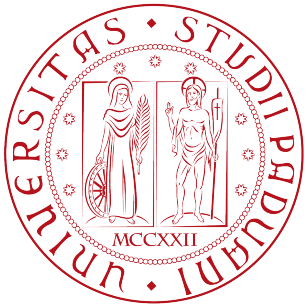
DATA FOR PREDICTIVE CONTROL OF ELECTRIC DRIVES

The Model-Free and the Data-Driven Approaches

PAOLO GHERARDO CARLET



Department of Industrial Engineering
University of Padova

September 2021 – version 2.0

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

University of Padova
Department of Industrial Engineering

Ph.D. Course in
Electrical Energy Engineering

XXXIV CYCLE

DATA FOR PREDICTIVE CONTROL OF ELECTRIC DRIVES

Candidate PAOLO GHERARDO CARLET

Supervisor PROF. SILVERIO BOLOGNANI

Ph.D. School Coordinator PROF. GIULIO ROSATI

Curriculum Coordinator PROF. LUIGI ALBERTI

ABSTRACT

This dissertation includes many research activities in the area of Model Predictive Control (MPC) of Permanent Magnet Synchronous Motor (PMSM) drives. Predictive controllers represent indeed a promising architecture in electric drives applications, because of high dynamic performances, intuitive tuning, possibility to include constraints in the control problem solution and flexibility in the definition of the control objectives.

Two main problems arise for the real-time implementation of the architecture, namely the high computational burden and the parameter sensitivity. Nowadays, the former issue is nearly considered overcome. Improvements in the computational power of the new generation of microprocessors, new hardware such as Field Programmable Gate Array (FPGA)s and advanced computing algorithms are available. The second problem is still open and different approaches are under investigation. This work focuses on those methods based on the exploitation of measured data to enhance the performances of purely model-based designed predictive controllers. The contribution of the thesis is structured in two parts.

The first part deals with the model-free predictive control of synchronous motor drives, in the context of finite-set predictive control. As previously mentioned, one of the main drawbacks of model-based control is the need of an accurate knowledge model of the plant. However, parameters variations or mismatches, as well as of model non-linearity and uncertainties always affect the model parameters identification. A model-free approach allows to avoid such issues by using a non-parametric adaptive model. The key advantage of this method is that the control is self-adapting to any synchronous motor and any operating condition, thus easing the matching between motor and inverter. The current variations predictions are estimated by means of past measurements, stored into look-up-tables. As a further step, the existing look-up-table based parameter-free techniques are overcome. In particular, the past current measurements are manipulated by a recursive least square filter. Finally, Discrete Space Vector Modulation (DSVM) technique is coupled to the model-free approach. Compared to previous parameter-free solutions, the latter modulation strategy allows to achieve a significantly lower current harmonic distortion at the same control frequency, while keeping the switching frequency value at bay.

The second part of the thesis deals with the development of data-driven control methods for the control of electric drives. A possible transition from model-based towards data-driven optimal current

control of synchronous motors is discussed. The discussion starts from a model predictive current control algorithm which uses the parametric model of the motor. Second, the Prediction Error Method (PEM) is analyzed, where a state-space model is identified from data to estimate the motor current dynamics. Moving toward data-driven controls, the Subspace Predictive Control (SPC) is proposed, 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), where raw data are directly exploited by the controller. Some of these techniques are 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. The theory behind data-driven controller is then applied to the disturbance estimation problem, namely the behavioral control theory. A data-driven moving horizon estimator is derived, analyzed and applied to the estimation of load torque disturbances. The design of the cost function behind the estimation problem is deeply discussed.

All the control schemes proposed or presented in the thesis are real-time implemented on embedded hardware. Simulations and experimental results are widely used to validate the research activities.

SOMMARIO

Questo lavoro di tesi include attività di ricerca svolte nell'area del controllo predittivo di azionamenti con motori sincroni a magneti permanenti. I controllori predittivi rappresentano un'architettura promettente, caratterizzata da alte prestazioni dinamiche, facilità di progettazione, dalla facilità nell'inclusione dei vincoli nella soluzione del problema di controllo e dalla flessibilità nella definizione dell'indice di prestazione dell'azionamento.

Tali architetture soffrono di due problematiche fondamentali nella implementazione in tempo reale: l'elevato onere computazionale e la sensitività parametrica. Al giorno d'oggi, il primo problema si può considerare superato, grazie al progresso della potenza di calcolo e allo sviluppo di algoritmi sempre più efficienti. La seconda problematica è tutt'ora aperta. Questa tesi si concentra su algoritmi che mirano a sfruttare le misure raccolte dall'azionamento per migliorare le prestazioni del controllore predittivo. Il lavoro di tesi è stato suddiviso in due parti principali.

La prima parte si focalizza su una tipologia di controllo predittivo noto come *model-free*. Il vantaggio chiave di questo metodo è che l'azionamento è in grado di adattarsi a qualsiasi motore sincrono e a qualsiasi condizione di lavoro. Le correnti future di un motore sincrono sono stimate a partire da dati rilevati in precedenza e memorizzati in tabelle di look-up. In un secondo momento, il precedente controllore *model-free* è stato superato. In particolare, le misure di corrente sono state manipolate attraverso un filtro ricorsivo ai minimi quadrati. Infine, il controllore *model-free* è stato ulteriormente migliorato implementando la tecnica *discrete space vector modulation*. Rispetto alle precedenti soluzioni *model-free*, quest'ultima strategia di modulazione permette di ottenere una distorsione armonica significativamente inferiore nelle correnti a parità di frequenza di controllo, non incrementando eccessivamente il valore della frequenza di commutazione.

La seconda parte della tesi riguarda schemi di controllo denominati *data-driven*, sempre nell'ambito del controllo predittivo di azionamenti elettrici. In particolare, si propone una possibile transizione da un controllo di corrente predittivo progettato unicamente a partire da un modello parametrico, a un controllo basato unicamente basato su dati. Il primo schema considerato è un controllore predittivo di tipo *model-based* standard. In secondo luogo, viene analizzata l'architettura PEM-MPC, nella quale viene identificato un modello in spazio di stato. Adottando un approccio sempre più *data-driven*, viene proposto il SPC, caratterizzato dall'identificazione di un modello ridotto costru-

ito tramite la decomposizione ai valori singolari di opportune matrici contenenti direttamente misure di corrente raccolte dal motore. L'ultimo passo è costituito da un approccio completo basato sui dati, denominato DeePC, in cui le misure raccolte vengono direttamente sfruttate dal controllore. Molte di queste tecniche sono state applicate per la prima volta alla progettazione del controllore di corrente di motori sincroni. Il medesimo approccio *data-driven* è stato successivamente applicato al problema della stima dei disturbi. In particolare si propone una tecnica di progettazione dello stimatore *moving-horizon*, analizzato e applicato alla stima dei disturbi della coppia di carico.

Tutti gli schemi di controllo proposti e descritti in questo lavoro di tesi sono stati implementati in tempo reale. Motivo per cui, tutte le innovative architetture di controllo proposte sono validate da numerosi test sperimentali.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications.

JOURNAL PAPERS

Paolo Gherardo Carlet, Fabio Tinazzi, Silverio Bolognani, and Mauro Zigliotto. "An Effective Model-Free Predictive Current Control for Synchronous Reluctance Motor Drives." In: *IEEE Transactions on Industry Applications* 55.4 (2019), pp. 3781–3790. DOI: 10.1109/TIA.2019.2910494.

Paolo Gherardo Carlet, Fabio Tinazzi, Mauro Zigliotto, and Silverio Bolognani. "Current Ripple Minimisation in Deadbeat Parameter-free Predictive Control of Synchronous Motor Drives." In: *IEEE Open Journal of Industry Applications* (2021), pp. 1–1. DOI: 10.1109/OJIA.2021.3113027.

Andrea Favato, Paolo Gherardo Carlet, Francesco Toso, Riccardo Torchio, and Silverio Bolognani. "Integral Model Predictive Current Control for Synchronous Motor Drives." In: *IEEE Transactions on Power Electronics* (2021), pp. 1–1. DOI: 10.1109/TPEL.2021.3081827.

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 Transactions on Industrial Electronics* (2019), pp. 1–1.

CONFERENCE PAPERS

Matteo Berto, Paolo Gherardo Carlet, Virginia Manzolini, and Luigi Alberti. "An Effective Ellipse Fitting Technique of the Current Response Locus to Rotating HF Voltage Injection in IPMSM for Sensorless Rotor Position Estimation." In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. 2018, pp. 391–396. DOI: 10.1109/IECON.2018.8591855.

S. Bolognani, P. G. Carlet, F. Tinazzi, and M. Zigliotto. "Fast and Robust Model Free Predictive Current Control for SynREL Motor Drives." In: *2018 IEEE ECCE*. 2018, pp. 5466–5472. DOI: 10.1109/ECCE.2018.8558321.

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.

Paolo Gherardo Carlet, Andrea Favato, Francesco Toso, and Silverio 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.

Paolo Gherardo Carlet, Francesco Toso, Andrea Favato, and Silverio 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.

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

Andrea Favato, Paolo Gherardo Carlet, Francesco Toso, and Silverio 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.

Andrea Favato, Francesco Toso, Paolo Gherardo Carlet, Matteo Carbonieri, and Silverio 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.

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.

Francesco Toso, Paolo Gherardo Carlet, Matthias Preindl, and Silverio Bolognani. "Active-Flux-Based Motion-Sensorless Control of PMSM Using Moving Horizon Estimator." In: *2018 IEEE 9th International Symposium on Sensorless Control for Electrical Drives (SLED)*. 2018, pp. 78–83. DOI: 10.1109/SLED.2018.8486107.

Francesco Toso, Milo De Soricellis, Paolo Gherardo Carlet, Andrea Favato, and Silverio 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.

Francesco Toso, Riccardo Torchio, Andrea Favato, Paolo Gherardo Carlet, Silverio Bolognani, and Piergiorgio 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.

SUBMITTED PAPERS

Paolo Gherardo Carlet, Andrea Favato, Saverio Bolognani, and Florian Dorfler. "Data-Driven Continuous-Set Predictive CurrentControl for Synchronous Motor Drives." In: *IEEE Transactions on Power Electronics* (2021). Submitted.

Andrea Favato, Paolo Gherardo Carlet, Francesco Toso, Riccardo Torchio, Mattia Bruschetta, Ruggero Carli, Bolognani Silverio, and Jose Rodriguez. "Fast Solver for Implicit Continuous Set ModelPredictive Control of Electric Drives." In: *IEEE Transactions on Industrial Informatics* (2021). Submitted.

ACKNOWLEDGMENTS

I am immensely grateful to Prof. Silverio Bolognani, supervisor of my work. He was always open to face new problems and to engage stimulating discussions. I also want to thank him for the opportunity I had to visit the Automatic Control Laboratory of ETH in Zurich, which was a breakthrough experience during my Ph. D.

I would like to express my best gratitude to Mosè Castiello, technician of the Electric Drive Laboratory of Padova. He was always helpful to set-up the test benches, to solve software and hardware troubles and to explain complex practical issues in simple and concepts.

I express my gratitude to Prof. Florian Dorfler and Saverio Bolognani, who guided me during my visiting period at ETH. The discussions with them have left me with valuable technical advice and new ideas. I thank all the supporters and contacts I had through my experience, I personally believe that such a stimulating context is hard to find.

Many thanks go to all the people I have met in EDLab many of whom I consider first as friends, rather than colleagues. Among them, I would especially thank Andrea Favato and Francesco Toso.

Lastly, but definitely not the least, I am immensely grateful to my family who have always believed in me in all the decisions I have made and have supported me.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

LTI Linear Time Invariant

MPC Model Predictive Control

LQR Linear Quadratic Regulator

QP Quadratic Programming

NP Nondeterministic Polynomial time

ARX Auto Regressive with eXogenous inputs

RLS Recursive Least Squares

SVD Singular Value Decomposition

LUT Look-Up Table

DSP Digital Signal Processor

I/O Input/Output

FPGA Field Programmable Gate Array

FS Finite Set

CS Continuous Set

PWM Pulse Width Modulation

IGBT Insulated Gate Bipolar Transistor

MOSFET Metal Oxide Semicondutor Field Effect Transistor

DSVM Discrete Space Vector Modulation

THD Total Harmonic Distortion

EMF Electro Motive Force

PMSM Permanent Magnet Synchronous Motor

SPM Surface Permanent Magnet

IPM Interior Permanent Magnet

PMAREL Permamente Magnet Assisted Reluctance

SyR Synchronous Reluctance

MTPA Maximum-Torque-per-Ampere

PEM Prediction Error Method

SPC Subspace Predictive Control

DeePC Data-EnablEd Predictive Control

MB Model-Based

MF Model-Free

TDD Total Demand Distortion

1 INTRODUCTION

Climate change and environmental degradation are two of the most important challenges of this century. The European Union accepted these challenges in the 2019, announcing the ambitious *European Green Deal*. Adopting this strategy, the European Union aims to become the first climate-neutral continent within the 2050. Socio-economic consequences of a such innovative strategy are unavoidable.

COVID-19 pandemic has probably accelerated the timings of this long-term transition. In fact, the economic recovery after the pandemic represents a stimulating chance to transform European economy towards a resource-efficient, modern and competitive one. For this reason, one third og the 1.8 trillion euro investments from the European Recovery Plan, namely the *NextGenerationEU* plan, and the EU's seven-year budget will finance the Green Deal.

Two of the emerging policies driving the *NextGenerationEU* plan are the *Make it Green* and the *Make it Digital*. The former policy aims to the climate-neutral continent. Innovations of the public a private transport, energy-efficient buildings and renewable energy are the leverages to realize this scenario. The latter policy provides a further tool for the climate change challenge, which is the digitalization. Artificial intelligence will assume a key-role in the reduction of the human footprint. For instance, big-data and their smart usage could further reduce the impact of the transport, industry and buildings consumption.

Research and innovation are essentials in the realization of the Green Deal. Thus, the European funding program *Horizon Europe* finances both research and innovations with a budget of 95.5 billion of euro within the 2027. One third of the budget is dedicated to the clusters digital-industry-space and climate-energy-mobility. The areas of intervention of the funding campaign includes artificial intelligence, advanced computing and big-data, energy supply and storage and smart, electrified and efficient mobility.

Electric motors are an enabling technology in the climate change challenge. In fact, electric machines are present in almost every strategic field, such as transport, generation from renewable energy sources, industry and buildings. Generation from renewable energy is often enabled by electric generators, for instance the wind-turbines generation. Moreover, electrification of the transports requires again electric motors. For these reasons, it is evident that improvements in the electric motor technology have consequence in the efficiency improvement of the entire chain.

