 \\ 0 & 0 & \cdots & 0 & S \end{bmatrix}, \mathcal{R} = \begin{bmatrix} R & 0 & \cdots & 0 & 0 \\ 0 & R & \cdots & 0 & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & 0 & \cdots & R & 0 \\ 0 & 0 & \cdots & 0 & R, \end{bmatrix}$$

and observing that

$$\arg\min_{u()} J(x(k), u(), k) = \arg\min_{U} \tilde{J}(x(k), U(k), k) \tag{3.6}$$

where

$$\tilde{J}(x(k), U(k), k) = X^T(k)\mathcal{Q}X(k) + UT(k)\mathcal{R}U(k). \tag{3.7}$$

Recalling (3.5), it follows that

$$\begin{aligned} \tilde{J}(x(k), U(k), k) &= (\mathcal{A}x(k) + \mathcal{B}U(k))^T\mathcal{Q}(\mathcal{A}x(k) + \mathcal{B}U(k)) + U^T(k)\mathcal{R}U(k) \\ &= x^T(k)\mathcal{A}^T\mathcal{Q}\mathcal{A}x(k) + 2x(k)^T\mathcal{A}^T\mathcal{Q}\mathcal{B}U(k) + U^T(k)(\mathcal{B}^T\mathcal{Q}\mathcal{B} + \mathcal{R})U(k). \end{aligned} \tag{3.8}$$

Equation (3.8) has some important properties. The matrix $\mathcal{B}^T\mathcal{Q}\mathcal{B} + \mathcal{R}$ of the quadratic term is symmetric and positive-definite[3], since $\mathcal{R} > 0$.

3 A symmetric matrix M with real entries is positive-definite if the real number z^TMz is positive for every nonzero real column vector z

The minimum of (3.8) can be computed by setting to zero its gradient with respect to $U(k)$. Thanks to the property of positive-definiteness, the matrix $\mathcal{B}^T\mathcal{Q}\mathcal{B}+\mathcal{R}$ is invertible. It results the optimal control sequence

$$U^*(k) = -(\mathcal{B}^T\mathcal{Q}\mathcal{B}+\mathcal{R})^{-1}\mathcal{B}^T\mathcal{Q}\mathcal{A}x(k). \tag{3.9}$$

Note that this solution depends on the prediction of the state, along the considered horizon, based on the current state $x(k)$. In fact, defining

$$\mathcal{K} = (\mathcal{B}^T\mathcal{Q}\mathcal{B}+\mathcal{R})^{-1}\mathcal{B}^T\mathcal{Q}\mathcal{A} = \begin{bmatrix} \mathcal{K}(0) \\ \mathcal{K}(1) \\ \vdots \\ \mathcal{K}(N-1) \end{bmatrix}, \qquad \mathcal{K}(i) \in \mathbb{R}^{m\times n} \tag{3.10}$$

it is possible to write

$$U^*(k) = -\begin{bmatrix} \mathcal{K}(0) \\ \mathcal{K}(1) \\ \vdots \\ \mathcal{K}(N-1) \end{bmatrix} x(k) \tag{3.11}$$

or

$$u^*(k+i) = -\mathcal{K}(i)x(k), \qquad i = 0, 1, ..., N-1. \tag{3.12}$$

The resulting control law in (3.12) is an open loop solution, at least for $i > 0$. However, by applying the RH principle, the state feedback control law is obtained

$$u^{MPC}(k) = -\mathcal{K}(0)x(k). \tag{3.13}$$

It is worth noting that in the nominal case here considered, therefore in absence of disturbances or modeling errors, closed-loop and open-loop solutions coincide, while in the presence of disturbances or model uncertainties only the first solution of the sequence coincides.

3.3.2 *Tracking reference signals and disturbances*

The basic MPC formulation previously introduced is now extended to cope with tracking problems and systems affected by disturbances. Let the system be described by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Md(k) \\ y(k) &= Cx(k) + d(k) \end{aligned} \tag{3.14}$$

where $d \in \mathbb{R}^p$ is a disturbance, assumed to be known at the current time instant k and, possibly, in the future prediction horizon. the

objective is tracking the output variable reference signal y^{ref} in the future prediction horizon. It is worth noting that

$$\begin{aligned} x(k+i) &= A^i x(k) + \sum_{j=0}^{i-1} A^{i-j-1}(Bu(k+j) + Md(k+j)) \\ y(k+i) &= Cx(k+i) + d(k+i) \end{aligned} \tag{3.15}$$

and the quadratic cost function to be minimized includes a term weighting the future error variable

$$\begin{aligned} J(x(k), u(), k) = \sum_{i=0}^{N-1} \|y^{ref}(k+i) - y(k+i)\|_Q^2 + \|u(k+i)\|_R^2 + \\ + \|y^{ref}(k+N) - y(k+N)\|_S^2 \end{aligned} \tag{3.16}$$

Based on the previous formula, it is possible to write

$$Y(k) = \mathcal{A}_c x(k) + \mathcal{B}_c U(k) + \mathcal{M}_c D(k) \tag{3.17}$$

where

$$\mathcal{A}_c = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N-1} \\ CA^N \end{bmatrix}, D(k) = \begin{bmatrix} d(k) \\ d(k+1) \\ \vdots \\ d(k+N-1) \\ d(k+N) \end{bmatrix},$$

$$\mathcal{B}_c = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 & 0 \\ CAB & CB & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \cdots & CB & 0 \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CAB & CB \end{bmatrix}$$

$$\mathcal{M}_c = \begin{bmatrix} CM & 0 & 0 & \cdots & 0 & 0 \\ CAM & CM & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ CA^{N-2}M & CA^{N-3}M & CA^{N-4}M & \cdots & CM & 0 \\ CA^{N-1}M & CA^{N-2}M & CA^{N-3}M & \cdots & CAM & CM \end{bmatrix}$$

Now letting

$$Y^{ref}(k) = \begin{bmatrix} y^{ref}(k+1) \\ y^{ref}(k+2) \\ \vdots \\ y^{ref}(k+N-1) \\ y^{ref}(k+N) \end{bmatrix}$$

Again it is obtained that

$$\arg\min_{u()} J(x(k), u(), k) = \arg\min_{U} \tilde{J}(x(k), U(k), k) \tag{3.18}$$

where

$$\begin{aligned}\tilde{J}(x(k), U(k), k) &= (Y^{ref}(k) - Y(k))' Q (Y^{ref}(k) - Y(k)) + U'(k) \mathcal{R} U(k) = \\ &= (Y^{O}(k) - \mathcal{A}_c x(k) - \mathcal{B}_c U(k) - \mathcal{M}_c D(k))' Q (Y^{ref}(k) - \mathcal{A}_c x(k) - \\ &\mathcal{B}_c U(k) - \mathcal{M}_c D(k)) + U'(k) \mathcal{R} U(k)\end{aligned} \tag{3.19}$$

This cost function must be minimized, as before, with respect to the future values $u(k+i)$ of the control, and possibly under suitable constraints on x, u, y. It is worth mentioning that $\tilde{J}$ depends on the future reference signals and disturbances, which often result unknown. In these cases, it is reasonable assuming

$$y^{ref}(k+i) = y^{ref}(k), \qquad d(k+i) = d(k), \tag{3.20}$$

i.e., the reference is considered to be constant within the prediction horizon, as well as the disturbance.

3.3.3 *Constant Reference Tracking*

Assume that the reference signal is constant, i.e. $y^{ref}(k) = y^{ref}$. In this case, provided that $p \leqslant m$ and the system does not have any invariant zero, it is possible to compute a steady-state $(\bar{x}, \bar{u})$ such that

$$\begin{aligned}\bar{x} &= A\bar{x} + B\bar{u} \\ y^{ref} &= C\bar{x}\end{aligned} \tag{3.21}$$

Thus, the performance index to be minimized as

$$\begin{aligned}J(x(k), u(), k) = \sum_{i=0}^{N-1} &\|y^{ref}(k+i) - y(k+i)\|_Q^2 + \|u(k+i) - \bar{u}\|_R^2 + \\ &+ \|y^{ref}(k+N) - y(k+N)\|_S^2\end{aligned} \tag{3.22}$$

so that $J = 0$ for $y(k+i) = y^{ref}$ and $u(k+i) = \bar{u}$. Then, it also a common design choice to consider as decision variable the input increments $\Delta u(k+i) \triangleq u(k+i) - u(k+i-1)$. It is then easy to extend the previous considerations to this case. However, the computation of the steady-state pair $(\bar{x}, \bar{u})$, which leads to zero output error, cannot be performed in case of unknown disturbances and/or modeling errors. Hence, it is advisable to resort alternative formulations guaranteeing the incorporation of an integral action in the control law. This will be the focus of Chapter 6.

3.4 CONTROL HORIZON

A large value of N is often chosen to include in the prediction horizon all the main process dynamics, so that also the number of optimization variables $u(k)$, $u(k+1)$, ..., $u(k+N-1)$ is large at the price of an heavy computational load associated with the solution of the optimization problem to be solved on-line. For this reason, and in order to obtain a less aggressive control action , many industrial algorithms allow one to define a control horizon $0 < N_u < N$ and to consider the additional constraints

$$u(k+i) = u(k+i-1), \qquad i = N_u, ..., N-1 \tag{3.23}$$

or

$$\Delta u(k+i) = 0, \qquad i = N_u, ..., N-1 \tag{3.24}$$

when the system is described in incremental form. Correspondingly, the performance index to be considered, for example becomes

$$J(x(k), \Delta u(), k) = \sum_{i=0}^{N-1} \|y^{ref}(k+i) - y(k+i)\|_Q^2 + \sum_{i=0}^{N_u-1} \|\Delta u(k+i)\|_R^2 + \|y^{ref}(k+N) - y(k+N)\|_S^2 \tag{3.25}$$

and all the previous developments can be easily extended to include in the problem formulation this new set of constraints. Note that these constraints can be explicitly considered when computing the output prediction.

3.5 QUADRATIC PROGRAMMING SOLVERS

A rich set of good QP algorithms is available today. Different algorithms have been investigated, based on the so called Karush-Khun-Tucker optimality conditions of the problem (3.1) [18]. Regarding the specific problem of MPC, on one hand, a possible alternative to online optimization would be explicit MPC, which pre-solves the optimization problem offline [19]. In particular, the algorithm considers the set of all parameters of the problem, then it founds the control action for all the possible regions of such space. Explicit MPC allows one to solve the optimization problem off-line for a given range of operating conditions of interest. By exploiting multi-parametric programming techniques [20], explicit MPC computes the optimal control action offline as an explicit function of the state and reference vectors, so that online operations reduce to a simple function evaluation. Such a function is piece-wise affine in most cases, so that the MPC controller

maps into a lookup table of linear gains. However, as the dimension of parameters space grows in dimension, the logarithmic search time required in the online stage and polynomial memory occupancy make explicit MPC feasible only for small problems, with few inputs and a short horizon. As an example, the multi-parametric toolbox [21] is a popular MATLAB toolbox which implements various explicit methods, such as multi-parametric quadratic programming, that aim at providing approximation of the solution to the QP problems at the benefit of reduced offline and online complexity.

On the other hand, iterative methods solve QP problems online and most of them can be classified according to the way they handle the inequality constraints [22]. The most common methods are:

- Active Set (AS) methods start with a guess which inequality constraint will hold with equality (which is called active) at the optimal solution and solve the resulting equality constrained QP problem. If it turns out that this guess was not correct, it is updated until the optimal solution is found [23] [24].

- Interior Point (IP) methods, which are based on the idea of removing the inequality constraints and to penalizing constraint violations in the objective function. However, a non-quadratic penalty term (e.g. logarithmic) needs to be used. The resulting equality-constrained nonlinear programming problem is then solved by means of, for instance, Newton's method [18];

- Gradient Projection (GP) methods iteratively compute a step towards the solution of the unconstrained QP problem and then project this step onto the feasible set [25],[26].

When assessing suitability of a given MPC algorithm in general or for a specific MPC problem at hand, the following numerical aspects should be considered:

1. Type of inequality constraints, i.e., whether (3.1) includes only input or also output bounds, or even general constraints;

2. cost function type: if all matrices Q, R and S are positive definite, the resulting QP is strongly convex, which is a desirable feature for most MPC algorithms to work reliably;

3. System dynamics: if system dynamics are unstable, the QP problem formulation tends to become ill-conditioned;

Another criterion of paramount importance when choosing MPC algorithms is numerical performance in terms of computational speed. In particular, worst-case execution times are of highest importance as they ensure that the algorithm can run online reliably within each sampling period [22].

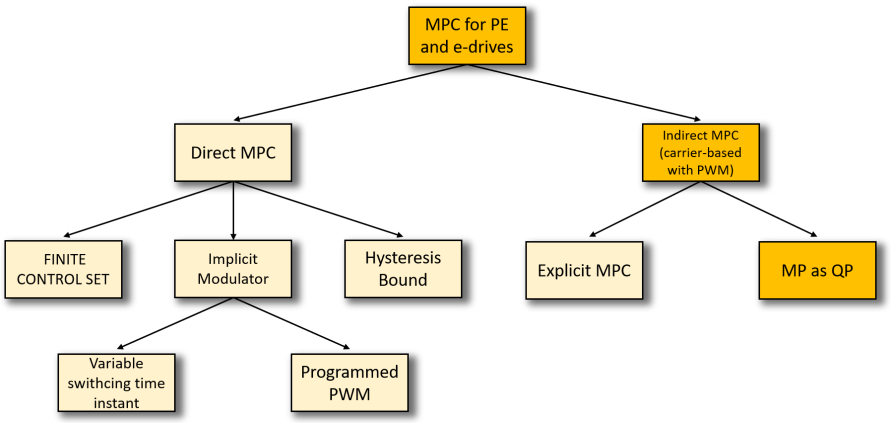


Figure 3.3: Classification method of MPC strategies for power electronics and electric drives applications [27].

3.6 MPC IN POWER ELECTRONICS AND ELECTRIC DRIVES

MPC has established itself as a promising control methodology in power electronics [27, 28]. Several variants of MPC have been developed and implemented in power converters used in applications such as electrical drives. MPC schemes can be classified into two main categories. On one hand, MPC is implemented as an indirect controller, i.e., the controller computes the modulating duty ratio which is fed into a modulator for generation of the switching commands. On the other hand, MPC can be designed as a direct controller exploiting the discrete nature of the power converter, thus dedicated modulator is not required. Consequently, the elements of the control input vector are the switching signals, implying that it is an integer vector.

The survey in [27], a well-detailed description of different MPCs formulation is provided. In particular, all different approaches in power electronics applications are summarized in Figure 3.3. Direct MPC schemes include controllers with reference tracking, hysteresis bounds and implicit modulator. Direct MPC with hysteresis bounds was the first rudimentary version of this type of controllers developed for power electronic converters. This algorithm employs hysteresis bounds within which the variables of interest, such as the stator currents. The second group of direct MPC strategies can be further divided into two subgroups. The first one includes methods that manipulate not only the switching signals, but also their application time in an attempt to emulate the behavior of Pulse-Width-Modulation (PWM) techniques. This is done by computing both the optimal switch positions and the associated duty cycles. The second group consists of direct MPC methods that are combined with programmed PWM, i.e., modulation methods that uses a fixed modulation interval. The switching pattern and the switching instants are computed offline based on some optimization criteria, such as

minimization of the current Total Harmonic Distortion (THD). Programmed PWM is implemented in the form of selective harmonic elimination [29–31], or optimized pulse patterns [32].

Direct MPC with reference tracking, also known as Finite Control Set (FCS)-MPC, is probably the most favored method in academia due to its well-reported advantages such as its intuitive design procedure and straightforward implementation [33]. In these methods, the converter DC-bus voltage is not modulated within one control period. Thus, the voltage that can be applied to the motor is selected among a finite set of candidates that depends on the power converter topology. In case of a three-phase two levels inverter, there are eight candidate voltages, which correspond to the eight possible switch configurations (see Chapter 2). The FCS-MPC offers several advantages. Firstly, the modulator removal eases the current regulation scheme, directly taking into account the discrete nature of a power converter. Secondly, the average switching frequency is reduced with respect to the PWM-based solutions. The FCS-MPC results, an effective control method for high power converters [13], where the minimization of switching losses can lead to an increase of the inverter efficiency. Finally, the MPC framework is suitable for including additional constraints, e.g. maximum output currents. The main drawback of using FCS predictive controllers is the computational burden that increases exponentially with the prediction horizon. A nondeterministic polynomial time-hard optimization problem has to be solved on-line at each control period. This represents a relevant issue for real-time applications. However, the availability of more powerful micro-controllers and more efficient solvers for the optimization task permit, nowadays, to overcome the issue.

The MPC-type which are considered in this work is highlighted in Figure 3.3. Recently, the combinatorial complexity has been severely reduced in FCS-MPC, with a consequent gain in its popularity [34]. However, regarding the MPC as QP problem, also known as Continuous Control Set (CCS)-MPC, implementing efficient algorithms which can solve the optimization problem in limited hardware at relatively high sampling frequency still represent a challenge. Some recent publications are going into this direction, addressing the problem of designing efficient methods for online CCS-MPC for motor control and embedded in a low-cost platform [26],[35]. Thus, in this work it will be presented an efficient custom QP solver for electric drives applications, whose computational complexity can be easily evaluated and the worst-case computational cost is uniquely determined.

4

DATA-DRIVEN CONTROL THEORY

This chapter aims to provide a short introduction on data-driven control theory, which will be useful for developing data-driven lgorithms for electric drives purposes. In general, the behavior of three-phase electric machines is described by the voltage balance equations. This model is characterized by motor parameters, such as stator and/or rotor phase resistance of the conductors, self and mutual inductances, depending on the type of machine. In many cases, this model is available, and different control strategies can be implemented. Model-based techniques have been largely investigated in literature, showing promising performances. However, in applications where the control needs to be design for an unknown machine, these techniques may be useless. The idea of data-driven control approach arises to overcome this limit. Instead of design the control based on motor model, this strategy considers only Input/Output (I/O) data collected from the motor (for instance voltages and currents). Several data-driven strategies can be pursued. In this work, the tracking problem is stated as a constrained Quadratic Programming (QP) problem, pursuing an Model Predictive Control (MPC)-type approach. Among possible data-driven strategies, the Data EnablEd Predictive Control (DeePC) and the Subsapce Predictive Control (SPC) are presented.

4.1 MODEL PREDICTIVE CONTROL FRAMEWORK

Consider the discrete-time system given by:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k), \end{aligned} \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}$,$B \in \mathbb{R}^{n\times m}$,$C \in \mathbb{R}^{p\times n}$,$C \in \mathbb{R}^{p\times n}$,$D \in \mathbb{R}^{p\times m}$ $x(t) \in \mathbb{X}^n$, $u(k) \in \mathbb{R}^m$,$y(t) \in \mathbb{R}^p$. Giving a desired reference trajectory $r = [r_0, r_1..]$, input and output constraints, apply control inputs such that the output of the system (4.1) tracks the reference trajectory. In the case when the model of the system is known in state space form, the MPC strategy can be stated. MPC relies on a QP problem type. The following optimization algorithm is considered:

$$\begin{aligned}
\min_{u,x,y} &= \sum_{k=0}^{N-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right) \\
\text{subject to} \quad & x_{k+1} = Ax_k + Bu_k \\
& y_k = Cx_k + Du_k \\
& x_0 = x_t \\
& u_k \in \mathcal{U} \\
& y_k \in \mathcal{Y}, \qquad \forall k \in \{0, ..., N-1\},
\end{aligned} \tag{4.2}$$

where $N \in \mathbb{Z}_{>0}$ is the time horizon, $Q \in \mathbb{R}^{p \times p}$ is the output cost matrix and $R \in \mathbb{R}^{m \times m}$ is the control cost matrix. The classical MPC problem (4.2) is solved considering the receding horizon principle, i.e only the optimal input solution u_0^* is applied to the system. The optimization problem is resolved at each time step.

One of the crucial ingredient of the MPC is that it requires the precise knowledge of the system to be controlled. Furthermore, when the state is not completely measured, one need to consider an estimated state for setting initial conditions. For complex system, this can be a cumbersome and expensive process. For these reasons, it is presented a data-driven algorithm that does not require any particular model of the process.

4.2 DATA-ENABLED PREDICTIVE CONTROL

Classical system theory relies on a representation of the process that includes a mathematical model of the system, as presented in Section 4.1. Here, the point of view of the *Behavioral System Theory*[1] is considered, where the nature of a dynamical system is identified by the subspace of signal space in which trajectories of the system live. The idea behind DeePC relies on this concept.

Some useful definitions are given in the following.

Definition 4.2.1 *A dynamical system is a 3-tuble $(\mathbb{Z}_{>0}, \mathbb{W}, \mathfrak{B})$, where $\mathbb{Z}_{\geqslant 0}$ is the discrete-time axis, $\mathbb{W}$ is a signal space, $\mathfrak{B} \subseteq \mathbb{W}^{\mathbb{Z}_{\geqslant 0}}$ is the behaviour.*

Definition 4.2.2 *Let $(\mathbb{Z}_{\geqslant 0}, \mathbb{W}, \mathfrak{B})$ be a dynamical system.*

(i). It is linear if $\mathbb{W}$ is a vector space and $\mathfrak{B}$ is a linear subspace of $\mathbb{W}^{\mathbb{Z}_{\geqslant 0}}$.

(ii). It is time invariant if $\mathfrak{B} \subseteq \sigma\mathfrak{B}$ where $\sigma : \mathbb{W}^{\mathbb{Z}_{\geqslant 0}} \to \mathbb{W}^{\mathbb{Z}_{\geqslant 0}}$ is the forward time shift defined by $(\sigma w)(t) = w(t+1)$ and $\sigma\mathfrak{B} = \{\sigma w | w \in \mathfrak{B}\}$.

(iii). It is complete if $\mathfrak{B}$ is closed in the topology of point-wise convergence.

1 See [36] for a complete description.

If a dynamical system satisfies (i)-(iii) is equivalent to finite dimensionality of $\mathbb{W}$. The system class $(\mathbb{Z}_{\geqslant 0}, \mathbb{R}^{m+p}, \mathfrak{B})$ is denoted by $\mathfrak{L}^{m+p}$ where $m, p \in \mathbb{Z}_{\geqslant 0}$. With abuse of notation, it is possible to identify a dynamical system in $\mathfrak{L}^{m+p}$ by simply indicate its behavior $\mathfrak{B}$. It is defined now the trajectory of the system truncated to a window of length T as the following set:

$$\mathfrak{B} = \{\omega \in (\mathbb{R}^{m+p})^T \mid \exists v \in \omega \in \mathfrak{B} \text{ s.t. } \omega_t = v_t, \ 1 \leqslant t \leqslant T\}$$

This represents the set of trajectories truncated to a window of length T. Without loss of generality it can be indicated $\mathfrak{B}$ as the product space of two sub-behaviours $\mathfrak{B}^u$ and $\mathfrak{B}^y$, where $\mathfrak{B}^u \subseteq (\mathbb{R}^m)^{\mathbb{Z}_{\geqslant 0}}$ and $\mathfrak{B}^y \subseteq (\mathbb{R}^p)^{\mathbb{Z}_{\geqslant 0}}$ are the space of input and output signals respectively, that correspond to any trajectory ω that can be written as $\omega = [u, y]^T$.

Definition 4.2.3 *A system $\mathfrak{B} \in \mathfrak{L}^{m+p}$ is said to be controllable if for every $T \in \mathbb{Z}_{>0}$, $\omega^1 \in \mathfrak{B}_T$, $\omega^2 \in \mathfrak{B}$ there exists $\omega \in \mathfrak{B}$ and $T' \in \mathbb{Z}_{>0}$ such that $\omega_t = \omega^1_t$ for $1 \leqslant t \leqslant T$ and $\omega_t = \omega^2_{t-T-T'}$ for $t > T + T'$.*

In a brief, a behavioral system is said to be controllable if any two trajectories can be patched together in a finite time.

The following definition denotes one of the most important properties on which DeePC is based.

Definition 4.2.4 *Persistency of Exitation Let $L, T \in \mathbb{Z}_{>0}$ such that $T \geqslant L$. The signal $u = [u_1, ...u_T]^T \in \mathbb{R}_{Tm}$ is said to be persistently exiting of order L if the Hankel matrix:*

$$\mathscr{H}_L(u) := \begin{pmatrix} u_1 & \cdots & u_{T-L+1} \\ \vdots & \ddots & \vdots \\ u_L & \cdots & u_T \end{pmatrix}$$

is full of rank.

One can also represent the behavioral system in the traditional state space representation. The state space representation of minimum order (i.e the one with the smallest state dimension) is called *minimal representation* and it is indicated as n. Another important parameter that is introduced is the *lag* of the system, indicated as $\ell(\mathfrak{B})$, which is defined as the smallest integer $\ell \in \mathbb{Z}_{>0}$ such that the observability matrix $\mathscr{O}_\ell(A, C) := [C, CA, ..., CA^{\ell-1}]^T$ has rank $n(\mathfrak{B})$. These definitions are used in the following Lemma.

Lemma 4.2.1 *Let $\mathfrak{B} \in \mathfrak{L}^{m+p}$ and $\mathfrak{B}(A, B, C, D)$ a minimal input/output state representation. Let T_{ini},$N \in \mathbb{Z}_{>0}$ with $T_{ini} \geqslant \ell(\mathfrak{B})$ and $[u_{ini}, u, y_{ini}, y]^T \in \mathfrak{B}_{T_{ini}+N}$.*
Then there exist a unique $x_{ini} \in \mathbb{R}_{n(\mathfrak{B})}$ such that

$$y = \mathscr{O}_N(A, C)x_{ini} + \mathscr{T}(A, B, C, D)u$$

The Lemma 4.2.1 means that, if a sufficiently long collection of initial I/O data of the system $[u_{ini}, y_{ini}]^T$ is given, then the state to which the system is driven by the sequence of inputs u_{ini} is univocally determined. Now the Fundamental Lemma is introduced. This is the main results of the work presented in [37].

Lemma 4.2.2 (Fundamental Lemma) *Consider a controllable system Let* $\mathfrak{B} \in \mathcal{L}^{m+p}$. *Let* $T, t \in \mathbb{Z}_{>0}$ *and* $\omega = [u, y]^T \in \mathfrak{B}_T$. *Assume* u *to be persistently exiting of order* $t + n(\mathfrak{B})$.
Then colspan$(\mathscr{H}_t(\omega)) = \mathfrak{B}_t$.

Note that in order to satisfy the persistency of excitation, it is required that $T \geqslant (m+1)(t+n(\mathfrak{B}))-1$. The Lemma 4.2.2 says that the system identification procedure of the model can be replaced by constructed any trajectories of a Linear Time Invariant (LTI) system using a finite number of data sampled generated by a sufficient rich input signal. The Lemma 4.2.2 affirms that each trajectory of the system can be described by a linear combination of the Hankel matrix columns. This allow to exploit this matrix to predict future behavior of the LTI system, and design optimal control inputs.

4.3 ALGORITHM FOR DEEPC

This section is dedicated to the explanation of how the data driven control is derived from the results of Lemma 4.2.2. A complete explanation can be found in [38]. Other references of previous or similar works can be found here [39],[40],[41].

4.3.1 *Input/Output Data Collection*

It is considered that the data is generated by an unknown controllable LTI system $\mathfrak{B} \in \mathcal{L}^{m+p}$. Choose $T, T_{ini}, N \in \mathbb{Z}_{>0}$ such that $T \geqslant (m+1)(T_{ini}+N+n(\mathfrak{B}))-1$. Consider:

$$u^d = [u_1^d, ..., u_T^d]^T \in \mathbb{R}^{Tm} \qquad y^d = [y_1^d, ..., y_T^d]^T \in \mathbb{R}^{Tp}$$

that are sequence of I/O collected (indicated with superscript d). Assume u^d to be persistently exited of order $T_{ini}+N+n(\mathfrak{B})$. Consider now the following partition, which divides the I/O samples in *past* and *future* data:

$$\begin{pmatrix} U_P \\ U_F \end{pmatrix} := \mathscr{H}_{T_{ini}+N}(u^d), \qquad \begin{pmatrix} Y_P \\ Y_F \end{pmatrix} := \mathscr{H}_{T_{ini}+N}(y^d)$$

where:

- U_P contains the first T_{ini} block rows of $\mathscr{H}_{T_{ini}+N}(u^d)$ (respectively for Y_P with $\mathscr{H}_{T_{ini}+N}(y^d)$);

- U_F contains the remained N block rows of $\mathscr{H}_{T_{ini}+N}(u^d)$ (respectively for Y_F with $\mathscr{H}_{T_{ini}+N}(y^d)$);

4.3.2 *State estimation and trajectory prediction*

By exploiting Lemma 4.2.2, it is possible to reconstruct any $T_{ini} + N$ length trajectory of a system $\mathfrak{B}_{T_{ini}+N}$. A trajectory $[u_{ini}, y_{ini}, u, y]^T$ belongs to $\mathfrak{B}_{T_{ini}+N}$ if and only if it exists $g \in \mathbb{R}^{T-T_{ini}-N+1}$ such that:

$$\begin{pmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix} \tag{4.3}$$

Intuitively, the trajectory $[u_{ini}, y_{ini}]^T$ identifies the initial state x_{ini} of the system, which is the initial point for the prediction of future trajectory.

4.3.3 *DeePC Implementation*

Given a desired output reference trajectory $r = [r_0, r_1, ..] \in (\mathbb{R}^p)^{\mathbb{Z}}_{>0}$, the aim is to build a controller that finds optimal inputs for the unknown system exploiting the procedure just described. The ingredients for building the optimization problem are the following:

- $N \in \mathbb{Z}_{>0}$ the time horizon length;
- past I/O data $[u_{ini}, y_{ini}]^T \in \mathfrak{B}_{T_{ini}}$, $T_{ini} > \ell(\mathfrak{B})$;
- input constraint set $\mathcal{U} \subseteq \mathbb{R}^m$;
- output constraint set $\mathcal{Y} \subseteq \mathbb{R}^p$;
- output cost matrix $Q \in \mathbb{R}^{p \times p}$;
- input cost matrix $R \in \mathbb{R}^{m \times m}$.

Now, the following optimization problem is set, following the optimization framework of MPC of Section 4.1;

$$
\begin{aligned}
\min_{g,u,y} &= \sum_{k=0}^{N-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right) \\
\text{subject to} \quad & \begin{pmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix} \\
& u_k \in \mathcal{U} \\
& y_k \in \mathcal{Y}, \qquad \forall k \in \{0, ..., N-1\}
\end{aligned} \tag{4.4}
$$

Here, u and y are not independent decision variables, since they are completely described by the matrices U_F and Y_F respectively. The following algorithm represents the DeePC implementation:

Algorithm 1

Input: $[u^d, y^d]^T \in \mathfrak{B}_T$, reference trajectory $r \in \mathbb{R}^{Np}$, past data $[u_{ini}, y_{ini}]^T \in \mathfrak{B}_{T_{ini}}$, constraint sets $\mathcal{U}$ and $\mathcal{Y}$ and performance matrices Q and R.

1: Solve (4.4) for g^*.

2: Compute the optimal input sequence $u^* = U_F g$.

3: Apply the receding horizon.

4: Set $t = t + 1$ and update u_{ini} and y_{ini} to the T_{ini} most recent input/output measurements.

5: Return to 1.

4.4 SOLUTION OF THE PROBLEM

4.4.1 *Constrained solution of DeePC*

The optimization problem in (4.4) requires in general an iterative QP solver. In this work, applications of such algorithms will be presented in Part iii, and output constraints are not considered. This approximation of the problem (4.4) speeds up the algorithm and the computation is reduced. The main disadvantage might be the undesirable dynamics of the outputs, but in most of the presented applications, a tolerance on the overshoot during step transients is allowed, while the input constraints are more stringent. Thus, outputs constraints will not be considered from here on out. Input constraints are assumed to be linear in the input variable u. The problem in (4.4) can be rewritten as:

$$\begin{aligned} &\min_{u,y,g} \ \left(\|y^{ref} - y\|_Q^2 + \|u\|_R^2\right) \\ &\text{s.t.} \begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}, \ lb \leqslant Fu \leqslant ub \end{aligned} \tag{4.5}$$

$lb, ub \in \mathbb{R}^{mN}$ and $F \in \mathbb{R}^{s \times pN}$, where s denoted the number of constraints. The optimization problem is first solved with respect to g. Thus, both input u and output y are eliminated from the minimization problem according to the equality constraints $U_F g = u$ and $Y_F g = y$. Exploiting these relations and explicitly the norm-2 operator, the following problem is written:

$$\begin{aligned} &\min_{g} \ \left((y^{ref} - Y_F g)^T Q (y^{ref} - Y_F g) + g^T U_F^T R U_F g\right) \\ &\text{s.t.} \begin{bmatrix} U_P \\ Y_P \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix}, \ lb \leqslant F U_F g \leqslant ub \end{aligned} \tag{4.6}$$

After a few manipulations, a standard QP problem is obtained, containing both equalities and inequalities constraints:

$$\begin{aligned} &\min_{g} \ \left(\frac{1}{2} g^T 2(Y_F^T Q Y_F^T + U_F^T R U_F) g - 2(y^{ref,T} Q Y_F) g + y^{ref,T} Q y^{ref}\right) \\ &\text{s.t.} \begin{bmatrix} U_P \\ Y_P \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix}, \ lb \leqslant F U_F g \leqslant ub \end{aligned} \tag{4.7}$$

From equation (4.7), it is quite straightforward to identify the Hessian matrix and the liner term:

$$H := 2(Y_F^T Q Y_F + U_F^T R U_F) \qquad c := -2 Y_F^T Q y^{ref} \tag{4.8}$$

Constraints can be condensed in a single expression as follows:

$$\begin{bmatrix} u_{ini} \\ y_{ini} \\ lb \end{bmatrix} \leqslant \begin{bmatrix} U_P \\ Y_P \\ U_F \end{bmatrix} g \leqslant \begin{bmatrix} u_{ini} \\ y_{ini} \\ ub \end{bmatrix} \tag{4.9}$$

The standard qp problem formulation is then obtained:

$$\begin{aligned} &\min_{g} \ \frac{1}{2} g^T H g + c^T g \\ &\text{s.t.} \ \ lb_A \leqslant A g \leqslant ub_A \end{aligned} \tag{4.10}$$

All the dimensions of just defined matrices are listed below:

$$\begin{aligned} g &\in \mathbb{R}^{mL\times 1}, \quad H \in \mathbb{R}^{L\times L} \\ c &\in \mathbb{R}^{L\times 1}, \quad A \in \mathbb{R}^{((m+p)T_{ini}+mN)\times L} \\ U_F &\in \mathbb{R}^{mN\times L}, \quad Y_F \in \mathbb{R}^{pN\times L} \\ U_P &\in \mathbb{R}^{mT_{ini}\times L}, \quad Y_P \in \mathbb{R}^{pT_{ini}\times L} \\ lb_A, ub_A &\in \mathbb{R}^{((m+p)T_{ini}+mN)\times 1} \end{aligned} \tag{4.11}$$

where $L := T - T_{ini} - N + 1$. The problem can be solved by standard QP solvers, e.g. using quadprog Matlab command or third parties solvers such as `qpOASES` or `OSQP`. In this work, `qpOASES` has been extensively uses for his robustness, simplicity and the possibility to interface it with Matlab Simulink and real time prototyping platforms.

4.4.2 *Unconstrained solution of* DeePC

Sometimes it is preferred to avoid the online implementation of a QP solver, especially due to computation hardware limitations. In particular, the problem dimension of the considered a data-driven control method could increase significantly with the size of I/O collected data. For this reason, it is preferable neglecting input constraints and finding a closed-form expression of the control law.

4.4.2.1 *Case I: linear system and noise-free output measurements*

A direct solution for the control problem stated in (4.7) can be obtained neglecting both output and input constraints. The equality constraints on vector g suggest that any N-long future trajectory of a system that starts from the input-output sequence $[u_{ini}, y_{ini}]^T$ can be calculated as:

$$g = M_P^{\dagger} w_{ini} + \Phi z, \quad M_P = \begin{bmatrix} U_P \\ Y_P \end{bmatrix} \tag{4.12}$$

where $M_P^{\dagger}$ is the Moore-Penrose pseudo-inverse of the past trajectories Hankel matrix M_P, whereas Φ is a basis of the kernel of M_P and w_{ini} is a compact expression of the vector $[u_{ini}, y_{ini}]^T$. It is worth noticing that both Φ and $M_P^{\dagger}$ can be pre-computed off-line using the training data set. The just obtained expression for g is replaced in the unconstrained version of (4.10). Thus, a reduced order optimization problem is obtained, whose unknown is just the vector z (considering the hessian term H as $\frac{H}{2}$, with a slight abuse of notation with respect the one defined in (4.8)):

$$\min_z \ (M_p^\dagger w_{ini} + \Phi z)^T H (M_p^\dagger w_{ini} + \Phi z) + d^T (M_p^\dagger w_{ini} + \Phi z) \quad (4.13)$$

Expanding the matrix products and ignoring terms that do not depend on z:

$$\min_z \ \frac{1}{2} z^T \left(2\Phi^T H \Phi\right) z + \left(2 w_{ini}^T (M_p^\dagger)^T H + d^T\right) \Phi z \quad (4.14)$$

The new Hessian matrix and linear term of the problem are respectively:

$$\hat{H} := 2\Phi^T H \Phi$$

$$\hat{d} := \Phi^T \left(2 H M_p^\dagger w_{ini} + d\right)$$

The new problem can be directly solved, provided that the reduced Hessian is positive definite. It is again remarkable the fact the Hessian inversion can be pre-computed off-line, reducing the overall computational effort The unconstrained solution of z can be computed as:

$$z^* = -\hat{H}^{-1} \ \hat{d} \quad (4.15)$$

Then, the unconstrained optimal control input solution is found substituting (4.15) in (4.12), recalling that $u = U_F g$:

$$\begin{aligned} u^* &= U_F g^* = U_F (M_p^\dagger w_{ini} + \Phi z^*) \\ &= U_F (M_p^\dagger w_{ini} - \Phi \hat{H}^{-1} \ \hat{d}) \\ &= U_F (M_p^\dagger w_{ini} - \Phi \hat{H}^{-1} \Phi^T \left(2 H^T M_p^\dagger w_{ini} + d\right)) \\ &= \underbrace{U_F \left(M_p^\dagger - 2\Phi \hat{H}^{-1} \Phi^T H^T M_p^\dagger\right)}_{K_{ini}} w_{ini} + \underbrace{2 U_F \Phi \hat{H}^{-1} \Phi^T Y_F^T Q}_{K_r} y^{ref} \end{aligned} \quad (4.16)$$

$$\Rightarrow u^* = K_{ini} w_{ini} + K_r y^{ref} \quad (4.17)$$

The optimal unconstrained solution is expressed as a combination of the initial condition vector w_{ini} and the reference vector y^{ref}, as reported in (4.17). Only the first optimal input is then applied, in a receding horizon manner. Note that he matrices K_{ini} and K_r can be computed offline. Thus, the computational cost of this algorithm reduces to a number of multiplications that depends linearly on T_{ini} and y^{ref}.

4.4.2.2 *Case II: linear system and output measurements with noise*

In real systems, such as electric motors, the speed and current measurements are affected by noise. In particular, one can consider that the output measurements can be affected by bounded but unknown noise. The problem formulation should be modified in this case. Let consider the following modified cost function:

$$\begin{aligned} &\min_{u,y,g,\sigma_y} \quad \left(\|y^{ref} - y\|_Q^2 + \|u\|_R^2 + \eta\|\sigma_y\|^2 + \beta\|g\|^2\right) \\ &\text{s.t.} \begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{4.18}$$

The term σ_y in (4.18) has been introduced as slack variable on initial conditions. The noise represents uncertainties in measured data, and this uncertainty is penalized in the cost function. In addiction, a term that penalized the norm of g is taken into consideration, as it introduces robustness in the optimization problem [42]. This idea relies on the fact that the optimal solution should be relatively "small" in presence of unknown uncertainties, otherwise it could diverge. The optimization problem is similarly solved as the one in Section 4.4.2.1. The complete passages for solving the problem (4.18) can be found in Appendix C.1. Again, it can be shown that the optimal control input is the results of linear combinations between the initial conditions and reference.

4.5 SUBSPACE PREDICTIVE CURRENT CONTROL

An alternative way for building a data-driven controller was proposed by Van Overschee and De Moor in 1999 [43], namely the SPC. The idea behind this approach is slightly different with respect the DeePC, even if the initial data and the algorithm output are similar. In particular, a collection of measured I/O trajectories are used also by SPC as initial data. The output of the SPC algorithm is a data-driven controller and not a state-space representation of the system, similarly to the DeePC.

As a preliminary step, measured data are reorganized into the already defined Hankel matrices U_P, U_F, Y_P and Y_F. Then, the idea consists into building a linear predictor of the form $Y_F = L_w[U_P \;\; Y_P]^T + L_u U_F$ between, on one hand, future outputs and, on the other hand, initial trajectory and future inputs

The predictor matrix $L = [L_w, \; L_u]$ is computed as the least square solution of:

$$\min_{L} \left|\left| Y_F - L \begin{bmatrix} U_P \\ Y_P \\ U_F \end{bmatrix} \right|\right|^2 \tag{4.19}$$

L can be found by means of the Moore-Penrose pseudo-inverse, i.e. $L = Y_F[U_P, Y_P, U_F]^\dagger$. Alternatively, numerically stable algorithms can be implemented, such as QR-decomposition. The main difference between SPC and DeePC is the technique adopted to get rid of noise in the measurements. On one hand, a slack variable is included in the DeePC algorithm. On the other hand, a Singular Value Decomposition (SVD) of the initial trajectory predictor matrix L_w is performed, i.e. $L_w = U_1 S_1 V_1^T$. Inspecting S_1, dominant singular values are found and L_w is reconstructed using a truncated version of S_1. The number of dominant values give also an information of the system dimension, that can be compared to the expected one.

The SPC uses the same optimization framework of Section 4.1:

$$\min_{u} \left((r-y)^T Q (r-y) + u^T R u \right) \text{ s.t. } y = L_w [u_{ini}, y_{ini}]^T + L_u u \tag{4.20}$$

Main differences with the DeePC algorithm are the equality constraints and the problem unknown. Equality constraints depend, in fact, on how training data are exploited. On the other hand, the problem is solved for the future output sequence u and not for the vector g. A closed form solution of (4.20) can be found, since the unconstrained problem is here considered. Moreover, only the first optimal input of the sequence u is applied, in a receding horizon manner. In conclusion, the control law can be resumed in a compact form as:

$$\begin{aligned} u(k) &= L_r^c r - L_w^c [u_{ini}, y_{ini}]^T \\ L_r^c &= M_{aux}(1:m,:) L_u^T Q, \\ M_{aux} &= (R + L_u^T Q L_u)^{-1}, \\ L_w^c &= L_r^c L_w \end{aligned} \tag{4.21}$$

Algorithm 2 Subspace Predictive Control SPC.

Input: Input-output measurements $[u_d, y_d]^T$ for the off-line stage; reference trajectory r, performance matrices Q and R and initial trajectory u_{ini} and y_{ini} for the on-line stage.

1: (Training) Build Hankel matrices U_P, U_F, Y_P and Y_F.

2: (Off-line) Find the solution of the least squares problem $\min_{L_u, L_w} \|Y_f - (L_u \;\; L_w)(U_P \;\; Y_P \;\; U_F)^T\|^2$ for L_w and L_u.

3: (Off-line) Compute the SVD decomposition $L_w = U_1 S_1 V_1^T$ and approximate L_w by inspecting the dominant singular values.

4: (On-line) Apply the first solution of the control sequence (unconstrained solution): $u = (R + L_u^T Q L_u)^{-1} L_u^T Q (r - L_w [u_{ini}, y_{ini}]^T)$.

5 MODEL ORDER REDUCTION

In this chapter, a brief introduction to Model Order Reduction (MOR) techniques is presented. In particular, the parametric Proper Orthogonal Decomposition (POD) method is discussed, which will be exploited in Chapter 9 for building a reduced data-driven model of the electric drive.

5.1 INTRODUCTION

The real-time control of large-scale dynamical systems, due to the inherent large-scale nature of the models, leads to unmanageable demands on computational resources. MOR aims to reduce the computational burden by generating reduced models that are faster and cheaper to manage, yet accurately represent the original large-scale system behavior. This means that in an early stage of the process, the most basic properties of the original model must already be present in the smaller approximation. At a certain moment the process of reduction is stopped. At that point all necessary properties of the original model must be captured with sufficient precision. All of this has to be done automatically. In more recent years much research has been done in the area of the MOR. Consequently, a large variety of methods is available. Some are tailored to specific applications, others are more general. Some basic notions that are necessary to understand how model reduction techniques work are given in the following sections. In particular, the focus is on the broad class of problems for which the equations representing the system dynamics depend on a set of parameters and the goal is to characterize system response for different values of the parameters. These parameters may enter the models in many ways, representing, for example, material properties, system geometry, system configuration, initial conditions, and boundary conditions. This parametric dependence presents a unique set of challenges for model reduction, since one cannot create a new reduced model for every change in the parameter values. Hence, the desired approach is to generate a parametric reduced model, that approximates the original full-order dynamical system with high fidelity over a range of parameters. This is the goal of parametric model reduction. The following notes follow the approach presented in [44], where a complete and exhaustive survey on parametric model reduction is presented. Furthermore, some very well-written textbooks have been consulted [45, 46]

5.2 PROJECTION-BASED MODEL REDUCTION

In this section, the system structure that is for interest for the MOR algorithms are presented. The focus in on the projection-based strategies, which, in principle, rely on the concept of truncating the solution of the original system in an appropriate basis.

5.2.1 *Parameterized Dynamical System*

Consider the linear dynamical system, parameterized with d parameters $p = [p_1, ..., p_d]^T \in \Omega \subset \mathbb{R}^d$, with Ω bounded domain:

$$\begin{aligned} D(p)\dot{x}(t,p) &= K(p)x(t,p) + B(p)u(t) \\ y(t,p) &= C(p)x(t,p) \\ x(0,p) &= 0 \end{aligned} \tag{5.1}$$

where $t \in [0,\infty)$, $x(t,p) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ and $y(t,p) \in \mathbb{R}^q$ represent the inputs (excitations) and outputs (measurements) of the model. The state space matrices have the dimensions $D(p)$, $K(p) \in \mathbb{R}^{n\times n}$, $B(p) \in \mathbb{R}^{n\times m}$ and $C(p) \in \mathbb{R}^{q\times n}$. Thus dimension of parametric model (5.1) is n. The following are considered: for every $p \in \Omega$, $D(p)$ is non-singular and the original model (5.1) is asymptotically stable for every $p \in \Omega$. This means that the eigenvalues of the matrix $\lambda D(p) - K(p)$ have negative real parts.

The aim is to represent the original large-scale model (5.1), where n is usually very large (for example hundreds of thousands), with a reduced model of the form:

$$\begin{aligned} D_r(p)\dot{x}_r(t,p) &= K_r(p)x_r(t,p) + B_r(p)u(t) \\ y_r(t,p) &= C_r(p)x_r(t,p) \\ x_r(0,p) &= 0 \end{aligned} \tag{5.2}$$

Such that the reduced output $y_r(t,p) \in \mathbb{R}^q$ is a good approximation of $y(t,p)$ with respect to a certain error measure. The reduced state-vector has length $r \ll n$ and the reduced state-space matrices have dimensions $D_r(p), K_r(p) \in \mathbb{R}^{r\times r}$, $B_r(p) \in \mathbb{R}^{r\times m}$ and $C_r(p) \in \mathbb{R}^{q\times r}$. Thus, the dimension is reduced from n to $r \ll n$. A graphical description of this process is given in Figure 5.1.

5.2.2 *Projection-Based Model Reduction*

Parametric model reduction can be approached from a variety of viewpoints. This work focuses on projection-based approaches. Firstly, consider the general projection-based reduction approach for a system with no parametric dependence; i.e.,

$$\begin{aligned} D\dot{x}(t) &= Kx(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{5.3}$$

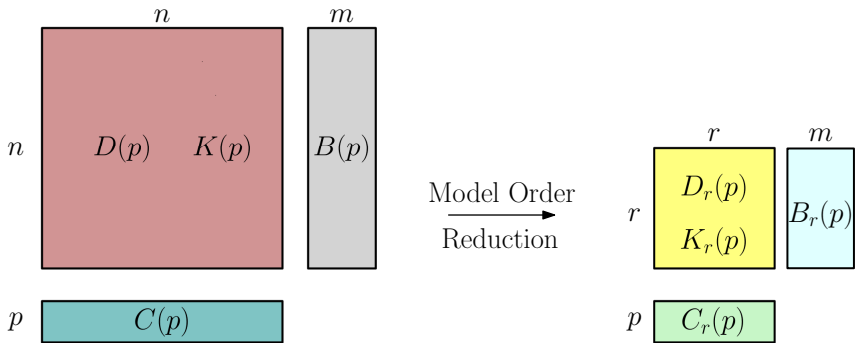


Figure 5.1: Model Order Reduction Principle.

The goal is to approximates the unknown state quantities in a basis of reduced dimension and projects the system (5.3) onto a suitably defined low-dimensional subspace. In particular, choose r-dimensional test and trial subspaces, denoted by $\mathcal{V}$ and $\mathcal{W}$, respectively, where $r \ll n$. Define the associated basis matrices $V \in \mathbb{R}^{n\times r}$ and $W \in \mathbb{R}^{n\times r}$, where $\mathcal{V} = \mathrm{Ran}(V)$ and $\mathcal{W} = \mathrm{Ran}(W)$, with Ran denoting the range. Using the approximation that the full state $x(t)$ evolves in the r-dimensional subsbace $\mathcal{V}$, it can be written as $x(t) \approx Vx_r(t)$, where $x_r(t) \in \mathbb{R}^r$. Using this approximation in (5.3):

$$DV\dot{x}_r(t) = KVx_r(t) + Bu(t) + res \tag{5.4}$$

then, defining the residual $res = DV\dot{x}_r(t) - KVx_r(t) - Bu(t)$, the most used method to find the matrix W is to impose the so-called Petrov-Galerkin condition, i.e.:

$$W^T(DV\dot{x}_r(t) - KVx_r(t) - Bu(t)) = 0 \tag{5.5}$$

In this way, the residual is constrained to be orthogonal to the subspace defined by W. The reduced system becomes:

$$\begin{aligned} D_r\dot{x}_r(t) = K_rx_r(t) + B_ru(t)) = 0 \\ y_r(t) = C_rx_r(t) \end{aligned} \tag{5.6}$$

where the reduced matrices are given by

$$\begin{aligned} D_r &= W^TDV \\ K_r &= W^TKV \\ B_r &= W^TB \\ C_r &= CV \end{aligned} \tag{5.7}$$

5.2.3 *Methods for Model Reduction*

There are several approaches that can be used to construct a Reduced Order Model (ROM). A depth description of such methods can be found in [46]. A list of possible algorithms is given in the following:

1. Rational Interpolation methods, (moment-matching and Padé-type approximation methods), which in principle find an approximation of the system transfer function;
2. Balanced Truncation methods, which is one of the most common techniques for approximating linear dynamical systems without parametric dependency. In the parametric setting, this strategy can be employed to construct local reduced models at given parameter values;
3. POD, which, in contrast to the rational interpolation methods that formulate the basis computation task in the frequency domain, POD formulations typically use the time domain.

This work presents the POD methods, which will be used in Chapter 9.

5.3 PROJECTION FRAMEWORK FOR PARAMETERIZED SYSTEMS

In the case of a system with no parametric dependence, the reduced quantities in (5.7) are precomputed constant matrices, and the reduced model can be evaluated with no further reference to the full model. However, in the case of a parameterized system, the reduced model will also be parameter dependent. Several challenges arise in achieving efficient construction and evaluation of a reduced model as the parameters vary.
The first challenge is how to introduce parametric dependence into the basis matrices V and W. One option is to construct "global" basis matrices over the parameter space, that is, a single matrix V and a single matrix W, each of which captures parametric dependence by embedding information regarding the entire parameter space (e.g., information collected by sampling multiple parameter values).
A second option is to construct "local" basis matrices. That is, consider K parameter sample points $\hat{p}_1, ..., \hat{p}_K$. For the realization of the dynamical system corresponding to $\hat{p}_i$ (i.e., $D(\hat{p}_i)\dot{x}(t) = K(\hat{p}_i)x(t) + B(\hat{p}_i)u(t), y(t) = C(\hat{p}_i)x(t)$), the state-space matrices are constant, and one computes appropriate local basis matrices V_i and W_i (i.e., V_i and W_i denote the basis matrices corresponding to the parameter $\hat{p}_i$).

A second challenge is achieving efficient evaluations of the parametric reduced model in the case that a global basis is used. For example, for a given V and W consider evaluating $K_r(p) = W^T K(p) V$. For general parametric dependence, K_r cannot be pre-computed; instead, evaluating the reduced model for a new parameter value p requires computing $K(p)$ and subsequent pre and post-multiplication by W^T and V, respectively. These operations all depend on the (large) dimension n of the original problem. Fortunately, in some cases the structure of the problem admits an efficient strategy. For example,

consider the case of affine parameter dependence[1] with $M+1$ terms:

$$K(p) = K_0 + \sum_{i=0}^{M} f_i(p) K_i \tag{5.8}$$

where the scalar function f_i determine the parametric dependency, which can be nonlinear, and $K_i \in \mathbb{R}^{n\times n}$ for $i = 0, ..., M$ are parameter independent. Then the reduced matrix is given by

$$K_r(p) = W^T K(p) V = W^T K_0 V + \sum_{i=0}^{M} f_i(p) W^T K_i V \tag{5.9}$$

For affine parametric dependence in the other matrices $D(p)$, $B(p)$, and $C(p)$, analogous expressions can be derived for the reduced-order counterparts. The two most important advantages of an affine parameterization are listed below:

1. Once the basis matrices V and W are chosen, the component reduced-order matrices (e.g., $W^T K_i V, i = 0, ..., M$) can be a priori computed in the offline phase. Hence, the reduced model for a given p can be constructed without referring back to the original system, thus having a small online cost;

2. the ROM has the same parametric structure as the original one, which may be appealing to designers who work with these models. Note that the affine representation in (5.8) always holds for any $K(p)$ by letting $M = n^2$ and choosing K_i as matrices with only one nonzero entry. However, for the affine representation to have the computational advantages discussed above, one needs $M \ll n^2$ and explicit expressions for $f_i(p), i = 1, ..., M$.

5.4 PROPER ORTHOGONAL DECOMPOSITION

Due to its broad applicability to linear and nonlinear systems, the POD has become widely used in many different application domains as a method for computing the reduced basis [44].

5.4.0.1 *Time Domain POD*

POD was introduced for the analysis of turbulent flows by Lumley in [47]. POD basis vectors are computed empirically using sampled data collected over a range of relevant system dynamics, typically using the method of snapshots, introduced by Sirovich [48]. Consider a set of snapshots, $x_1, x_2, ..., x_{n_s}$, which are state solutions computed at different instants in time and/or different parameter values, where $x_j \in \mathbb{R}^n$ denotes the jth snapshot and one collects a total of $n_s < n$

1 This is the situation that will be presented in Chapter 9.

snapshots. More specifically, write $x_j = x(t_j; p_j)$, where t_j and p_j are, respectively, the time and parameter values for the jth snapshot. Define the snapshot matrix $X \in \mathbb{R}^{n \times n_s}$ whose jth column is the snapshot x_j. The Singular Value Decomposition (SVD) of X is:

$$X = U\Sigma Y^T \tag{5.10}$$

where the columns of the matrices $U \in \mathbb{R}^{n \times n_s}$ and $Y \in \mathbb{R}^{n_s \times n_s}$ are the left and right singular vectors of X, respectively. $\Sigma \in \mathbb{R}^{n_s \times n_s} = diag(\sigma_1, \sigma_2, ..., \sigma_{n_s})$, where $\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_{n_s} \geqslant 0$, are the singular values of X. The POD basis V is selected as the r left singular vectors of X that correspond to the r largest singular values, generating an orthonormal basis. The POD basis is optimal in the sense that, for an orthonormal basis of size r, it minimizes the least squares error of snapshot reconstruction,

$$\min_{V \in \mathbb{R}^{n \times r}} \|X - VV^T X\|_F^2 = \min_{V \in \mathbb{R}^{n \times r}} \sum_{i=1}^{n_s} \|x_i - VV^T x_i\|_2^2 = \sum_{i=r+1}^{n_s} \sigma_i^2. \tag{5.11}$$

The square of the error in (5.11) is given by the sum of the squares of the singular values corresponding to those left singular vectors not included in the POD basis. Thus, the singular values provide quantitative guidance for choosing the size of the POD basis. A typical approach is to choose r so that

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} > \kappa, \tag{5.12}$$

where κ is a desired tolerance. The numerator of (5.12) is often referred to as the energy captured by the POD modes. Since the POD basis is constructed from sampled solutions, the POD method makes no assumptions about the form of the full model; POD applies to both linear and nonlinear systems, as well as to parameter varying systems. It is important to note that the optimality of the POD basis applies only to error in reconstruction of the snapshots, not to the error in solution of the reduced model. Thus, the choice of snapshots is critical to the quality of the reduced model.

5.4.0.2 *Adaptive Parameter Sampling via Greedy Search*

The choice of parameter sampling points is a critical issue that arises in all methods of calculating the basis. In POD methods, one must select the parameter samples to which the snapshots are computed. For problems with a small number of parameters, a structured or random sampling method (e.g., grid-based sampling) is the simplest approach and, with a sufficiently large number of samples, will generate a rich dataset covering the parameter space. Greedy sampling methods for model reduction approach the task of choosing parameter sample points one by one in an adaptive manner. The general

steps in the greedy sampling approach are as follows. First, given a current reduced model,

$$\begin{aligned} D_r(p)\dot{x}_r(t,p) &= K_r(p)x_r(t,p) + B_r(p)u(t) \\ y_r(t,p) &= C_r(p)x_r(t,p) \end{aligned} \tag{5.13}$$

find the parameter value for which the error between the reduced model and the full model is largest:

$$\hat{p} = \arg\max_p \|y(\cdot\,;p) - y_r(\cdot\,;p)\|_2 \tag{5.14}$$

Second, solve the full model at $\hat{p}$ to generate new information with which to update the reduced model. Then, with the updated reduced model, repeat these two steps until the error is acceptable. This method has since been applied in conjunction with POD methods [49–51] and rational interpolation methods [52]. The key advantage of the greedy approach is that the search over the parameter space embodies the structure of the problem, so that the underlying system dynamics guide the selection of appropriate parameter samples.

5.5 DISCUSSION

The main steps which are involved in projection-based model reduction of parameterized systems can be summarized as follows:

1. Choose the parameter values at which to sample. This may be done at once as a first step or in an iterative fashion.
2. Evaluate the full model for each sampled parameter, which might involve evaluating system matrices and solving linear systems, or evolving the full model dynamics, depending on the reduction method chosen.
3. Build the reduced basis, using a local or global strategy.
4. Project the equations to produce the reduced-order state-space matrices.
5. Use the resulting reduced model for simulation, optimization, control, or uncertainty quantification.

POD is the most generally applicable among the possible methods, since it relies only on snapshots of the underlying system solutions. As a result, the POD basis can be computed easily, even when the simulation is a black-box code. Finally, the POD can also be applied to general nonlinear problems, since computation of the POD basis does not rely on a specific problem structure.

Part II

MODEL PREDICTIVE CONTROL FOR ELECTRIC DRIVES

6

INTEGRAL MODEL PREDICTIVE CURRENT CONTROL FOR SYNCHRONOUS MOTOR DRIVES

Model Predictive Control represents an affirmed optimal control strategy, able to handle multi-variable systems and their input-output constraints. However, Model Predictive Control (MPC) does not provide an integral control action for reference tracking control problems. Several methods have been proposed to overcome this limitation. Standard MPC methods include a disturbance observer to handle unmodeled uncertainties, such as external disturbances and parameter mismatches. Among these formulations, the authors focus on the velocity form MPC, which considers the incremental formulation of the motor state space model. This formulation gets rid of the bias errors in reference tracking problems. In this chapter, the MPC paradigm is applied to the current control of synchronous motor drives. The intent is to compare the velocity form and the MPC with disturbance observer. A theoretical analysis of the MPC coupled with disturbance observers and the equivalence between these formulations and the velocity form is presented. Input constraints are included in the MPC optimization process, thus requiring an online quadratic programming solver. Experimental tests consider a 1 kW anisotropic synchronous motor. Numerical aspects regarding the optimization problem are investigated for both methods.

6.1 INTRODUCTION

Permanent Magnet Synchronous Motor (PMSM) drives are nowadays an affirmed technology in applications where high performance motion control is required. In fact, PMSMs present significant merits due to minimal rotor Joule losses, high power density and large torque-to-inertia ratio. Despite these advantages, designing high precision controllers for PMSM drives is a rather challenging problem because of (i) the complex and non-linear dynamics to be controlled and (ii) the presence of disturbances (also time varying), due to load changes, uncertainties on the mechanical/electrical parameters of the system and sensitivity to environmental operating conditions.

The research interest on model-based algorithms for electric drive applications is constantly growing. Improvements in the computational power of modern Digital Signal Processor (DSP)s and the development of dedicated hardware solutions are making these algorithms suitable for industrial implementations. MPC represents a promising strategy for electric drives applications [8],[9]. MPC theory relies on

the presence of two main features: a model of the process to be controlled and a proper cost function to be minimized. Considering the PMSM current-reference tracking problem, MPC exploits the motor model to compute optimal voltage inputs over a future finite receding horizon time.

In general, the higher is the accuracy of the model, the better the MPC performs. Concerning PMSM current control, the machine model presents a non-linear behavior, e.g., the non-linearity induced by magnetic saturation effects. Thus, an accurate description of current dynamics would imply the modeling of complex non-linear relations among electric quantities. As a drawback, such a precise representation of the model could drastically increase the controller complexity, reducing its reliability. As a consequence, a relatively simple linear model is widely adopted for current tracking MPCs. When a non-linear representation of the system is needed, efficient numerical methods are required to reduce the computational complexity, as proposed in [53] and [23].

In literature, two main MPC strategies have been proposed for power electronics and electric drives applications. The first refers to the Finite Control Set (FCS)-MPC, where the discrete nature of the inverter is exploited for solving the optimization problem [54],[55]. In details, the solution of the current tracking problem is found among the eight possible switch configurations of a two level voltage-source inverter. This solution leads the controller to be very attractive for electric drives applications, where high sampling frequencies are usually adopted. The second refers to the Continuous Control Set (CCS)-MPC, where MPC is stated as a Quadratic Programming (QP) problem type [56–58]. The problem can be formulated in explicit or implicit form. Implementations of explicit MPC are proposed and analyzed in [59]. Similarities and differences between implicit and explicit formulations have been studied in [60]. Newly, in [35, 61], the CCS-MPC has been successfully implemented as PMSM current control, fulfilling all the requirements of embedded control. Moreover, recent publications include also the nonlinear MPC framework [62]. The authors remand to [27] for a complete and exhaustive survey on MPC applications in power electronics field.

Several advantages are gained by adopting an MPC scheme. First, MPC can easily handle multiple-input multiple-output systems [7], whereas a decoupling of the dq axes and compensation strategies are required by traditional Proportional-Integral-Derivative (PID) controllers to keep a simple implementation [63]. Furthermore, constraints related to the system physics can be lightly included in the optimization problem (e.g. see [64]).

One of the main drawbacks that affects MPC performances is represented by model uncertainties. In particular, a defective knowledge of the motor parameters and the presence of unmodeled external distur-

bances cause an error in current prediction. This causes the control action to guide the system along a different trajectory from the reference one, producing a bias error between the reference and actual controlled variables. Many solutions have been proposed to overcome this limit. In [65], an overview on disturbance and uncertainty estimation and attenuation techniques for PMSMs is presented. In [66], a robust MPC controller is designed adopting an extended state observer, reducing the sensitivity against the stator inductance mismatch. Compared to [66], the disturbance observer or the velocity form are designed to provide integral action to the control, rather than estimate inductive parameters. Moreover, a moving horizon estimator is implemented as disturbance observer for a PMSM in [67]. Recently, in [68], the MPC algorithm has been coupled with resonant controllers to reduce the effect of unmodeled periodic disturbances. Even for the FCS-MPC a recent research proposes a Disturbance Observer (DOB) to improve the robustness against uncertain parameters for three-phase inverters [69]. Furthermore, in [70–72], the multistep-MPC has proven to be a viable method for improving the transient and steady-state behavior of the predictive current controller for medium voltage drives.

Only very few publications have analyzed the design of CCS-MPCs for electric drives. Inspired by [73], the offset-free MPC that includes a DOB is considered. In particular, a steady-state Kalman filter DOB is reviewed. Moreover, the velocity form of MPC is presented, which hereinafter will be referred as Integral-Model Predictive Control (I-MPC). In principle, it is obtained by considering the incremental formulation of the motor state-space model. In literature, velocity form of CCS-MPCs have been recently presented for the current control of an induction motor [74] and a PMSM [75]. However, the strategy presented in [75] belongs to the FCS-MPC version. The purposes of this work are manifold and listed in the following. First, guidelines for the design of CCS offset-free MPC with DOBs are given. Second, experimental validation of the benefits introduced by the integral action in presence of model uncertainties are provided. Third, the comparison between the two control solutions is conducted with experiments, in terms of performance and computational aspects, giving indications of when one solution may be preferred over the other. Fourth, the robustness against motor parameter variations is analyzed for the velocity form and MPC with steady-state Kalman filter DOB. Finally, the focus is on numerical problems and constraints inherent in real-time control for electrical drive applications.

6.2 MODEL PREDICTIVE CURRENT CONTROL OF A PMSM DRIVE

The idea behind MPC approach consists in predicting the future behavior of a system exploiting an internal model. Thanks to the predic-

tion step, MPC computes the optimal control action over time, minimizing an appropriate cost function. The internal model adopted by MPC is often the discrete-time state-space representation of the process:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + h(k) \\ y(k) = Cx(k) \end{cases} \tag{6.1}$$

where $x \subseteq \mathbb{X}^n, u \subseteq \mathbb{U}^m, y \subseteq \mathbb{R}^p$ are the state, input, and output vectors, respectively, whereas $\mathbb{X}^n$, $\mathbb{U}^m$ and $\mathbb{R}^p$ are their admissible sets. A disturbance term $h(k)$ is included too.

The state-space representation of the PMSM electrical variables dynamics, expressed in the dq reference frame synchronous with the rotor flux [57] is considered. The matrices and vectors in (6.1) assume the following form:

$$A = \begin{bmatrix} 1 - \frac{T_s R}{L_d} & \frac{T_s \omega_{me}(k) L_q}{L_d} \\ -\frac{T_s \omega_{me}(k) L_d}{L_q} & 1 - \frac{T_s R}{L_q} \end{bmatrix}; B = \begin{bmatrix} \frac{T_s}{L_d} & 0 \\ 0 & \frac{T_s}{L_q} \end{bmatrix}$$

$$x(k) = \begin{bmatrix} i_d \\ i_q \end{bmatrix}; u(k) = \begin{bmatrix} u_d \\ u_q \end{bmatrix}; h(k) = \begin{bmatrix} 0 \\ -\frac{T_s \omega_{me}(k) \Lambda_{pm}}{L_q} \end{bmatrix}$$

where T_s is the sample time, i_d and i_q are the dq axis currents, u_d and u_q are the dq applied voltages. C in (6.1) is the identity matrix. Moreover, L_d and L_q are the dq PMSM inductances, respectively, and R is the stator phase resistance. The back-Electro Motive Force (back-EMF) induced by the presence of permanent magnets is modeled as a disturbance term $h(k)$. This term depends on the electric speed $\omega_{me}(k)$ and on the flux linkage Λ_{pm}. Instead of using $u(k)$ as control input, it is convenient to adopt the input increment $\Delta u(k) = u(k) - u(k-1)$. The optimal future N input increments $\Delta u(k+z)$ to be applied to the PMSM are computed on the basis of the following cost function J minimization process:

$$\begin{aligned} \min_{\Delta u(\cdot)} J = \min_{\Delta u(\cdot)} & \sum_{z=1}^{N-1} \|y^*(k+z) - y(k+z)\|_Q^2 + \\ & + \sum_{z=0}^{N-1} \|\Delta u(k+z)\|_R^2 + \|y^*(k+N) - y(k+N)\|_S^2 \\ \text{s.t. } & x(k+z+1) = Ax(k+z) + Bu(k+z) + h(k+z) \\ & \Delta u(k+z) = u(k+z) - u(k+z-1) \end{aligned} \tag{6.2}$$

where N is the prediction horizon length, $y = [i_d, i_q]^T$ is the predicted current vector, $y^* = \left[i_d^*, i_q^*\right]^T$ is the reference current vector, and Q, R, S are weighting matrices.

6.3 INTEGRAL ACTION

The MPC optimality solution is measured by the cost function value. However, if a model mismatch occurs in matrices A, B or in the vector $h(k)$, the calculated voltage reference $\Delta u(k)$ does not guarantee an unbiased current reference tracking, even if the value of the cost function is zero. A conventional MPC scheme without a disturbance observer, in fact, introduces only a proportional action for a tracking problem.

In this section an integral action is included inside the MPC problem formulation (6.2). A different form of the state-space representation, known as velocity form, is required to guarantee this feature. The relevant difference between standard MPC and I-MPC is that the latter considers an incremental state space model formulation (see (6.3)). The new state is obtained by subtracting the equation (6.1) evaluated in the $k+1$ sample from the same equation evaluated at sample k, defining $\Delta x(k+1) = x(k+1) - x(k)$:

$$\Delta x(k+1) = A\Delta x(k) + B\Delta u(k) \tag{6.3}$$

The back-EMF term h is elided, since both motor speed and permanent magnet flux does not change significantly within a sample period. The I-MPC problem statement can be rewritten in a standard MPC form using the state augmentation technique. First, the output equation of (6.1) has to be rearranged in a incremental form:

$$\begin{aligned} &y(k+1) - y(k) = C(x(k+1) - x(k)) = C\Delta x(k+1) \\ &\rightarrow y(k+1) = CA\Delta x(k) + y(k) + CB\Delta u(k) \end{aligned} \tag{6.4}$$

The augmented form of state equation (6.1) is reached by stacking Δx and y in a new state variable denoted x_m:

$$\begin{aligned} \begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix} &= \begin{bmatrix} A & 0_{n\times p} \\ CA & \mathbb{1}_{p\times p} \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B \\ CB \end{bmatrix} \Delta u(k) \\ y_m(k) &= \begin{bmatrix} 0_{p\times n} & \mathbb{1}_{p\times p} \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} \end{aligned} \tag{6.5}$$

Then, the system is rewritten in a conventional state-space form with the new augmented state variable $x_m \in \mathbb{R}^{n+p}$ and new matrices denoted A_m and B_m, derived from (6.5):

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m \Delta u(k) \\ y_m(k) &= C_m x_m(k) \end{aligned} \tag{6.6}$$

The analysis of the new system permits to understand the difference between a conventional MPC and an I-MPC. The eigenvalues of the augmented system matrix A_m are in fact:

$$\begin{aligned} \det(\lambda \mathbb{1}_{n+p} - A_m) &= \det \begin{bmatrix} \lambda \mathbb{1}_n - A & 0_{n\times p} \\ -CA & \lambda \mathbb{1}_p - \mathbb{1}_p \end{bmatrix} \\ &= (\lambda - 1)^p \det(\lambda \mathbb{1}_n - A) \end{aligned} \tag{6.7}$$

where p integrators are added to the eigenvalues of original matrix A thanks to the incremental formulation. This fact proves why I-MPC, differently to MPC, assures the rejection of constant disturbances in the output reference tracking.

6.3.1 *Constrained solution of I-MPC*

MPC framework permits to easily handle input or output inequality constraints. However, a constrained optimization problem requires in general a high computational effort. In the last years, some real-time oriented QP solvers have been proposed [24] for solving this class of problems. In particular, the solver `qpOASES` has both a MATLAB Simulink and a dSPACE interface, allowing real-time testing on dSPACE hardware, e.g. MicroLabBox. All QP solvers require a standard constraints matrix in the form $M\Delta U \leqslant b$ to compute the constrained solution. The simplest constraints that can be included in (6.2) are the input voltage ones. In the case of a three-phase two-level inverter, the feasible voltage set is represented by an hexagonal convex region in a stationary reference frame, reported in Figure 6.1. Considering a one-step-long prediction horizon and an incremental formulation for the input voltages, all six constraints inequalities are described by the following matrix expression:

$$\begin{aligned} m^T T^{-1} \begin{pmatrix} \Delta u_d(k) \\ \Delta u_q(k) \end{pmatrix} &\leqslant -m^T T^{-1} \begin{pmatrix} u_d(k-1) \\ u_q(k-1) \end{pmatrix} + u_{lim}^T \\ m &= \begin{bmatrix} \sqrt{3} & 0 & -\sqrt{3} & -\sqrt{3} & 0 & \sqrt{3} \\ 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix} \\ u_{lim} &= 2/\sqrt{3}\, U_{DC} \begin{pmatrix} 1 & 1/2 & 1 & 1 & 1/2 & 1 \end{pmatrix} \end{aligned} \tag{6.8}$$

where T^{-1} is the well-known inverse Park transformation. It is reminded that the model used for the predictions is the dq one, as

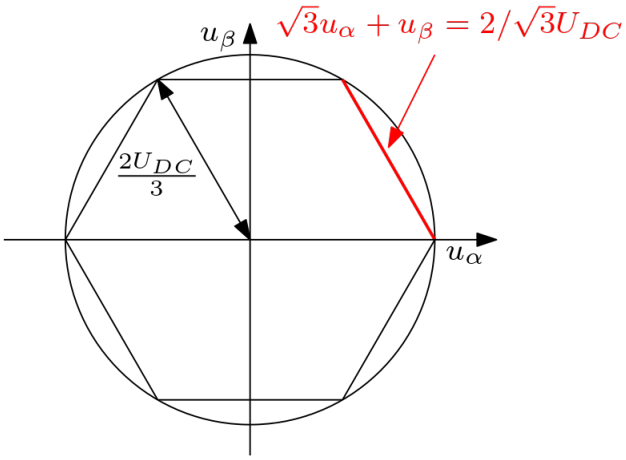


Figure 6.1: Voltage constraints in a stationary reference frame. One of the six equations that define the feasible set is reported as an example.

presented in Section 6.2. As an example, the first column defines the equation of constraint in Figure 6.1. The (6.8) can be rewritten in the form $M\Delta U \leqslant b$ straightforwardly, where the matrix M is defined as $m^T T^{-1}$.