High efficiency and high performances are met by PMSMs. These machines have high torque and power density, specifications that are fundamentals also from the commercial point of view. Performances are strongly influenced by the materials adopted for the permanent magnets. Rare-earth magnets guarantee the highest torque and power density, such as Neodymium based magnets. Unfortunately, price and performances do not meet, since rare-earth materials and their production is almost a Chinese monopoly (almost 80% of the production). This means also that the price fluctuates significantly. Thus, Synchronous Reluctance (SyR) motors are becoming more and more popular in the industry. These machines are designed without permanent magnets, reducing the overall price. However, lower torque and power density are achieved, limiting the fields of application of such motor technology.

Further improvements of the electro-mechanic conversion efficiency performed by PMSM can be achieved both by means of design optimization and improvements of the control strategies. On one hand, advanced computing and artificial intelligence help the optimization at the design stage. Differential evolution methods [83] can be used to accomplish the design specifications, finding the best compromise for the application. On the other hand, advanced control strategies are available nowadays to improve the performances of the electric motor drive, without acting on the machine design. This solution is often of particular interest for electric motor manufacturers. In fact, the power converter that control the motor could be manufactured by a different company.

MPC have been investigated for two decades as advanced control method both for grid-connected converter and synchronous motor drives. MPC is an optimization based technique used in regulation or tracking control problems. 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. This method has already found effective industrial application in the first field [74], i. e. grid-connected converters. In this area, in fact, the efficiency of the converter can be drastically enhanced. MPC represents indeed an example of advanced computing for the application. In fact, MPC was designed for slowly-evolving systems, such as chemical processes, and not for real-time embedded fast applications, such as the control of electric motor drives. The difference of time scales is about hundreds order of magnitudes, from seconds or minutes to milli-seconds or hundreds of micro-seconds.

Research on theoretical aspects related to MPC is steadily growing, pushed by the new interest in data analysis [78]. One of the most promising aspects under investigation regards the exploitation of measured data to further improve the performances of MPCs. These aspects are of great interest also for the electric motor drives. Exploit-

ing data measured from PMSM opens the possibility of designing new smarter and data-driven controllers. In particular, two main strands are found, depending on the how data are exploited in the MPC framework. On one hand, data could be used to build adaptive controllers. Such control type is becoming popular in the electric drive area as *model-free* [46] methods. The intuition behind these methods is that data are capable to predict the fast electric drives dynamics even more accurately than analytical models. It is reminded that the computational time bottleneck is fundamental in embedded applications, limiting the complexity of analytical models. Thus, the model-free paradigm represents a promising alternative to the parametric analytical paradigm. On the other hand, data could be exploited to commission an offline data-driven model [18]. The offline commissioning solves the time limitations issue. Thus, even advanced machine learning methods are available for the model commissioning [72].

This thesis contains several contributions in the field of model-free and data-driven control of electric drives. In fact, the objective pursued in the research activity is to improve the performance of model predictive control in the electric drives area, through the effective usage of data collected from the plants. The performances and efficiency of the overall drive are considered as essential features, which not always meet. Performances are often what the market ask for. However, efficiency is becoming more and more relevant, pushed also by renewed social attention on the climate challenge highlighted by the European guidelines. Academics and researchers in the industrial engineering area are asked to find a road in there. This dissertation wants to provide some humble contributions towards smarter and more data-driven control of electric drives. Moreover, the author hopes to stimulate discussions and to promotes interests in other future research projects.

1.1 OUTLINE OF THE THESIS

Hereinafter the dissertation structure is briefly presented.

PART I presents the theoretical foundation and the mathematical background. The part is divided in three chapters:

- Chapter 2 presents the Permanent Magnet Synchronous motor. Mathematical model of the motor is derived including some non linear phenomena, such as the iron saturation.
- Chapter 3 provides a brief presentation of Model Predictive Control, with particular focus on the power electronics area. In particular, Finite Set and Continuous Set strategies are discussed.

- Chapter 4 presents some ideas and theorems of the behavioral system theory, which is widely exploited in data-driven control.

PART II resumes the contributions in the area of model-free control of electric drives and it is structured in three main chapters.

- Chapter 5 presents some relevant contributions concerning the real-time implementation of model-free controls, with particular focus on the stagnation problem.
- Chapter 6 illustrates an innovative parameter-free method which exploits recursive least square estimators to build an adaptive model-free controller.
- Chapter 7 discusses a further contribution to the parameter-free solution, improving the previous technique. In a nutshell, the model-free paradigm and the discrete space vector modulation are coupled. This permits to improve the performances of the drive, reducing both the Joule losses and the switching frequency.

PART III reports the contributions in the sphere of data-driven control of electric drives, which is divided in two parts.

- Chapter 8 discusses the novel data-driven control paradigm. The electric drives field is one of the first area of application of this control methods. The technique is discussed providing also some design guidelines.
- Chapter 9 presents the discussion of a new disturbance estimator based on the behavioral system theory, similar to a data-driven moving horizon estimator. The estimator equations are derived, then it is applied to the problem of estimating load torques disturbances.

Part I

BACKGROUND KNOWLEDGE

The models behind the essential components of an electric drive are developed, focusing on synchronous motors and the power electronics. Some of the non-linear phenomena that affects the control of these plants are presented and included in the modeling stage, further complicating the model of the drive. This thesis presents predictive control techinques which goes to an opposite direction. In details, in the following parts simple adaptive models are presented, exploiting measured data to take into account of all the non linearities. In this part a brief description of model predictive control is provided, focusing on the power electronics applications and on the concepts of finite-set and continuous set methods. Finally, some concept regarding the behavioral system theory are described in order to ease the understanding of the data-drive control paradigm.

2

SYNCHRONOUS MOTORS ELECTRIC MODEL

PMSMs are AC machines characterized by a phase stator windings, usually star-connected, displaced $\frac{2\pi}{3}$ electrical radiants in space. The stator windings consist of individual coils connected and wounded in the stator slots, in order to approximate a sinusoidal distribution along the stator circumference. This sinusoidal distribution can be replicated around the circumference for several times, depending on the motor pole-pairs number p. The rotor is separated from the stator by the so called airgap. Permanent magnets are placed in the rotor, producing a flux which do not depends on the stator current. The periodicity of the magnets around the rotor circumference matches the stator number of pole-pairs number. Magnets materials depend on the torque density that need to be achieve. Rare-earth magnets, such as Neodymium of Samarium-Cobalt, are the one that guarantee the highest performances, achieving also the most compact designs. However, the cost of these materials is quite high. Thus, even magnet with lower performance or even pure SyR are of interest in many industrial applications. The interaction between the magnet flux and the armature current flux produces the motor torque.

PMSM are usually distinguished on the basis of the rotor geometry. Figure 2.2 reports some of the most relevant categories. Surface Permanent Magnet (SPM) machines are characterized by a magnets displacement along the rotor surface Figure 2.2a. Since the magnetic relative permeability of the magnets is similar to the one of the air, the magnetic behavior of the rotor is isotropic, i. e. the flux finds an uniform iron path along the rotor circumference. In Interior Permanent Magnet (IPM) motors or Permamente Magnet Assisted Reluctance (PMAREL) motors magnets are inserted inside the rotor iron (Figure 2.2b), resulting in an anisotropic magnetic behavior. In fact, the magnetic reluctance seen by the stator depends on the rotor position, since the stator armature-current produce a flux which finds different iron paths depending on the rotor position. The machine presents a so called saliency or a reluctance effect, which can add a beneficial contribution to the total torque. An extreme example of anisotropic topology is the one of the SyR machines, as shown in Figure 2.2c. No magnets are mounted on the rotor and the motor torque is entirely produced by the reluctance effect.

In the following subsection a model is derived for permanent magnet synchronous machines. In the modeling it is assumed a slot-less sinusoidal stator winding distribution and a sinusoidally distributed magnetic field at the airgap. This permits a simple model of the ma-

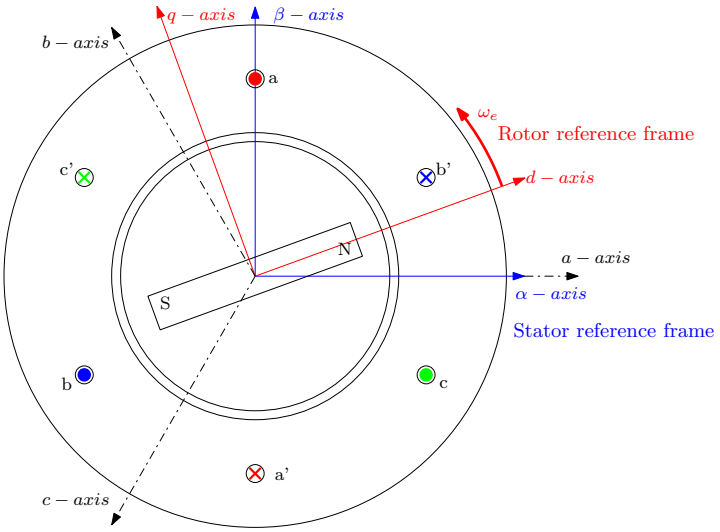
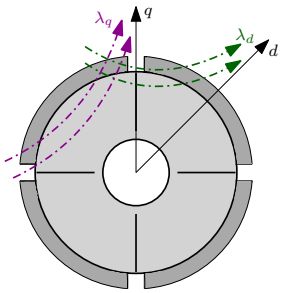


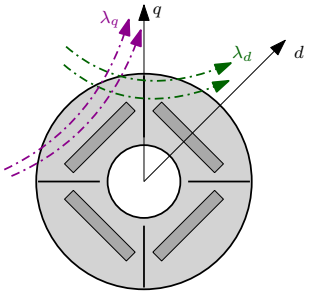
Figure 2.1: Overview of the stator and rotor reference frames commonly used to represent PMSM: one pole-pair machine.

chine, which results suitable for the application of many control architectures, included optimization-based methods, e. g. MPC. Before going throw more formal details, it is worth listing the three fundamental reference frame that are considered for deriving the motor models (Figure 2.1):

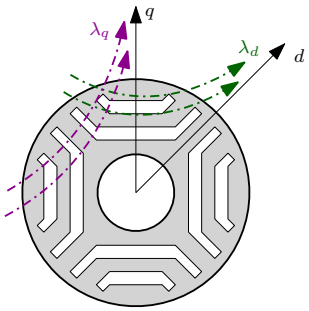
- the stationary abc reference frame is the one defined by the stator windings coils, and it is depicted by the dashed lines in the figure;
- the stationary $\alpha\beta$ system, composed by two orthogonal axes and aligned along the phase a axis (depicted with solid blue lines);
- the rotating dq reference frame, aligned along the permanent magnet flux and synchronous with it, depicted by solid red lines. The d-axis angle is denoted as $\theta_e = p\theta_m$, being the θ_m the rotor position with respect to the a-axis. Moreover, the same axis rotates at an angular speed $\omega_e = p\omega_m$, being the ω_m the angular rotor speed. It is worth highlighting that the electric speed and position are related to the periodicity of the electric quantities. This clarifies why they are influenced by the pole-pairs number.



(a) SPM motor.



(b) IPM motor.



(c) SyR motor.

Figure 2.2: PMSM topologies.

2.1 VOLTAGE BALANCE: IDEAL MAGNETIC MODEL

The motor voltage balance equation of each phase is composed by the resistive voltage drop and the contribution of the flux linkage derivative [48]:

$$\begin{aligned} u_a &= R_s i_a + \frac{d\lambda_a}{dt} \\ u_b &= R_s i_b + \frac{d\lambda_b}{dt}, \\ u_c &= R_s i_c + \frac{d\lambda_c}{dt} \end{aligned} \tag{2.1}$$

where u_a,u_b and u_c are the terminal voltages, i_a,i_b and i_c are the phase currents and λ_a,λ_b and λ_c the stator flux linkages. The time dependence of all these signals is neglected here to keep a compact and simple notation. Moreover, the stator windings resistance R_s is assumed equal for each phase.

The voltage induced by the stator flux derivative has two main contributions. In particular, the first contribution is due to the fluxes produced by the stator currents, whereas the second one is due to the fluxes produced by the permanent magnets mounted on the rotor, if present:

$$\begin{aligned} \lambda_a &= L_a i_a + M_{ab} i_b + M_{ac} i_c + \lambda_{pm,a} \\ \lambda_b &= L_b i_b + M_{ba} i_a + M_{bc} i_c + \lambda_{pm,b}. \\ \lambda_c &= L_c i_c + M_{ca} i_a + M_{cb} i_b + \lambda_{pm,c} \end{aligned} \tag{2.2}$$

L_a, L_b and L_c are the phase self-inductances, whereas $M_{ab} = M_{ba}$, $M_{bc} = M_{cb}$ and $M_{ac} = M_{ca}$ are the mutual inductances between the phases. It is remarked that in this section the iron saturation phenomena is neglected.

Self and mutual inductances, as well as the permanent magnet flux contributions are position dependent quantities and they have a periodicity related to the electric angle θ_e. In order to be more clear, the position dependency is provided for the permanent magnet flux terms:

$$\begin{aligned} \lambda_{pm,a} &= \Lambda_{pm}\cos(\theta_e) \\ \lambda_{pm,b} &= \Lambda_{pm}\cos(\theta_e - \frac{2}{3}\pi) \\ \lambda_{pm,c} &= \Lambda_{pm}\cos(\theta_e - \frac{4}{3}\pi) \end{aligned}, \tag{2.3}$$

where Λ_{pm} is the amplitude of the permanent magnet flux-linkage. The value of Λ_{pm} is influenced by many factors, such as the rotor geometry, the materials, the airgap thickness and the number of turns in the stator windings.