The same approach used for a one-step-long prediction horizon can be easily extended in the case of longer horizons. A bigger M matrix has to be defined, in particular its size is $(2N) \times (6N)$. The motor model, in fact, has two control inputs, i.e., the dq voltage components, and six constraints equations, i.e., the ones defined in (6.8).

6.4 VELOCITY FORM AND DISTURBANCE MODEL EQUIVALENCE

The offset-free tracking can be achieved by different MPC formulations. In particular, the velocity form of MPC is considered as an alternative strategy with respect to disturbance model and observers. However, recent results reported in [73] show that the velocity form represents an equivalent formulation of a particular case of disturbance and model observer. To have a clear and concise explanation, the steady-state Kalman filter estimator is considered. This represents the benchmark in linear offset-free MPC. In the following, the main result and new considerations about the implementation of offset-free MPC in electric drives applications are presented. It is worth noting that the following theoretical results hold for Linear Time Invariant (LTI) systems. The matrix A in Section 6.2 contains speed dependence terms. As suggested in [35], the system is linearized for a constant value of speed (i.e., the base speed Ω_N). Then, if a different speed is set, the DOB will try to estimate it as a discrepancy from the "nominal" model.

6.4.1 *Disturbance and Model Observer*

To get an offset-free tracking an augmented model including the disturbance as state variable is first defined; then, a state observer is designed to estimate the disturbance. Let us consider the system (6.1). The general augmented model is introduced:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_d d(k) \\ d(k+1) &= d(k) \\ y(k) &= Cx(k) + C_d d(k) \end{aligned} \tag{6.9}$$

where $d \in \mathbb{R}^{n_d}$ is the disturbance state. Once defined, the augmented state $\xi = [x; d]$, (6.9) can be rewritten as:

$$\begin{aligned} \xi(k+1) &= A_a \xi(k) + B_a u(k) \\ y(k) &= C_a \xi(k) \end{aligned} \tag{6.10}$$

where $A_a := \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix}$, $B_a := \begin{bmatrix} B \\ 0 \end{bmatrix}$ and $C_a := \begin{bmatrix} C & C_d \end{bmatrix}$. It is assumed that the system (6.10) is observable. The disturbance state d is not controllable by the input in (6.10). However, using the above assumption, the disturbance estimation is used to remove its influence from the controlled variable u. The disturbance estimation consists of two steps.

In the former step, known as prediction step, the future state and output are predicted by means of the augmented model:

$$\begin{aligned} \hat{\xi}^*(k) &= A_a \hat{\xi}(k-1) + B_a u(k-1) \\ \hat{y}^*(k) &= C_a \hat{\xi}^*(k) \end{aligned} \tag{6.11}$$

$\hat{\xi}^*(k)$ and $\hat{y}^*(k)$ denote the predicted state and output, respectively. In the latter, known as filtering step, the augmented state $\xi(k)$ is estimated exploiting the measurements y, and the estimated state is denoted as $\hat{\xi}(k)$:

$$\hat{\xi}(k) = \hat{\xi}^*(k) + K_a e(k) \tag{6.12}$$

where $e(k) = y(k) - \hat{y}^*(k)$ indicates the output prediction error and $K_a = [K_x; K_d] \in \mathbb{R}^{n+n_d}$ is the observer gain matrix for the augmented system (the block K_x multiplies the state x, whereas K_d multiplies the disturbance). The two steps (6.11) and (6.12) can be also condensed in the following single equation:

$$\hat{\xi}(k) = A_a \hat{\xi}(k-1) + B_a u(k-1) + K_a e(k) \tag{6.13}$$

The asymptotic stability of the observer (6.13) is guaranteed if the matrix $[A_a - K_a C_a A_a]$ (i.e., the observability matrix) has all eigenvalues within the unit circle. In [73, Proposition 17], it is reported the procedure for designing offset-free MPC schemes using a disturbance model.

6.4.2 *Velocity Form*

The velocity form, named I-MPC, exploits the definitions of the state increment $\Delta x(k) = x(k) - x(k-1)$ and input increment $\Delta u(k) = u(k) - u(k-1)$. The nominal model can be rewritten introducing an augmented state variable $\xi_\Delta = [\Delta x; y]$:

$$\begin{aligned} \xi_\Delta(k+1) &= A_\Delta \xi(k) + B_\Delta \Delta u(k) \\ y(k) &= C_\Delta \xi(k) \end{aligned} \tag{6.14}$$

where $A_\Delta := \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix}$, $B_\Delta := \begin{bmatrix} B \\ CB \end{bmatrix}$ and $C_\Delta := \begin{bmatrix} 0 & I \end{bmatrix}$ In general, the outputs y are measured, whereas the state Δx may not be measurable. In this case a state observer is required too. According to the velocity form approach, the disturbance is estimated by means of the same Kalman filter presented in (6.11) and (6.12), which can be written as:

$$\hat{\xi}_\Delta(k) = A_\Delta \hat{\xi}_\Delta(k-1) + B_\Delta \Delta u(k-1) + K_\Delta e(k) \tag{6.15}$$

where $K_\Delta := [K_{\Delta_x}; K_y]$. Since the outputs are measured in our application, it is a common choice to set $K_y = I$, thus $\hat{y}(k) = y(k)$. It follows that the matrix $[A - K_{\Delta_x} CA]$ needs to have the eigenvalues inside the unit circle to have an asymptotically stable observer.

6.4.3 *Equivalence Between Velocity Form and Disturbance Model and Observer*

The equivalence result between the velocity form and the disturbance model/observer form was shown firstly in [76]. The same authors gave a final and complete description in [73]. The most relevant result of the work, is here reported:

Theorem 6.4.1 *[73, Theorem 22] Consider the velocity form model (6.14) and observer (6.15), with a stable output observer gain $K_\Delta = [K_{\Delta_x}; I]^T$. This is equivalent to use the following disturbance model and observer gains:*

$$B_d = K_{\Delta_x};\ C_d = I - CK_{\Delta_x};\ K_x = K_{\Delta_x};\ K_d = I \tag{6.16}$$

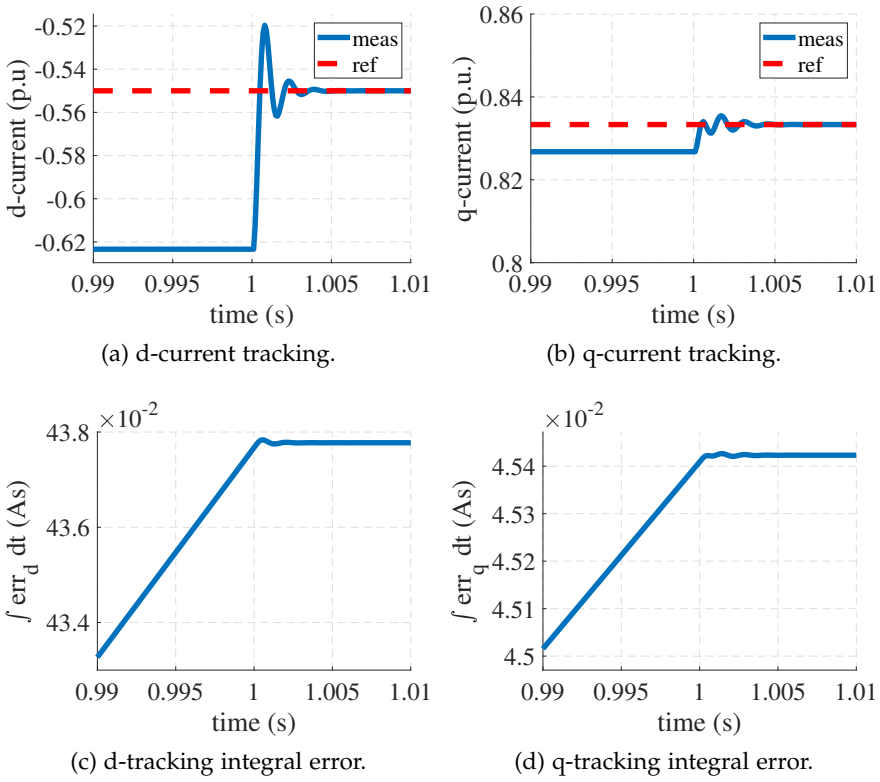
Moreover, if the state is measurable, the matrices become:

$$B_d = I;\ C_d = 0;\ K_x = I;\ K_d = I \tag{6.17}$$

Assuming (6.17), the observability matrix $[A_a - K_a C_a A_a]$ has all its eigenvalues equal to zero.

6.5 SIMULATION RESULTS

In this section, the effectiveness of the integral action in the control loop is pointed out. This is shown by considering the effect of parameter mismatches neglecting the velocity form or the presence of



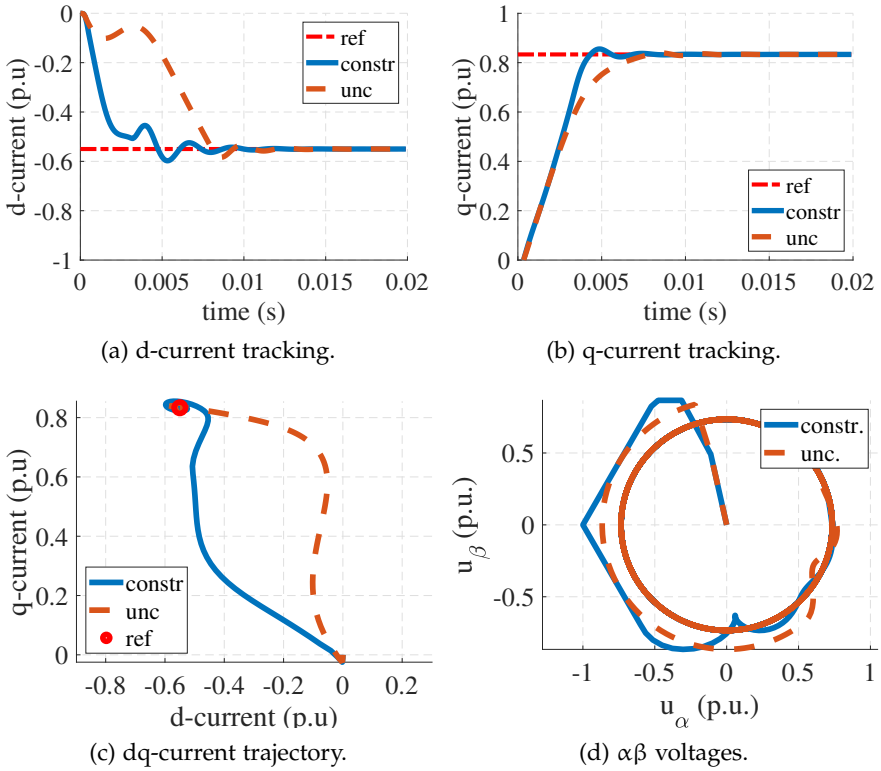
(a) d-current tracking.

(b) q-current tracking.

(c) d-tracking integral error.

(d) q-tracking integral error.

Figure 6.2: MPC to I-MPC switch: L_d and L_q mismatch. The inductance parameters of the motor are halved. At first, the current tracking is performed using MPC with no observer. At t=1s the I-MPC is selected.



(a) d-current tracking.

(b) q-current tracking.

(c) dq-current trajectory.

(d) αβ voltages.

Figure 6.3: I-MPC: Unconstrained vs Constrained implementation. Voltage are normalized with respect to $2/3U_{DC}$.

a state observer. Secondly, a comparison between the unconstrained and constrained formulation of the I-MPC is given in transient conditions. Finally, tuning guidelines are provided.

6.5.1 *Benefits of the Integral Action in Presence of Parameter Mismatches*

The Interior Permanent Magnet (IPM) motor parameters are reported in Table 2, whereas the motor lamination is shown in Figure 6.4. The control uses the model (6.1) to predict future currents and the algorithm is executed in discrete time, with sample period $T_s = 100$ µs. The simulation is performed as follows: the IPM motor speed is kept constant; in the first part, the MPC with velocity form or observer has been used as current controller. Then, a switch with the I-MPC algorithm is performed. This allows for comparing the two strategies, focusing on the dq current reference tracking and the integral action of the current error between references and measurements.

Thanks to commissioning procedures, both the inductance L_d and L_q change of almost a factor 2 inside the feasible operating current range of the machine. Thus, motor inductances during simulations are halved, approximating the steady-state full load operation. In Figure 6.2(a) and Figure 6.2(b), it is possible to notice that, after the

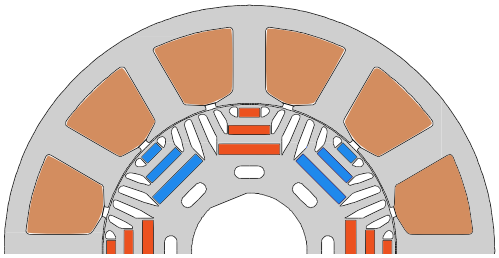


Figure 6.4: Structure of the IPM machine prototype (12 slots, 8 poles).

switch, I-MPC compensates effectively the steady state error in the reference tracking. Moreover, in Figure 6.2(c) and Figure 6.2(d), the rise of the error integral is stopped when I-MPC is applied at instant 1 s. This means that I-MPC grants an integral action and so an unbiased tracking reference.

6.5.2 *Unconstrained vs Constrained Implementation*

The main benefits of implementing a constrained version of I-MPC controller are obtained when the solution of the unconstrained problem exceed the voltage constraints. In this condition, the policy adopted for the unconstrained solution is to project the optimal voltage computed keeping constant the ratio between its components. The maximum value of the voltage vector magnitude is $U_{DC}/\sqrt{3}$, i.e. the value of the radius of the circle inscribed in the hexagonal region shown in Figure 6.1. This policy does not assure optimal results, as proved by the simulation summarized by Figure 6.3. The motor is driven in current control mode, keeping its speed constant at the nominal value. A current reference equal to the nominal Maximum Torque per Ampere (MTPA) point is applied.

The current transient is much faster in the constrained case for both the d and q current components (Figure 6.3(a) and Figure 6.3(b)). This is reasonably due to the fact that unconstrained I-MPC does not entirely exploit the available DC-bus voltage (Figure 6.3(d)). Nevertheless, the selection of a consistently different current trajectory greatly influences the transient (Figure 6.3(c)). Projecting the voltage maintaining the phase, in fact, avoids the fact that the d-axis current has a faster dynamic, due to a lower inductance (Table 2). This justifies the constrained I-MPC choice of increasing the d current value first, inducing a trajectory which is longer in the dq current plot (Figure 6.3(c)), but actually much faster in the time domain.

6.5.3 *Tuning Guidelines*

One of the main issue about MPC design and optimal control in general is how the choice of weight matrices in (6.2) affects the con-

Table 2: Interior permanent magnet motor nominal parameters.

IPM Motor Data			
Pole Pair Number		4	
Phase resistance	R	1.5	Ω
Direct inductance	L_d	34	mH
Quadrature inductance	L_q	86	mH
Nominal current	I_N	6	A(pk)
PM flux-linkage	Λ_{pm}	0.2	Vs
DC-bus voltage	U_{DC}	300	V
Nominal Power	P_N	1	kW
Nominal Speed	n_N	1000	rpm

trol performances. The answer is still not uniform in literature, and several heuristic techniques have been presented. In this work, the penalty on the tracking terms (i.e. the weight matrices Q and S) have been set equal to the identity matrices, and the matrix R that penalizes the control input increments is manually tuned. The relative weights among this matrices determine the control performance. On one hand, a relatively low penalty on the control input compared to the one on tracking term $y^* - y$ causes a faster current dynamics, but with the drawback of consuming more energy (i.e. the control algorithm will find higher values of Δu) and current overshoots are emphasized. On the other hand, larger values of R reduce the overshoot, having an impact also on the current rising-time.

The effect of the regularization term on the control action can be evaluated by considering the MPC problem (6.2) written for $N = 1$. Moreover, it is assumed that the input constraints are satisfied. The MPC in (6.2) can be condensed in matrix form:

$$\min_{\Delta u} \frac{1}{2}\Delta u^T H \Delta u + \Delta u^T c \tag{6.18}$$

where

$$H = 2(B^T Q B + R) = 2\begin{bmatrix} (\frac{T_s}{L_d})^2 q + r & 0 \\ 0 & (\frac{T_s}{L_q})^2 q + r \end{bmatrix} \tag{6.19}$$
$$c = -2B^T Q(y^r - Ax(k) - Bu(k-1) - d(k))$$

The matrix C has been omitted since it is the identity matrix. If $r = 0$, the algorithm performs as a dead-beat predictive controller. In fact, the optimal solution would be (ignoring the term Q):

$$\Delta u^* = -H^{-1}c = B^{-1}(y^r - Ax(k) - Bu(k-1) - d(k)) \tag{6.20}$$

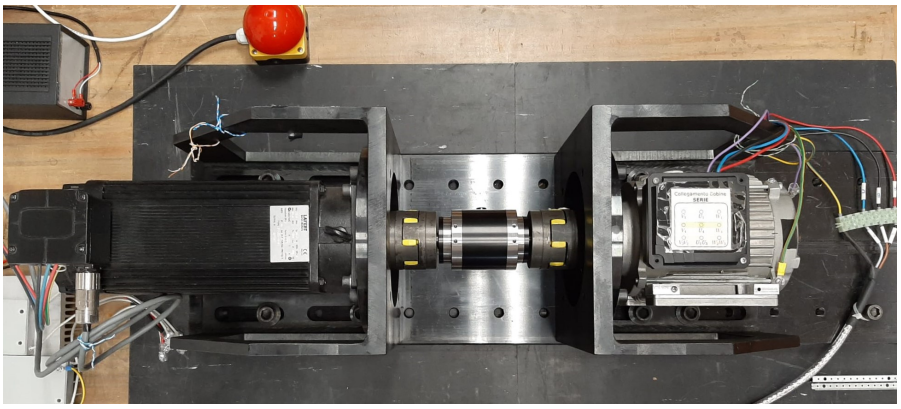


Figure 6.5: Test bench.

and the value is computed in order to achieve the reference in one time step interval, i.e. the control input is computed based on the error between the reference and the predicted output. However, the control input variation is introduced to mitigate this effect. In fact, choosing $r \neq 0$ the solution is weighted by:

$$\Delta u^* = \begin{bmatrix} \frac{1}{(\frac{T_s}{L_d})^2 q + r} & 0 \\ 0 & \frac{1}{(\frac{T_s}{L_q})^2 q + r} \end{bmatrix} c \tag{6.21}$$

Now, the control input variation can be modified by acting on r. As can be seen from (6.21), the value of r should depend on the sampling time T_s and the inductance values. As long as the r coefficient is increased, the current dynamics is slowed. This effect is typical of transients, where large values of Δu are usually required. The computation with a larger prediction horizon is more complicated, but the same idea holds.

In the literature, finding a systematic way to design control parameters when constraints are included is still a challenge. Thus, a rigorous calculation of tuning weights is found to be cumbersome. From the author's perspective, there is always a mismatch between models which are used in simulations for finding the tuning parameters, and the real test bench experiments. The model always partially represents the real physics. The effects of noise, external disturbances, and sensor accuracy are just a few examples of phenomena that are very difficult to represent exactly in the simulation environment. Thus, the tuning parameters are always adjusted in the experimental stage. Previous considerations provide guidelines to find the order of magnitude of the regularization term, which is often related to the values of the dq inductances.

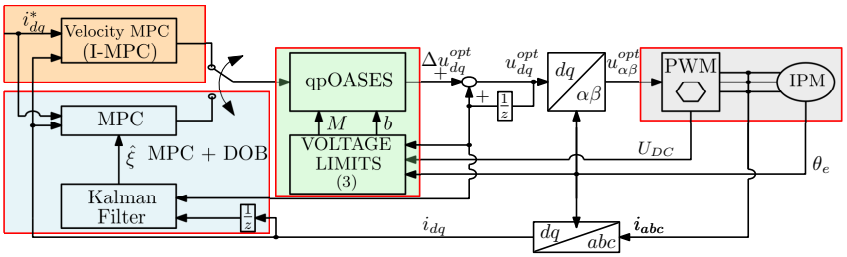


Figure 6.6: Control scheme architectures.

6.6 EXPERIMENTAL RESULTS

The test bench in Figure 6.5 consists of a master Surface Permanent Magnet (SPM) motor directly connected to the motor under test. The drive has been tested experimentally using a dSPACE MicroLabBox, a compact development system for laboratory purposes which has dual-core real-time processor at 2 GHz and dedicated electric motor control features, which provides the six commands to the inverter. After initialization, the program reads the phase currents from the inverter current sensors, the mechanical speed and rotor position from the encoder placed on the motor shaft. Speed is hold by the master motor, while the current loop controller (MPC/I-MPC) calculates the voltage references for the Pulse-Width-Modulation (PWM), which gives the duty cycle signals for the six gates of the inverter connected to the motor under test. Current and speed measurements are synchronized with the PWM.

The first part of this section is devoted to show the effectiveness of the integral action introduced by I-MPC scheme. Tests have been carried out considering the velocity form MPC formulation and the MPC without any disturbance observer. The effect of parameter mismatch is analyzed. A schematic representation of the control scheme architecture is shown in Figure 6.6. The experiments on dynamic performance and parameter sensitivity have been conducted with the already presented IPM motor (see Table 2).

The second part of this section considers the comparison between the I-MPC and the MPC with steady-state Kalman filter as DOB. The aim is to show the effectiveness of both control solutions with a particular attention to numerical aspects. Regarding these two control schemes, the computational burden in the worst-case scenario (i.e. when the solver performs the maximum number of iterations) is about 70 µs for both the algorithms. These tests are carried out considering a PWM switching frequency of 8 kHz to allow the implementation of the constrained controller on the dSPACE MicroLabBox. This choice allows the controller to be implemented with $N = 3$. The computational cost of the MPC is strictly related to the prediction horizon length. The dimension of the problem grows as far as this parameter increases. Thus, the computation time required by the

solver is highly correlated with the selected prediction horizon too. The computational effort can be reduced if a customized solver is designed, allowing the prediction horizon, the sampling frequency or both to be increased, with benefits in term of dynamic responses and disturbance rejection properties. This study will be presented in Chapter 7.

6.6.1 *Sensitivity Analysis*

It is worth mentioning that the adopted IPM motor exhibits a high harmonic content on the back-EMF, since it is designed with a tooth wound stator winding. Hereafter, a comparison between the two proposed predictive control strategies is shown in terms of parameter sensitivity.

As just mentioned, the main advantage of I-MPC over MPC is the integral action that vanishes bias errors in the reference tracking when constant disturbances, as parameter mismatches, are presented in the control loop. In Figure 6.7, a complete sensitivity analysis of all the parameters presented in both MPC and I-MPC controllers is shown. Nominal control values of L_d, L_q, R and Λ_{pm} have been changed of 10%, 20%, 50% and 100% during a steady state MTPA working point condition, operating at 500 rpm. As can be noticed, MPC shows both bias and variance increasing errors in tracking reference when the disturbance grows. Only the resistance R does not badly affect the control performance, as expected. On the other hand, I-MPC seems to be robust to all the parameters mismatches. However, it is worth mentioning that there are cases, where a constant disturbance can increase the variance error in tracking a certain reference, as shown in next paragraphs. In fact, even offset-free MPCs have their intrinsic bandwidth limits and thus, if the disturbance is too severe, variance errors arise and the stability can be compromised.

6.6.2 *MPC Performance Comparison*

In this section, the offset-free MPC with disturbance observer (Section 6.4.1) designed as in Section 6.4.3 and the I-MPC (Section 6.4.2) schemes are compared. For this purpose, the IPM motor has been tested in different operating conditions. Three tests are presented: a current transient, a sensitivity analysis in steady-state and during a speed transient. These experiments have been conducted using the regularization term in the cost function (6.2) equal to $R = 0.003I$ for both the schemes, where I is the identity matrix of suitable dimension. The dq currents have been normalized with respect the nominal peak current in Table 2.

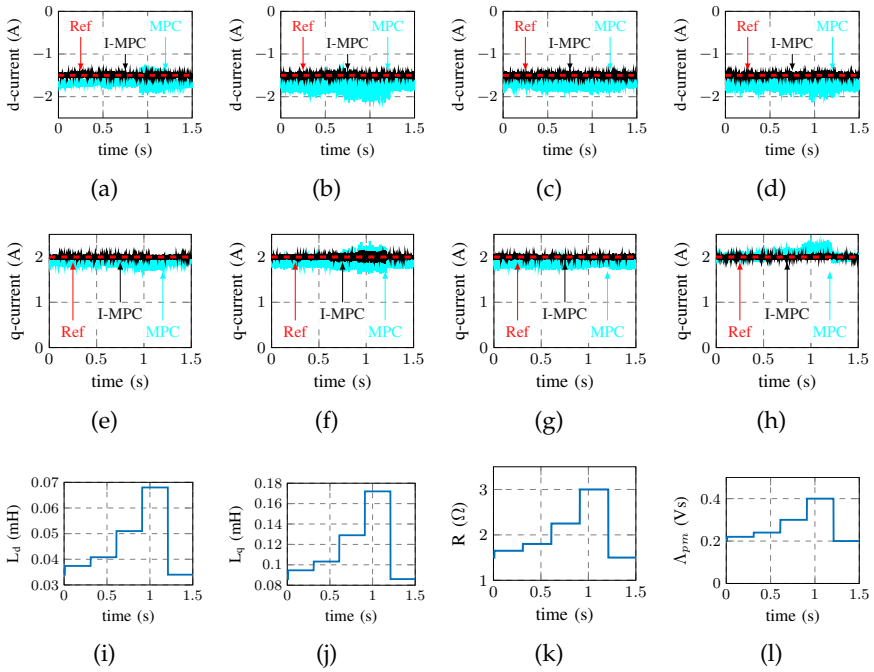
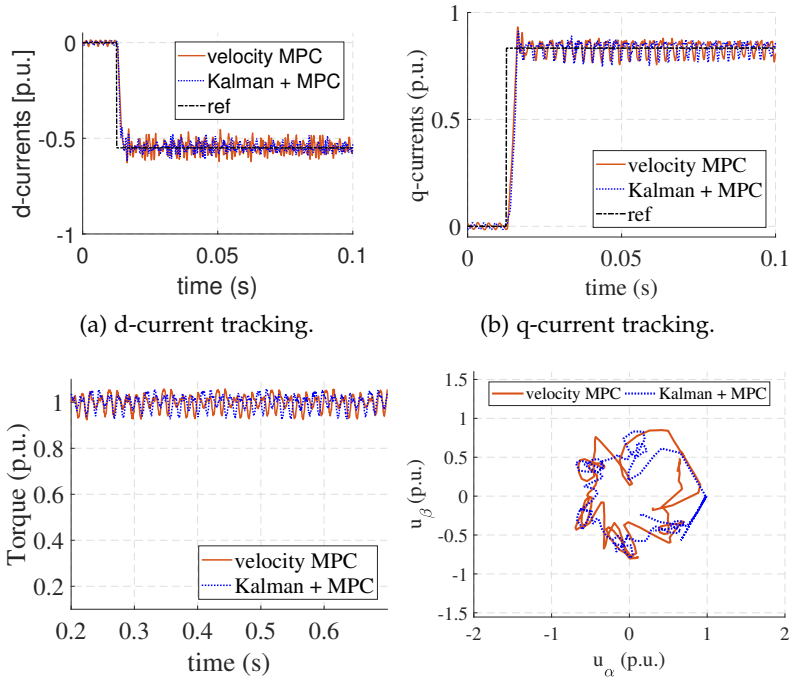


Figure 6.7: Parameter sensitivity test: nominal control values of L_d ((a),(e),(i)) L_q ((b),(f),(j)), R((c),(g),(k)) and Λ_{pm} ((d),(h),(l)) have been changed of 10%, 20%, 50% and 100% during a steady state MTPA working point condition, operating at 500 rpm.

6.6.2.1 *Current Transient*

A current step transient is shown in Figure 6.8, where an MTPA working point has been set as reference. The test is performed the drive base speed. The two MPC schemes have the same performance as confirmed by the dq current transients in Figure 6.8(a) and Figure 6.8(b). Furthermore, the steady-state torque waveform is reported in Figure 6.8(c). The oscillations are mainly due to the master PMSM that introduces harmonics. Figure 6.8(d) proves that the solver finds the optimal solution to the voltage limits in the first instants. The number of iterations required by the solver for finding the solution during the transient are the same for both MPC control schemes. The test has been repeated at 10% of the base speed and it is reported in Figure 6.9. By comparing Figure 6.8(a), Figure 6.8(b) and Figure 6.9, it is possible to note that the speed and the external mechanical system represented by the master motor impacts on the control performance. In fact, both the MPC solutions show very good tracking performance at low speed with a lower harmonic contents. However, the dynamic is influenced by the lower value of the back-EMF. As can be seen from Figure 6.9, the two algorithms show an overshoot and an undershoot, then the controllers stabilize the currents at the desired reference. The different behavior can be justified by the choice of the tuning parameters, which have been selected in order to satisfy acceptable perfor-



(a) d-current tracking.

(b) q-current tracking.

(c) Measured Torque.

(d) Voltage references.

Figure 6.8: Step current at nominal speed.

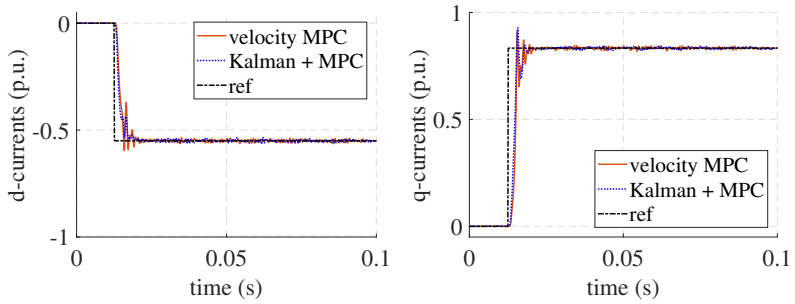
mance in all the speed range. This allows both to maintain a simple control structure and to perform offline hessian computations. Better results could be obtained with the addition of a feed-forward term, but this goes beyond the scope of the work. With the presented experiments, it has been shown that both the algorithms have similar performance during transients, whose dynamics almost overlap.

6.6.2.2 L_q *Sensitivity*

The sensitivity analysis test is shown in Figure 6.10. The value of L_q has been doubled at time instant 0.18 s, similar as before, during a steady-state operating condition. This test also confirms that the two control schemes operate in the same manner. Both solutions are able to get rid of this parameter mismatch without introducing an offset in current tracking. In particular, the d-axis current is not affected by L_q mismatch. However, the q-axis current increases its variance error.

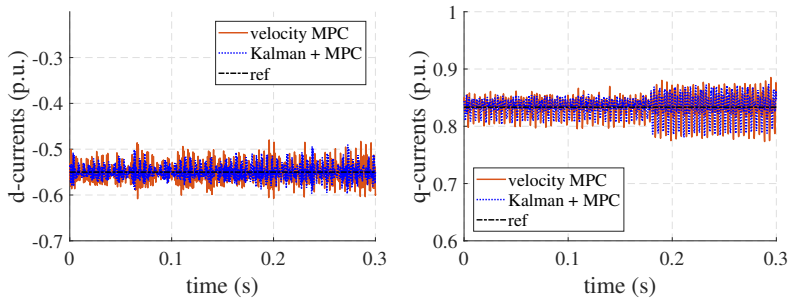
6.6.2.3 *Speed Variation*

The last test aims at showing the effect of speed variation in current tracking capacity of the two MPC scheme, combined with a L_q parameter variation. The test is performed as follows: the current MTPA working point is set as current reference and steady-state condition is reached. Then, starting from 100 rpm, the master motor performs a speed ramp up to 700 rpm. Results of this test are shown in Fig-



(a) d-current tracking. (b) q-current tracking.

Figure 6.9: Step current at 10% of the nominal speed.



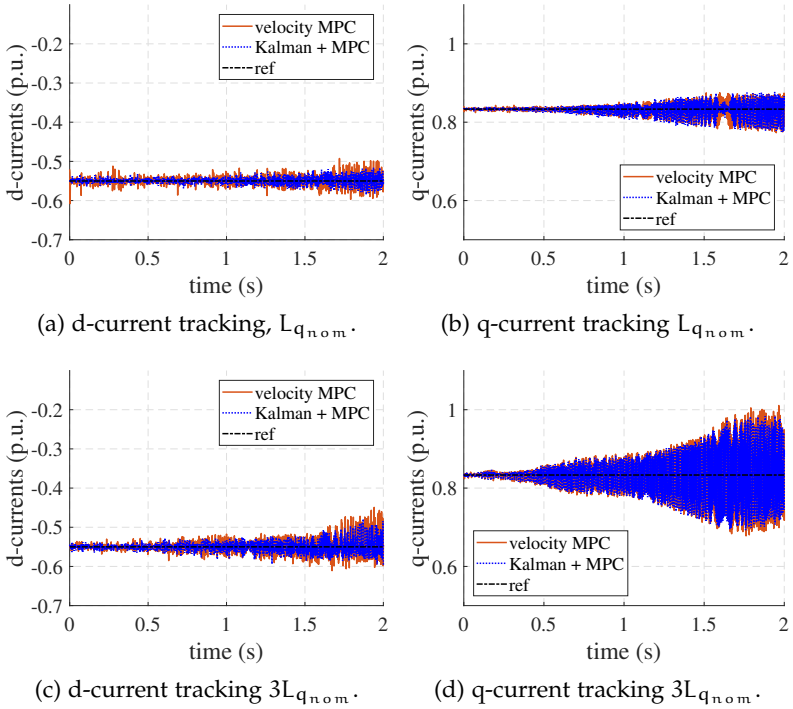
(a) d-current tracking. (b) q-current tracking.

Figure 6.10: Steady state test. The L_q value of the control is doubled at time t =0.18 s.

ure 6.11. The Figure 6.11(a) and Figure 6.11(b) report the dq current tracking during the speed transient, using the nominal value of the q-axis inductance. Instead, Figure 6.11(c) and Figure 6.11(d) represent the same test, but using an L_q value which is three times the nominal one. This test highlights some limitations of these algorithms. In fact, the combination of the speed variation with the discrepancy of the L_q parameter negatively affects the performance of the control. The variation error increases slightly with speed while maintaining the L_q nominal value. However, the oscillations increase significantly if there is a large deviation of the parameter, especially in the current of the q axis. Acceptable performances could be obtained by using different weighting factors in the cost function for different speed values.

6.6.3 *Computational Cost*

One of the most important aspects in MPC applications is the computational cost of the algorithm. This depends on several factors, such as the choice of the solver, the conditioning number of the matrix that must be inverted to solve the optimization problem and the type of DOB that is used to eliminate offset errors. In fact, one should also



(a) d-current tracking, $L_{q_{nom}}$. (b) q-current tracking $L_{q_{nom}}$.

(c) d-current tracking $3L_{q_{nom}}$. (d) q-current tracking $3L_{q_{nom}}$.

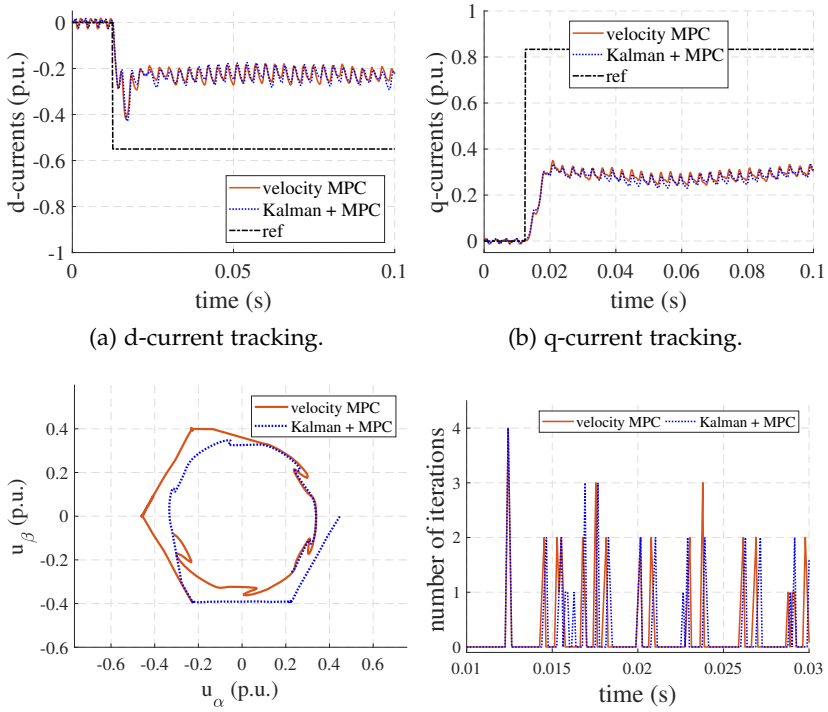
Figure 6.11: Speed ramp from 100 to 700 rpm. Comparison between $L_{q_{nom}}$ and $3L_{q_{nom}}$.

include the calculation time required by the DOB, because it is part of the standard MPC algorithms.