Equation (2.1) can be written in a compact matrix form. Matrix notation is widely used in this work since it results particularly suitable for the implementation of MPC algorithms. First it is worth defining the back-Electro Motive Force (EMF) along the three stator axes, denoted as e_a, e_b and e_c:

$$\begin{aligned} e_a &= \frac{d\lambda_{pm,a}}{dt} = -\Lambda_{pm}\omega_e \sin(\theta_e) \\ e_b &= \frac{d\lambda_{pm,b}}{dt} = -\Lambda_{pm}\omega_e \sin(\theta_e - \frac{2}{3}\pi) \\ e_c &= \frac{d\lambda_{pm,c}}{dt} = -\Lambda_{pm}\omega_e \sin(\theta_e - \frac{4}{3}\pi) \end{aligned}. \tag{2.4}$$

Thanks to this definition, the compact matrix form of the voltage balance equation 2.2 results:

$$\mathbf{u}_{abc} = R_s\mathbf{i}_{abc} + \frac{d\boldsymbol{\lambda}_{abc}}{dt} = R_s\mathbf{i}_{abc} + \frac{d\mathbf{L}_{abc}\mathbf{i}_{abc}}{dt} + \boldsymbol{e}_{abc}. \tag{2.5}$$

All the phase signals are condensed in vectors:

$$\mathbf{u}_{abc} = \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}, \quad \mathbf{i}_{abc} = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}, \quad \boldsymbol{\lambda}_{abc} = \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix}, \quad \boldsymbol{e}_{abc} = \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}, \tag{2.6}$$

whereas self and mutual inductances are stacked in the symmetric matrix:

$$\mathbf{L}_{abc} = \begin{bmatrix} L_a & M_{ab} & M_{ac} \\ M_{ba} & L_b & M_{bc} \\ M_{ca} & M_{cb} & L_c \end{bmatrix}. \tag{2.7}$$

Three-phase systems are often represented in the $\alpha\beta0$ reference frame. The essential details about the transformation from abc to $\alpha\beta0$ frames can be found in AppendixA.1. Since PMSM are balanced three-phase systems, the homopolar component (A.1) is null, thus neglected. The simplified Clarke transformation matrix (A.2) is applied to (2.5), obtaining:

$$\mathbf{u}_{\alpha\beta} = R_s \mathbf{i}_{\alpha\beta} + \frac{d\boldsymbol{\lambda}_{\alpha\beta}}{dt} = R_s \mathbf{i}_{\alpha\beta} + \frac{d\mathbf{L}_{\alpha\beta}\mathbf{i}_{\alpha\beta}}{dt} + \boldsymbol{e}_{\alpha\beta}, \tag{2.8}$$

where $\mathbf{L}_{\alpha\beta} = \mathbf{T}_{\alpha\beta}\mathbf{L}_{abc}\mathbf{T}_{\alpha\beta}^{T}$ and

$$\mathbf{u}_{\alpha\beta} = \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix}, \quad \mathbf{i}_{\alpha\beta} = \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}, \quad \boldsymbol{\lambda}_{\alpha\beta} = \begin{bmatrix} \lambda_\alpha \\ \lambda_\beta \end{bmatrix}, \quad \boldsymbol{e}_{\alpha\beta} = \begin{bmatrix} e_\alpha \\ e_\beta \end{bmatrix}. \tag{2.9}$$

It is noticed that the term $R_s \mathbf{i}_{\alpha\beta}$ should be interpreted as an element-wise product. In other words, it represents a diagonal matrix, having all the diagonal elements equal to the stator resistance R_s.

Finally, the Park transformation (A.4) is applied to (2.8). This simplifies all the position dependent parameters, i. e. inductances and back-EMFs, to constants:

$$\mathbf{u}_{dq} = R_s \mathbf{i}_{dq} + \frac{d\boldsymbol{\lambda}_{dq}}{dt} + \omega_e \boldsymbol{\lambda}_{dq} = R_s \mathbf{i}_{dq} + L_{dq}\frac{d\mathbf{i}_{dq}}{dt} + \omega_e \mathbf{J}\mathbf{L}_{dq}\mathbf{i}_{dq} + \boldsymbol{e}_{dq}, \tag{2.10}$$

where the dq vectors and matrices are:

$$\mathbf{u}_{dq} = \begin{bmatrix} u_d \\ u_q \end{bmatrix}, \quad \mathbf{i}_{dq} = \begin{bmatrix} i_d \\ i_q \end{bmatrix}, \quad \boldsymbol{\lambda}_{dq} = \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix}, \quad \boldsymbol{e}_{dq} = \begin{bmatrix} 0 \\ \omega_e \Lambda_{pm} \end{bmatrix}. \tag{2.11}$$

When transforming quantities in the rotating reference frame, an additional skew-symmetric matrix

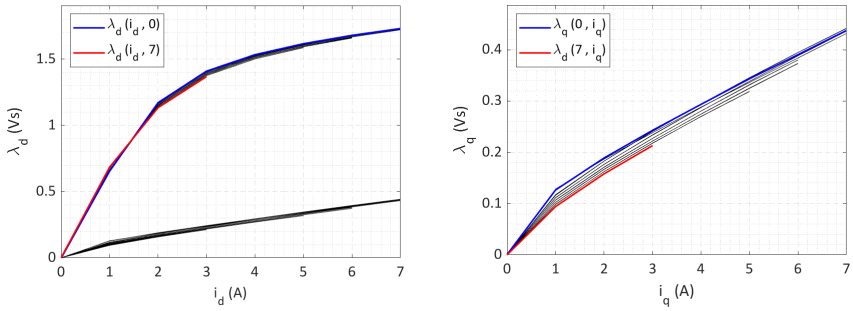
$$\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.12}$$

appears in the equation, because of the Park operator derivative (A.6). Even if motional cross-coupling terms appear in the model, the inductance matrix becomes simply:

$$\mathbf{L}_{dq} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix}. \tag{2.13}$$

Depending on the motor topology, the two inductances can be equal, e. g. SPM machines, or different, e. g. IPM, PMAREL and SyR motors. For sake of completeness, the matrix equation (2.10) is decomposed in the two d and q components:

$$\begin{aligned} u_d &= R_s i_d + L_d \frac{di_d}{dt} - \omega_e L_q i_q \\ u_q &= R_s i_q + L_q \frac{di_q}{dt} + \omega_e L_d i_d + \omega_e \Lambda_{pm} \end{aligned} \tag{2.14}$$



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

Figure 2.3: Example of non-linear flux-current curves: SyR motor case.

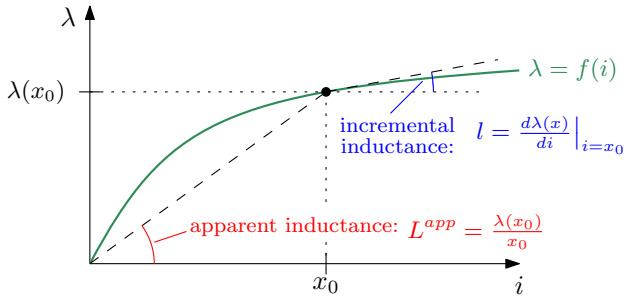


Figure 2.4: Difference between apparent and differential inductances.

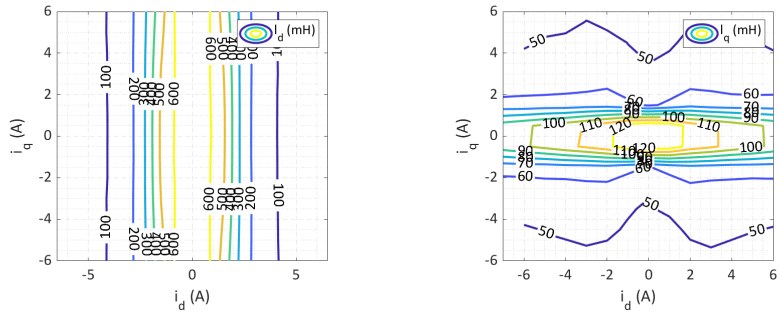
The proposed simple linear model is particularly suitable for isotropic SPM and IPM, characterized by an almost linear magnetic saturation curve. Its simplicity is one of the main benefits for the control design purpose. On the opposite, the model could be very rough for SyR machines (see Figure 2.3), for which the iron-saturation phenomena should be considered for the current regulators tuning.

2.2 VOLTAGE BALANCE INCLUDING THE FLUX NON-LINEARITY

A dq model is presented in this section, suitable for highly saturated motors, such as SyR machines. The procedure that brings the voltage balance equation from the three-phase reference frame to the rotating one is similar to the one described in the previous section. However, being the flux-current relationships non linear it is hard to define self and mutual inductances. For this reason, a slightly different analysis is required.

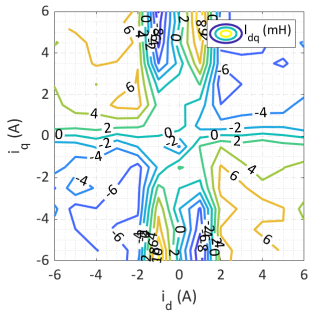
It is useful to rewrite (2.10), highlighting the introduced non linearity in the flux-linkages expressions:

$$\mathbf{u}_{dq} = R_s \mathbf{i}_{dq} + \frac{d\boldsymbol{\lambda}_{dq}(\mathbf{i}_{dq})}{dt} + \omega_e \boldsymbol{\lambda}_{dq}(\mathbf{i}_{dq}) \tag{2.15}$$



(a) l_d

(b) l_q



(c) l_{dq}

Figure 2.5: Example of incremental inductances maps: SyR motor case.

The chain rule is applied to this expression in order to extract the dependency on the motor currents:

$$\frac{df(x_1, x_2, \ldots, x_n)}{dt} = \frac{\delta f}{\delta x_1}\Big|_{x_1 = x_1^0} \frac{\delta x_1}{\delta t} + \cdots + \frac{\delta f}{\delta x_n}\Big|_{x_n = x_n^0} \frac{\delta x_n}{\delta t} \tag{2.16}$$

Applying the chain rule to the voltage balance equation, it is obtained:

$$\frac{d\boldsymbol{\lambda}_{dq}}{dt} = \mathbf{f}(\mathbf{i}_{dq}) \approx \boldsymbol{\lambda}_{dq}(\mathbf{i}_{dq})\Big|_{\mathbf{i}_{dq} = \mathbf{x}_0} + \frac{\delta \boldsymbol{\lambda}_{dq}}{\delta \mathbf{i}_{dq}}\Big|_{\mathbf{i}_{dq} = \mathbf{x}_0} \frac{\delta \mathbf{i}_{dq}}{\delta t} \tag{2.17}$$

In the electric drives field, the Jacobian, i. e. the matrix containing the partial derivatives of the fluxes with respect to the current, hereinafter is denoted as $\mathbf{l}_{dq}$. It is known also as the matrix of the self and cross-differential inductances:

$$\begin{aligned} l_d(\mathbf{i}_{dq}) &= \frac{d\lambda_d}{di_d}, \quad l_{dq}(\mathbf{i}_{dq}) = \frac{d\lambda_d}{di_q} \\ l_{qd}(\mathbf{i}_{dq}) &= \frac{d\lambda_q}{di_d}, \quad l_q(\mathbf{i}_{dq}) = \frac{d\lambda_q}{di_q} \end{aligned} \tag{2.18}$$

Moreover, the linearized flux $\boldsymbol{\lambda}_{dq}(\mathbf{i}_{dq})\Big|_{\mathbf{i}_{dq}}$ is often written as function of the apparent inductances, defined as:

$$\begin{aligned} L_d^{app}(\mathbf{i}_{dq}) &= \frac{\lambda_d(\mathbf{i}_{dq})}{i_d}, \quad L_{dq}^{app}(\mathbf{i}_{dq}) = \frac{\lambda_d(\mathbf{i}_{dq})}{i_q} \\ L_{qd}^{app}(\mathbf{i}_{dq}) &= \frac{\lambda_q(\mathbf{i}_{dq})}{i_d}, \quad L_q^{app}(\mathbf{i}_{dq}) = \frac{\lambda_q(\mathbf{i}_{dq})}{i_q} \end{aligned}. \tag{2.19}$$

In the following, the matrix of apparent inductances is denoted as $\mathbf{L}_{dq}^{app}(\mathbf{i}_{dq})$. Figure 2.4 provide a graphical interpretation of the difference between the concepts of differential and incremental inductances. Figure 2.5 shows an example of incremental inductances maps for a SyR motor.

The dq voltage balance model, accounting of the non-linear magnetic characteristics, results:

$$\mathbf{u}_{dq} = R_s\mathbf{i}_{dq} + \mathbf{l}_{dq}(\mathbf{i}_{dq})\frac{d\mathbf{i}_{dq}}{dt} + \omega_e L_{dq}^{app}(\mathbf{i}_{dq})\mathbf{i}_{dq} \tag{2.20}$$

2.3 STATE-SPACE REPRESENTATION

The state-space representation is probably the most widespread framework to represent dynamical systems in the modern control engineering. Moreover, the MPC adopts the discrete-time version of these representations for the controller synthesis. As a consequence, the continuous-time and discrete-time state-space model of the voltage balance equation (2.10) are derived. A state-space representation of the non-linear model can be obtained following the same steps described in this subsection for the linear one.

According to the state-space framework, Linear Time Invariant (LTI) system dynamics are described by n states $\mathbf{x}(t)$, i. e. variables which fully describe the behavior of a plant. These plant states are not always fully available and measurable. The subset of available signals is named output $\mathbf{y}(t)$, which has dimension $p \leqslant n$. The control acts on the system through m inputs $\mathbf{u}(t)$. States dynamics are described by:

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}_c\mathbf{x}(t) \end{aligned}, \tag{2.21}$$

where $\mathbf{A}_c$ is the state matrix, $\mathbf{B}_c$ is the input matrix and $\mathbf{C}_c$ is the output matrix.

The PMSM voltage balance equation (2.10) can be easily written using this formalism, by posing:

$$\mathbf{A}_c = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_e(t)\frac{L_q}{L_d} \\ -\omega_e(t)\frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \tag{2.22}$$

and rearranging (2.10) in

$$\frac{d\mathbf{i}_{dq}(t)}{dt} = \mathbf{A}_c(t) + \mathbf{B}_c\mathbf{u}_{dq}(t) + \mathbf{B}_c\mathbf{e}_{dq}(t). \tag{2.23}$$

It is worth noticing that the state matrix is time variant, since it depends on the operating speed. Thus, the system results linear parameter varying, in the parameter ω_e. Moreover, the back-EMF can be interpreted as a time variant disturbance acting on the plant. The states of the PMSM voltage balance equation are the dq current, which coincide to the outputs. Finally, the input is the vector of dq voltages. Other solutions can be found in literature, e. g. the motor flux linkages can be assumed as states.

Modern drive systems are almost exclusively implemented on a digital hardware with constant sampling interval, denoted T_s. Several techniques are available for discretizing model (2.23), starting from Euler methods, to the Tustin methods or even exact discretization. The discretized form of (2.21) results

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \tag{2.24}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the discrete form of $\mathbf{A}_c$ and $\mathbf{B}_c$, respectively. Their expressions depend on the adopted discretization technique.

The computational effort is the balance needle, when choosing the discretization method. Exact expressions require the computation of matrix exponentials and integrals [36]. The expressions are here reported, included the full computation for $\mathbf{A}$ as example:

$$\mathbf{A} = e^{\mathbf{A}_c T_s} = e^{-T_s \frac{R_s L_\Delta}{L_d L_q}} \begin{bmatrix} c_1 - \frac{R_s L_\Sigma}{L_d L_q} s_1 & \omega_e s_1 \\ -\omega_e s_1 & c_1 + \frac{R_s L_\Sigma}{L_d L_q} s_1 \end{bmatrix}. \tag{2.25}$$

$$\mathbf{B} = \left(\int_0^{T_s} e^{\mathbf{A}_c \tau} d\tau \right) \mathbf{B}_c$$

being L_Σ and the L_Δ the average and the semi-sum of the inductances L_d and L_q, $c_1 = \cosh(\lambda T_s)$, $c_2 = \frac{\sinh(\lambda T_s)}{\lambda}$ and $\lambda = \sqrt{(\frac{R_s L_\Sigma}{L_d L_q})^2 - \omega_e^2}$. The full expression of $\mathbf{B}$ is even more complicated and it can be found in [36]. For this reason, the exact method is rarely adopted in drives applications, even if new FPGA hardware unlocks these computations.

Taylor approximations of the exact discretization process reach often the desired accuracy. Since the sampling rate is relatively high with respect to the dynamics, first order truncation is sufficient. The forward Euler difference is indeed the most widespread solution and also the most adopted for MPC architectures. According to this method, current derivatives are approximated as

$$\frac{d\mathbf{i}_{dq}}{dt} \approx \frac{\mathbf{i}_{dq}(k+1) - \mathbf{i}_{dq}(k)}{T_s}. \tag{2.26}$$

The resulting discrete-time state-space model of the voltage balance equation is:

$$\mathbf{i}_{dq}(k+1) = \mathbf{A}(k)\mathbf{i}_{dq} + \mathbf{B}\mathbf{u}_{dq}(k) + \mathbf{B}\mathbf{e}_{dq}(k). \tag{2.27}$$

where

$$\mathbf{A} = \mathbf{I} + T_s\mathbf{A}_c = \begin{bmatrix} 1 - T_s\dfrac{R_s}{L_d} & T_s\omega_e(k)\dfrac{L_q}{L_d} \\ -T_s\omega_e(k)\dfrac{L_d}{L_q} & 1 - T_s\dfrac{R_s}{L_q} \end{bmatrix}, \mathbf{B} = T_s\mathbf{B}_c = \begin{bmatrix} \dfrac{T_s}{L_d} & 0 \\ 0 & \dfrac{T_s}{L_q} \end{bmatrix}. \tag{2.28}$$

2.4 ELECTROMAGNETIC TORQUE AND MECHANICAL EQUATION

As a final step in the motor modeling description, the electromagnetic torque expression is derived. The non linear magnetic model is considered for sake of generality. In addition, the torque expression will be exploited in a simple model that approximate the dynamic of the electro-mechanical energy conversion.

The active power absorbed by the motor is computed as:

$$p = \frac{3}{2}\mathbf{u}_{dq}^T\mathbf{i}_{dq} \tag{2.29}$$

It is reminded that the factor 3/2 is due to the fact that the Clarke transformation is not power invariant (see AppendixA.1). Substituting (2.15) in the just presented equation, it is obtained:

$$\begin{aligned} p &= \frac{3}{2}\left(R_s\mathbf{i}_{dq} + \frac{d\boldsymbol{\lambda}_{dq}}{dt} + \omega_e\boldsymbol{\lambda}_{dq}\right)^T\mathbf{i}_{dq} \\ &= \frac{3}{2}R_s\mathbf{i}_{dq}^T\mathbf{i}_{dq} + \frac{3}{2}\left(\frac{d\boldsymbol{\lambda}_{dq}}{dt}\right)^T\mathbf{i}_{dq} + \frac{3}{2}p\omega_m\boldsymbol{\lambda}_{dq}^T\mathbf{i}_{dq} \end{aligned} \tag{2.30}$$

Assuming that the flux $\boldsymbol{\lambda}_{dq}$ derivative is null (which is coherent with the simplified modeling of the motor presented), the power expression contains two terms:

- the Joule losses produced in the motor windings $\frac{3}{2}R_s\mathbf{i}_{dq}^T\mathbf{i}_{dq}$;
- the electromagnetic power $\frac{3}{2}p\omega_m\boldsymbol{\lambda}_{dq}^T\mathbf{i}_{dq}$.

The PMSM torque is computed by dividing the electromagnetic power by the mechanical speed:

$$\tau = \frac{3}{2}p\boldsymbol{\lambda}_{dq}^T\mathbf{i}_{dq} = \frac{3}{2}p(\lambda_d i_q - \lambda_q i_d). \tag{2.31}$$

In case of linear magnetic model a useful and meaningful expression of the torque is obtained:

$$\tau = \frac{3}{2}p(L_d - Lq)i_d i_q + \frac{3}{2}p\lambda_{pm}i_q \tag{2.32}$$

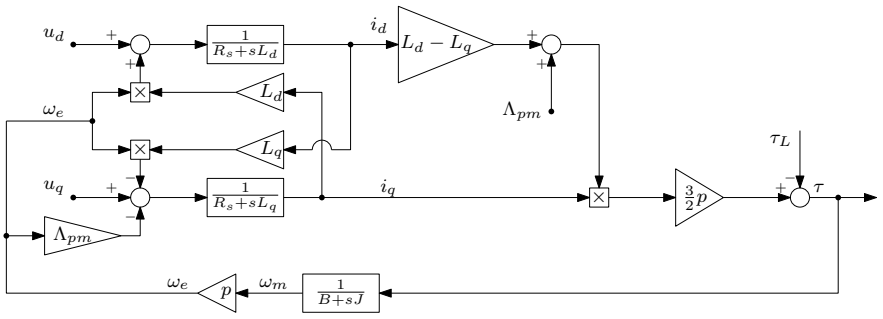


Figure 2.6: Overall scheme of the linear motor model.

Two different contributions are easily recognized in the expression, and they are due to two different principles. On one hand, the first term is due to the interaction among the rotor permanent-magnet flux and the stator currents. On the other hand, the second term called the reluctance torque, is due to the interaction between the stator current flux and the rotor magnetic anisotropic geometry (if present). SPM motor torques are characterized by the permanent magnet contribution only. On the opposite, SyR machines are characterized by the lack of magnets, resulting in a torque which is only due to the reluctance principle. PMAREL and IPM motors are in between the two cases presented, taking advantages of both the contributions.

An accurate modeling of the mechanical dynamic of a PMSM results quite complicated. Non-linear viscous friction, backslash and many other non-linear phenomena could affect the system. For this reason, a simplified first order model is often adopted

$$J\frac{d\omega_m}{dt} = B\omega_m + \tau + \tau_L, \tag{2.33}$$

at least for control design purposes. In the equation J represent the motor inertia, B accounts for a linear viscous friction and τ_L resumes all the loading torques that are independent from the motor position and speed. The overall motor model, including both the electrical and mechanical dynamic, is shown in Figure 2.6.

2.5 POWER CONVERTER MODEL

The power converter represents the core element of an electric drive circuit. The converter manages the energy conversion from DC electric quantities to AC ones. Figure 2.7 reports the example of a three-phase two-levels converter, known also as three-phase two-levels inverter, which is the topology considered in this thesis since it is probably the most widespread in industry.

A DC capacitor stabilizes the voltage and filters high frequency disturbances. The regulation of the available DC-bus voltages is performed by the six electronic switches. Depending on the application, the switches are realized by means of different technologies.

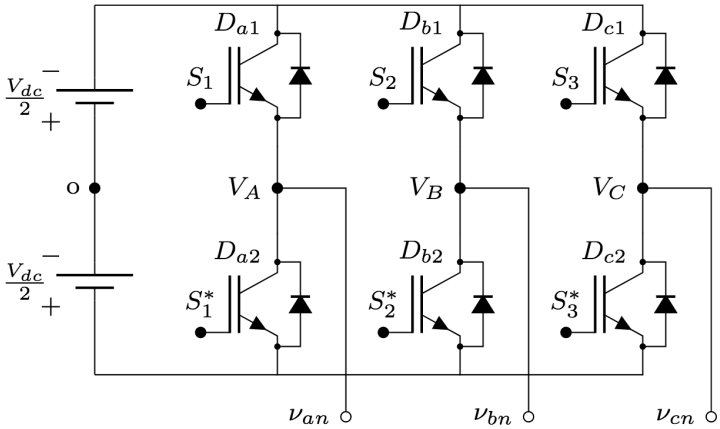


Figure 2.7: Three-phase two-levels voltage source inverter topology.

Insulated Gate Bipolar Transistor (IGBT)s are preferred in case of high voltages and currents and relatively low switching frequencies. In case of opposite specifications, Metal Oxide Semicondutor Field Effect Transistor (MOSFET)s are adopted. Different strategies are available to perform the *modulation* of this voltage, i. e. to select the state of the switches (on or off), see for instance Section 3.1. Anti-parallel diodes are added in the circuitry to allow the reverse conducting in the switches.

In case of a three-phase inverter, three branches of switches are present, namely *legs*. Each leg is formed by a couple of switches. Regardless the modulation strategy applied, switches lying in the same leg are turned on and off in an alternative fashion. This precaution needs to be adopted, in order to avoid short-circuit of the DC source. As a consequence, eight admissible combinations of switches are available, known also as *fundamental voltages*. These combinations, listed in Table 1, are associated to eight fundamental voltage vectors in the stationary $\alpha\beta$ reference frame, shown in Figure 2.8. The voltage vectors $\alpha\beta$ components can be parametrized as function of the vector index *idx*:

$$\begin{aligned} u_{\alpha,idx} &= \frac{2}{3} U_{DC} \cos\left(\frac{2\pi}{3}(idx-1)\right) \\ u_{\beta,idx} &= \frac{2}{3} U_{DC} \sin\left(\frac{2\pi}{3}(idx-1)\right) \end{aligned}, \; idx = 1,2,\ldots,6. \tag{2.34}$$

which are also named *active* vectors. The two zero vectors u_0 and u_7 have obviously null components.

The abc representation reported in the same figure is relevant, too. It provides three hints regarding regarding the transition between the fundamental vector.

First, the transitions that involves vectors on opposite edges are the most expensive, in terms of switching effort. In fact, all the switches of the converter need to change their states. Fort this reason, a well

Table 1: Configuration of the inverter switches for the eight fundamental voltages. State 1 means that the switch is on, whereas 0 means that the switch is off.

Voltage	S_1	S_2	S_3	u_{an}	u_{bn}	u_{an}
u_0	0	0	0	0	0	0
u_1	1	0	0	$2/3U_{DC}$	$-1/3U_{DC}$	$-1/3U_{DC}$
u_2	1	1	0	$1/3U_{DC}$	$-2/3U_{DC}$	$1/3U_{DC}$
u_3	0	1	0	$-1/3U_{DC}$	$2/3U_{DC}$	$-1/3U_{DC}$
u_4	0	1	1	$-2/3U_{DC}$	$1/3U_{DC}$	$1/3U_{DC}$
u_5	0	0	1	$-1/3U_{DC}$	$-1/3U_{DC}$	$2/3U_{DC}$
u_6	1	0	1	$1/3U_{DC}$	$-2/3U_{DC}$	$1/3U_{DC}$
u_7	1	1	1	0	0	0

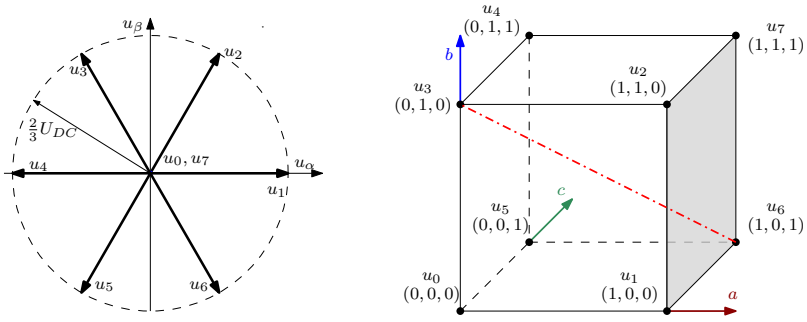


Figure 2.8: The fundamental voltage vectors in the stator reference frames $\alpha\beta$ and abc.

designed Finite Set (FS) MPC should avoid these transitions. On of these transition is highlighted in Figure 2.8 by a dash-dotted red line.

Secondly, transitions between fundamentals vectors located on the same face of the three-dimensional cube require always no more than two transition. In Figure 2.8 it is highlighted the case of u_1, u_2, u_6 and u_7, which represents the right face of the cube.

A third interesting fact that is evident form the abc representation regards the two zero fundamental vectors u_0 and u_7. In particular, any active voltage vector can be reached from one of the two null vectors with only two commutations.

2.5.1 *Inverter non-idealities*

In the previous section an ideal model was proposed for a three-phase inverter. This model is widely adopted in the control of electric drives, because of its simplicity. However, if a detailed description of motor currents dynamics is desired, the just developed representation could result too rough. In fact, many non-ideal phenomena were neglected, mostly due to the real functionality of th electronics devices. This

subsection aims to introduce some of them, in order to grasp the complexity of the actual system.

First of all, all the electronics components, i. e. diodes and transistors, are not perfect conductors. Voltage drops are introduced in the circuit, due to the activation voltage and the internal resistance. This means that when these devices are connected to the motor windings, they increase at least the overall resistance. In other words, a more accurate model of a drive should include both the motor stator resistance and the voltage drop due to the electronic devices. Being more precise, the additional voltage drop is different if a transistor or the related anti-parallel diode is conducting.

A second source of uncertainty in the inverter model is due to the transition between on and off states of the electronic devices. These transitions require a certain amount of time, which depend to the technology of the device, too. From the point of view of the equivalent voltage applied to the motor windings, these phenomena adds a further discrepancy with respect to the ideal model.

Finally, as a last example of inverter non-linearity, the dead-time effect is reported. Dead-time are introduced in the transistors control to avoid short circuits on the DC-bus when two transistors of the same leg are switched. In fact, the transitions between on and off state are not instantaneous, as mentioned above. After commanding the turning-off condition to a switch that was in a conducting mode, a short time interval is added, before commanding the activation of the other switch of the same leg. The dead-time is a short time interval added when a switch is commanded to be turned-off that was in a conduction mode, before activating the other switch on the leg. During this time the switch ends the transients toward the off-state and both the switches of the legs are off before the turning-on command.

3 MODEL PREDICTIVE CONTROL

The Model Predictive Control (MPC) was introduced in the late 1960s, with the introduction of parametric state-space models. This control methodology gained quickly popularity in many applications, particularly where accurate models were available, e. g. in aerospace and chemistry. The design of this class of regulators consists of two steps. First, the dynamics of the process are modeled, often by means of simple parametric LTI models, whose parameters are obtained with system identification routines. Modeling and identifying the plant is often the most expensive and time-consuming part of control design [19]. Then, the second step is the design of an optimal control law, based on the identified plant model.

According to the MPC framework, the control actions, i. e. the plant inputs, are obtained solving an open-loop optimal control problem for a finite prediction horizon, at each control period. The regulator relies on the system model to predict the process states in a future horizon. Then, the optimal control sequence is found by minimizing a pre-defined cost function. The cost function commonly includes a penalty on a certain set-point tracking and a penalty on the input effort.