In this work, both I-MPC and MPC with steady-state Kalman filter adopt `qpOASES` as solver. The authors decided to implement `qpOASES` because it is a free, widely used, open source QP solver (thus it permits the reproducibility) and it allows the online implementation on the platform dSPACE MicroLabBox. Regarding the QP conditioning, it has been observed that for this particular configuration the velocity form results of the same conditioning of the standard formulation, since the Hessian of the cost function and the constraint matrix have the same condition number. In particular, the Hessian matrices have a condition number approximately equal to one, which indicates that the problem is well conditioned. Therefore, even working with a single precision, an accurate solution can be guaranteed.

The linear term of the cost function is computed for an incremental state in the case of the I-MPC. This can badly affect the convergence speed of the iterative solver, which is handling the constrained QP problem. However, the authors studied the number of iterations required by the solver during limit conditions for both I-MPC and MPC coupled with steady-state Kalman filter. The results of test are shown in Figure 6.12.

The test procedure is as follows: the IPM motor is hold at nominal speed by the master motor. The DC value of the bus is 300 V. However,



(a) d-current tracking.

(b) q-current tracking.

(c) Voltage references.

(d) Solver iteration number.

Figure 6.12: Iteration test: comparison between velocity MPC and MPC with steady-state Kalman filter. Unfeasible references are set and the solver finds solutions at voltage limits.

its value is limited for the control to 150 V via software. Therefore, a current reference is given that cannot be realized by the current control. As can be seen from Figure 6.12(a) and Figure 6.12(b), the I-MPC and the MPC with Kalman cannot reach the references, and the system stabilizes its working point accordingly with the bus limit. In fact, Figure 6.12(c) indicates that both controllers find optimal solutions to the voltage limits. It is worth noting that, in Figure 6.12(c), there is a drop of the voltage in the commutation between phases due to the motor inductances, which refers to the voltage notch [77]. The number of iterations performed by `qpOASES` are compared in Figure 6.12(d), and shows that they are comparable. This confirms that both strategies work with the same computation time. The number of iterations indicates the convergence ratio of the solution. As can be seen, when no iterations are performed, the controller acts as a Linear Quadratic Regulator (LQR); i.e. when the unconstrained solution is still feasible and no constraints are violated. Further numerical properties of these algorithms could be investigated by considering other solvers.

6.6.4 *Final Considerations and Remarks*

The experimental results are summarized with some considerations concerning the case for which one of the proposed control architecture may be preferred with respect to the other. One of the most important concerns about MPC design in industrial applications (a part of the computational aspects) arises from the parameter choice of the controller. If a Kalman filter-type observer is selected, one should consider which type of noise affects the system. In particular, an estimation of the covariance matrices for model and process noise is required. This means that the size of the parameter vector increases. From the authors' perspective, reducing the number of tuning parameters leads to an attractive method for industrial applications. In particular, when the number of parameters to be tuned want to be reduced (thus making the process simple also for non-high specialized personnel) and when the performance requirements are not so demanding, the velocity form MPC should be preferred for its simplicity.

In this work, the focus has been the MPC framework where both observer and velocity form can be applied. Nevertheless, the observer can also be used in other control architectures, whereas the velocity form is restricted to the MPC framework. Instead, in the case that a specific closed-loop performance is required and, for instance, accurate knowledge of sensors is also available, the implementation of MPC with DOB is preferred. The author is also aware that very recent publications in the control community are going in the direction of "self-calibration" of optimization-based controllers [78]. This can

represent a promising strategy to overcome the *curse* of MPCs tuning. It is highlighted that the aim is not to defend one solution over the other, but to find some interesting and practical aspects which can be of interest for the scientific community and industrial applications.

6.7 DISCUSSION

In this work the velocity form of a continuous control set model predictive controller has been proposed to control the current loop of an interior permanent magnet synchronous motor. This formulation has been compared with the MPC algorithm coupled with a disturbance observer, using a steady-state Kalman filter. The theoretical analysis of these two schemes has been reported, showing that the incremental state space formulation, which characterized the velocity form of MPC, corresponds to a particular choice of the standard MPC with the disturbance observer. These formulations allow overcoming traditional issues known for model predictive control approach, such as steady state bias errors due to parameter uncertainties and constant external disturbances. Theory validation is achieved by comparing the two controllers in different operating conditions. In particular, the control behavior in case of parameter mismatches has been studied. The proposed current MPC in velocity form is able to compensate the steady state error due to mismatches, rejecting bias errors. Furthermore, tuning guidelines for the MPC have been provided. Moreover, it has been shown that the two MPC versions have similar performances. Thus, the velocity MPC can be considered as an easy and quick method for designing offset-free MPCs. The computational cost of the schemes has been analyzed through experimental results, using the open source solver `qpOASES` for the optimization problem. Future research will consider a multilevel converter topology, which usually presents lower harmonic content. In addition, other solvers and a customized design of the entire controller will be considered to further investigate numerical properties.

7

FAST SOLVER FOR IMPLICIT CONTINUOUS SET MODEL PREDICTIVE CONTROL OF ELECTRIC DRIVES

This chapter presents an effective method for solving Continuous Control Set (CCS)-Model Predictive Control (MPC) for the current control loop of synchronous motor drives with input constraints, allowing for reaching the maximum voltage feasible set. The related quadratic programming problem requires an iterative solver to find the optimal solution. The real-time certification of the algorithm is of paramount importance to move the technology toward industrial-scale applications. The total number of operations can be computed in the worst-case scenario; thus the maximum computational time is known a priori. The solver is deeply illustrated, showing its feasibility for real-time applications in the microseconds range by means of experimental tests. Promising results are obtained with respect to general purpose solvers.

7.1 INTRODUCTION

MPC is an advanced optimization-based control strategy that is gaining more and more popularity in the power electronics framework [8, 28]. Increasing availability of computational power and enhancement of optimization strategies have made MPC suitable for fast dynamics, as electric drives. In this framework, MPC has been mostly implemented in the Finite Control Set (FCS) form [79–81]. As been reported in Chapter 3, second type of MPC refers to the CCS-MPC technique, which requires the modulator for synthesizing the optimal voltage reference. However, it provides fixed switching frequency of the converter and it works efficiently with longer sampling intervals [82]. Several improvements on the computational performances of solvers adopted for handling the optimization problems underlying the MPC led to the establishment of this technology. The MPC problem can be formulated in explicit or implicit form, whose similarities and differences are discussed in [60] (see also Chapter 3). The explicit form considers the optimization problem as a piece-wise linear function of the parameters and solves it offline [19]. However, the computational cost grows significantly with the parameter space [83], thus limiting its applicability for industrial applications. Implementation of explicit MPCs is proposed and analyzed in [57, 59]. The implicit form, instead, requires an online iterative algorithm for solving the Quadratic Programming (QP) problem. A significant reduc-

Table 3: Overview of some QP solvers. The acronyms are: IP: Interior point; AS: Active Set; ADMM: Alternating Method of Multipliers, FGM: Fast Gradient Method.

Solver	References	Target	Methods	Licence
`FORCES Pro`	[84]	NMPC, very general QPs	IP, ADMM, FGM	proprietary
`HPIPM`	[85]	LQP	IP	2-clause BSD
`ODYS`	[83]	QP, Embedded MPC	AS	proprietary
`OSQP`	[86]	QP	ADMM	Apache 2.0 License
`qpOASES`	[87]	QP	AS	2-clause BSD

Table 4: Survey of MPCs for electric drive applications. The acronyms are: LTI: Linear Time Invariant; LPV = Linear Parameter Variant; NL: Non Linear; E-MPC: Explicit-MPC.

References	Year	Controlled Variable	Model	Method	Constraints		Tested Motors	Platform
					Input	Output		
[57]	2009	Speed&Currents	LTI	E-MPC	✓	✓	SPM	dSPACE 1004
[88]	2012	Speed+Torque/Flux	LPV	E-MPC	✓	✓	IM-PMSM	Sharc ADSP 21062 + TMS320C6713
[58]	2013	Currents	LPV	FGM	✓		IPM	TMS320F240
[61]	2015	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[89]	2019	Currents	LPV	AS	✓		IPM	dSPACE 1006
[83]	2021	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[62]	2021	Currents	NL	AS	✓		SRM	dSPACE 1007

tion of the computational burden of CCS-MPCs has been achieved by implementing the unconstrained version of the QP problem, where a closed-form expression of the CCS-MPC solution exists. However, this approximation suppresses the MPC advantage of including feasible limits in the control law. This has spurred a renewed research interest towards constrained solutions.

The development of efficient QP solvers is now enabling the real-time implementation of CCS-MPCs. Gradient methods [58, 90] and active set methods [87] are the two most widespread kind of solvers for the considered application. In Table 3, a set of existing QP solvers and some of their principal features are reported. Software packages containing solvers for fast embedded optimization are available too, e.g., `acados` [91]. Moreover, solver-libraries designed for MPC problems have been presented, e.g., `MATMPC` [92].

The constrained CCS-MPC for electric motor drive applications has gained interest over the last decade. Table 4 resumes some of the most relevant papers related to this topic. In [83], an active set solver has been successfully implemented on a Texas Instrument Digital Signal Processor (DSP) for CCS-MPC torque control of a Permanent Magnet Synchronous Motor (PMSM), proving that the technology is mature for industrial applications. The work is based on the results of

[93], where the same authors provided the exact complexity certification of the Golfarb-Idnani algorithm [94]. Thus, in these works, an upper bound of the computation time in the worst-case scenario has been demonstrated. Moreover, in [62], a nonlinear CCS-MPC is presented for the Synchronous Reluctance Motor (SyRM). The work proves the flexibility of the MPC framework in tackling the nonlinear flux-current characteristics of such motors. However, the QP solver adopted for the specific application is a general purpose one, which is effective at the price of an increased complexity. Nevertheless, the computational burden is of paramount importance for embedded applications, and the emerging need from recent research advances is to design purpose-built algorithms.

In this work, we propose an efficient and fast method for solving the specific QP problem arising from MPC implementations for applications where a power converter is adopted. In particular, the algorithm is presented for the current control of synchronous motors, where limited computational hardware is usually available. The MPC implementation adopts a linear time-invariant model of the dynamic and considers linear input voltages constraints.

The model is formulated in the dq rotating reference frame, making the voltage feasible set rotating synchronously with the rotor position. The proposed QP solver takes advantage of the specific shape of the feasible set and the cost function to achieve a computationally efficient formulation that allows for exploiting the maximum voltage deliverable by the converter, i.e., a hexagonal region in the stationary reference frame, centered in the origin. The proposed algorithm adopts the choice of $N = 3$ prediction steps and $N_u = 1$ control horizon length, which has been assessed as a good trade-off between accuracy and computational effort [83].

A complete description of the QP solver is presented, allowing an easier replication of the algorithm. Moreover, computational performances in the worst-case scenario are assessed both in terms of number of operations and in experimental conditions. The high accuracy of the algorithm is then assessed comparing with `qpOASES`.

7.2 MATHEMATICAL MODEL

The electrical dynamic equations of the synchronous motor is described with respect to the rotating dq-reference frame as follows (for simplicity we omit the time-dependence) [95]:

$$\frac{d}{dt} i_{dq} = A_c i_{dq} + B_c \left(u_{dq} + w_{dq} \right), \tag{7.1}$$

where $i_{dq} = [i_d \; i_q]^T$, $u_{dq} = [u_d \; u_q]^T$, $w_{dq} = [w_d \; w_q]^T$ and

$$A_c = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_e \frac{L_q}{L_d} \\ -\omega_e \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix}, \quad B_c = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix}, \tag{7.2}$$

where u_d, u_q, i_d and i_q are the dq-axis voltages and currents, respectively, R_s is the windings resistance, ω_e the electrical speed, whereas L_d and L_q are the dq-axis stator inductances. Assuming perfect knowledge of the parameters and neglecting model inaccuracies, the vector w_{dq} includes only the back-Electro Motive Force (back-EMF), i.e., $w_{dq} = [0 \; -\omega_e \lambda_{pm}]^T$, being λ_{pm} the permanent magnet flux linkage.

Discretizing model (7.1) adopting Euler integration with sampling time T_s, yields to the following discrete-time model

$$x(k+1) = Ax(k) + Bu(k) + Bw(k), \tag{7.3}$$

where $B = T_s B_c$, $A = I + T_s A_c$, being I the identity matrix, and where $x = [i_d, i_q]^T$ denotes the state and the subscript on $u(k)$ and $w(k)$ are neglected for simplicity.

In practice, significant unmodeled disturbances and nonlinear dynamics are present, due, e.g., to iron saturation, parasitic effects and harmonic modes introduced by the non-ideal rotor geometry, and to the speed dependence of matrix A_c, which has been obtained by linearizing around the nominal speed. To address this issue in real experiments an observer is implemented to obtain an estimate of both $x(k)$ and $w(k)$ for all k, thus improving the tracking performance of the controller and guaranteeing offset free tracking. More precisely, the state of the model (7.3) is augmented with the disturbance w, and it is estimated via a Kalman Filter as proposed in [96].

7.3 MODEL PREDICTIVE CONTROL OF PMSM CURRENTS

The Linear Time Invariant (LTI) model (7.3) is used as the prediction model in the MPC problem, whose ultimate goal is to track the desired currents profiles $x^{ref} = [i_d^{ref} \; i_q^{ref}]^T$.

The following quadratic functional cost is adopted

$$\begin{aligned} J = &\sum_{j=0}^{N-1} \left(\|x^{ref}(k+j+1) - x(k+j+1)\|_Q^2 \right) + \\ &+ \sum_{i=0}^{N_u-1} \left(\|\Delta u(k+i)\|_R^2 \right) + \|x^{ref}(k+N) - x(k+N)\|_S^2 \end{aligned} \tag{7.4}$$

where Q, S and R are weighting matrices and $\Delta u(k)$ denotes the differential input, i.e., $\Delta u(k) = u(k) - u(k-1)$.

At each time step k, the optimal control move is obtained by solving the following optimal control problem:

$$\begin{aligned} &\min_{\Delta u, x} \quad J \\ &\text{s.t. } x(k+1) = Ax(k) + Bu(k) + Bw \\ &\qquad \Delta u(k) = u(k) - u(k-1) \\ &\qquad u_{\alpha\beta}(k+j) = T(\theta_e)u(k+j) \in \mathbb{U}_{\alpha\beta},\ j = 0, 1, ..., N_u \end{aligned} \tag{7.5}$$

where $T(\theta_e)$ is the Park anti-transformation matrix, being θ_e the electric angle, $\mathbb{U}_{\alpha\beta}$ represents the feasible voltage region for $u_{\alpha\beta} = \begin{bmatrix} u_\alpha & u_\beta \end{bmatrix}^T$, and where the term w is assumed to be constant in the prediction horizon and equal to the estimate $\hat{w}(k-1)$ provided by the Kalman filter at the previous step.

Observe that $\mathbb{U}_{\alpha\beta}$ is the maximum voltage deliverable by the converter, i.e., an hexagonal region in the stationary reference frame, centered in the origin. Effective methods to manage the transition from sinusoidal output voltage with a linear voltage gain characteristic to six-step operation can be implemented [15]. A feasible solution verifies the following inequalities:

$$\begin{aligned} &F\Delta u(k) \leqslant f \\ &F = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ \sqrt{3} & 0 & -\sqrt{3} & -\sqrt{3} & 0 & \sqrt{3} \end{bmatrix}^T T(\theta_e), \\ &f = \frac{2u_{DC}}{\sqrt{3}} \begin{bmatrix} 1 & 0.5 & 1 & 1 & 0.5 & 1 \end{bmatrix}^T - Fu(k-1) \end{aligned} \tag{7.6}$$

Assuming, as anticipated, that $N = 3$ and $N_u = 1$ (and so it will be hereafter) the condensed problem of (7.5) results in

$$\begin{aligned} &\min_{\Delta u(k)} \ J(\Delta u(k)) := \frac{1}{2}\Delta u(k)^T H \Delta u(k) + c^T \Delta(k) + \text{const}, \\ &\quad \text{s.t.} \quad F\Delta u(k) \leqslant f \end{aligned} \tag{7.7}$$

where $H \in \mathbb{R}^{2\times 2}$, $c \in \mathbb{R}^{2\times 1}$ and where $F\Delta u(k) \leqslant f$ defines a hexagon, which is the rotated and translated version of $\mathbb{U}_{\alpha\beta}$ according to the transformation $T(\theta_e)$ and $u(k-1)$. Clearly the region $F\Delta u(k) \leqslant f$ is composed of six segments lying on six lines that, hereafter, we denote as ℓ_i for $i = 1, ...6$, being ℓ_i associated to the constraint $F(i,:)\Delta u(k) \leqslant f(i)$. Now, let Δu^* be the optimal solution of (7.7), then the applied control input at time k is

$$u^*(k) = u^*(k-1) + \Delta u^*(k).$$

In the $\alpha\beta$-plane the optimal control input is then $u_{\alpha\beta}(k) = T(\theta_e)u(k)$.

We conclude this section by considering a particular scenario that will play an important role in the proposed algorithm, namely the

case where only one constraint is imposed, e.g., the constraint associated with the line ℓ_{i^*}. The optimal control problem to be solved is

$$\min_{\Delta u(k)} \quad J(\Delta u(k)) \qquad \text{s.t. } F(i^*,:)\Delta u(k) \leqslant f(i^*)$$

and the solution is computed by solving the almost equivalent unconstrained problem

$$\min_{\Delta u(k)} \quad J(\Delta u(k)) + \lambda_{i^*}(F(i^*,:)\Delta u - f(i^*))^2 \tag{7.8}$$

for λ_{i^*} sufficiently high to guarantee a negligible distance from the constraint. Basically, in (7.8) the functional cost J has been augmented with a penalty function forcing the solution to stay on the active constraint. Note that the problem in (7.8) admits a simple analytical solution.

7.4 ALGORITHM DESCRIPTION

In this section, the QP solver algorithm is described. First, the initial assumptions are listed, then, the algorithm is discussed, considering different cases that may occur during the search of the solution. The solver is based on the idea of solving a constrained QP problem with the constraints described in (7.6), with a finite succession of iterations, each one being analytical solutions of unconstrained problems. In particular, at the beginning, the optimal solution of the unconstrained version of problem (7.7) is computed in closed form as

$$\Delta u^{uc} = -H^{-1}c, \tag{7.9}$$

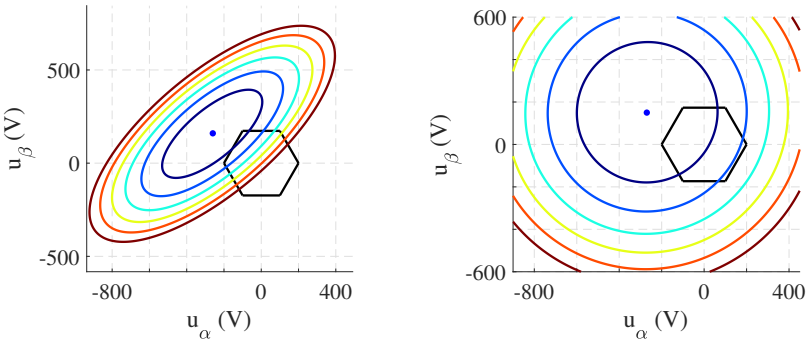
assessing the maximum number of possible active constraints, that depends on the relative position of the hexagon defining the voltage limits with respect to the unconstrained optimal solution. Four different situations can then occur:

0. The solution is optimal and feasible, i.e., the optimal unconstrained solution lies within the feasible voltage set, and it is applied as reference for the inverter;
1. one constraint is not satisfied (see for example Figure 7.2);
2. two consecutive constraints are not satisfied (see Figure 7.3);
3. three consecutive constraints are not satisfied (see, e.g., Figure 7.4).

Notice that, in the figures, the unconstrained solution

$$u^*_{\alpha\beta} = T(\theta_e)\,(u(k-1) + \Delta u^*)$$

and the respective feasible set $\mathbb{U}_{\alpha\beta}$ are represented in the $\alpha\beta$-plane. As mentioned before, a simple roto-translation allows for mapping



(a) $L_d \neq L_q$ (b) $L_d = L_q$

Figure 7.1: Cost function contours in αβ-plane for anisotropic and isotropic motor. The hexagon represents the feasible voltage set, while the center of the contours is the unconstrained solution.

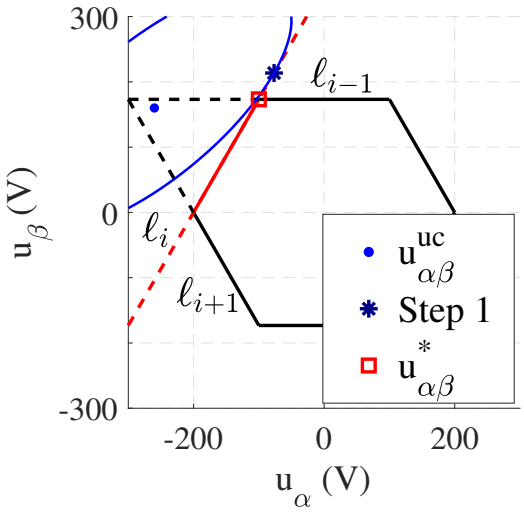
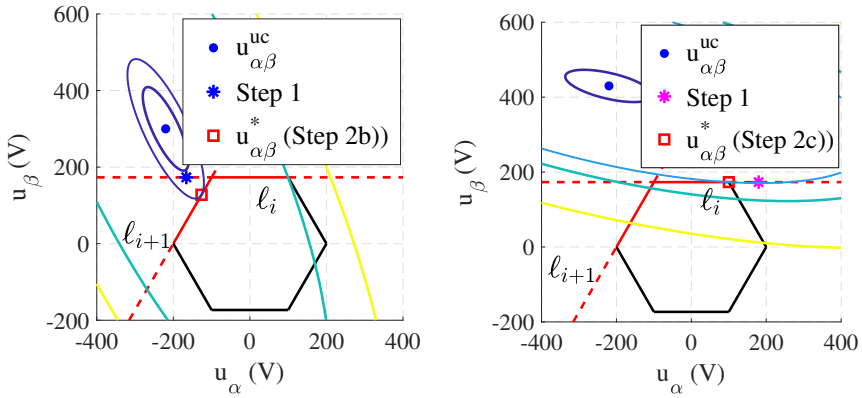


Figure 7.2: One constraint violation: the solution is found by adding the penalty function represented by the red line to the cost.

the dq-plane into the αβ one, making the representations equivalent in qualitative terms. In order to better clarify the last three points, we propose a deepening on the shape of the function cost and the relative positioning of the hexagonal constraint.

7.4.1 *Regions of Violated Constraints*

In this section we expressed the functional cost J in terms of $u_{\alpha\beta}$. It turns out that the level set curves are centered in $u^{uc}_{\alpha\beta}$, where the eccentricity depends on the ratio of the dq inductances and the orientation depends on the electric angle. Figure 7.1 shows two examples of cost function shapes in the αβ-plane, where Figure 7.1(a) represents the case of an Interior Permanent Magnet (IPM) motor, while



(a) Two steps solution.

(b) Vertex solution.

Figure 7.3: Two constraint violations. Figure 7.3(b) represents the case 2c), where the one constraint violation procedure is followed, and the solution is on an intersection.

Figure 7.1(b) shows the case of an SPM motor. In both cases, a representative hexagonal feasible voltage set in the $\alpha\beta$-plane induced by (7.6) is reported. Notice that, due to the magnetic isotropic typical of an Surface Permanent Magnet (SPM), the level curves become circumferences. Consequently, depending on the type of motor, a priori knowledge of the cost function shape can be obtained.

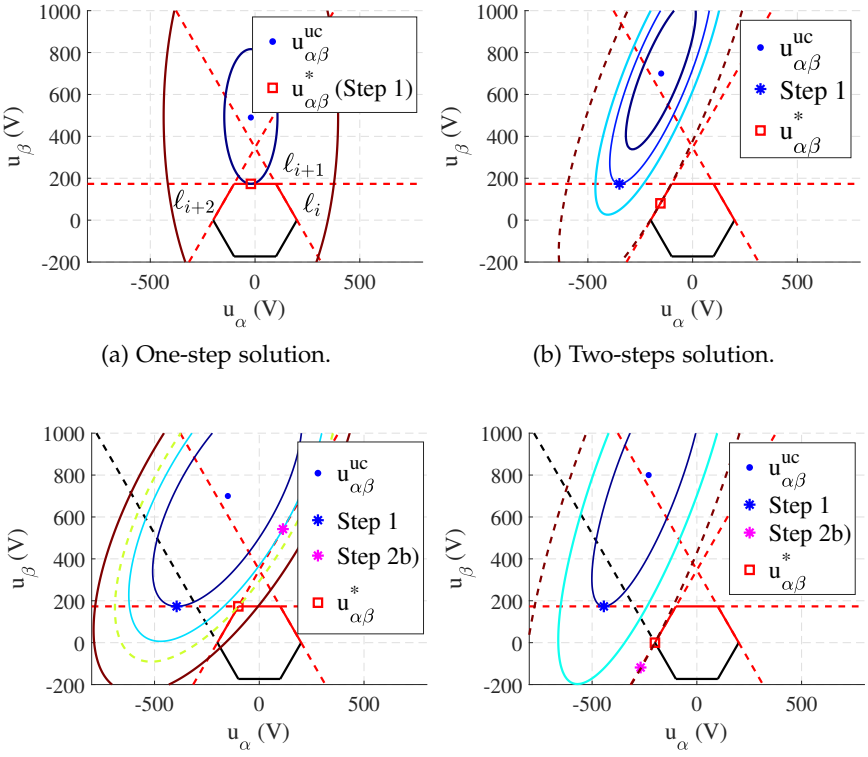
By analyzing the six inequalities associated with the rows of F and f in (7.6) and relying on the convexity of the level curves of the cost function, it is possible to relate the relative position of the unconstrained optimal solution with respect to the hexagonal feasible voltage set with the number of violated constraints. In Figure 7.5, a graphical representation is shown, where each region is identified by a number from 0 to 3, which indicates the number of violated constraints of an optimal solution located in the corresponding region. In the following, for each of these cases, a tailored solution strategy is described.

7.4.2 *Algorithm Steps*

A detailed description of the steps of the algorithm is proposed below, categorized according to the relative location of the unconstrained solution Δu^{uc} with respect to the cost function (see Figure 7.5).

7.4.2.1 *One Constraint Violation*

If Δu^{uc} lies within the triangle adjacent to the segment of the hexagon lying on the line ℓ_i (see Figure 7.5), the feasible optimal solution certainly lies on the segment itself. A representative example is shown in Figure 7.2, where solutions are plotted, as before, in the $\alpha\beta$-plane.



(a) One-step solution.

(b) Two-steps solution.

(c) Vertex solution (first case).

(d) Vertex solution (second case).

Figure 7.4: Three constraint violations.

The feasible solution is then computed according to the following steps:

1. The solution of the problem (7.8) is computed considering as constraint ℓ_i[1], i.e., with $i^* = i$, namely Δu^{oa};

2. a feasibility check is then operated, computing

$$h_j := F(j,:)\Delta u^{oa} - f(j) \leqslant 0 \text{ for } j = i-1, i+1; \tag{7.10}$$

   a) if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$ and lies on the segment;

   b) if Δu^{oa} violates the next constraint[2] ℓ_{i+1}, i.e., if $h_{i+1} > 0$, thus Δu^* is the intersection between ℓ_i with ℓ_{i+1};

   c) if $h_{i-1} > 0$, Δu^* is the intersection between ℓ_i with ℓ_{i-1}.

1 To simplify the description of the algorithm we refer to constraint ℓ_i as the constraint lying on line ℓ_i.

2 The constraints are cycled through with a counterclockwise direction in a circular buffer fashion, i.e., the subscript $i + n =$ mod $(i - 1 + n, 6) + 1$ for any integer n where mod is the modulo operator.

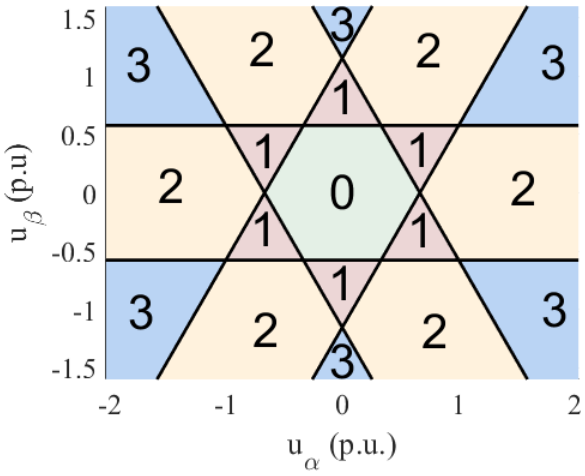


Figure 7.5: Portions of αβ-plane identified by the hexagon lines identified by the number of input constraints that are overcome simultaneously by the unconstrained optimal solution which lies within that region.

7.4.2.2 *Two Constraint Violations*

The case where the unbounded solution lies in a region that crosses two consecutive boundaries of the feasible set is illustrated in Figure 7.3. In this case, Δu^* lies on one of two *consecutive* constraints ℓ_i, ℓ_{i+1}, for some i in the range $[1, 6]$. More precisely, depending on the location of the unconstrained solution and the cost function shape, three different situations can occur:

- the feasible solution lies within one of the hexagon segments that lies either on ℓ_i or on ℓ_{i+1} (see Figure 7.3(a));
- the feasible solution lies on the intersection of ℓ_i and ℓ_{i+1};
- the feasible solution lies on one of the extremal vertexes of one of the hexagon segments, i.e., those that are not originated by the intersection of ℓ_i and ℓ_{i+1} (see Figure 7.3(b)).

The feasible solution is then computed according to the following steps:

1. The solution to the problem (7.8) is computed considering as constraint ℓ_i, i.e., with $i^* = i$, namely Δu^{oa};
2. a feasibility check is then operated, computing h_j as in (7.10) for $j = i+1, i-1$;
   a) if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$;
   b) if $h_{i+1} > 0$, thus Δu^* lies on ℓ_{i+1} itself and the *one constraint violation* procedure can be applied (Figure 7.3(a));
   c) if $h_{i-1} > 0$, Δu^* is intersection between ℓ_i with ℓ_{i-1} (Figure 7.3(b)).

7.4.2.3 *Three Constraints Violations*

The last possible condition is reported in Figure 7.4, where the unconstrained solution detects three violated boundaries in which the optimal feasible solution can lie. With the same notation adopted in the previous section, the three consecutive constraints are denoted by ℓ_i, ℓ_{i+1} and ℓ_{i+2} for some i in the range $[1, 6]$. Three different situations can then occur:

- the feasible solution lies within one of the hexagon segments that lies or on ℓ_i or on ℓ_{i+1} or on ℓ_{i+2};
- the feasible solution lies on the intersections of ℓ_i and ℓ_{i+1} and ℓ_i;
- the feasible solution lies on one of the extremal vertexes of one of the hexagon segments, i.e., those that are not originated by the intersection of ℓ_i and ℓ_{i+1} and ℓ_{i+2}.

The feasible solution is then computed according to the following steps:

1. The solution to the problem (7.8) is computed considering as constraint ℓ_{i+i}, i.e., with $i^* = i + 1$ (the central violated constraint), namely Δu^{oa};
2. A feasibility check is then operated, computing h_j as in (7.10) for $j = i + 2, i$;
   a) if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$ (Figure 7.4(a));
   b) if $h_{i+2} > 0$ (or $h_i > 0$), Δu^* certainly lies on $\ell_i + 2$ (or ℓ_i respectively) and the *one constraint violation* procedure can be applied; Figure 7.4(b), Figure 7.4(c) and Figure 7.4(d) summarized the possible situations.

7.5 COMPUTATIONAL ANALYSIS

In the previous section we described an algorithm to solve the constrained QP (7.5) that exploits the cost function shape to provide a solution in a fixed number of operations. The procedure results to be accurate and efficient in terms of computational burden. The computational performance of the algorithm is hereafter compared with those of the open-source tool `qpOASES`.

7.5.1 *Comparison with `qpOASES`*

In this section, we show that the proposed method achieves competitive performances and requires a low computational effort. `qpOASES`

Table 5: Overview of the SyRM motor and control parameters.

Motor Data	Symbol	Value
Pole pairs	p	2
Phase resistance	R	$1\,\Omega$
d-axis inductance	L_d	$0.2\,H$
q-axis inductance	L_q	$0.06\,H$
Nominal current	I_N	$6\,A_{peak}$
Nominal d current	$I_{N,d}$	$3\,A$
Nominal q current	$I_{N,q}$	$5.2\,A$
Nominal speed	Ω_N	$700\,rpm$
Prediction Horizon	N	3
Control Horizon	N_u	1
Input Weight d-axis	r_d	0.0001
Input Weight q-axis	r_q	0.0002
Prediction Error Weight	q_s	1

Table 6: Averaged time comparison between the proposed method and `qpOASES` for different number of steps.

	Avg Time (µs)	
	Proposed Method	qpOASES
Step 0	1.6	28.2
Step 1	1.65	30.4
Step 1.5	1.67	30.4
Step 2	1.7	30.0
Step 2.5	1.72	30.0

is an open source software, it is very robust and suitable for medium-small scale problems, and it can be considered as a very good benchmark in terms of solution accuracy. The MPC problem is built considering the motor under test (Table 5). The averaged time required by the proposed method is compared with the one of the open source solver `qpOASES` in `MATLAB` environment. The solvers were run on a Intel(R) Core(TM) i7-8700 CPU 3.20GHz and they were tested considering all the possible cases presented in Section 7.4. The problems were run one million times, averaging the time spent for the computation.

The numerical results are presented in Table 6. The following values have been chosen to distinguish among the cases:

- 0 if the unconstrained solution is feasible;

- 1 the algorithm performs the one constraint violation procedure, and the new solution is feasible (step 2 a) in 7.4.2);
- 1.5 if the one constraint violation indicates a solution found in the intersection (step 2 b) or c));
- the value of 2 refers to the case where two of (7.8) are solved.
- finally, 2.5 corresponds to the worst-case scenario where, in the two and three constraint violations, the solutions is on a vertex after solving (7.8) twice.

As can be noticed, the proposed solver finds the MPC solution with cheaper computational cost with respect to `qpOASES`. We highlight that, for the worst-case scenario, where the solution violates three constraints and the optimal one is on a hexagon vertex, the average computation time remains very limited. Thus, the execution time of this case is considered the upper bound for the control algorithm cost.

7.5.2 *Worst-Case Computational Cost*

Considering the relevance of real-time feasibility certification for algorithms that aspire to large-scale industrial applications, we propose a worst-case performance analysis. Since the computational performances strongly rely on the specific implementation, we propose a quantification of the computational cost required by the proposed method with the total number of algebraic operations performed to find the solution in the worst-case scenario. The sequence of operations is listed below:

1. Compute the unconstrained solution; (7.9);
2. feasibility check (7.10) $\Rightarrow$ solution unfeasible ;
3. find new solution (7.8);
4. feasibility check (7.10) ; $\Rightarrow$ solution unfeasible;
5. find new solution (7.8) ;
6. feasibility check (7.10); $\Rightarrow$ solution unfeasible;
7. find the solution among the vertices.

The corresponding number of operations is reported in Table 7. It is worth noting that step 7) does not require any additional algebraic operations. In conclusion, the peculiarities of the proposed method are highlighted: the maximum number of steps are fixed; hence, the total number of operations can be evaluated and the maximum run time can be easily determined depending on the specific hardware.

Table 7: Total number of algebraic operation of the proposed QP method in the worst-case scenario.