MPC differs from other optimization-based techniques, e. g. Linear Quadratic Regulator (LQR), in the way of applying the optimized control sequence. In particular, the control action is applied in a receding horizon manner. This means that only the first control input of the sequence is applied each control step, discarding all the others. The control sequence is recomputed each control step, taking into account new measurements collected from the plant. As a consequence, the control inputs are adjusted to improve the set-point tracking performances.

The procedure behind the MPC can be resumed as:

- Prediction of the system state dynamics with a model for a selected number of future time steps, the so called prediction horizon (see Figure 3.2).
- Resolution of an online optimization problem, in order to obtain the optimal sequence of actions determined by the cost function.
- Adoption of a receding horizon strategy, which in turns implies the application of the first control input of the computed sequence to the plant.

The controller design is described hereinafter (Figure 3.1). In the description it is assumed that the full states of the system are avail-

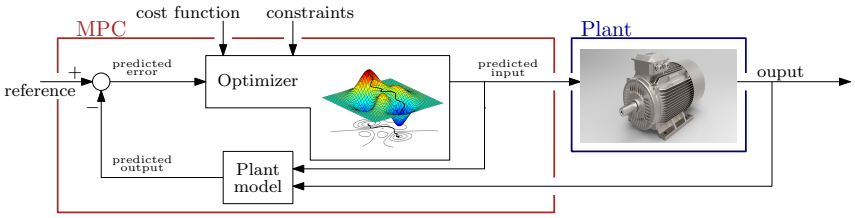


Figure 3.1: Block scheme of a MPC controller

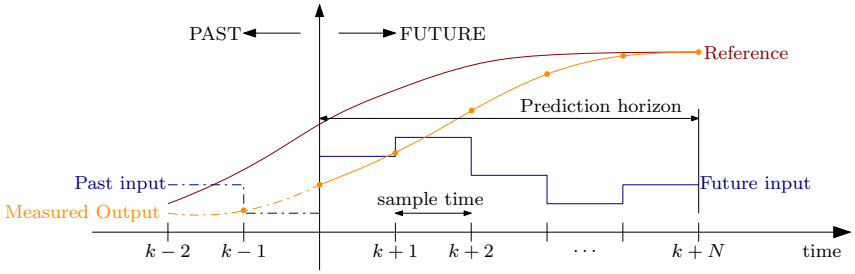


Figure 3.2: MPC working principle.

able, thus state and outputs are equivalent. This is due to the fact that this hypothesis holds for the current control of synchronous motor drives, which is the main application described in this dissertation. In most of the industrial applications, included the ones in the power electronics area, discrete-time LTI or linear parameter varying state-space models are used to represent the plant dynamics of interest, e. g. (2.27). According the state-space framework, the future states of the system can be predicted starting from (2.24):

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\
\mathbf{x}(k+2) &= \mathbf{A}\mathbf{x}(k+1) + \mathbf{B}\mathbf{u}(k+1) = \mathbf{A}^2\mathbf{x}(k) + \mathbf{A}\mathbf{B}\mathbf{u}(k) + \mathbf{B}\mathbf{u}(k+1) \\
\mathbf{x}(k+3) &= \mathbf{A}^3\mathbf{x}(k) + \mathbf{A}^2\mathbf{B}\mathbf{u}(k) + \mathbf{A}\mathbf{B}\mathbf{u}(k+1) + \mathbf{B}\mathbf{u}(k+2) \\
&\cdots \\
\mathbf{x}(k+N) &= \mathbf{A}^N\mathbf{x}(k) + \sum_{z=0}^{N-1} \mathbf{A}^{N-z-1}\mathbf{B}\mathbf{u}(k+z)
\end{aligned} \tag{3.1}$$

A compact matrix expression of the predicted states sequence is obtained by stacking the future states in the vector $\mathbf{X} = [\mathbf{x}(k+1), \ldots, x(k+N)]^T$ and future inputs in $\mathbf{U} = [\mathbf{u}(k), \ldots, u(k+N-1)]^T$:

$$\mathbb{A} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^N \end{bmatrix}, \quad \mathbb{B} = \begin{bmatrix} \mathbf{B} & 0 & \cdots & 0 \\ \mathbf{A}\mathbf{B} & \mathbf{B} & 0 & \vdots \\ \vdots & \vdots & \ddots & \\ \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \tag{3.2}$$

$$\rightarrow \mathbf{X} = \mathbb{A}\mathbf{x}(k) + \mathbb{B}\mathbf{U}.$$

In a nutshell, the future predicted states sequence depends only on the initial condition $\mathbf{x}(k)$ and the sequence of future inputs $\mathbf{U}$.

The second step in the design is the definition of an online problem, i. e. the cost function selection. In literature, most of applications adopt a cost which is quadratic with respect to the optimization variable, i. e. the input sequence $\mathbf{U}$. In particular, the cost function includes the Euclidean norm of the tracking error and the input effort:

$$\begin{aligned} J &= q(\mathbf{x}^*(k+1) - \mathbf{x}(k+1))^2 + r\,\mathbf{u}(k)^2 + \\ &+ q(\mathbf{x}^*(k+2) - \mathbf{x}(k+2))^2 + r\,\mathbf{u}(k+1)^2 + \\ &+ q(\mathbf{x}^*(k+3) - \mathbf{x}(k+3))^2 + r\,\mathbf{u}(k+2)^2 + \cdots + \\ &+ q(\mathbf{x}^*(k+N) - \mathbf{x}(k+N))^2 + r\,\mathbf{u}(k+N-1)^2 \\ &= \sum_{z=1}^{N} \left(\|\mathbf{x}^*(k+z) - \mathbf{x}(k+z)\|_{\mathbf{Q}}^2 + \|\mathbf{u}(k+z-1)\|_{\mathbf{R}}^2 \right) \end{aligned} , \tag{3.3}$$

subject to

$$\mathbf{x}(k+z+1) = \mathbf{A}\mathbf{x}(k+z) + \mathbf{B}\mathbf{u}(k+z) \tag{3.4}$$

$$\mathbf{y}(k+z) = \mathbf{C}\mathbf{x}(k+z) \tag{3.5}$$

$$\mathbf{u}_{min} \leqslant \mathbf{u}(k+z) \leqslant \mathbf{u}_{max} \tag{3.6}$$

$$\mathbf{y}_{min} \leqslant \mathbf{y}(k+z) \leqslant \mathbf{y}_{max}. \tag{3.7}$$

where q and r are weighting coefficients, whereas $\mathbf{Q}$ and $\mathbf{R}$ are often weighting diagonal matrices, having q and r as diagonal elements, respectively. It is reminded that these weighting matrices may have non-equal diagonal element and be non-diagonal in the general case.

The equality constraints (3.4) and (3.5) introduce the knowledge of the plant model in the optimization, while the inequality constraints (3.6) and (3.7) define input and output constraints. $\mathbf{x}^*$ is the reference to track, the matrix $\mathbf{Q}$ weights the tracking error, whereas $\mathbf{R}$ penalizes the input effort.

The expressions of $\mathbf{x}(k+z+1)$ and $\mathbf{u}(k+z)$ can be substituted in the cost function (3.3), and the formulation can be condensed in compact matrix form:

$$\begin{aligned} &\min_{\Delta \mathbf{U}} \left(\mathbf{X}^* - \mathbf{X}\right)^T \mathbb{Q} \left(\mathbf{X}^* - \mathbf{X}\right) + \mathbf{U}^T \mathbb{R} \mathbf{U} \\ &\text{s.t. } \mathbf{X} = \mathbb{A}\mathbf{x}(k) + \mathbb{B}\mathbf{U}, \end{aligned} \tag{3.8}$$

where $\mathbf{X}^* = [\mathbf{x}^*(k+1), \ldots, \mathbf{x}^*(k+N)]^T$, while $\mathbb{Q}$,$\mathbb{R}$ are positive definite diagonal square weighting matrices.

The just defined optimization problem is a Quadratic Programming (QP) problem type and it can be represented using a standard form:

$$\min_{\mathbf{U}} J = \frac{1}{2}\mathbf{U}^T \mathbf{H} \mathbf{U} - \mathbf{d}^T \mathbf{U} + c. \tag{3.9}$$

$\mathbf{H} = \mathbb{B}^T\mathbb{Q}\mathbb{B} + \mathbb{R}$ represents the $N \times N$ Hessian matrix, $\mathbf{d} = \left(\mathbf{X}^* - \mathbb{A}\boldsymbol{x}(k)\right)^T\mathbb{Q}\mathbb{B}$ the $N \times 1$ linear term vector and $c = \left(\mathbf{X}^* - \mathbb{A}\boldsymbol{x}(k)\right)^T\left(\mathbf{X}^* - \mathbb{A}\boldsymbol{x}(k)\right)$ a constant term.

Being at least $\mathbb{R}$ positive definite, the problem is strictly convex, i. e. a unique solution always exists. Moreover, this problem class can be solved in closed form, without using any iterative solver, when no constraints are included in the optimization, i. e. when (3.4) and (3.5) are neglected. This aspect is of particular interest for embedded applications, e. g. the control of electric drives, where the computational time needs to be of the order of hundreds of µs.

Taking the gradient of J and imposing its expression to zero, the optimal unconstrained solution results:

$$\nabla J = 0 = \mathbf{H}\mathbf{U} - \mathbf{d} \Rightarrow \mathbf{U}^{opt} = \mathbf{H}^{-1}\mathbf{d}. \tag{3.10}$$

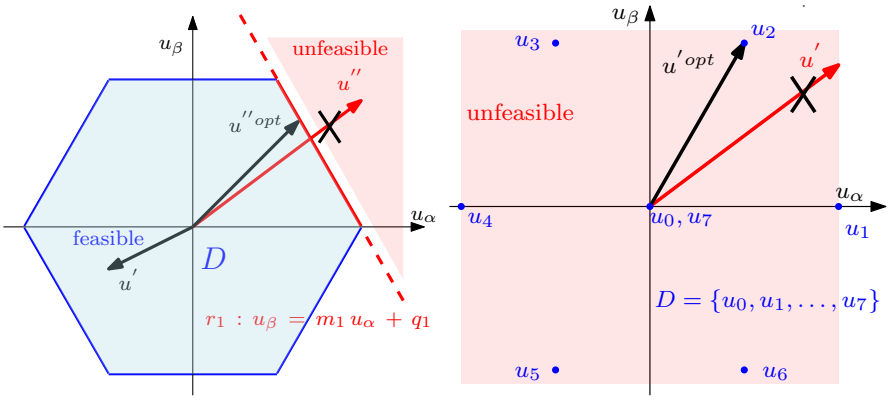
Finally, according to a receding horizon policy, only the first solution $\mathbf{u}^{opt}(k)$ is applied to the plant. All the other solutions are discarded.

MPC represents a promising architecture in electric drives applications, because of high dynamic performances, relatively easy tuning and possibility of including constrains in the problem resolution [69, 77]. However, the higher computational cost with the respect to PI controllers was a critical disadvantage in the last decades for the real-time implementation. Nowadays fast processors and new platforms hardware, e.g. Digital Signal Processor (DSP)-FPGA solutions, are available, thus interest in MPC is steadily growing [77].

Concerning the specific area of the electric motor drives, technology advances in the production of new motors, such as IPM motors, PMAREL motors and pure SyR motors expand the application areas of the predictive control paradigm [64]. In fact, various aspects concerning the optimal functioning of the motor drive can be considered in the MPC problem, such as the reduction of common-mode voltage [33], field-weakening operations of IPM motors [51], thermal stress of the power converters [24] or even special machines, such as dual three-phase motors [54]. Many successful implementations of this control strategy can be found for almost every synchronous motor topologies, such as SPM [7] and IPM motors [58] or SyR motors [3]. Even if most of these examples are related to the regulation of electric quantities of the motor, e. g. currents, electromagnetic torque or fluxes, MPC has been used in the regulation of the motors speed [13, 31, 84], too.

3.1 FINITE SET AND CONTINUOUS SET METHODS

When MPC is applied in the control of an electric motor drive system, two main categories of methods are found in literature, which refers to Continuous Set (CS) and FS methods.



(a) CS-MPC feasible region. (b) FS-MPC feasible set.

Figure 3.3: CS-MPC feasible region: u' is a feasible solution, whereas u'' an unfeasible one. In the second case the optimum u''^{opt} lies on the hexagon edge. FS-MPC feasible set: only the verteces are feasible points. The optimum must be one of the verteces.

Table 2: 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	[89]	NL-MPC, very general QPs	IP, ADMM, FGM	proprietary
HPIPM	[30]	LQP	IP	2-clause BSD
ODYS	[16]	QP, Embedded MPC	AS	proprietary
OSQP	[71]	QP	ADMM	Apache 2.0 License
qpOASES	[29]	QP	AS	2-clause BSD

Techniques falling into the former group are also known as indirect [7, 41, 66]. These methods are indirect meaning that a modulation strategy must be coupled to the MPC, e. g. the Pulse Width Modulation (PWM). As a consequence, the control input constraints, i. e. the voltage applied to the motor windings, are defined by a polytopic region in a stationary reference frame, which depends on the power converter topology. In case of a three-phase two-levels inverter the region has an hexagonal shape (see Figure 3.3a). The QP problem associated to these methods is solved in an unconstrained fashion or in the constrained fashion, implementing a proper solver as in [76]. In the cited paper, for instance, an open source software named qpOASES [28], developed by H. J. Ferrau et all., is used to solve the constrained linear QP problem associated to the MPC algorithm. qpOASES is a C++ software package which exploits a parametric active-set technique as solver, which implements several tools to improve the numerical stability of the solution and it provides an interface of the software package with other external software, in particular MATLAB Simulink. A list of other available solvers for online QP optimization is reported in Table 2. The main drawback of CS approach is

Table 3: Survey of MPCs for electric drive applications. The acronyms are: NL: Non Linear; E-MPC: Explicit-MPC.

References	Year	Controlled Variable	Model	Method	Constraints		Tested Motors	Platform
					Input	Output		
[7]	2009	Speed&Currents	LTI	E-MPC	✓	✓	SPM	dSPACE 1004
[55]	2012	Speed+Torque/Flux	LPV	E-MPC	✓	✓	IM-PMSM	Sharc ADSP 21062 + TMS320C6713
[67]	2013	Currents	LPV	FGM	✓		IPM	TMS320F240
[15]	2015	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[34]	2019	Currents	LPV	AS	✓		IPM	dSPACE 1006
[16]	2021	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[90]	2021	Currents	NL	AS	✓		SRM	dSPACE 1007

that the modulation strategy adds further computational burden to the overall scheme. Moreover, the efficiency of the drive is not significantly improved by the technique, even if dynamic performances of the plant are improved. Thus, its applications are limited to the areas where fast dynamics are required. Despite these disadvantages, more and more MPC solutions of this type are being publishing in the last years, see for instance Table 3. This suggests that many improvements are expected in the very next future.

The FS techniques are also known as direct [44, 79], and they are valid candidate in the current control of PMSM. According to these techniques, the converter DC-bus voltage is not modulated within one control period. Thus, the voltage that can be applied to the PMSM windings 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, as remarked in Figure 3.3b. Advantages of these methods are bringing them close to effective industrial applications. Firstly, the modulator removal eases the regulation scheme, directly taking into account the discrete nature of a power converter. Secondly, the average switching frequency of power converters are reduced with respect to the PWM-based solutions, reducing the inverter commutation losses. In addition, the switching frequency can be manipulated by properly tuning the cost function [22], accomplishing desired design specifications. These techniques, in fact, represent an effective control method for high power converters [74], where the minimization of switching losses can lead to an increase of the inverter efficiency. For this reason, the industry finds it more attractive than CS techniques. Finally, the simplicity of the FS allows to include additional constraints in the MPC cost function, e.g. maximum output currents, or to design non-linear costs. The main drawback of FS-MPCs is the computational burden, which increases exponentially with the prediction horizon. It scales much faster than CS algorithms with increasing prediction horizons. A Nondeterministic Polynomial time (NP)-hard optimization problem has to be solved online at each control period. This represents a relevant issue for real-time applications. However, the availability of more powerful microcontrollers and more efficient solvers for the optimization task permit,

nowadays, to overcome the issue. The most widespread solvers proposed for such problem class are the *branch and bound* and the *sphere decoding* [74]. Heuristic methods are available to further speed up the optimization solution, too. More details regarding the solvers for FS-MPC are given in Chapter 7.4. A second drawback is the increase of the motor currents distortion [45], which in turns can increase the Joule losses if too large sampling times are used or if the method is applied to machines characterized by small inductances.

4

BEHAVIORAL SYSTEM THEORY

The data-driven predictive controllers presented in this thesis are based on the *behavioral system theory*. This approach to the linear control theory is not so popular in literature, in particular when considering electric drives applications. For this reason, a brief introduction about this framework of the control engineering is presented. It is remarked that the aim of this section is to provide some ideas and concepts of the topic, without including too much mathematical formalism. A complete and formal description of the behavioral system theory can be found in [85]. Even if the first works were written around the 2000s, many recent theoretical contributions have been published in the last years, for instance [56] and [96]. This confirms that the topic is still very hot in the control community.

The notation of this section is slightly different from the rest of this thesis. In fact, it is chosen to be as coherent as possible with the notation used by Willems [85], whose research group was one of the first that worked on the topic.

4.1 PRELIMINARIES AND NOTATION

Classical control design techniques are essentially model-based, meaning that a representation of the plant, i. e. a model, is built to correctly design the control law. This is the case of MPC methods presented in Section 3. On the contrary, data-driven controllers do not need such prior knowledge on the system, and they provide controller design procedures that requires only on a dataset of sampled Input/Output (I/O) measurements from the plant.

According to the behavioral system theory, the mathematical model that was used to represent the system nature is replaced by the subspace of the system signals. In particular, this subspace contains all the trajectories of the system dynamics. Even if it is not trivial, this methodology represents the standard way of proceeding for real-life control applications characterized by high dimensional models. In fact, the controllers design is performed starting from observations of trajectories, since it is rarely available a parametric representation of the plant [57].

It is first introduced the concept of *trajectory*, or behavior, of an LTI system. The trajectory is a sequence of samples of the tuple $w(z) = [u(z); y(z)]$, where $u(z)$ and $y(z)$ are the system input and output, respectively. A T-long trajectory is, for instance, $w(1), w(2), \ldots, w(T)$.

Within a T-long trajectory, it is possible to define some L-long subset, such that $1 \leqslant L \leqslant T$, which are simply snapshot on the plant dynamics:

$$
\begin{aligned}
&[w(1), w(2), \ldots, w(L)] \\
&[w(2), w(3), \ldots, w(L+1)] \\
&\ldots \\
&[w(T-L+1), w(T-L+2), \ldots, w(T)]
\end{aligned}
$$

The corner stone of the behavioral system theory is probably the Willems *Fundamental Lemma*. This lemma aims to answer the following question: which conditions need to be verified by these L-long windows in order to span the whole space of trajectories that the system can produce? [85] The answer to this question will conclude this overview on the behavioral system theory.

In the behavioral language, a discrete-time dynamical system $\mathfrak{B}$ is defined as a 3-dimensional tuple $(\mathbb{T}, \mathbb{W}, \mathfrak{B})$ where $\mathbb{T} \subseteq \mathbb{R}$ is the time, $\mathbb{W}$ represents a signal space and $\mathfrak{B} \subseteq \mathbb{W}^{\mathbb{T}}$ the behavior. The analysis is restricted to discrete-time causal finite-dimensional systems, thus the adopted time axis is $\mathbb{T} = \mathbb{Z}_{\geqslant 0}$. Some simple concepts of the linear control theory, such as the concepts of linearity of a system and time-invariance, are reported in this framework.

Definition 4.1.1. *[57] Let $(\mathbb{Z}_{\geqslant 0}, \mathbb{W}, \mathfrak{B})$ be a dynamical system.*

1. *It is linear if $\mathbb{W}$ is a vector space and $\mathfrak{B}$ is a linear subspace of $\mathbb{W}^{\mathbb{Z}_{\geqslant 0}}$.*
2. *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 $\sigma\mathfrak{B} : \{\sigma w | w \in \mathfrak{B}\}$.*

The system class $(\mathbb{Z}_{\geqslant 0}, \mathbb{R}^w, \mathfrak{B})$ can be denoted by $\mathfrak{L}^w$, which is represented hereinafter only by its behavior $\mathfrak{B}$. Some important invariants associated to the system class are:

- $\mathbf{w}$ is the variable cardinality, if $\mathfrak{B} \in \mathfrak{L}^w$, then $\mathbf{w}(\mathfrak{B}) := w$;
- $m \in \mathbb{Z}_{\geqslant 0}$ is the input cardinality;
- $p \in \mathbb{Z}_{\geqslant 0}$ is the output cardinality defined as $p := \mathbf{w}(\mathfrak{B}) - m$;
- $n(\mathfrak{B})$ si the state cardinality. Every behavior $\mathfrak{B} \in \mathfrak{L}^{w=m+p}$ admits an equivalent I/O representation in state space form: $\mathfrak{B}(A, B, C, D)$ that can be computed as:

$$
\begin{aligned}
\mathfrak{B}(A, B, C, D) &:= \{w \in (\mathbb{R}^{m+p})^{\mathbb{Z}}\}, \\
\text{such that} \quad & \sigma x = Ax + Bu \\
\text{and} \quad & y = Cx + Du
\end{aligned}
$$

with matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. Thus, the state cardinality $n = n(\mathfrak{B})$ defines the the order of the minimal representation $\mathfrak{B}(A, B, C, D)$.

- $l(\mathfrak{B}) \in \mathbb{Z}_{\geqslant 0}$ is the lag of the system, defined as the smallest integer l such that the observability matrix:

$$\mathcal{O}_l(A, C) := [C, CA, \ldots, CA^{l-1}]^T$$

has rank $n(\mathfrak{B})$.

A restriction of the behavior in a time window of length T as:

$$\mathfrak{B}_T = \{w_p \in (\mathbb{R}^{m+p})^T | \text{ there is } w_f, \text{ such that } w_p \wedge w_f \in \mathfrak{B}\}$$

where $w_p \wedge w_f$ denotes the concatenation of trajectories w_p and w_f. This represents the set of all the T-long trajectories of the system.

A system behavior $\mathfrak{B}$ can be rewritten also as the product space of two sub-behaviors $\mathfrak{B}^y \subseteq (\mathbb{R}^m)^{\mathbb{Z}_{\geqslant 0}}$ and $\mathfrak{B}^u \subseteq (\mathbb{R}^p)^{\mathbb{Z}_{\geqslant 0}}$ which are respectively, the input and output signal spaces. Thus, any trajectory w can be written highlighting the input-output partition:

$$w := \begin{bmatrix} u \\ y \end{bmatrix}.$$

A fundamental property of a behavior is the controllability:

Definition 4.1.2. *[57] A system is said to be controllable if for every $T \in \mathbb{Z}_{\geqslant 0}$, $w^1 \in \mathfrak{B}_T$, $w^2 \in \mathfrak{B}$ there exist $w \in \mathfrak{B}$ and $T' \in \mathbb{Z}_{\geqslant 0}$ such that $w_t = w^1_t$ for $1 \leqslant t \leqslant T$ and $w_t = w_{t-T-T'}$ for $t > T + T'$.*

This concept of controllability has a very practical meaning. In fact, this property is satisfied when any two trajectories of a given system can be patched together in finite time.

Selected a number of rows equal to L, the Hankel matrix associated to the signal $w = [w_1, w_2, \ldots, w_T]$ is defined as:

$$\mathcal{H}_L(w) = \begin{bmatrix} w(1) & w(2) & \ldots & w(T-L+1) \\ w(2) & w(3) & \ldots & w(T-L+2) \\ \vdots & \vdots & & \vdots \\ w(L) & w(L+1) & \ldots & w(T) \end{bmatrix}. \tag{4.1}$$

In order to develop a controller based on behavioral system theory, it is crucial the *persistency of excitation* condition:

Definition 4.1.3. *[85] Given, $L, T \in \mathbb{Z}_{\geqslant 0}$ such that $T \geqslant L$. The signal $f \in \mathbb{R}^{Tm}$, is said to be persistently exciting of order L if the Hankel matrix $\mathcal{H}_L(f)$ is full rank.*

In other words, a given signal is persistently exciting of a given order (assume L), if the trajectories that form the column of (4.1) are not trivial.

Given an initial trajectory, $w_{ini} \in \mathfrak{B}_{T_{ini}}$, with $T_{ini} \geqslant l(\mathfrak{B})$, and an input sequence u_r, the state can output can be steered to a desire value:

Lemma 4.1.1. *[57] Let $\mathfrak{B} = (A, B, C, D)$ and $\mathfrak{B} \in \mathcal{L}^{m+p}$ be a minimal input/state/output representation.*
Then for all $w_{ini} \in (\mathbb{R}^{m+p})^{T_{ini}}$, with $T_{ini} \geqslant l(\mathfrak{B})$ and $w_r = [u_r, y_r]^T$,

$$w_{ini} \wedge w_r \in \mathfrak{B}_{T_{ini}+T_r} \Rightarrow \textit{there is unique } x_{ini} \in \mathbb{R}^{n(\mathfrak{B})} \textit{ such that } y_r = \mathcal{O}_{T_r}(A, C)x_{ini} + \mathcal{T}_{T_r}(A, B, C, D)u_r,$$

where $\mathcal{T}_{T_r}$ is the lower triangular Toeplitz matrix:

$$\mathcal{T}_{T_r}(A, B, C, D) = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{T_r-2}B & \dots & CB & D \end{bmatrix}. \tag{4.2}$$

It is highlighted that, if matrices A, B, C, D are known, it the initial state x_{ini} is obtainable. This lemma coupled with the theory of the MPC represents a first step towards the design of predictive controllers from a set of measurements.

As conclusion of this short review on the behavioral system theory, the *fundamental lemma* is reported:

Lemma 4.1.2. *[85] Given $T, t \in \mathbb{Z}_{\geqslant 0}$. Consider a controllable, linear and time invariant system $\mathfrak{B}$ with $w = \mathrm{col}(u, y) \in \mathfrak{B}_T$ a T samples long trajectory, also consider a persistently exciting input u of order $t + n(\mathfrak{B})$.*
Then, any t samples long trajectory $w \in \mathfrak{B}_t$ can be written as a linear combination of the columns of $\mathcal{H}_t(w)$, i.e.

$$\mathrm{colspan}(\mathcal{H}_t(w)) = \mathfrak{B}_t \,.$$

This lemma provides the enabling result for data-driven controls. In fact, it states that any trajectory of the system can be described, or a predicted, by a predictor computed as linear combination of Hankel matrix columns (4.1). Which in turns means that, given a measured trajectory w, a controlling input u_r and an initial state w_{ini} that meet imposed constraint (i.e. persistency of excitation), any future N-step-long trajectory can be estimated as:

$$w_{ini} \wedge w_r \in \mathfrak{B}_{T_{ini}+T_r} \iff \exists g, \text{ such that } \mathcal{H}_{T_{ini}+T_r}(w)g = \mathrm{col}(w_{ini}, \mathrm{col}(u_r, y)), \tag{4.3}$$

where the equation:

$$\mathcal{H}_{T_{ini}+T_r}(w)g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u_r \\ y \end{bmatrix} \tag{4.4}$$

is a linear system having as unknowns the auxiliary variable g and the future output trajectory y. A more detailed explanation and proofs of all the lemmas reported in this chapter can be found in [85].

The take-to-home messages of this brief overview on the behavioral system theory are resumed:

- the lemma (4.1.2) permits to substitute the system identification procedure of the model-based design. Parametric models are replaced by linear combinations of available snapshots on the system behavior;
- the fundamental lemma (4.1.2) provides also guidelines for the choice of the length of these snapshots;
- lemma (4.1.1) assures that, knowing the initial condition of a system, it is possible to uniquely predict its output, when excited by a given input sequence, starting from data samples rearranged in proper Hankel matrices (4.1).

Part II

MODEL-FREE PREDICTIVE CONTROL

Mode-free predictive control is an emerging branch of the finite-set model predictive control of electric drives. The voltage balance model of synchronous motors is replaced by non-parametric adaptive models in the estimation of the future machine currents. All the parameters of this non-parametric model are online estimated by means of observers. The accuracy of the states prediction achieved with these methods is impressive, considering that only a few a-priori knowledge of the plant is available. Promising results are obtained also in terms of overall drive efficiency. A reduction of the current harmonic distortion and reduction of the switching frequency can be achieved with this method, reducing the Joule losses and the commutation losses, respectively.

5 MODEL-FREE PREDICTIVE CURRENT CONTROL

In model-based predictive current control schemes the current predictions are based on the state-space model of the system, conveniently written in a dq reference frame synchronous to the rotor position (2.23). The prediction accuracy is strongly influenced by the knowledge of the parameters, for which several identification techniques are available [35, 58, 61, 65].

Several approximations are commonly introduced to simplify the system description, such as linear magnetic flux-current characteristics (2.10), constant stator resistance and absence of magnetic cross-coupling between the direct and quadrature axes. As a consequence, parameter mismatches always affect the model, because of the electric load changes induced by either the iron-saturation (2.20) (particularly evident in SyR motors) or the temperature variation. Parameters mismatch can be reduced by the use of observers, which anyway pose the problem of convergence, stability and accuracy. Online parameter tracking also improves any model-based scheme, at the price of higher computational and tuning efforts.