PHASE	SUM./SUB.	PRODUCTS	DIVISIONS
1)	7	10	2
2)	15	12	-
3)	15	28	2
4)	15	12	-
5)	15	28	2
6)	15	12	-
7)	-	-	-
TOTAL	82	102	6

7.6 EXPERIMENTAL RESULTS

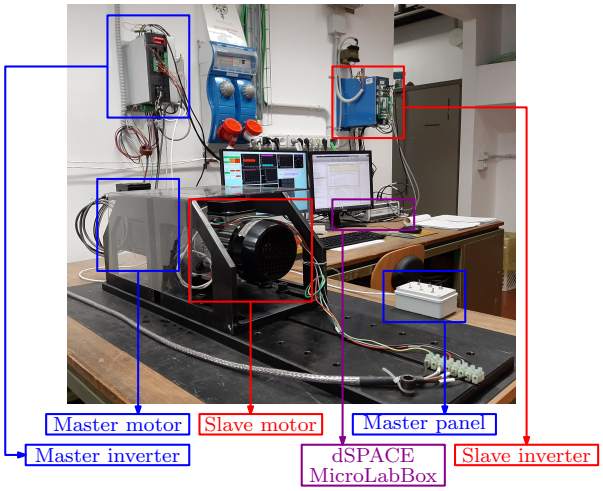
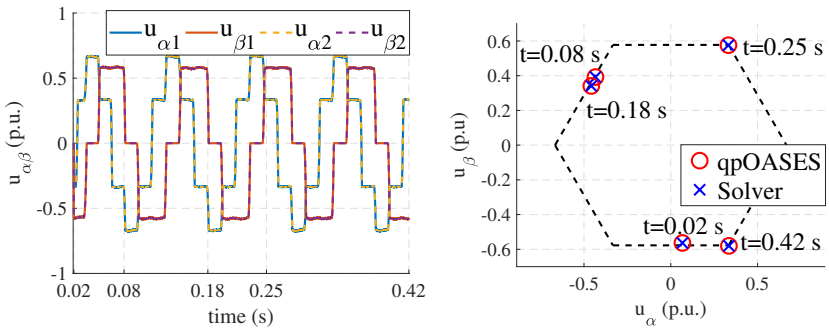


Figure 7.6: Test-bed layout.

The test bench adopted for the experiments is shown in Figure 7.6. It consists of a master motor (surface-mounted PMSM) directly connected to the SyRM motor under test. The controllers have been implemented in a dSPACE MicroLabBox, a compact development system for laboratory purposes which has dual-core real-time processor at 2 GHz and dedicated electric motor control features, which provides the gate signals for the inverter and performs all the acquisitions from current and position sensors.

7.6.1 *Solution Accuracy*

First, a reduced value of the DC bus was set for the controllers in order to achieve an unfeasible working point, where the solvers are



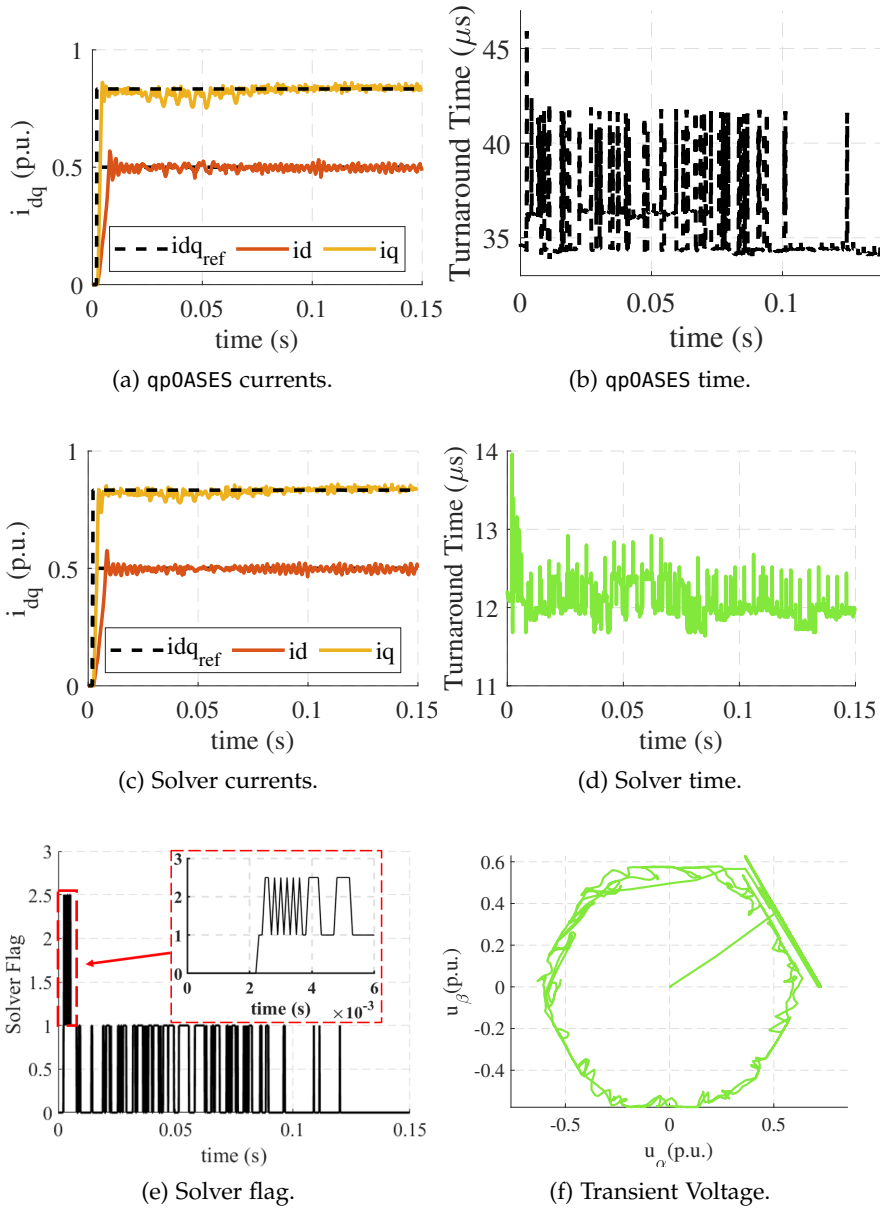
(a) αβ Voltages. (b) Solution Snapshots.

Figure 7.7: Comparison between qpOASES and the proposed method when an unfeasible working point is set. The subscript 1 in Figure 7.7(a) refers to qpOASES, while the other one to the proposed method.

pushed to find solutions along the hexagon edges. During the test qpOASES was used as solver and it is considered as the benchmark. Then, the recorded data were used to build the MPC problem offline and the voltage solutions were computed with the proposed method for each time-step. This experiment is shown Figure 7.7. The voltages are plotted in per unit with respect the average value of the DC bus. The solutions found by qpOASES and our solver overlap, as confirmed by the voltage reference in Figure 7.7(a). In particular, some test points are plotted in Figure 7.7(b) in the αβ-plane for the selected time snapshots indicated in Figure 7.7(a). This confirms that the solutions are found at limits and they are numerically identical.

7.6.2 *Solver Performance*

To evaluate the solver performance in terms of computational cost we propose a current step transient in Figure 7.8, where the master motor keeps the motor under test at its nominal speed, and the nominal current value is set as reference. This condition represents one of the most stressing tests for evaluating current controller performance. Thus, it is expected that this represents the worst-case scenario for the computational cost of the algorithms. Figure 7.8(a) and Figure 7.8(c) show the dq currents transients, normalized with respect to the nominal peak current value. The performance are very similar for the two methods, with good tracking and steady-state accuracy. The most interesting part of this test can be observed in Figure 7.8(b) and Figure 7.8(d), where the turnaround time of the algorithms is measured by dSPACE. The time required by measurements acquisitions and elaborations is about 8 µs. As an interesting result, it can be observed that, while the proposed solver maintains a good accuracy in tracking the currents, the good computational performance presented in Table 7 are replicated in experimental environment. The former shows



(a) qpOASES currents. (b) qpOASES time.

(c) Solver currents. (d) Solver time.

(e) Solver flag. (f) Transient Voltage.

Figure 7.8: Transient test: Comparison between qpOASES and the proposed method at nominal currents and speed. In Figure 7.8(e) the flag values are described in Section 7.5.1 The average time for measurements acquisition and elaboration is about 8 µs.

a peak at the first instant with 14 µs, while the average time stabilizes at 12 µs. In Figure 7.8(e) is also reported a flag value which indicates at which step the proposed solver finds the optimal solution. Thus, in the worst-case situation, the proposed solver needs 14 µs to find the optimal voltages. Finally, the voltage solutions are compared in Figure 7.8(f), which confirms that solutions are found on the limits during this transient.

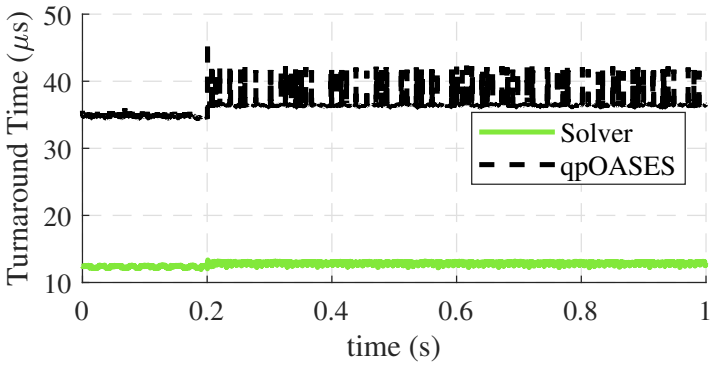


Figure 7.9: Turnaround time of the QP algorithms considering an unfeasible working point.

As a last confirmation of the averaged time required by the solvers (i.e., the proposed algorithm and `qpOASES`) to perform all the computations, we report the turnaround time during the test of Figure 7.7 in Figure 7.9, where an unfeasible working point is considered. We highlight that the time required by `qpOASES` is inevitably higher, because the generality of the considered solver increases the required time to solve the problem.

7.7 DISCUSSION

In this work, a MPC strategy for the current control loop of synchronous motor drives has been presented, including the input voltage feasible set. A fast and effective method for solving in real-time the quadratic programming problem related to the MPC has been proposed. The number of steps in the worst-case is fixed, thus the number of operations can be assessed in advance. One of the main features of the proposed method is that real-time certification can be achieved. Experimental results have been performed by using the open source solver `qpOASES` as benchmark for comparing the optimal solution. The proposed solver has shown very good accuracy and, considering the worst-case scenario, the optimal voltage is found in few microseconds, making it promising for its implementation in large-scale, real-time industrial applications.

Part III

DATA-DRIVEN CONTROL FOR ELECTRIC DRIVES

8

DATA-DRIVEN CONTINUOUS-SET PREDICTIVE CURRENT CONTROL FOR SYNCHRONOUS MOTOR DRIVES

Optimization-based control strategies are an affirmed research topic in the area of electric motors drives. These methods typically rely on an accurate parametric representation of the motor equations. This chapter exploits the concept presented in Chapter 4 with the aim of proposing a potential transition from model-based towards data-driven optimal control strategies. As starting point, the Model Predictive Control (MPC) paradigm is considered, which exploits the parametric model of the motor. Second, the discussion shifts to the Prediction Error Method (PEM), where a state-space model is identified from data. Moving toward data-driven controls, the Subsapce Predictive Control (SPC) is presented, where a reduced model is constructed based on the singular value decomposition of raw data. The final step is represented by a complete data-driven approach, named Data EnablEd Predictive Control (DeePC). The theory behind these techniques is reviewed and design applied for the first time to the design of the current controller of synchronous permanent magnet motor drives. Design guidelines are provided to practitioners for the proposed application and a way to address offset-free tracking is discussed. Experimental results demonstrate the feasibility of the real-time implementation and provide comparisons between model-based and data-driven controls.

8.1 INTRODUCTION

The interest in data analysis is constantly growing, supported by an unprecedented availability of computational power and memory storage, as well as advances in optimization, statistics and machine learning. This leads to an increasing attention towards data-enabled methods in all branches of science and engineering. This revolution has a significant impact on the control engineering too. *Data-driven control* design consists in synthesizing a controller using the data collected on the real system, without defining and identifying a parametric model for the plant [39]. This is in contrast with model-based approaches, which rely on plant modeling and identification procedures. The epitome of this model-based paradigm is arguably the Model Predictive Control (MPC), which has been applied to power electronics control tasks for two decades, reaching an industrial and commercial level [28].

Continuous Control Set (CCS)-MPC methods for Permanent Magnet Synchronous Motor (PMSM) current control, which is the focus of this work, rely on a state-space model of the motor to build the predictive controller [89, 96, 97]. The parameters of this model can be obtained by performing a careful experimental characterization. These procedures often include many different tests and they require specific measuring devices and proper test-bed setups. Then, the resulting accurate model can be exploited in real-time by means of look-up-tables. Alternatively, parameters could be estimated via offline [98] or online [99] procedures. Self-commissioning and auto tuning techniques are also consolidated strategies. In [100], an exhaustive survey of research and state-of-art parameter identification and self-commissioning methods for AC motor drives is discussed. In particular, these approaches are of interest when high performance control is required with sensorless applications. Finally, many methods have been proposed in literature to improve the robustness against parameter variations [66, 75, 101–103], although most of these strategies are implemented for Finite Control Set (FCS)-MPCs.

The key idea behind data-driven predictive controllers is to avoid the model identification stage entirely, and design the controller directly from collected Input/Output (I/O) data, e.g., voltage/current samples. This approach overcomes the challenges of model selection and identification, resulting of particular interest for many industrial applications [104]. However, there are just a few examples of data-driven control applications for electric motor drives. In [105], an observer is coupled to an MPC to update the PMSM model, improving its reliability. However, this approach still relies on a parametric model. Many effective techniques have been presented which go toward the data-driven paradigm, named model-free [106–108] or parameter-free [109] algorithms. In particular, [106] and [107] propose to online update a non parametric model, but they rely on the hypothesis that there are no data available for guessing an initial controller, which might be too restrictive.

In this chapter a transition from model-based to data-driven control design is presented, considering as application the current control of PMSMs. This control task serves as a well-understood benchmark for new methods, despite the fact that other traditional non-data-driven methods yield satisfactory results for this application. The considered framework on optimization-based control schemes, i.e. MPC-type solutions. Firstly, state-of-the-art of CCS-MPC is recalled, whose model is obtained through a previous motor characterization. Then, the discussion moves step by step towards more data-driven control designs, exploiting just voltage and current measurements collected from the motor. The PEM technique coupled with MPC is presented, which is a consolidate solution for identifying a parametric model from data [110]. A further step is represented by the SPC [43], where the

collected data are processed offline by means of a least-square program, and the resulting ARX predictor is de-noised by singular value thresholding. This pseudo-identification procedure is used to build a linear predictor for the currents dynamics. Finally, a completely data-driven control algorithm is presented, named the DeePC [38, 42], where the system identification process is totally avoided and the collected data are directly used in the controller. This technique has already found application in power electronics [40, 111, 112].

The contributions of this work are manifold:

- the perspective of data-driven control design using a predictive control framework is deeply analyzed;
- the practical real-time implementation of data-driven methods is demonstrated, which is not trivial since data-driven methods are expensive in terms of computation and samples;
- it is shown that data-driven paradigm can be a systematic design tool for PMSM current controllers;
- discussion about computational aspects of the presented control strategies is provided;
- as a technical contribution, the problem of the offset-free tracking for the SPC and DeePC methods is addressed;
- guidelines for the choice of the control parameters and excitation input signals for this application are provided.

A relevant advantage of data-driven strategies is that they can be easily implemented as automatic procedures that excite the system with predefined input signals, perform offline calculations, and deliver a ready-to-use control law. No special skills or specialized commissioning personnel are required to set up the procedure. This approach could be interesting for some industrial challenges. For example, in compressor for refrigeration equipment or submersible pumps, offline characterizations cannot be performed when PMSMs are inaccessible. Another case of interest is multi-purpose drives, where algorithms suitable for different PMSM topologies are needed. In addition, PMSM and inverter manufacturers are often different companies and they were never meant to be integrated in the same application. Moreover, if the motor drive needs to be manually re-tuned during its life-cycle, data-driven procedures represent a simple and reliable method to adapt the initial design.

8.2 MODEL-BASED MPC OF PMSM CURRENTS

According to the MPC paradigm, the future control input sequence $u = [u(k), u(k+1), \dots u(k+N-1)]^T$ is optimized in order to steer

the predicted future output $y = [y(k+1), y(k+2), \dots y(k+N)]^T$ to a desired reference $r = [r(k+1), r(k+2), \dots r(k+N)]^T$. Only the first optimal input of the sequence $u(k)$ is applied to the plant (receding horizon principle). Thus, the following optimization problem is solved at each control period:

$$\min_{u,x,y} \quad \left(\|y - r\|_Q^2 + \|u\|_R^2 \right) \tag{8.1a}$$

$$\text{subject to} \quad x(k+1) = Ax(k) + Bu(k), \; y = Cx, \tag{8.1b}$$

$$u(k) \in \mathcal{U}, \; k = 0, \dots, N-1 \tag{8.1c}$$

where N is the prediction horizon, $Q \geqslant 0$ and $R > 0$ are two weighting matrices, A and B represent the state space model used to predict the output $y = Cx$, and $\mathcal{U}$ is the input feasible set. If the the set $\mathcal{U}$ is neglected, the problem is referred to unconstrained, and it has a closed-from solution of reduced computational burden. On the opposite, if the constraints are included, the optimization problem becomes a Quadratic Programming (QP) which requires an online solver like `qpOASES`, as in [95], but it is still easily solvable in real-time.

In the context of PMSM currents control, future currents are estimated by exploiting a parametric model, based on the voltage balance equations, represented in the dq reference frame, synchronous with the rotor flux. The equations are arranged in a state-space form:

$$\begin{gathered} i_{dq}(k+1) = A i_{dq}(k) + B u_{dq}(k) + B h(k) \\ A = \begin{bmatrix} 1 - R_s \dfrac{T_s}{L_d} & \omega_e \dfrac{L_q}{L_d} T_s \\ -\omega_e \dfrac{L_d}{L_q} T_s & 1 - R_s \dfrac{T_s}{L_q} \end{bmatrix}, \; B = \begin{bmatrix} \dfrac{T_s}{L_d} & 0 \\ 0 & \dfrac{T_s}{L_q} \end{bmatrix} \end{gathered} \tag{8.2}$$

where R_s is the stator winding resistance, T_s is the sampling period, ω_e is the electric angular speed and L_d and L_q are the d and q-axis inductances, respectively. Moreover, i_{dq} and u_{dq} are the dq currents and voltages, respectively. u_{dq} are the inputs of the system whereas i_{dq} are the states. Finally, $h = [0 \; -\omega_e \Lambda_{pm}]^T$ is the back-Electro Motive Force (back-EMF) due to permanent magnet flux linkage Λ_{pm}. In the considered application, the full state, i.e. motor currents, is measurable. This model neglects the cross-saturation phenomena, as well as iron-saturation and back-EMF harmonics effects. Thus, the model can result as oversimplified for some PMSM topologies, such as pure reluctance motors. However, many CCS-MPCs proposed in literature work with even more simplified models, obtaining indeed good results. In particular, the dependence of matrix A on the operating speed ω_e is neglected, preferring a constant A matrix for the real-time implementation [35]. It is worth noticing that the data-driven paradigm overcomes the issues coming from assumptions on the model structure.

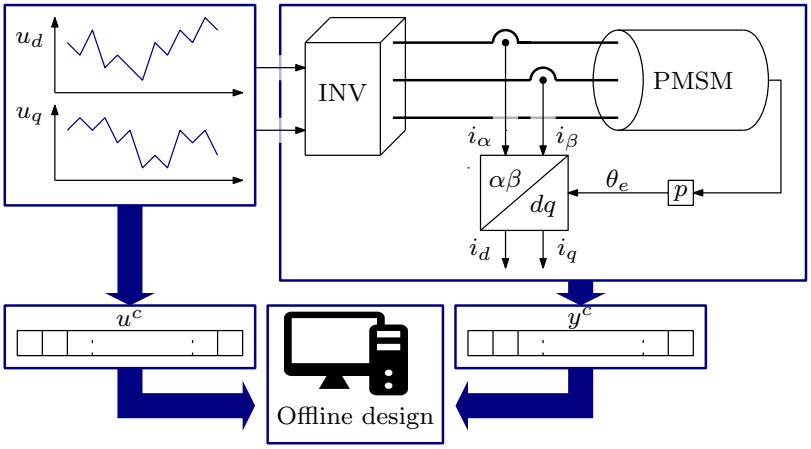


Figure 8.1: Scheme of the data collection process.

An integral action is included in the MPC formulation by means of the velocity form of the MPC problem (8.1) [73], in order to achieve an unbiased current reference tracking (see Chapter 6). The discussion about the offset-free data-driven control is given in Section 8.3.3.

8.3 TOWARDS DATA-DRIVEN CONTROL OF PMSM DRIVES

Concerning the currents control of PMSMs, a data-driven controller is built from the input dq voltages u_{dq} and the measurements of the resulting dq currents i_{dq}. Then, a current reference tracking problem is stated, completely analogous to the one presented in (8.1). In contrast to the parametric model (8.2) used in the MPC solution, a non-parametric model is adopted, consisting of raw measurements arranged in a matrix representation. The construction of this model happens offline, therefore it is not an adaptive controller. A data-driven controller design procedure consists of two steps:

- A *data collection step*, followed by offline rearrangement of the voltages/currents samples into proper matrices;
- An *online program*, when the tracking problem is solved, with the voltages/currents samples matrices acting as a constraint. In this online step, the controller has access to the latest I/O (voltage/current) samples and optimizes the predictions over an horizon of N steps.

8.3.1 *Data Collection and Offline Computations*

All the considered data-driven designs begin from the collection of a T-long sequence of I/O voltages u^c and currents y^c measurements (Figure 8.1). The sequence $u^c = [u_1^c; u_2^c; ...; u_T^c] \in \mathbb{R}^{2T}$ contains the inverter reference voltages and it fulfills the *persistency of excitation*

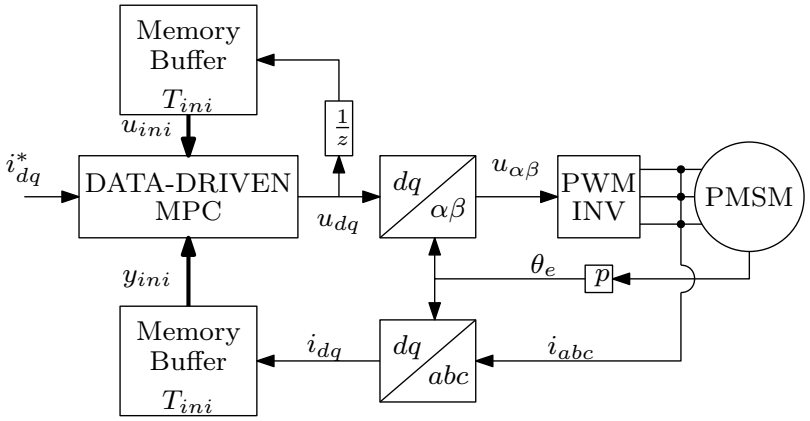


Figure 8.2: Block scheme of the online controller.

Table 8: Overview of matrices dimensions for the considered PMSM current control application.

Matrix	$\mathscr{H}(u_c)$	$\mathscr{H}(y_c)$	U_P	Y_P	U_F	Y_F	P_w	P_u	M	Φ	A	B
rows	$2(T_{ini}+N)$	$2(T_{ini}+N)$	$2T_{ini}$	$2T_{ini}$	2N	2N	2N	2N	L	L	2	2
columns	L	L	L	L	L	L	$4T_{ini}$	$2T_{ini}$	$4T_{ini}$	2N	2	2

requirement [37, Corollary 2], that is the Hankel matrix of inputs in (8.4) needs to have full raw rank. The selection of the input signal is further discussed in Section 8.4.1. The resulting output sequence contains the dq currents $y^c = [y_1^c; y_2^c; \dots; y_T^c] \in \mathbb{R}^{2T}$.

8.3.1.1 *PEM-MPC*

In the PEM-MPC method, the coefficients of the state-space matrices A and B used in (8.1b) are inferred from data by means of an ordinary least-square problem[1] that involves the sequence u_c and y_c:

$$\min_{A,B} \sum_{k=1}^{T-1} \|x_c(k+1) - Ax_c(k) - Bu_c(k)\|^2 \tag{8.3}$$

The main difference between the resulting model and the parametric voltage balance equation (8.2) is that the latter inherently requires the ad-hoc identification procedures to identify all the electric parameters (R_s, L_d, L_q, Λ_{pm}). The PEM method, instead, does not enforce any parametrization of the model and the resulting matrices can, in general, have a structure that is different from the one of (8.2).

8.3.1.2 *SPC*

The theory presented in Chapter 4 for the SPC is here summarized. In the SPC algorithm, the state-space model (8.1b) is replaced by a

1 It is referred to [40] for a discussion on how to solve this problem numerically.

different algebraic constraint that relates the future currents trajectory with the past T_{ini} voltages/currents samples and the future N input voltage samples.

To obtain this model, two Hankel matrices $\mathscr{H}(u_c)$ and $\mathscr{H}(y_c)$ are built using the collected sequences u^c and y^c:

$$\mathscr{H}(u_c) := \begin{bmatrix} u_1 & u_2 & \cdots & u_{T-T_{ini}-N+1} \\ u_2 & u_3 & \cdots & u_{T-T_{ini}-N+2} \\ \vdots & \vdots & & \vdots \\ u_{T_{ini}+N} & u_{T_{ini}+N+1} & \cdots & u_T \end{bmatrix}. \tag{8.4}$$

The output matrix $\mathscr{H}(y_c)$ is built in an analogous way from the samples y_c. Then, the matrices are partitioned in *Past* and *Future* sub-blocks:

$$\begin{bmatrix} U_P \\ U_F \end{bmatrix} := \mathscr{H}(u^c), \qquad \begin{bmatrix} Y_P \\ Y_F \end{bmatrix} := \mathscr{H}(y^c), \tag{8.5}$$

where U_P contains the first T_{ini} block rows of $\mathscr{H}(u^c)$, i.e. $2T_{ini}$ rows, and U_F the remaining N block rows. The block Hankel matrices Y_P and Y_F are similarly obtained. The dimensions of all the presented matrices are summarized in Table 8 for convenience. The I/O block Hankel matrices U_P, U_F, Y_P and Y_F are used in the SPC design to construct an auto-regressive model with exogenous inputs (ARX) as predictor [113]:

$$y = P_w \begin{pmatrix} u_{ini} \\ y_{ini} \end{pmatrix} + P_u u, \tag{8.6}$$

where $u_{ini}, y_{ini} \in \mathbb{R}^{2T_{ini}}$ are the past dq voltage and current samples, respectively, $u, y \in \mathbb{R}^{2N}$ are the future ones. The matrices P_w and P_u are computed as in Chapter 4, Section 4.5. The term multiplied by P_w is used to set the initial condition of the prediction. A Singular Value Decomposition (SVD) of the initial trajectory predictor P_w can be performed to mitigate the noise effect in the data [43]. Only the dominant singular values are used to construct a reduced-rank matrix.

8.3.1.3 *DeePC*

The design of a DeePC controller is purely data-driven, as the data block Hankel matrices defined in (8.5) are used in their raw form in the controller. This method is based on the so called *Fundamental Lemma* of *behavioral system theory* [37] (see Chapter 4, Section 4.2), which guarantees that (under persistency-of-excitation assumptions

on u^c) any trajectory of the system needs to satisfy, for a unique $g \in \mathbb{R}^L$, the linear equations

$$\begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}. \tag{8.7}$$

Implicitly, (8.7) serves as a predictor of the future N-long I/O voltages/currents trajectory (u, y) based on T_{ini}-long I/O initial trajectory (u_{ini}, y_{ini}). If one considers (u, y) as free optimization variables, the vector g that satisfies the first two block-equations of (8.7) can be expressed explicitly as

$$g = \begin{bmatrix} U_P \\ Y_P \end{bmatrix}^{\dagger} \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \Phi z = M \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \Phi z, \tag{8.8}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse operator, and Φ represents a basis of the kernel of M. Both Φ and M can be computed offline using standard linear algebra routines. This decomposition allows expressing the future trajectory as a function of the lower-dimensional variable z, and turns out to be useful in the online phase of the unconstrained control problem, as explained in the next subsection.

8.3.2 *Computational Aspects Regarding the Online Program*

In the online stage, the MPC tracking problem (8.1) is solved, but with different representations in place of (8.1b) depending on the adopted data-driven method. Both the unconstrained and constrained solutions are now discussed for each data-driven method, clarifying the practicality of their real-time implementation from the computational burden point of view.

8.3.2.1 *PEM-MPC*

PEM-MPC algorithm is completely analogous to a standard model-based MPC, from the point of view of the online program. It is worth remembering that two possible online controllers can be obtained, depending on the presence or not of the constraints (8.1c). If the problem is unconstrained ((8.1c) is absent), the PEM-MPC yields a linear feedback controller [96] of the form $u = K^r r + K^x x(k)$. On the other hand, the QP problem requires an iterative solver as in [95], if input constraints are included. In both situations, the complexity of the PEM-MPC is the same of a standard model-based MPC, which is amenable for real-time implementation on adequate hardware. The dimension of the decision variable coincides with the one of $u \in \mathbb{R}^{2N}$,

thus it scales linearly with the prediction horizon. In the considered application, the full state of the system is available, but in general the PEM-MPC requires a state estimator. The other two data-driven methods, SPC and DeePC, do not require a state estimator, since they naturally work with the plant outputs.

8.3.2.2 *SPC*

The SPC algorithm solves the same tracking problem (8.1) as in MPC or PEM-MPC, but with the state-space model (8.1b) replaced by the predictor (8.6):

$$\min_{u,y} \quad \|y - r\|_Q^2 + \|u\|_R^2 \tag{8.9a}$$

$$\text{subject to} \quad y = P_w \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + P_u u \tag{8.9b}$$

$$u(k) \in \mathcal{U},\ k = 0, \dots, N-1 \tag{8.9c}$$

Similarly to the PEM-MPC, if the constraints (8.9c) are not present, then the problem can be solved in closed-form by substituting the predictor equation (8.9b) into (8.9a) and by setting the gradient of the resulting convex quadratic cost to zero. The resulting online controller is a linear feedback of the form $u = K^r r + K^{ini}[u_{ini}, y_{ini}]^T$. If the constraints (8.9c) are present, the minimization program can be solved online, at the same computational complexity of the PEM-MPC one. In fact, the computational burden depends on the length of u.

8.3.2.3 *DeePC*

The DeePC algorithm, because of the implicit form of the algebraic constraint, requires the minimization over the decision variables g, u, y:

$$\min_{g,u,y} \quad \|y - r\|_Q^2 + \|u\|_R^2 + \lambda_g \|g\|^2 \tag{8.10a}$$

$$\text{s.t.} \begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix},\ u(k) \in \mathcal{U},\ k = 0, \dots, N-1 \tag{8.10b}$$

where λ_g adds a regularization on the decision variable g. In fact, if noisy data are used, the Hankel matrices are full raw rank, but the realized control error in (8.10a) could be different from the predicted one. Thus, the term $\lambda_g \|g\|^2$ helps to robustify the control problem [114, Section III.C]. In the unconstrained case, the problem can be solved directly using the null-space representation presented in (8.8).

The future currents and voltages sequences u and y are replaced in (8.10a) with $U_F g$ and $Y_F g$, respectively, obtaining

$$\min_z \|Y_F(M \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \Phi z) - r\|_Q^2 + \|U_F(M \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \Phi z)\|_R^2 \quad (8.11)$$

The solution of the problem is available in closed form as $z^{opt} = H^{-1} d^T$, where the Hessian matrix H and the linear term d are defined as:

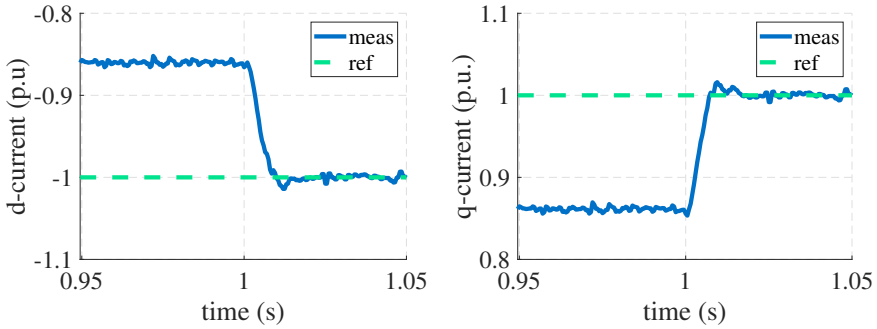
$$\begin{aligned} H &:= \Phi^T Y_F^T Q Y_F \Phi + \Phi^T U_F^T R U_F \Phi \\ d &:= \left(r - Y_F M \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix}\right)^T Q \Phi - \left(U_F M \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix}\right)^T R \Phi. \end{aligned} \quad (8.12)$$

The Hessian inversion can be evaluated offline with proper numerical techniques, further reducing the complexity of the scheme. More details on the closed-form solution of the unconstrained DeePC can be found in [40]. Starting from the optimal value of z^{opt}, (8.8) is used to compute g^{opt}, and, finally, the sequence of optimal input u^{opt}. It is still possible to condense this controller in a feedback law similar to the SPC, with a decision variable that scales linearly with the prediction horizon length. The constrained solution of (8.11) would instead require an online QP solver. However, the dimension of the decision variable g can be large, as it depends on the number of samples used in (8.7). Thus, the real-time implementation of the DeePC algorithm is still a challenging problem.

In conclusion, three main aspects differentiate the SPC and DeePC methods [114]: the way the predictor is built, the underlying prediction model and the variables over which the QP problem is solved. In fact, the SPC forces a least-square fit to a Linear Time Invariant (LTI) system model, whereas the DeePC does not.DeePC Thus, SPC is more suited for LTI systems or linear parameter varying ones. On the other hand, DeePC exhibits interesting features also when applied to nonlinear system, e.g. the grid connected inverter application shown in [40]. Finally, SPC solves the tracking problem in the input u, whereas the DeePC in g.

8.3.3 *Integral Action*

An integral action is needed to avoid bias errors in the currents reference tracking for the SPC and DeePC algorithms. For instance, the back-EMF induced by the magnets acts as a constant disturbance in the voltage equation, inducing a steady state error in reference tracking. Following the idea detailed in Chapter 6, the introduction of this framework also for data-driven controllers is provided. For the MPC and PEM-MPC algorithms, the integral can be included by formulating the optimization problem in its velocity-form [73]. The idea is to



(a) d-current. (b) q-current.

Figure 8.3: Offset-free tracking error: simulation of DeePC algorithm with no integral action (before $t = 1s$) and when the offset-free implementation after $t = 1s$.

perform the data collection stage filling the matrices with incremental data, e.g. $\Delta y = y(k) - y(k-1)$. For instance, the DeePC problem in (8.10) is written as follows:

$$\begin{aligned} \min_{g,\Delta u,\Delta y} \quad & \|\Delta y - r'\|_Q^2 + \|\Delta u\|_R^2 + \lambda_g \|g\|^2 \\ \text{subject to} \quad & \begin{bmatrix} U'_P \\ Y'_P \\ U'_F \\ Y'_F \end{bmatrix} g = \begin{bmatrix} \Delta u_{ini} \\ \Delta y_{ini} \\ \Delta u \\ \Delta y \end{bmatrix} \\ & u(k) = u(k-1) + \Delta u(k) \in \mathcal{U}, \ k = 0, 1..., N-1 \\ & r(k)' = r(k) - y(k), \ k = 1, ..., N \end{aligned} \tag{8.13}$$

$[U'_P Y'_P U'_F Y'_F]^T$ are the Hankel matrices filled with incremental data. The optimization problem (8.13) is solved for g, then $\Delta u(k)$ is found.

The effectiveness of the proposed solution is shown in Figure 8.3. The nominal current reference has been set and steady state is reached, while the motor is kept at nominal speed. Before time $t = 1s$, the standard data-driven formulation is considered as controller. As can be seen, a bias appears in the tracking. At time $t = 1s$, the controller designed with incremental data is selected and the bias is removed.

8.4 EXPERIMENTAL VALIDATION

The experimental validation on an interior permanent magnet motor is proposed. The nominal parameters of the considered machine are reported in Table 9. All the algorithms are real-time implemented on the dSPACE MicroLabBox at a sample rate of $T_s = 100\mu s$.

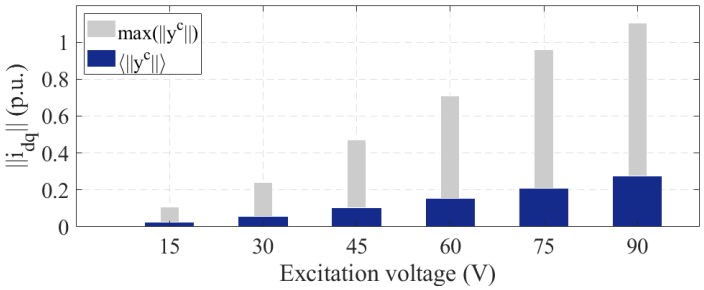
Table 9: Overview of the motor parameters.