A different approach is considered in this chapter. The conventional description of the system is abandoned, moving to a model-free concept firstly introduced by Lin et al. [49]. The motor current variations caused by the application of each of the eight inverter base voltage vectors Figure 2.8 over a switching interval are stored in two Look-Up Table (LUT)s, one for each axis. The LUTs content is continuously updated online. A key parameter of a model-free predictive current control is the LUTs update frequency.

If the frequency is too low, the scheme suffers of *stagnation* problem, described in [50]. In a nutshell, if a voltage vector is not applied for many consecutive time steps, the stored information regarding the related current variation is obsolete and unreliable. Long stagnation periods could compromise even the system stability.

In order to avoid stagnation, several methods have been presented. They can be divided in two groups: *direct* and *indirect* anti-stagnation methods. A first simple, but bulky, method for increasing the update frequency is obtained with the direct anti-stagnation algorithm proposed by the same authors in [50]. A minimum refresh frequency is imposed for the current LUTs. If one of the base voltages is not applied for a predefined time window, the voltage vector is forcedly imposed as next voltage reference. In other words, MPC optimum detection is neglected and the updated information is retrieved at the cost of an increase of the current ripple.

A smart anti-stagnation algorithm is proposed in [20], where an indirect LUTs reconstruction is presented. Current LUT update is based on the mathematical relationships that link the inverter input voltages. The knowledge of the last three current variations permits an approximated update of the other LUTs elements. The update is obtained without penalizing the current ripple as in [50], since the updating frequency is high enough to prevent stagnations. Furthermore, the current ripple and the update frequency are not so strictly linked, as in the previous model-free predictive control solutions [49, 50].

In this chapter an up-to-date algorithm is described, which further increases the updating frequency of the LUTs, while maintaining the cost function minimization and by taking advantage only on the past current measurements. The LUTs update is guaranteed after each current measurement and it skips any stagnation completely.

The scope of the model-free control architecture is to avoid the need of motor parametric models. This is a particularly valuable feature for plug-and-play drives connected to motors with few or no parameters available. Examples are general purpose drives and ac drives for compressors adopted in cooling equipment (sometimes the motor is not accessible at all). In this light, the model-free predictive control is a promising control strategy for electric motors and it is worth attention from researchers.

5.1 THEORY OF DEADBEAT MODEL-FREE CONTROL

The idea underlying this first Model-Free (MF) solution is to keep the computational cost as low as possible. For this reason, the cost function behind the MPC (3.3) is simplified as much as possible. In details the input effort is not penalized ($r = 0$) and the prediction horizon N is set equal to one. Under these assumptions, the cost function J defined assumes the following form:

$$J(\mathbf{u}_z) = \|\mathbf{i}^*_{dq} - \mathbf{i}^z_{dq}(k+1)\|^2 \tag{5.1}$$

The MPC is equivalent to a deadbeat control. It is reminded that $\mathbf{i}^*_{dq} = \left[i^*_d; i^*_q\right]$ is the reference current vector and $\mathbf{i}^z_{dq} = [\widehat{i}_d; \widehat{i}_q]$ is the dq vector of the predicted currents due to one of the eight base inverter voltage vectors $\mathbf{u}_z$ (Figure 2.8). Of course, more complex cost functions can be adopted at the price of a higher computational load and an additional tuning of cost function gains [32, 88]. For instance, a thermal stress based MPC was implemented in [24]. However, the performance of a model predictive current control depends on the current prediction accuracy, which is the topic of this chapter. In this light, the adoption of (5.1) seems appropriate, since it allows a fair comparison with other predictive control techniques.

Following the principle of simplicity, the just obtained optimization problem is solved by means of enumeration. This means that the cost is evaluated for each dq base voltage, whose components are:

$$\begin{cases} \mathbf{u}_{dq}^{z} & = \frac{2}{3} U_{dc} \left[\cos((z-1)\frac{\pi}{3} - \theta_e), \sin((z-1)\frac{\pi}{3} - \theta_e) \right]^{T} \quad z \in [1, \dots, 6] \\ \mathbf{u}_{dq}^{z} & = \left[0, 0 \right]^{T} \qquad\qquad\qquad\qquad z = 0, 7 \end{cases} \tag{5.2}$$

where U_{dc} is the inverter bus voltage. The pedix dq, which denotes the reference frame, is omitted in the following to ease the notation.

The model-free control is applied to a SyR machine, whose electric parameter are time-varying and depend on the working point. The considered motor magnetic flux linkages suffer of severe saturation effects, as proved by Fig. 5.1, in particular non-null cross-saturation inductances (see Section 2.2). Therefore, a specific motor characterization is required to achieve the correct system model, which usually implies laboratory tests and facilities, and data post-processing. The commissioned parameter should be then inserted in a non-linear model, as the one proposed in (2.20).

It is convenient to introduce the expression of motor current variations due to the application of a voltage vector. Their expression are easily derived from the continuous-time state-space model of the current derivatives (2.23):

$$\Delta \mathbf{i}^{z}(k+1) = T_s \boldsymbol{A}_c \mathbf{i}(k) + T_s \mathbf{B}_c \mathbf{u}^{z}(k) \tag{5.3}$$

Opposite to model-based control, the considered model-free algorithm uses only the current measurements without information about motor parameters. Each of the seven base voltage vectors, if applied, results in current variations on both d and q axes that can be stored in two different LUTs. The seven current variations at time step k due to the seven voltage vectors are supposed to be known and actual:

$$\mathbf{i}(k) - \mathbf{i}(k-1) = \Delta \mathbf{i}^{z}(k) \tag{5.4}$$

According to the FS policy, the cost function (5.1) is evaluated seven times at each control period. The next-step currents are estimated as follows:

$$\begin{aligned} \widehat{\mathbf{i}}^{z}(k+1) &= \mathbf{i}(k) + \Delta \mathbf{i}^{z}(k) \\ \widehat{\mathbf{i}}^{z}(k+2) &= \widehat{\mathbf{i}}^{z}(k+1) + \Delta \mathbf{i}^{z}(k+1) \end{aligned} \tag{5.5}$$

where $\mathbf{i}(k)$ are the current measurements at time k and $\Delta \mathbf{i}^{z}(k)$ is the current variation due to the vector $\mathbf{u}^{z}(k)$ which has been already decided at time $(k-1)$. The voltage vector to be applied at time $(k+1)$ is selected by means of (5.1).

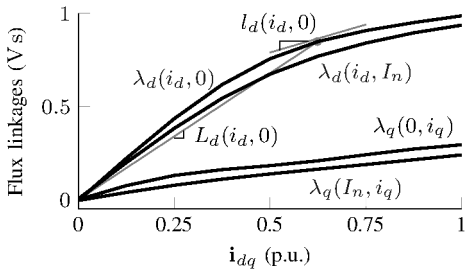


Figure 5.1: Magnetic flux linkages of the considered SyR motor at different currents and an example of d-axis apparent L_d and differential l_d inductances.

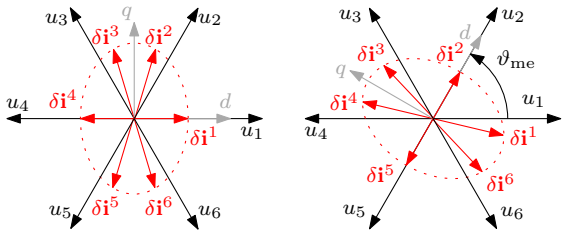


Figure 5.2: Current variations $\delta\mathbf{i}^z$ in a SyR motor with the six active vectors $\mathbf{u}^z$ at different rotor position: $\vartheta_e = 0$ (left) and $\vartheta_e = \pi/3$ (right).

Each of the current variations in (5.4) can be split in two components:

$$\Delta\mathbf{i}^z(k) = \delta\mathbf{i}^0(k) + \delta\mathbf{i}^z(k) \tag{5.6}$$

where $\delta\mathbf{i}^0 = [\delta i_d^0; \delta i_q^0]$ is the natural response and $\delta\mathbf{i}^z = [\delta i_d^z; \delta i_q^z]$ is the forced response to an active voltage vector $\mathbf{u}^z$. In case of $\mathbf{u}^0$, it is straightforward that $\Delta\mathbf{i}^0 = \delta\mathbf{i}^0$.

Since the values of $\Delta\mathbf{i}^z$ are obtained from measurement, they inherently contain information about the real behaviour of the SyR motor close to the actual working point. Fig. 5.2 shows an example of current variations $\delta\mathbf{i}^z(k)$, $z \in [1 \dots 6]$, relative to two different rotor positions. For simplicity, it is assumed $\mathbf{i}(k) = 0$. Due to the anisotropy of the SyR motor, the same active voltage vectors induce different current variations, depending on the rotor position ϑ_e. In general, it is $\Delta\mathbf{i}^z = f(\mathbf{i}, z, \vartheta_e)$. Similarly, the current variations associated to the natural response depends on the actual currents and the motor back-electromotive force, i.e. the speed ω_e, so that $\delta\mathbf{i}^0 = f(\mathbf{i}, \omega_e)$. LUTs can be initialized with all zeros. In fact, the algorithm would automatically proceeds applying all the base voltage to recover the response of the system, if a zero current reference is commanded. Alternatively, opposite base voltages could be applied in a row to obtain a smoother and safer initial start-up, i.e., (u_1, u_4), (u_2, u_5) and (u_3, u_6).

The LUTs content should be updated with the highest possible frequency to assure that the stored current variations return reliable information about the actual working point of the SyR motor. However,

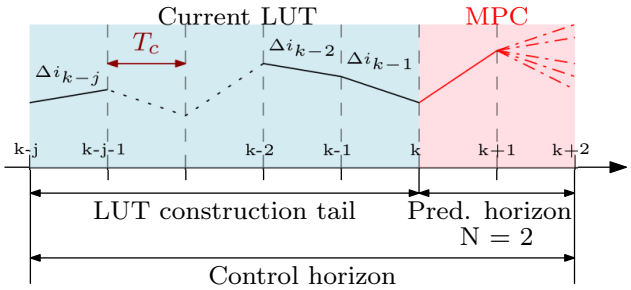


Figure 5.3: Discrete time representation of the control horizon.

only one voltage vector per control period step can be applied with a finite-set model predictive control algorithm. It implies that only one pair of current measurement variations $\Delta\mathbf{i}$ stored in the LUTs can be updated based on the latest current measurement. For the sake of current prediction accuracy, the remaining six pairs of $\Delta\mathbf{i}$ should be updated with a different strategy.

5.2 THE UPDATING OF THE CURRENT VARIATIONS LUTS

As mentioned above, a possible drawback of the model-free approach is the *stagnation* effect [20, 50]. It happens when one (or more) element of the LUTs are not updated for many time steps, which means that a voltage vector has not been used for several switching cycles.

The FS algorithm applies just one (out of seven) base voltage vector for an entire control period T_c. The anti-stagnation solution proposed in [8, 20] uses the current variations relative to the last three periods to reconstruct all the other (older) four ones. A weak point is that the reconstructions are made for use in the prediction horizon time span, when the rotor electrical position ϑ_e may have changed.

There are also a couple of possible situations that can corrupt the mechanism. The first case is when only two voltage vectors are applied for a long interval and thus only two couple of LUTs elements are actually updated. Even worse the second case, when only one vector is applied. These two cases may prevent a stable length of the LUTs construction tail (see Fig. 5.3). The problem of the stagnation, i.e. the lengthening of the LUTs construction tail, was quite evident since the very beginning. A constant tail length was guaranteed in the seminal paper [50] at the price of non-optimal base voltage vector choices, which cause current ripple inhomogeneities.

The weak points outlined above require an indispensable improvement for the practical application of the model-free control paradigm. In a nutshell, the goal is to maximize the updating frequency of the current variations $\Delta\mathbf{i}^z$. The goal can be obtained by the method described in Sect. 5.2.1 and 5.2.2. Furthermore, the rotor electrical position variation can be accounted for as explained in Sect. 5.2.3.

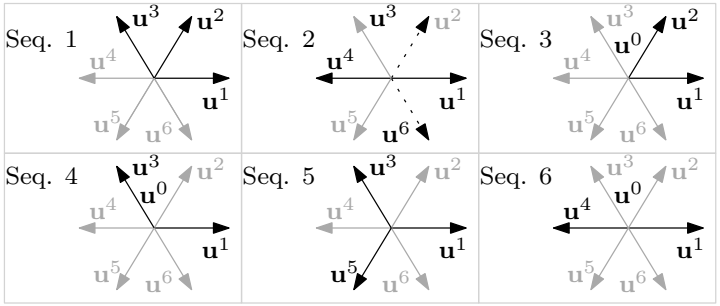


Figure 5.4: Voltage vector sequences. The rotation by multiples of $\pi/3$ of the selected (black) vectors returns the same sequence type. Dotted vectors in sequence 2 means that the sequence is obtained by either vector $\mathbf{u}^2$ or $\mathbf{u}^6$.

5.2.1 *The voltage triplet identification*

The proposed LUTs update method is based on the identification of particular combinations of three different voltage vectors referred to as *triplet*, whereas the specific combination they form is called *sequence*.

Six sequences can be defined as shown in Fig. 5.4. It is important to underline that the highlighted vectors are just examples to graphically define the sequences. For instance, sequence 1 is formed by either the vectors $(1,2,3)$ or $(2,3,1)$, as well as $(3,1,2)$. Furthermore, sequence 1 is formed by either the vectors $(3,4,5)$ or $(5,6,1)$, including all the possible permutations.

The sequences can be used to define mathematical relationships that allow to reconstruct the four current variations due to the remaining four voltage vectors, as it is reported in Sect. 5.2.2. A practical example is also reported in [8]. Since the current reconstructions depends on the sequence type, it is convenient to define an algorithm for the sequence identification.

The first step of the solution proposed consists in defining a buffer $B_u = [\mathbf{u}^z(t_3); \mathbf{u}^z(t_2); \mathbf{u}^z(t_1)]$ containing the last three applied voltage vectors, where $\mathbf{u}^z(t_1)$ corresponds to the latest applied voltage vector and $\mathbf{u}^z(t_3)$ to the oldest one, that is $t_3 < t_2 < t_1$. The buffer is updated by following two rules:

- the three voltage vectors must be different;
- the buffer is time oriented, i.e. the indexes are stored according to the chronological order of the related voltages.

The buffer updating rules guarantees that the three voltage vectors are suitable to form a triplet.

The k-permutations of n base vectors are the different ordered arrangements of a k-element subset. In the present case, $n = 7$, $k = 3$ and the permutations are $n!/(n-k)! = 210$, so that the identification of the triplets is not trivial.