Motor Data	Symbol	IPM
Pole pairs	p	3
Phase resistance	R	$1\,\Omega$
d-axis inductance	L_d	$0.010\,H$
q-axis inductance	L_q	$0.014\,H$
PM flux-linkage	Λ_{pm}	$0.26\,V\,s$
Nominal current	I_N	$6.2\,A_{rms}$
Nominal d current	$I_{N,d}$	$-1.1\,A$
Nominal q current	$I_{N,q}$	$8.7\,A$
Nominal speed	Ω_N	$1000\,rpm$

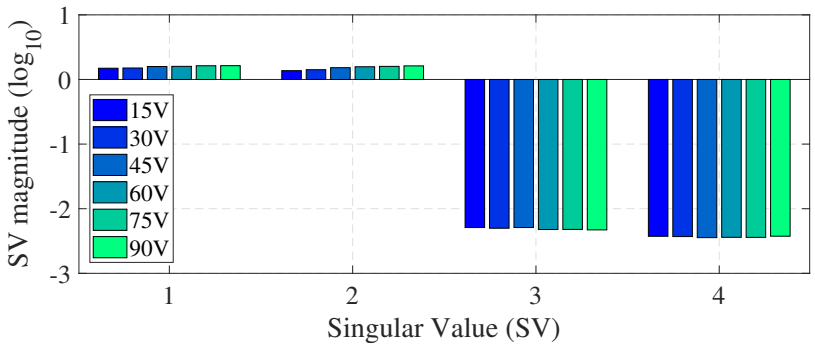
8.4.1 *Data Acquisition Step*

The test designed to collect I/O data from the Interior Permanent Magnet (IPM) motor consists of excitation with a random (detailed below) dq voltage vectors sequence u^c and the measurement of the dq currents. Thanks to this choice, the rotor is not required to be locked or to be maintained at standstill by another motor. The selected zero-mean voltage sequence induces zero-mean currents and, consequently, a zero-mean torque. Since the mechanical dynamic is much slower than the electric one, the rotor remain at standstill even if instantaneously the torque could be not zero. In addition non-linear frictions help to avoid rotations of the motor.

The criteria to select the voltages amplitude is here discussed. The motor is driven by a two-level voltage source inverter with a DC bus voltage of 300 V. The voltage sequence is generated by picking the values from a uniform probability distribution in the interval $[-u_{exc}, u_{exc}]$. A test is proposed to analyze the effects of u_{exc} on the sequence y^c and the data-driven design. Figure 8.4(a) refers to several excitation tests, characterized by different values of u_{exc}. On one hand, the maximum excitation voltage should be limited to avoid over-currents, preserving a safe motor operation. The figure, in fact, shows that the mean value of the currents samples are quite low with respect to the nominal value. However, the nominal current value, for the proposed motor, is achieved using $u_{exc} = 90\,V$, i.e. the 30% of the DC bus voltage. Higher excitation voltages should be avoided. On the other hand, a too low voltage excitation could lead at least to current sampling issues. Moreover, one needs to take into consideration also other problems, i.e. if the information carried by the data is rich enough to describe the current dynamics. The Pulse-Width-Modulation (PWM) synthesis of low voltages could emphasize some



(a) Output currents during the excitation tests for different u_{exc} values: mean and maximum output currents values.



(b) Singular values analysis of P_w for different u_{exc} values.

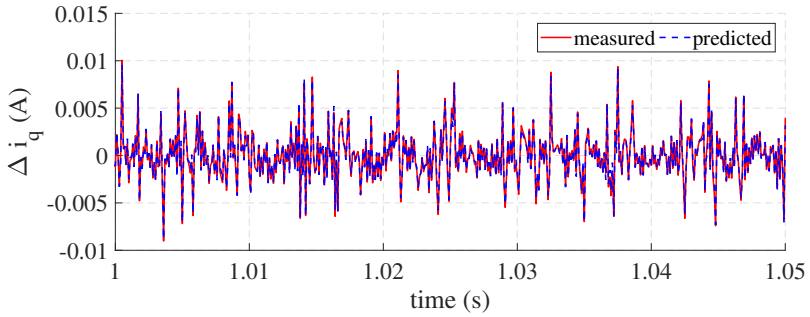
Figure 8.4: Overview of some key parameters of the data collection test.

inverter non-linearities, e.g. not properly compensated dead-times, that are not of interest of our identification. In order to evaluate if the data are collected properly, the dominant singular values of the matrix P_w are analyzed (see the logarithmic plot in Figure 8.4(b)). The number of dominant values should be coherent with the anticipated dimension of the state, see Table 8. Two dominant values characterize the considered dynamic, as expected.

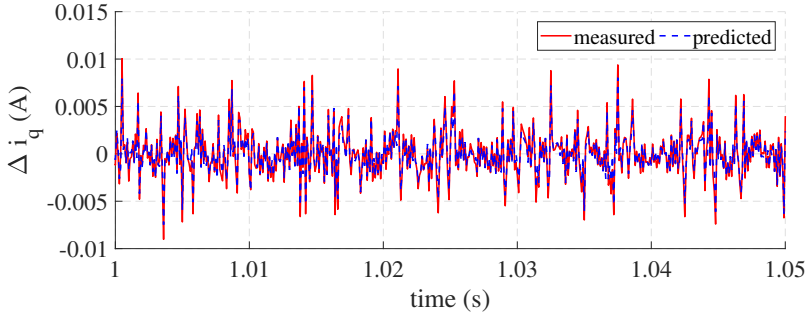
8.4.2 *Parameters Selection*

In this section, the problem of parameters selection for designing the data-driven controls is addressed. The prediction horizon length N is chosen according to the MPC framework, i.e. $N = 3$. This value is a good trade-off between accuracy and computational effort for this application [35]. Moreover, all the these controllers share the same cost function; thus, equal weighting matrices Q and R are chosen. In particular, Q is the identity matrix, whereas R is the identity scaled by a factor 0.0001. The robust formulation of the DeePC is considered, and the related parameter in (8.10) has been set to $\lambda_g = 0.1$.

Two parameters that characterize the data-driven algorithms are the length of the initial trajectory T_{ini} and the number of samples T. The trajectory $[u_{ini}, y_{ini}]^T$ replaces the initial condition for the prediction. Thus, it determines the inherent system state, and the parameter



(a) PEM-MPC.



(b) DeePC.

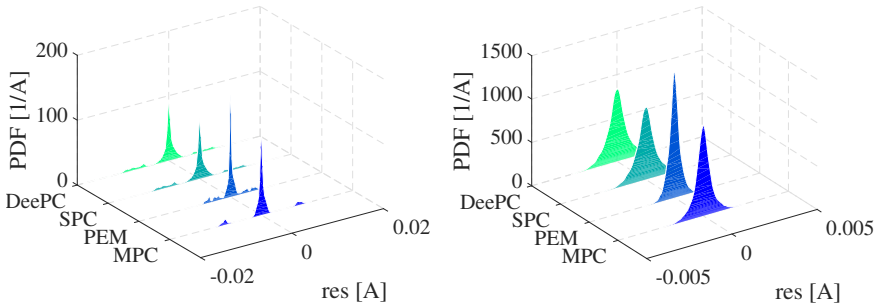
Figure 8.5: Accuracy of the data-driven predictors in the estimation of the q-axis current variation.

T_{ini} provides a complexity for the model. In [37], the system lag[2] l is used to find a lower bound for T_{ini}. In particular, if $T_{ini} \geqslant l$ the system prediction is uniquely determined. Thank to this criteria, the value of T_{ini} can be chosen even without knowing the system dimension, but using an estimate of it. Since the system lag is known for the considered application (i.e $l = 1$), it is set $T_{ini} = 1$. The length T of the recorded I/O vectors should be long enough to make sure that the Hankel matrices have full rank. The *Fundamental Lemma* in [37] gives a lower bound for T, whose value for the considered application is $T \geqslant 3(T_{ini} + N + 2) - 1$. The parameters T is set to 100 samples, which satisfies the inequality. All dimensions of the matrices can be computed using Table 8.

8.4.3 *Accuracy of the Data-Driven Predictor*

The accuracy of the data-driven predictors is investigated in this subsection, taking the model-based MPC as benchmark. This analysis is performed during steady state operation, when the motor is working at the nominal Maximum Torque per Ampere (MTPA) current point (see Table 9) at standstill. During the tests the currents are regulated by standard Proportional-Integral-Derivative (PID) controllers.

2 The lag l of a linear system is the smallest integer value for which the observability matrix $\mathcal{O} = [C\ CA ... CA^{l-1}]^T$ has full rank.



(a) d-axis current. (b) q-axis current.

Figure 8.6: Residual analysis of the prediction error: probability distribution function of the residuals for the presented predictors.

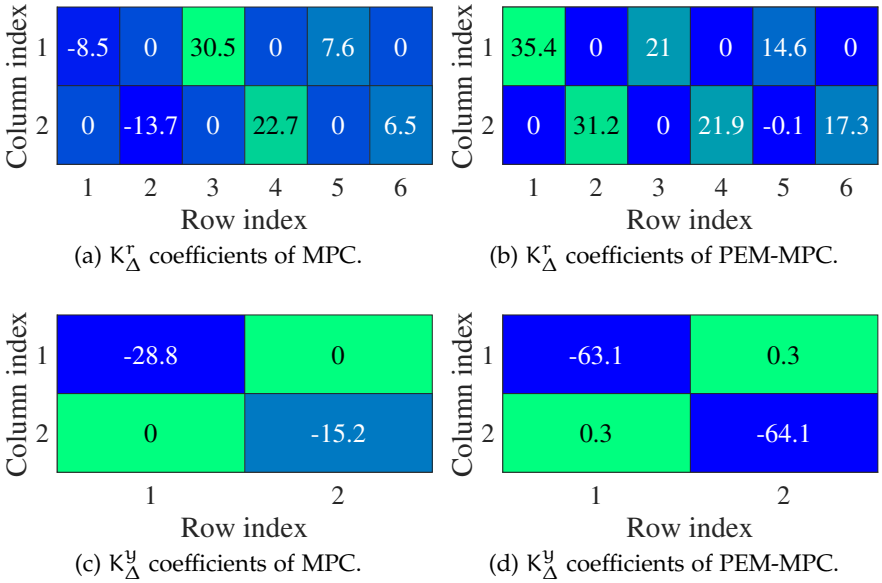
The interest is on the open-loop prediction accuracy of the methods. This means that the predictors are fed by current measurements and the reference voltages computed by the PIDs. A first qualitative information on the accuracy is provided by Figure 8.5(a) and Figure 8.5(b). The figures show the comparison between the measured q-axis current increments and the predicted ones, using respectively the predictor obtained with (8.3) and (8.11). A good correspondence between measurements and predictions for both the controllers is observed.

The residuals between measured and estimated currents are considered as performance index, as suggested in [105]. The results of this analysis are reported in Figure 8.6. The figures show the estimated probability density function of the d and q residuals for all the described predictors. From literature [105], a zero mean normal distribution of the residuals is expected, which is coherent with the obtained results. The PEM-MPC predictor appears the most accurate one, proving that using data to validate the commissioning tests is an interesting tool. This could be observed also by the time-domain figure Figure 8.5(a), since the predicted currents almost overlap the measured ones.

8.4.4 *Online Unconstrained Controller*

In this a comparison between model-based and data-driven designed controllers in terms of step current reference response is provided. In particular, the reference r is changed from zero to the nominal MTPA. The model-based MPC adopts the motor parameters which were previously obtained by means of characterization procedures (see Table 9). All the data-driven controllers are designed from the same data recording, in particular the one defined by a $u_{exc} = 50\,V$.

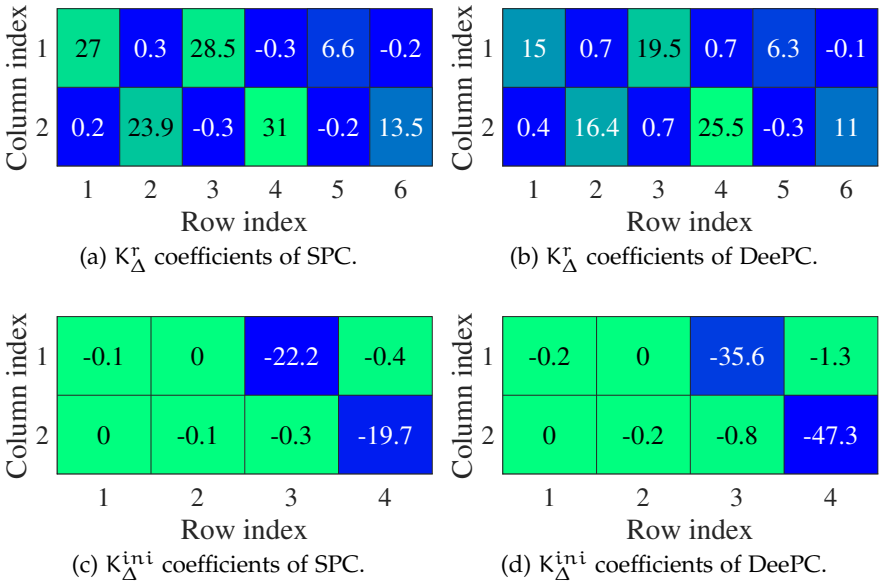
The online controllers are implemented in their unconstrained versions, to provide a fair comparison between all the methods. In fact, the constrained version of the DeePC results too computationally demanding. The resulting control laws are reported in Figure 8.7 and



(a) K_Δ^r coefficients of MPC. (b) K_Δ^r coefficients of PEM-MPC.

(c) K_Δ^y coefficients of MPC. (d) K_Δ^y coefficients of PEM-MPC.

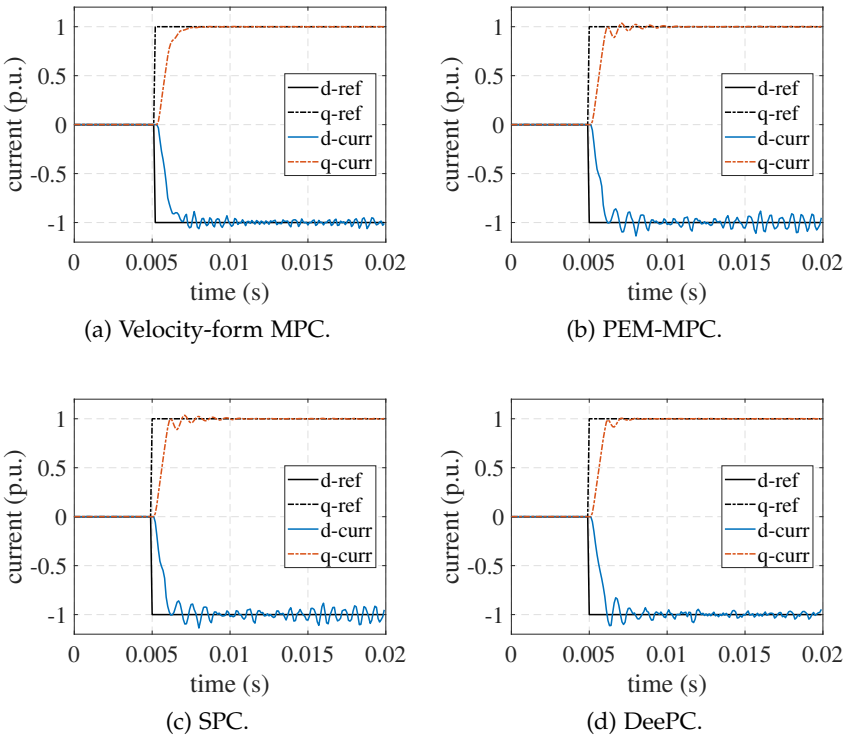
Figure 8.7: Values of K_Δ^r and K_Δ^y elements, representing the feedback laws obtained for the MPC and the PEM-MPC with integral action.



(a) K_Δ^r coefficients of SPC. (b) K_Δ^r coefficients of DeePC.

(c) K_Δ^{ini} coefficients of SPC. (d) K_Δ^{ini} coefficients of DeePC.

Figure 8.8: Values of K_Δ^r and K_Δ^{ini} elements, representing the feedback laws obtained for the the SPC and the DeePC with integral action.



(a) Velocity-form MPC.

(b) PEM-MPC.

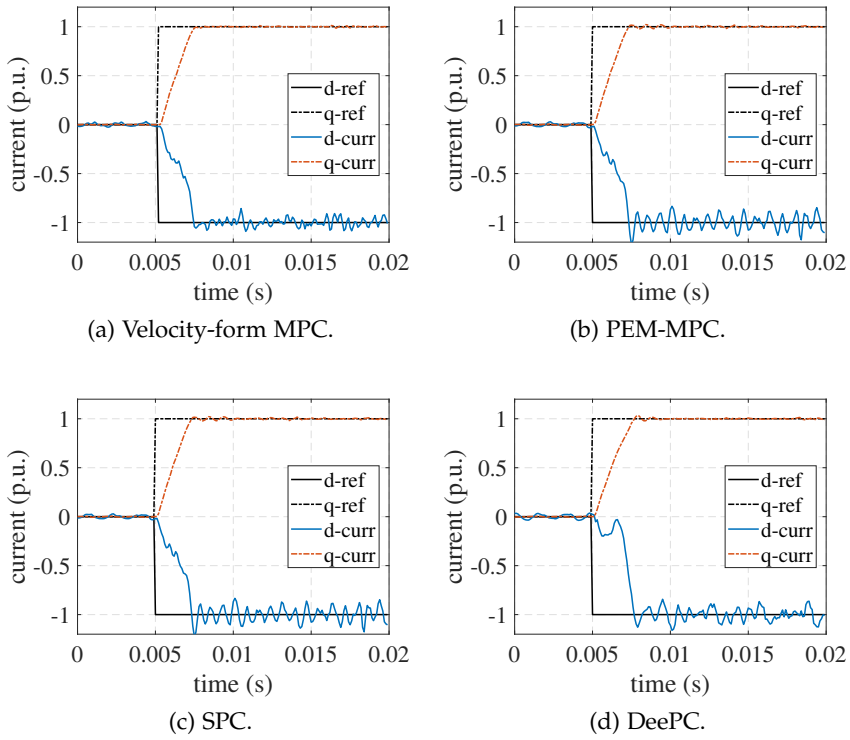
(c) SPC.

(d) DeePC.

Figure 8.9: Comparison of the step responses of model-based and data-driven current controllers at standstill.

Figure 8.8. Namely, the MPC matrices are compared with the ones obtained using the PEM-MPC in Figure 8.7, while the SPC and DeePC control laws are compared in Figure 8.8. The colored heatmaps help to compare the magnitude of the coefficients. Although the coefficient values differ among methods, their distribution is very similar, for example in the location of the zeros. This means that the resulting control action based on data could also take into account the cross coupling terms. In particular, by observing Figure 8.7, the MPC and PEM-MPC control laws are slightly different. Instead, the SPC and DeePC control laws in Figure 8.7 are very similar, thus comparable performances are expected.

The step responses are compared at standstill in Figure 8.9. It is relevant that the data-driven designs allow achieving similar performances with respect to the model-based controller. In fact, the commissioning effort of all the proposed algorithms in terms of measurement apparatus, number of carried out tests and their complexity and duration is much lighter compared to the characterization required to build an accurate model-based controller [100]. Among data driven controllers, the DeePC is considered the most data-oriented algorithm, because it uses raw data without any pre-processing. Despite the direct exploitation of raw data, it has almost the same performance as the others. The same step response analysis also at nominal



(a) Velocity-form MPC.

(b) PEM-MPC.

(c) SPC.

(d) DeePC.

Figure 8.10: Comparison of the step responses of model-based and data-driven current controllers at nominal speed.

speed rate in Figure 8.10 is reported. This test confirms the effectiveness of the integral action included in the data-driven control framework. The back-EMF and the state transition matrix A of the IPM motor model (8.2) depend on the operating speed. Thus, a bias in the current tracking should be observed if the integral action is missing (as in Figure 8.3). The model-based MPC grants the overall best performances, as expected, since the effort paid for the commissioning. However, wit is highlighted that the proposed data-driven methods are still very effective for the q-axis current. Moreover, other tools can be used to further improve their behavior, in particular the one of the DeePC (see [38]). In addition, accordingly to [115], a feed-forward term can be nested in the controller to improve disturbance rejection performances, without penalizing the overshoot in the dynamics. It is therefore believed that there is much unexplored potential to improve the performance to data-driven controllers.

8.5 DISCUSSION

In this work a transition path from model-based to data-driven design of PMSM current controllers is presented. Different data-driven algorithms are considered: the prediction error method model predictive control, the subspace and the data-enabled predictive controls.

All the algorithms were online implemented in the unconstrained version, proving their online feasibility. Similar accuracy between model-based and data-driven predictors is demonstrated with experimental data. Experimental results show that all these controllers have comparable performance, considering the MPC with an accurate model as benchmark. Moreover, among data-driven controllers, the DeePC performs well both in steady-state and dynamics.

There are several challenges to address in the future. First, comparison between data-driven and self-commissioning techniques would be valuable. This could help to design effective strategies for the excitation voltage signals. Second, the extension of data-driven methods for nonlinear system is at the beginning. The possibility to automatically include the motor and inverter nonlinearities in the control law is of particular interest. Third, finding computationally efficient methods for implementing high-dimensional data-driven methods that include constraints in real-time is still an open challenge. Finally, other future research will focus on online adaptation of data-driven control structures and applications to other drives problems.

9

REDUCING DATA-DRIVEN CONTROL ALGORITHM COMPLEXITY VIA MODEL ORDER REDUCTION

This chapter presents some preliminary results that has been obtained during the last period as Ph.D. student. A possible approach for reducing the computational cost in the online stage of data-driven control discussed in previous chapters is presented.

9.1 INTRODUCTION

The aim of this chapter is to propose a method for enabling the real-time implementation of the Data EnablEd Predictive Control (DeePC) control algorithm, exploiting the Quadratic Programming (QP) solver formulation presented in Chapter 7. In the previous chapter, it was shown that, as the number of data used to build a data-driven model of the electric drive increases, the computational cost of the DeePC grows such that real-time feasibility is compromised. The problem class considered has the objective function that is quadratic in the unknowns, whose values must be delimited with respect to some constraints, and the size of the problem is large. The key idea is to exploit Model Order Reduction (MOR) techniques presented in Chapter 5, approximating the large-scale system with dimension n with a reduced order problem with order r. The reduced system is designed in order to preserves a prescribed accuracy of the solution with respect to the original problem. The main advantage of this strategy is that the huge amount of computing cost is offline, and the online cost is dramatically reduced. The steps from data collection to the reduced problem can be fully automated, and the introduction of the MOR procedure reduces the computational complexity for the optimization problem solver to solve in real time. This method is called Data-Driven QP MOR, and it is presented for electric drives applications.

9.2 PROBLEM STATEMENT

The problem of designing a data-driven controller for the current control loop of Permanent Magnet Synchronous Motor (PMSM)s is analyzed. Data-driven controls have recently become popular among the control research community, thanks to the digital revolution that is bringing a huge change in all scientific fields. The availability of data has raised the big question of how to use them to increase the efficiency and safety of industrial processes. In electric drives applica-

tions, the design of a data-driven controller aims at controlling motor currents simply by using the system's input and output data. This procedure requires two main steps. The first one is the acquisition of a certain number of sampled input (voltages) and output (currents) data. Next, this data replaces the motor model in a Model Predictive Control (MPC) framework. No identification step is preformed and the optimization problem is usually a QP problem that needs to be solved in real-time.

The following sections briefly recall the key ingredients of the previous chapters which are combined together for building the proposed algorithm. These features are listed below:

- the DeePC algorithm, whose related constrained QP problem dimension strictly depends on the number of used data;
- the custom QP solver proposed for MPC of electric drives;
- the adopted MOR technique, which represents the link between the data-driven algorithm and the QP solver.

9.3 DEEPC ALGORITHM FOR ELECTRIC DRIVES

The DeePC algorithm minimizes the following cost function with respect to the decision variable g:

$$\begin{aligned} &\min_{g,u,y} \sum_{k=0}^{N-1} \left(\|y(k) - r(k)\|_Q^2 + \|u(k)\|_R^2 \right) \\ &\text{s.t.} \begin{pmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{pmatrix} g = \begin{pmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{pmatrix}, \; u_k \in \mathcal{U}, \; k = 0, \dots, N-1 \end{aligned} \tag{9.1}$$

where U_P, Y_P, U_F, Y_F are Hankel matrices that contain the sampled data and they have the following structure:

$$\mathscr{H}(u) := \begin{pmatrix} u_1 & u_2 & \cdots & u_{T-L_{Hank}+1} \\ u_2 & u_3 & \cdots & u_{T-L_{Hank}+2} \\ \vdots & \vdots & & \vdots \\ u_{L_{Hank}} & u_{L_{Hank}+1} & \cdots & u_T \end{pmatrix} \tag{9.2}$$

These matrices are obtained by partitioning the matrix (9.2) in past and future sub-blocks:

$$\begin{pmatrix} U_P \\ U_F \end{pmatrix} := \mathscr{H}(u^c), \quad \begin{pmatrix} Y_P \\ Y_F \end{pmatrix} := \mathscr{H}(y^c) \tag{9.3}$$

where U_P contains the first T_{ini} block rows of $\mathscr{H}(u^c)$, and U_F the remaining N block rows. The block Hankel matrices Y_P and Y_F are similarly obtained. The design parameters are:

- N, which is the prediction horizon;
- T_{ini}, which is the number of past samples to reconstruct initial trajectory;
- T, representing the total number of samples.

The QP problem (9.1) includes inequality constraints for the voltage solution vector and a solver is required. The dimension of the matrices strictly depends on the choice of these parameters. One of the main drawbacks of this method is represented by the presence of noise in output measurements. To get rid of part of the noise effects, the number of required samples drastically increases. Thus, the real-time implementation of this strategy may be compromised.

9.4 MODEL ORDER REDUCTION

Some insightful about MOR strategies has been provided in Chapter 5. The main idea is that a high-dimensional state vector actually belongs to a low-dimensional subspace. Given a general system of equations $Ax = b$, with unknown vector $x \in \mathbb{R}^n$, the aim is to find a projection matrix $V \in \mathbb{R}^{n\times r}$ such that $x \approx Vx_r$, with $x_r \in \mathbb{R}^r$ reduced order solution vector.

9.4.1 *Proper Orthogonal Decomposition*

Proper Orthogonal Decomposition (POD) is a suitable strategy for approximating large-scale dynamical system with a reduced order problem. This method consists in capturing dominant information of the physical phenomenon through already calculated solutions, and the construction from these of a small subspace with an orthonormal base. Here, the idea is to exploit this method not for obtaining a reduced data-driven model, but to directly reduce the optimization problem. From (9.1), it can be observed that the optimization problem depends on a number of parameters (e.g., current reference, initial conditions). Hence, the reduced model should represent this parameter dependency of the original system. The class of MOR algorithms which aims at solving this problem are called Parametric MOR (pMOR). Parametric model order reduction targets the broad class of problems for which the equations governing the system behavior depend on a set of parameters. A general procedure to obtain a parametric reduced problem with POD follows these steps:

1. define the set of parameters and their range of variability;

2. solve the problem with an initial value of the parameters;
3. orthonormalize the solution vector v (using, e.g., Singular Value Decomposition (SVD), QR or Modified-Gram–Schmidt algorithm) with respect to previous solutions;
4. add the vector v to the projection matrix V;
5. compute the reduced order model using V;
6. test the accuracy of the solution with a number of random parameter values (greedy methods);
7. choose the combination of parameters which give the maximum error;
8. repeat the procedure from step 3 to 7 until the error is less than a desired tolerance.

9.5 IMPLEMENTATION

This section presents a step-by step procedure to combine a quadratic programming problem with model order reduction techniques for improving the real-time feasibility of the data-driven control algorithm. The first step is to consider the unconstrained QP problem. In this framework, all the inequality constraints are ignored, but the equality constraints are considered. The MOR procedure is applied to this problem, and the solution is compared with the one obtained with `qpOASES`[1] that solves the original problem. Once the compared solutions are identical, except for a determined tolerance, the inequality constraints are included in the optimization problem and reduced with MOR. The solution accuracy is again compared and validated with `qpOASES`. The standard condensed formulation of (9.1), presented in Chapter 8, is:

$$\min_{g} \frac{1}{2} g^T H g + c^T g \tag{9.4}$$

where $H \in \mathbb{R}^{L\times L}, g \in \mathbb{R}^{L}, c \in \mathbb{R}^{L}$. To combine the data-driven algorithm with the QP solver, some required preliminary assumptions are listed below:

- regarding the solver presented in Chapter 7, it considers as decision variable Δu. In (9.13), one can easily express Δu as function of g, i.e., $\Delta u = u(k-1) + U_F g$;
- it is introduced the control horizon $N_u = 1$ and the prediction horizon $N = 3$, according to the QP solver implementation.

1 see Chapter 6 and Chapter 7 for more information about `qpOASES` solver.

9.5.1 *Equality Constrained QP*

Starting from (9.1), the associated equality constrained optimization problem is:

$$\begin{aligned}
&\min_{g,\Delta u,y} \sum_{i=0}^{N} \|ref(k+i+1) - y(k+i+1)\|_Q^2 + \\
&\qquad \sum_{i=0}^{N_u-1} \|\Delta u(k+i)\|_R^2 + \lambda_g \|g\|^2 \\
&\text{subject to} \quad \begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ \Delta u \\ y \end{bmatrix} \\
&\qquad u(k) = u(k-1) + \Delta u(k) \in \mathcal{U}, \; k = 0, 1..., N-1
\end{aligned} \tag{9.5}$$

Then, using the relations $U_F g = \Delta u$ and $Y_F g = \Delta y$, the problem (9.5) can be solved with respect to the vector g. Condensing the problem (9.5) in the standard QP form it results:

$$\begin{aligned}
&\min_{g} \; \frac{1}{2} g^T H g + c^T g \\
&\text{s.t.} \; A_e g = b_e
\end{aligned} \tag{9.6}$$

where:

$$\begin{aligned}
&H = 2(Y_F^T Q Y_F + \lambda_g I_{L\times L} + U_F^T R U_F); \\
&c = -2 Y_F^T Q ref, \quad A_e = \begin{bmatrix} U_P \\ Y_P \end{bmatrix}, \; b_e = \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix}
\end{aligned} \tag{9.7}$$

and $I_{L\times L}$ is the identity matrix of indicated dimension. There are several methods that can be used to solve (9.7), such as the null space method or range space method. Assuming the objective function of the minimization problem to be a convex function, the necessary conditions are also sufficient for optimality [18]. Let's consider the Karush–Kuhn–Tucker (KKT) system associated with problem (9.5):

$$\underbrace{\begin{bmatrix} H & A_e^T \\ A_e & 0 \end{bmatrix}}_{KKT_{eq}} \underbrace{\begin{bmatrix} g^* \\ \mu_e^* \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} -c \\ b_e \end{bmatrix}}_{c_{eq}} \tag{9.8}$$

The superscript $*$ indicates the optimal solution. Differently from Chapter 7, the system in (9.8) is considered instead of using penalty functions. In fact, the resulting optimization problem with penalty function could be ill-conditioned because of the introduction of large weight parameters on the constraints.

The MOR strategy is applied to the relation (9.8). One can notice that the optimization problem does not have constant matrices during time. The current reference can change, and the initial conditions are update at every control sample. This suggests that the reduced problem should be able to represent the original one in the entire parameter space. The adopted technique is represented by a parametric POD, which is a state-of-art MOR techniques [44].

The goal is to represent (9.8) with an explicit affine parameters dependence. The parameter space includes the initial conditions $w_{ini} = [u_{ini}\ y_{ini}]^T \in \mathbb{R}^{(m+p)T_{ini}}$ and the current reference $ref = i^r_{dq} \in \mathbb{R}^{pN}$, where m is the dimension of the system input ($m = 2$) and p the dimension of system output ($p = 2$), and i^r_{dq} is the dq current reference instant k. The dimension of these vectors depends on the choice of N and T_{ini}. In particular, the problem (9.8) can be rewritten as in the following:

$$KKT_{eq}x = (c_1 ref + c_2 w_{ini}), \tag{9.9}$$

$c_{ref} \in \mathbb{R}^{(L+(m+p)T_{ini})\times pN}$ and $c_{ref} \in \mathbb{R}^{(L+(m+p)T_{ini})\times(m+p)T_{ini}}$. Applying the POD method to (9.9), a reduced order problem is obtained, i.e., the algorithm finds the matrix $V \in \mathbb{R}^{(L+(m+p)T_{ini})\times r}$ using Galerkin projection (see Chapter 5), introducing a reduced variable g_r s.t. $g \approxeq V g_r$. The introduction of this definition in (9.9) leads to:

$$KKT_{eq}V g_r \approxeq (c_1 ref + c_2 w_{ini}) + res \tag{9.10}$$

and the matrix V is chosen such that the projection of residual res onto the space spanned by the column of V is in the null space of V, i.e., $V^T res = 0$. Thus, the reduced order system is obtained:

$$\hat{KKT}_{eq} g_r = (\hat{c}_1 ref + \hat{c}_2 w_{ini}) \tag{9.11}$$

where:

$$\hat{KKT}_{eq} = V^T KKT_{eq} V;\ \ \hat{c}_1 = V^T c_1;\ \ \hat{c}_2 = V^T c_2 \tag{9.12}$$

Four matrices $\hat{KKT}_{eq} \in \mathbb{R}^{r\times r}$, $\hat{c}_1 \in \mathbb{R}^{r\times mN}$, $\hat{c}_2 \in \mathbb{R}^{r\times(m+p)T_{ini}}$, and V are obtained as output of the algorithm. There are several ways to implement the POD method. A draft of the implemented algorithm is reported in the Algorithm 3. In particular, a pseudo-random approach has been selected. The idea is to build the projection matrix V by randomly exploring the parameter space. This approach has shown promising results in different applications [116–118]. Furthermore, the residual evaluation is performed considering only the variable g, because it is the variable of interest and the Lagrange multipliers are unnecessary to obtain the final solution.

Algorithm 3 pMOR Algorithm of (9.8)

Input: Matrices KKT_{eq}, b_e and c, which are parametric in u_{ini}, y_{ini} and ref. The parameters ranges are: $u_{ini} = [u_{min}, u_{max}]$,$y_{ini} = [i_{min}, i_{max}]$, $ref = [ref_{min}, ref_max]$.
Step {0} Set initial values for parameters, i.e., $u_{ini} = \frac{u_{min}+u_{max}}{2}$, $y_{ini} = \frac{y_{min}+y_{max}}{2}$, $ref = \frac{ref_{min}+ref_{max}}{2}$
Set $res_\star = +\infty$
Set N_{rand} (e.g. $N_{rand} = 20$)
Set a desired value of η (e.g. $\eta = 10^{-4}$)
while $res_\star > \eta$ **do**
  Step {1} Find the solution $[g; \mu_e]$ of (9.8)
  Step {2} Update the orthonormal basis V
  Step {3} Generate/Update the reduced order problem
  Step {4} Generate N_{rand} random values of $u_{ini} = [u_{min}, u_{max}]$, $y_{ini} = [i_{min}, i_{max}]$, $ref = [ref_{min}, ref_{max}]$
  for $h = 1, \cdots, N_{rand}$ **do**
    Select the h-th random set of parameters,
    Step {5} Find the solution $[\hat{g}; \hat{\mu}_e]$ of the reduced order problem
    {6} Evaluate the residual, res, with respect to the full order problem
  end for
  Step {7} Find the set of parameters generated at step 4 which maximizes the residual and assign the corresponding maximum residual to $res_\star$. The new parameter set is used for solving step 1
end while
Output: Reduced order model and the projection matrix, i.e., $\hat{KKT}_{eq} = V^T KKT_{eq} V$; $\hat{c}_1 = V^T c_1$; $\hat{c}_2 = V^T c_2$

9.5.2 *Include Inequality Constraints*

The inequality constraints on the input are the voltage limit equations of the inverter, that represent a hexagon centered to the origin in a stationary reference frame. In a rotating reference frame (dq-reference frame), the hexagon rotates with an angular speed which is the electro-mechanical speed of the motor $\omega_{me} = \frac{d\theta_{me}}{dt}$. The standard constrained QP problem is represented in (9.13).

$$\begin{aligned} \min_g \ & \frac{1}{2} g^T H g + c^T g \\ \text{s.t.} \quad & A_e g = b_e \\ & A_{in} g \leqslant b_{in} \end{aligned} \tag{9.13}$$

where matrices A_{in} and b_{in} describe the hexagon and are reported below:

$$F = MT(\theta_e) = \begin{bmatrix} 1 & \sqrt{3} \\ 1 & 0 \\ 1 & -\sqrt{3} \\ -1 & -\sqrt{3} \\ -1 & 0 \\ -1 & \sqrt{3} \end{bmatrix} \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix}, \tag{9.14}$$

$$A_{in} = FU_F,$$

$$b_{in} = \frac{2u_{DC}}{\sqrt{3}} b - Fu_{k-1},\ b = \begin{bmatrix} 1 & 0.5 & 1 & 1 & 0.5 & 1 \end{bmatrix}^T$$

u_{DC} is the DC bus voltage, $T(\theta_e)$ is the dq to $\alpha\beta$ matrix transformation, and u_{k-1} is the dq voltage at instant $k-1$. The hexagon is represented as six lines with proper angular coefficients. Introducing the solver algorithm presented in Chapter 7, if an unfeasible solution is found by the unconstrained QP problem, the number of constraints that can be violated simultaneously can be one, two or three.

Briefly, if a constraint is violated, a penalty term on the cost function that penalized the related constraint is added. The possible steps of the QP solver are described in Chapter 7. Here, it is recalled the case where one constraint of the hexagon is violated. In fact, since the algorithm considered one violated constraint at a time, the procedure at each step is closed to the case of one constraint violation. Given the unconstrained solution of the problem (9.5) $\Delta u^{uc} = U_F g^{uc}$, if it violates one constraint (represented by an edge of the hexagon and identified with index i, $i = 1, ..., 6$), the feasible optimal solution certainly lies on the segment itself. The feasible solution is computed according to the following passages:

1. The solution of the problem (9.13), namely Δu^{oa}, is computed by considering the penalty related to the constraint i;
2. a feasibility check is then operated, computing

$$h_j := F(j,:)\Delta u^{oa} - f(j) \leqslant 0 \text{ for } j = i-1, i+1; \tag{9.15}$$

   a) if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$ and lies on the segment;
   b) if Δu^{oa} violates one of the two adjacent constraints, the solution is on the related intersection.

If the solution is unfeasible and it violates the constraint j (or the constraint j is considered at the current step if more than one boundary

has been crossed), the following associated KKT system is considered:

$$\underbrace{\begin{bmatrix} H & A_e^T & A_{in}(j,:)^T \\ A_e & 0 & 0 \\ A_{in}(j,:) & 0 & 0 \end{bmatrix}}_{KKT_{in}} \begin{bmatrix} g^* \\ \mu_e^* \\ \mu_{in}^* \end{bmatrix} = \underbrace{\begin{bmatrix} -c \\ b_e \\ b_{in}(j) \end{bmatrix}}_{rhs} \tag{9.16}$$

As before, the aim is to find a reduced problem of (9.16). In this case, the parameter space results larger than the equality constrained problem. In fact, the new parameters introduced by the inequality constraints are: the electric position θ_e, the bus DC value u_{DC}, and the index i of the considered constraint among the six which describe the hexagon. Then, the algorithm proceeds as before with larger set of parameters. The problem (9.16) has to be written explicating its parameter dependence, as in the following:

$$\begin{aligned} KKT_{in} &= H_1 + H_2 M(j,1)\cos(\theta_e) + H_3 M(j,2)\cos(\theta_e) \\ &\quad + H_4 M(j,1)\sin(\theta_e) + H_5 M(j,2)\sin(\theta_e); \\ rhs &= c_1 ref + c_2 [u_{ini}; y_{ini}]^T + c_3 u_{DC} b(j) \\ &\quad + c_4 M(j,1)\cos(\theta_e) u_{k-1} \\ &\quad + c_5 M(j,2)\cos(\theta_e) u_{k-1} \\ &\quad + c_6 M(j,1)\sin(\theta_e) u_{k-1} \\ &\quad + c_7 M(j,2)\sin(\theta_e) u_{k-1}; \end{aligned} \tag{9.17}$$

Thus, an affine representation of (9.16) with respect to the parameters is obtained. The goal of MOR is to find a reduced system by finding a projection matrix V_{in}.

9.6 VALIDATION TEST

In this section, some preliminary results are reported. In particular, the method has been validated by evaluating the accuracy of the solution found by the reduced system of (9.8) and (9.17).

9.6.1 *Case Study*

The DeePC algorithm in its incremental formulation has been considered, applied for controlling the current of the same Interior Permanent Magnet (IPM) used in Chapter 8, whose parameters are given in Table 10. The following parameters for the DeePC have been selected:

$$T_{ini} = 1; N = 3; N_u = 1; T = 100; \tag{9.18}$$

remebering from Chapter 4 that $T \geqslant (m+1)(T_{ini} + N + n_B) - 1$, where n_B is an estimation of the state dimension, it results $L =$

Table 10: IPM motor parameters.

Motor Data	Symbol	IPM
Pole pairs	p	3
Phase resistance	R	$1\,\Omega$
d-axis inductance	L_d	$0.010\,H$
q-axis inductance	L_q	$0.014\,H$
PM flux-linkage	Λ_{pm}	$0.26\,V\,s$
Nominal current	I_N	$6.2\,Arms$
Nominal d current	$I_{N,d}$	$-1.1\,A$
Nominal q current	$I_{N,q}$	$8.7\,A$
Nominal speed	Ω_N	$1000\,rpm$

$T - T_{ini} - N + 1 = 100$. Thus, the dimension of the original high-dimensional system (9.8) is $L + (m+p)T_{ini} = 104$, and for the system with inequalities (9.16) is $L + (m+p)T_{ini} + 1 = 105$, because it is considered one constraint at a time. The tolerance for computing the Reduced Order Model (ROM) has been set to $\ni = 0.0001$.

9.6.2 *MOR of Equality Constrained Problem*

The Algorithm 3 starts with an initial set of parameters. At a generic iteration, the high-dimensional problem is solved (step 1), the matrix V is updated by adding the solution by orthonormalization procedure (step 2). The updated reduced model is built (step 3) and N_{rand} number of parameters are generated randomly (step 4). The reduced problem is tested (step 5) and the residual is evaluated (step 6). Then, the while loop iterates until the residual between the solution of the reduced model and the high-dimensional problem (step 7) is below the tolerance μ.

In Figure 9.1, it is reported the trend of the residual during the MOR process. As can be seen, the algorithm stops after six iteration. It is highlighted that this number also indicates the dimension of the reduced problem. It can be observed that the procedure stops with a residual which approaches zero. This means that the resulting system is drastically reduced in size and it describes the original high-dimensional problem very accurately. The reduced problem is then tested with another set of random parameters, which has not been used during the building process. This is considered as a validating test of the model. Furthermore, a comparison with the solution found by the solver `qpOASES` has been performed. In Figure 9.2, there are reported two examples of how the reduced model finds a very accurate solution. These two tests have been conducted with two different sets

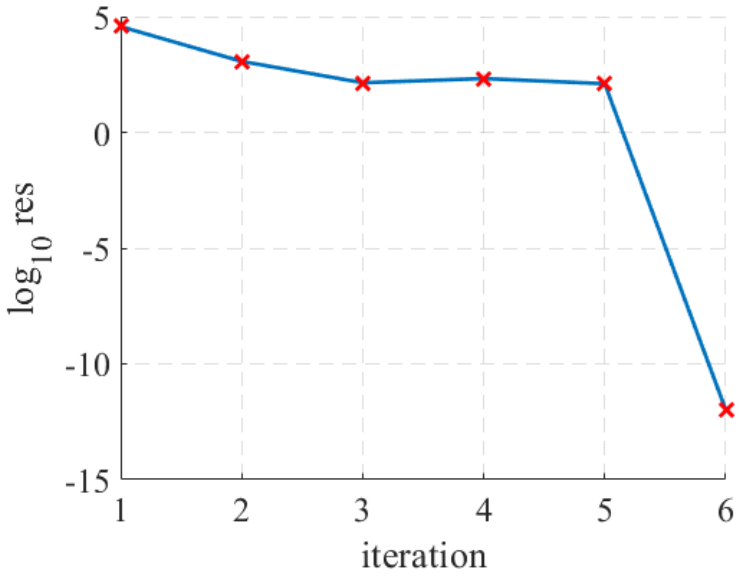
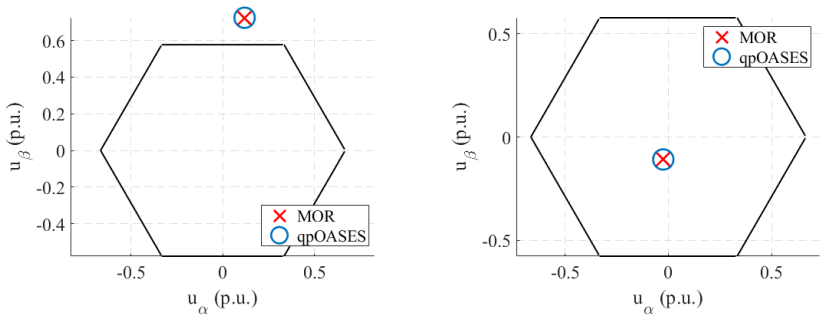


Figure 9.1: Iterations performed by the MOR algorithm for reducing problem (9.8).

of parameters, randomly chosen in the parameter space. The solver `qpOASES` computes the solution of the original high-dimensional problem (9.6), while the cross in Figure 9.2(a) and Figure 9.2(b) report the solutions found with the reduced model. As can be seen, these solutions overlap. This was expected since the residual in Figure 9.1 at the final iterate is almost zero. The evident computational advantage is that the problem has been reduced in size from dimension 104 to dimension 6.

9.6.3 *MOR of Inequality Constrained Problem*

The same validation procedure presented in the previous section can be applied to the problem (9.16). The MOR algorithm proceeds in the same way of Algorithm 3. In Figure 9.3, it has been reported the iterations performed by the MOR procedure, and now the reduced model has dimension 9. As before, the algorithm ends with the last iteration which approaches zero residual with respect to the high-dimensional problem. As before, the obtained problem is validated by comparing its solution with `qpOASES`. In this case, the parameters are chosen such that the unconstrained solution is found outside one constraint of the hexagon, since the QP solver presented in Chapter 7 consists in a finite succession of solutions found by considering one constraint at time. Two examples of results have been reported in Figure 9.4. Firstly, the proposed method uses the reduced system with the equality constraints to find the unconstrained solution. Then, a check of the feasibility is performed, and it results that one constraint



(a) Test 1. (b) Test 2.

Figure 9.2: Validation examples: the solution computed by `qpOASES` of the high-dimensional problem (9.8) with equality constraints is compared with the one obtained with the reduced model, for a given set of parameters. This problem is exploited for finding the unconstrained solution.

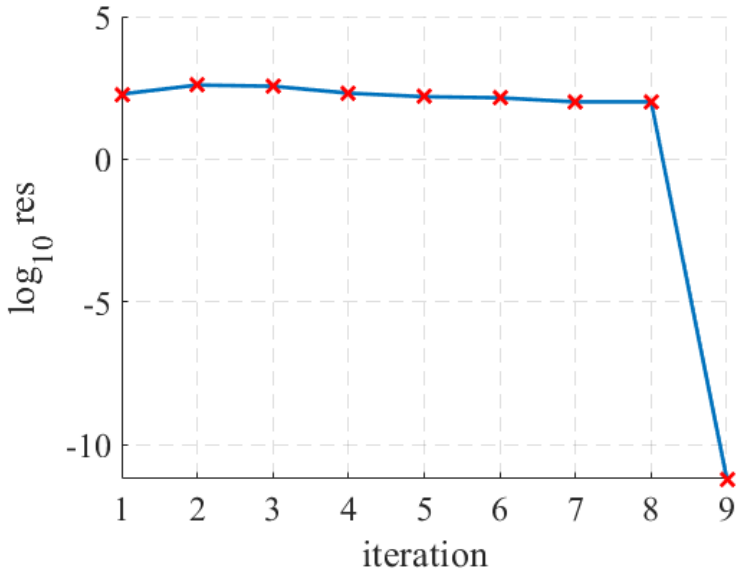
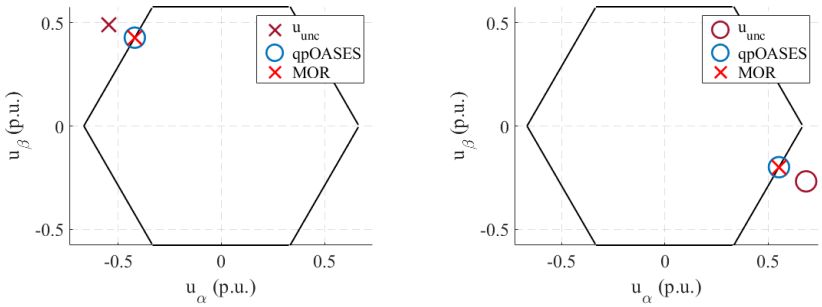


Figure 9.3: Iterations performed by the MOR algorithm for reducing problem (9.8).

has been violated. Thus, the reduced problem with the considered inequality constraint is solved. This solution is compared with the one obtained by `qpOASES` with the original high-dimensional problem. As result, the solution accuracy is preserved, while the problem has been reduced in size of a factor 10.

9.7 DISCUSSION AND FUTURE CHALLENGE

In this chapter, a possible way to overcome the problem of the computational complexity of the DeePC algorithm has been proposed.



(a) Test 1. (b) Test 2.

Figure 9.4: Validation examples: the solution computed by `qpOASES` of the high-dimensional problem (9.16) with equality constraints is compared with the one obtained with the reduced model, for a given set of parameters. This problem is exploited for finding the unconstrained solution.

In particular, when inequality input constraints are considered, the cost for solving the optimization problem becomes a cumbersome aspect, and the real-time feasibility of the algorithm is completely lost. The proposed method aims at using more efficiently the custom QP solver presented in Chapter 7, by considering a MOR technique, which use a POD method, that drastically reduces the size of the original high-dimensional data-driven model, while preserving accuracy of the solution. In particular, a parametric representation of the data-driven model has been obtained, and the reduced parametric problem has been exploited at solving the constrained MPC using the QP solver for the current control of a synchronous motor. Preliminary results demonstrate that this method can drastically reduce the size of the data-driven model, at least of a factor 10. The accuracy of the solution is preserved, and tests have been carried out that validate the proposed strategy by comparing the solution found by the solver `qpOAES`.

Future researches will focus on the possibility to further reduce the size of the problem, testing other MOR techniques. There are several challenges related to this problem: the first one is strictly related with the data-driven strategy of the DeePC, and it is the possibility to take into account nonlinearities of the system, and then exploits a suitable reduction strategy. Furthermore, a further investigation in the reduced order modeling literature will help to find systematic way to choose the size of the reduced model, while preserving a desired accuracy.

CONCLUSIONS AND FUTURE DEVELOPMENTS

The research carried out in this thesis covered different aspects of control of electric drives. In particular, optimization-based control techniques for synchronous motor drives have been studied, with a lot of emphasis on computational aspects and deployments of such algorithms.

Two complementary paradigms to control the machine have been considered, i.e., the model-based and the data-driven approach. The theory of Model Predictive Control (MPC) has been presented. It has been highlighted the main MPC advantages of easily handling multiple-input-multiple-output systems and the possibility to directly include the physical limits of the machine into the optimization problem. The most widespread MPC method uses in power electronics application is represented by the finite-set strategy. However, my thesis analyzed the other approach to MPC, the continuous-set one, which includes a modulation strategy for the inverter. With the presence of input/output constraints, a Quadratic Programming (QP) problem needs to be solved, requiring the presence of an iterative algorithm.

Two main aspects of MPC have been investigated: the integral action of MPC and an efficient method for solving continuous-set problems in real-time. One one hand, it has been shown the comparison between the velocity-form MPC and the MPC coupled with a Kalman filter observer by means of experiments, with a focus on the computational performance of the two approaches, adopting the solver `qpOASES` for its versatility. The equivalence of these two methods has been pointed out. The velocity-form MPC, which considers the incremental formulation of the motor model, turns out to be an easy and effective method for overcoming bias errors introduced by model deviations, with a reduced number of parameters to be tuned, thus making the process simple also for non-high specialized personnel. On the other hand, the problem implementing efficient algorithms which can solve the optimization problem in limited hardware at high control frequency has been addressed. It has been presented an efficient custom QP solver for electric drives applications, whose computational complexity can be easily evaluated and the worst-case computational cost is uniquely determined.

Then, data-driven controls have been presented for controlling motor currents using only input and output data collected from the machine. Fundamental theory regarding these methods have been recalled, in the case of linear systems. The Subsapce Predictive Control (SPC) and the Data EnablEd Predictive Control (DeePC), which

solve an optimization problem have been analyzed and compared with the model-based approach, showing promising performance.

The computational burden of data-driven algorithms is currently under investigation. If a lot of data are needed, the complexity of the optimization problem grows significantly, precluding the real-time feasibility. The idea is to reduce the problem size via model order reduction techniques, which find an approximation of the original optimization problem by projecting it onto a smaller subspace. The result is a considerably smaller problem, while keeping accuracy with respect the original high-fidelity model. Then, the optimization problem could be solved exploiting the proposed solver.

Part IV

APPENDIX

A

SPACE VECTORS

In this chapter are presented the mathematical instruments that are essential to understand the analysis and modeling of electric machines. In particular, the notion of space vector is introduced, together with the system coordinate transformation which are adopted in this work.

A generic three-phase system is a quadrupole which characterized by three terminals, called phases, and a fourth connector, the neutral terminal. In Figure A.1 it is represented a general three-phase system as a box, assuming the user convention. The behaviour of the system is determinated by three currents (the phase currents $i_a(t)$, $i_b(t)$, $i_c(t)$) and a triad of voltages $u_a(t)$, $u_b(t)$, $u_c(t)$ that refer to the neutral wire. The difference between phase voltages gives the so called line-to-line voltages $u_{ab}(t)$, $u_{bc}(t)$, $u_{ca}(t)$, and at any instant their sum is zero for the Kirchhoff's voltage law. In principle, these magnitudes are function of time. The introduction of appropriate mathematical tools such as space vectors leads to introduce a considerable simplification in the description of three-phase systems. A general definition of space vector is given in (A.1).

$$\underline{g}(t) = \frac{2}{3}\left(g_a(t) + g_b(t)e^{i\frac{2\pi}{3}} + g_c(t)e^{i\frac{4\pi}{3}}\right) \tag{A.1}$$

The variables $g_{a,b,c}(t)$ are real functions that may, for example, be the three supply voltages or the phase currents, and they do not need to be sinusoidal. Equation (A.1) allows to introduce a complex vector $\underline{g}(t)$ that is associated to the triad $g_{a,b,c}(t)$. First, consider the case that the three-phase functions of voltages and currents are instantaneously balanced, i.e. (A.2) holds for all the values of t.

$$g_0(t) = \frac{g_a(t) + g_b(t) + g_c(t)}{3} = 0 \tag{A.2}$$

Equation (A.1) is a complex function, and it can be decomposed in its real and imaginary part:

$$\begin{aligned} \mathrm{Re}[\underline{g}(t)] &= g_\alpha(t) = \frac{2}{3}\left(g_a(t) - \frac{1}{2}g_b(t) - \frac{1}{2}g_c(t)\right) \\ \mathrm{Im}[\underline{g}(t)] &= g_\beta(t)\frac{2}{3}\left(-\frac{\sqrt{3}}{2}g_b(t) + \frac{\sqrt{3}}{2}g_c(t)\right) \end{aligned} \tag{A.3}$$

If the relation (A.2) does not hold for all t, the system is said to be unbalanced. The $\alpha\beta$ transformation is still possible. The general g_{abc} can be decomposed as:

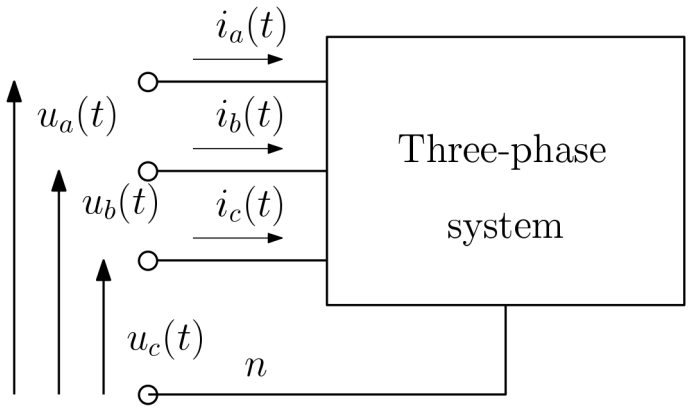


Figure A.1: Representation of a space vector in αβ reference frame.

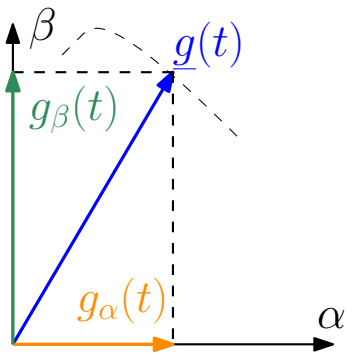


Figure A.2: Representation of a space vector in αβ reference frame.

$$\begin{aligned} g_a &= g'_a + g_0 \\ g_b &= g'_b + g_0 \\ g_c &= g'_c + g_0 \end{aligned} \tag{A.4}$$

where g'_{abc} represents the balance component of g_{abc}. The definition in (A.1) can be still applied to (A.4) and it is demonstrated that:

$$\begin{aligned} \underline{g}(t) = \frac{2}{3}\left(g_a(t) + g_b(t)e^{j\frac{2\pi}{3}} + g_c(t)e^{j\frac{4\pi}{3}}\right) = \\ \frac{2}{3}\left(g'_a(t) + g'_b(t)e^{j\frac{2\pi}{3}} + g'_c(t)e^{j\frac{4\pi}{3}}\right) \end{aligned} \tag{A.5}$$

i.e. the transformation does not contain the homopolar component.

The definition in (A.3) with the following notation: $\underline{g} = g_\alpha + jg_\beta$, indicates a function of time which is called *space vector*. In Figure A.2, the space vector is represented with the two orthogonal component in a complex plane, where the real axis is called α-axis and the β-axis is

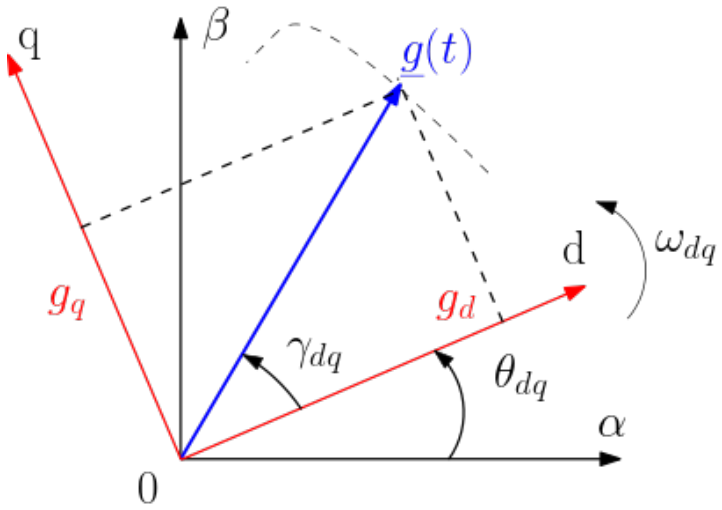


Figure A.3: Representation of a space vector in rotating dq reference frame.

the imaginary one. This transformation can be represented in matrix form as following:

$$\begin{bmatrix} g_\alpha \\ g_\beta \\ g_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & \frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} g_a \\ g_b \\ g_c \end{bmatrix} \tag{A.6}$$

$$\Rightarrow \; g_{\alpha\beta 0} = T_{abc\rightarrow\alpha\beta}\; g_{abc}$$

The relation in (A.6) is the known as Clarke's transformation or $\alpha\beta 0$-transformation, and it is employed to simplify the analysis of three-phase circuits. In (A.6) and from here on, the explicit time dependence is omitted to simplify the notation. This relation is *amplitude conservative*, thank to the term $\frac{2}{3}$. The g_0 component is also known as the homopolar component.

Another useful mathematical tool is the so called Park's transformation. Consider the Figure A.3, which shows that is possible to represent the space vector $\underline{g}\alpha\beta = |\underline{g}|e^{j\theta_{\alpha\beta}}$ with respect to a rotating orthogonal reference frame, whose axes are called dq axes. The origin of these reference frame coincides with the one of $\alpha\beta$ axes, but it rotates with an angular speed $\omega_{dq} = \frac{d\theta_{dq}}{dt}$. The rotational angle θ_{dq} is in general a function of time. Observing the Figure A.3, the following equivalent representation holds:

$$\begin{aligned} \underline{g}_{\alpha\beta} &= |\underline{g}|e^{j\theta_{\alpha\beta}} = |\underline{g}|e^{j(\gamma_{dq}+\theta_{dq})} = \underline{g}_{dq}e^{j\theta_{dq}} \\ &\Rightarrow \; \underline{g}_{dq} = \underline{g}_{\alpha\beta}e^{-\theta_{dq}} \end{aligned} \tag{A.7}$$

Equation (A.7) points out the relation between $\alpha\beta$ and dq reference frame, which in compact matrix notation becomes:

$$\begin{bmatrix} g_d \\ g_q \end{bmatrix} = \begin{bmatrix} \cos\theta_{dq} & \sin\theta_{dq} \\ -\sin\theta_{dq} & \cos\theta_{dq} \end{bmatrix} \begin{bmatrix} g_\alpha \\ g_\beta \end{bmatrix} \tag{A.8}$$

$$\Rightarrow \; g_{dq} = T_{\alpha\beta\to dq}\, g_{\alpha\beta}$$

The matrices $T_{abc\to\alpha\beta}$ and $T_{\alpha\beta\to dq}$ are the so called Clarke and Park transformation matrices respectively. The inverse transformation is computed by computing the inverse of the matrix $T_{\alpha\beta\to dq}$. This is an orthonormal matrix, i.e its inverse is equal to the transpose, as represented in (A.9).

$$T_{dq\to\alpha\beta} = T^{-1}_{\alpha\beta\to dq} = T^{T}_{\alpha\beta\to dq} = \begin{bmatrix} \cos\theta_{dq} & -\sin\theta_{dq} \\ \sin\theta_{dq} & \cos\theta_{dq} \end{bmatrix} \tag{A.9}$$

If one wants to consider even the homopolar component, the previous relations are obtained by imposing the conservation of such component. Matrices relations become:

$$\begin{aligned} T_{\alpha\beta 0\to dq0} &= \begin{bmatrix} \cos\theta_{dq} & \sin\theta_{dq} & 0 \\ -\sin\theta_{dq} & \cos\theta_{dq} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ T_{dq0\to\alpha\beta0} &= \begin{bmatrix} \cos\theta_{dq} & -\sin\theta_{dq} & 0 \\ \sin\theta_{dq} & \cos\theta_{dq} & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{A.10}$$

B

STATE-FEEDBACK CONTROL

B.1 INTRODUCTION

State-feedback control is a traditional way to stabilize a dynamical system to the origin. However, the presence of disturbances on the plant could significantly reduce the action of this controller. This report gives a comprehensive description of how the integral action can be included in state-feedback control in order to compensate constant disturbances acting on the input signal. The possibility to include a desired reference trajectory different from the origin is also described.

B.2 REFERENCE TRACKING

In the following, it is assumed that the open-loop dynamical system completely reachable and observable. It has been shown that state feedback control law brings the output $y(k)$ to zero asymptotically. The first step to introduce the possibility to make the output $y(k)$ track a generic constant set-point $r(k) = r$ is to reformulate the control input as follows (see Figure B.1).

$$u(k) = Kx(k) + v(k) \tag{B.1}$$
$$v(k) = Fr(k) \tag{B.2}$$

The state-space representation becomes:

$$\begin{aligned} x(k+1) &= (A + BK)x(k) + BFr(k) \\ y(k) &= Cx(k) \end{aligned} \tag{B.3}$$

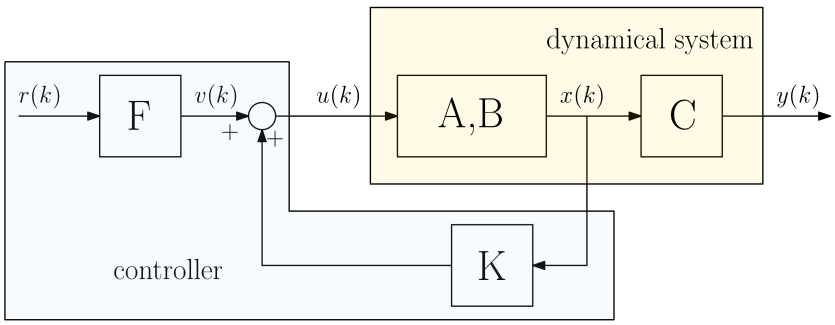


Figure B.1: Reference tracking with state-feedback control law.

The design requires to choose the gain F properly to ensure reference tracking. The DC gain from the reference to the output has to be a unit gain in order to have $y(k) \Rightarrow r$ asymptotically. This condition is shown in (B.4).

$$C(I-(A+BK))^{-1}BF = I \tag{B.4}$$

In the following the assumptions are that the number of inputs equal to the number of outputs and the DC gain from u to y is invertible, i.e. $C\,adj(I-A)B$ is invertible. Since the state feedback doesn't change the zeros in closed-loop, the relation in (B.5) holds.

$$C\,adj(I-A-BK)B = C\,adj(I-A)B \tag{B.5}$$

Thus, the left side of (B.5) is also invertible. Combining this result with (B.4), the gain F assumes the expression in (B.6).

$$F = (C(I-(A+BK))^{-1}B)^{-1} \tag{B.6}$$

The just presented procedure has the advantage to easily combine the state-feedback control law with reference tracking. One problem that may arise is the presence of model uncertainties and exogenous disturbances which affect the dynamical process. In this case, the design of the gain F is not suitable because it is based on the nominal representation of the process.

B.2.1 *Integral action for disturbance rejection*

There are many kind of disturbances and uncertainties that can affect the dynamical system. For instance, the control actuator usually contains several non-idealities. Furthermore, nominal parameters of the process may be different from the actual values, or they can be affected by different phenomena (temperature, pressure etc..). One way to increase the robustness with respect disturbances is to introduce an *integral action* in the control law. The following analysis considers the presence of an *input disturbance* $d(k)$ as represented in Fig. B.2.

Consider the problem of output regulation, i.e. the design of a controller that drives the output $y(k)$ to the reference $r(k) \equiv 0$, under the presence of the input disturbance $d(k)$. The first step is to augment the open-loop system with the integral of the output vector represented in (B.7).

$$z(k+1) = z(k) + y(k) \tag{B.7}$$

The equations in (B.8) represent the augmented open-loop state space model.

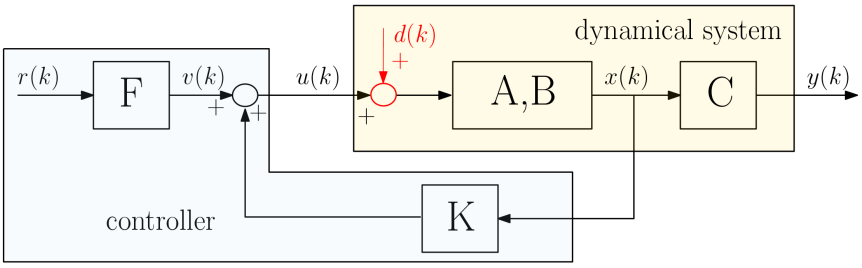


Figure B.2: Reference tracking with input disturbance $d(k)$.

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ z(k+1) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} B \\ 0 \end{bmatrix} d(k) \\ y(k) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} \end{aligned} \tag{B.8}$$

The state $z(k)$ is seen to be the integral of the error between the reference r, and the output y. If it exists a compensator that stabilizes the system, in steady state results thst $z(l+1) = z(k)$, hence $y \Rightarrow r$ in steady state. Assuming that a stabilizing gain $[HK]$ can be designed for the system augmented with integral action of the form in (B.9) for the system (B.8), then the $\lim_{k\to\inf} y(k) = 0$ for all constant disturbances $d(k) \equiv d$.

$$u(k) = \begin{bmatrix} K & H \end{bmatrix} \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} \tag{B.9}$$

This can be easily verified by looking at the state-update matrix of the closed-loop system in (B.10).

$$\begin{bmatrix} A & 0 \\ C & I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} B & H \end{bmatrix} \tag{B.10}$$

This matrix has asymptotically stable eigenvalues by construction.

The last step is to generalize the just presented result for a generic set-point tracking. The idea is to use the same feedback gains designed earlier. Now the integral of the tracking error is considered, as in (B.11).

$$z(k+1) = z(k) + (y(k) - r(k)) \tag{B.11}$$

In Fig. B.3 it is represented the principle scheme of the controller.

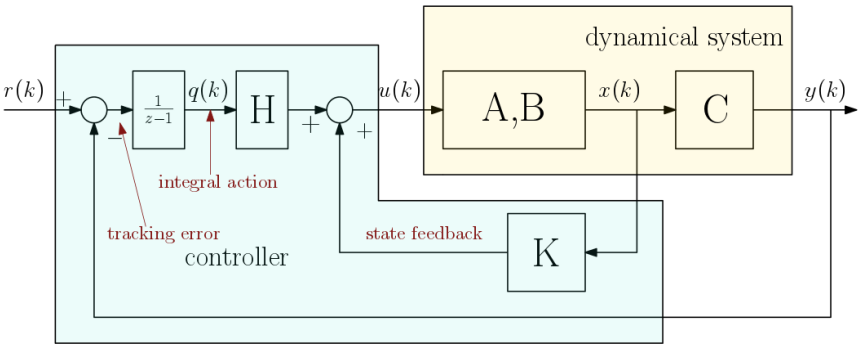


Figure B.3: Reference tracking with input disturbance d(k) and integral action.

C

APPENDIX TEST

C.1 APPENDIX SECTION TEST

Let's consider the modified cost function (4.18). Substituting the expressions of $u = U_F g$ and $y = U_F g$:

$$\begin{aligned} &\min_{g,\sigma_y} \ (\|y^{ref} - Y_F g\|_Q^2 + \|U_F g\|_R^2 + \eta\|\sigma_y\|^2 + \beta\|g\|^2) \\ &\text{s.t.} \begin{bmatrix} U_P \\ Y_P \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \end{bmatrix} \end{aligned} \tag{C.1}$$

Now, it is useful to develop matrix multiplications:

$$\begin{aligned} &\min_{g,\sigma_y} \ (g^T(Y_F^T Q Y_F + U_F^T R U_F + \beta)g + -2y_{ref}^T Q Y_F g + \sigma_y^T \eta \sigma_y) \\ &\qquad \text{s.t. } M_P \ g = w_{ini} + \begin{bmatrix} 0 \\ \sigma_y \end{bmatrix} \end{aligned} \tag{C.2}$$

As before, a solution g of (C.2) can be written as the sum of a particular solution and one that is in the kernel of $\begin{bmatrix} U_P \\ Y_P \end{bmatrix}$:

$$g = \hat{A}\left(w_{ini} + \begin{bmatrix} 0 \\ \sigma_y \end{bmatrix}\right) + \Phi z, \ \hat{A} := M_P^{\dagger} \tag{C.3}$$

Then, substituting (C.3) in (C.2):

$$\begin{aligned} &\min_{z,\sigma_y} \ (\hat{A}w_{ini} + \hat{A}_r\sigma_y + \Phi z)^T H(\hat{A}w_{ini} + \hat{A}_r\sigma_y + \Phi z) \\ &\qquad - 2y^{ref,T} Q Y_F(\hat{A}w_{ini} + \hat{A}_r\sigma_y + \Phi z) + \sigma_y^T \eta \sigma_y \end{aligned} \tag{C.4}$$

where $\hat{A}_r$ considers the last pT_{ini} columns of $\hat{A}$, and $H = Y_F^T Q Y_F + U_F^T R U_F + \beta$. Developing the matrix products:

$$\begin{aligned} &\min_{z,\sigma_y} \ \sigma_y^T(\hat{A}_r^T H \hat{A}_r + \eta)\sigma_y + z^T(\Phi^T H \Phi)z + 2z^T \Phi^T H \hat{A}_r \sigma_y \\ &\qquad + 2w_{ini}^T \hat{A}^T H \hat{A}_r \sigma_y + 2w_{ini}^T \hat{A}^T H \Phi z \\ &\qquad - 2y_{ref}^T Q Y_F \hat{A}_r \sigma_y - 2y_{ref}^T Q Y_F \Phi z \end{aligned} \tag{C.5}$$

Now, the following definitions are introduced:

$$
\begin{aligned}
\varepsilon &:= \begin{bmatrix} z \\ \sigma_y \end{bmatrix} \in \mathbb{R}^{L_z + p T_{ini}},\ L_z = \dim\{\ker(M_p)\} \\
z &= I_z \varepsilon,\ \ I_z := [I_{L_z \times L_z}\ 0_{L_z \times p T_{ini}}] \\
\sigma_y &= I_\sigma \varepsilon,\ \ I_\sigma := [0_{p T_{ini} \times L_z}\ I_{p T_{ini} \times p T_{ini}}]
\end{aligned}
\tag{C.6}
$$

where the term I is the identity matrix with specified dimension. Substituting (C.6) in (C.5), the optimization problem can be solved with respect to the new variable ε:

$$
\begin{aligned}
\min_{\varepsilon} \frac{1}{2}\ \varepsilon^T & \underbrace{2\Big(I_z^T \Phi^T H \Phi I_z + I_s^T \hat{A}_r^T H \hat{A}_r I_s + I_z^T 2\Phi^T H \hat{A}_r I_s + I_s^T \eta I_s\Big)}_{:=\hat{H}} \varepsilon \\
& \underbrace{- 2\Big(y_{ref}^T Q Y_F (\hat{A}_r I_s + \Phi I_z) - (w_{ini}^T \hat{A}_r^T H)(\hat{A}_r I_s + \Phi I_z)\Big)}_{:=\hat{d}^T} \varepsilon \\
& \Rightarrow \varepsilon^* = \underbrace{(\hat{H} + \hat{H}^T)^{-1}}_{:=G} \hat{d}
\end{aligned}
\tag{C.7}
$$

Then, remembering the previous definitions, the optimal control input can be obtained:

$$
\begin{aligned}
z^* &= I_z \varepsilon^* = I_z G \hat{d} \\
\sigma_y^* &= I_s \varepsilon^* = I_s G \hat{d}
\end{aligned}
\tag{C.8}
$$

$$
\begin{aligned}
g^* &= \hat{A} w_{ini} + \hat{A}_r I_s G \hat{d} + \Phi I_z G \hat{d} \\
&= \hat{A} w_{ini} + (\hat{A}_r I_s + \Phi I_z) G \hat{d} \\
&= \hat{A} w_{ini} + \\
&(\hat{A}_r I_s + \Phi I_z) G \Big(2\Big((\hat{A}_r I_s + \Phi I_z)^T Y_F^T Q y_{ref} - (\hat{A}_r I_s + \Phi I_z)^T (H \hat{A}_r w_{ini})\Big) \\
&= \Big(\hat{A} - 2(\hat{A}_r I_s + \Phi I_z) G (\hat{A}_r I_s + \Phi I_z)^T H \hat{A}_r\Big) w_{ini} + \\
&2\Big(\hat{A}_r I_s + \Phi I_z) G (\hat{A}_r I_s + \Phi I_z)^T Y_F^T Q\Big) y_{ref} \\
&u^* = U_F\ g^* \\
&\Rightarrow u^* = K_{ini}\ w_{ini} + K_r\ y_{ref}
\end{aligned}
\tag{C.9}
$$

Again, it has been shown that the optimal control input can be represented as linear combination of the initial conditions and desired reference. Even in this case, the matrices K_{ini} and K_r can be computed offline.

BIBLIOGRAPHY

[1] E. Commission. *Research and innovation strategy 2020-2024*. https://ec.europa.eu/info/research-and-innovation/strategy/strategy-2020-2024_en (cit. on p. 3).

[2] B. K. Bose. "Global Energy Scenario and Impact of Power Electronics in 21st Century." In: *IEEE Transactions on Industrial Electronics* 60.7 (2013), pp. 2638–2651. DOI: 10.1109/TIE.2012.2203771 (cit. on p. 4).

[3] E. C. for Power Electronics (ECPE). "ECPE Position Paper Power Electronics and Digitalisation (Smart-Inverter)." In: (2020) (cit. on p. 4).

[4] T. M. Jahns, G. B. Kliman, and T. W. Neumann. "Interior Permanent-Magnet Synchronous Motors for Adjustable-Speed Drives." In: *IEEE Transactions on Industry Applications* IA-22.4 (1986), pp. 738–747. DOI: 10.1109/TIA.1986.4504786 (cit. on p. 5).

[5] A. M. Research. "Global Electric Motor Market: Opportunities and Forecast 2018-2025." In: (2017) (cit. on p. 5).

[6] C. Mohtadi and D. W. Clarke. "Generalized Predictive Control, LQ, or Pole-Placement: A Unified Approach." In: *1986 25th IEEE Conference on Decision and Control*. 1986, pp. 1536–1541 (cit. on pp. 5, 21).

[7] C. E. García, D. M. Prett, and M. Morari. "Model Predictive Control: Theory and Practice-A Survey." In: *Automatica* (1989). ISSN: 00051098. DOI: 10.1016/0005-1098(89)90002-2 (cit. on pp. 5, 21, 56).

[8] P. Cortés, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez. "Predictive Control in Power Electronics and Drives." In: *IEEE Trans. Ind. Electron.* 55.12 (2008), pp. 4312–4324. ISSN: 02780046. DOI: 10.1109/TIE.2008.2007480 (cit. on pp. 5, 55, 79).

[9] A. Bemporad. "Model Predictive Control Design: New Trends and Tools." In: *Proceedings of the 45th IEEE Conference on Decision and Control*. 2006, pp. 6678–6683. DOI: 10.1109/CDC.2006.377490 (cit. on pp. 5, 55).

[10] P. Krause, O. Wasynczuk, S. Sudhoff, and S. Pekarek. *Analysis of Electric Machinery and Drive Systems*. Third. IEEE Press Series on Power and Energy Systems. Wiley, 2013. ISBN: 9781118024294 (cit. on p. 13).

[11] A. E. Fitzgerald, C. Kingsley, and S. Umans. *Electric Machinery*. McGraw-Hill Education, 2002. ISBN: 0073660094 (cit. on p. 14).

[12] D. White and H. Woodson. *Electromechanical Energy Conversion*. M.I.T. core curriculum program in electrical engineering. Wiley, 1959 (cit. on p. 14).

[13] T. Geyer. *Model Predictive Control of High Power Converters and Industrial Drives*. Ed. by Wiley. First Edition. 2017. ISBN: 978-1-119-01090-6 (cit. on pp. 16, 32).

[14] D. Holmes and T. Lipo. *Pulse Width Modulation for Power Converters: Principles and Practice*. IEEE Press Series on Power and Energy Systems. Wiley, 2003. ISBN: 9780471208143. URL: https://books.google.it/books?id=8LGi1AjSfpcC (cit. on pp. 17, 20).

[15] S. Bolognani and M. Zigliotto. "Novel digital continuous control of SVM inverters in the overmodulation range." In: *IEEE Trans. Ind. Appl.* 33.2 (1997), pp. 525–530. DOI: 10.1109/28.568019 (cit. on pp. 20, 83).

[16] S. Bowes and Y.-S. Lai. "The relationship between space-vector modulation and regular-sampled PWM." In: *IEEE Transactions on Industrial Electronics* 44.5 (1997), pp. 670–679. DOI: 10.1109/41.633469 (cit. on p. 20).

[17] E. Camacho and C. Bordons. *Model Predictive Control*. Ed. by Springer. Second Edition. 2007. Chap. 1, p. 3. ISBN: 978-1-85233-694-3 (cit. on p. 21).

[18] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511804441 (cit. on pp. 29, 30, 123).

[19] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. "The explicit linear quadratic regulator for constrained systems." In: *Automatica* 38.1 (2002), pp. 3–20. ISSN: 0005-1098. DOI: https://doi.org/10.1016/S0005-1098(01)00174-1 (cit. on pp. 29, 79).

[20] A. Alessio and A. Bemporad. "A Survey on Explicit Model Predictive Control." In: *Nonlinear Model Predictive Control: Towards New Challenging Applications*. Ed. by L. Magni, D. M. Raimondo, and F. Allgöwer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 345–369. ISBN: 978-3-642-01094-1. DOI: 10.1007/978-3-642-01094-1_29 (cit. on p. 29).

[21] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. "Multi-Parametric Toolbox (MPT)." In: *Hybrid Systems: Computation and Control*. Ed. by R. Alur and G. J. Pappas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 448–462. ISBN: 978-3-540-24743-2 (cit. on p. 30).

[22] D. Kouzoupis, A. Zanelli, H. Peyrl, and H. J. Ferreau. "Towards proper assessment of QP algorithms for embedded model predictive control." In: *2015 European Control Conference (ECC)*. 2015, pp. 2609–2616. DOI: 10.1109/ECC.2015.7330931 (cit. on p. 30).

[23] B. Houska, H. J. Ferreau, and M. Diehl. "An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range." In: *Automatica* 47.10 (2011), pp. 2279 –2285. ISSN: 00051098. DOI: 10.1016/j.automatica.2011.08.020 (cit. on pp. 30, 56).

[24] D. K. M. Kufoalor, B. J. T. Binder, H. J. Ferreau, L. Imsland, T. A. Johansen, and M. Diehl. "Automatic Deployment of Industrial Embedded Model Predictive Control Using qpOASES." In: *2015 European Control Conference (ECC)*. 2015, pp. 2601–2608 (cit. on pp. 30, 60).

[25] J. Kalmari, J. Backman, and A. Visala. "A toolkit for nonlinear model predictive control using gradient projection and code generation." In: *Control Engineering Practice* 39 (2015), pp. 56–66. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2015.01.002. URL: https://www.sciencedirect.com/science/article/pii/S096706611500012X (cit. on p. 30).

[26] P. Patrinos and A. Bemporad. "An Accelerated Dual Gradient-Projection Algorithm for Embedded Linear Model Predictive Control." In: *IEEE Transactions on Automatic Control* 59.1 (2014), pp. 18–33. DOI: 10.1109/TAC.2013.2275667 (cit. on pp. 30, 32).

[27] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel. "Model Predictive Control of Power Electronic Systems: Methods, Results, and Challenges." In: *IEEE Open Journal of Ind. Applications* (2020), pp. 1–1 (cit. on pp. 31, 56).

[28] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena. "Model Predictive Control for Power Converters and Drives: Advances and Trends." In: *IEEE Trans. on Ind. Electron.* 64.2 (2017), pp. 935–947. DOI: 10.1109/TIE.2016.2625238 (cit. on pp. 31, 79, 99).

[29] H. S. Patel and R. G. Hoft. "Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverters: Part I–Harmonic Elimination." In: *IEEE Transactions on Industry Applications* IA-9.3 (1973), pp. 310–317. DOI: 10.1109/TIA.1973.349908 (cit. on p. 32).

[30] H. S. Patel and R. G. Hoft. "Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverters: Part II — Voltage Control Techniques." In: *IEEE Transactions on Industry Applications* IA-10.5 (1974), pp. 666–673. DOI: 10.1109/TIA.1974.349239 (cit. on p. 32).

[31] R. P. Aguilera, P. Acuña, P. Lezana, G. Konstantinou, B. Wu, S. Bernet, and V. G. Agelidis. "Selective Harmonic Elimination Model Predictive Control for Multilevel Power Converters." In: *IEEE Transactions on Power Electronics* 32.3 (2017), pp. 2416–2426. DOI: 10.1109/TPEL.2016.2568211 (cit. on p. 32).

[32] M. Vasiladiotis, A. Christe, and T. Geyer. "Model Predictive Pulse Pattern Control for Modular Multilevel Converters." In: *IEEE Transactions on Industrial Electronics* 66.3 (2019), pp. 2423–2431. DOI: 10.1109/TIE.2018.2868294 (cit. on p. 32).

[33] J. Rodriguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortes, and U. Ammann. "Predictive Current Control of a Voltage Source Inverter." In: *IEEE Transactions on Industrial Electronics* 54.1 (2007), pp. 495–503. DOI: 10.1109/TIE.2006.888802 (cit. on p. 32).

[34] P. Karamanakos, T. Geyer, and R. P. Aguilera. "Long-Horizon Direct Model Predictive Control: Modified Sphere Decoding for Transient Operation." In: *IEEE Transactions on Industry Applications* 54.6 (2018), pp. 6060–6070. DOI: 10.1109/TIA.2018.2850336 (cit. on p. 32).

[35] G. Cimini, D. Bernardini, S. Levijoki, and A. Bemporad. "Embedded Model Predictive Control with Certified Real-Time Optimization for Synchronous Motors." In: *IEEE Trans. Control Syst. Technol.* (2020), pp. 1–8 (cit. on pp. 32, 56, 61, 102, 111).

[36] J. Polderman and J. Willems. *Introduction to Mathematical System Theory – A Behavioral Approach*. Jan. 1997. DOI: 10.1007/978-1-4757-2953-5 (cit. on p. 34).

[37] J. Willems, P. Rapisarda, I. Markovsky, and B. De Moor. "A note on persistency of excitation." In: *Syst. Control Lett.* 54 (Apr. 2005). DOI: 10.1016/j.sysconle.2004.09.003 (cit. on pp. 36, 104, 105, 112).

[38] J. Coulson, J. Lygeros, and F. Dörfler. "Data-Enabled Predictive Control: In the Shallows of the DeePC." In: *2019 18th European Control Conference (ECC)* (2018), pp. 307–312 (cit. on pp. 36, 101, 116).

[39] Z. S. Hou and Z. Wang. "From model-based control to data-driven control: Survey, classification and perspective." In: *Information Sciences* 235 (2013), pp. 3–35. ISSN: 00200255. DOI: 10.1016/j.ins.2012.07.014 (cit. on pp. 36, 99).

[40] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler. "Data-Enabled Predictive Control for Grid-Connected Power Converters." In: *58th Conf. Decision and Control (CDC)*. 2019, pp. 8130–8135. DOI: 10.1109/CDC40024.2019.9029522 (cit. on pp. 36, 101, 104, 108).

[41] I. Markovsky and P. Rapisarda. "Data-driven simulation and control." In: *Int. J. Control* 81.12 (2008), pp. 1946–1959. ISSN: 00207179. DOI: 10.1080/00207170801942170 (cit. on p. 36).

[42] J. Coulson, J. Lygeros, and F. Dörfler. *Distributionally Robust Chance Constrained Data-enabled Predictive Control*. 2020. arXiv: 2006.01702 [math.OC] (cit. on pp. 42, 101).

[43] W. Favoreel, B. De Moor, and M. Gevers. "SPC: Subspace Predictive Control." In: *IFAC Proc. Volumes* 32.2 (1999), pp. 4004 –4009. ISSN: 1474-6670. DOI: https://doi.org/10.1016/S1474-6670(17)56683-5 (cit. on pp. 42, 100, 105).

[44] P. Benner, S. Gugercin, and K. Willcox. "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems." In: *SIAM Review* 57.4 (2015), pp. 483–531. DOI: 10.1137/130932715 (cit. on pp. 45, 49, 124).

[45] A. Quarteroni and G. Rozza. *Reduced Order Methods for Modeling and Computational Reduction*. Springer Publishing Company, Incorporated, 2013. ISBN: 3319020897 (cit. on p. 45).

[46] W. Schilders, H. van der Vorst, and J. Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*. Mathematics in Industry. Springer Berlin Heidelberg, 2008. ISBN: 9783540788416. URL: https://books.google.it/books?id=WWi1evmAQPkC (cit. on pp. 45, 47).

[47] J. Lumley. "The Structures of Inhomogeneous Turbulent Flow." In: *Atmospheric Turbulence and Radio Wave Propagation* (1967), pp. 166 –178 (cit. on p. 49).

[48] L. Sirovich. "Turbulence and the dynamics of coherent structures. I - Coherent structures." In: *Quarterly of Applied Mathematics - QUART APPL MATH* 45 (Oct. 1987), pp. 561–571. DOI: 10.1090/qam/910463 (cit. on p. 49).

[49] O. Ghattas, T. Bui-Thanh, and K. Willcox. "Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space." In: *SIAM* 30 (Jan. 2008), pp. 3270–3288. DOI: 10.1137/070694855 (cit. on p. 51).

[50] B. Haasdonk and M. Ohlberger. "Reduced Basis Method for Finite Volume Approximations of Parametrized Linear Evolution Equations." In: *ESAIM: Mathematical Modelling and Numerical Analysis* 42 (Mar. 2008), pp. 277 –302. DOI: 10.1051/m2an:2008001 (cit. on p. 51).

[51] P. Benner, L. Feng, S. Li, and Y. Zhang. "Reduced-order modeling and rom-based optimization of batch chromatography." In: *Lecture Notes in Computational Science and Engineering* 103 (Jan. 2015), pp. 427 –435. DOI: 10.1007/978-3-319-10705-9__42 (cit. on p. 51).

[52] V. Druskin, C. Lieberman, and M. Zaslavsky. "On Adaptive Choice of Shifts in Rational Krylov Subspace Reduction of Evolutionary Problems." In: *SIAM Journal on Scientific Computing* 32 (Jan. 2010), pp. 2485–2496. DOI: 10.1137/090774082 (cit. on p. 51).

[53] M. Diehl, H. J. Ferreau, and N. Haverbeke. "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation." In: *Lect. Notes Control Inf. Sci.* 2009. ISBN: 9783642010934. DOI: 10.1007/978-3-642-01094-1-32 (cit. on p. 56).

[54] P. Kakosimos and H. Abu-Rub. "Predictive Speed Control with Short Prediction Horizon for Permanent Magnet Synchronous Motor Drives." In: *IEEE Trans. Power Electron.* PP.99 (2017), pp. 1–1. ISSN: 0885-8993. DOI: 10.1109/TPEL.2017.2697971 (cit. on p. 56).

[55] M. Preindl and S. Bolognani. "Model Predictive Direct Speed Control with Finite Control Set of PMSM Drive Systems." In: *IEEE Trans. Power Electron.* 28.2 (2013), pp. 1007–1015. ISSN: 08858993. DOI: 10.1109/TPEL.2012.2204277 (cit. on p. 56).

[56] M. Richter, M. E. Magana, O. Sawodny, and T. K. A. Brekken. "Nonlinear Model Predictive Control of a Point Absorber Wave Energy Converter." In: *IEEE Trans. Sustain. Energy* 4.1 (2013), pp. 118–126. ISSN: 1949-3029. DOI: 10.1109/TSTE.2012.2202929 (cit. on p. 56).

[57] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto. "Design and Implementation of Model Predictive Control for Electrical Motor Drives." In: *IEEE Trans. Ind. Electron.* 56.6 (2009), pp. 1925–1936. ISSN: 02780046. DOI: 10.1109/TIE.2008.2007547 (cit. on pp. 56, 58, 79, 80).

[58] M. Preindl, S. Bolognani, and C. Danielson. "Model Predictive Torque Control with PWM using fast gradient method." In: *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conf. and Exp. (APEC)*. 2013, pp. 2590–2597. DOI: 10.1109/APEC.2013.6520661 (cit. on pp. 56, 80).

[59] S. Mariethoz, A. Domahidi, and M. Morari. "Sensorless explicit model predictive control of permanent magnet synchronous motors." In: *2009 IEEE Int. Electr. Machines and Drives Conf.* 2009, pp. 1250–1257. DOI: 10.1109/IEMDC.2009.5075363 (cit. on pp. 56, 79).

[60] M. Kvasnica. "Implicit vs explicit MPC — Similarities, Differences, and a Path Towards a Unified Method." In: *2016 European Control Conference*. 2016, pp. 603–603. DOI: 10.1109/ECC.2016.7810353 (cit. on pp. 56, 79).

[61] G. Cimini, D. Bernardini, A. Bemporad, and S. Levijoki. "Online model predictive torque control for Permanent Magnet Synchronous Motors." In: *2015 IEEE Int. Conf. on Ind. Tech. (ICIT)*. 2015, pp. 2308–2313. DOI: 10.1109/ICIT.2015.7125438 (cit. on pp. 56, 80).

[62] A. Zanelli, J. Kullick, H. M. Eldeeb, G. Frison, C. M. Hackl, and M. Diehl. "Continuous Control Set Nonlinear Model Predictive Control of Reluctance Synchronous Machines." In: *IEEE Trans. Control Syst. Technol.* (2021), pp. 1–12. DOI: 10.1109/TCST.2020.3043956 (cit. on pp. 56, 80, 81).

[63] H. Zhu, X. Xiao, and Y. Li. "PI Type Dynamic Decoupling Control Scheme for PMSM High Speed Operation." In: *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*. 2010, pp. 1736–1739 (cit. on p. 56).

[64] S.-C. Carpiuc and C. Lazar. "Fast Real-Time Constrained Predictive Current Control in Permanent Magnet Synchronous Machine-Based Automotive Traction Drives." In: *IEEE Transactions on Transportation Electrification* 1.4 (2015), pp. 326–335. DOI: 10.1109/TTE.2015.2482223 (cit. on p. 56).

[65] J. Yang, W. Chen, S. Li, L. Guo, and Y. Yan. "Disturbance/Uncertainty Estimation and Attenuation Techniques in PMSM Drives-A Survey." In: *IEEE Trans. Ind. Electron.* 64.4 (2017), pp. 3273–3285 (cit. on p. 57).

[66] M. Yang, X. Lang, J. Long, and D. Xu. "Flux Immunity Robust Predictive Current Control with Incremental Model and Extended State Observer for PMSM Drive." In: *IEEE Trans. Power Electron.* 32.12 (2017), pp. 9267–9279. ISSN: 08858993. DOI: 10.1109/TPEL.2017.2654540 (cit. on pp. 57, 100).

[67] F. Toso, M. D. Soricellis, and S. Bolognani. "Simple and Robust Model Predictive Control of PMSM with Moving Horizon Estimator for Disturbance Compensation." In: *The Journal of Engineering* 2019.17 (2019), pp. 4380–4385. URL: https://digital-library.theiet.org/content/journals/10.1049/joe.2018.8052 (cit. on p. 57).

[68] Z. Zhou, C. Xia, Y. Yan, Z. Wang, and T. Shi. "Disturbances Attenuation of Permanent Magnet Synchronous Motor Drives Using Cascaded Predictive-Integral-Resonant Controllers." In: *IEEE Trans. Power Electron.* 33.2 (2018), pp. 1514–1527. ISSN: 08858993. DOI: 10.1109/TPEL.2017.2679126 (cit. on p. 57).

[69] H. T. Nguyen and J. Jung. "Disturbance-Rejection-Based Model Predictive Control: Flexible-Mode Design with a Modulator for Three-Phase Inverters." In: *IEEE Trans. Ind. Electron.* 65.4 (2018), pp. 2893–2903 (cit. on p. 57).

[70] R. Baidya, R. P. Aguilera, P. Acuña, S. Vazquez, and H. d. T. Mouton. "Multistep Model Predictive Control for Cascaded H-Bridge Inverters: Formulation and Analysis." In: *IEEE Transactions on Power Electronics* 33.1 (2018), pp. 876–886. DOI: 10.1109/TPEL.2017.2670567 (cit. on p. 57).

[71] P. Acuna, C. A. Rojas, R. Baidya, R. P. Aguilera, and J. E. Fletcher. "On the Impact of Transients on Multistep Model Predictive Control for Medium-Voltage Drives." In: *IEEE Transactions on Power Electronics* 34.9 (2019), pp. 8342–8355. DOI: 10.1109/TPEL.2018.2889565 (cit. on p. 57).

[72] R. Baidya, R. P. Aguilera, P. Acuña, T. Geyer, R. A. Delgado, D. E. Quevedo, and H. d. T. Mouton. "Enabling Multistep Model Predictive Control for Transient Operation of Power Converters." In: *IEEE Open Journal of the Industrial Electronics Society* 1 (2020), pp. 284–297. DOI: 10.1109/OJIES.2020.3029358 (cit. on p. 57).

[73] G. Pannocchia, M. Gabiccini, and A. Artoni. "Offset-free MPC explained: novelties, subtleties, and applications." In: *IFAC-PapersOnLine* 48.23 (2015), pp. 342 –351. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.11.304 (cit. on pp. 57, 61, 62, 63, 103, 108).

[74] Z. Wang, Z. Zheng, Y. Li, J. Sun, and Z. Deng. "A Robust Offset-Free Model Predictive Current Control for Induction Motor Based on Incremental Model and Incremental Current Observer." In: *2019 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*. 2019, pp. 1–5. DOI: 10.1109/PRECEDE.2019.8753363 (cit. on p. 57).

[75] X. Zhang, L. Zhang, and Y. Zhang. "Model Predictive Current Control for PMSM Drives with Parameter Robustness Improvement." In: *IEEE Trans. Power Electron.* 34.2 (2019), pp. 1645–1657. DOI: 10.1109/TPEL.2018.2835835 (cit. on pp. 57, 100).

[76] G. Pannocchia. "Offset-Free Tracking MPC: A Tutorial Review and Comparison of Different Formulations." In: *2015 Eur. Control Conf. ECC 2015*. Institute of Electrical and Electronics Engineers Inc., 2015, pp. 527–532. ISBN: 9783952426937. DOI: 10.1109/ECC.2015.7330597 (cit. on p. 63).

[77] H. Akagi, Y. Tsukamoto, and A. Nabae. "Analysis and Design of an Active Power Filter Using Quad-Series Voltage Source PWM Converters." In: *IEEE Trans. Ind Appl.* 26.1 (1990), pp. 93–98. DOI: 10.1109/28.52679 (cit. on p. 76).

[78] M. Forgione, D. Piga, and A. Bemporad. "Efficient Calibration of Embedded MPC." In: *Proc. of the 21st IFAC World Congress, Berlin, Germany*. 2020 (cit. on p. 76).

[79] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez. "Model Predictive Control—A Simple and Powerful Method to Control Power Converters." In: *IEEE Trans. on Ind. Electron.* 56.6 (2009), pp. 1826–1838. ISSN: 0278-0046. DOI: 10.1109/TIE.2008.2008349 (cit. on p. 79).

[80] T. Geyer and D. E. Quevedo. "Performance of Multistep Finite Control Set Model Predictive Control for Power Electronics." In: *IEEE Trans. Power Electron.* 30.3 (2015), pp. 1633–1644. DOI: 10.1109/TPEL.2014.2316173 (cit. on p. 79).

[81] F. Wang, X. Mei, J. Rodriguez, and R. Kennel. "Model predictive control for electrical drive systems-an overview." In: *CES Trans. on Electr. Machines and Systems* 1.3 (2017), pp. 219–230. ISSN: 2096-3564. DOI: 10.23919/TEMS.2017.8086100 (cit. on p. 79).

[82] A. A. Ahmed, B. K. Koh, and Y. I. Lee. "A Comparison of Finite Control Set and Continuous Control Set Model Predictive Control Schemes for Speed Control of Induction Motors." In: *IEEE Trans. Ind. Informat.* 14.4 (2018), pp. 1334–1346. DOI: 10.1109/TII.2017.2758393 (cit. on p. 79).

[83] G. Cimini, D. Bernardini, S. Levijoki, and A. Bemporad. "Embedded Model Predictive Control With Certified Real-Time Optimization for Synchronous Motors." In: *IEEE Trans. Control Syst. Technol.* (2020), pp. 1–8. DOI: 10.1109/TCST.2020.2977295 (cit. on pp. 79, 80, 81).

[84] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari. "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs." In: *International Journal of Control* 93.1 (2020), pp. 13–29. DOI: 10.1080/00207179.2017.1316017 (cit. on p. 80).

[85] G. Frison and M. Diehl. *HPIPM: a high-performance quadratic programming framework for model predictive control*. 2020. arXiv: 2003.02547 [math.OC] (cit. on p. 80).

[86] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. "OSQP: an operator splitting solver for quadratic programs." In: *Mathematical Programming Computation* 12.4 (2020), pp. 637–672. DOI: 10.1007/s12532-020-00179-2 (cit. on p. 80).

[87] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. "qpOASES: a parametric active-set algorithm for quadratic programming." In: *Mathematical Programming Computation* 6.1 (2014), pp. 327–363. DOI: https://doi.org/10.1007/s12532-014-0071-1 (cit. on p. 80).

[88] S. Mariethoz, A. Domahidi, and M. Morari. "High-Bandwidth Explicit Model Predictive Control of Electrical Drives." In: *IEEE Trans. Ind Appl.* 48.6 (2012), pp. 1980–1992. DOI: 10.1109/TIA.2012.2226198 (cit. on p. 80).

[89] S. Hanke, O. Wallscheid, and J. Böcker. "Continuous-Control-Set Model Predictive Control with Integrated Modulator in Permanent Magnet Synchronous Motor Applications." In: *2019 IEEE Int. Elect. Machines Drives Conf. (IEMDC)*. 2019, pp. 2210–2216. DOI: 10.1109/IEMDC.2019.8785122 (cit. on pp. 80, 100).

[90] S. Richter, C. N. Jones, and M. Morari. "Computational Complexity Certification for Real-Time MPC With Input Constraints Based on the Fast Gradient Method." In: *IEEE Trans. Autom. Control* 57.6 (2012), pp. 1391–1403. ISSN: 0018-9286. DOI: 10.1109/TAC.2011.2176389 (cit. on p. 80).

[91] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl. "acados: a modular open-source framework for fast embedded optimal control." In: *arXiv* (2020). arXiv: 1910.13753 [math.OC] (cit. on p. 80).

[92] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi. "MATMPC - A MATLAB Based Toolbox for Real-time Nonlinear Model Predictive Control." In: *2019 18th Europ. Control Conf. (ECC)*. 2019, pp. 3365–3370. DOI: 10.23919/ECC.2019.8795788 (cit. on p. 80).

[93] G. Cimini and A. Bemporad. "Exact Complexity Certification of Active-Set Methods for Quadratic Programming." In: *IEEE Trans. on Autom. Control* 62 (2017), pp. 6094–6109. DOI: 10.1109/TAC.2017.2696742 (cit. on p. 81).

[94] D. Goldfarb and A. Idnani. "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs." In: *Math. Program.* 27.1 (Sept. 1983), 1–33. ISSN: 0025-5610. DOI: 10.1007/BF02591962 (cit. on p. 81).

[95] F. Toso, P. G. Carlet, A. Favato, and S. Bolognani. "On-line Continuous Control Set MPC for PMSM drives current loops at high sampling rate using qpOASES." In: *2019 IEEE Energy Conv. Congr. and Exp. (ECCE)*. 2019, pp. 6615–6620. DOI: 10.1109/ECCE.2019.8912838 (cit. on pp. 81, 102, 106).

[96] A. Favato, P. G. Carlet, F. Toso, R. Torchio, and S. Bolognani. "Integral Model Predictive Current Control for Synchronous Motor Drives." In: *IEEE Trans. Power Electron.* (2021), pp. 1–1. DOI: 10.1109/TPEL.2021.3081827 (cit. on pp. 82, 100, 106).

[97] S. Bolognani, R. Kennel, S. Kuehl, and G. Paccagnella. "Speed and current Model Predictive Control of an IPM synchronous motor drive." In: *Int. Electric Machines Drives Conf. (IEMDC)*. 2011. DOI: 10.1109/IEMDC.2011.5994599 (cit. on p. 100).

[98] Q. Wang, G. Zhang, G. Wang, C. Li, and D. Xu. "Offline Parameter Self-Learning Method for General-Purpose PMSM Drives With Estimation Error Compensation." In: *IEEE Trans. Power Electron.* 34.11 (2019), pp. 11103–11115. DOI: 10.1109/TPEL.2019.2900559 (cit. on p. 100).

[99] A. Boglietti, A. Cavagnino, and M. Lazzari. "Experimental High-Frequency Parameter Identification of AC Electrical Motors." In: *IEEE Trans. Ind. Appl.* 43.1 (2007), pp. 23–29. DOI: 10.1109/TIA.2006.887313 (cit. on p. 100).

[100] S. A. Odhano, P. Pescetto, H. A. A. Awan, M. Hinkkanen, G. Pellegrino, and R. Bojoi. "Parameter Identification and Self-Commissioning in AC Motor Drives: A Technology Status Review." In: *IEEE Trans. Power Electron.* 34.4 (Apr. 2019), pp. 3603–3614. ISSN: 1941-0107. DOI: 10.1109/TPEL.2018.2856589 (cit. on pp. 100, 115).

[101] F. Wang, K. Zuo, P. Tao, and J. Rodríguez. "High Performance Model Predictive Control for PMSM by Using Stator Current Mathematical Model Self-Regulation Technique." In: *IEEE Trans. Power Electron.* 35.12 (2020), pp. 13652–13662. DOI: 10.1109/TPEL.2020.2994948 (cit. on p. 100).

[102] X. Liu, L. Zhou, J. Wang, X. Gao, Z. Li, and Z. Zhang. "Robust Predictive Current Control of Permanent-Magnet Synchronous Motors With Newly Designed Cost Function." In: *IEEE Trans. Power Electron.* 35.10 (2020), pp. 10778–10788. DOI: 10.1109/TPEL.2020.2980930 (cit. on p. 100).

[103] M. De Soricellis, D. Da Rù, and S. Bolognani. "A Robust Current Control Based on Proportional-Integral Observers for Permanent Magnet Synchronous Machines." In: *IEEE Trans. on Ind. Appl.* 54.2 (2018), pp. 1437–1447. DOI: 10.1109/TIA.2017.2772171 (cit. on p. 100).

[104] S. Yin, H. Gao, and O. Kaynak. "Data-Driven Control and Process Monitoring for Industrial Applications - Part I." In: *IEEE Trans. Ind. Electron.* 61.11 (2014), pp. 6356–6359. DOI: 10.1109/TIE.2014.2312885 (cit. on p. 100).

[105] A. Brosch, S. Hanke, O. Wallscheid, and J. Böcker. "Data-Driven Recursive Least Squares Estimation for Model Predictive Current Control of Permanent Magnet Synchronous Motors." In: *IEEE Trans. Power Electron.* 36.2 (2021), pp. 2179–2190. DOI: 10.1109/TPEL.2020.3006779 (cit. on pp. 100, 113).

[106] Y. Zhang, J. Jin, and L. Huang. "Model-Free Predictive Current Control of PMSM Drives Based on Extended State Observer Using Ultralocal Model." In: *IEEE Trans. Ind. Electron.* 68.2 (2021), pp. 993–1003. DOI: 10.1109/TIE.2020.2970660 (cit. on p. 100).

[107] F. Tinazzi, P. G. Carlet, S. Bolognani, and M. Zigliotto. "Motor Parameter-Free Predictive Current Control of Synchronous Motors by Recursive Least-Square Self-Commissioning Model." In: *IEEE Trans. Ind. Electron.* 67.11 (2020), pp. 9093–9100. DOI: 10.1109/TIE.2019.2956407 (cit. on p. 100).

[108] S. Aghaei Hashjin, S. Pang, E. H. Miliani, K. Ait-Abderrahim, and B. Nahid-Mobarakeh. "Data-Driven Model-Free Adaptive Current Control of a Wound Rotor Synchronous Machine Drive System." In: *IEEE Trans. Transport. Electrific.* 6.3 (2020), pp. 1146–1156. DOI: 10.1109/TTE.2020.3006722 (cit. on p. 100).

[109] N. A. Losic and L. D. Varga. "A current-free and parameter-free control algorithm." In: *IEEE Trans. Ind. Appl.* 30.2 (1994), pp. 324–332. DOI: 10.1109/28.287527 (cit. on p. 100).

[110] J. B. Jørgensen, J. K. Huusom, and J. B. Rawlings. "Finite horizon MPC for systems in innovation form." In: *50th IEEE Conf. Decision and Control and Eur. Control Conf.* 2011. DOI: 10.1109/CDC.2011.6161509 (cit. on p. 100).

[111] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler. *Quadratic Regularization of Data-Enabled Predictive Control: Theory and Application to Power Converter Experiments*. 2020. arXiv: 2012.04434 [eess.SY] (cit. on p. 101).

[112] P. G. Carlet, A. Favato, S. Bolognani, and F. Dörfler. "Data-driven predictive current control for synchronous motor drives." In: *IEEE Energy Convers. Congr. and Expo. (ECCE)*. 2020. DOI: 10.1109/ECCE44975.2020.9235958 (cit. on p. 101).

[113] B. Huang and R. Kadali. *Dynamic Modeling, Predictive Control and Performance Monitoring: A Data-driven Subspace Approach*. Springer-Verlag London, 2008, p. 242. ISBN: 978-1-84800-232-6. DOI: 10.1007/978-1-84800-233-3 (cit. on p. 105).

[114] F. Dörfler, J. Coulson, and I. Markovsky. *Bridging direct & indirect data-driven control formulations via regularizations and relaxations*. 2021. arXiv: 2101.01273 [math.OC] (cit. on pp. 107, 108).

[115] R. Kadali, B. Huang, and A. Rossiter. "A data driven subspace approach to predictive controller design." In: *Control Engineering Practice* 11.3 (2003), pp. 261–278. ISSN: 0967-0661. DOI: https://doi.org/10.1016/S0967-0661(02)00112-0 (cit. on p. 116).

[116] R. Torchio, L. Codecasa, L. Di Rienzo, and F. Moro. "Fast Uncertainty Quantification in Low Frequency Electromagnetic Problems by an Integral Equation Method Based on Hierarchical Matrix Compression." In: *IEEE Access* 7 (2019), pp. 163919–163932. DOI: 10.1109/ACCESS.2019.2952723 (cit. on p. 124).

[117] L. Codecasa and L. Di Rienzo. "Stochastic Finite Integration Technique for Eddy-Current Problems." In: *IEEE Transactions on Magnetics* 51.3 (2015), pp. 1–4. DOI: 10.1109/TMAG.2014.2356716 (cit. on p. 124).

[118] L. Codecasa and L. Di Rienzo. "Stochastic Finite Integration Technique Formulation for Electrokinetics." In: *IEEE Transactions on Magnetics* 50.2 (2014), pp. 573–576. DOI: 10.1109/TMAG.2013.2280522 (cit. on p. 124).

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Final Version as of September 28, 2021 (`classicthesis` version 4.2).