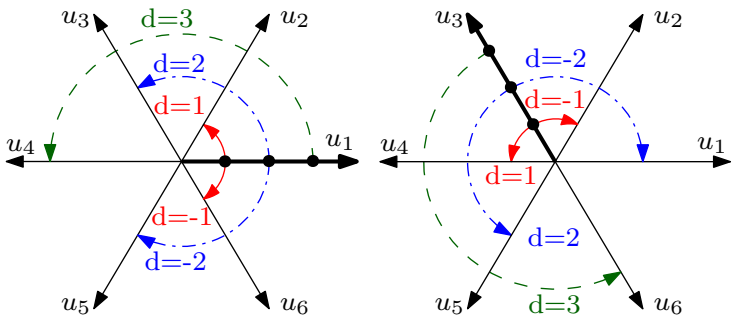


Figure 5.5: Examples of angular displacement and distance d calculation between active vectors. Reference vectors are $\mathbf{u}^1$ (left) and $\mathbf{u}^3$ (right), respectively.

As a second step, one defines the *distance* as the *normalized* phase displacement between $\mathbf{u}^z(t_1)$ and the other two voltage vectors of the buffer. The two distances are stored in a buffer $B_d = [d_1, d_2]$, where d_1 is the distance between $\mathbf{u}^z(t_1)$ and $\mathbf{u}^z(t_3)$, and d_2 is the distance between $\mathbf{u}^z(t_1)$ and $\mathbf{u}^z(t_2)$. The distances are calculated according to an anticlockwise positive direction and they are defined only in the set $[-2; -1; 0; 1; 2; 3]$. With reference to Fig. 5.5, with a voltage buffer $\mathbf{u}^z(t_1) = \mathbf{u}^1$, $\mathbf{u}^z(t_2) = \mathbf{u}^6$ and $\mathbf{u}^z(t_3) = \mathbf{u}^3$, the buffer of distances is $B_d = [2, -1]$.

The sequence identification process is shown in Fig. 5.6. It can manage all possible combination, including the special case of a zero-voltage vector in the three-element buffer. The remaining sequences can be identified by defining simple rules on the distances in the three-element buffers. For example, the sequence 1 is identified when one of following rule applies to the buffer B_d:

$$|d_1|, |d_2| \leqslant 2 \quad \text{and} \quad d_1 d_2 = 2 \quad \text{or} \quad d_1 d_2 = -1 \tag{5.7}$$

Sequence 2 is detected by the rule:

$$d_1 d_2 = -2 \quad \text{or} \quad d_1 = 3 \quad \text{or} \quad d_2 = 3 \tag{5.8}$$

The use of the distances to identify the sequence results in a computationally fast method. Only a handful of *if* statements is necessary to carry out the sequence identification, allowing the algorithm implementation even on basic microprocessors.

The sequence identification process is shown in Fig. 5.6. It manages all possible combinations, including the special case of a zero-voltage vector in the three-element buffer.

5.2.2 *Current variations reconstruction*

The vectorial relationships between the magnitude of the current variations can be calculated for each sequence. The mathematical equations are reported in Table 4. A practical example of LUTs updating is

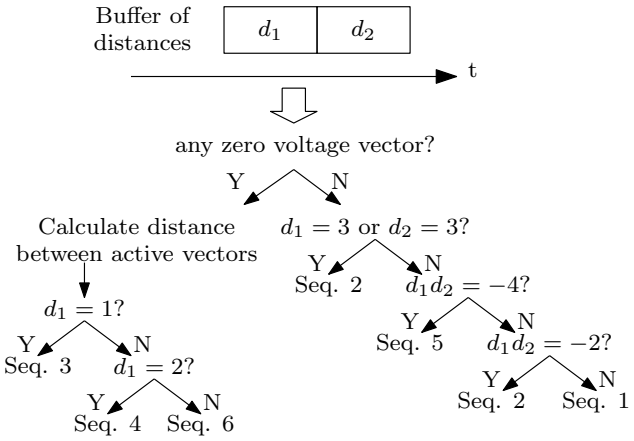


Figure 5.6: Sequence identification based on the buffer distances.

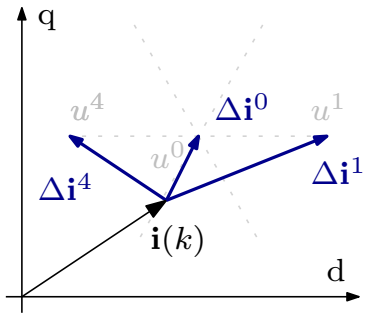


Figure 5.7: Sequence 6 current variations representation.

reported in [8]. For instance, one may consider the buffer of voltages equal to $(\mathbf{u}^4(k-2), \mathbf{u}^2(k-1), \mathbf{u}^1(k))$. The sequence identification procedure described in Sect. 5.2.1 result in the identification of sequence 2. Thus, the remaining four elements of the LUTs current variations, i.e. the ones corresponding to the voltage vectors u^0, u^3, u^5 and u^6, can be updated by means of the calculations reported at the second row in Table 4.

It is worth highlighting that sequence 6 does not allows the reconstruction of the remaining four elements. It represents a linearly dependent combination with two opposite active vectors and a zero voltage vector. Therefore, LUTs are not updated in that occurrences.

However, these combinations are still useful for the sake of current variations updating. For the sake of clarity, an example of sequence 6 $(1, 4, 0)$ and related current variations is reported in Fig. 5.7. The current variation $\Delta\mathbf{i}^0$ can be derived as the mean of the variations caused by the voltage vectors u^1 and u^4:

$$\Delta\mathbf{i}^0 = \frac{\Delta\mathbf{i}^1 + \Delta\mathbf{i}^4}{2} \tag{5.9}$$

The oldest current variation among $\Delta\mathbf{i}^0$, $\Delta\mathbf{i}^1$ and $\Delta\mathbf{i}^4$ is updated by a new value obtained by manipulating (5.9).

Table 4: Updating relationships for the current variations LUTs updating

Sequence	Rule	Rule	Rule	Rule
Seq. 1	$\Delta i^0 = \Delta i^1 + \Delta i^3 - \Delta i^2$	$\Delta i^4 = 2\Delta i^0 - \Delta i^1$	$\Delta i^5 = 2\Delta i^0 - \Delta i^2$	$\Delta i^6 = 2\Delta i^0 - \Delta i^3$
Seq. 2	$\Delta i^0 = \frac{1}{2}(\Delta i^1 + \Delta i^4)$	$\Delta i^3 = \Delta i^2 + \Delta i^4 - \Delta i^0$	$\Delta i^5 = 2\Delta i^0 - \Delta i^2$	$\Delta i^6 = 2\Delta i^0 - \Delta i^3$
Seq. 3	$\Delta i^4 = 2\Delta i^0 - \Delta i^1$	$\Delta i^3 = \Delta i^2 + \Delta i^4 - \Delta i^0$	$\Delta i^5 = 2\Delta i^0 - \Delta i^2$	$\Delta i^6 = 2\Delta i^0 - \Delta i^3$
Seq. 4	$\Delta i^4 = 2\Delta i^0 - \Delta i^1$	$\Delta i^2 = \Delta i^1 + \Delta i^3 - \Delta i^0$	$\Delta i^5 = 2\Delta i^0 - \Delta i^2$	$\Delta i^6 = 2\Delta i^0 - \Delta i^3$
Seq. 5	$\Delta i^0 = \frac{1}{3}(\Delta i^1 + \Delta i^3 + \Delta i^5)$	$\Delta i^2 = 2\Delta i^0 - \Delta i^5$	$\Delta i^4 = 2\Delta i^0 - \Delta i^1$	$\Delta i^6 = 2\Delta i^0 - \Delta i^3$

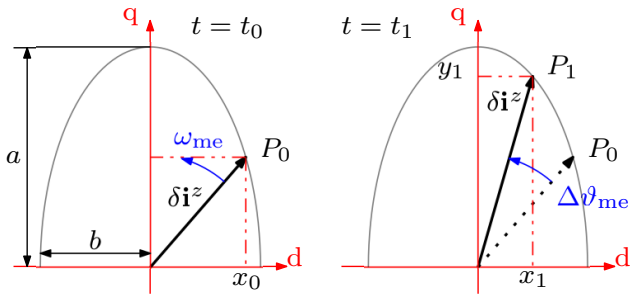


Figure 5.8: Projections of a vector rotating at a constant speed when $L_q < L_d$

5.2.3 *Compensation of the* dq *reference frame rotation*

Let consider a steady state working condition. The forced current responses components $[\delta i_d^z \; \delta i_q^z]^T$ induced by the z-th stationary voltage vector are sinusoidal (5.6), as the z-th voltage dq projections $[\delta u_d^z \; \delta u_q^z]^T$. Moreover, during steady state operation, a current variation due to u^0 and measured at any $(k-m)$ step, with $m > 1$, is equal to the one measured at $(k-1)$, without any further approximation. These considerations can be used to compensate the effect of the rotor rotation on the estimated $\Delta \mathbf{i}^z$.

At every T_c, the current measurement updates the most recent element of the triplet. The remaining two elements are older, e.g. they could be 2, 3 or more control periods old. Therefore, the current reconstruction of all the other current variations by means of the technique proposed may be affected by an error if the position variation effect is not properly compensated. An effective and simple anti-ageing technique is proposed in this section.

The current variations due to active vectors are sinusoidal and they depend on the electromechanical position ϑ_e. Therefore, it is possible to correct them by taking advantage of this property. After the anti-age compensation, every element of the triplet will be considered to be one T_c old.

First of all, it is worth reminding that $\delta \mathbf{i}^0$ is supposed to be known. Thus, the forced response $\delta \mathbf{i}^z(k-1)$ can be extrapolated also in the model-free approach:

$$\delta \mathbf{i}^z(k-1) = \Delta \mathbf{i}^z(k-1) - \delta \mathbf{i}^0(k-1) \tag{5.10}$$

The rotations of the vectors $\mathbf{u}^z$ describe a circle in the dq voltage plane, whereas the rotations of vectors $\delta \mathbf{i}^z$ describe an ellipse in the

dq current plane, due the motor magnetic anisotropy (Fig. 5.8). In particular, the ellipse semi-major axis of length a is placed along the lower inductance q-axis. The ellipse semi-minor axis of length b is thus placed along the d-axis.

The measured currents variations $\delta i^z(k-m)$, with $m \geqslant 2$, can be used to calculate $\delta i^z(k-1)$. Actually, the problem consists into estimating the future projections $(x_1(\vartheta_e), y_1(\vartheta_e))$ of a vector rotating at constant speed, starting from a previous position $(x_0(\vartheta_e), y_0(\vartheta_e))$ and knowing the time or angular displacement $\Delta\vartheta_{me}$ between the points (Fig. 5.8):

$$\Delta\vartheta_e = \omega_e(k-m+1)T_c \tag{5.11}$$

During each control period the motor speed is considered as a constant. In principle, the projections $(x_1 = \delta i_d^z(k-1), y_1 = \delta i_q^z(k-1)$ could be calculated as follows:

$$\delta i_d^z(k-1) = \delta i_d^z(k-m)\cos(\Delta\vartheta_e) - \delta i_q^z(k-m)\frac{a}{b}\sin(\Delta\vartheta_e) \tag{5.12}$$

A similar expression holds for the computation of $\delta i_q^z(k-1)$. Unfortunately, the ratio a/b is unknown, and it also depends on the current level that influences the motor magnetic saturation. Not excluding more sophisticated solutions, the easiest way is to neglect the term multiplied by the ratio, taking advantage of the fact that $\Delta\vartheta_e$ is very small in one control iteration. As a consequence, the projections can be approximated as follows:

$$\begin{cases} \delta i_d^z(k-1) \approx \delta i_d^z(k-2)\cos(\omega_e T_c) \\ \delta i_q^z(k-1) \approx \delta i_q^z(k-2)\cos(\omega_e T_c) \end{cases} \tag{5.13}$$

The compensation proposed in (5.13) was adopted. It is worth highlighting that the compensation requires the computation of just one cosine function and two multiplications.

5.3 RESULTS AND DISCUSSION

The experimental test bench adopted in this section consists of a SynR motor connected to an isotropic surface permanent magnet motor acting as a virtual load. Two different SynR prototype motors were considered, whose plate data have been reported in Table 5. The control drive algorithm was implemented in a *MicroLabBox* dSpace hardware. The control period T_c was set at 100 µs and it corresponds to the updating time of the voltage vector output. The bus voltage was set at 300 V.

The schematic of the proposed model-free-based AC drive is reported in Fig. 5.9. The proposed model-free algorithm was also implemented in simulation to gain additional information which ease

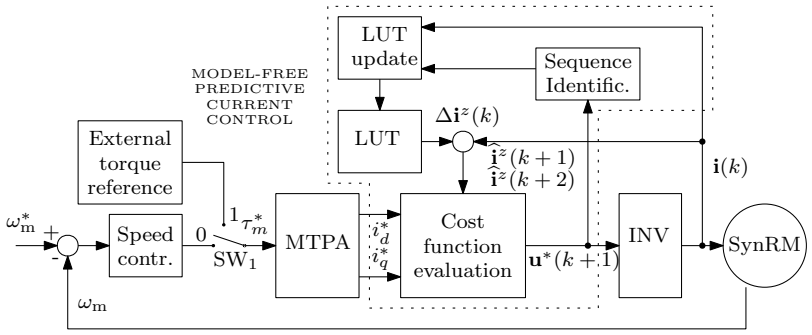


Figure 5.9: Model-free predictive current control (dotted) in an electric drive. The state of switch SW_1 determines the drive operating conditions.

Table 5: Nameplate data of the motors under test

Motor Data	Symbol	SyR$_1$	SyR$_2$
Pole pairs	p	2	2
Phase resistance	R	16 Ω	4.7 Ω
Direct inductance	L_d	1 H	0.4 H
Quadrature inductance	L_q	0.3 H	0.08 H
Nominal current	I_N	3 A	5.6 A
Nominal speed	ω_N	500 rpm	1500 rpm
Nominal torque	τ_N	8.1 N m	10.2 N m

the demonstration. The motor models implemented in the simulation were comprehensive of the magnetic saturations by means of LUTs as in Fig. 5.1.

In order to prove the feasibility of the proposed model-free approach, it is important to demonstrate the usefulness of the sequence identification. To this aim, several simulations were carried out and two different results are reported in the histograms of Fig. 5.10. The simulation was relative to motor SyR$_1$ (Table 5) and state $SW_1 = 0$ in Fig. 5.9, i.e. with an active speed control loop, at steady state and no load. The remarkable utilization of sequence 6 indicates the importance of considering also that sequence, neglected in [20]. The measured quantities are reported in *per unit* (p.u.) to ease the discussion and results comparisons. In particular, the currents were normalized with respect to $I_N/\sqrt{2}$ due to the adopted Maximum-Torque-per-Ampere (MTPA) strategy. The speed measurements were normalized with respect to the rated value ω_N.

5.3.1 *Anti-stagnation capability*

The anti-stagnation capability of the model-free predictive current control with the reconstruction method described in Sect. 6.2 is evaluated for the motor named SyR$_1$ (Table 5). Two model-free predic-

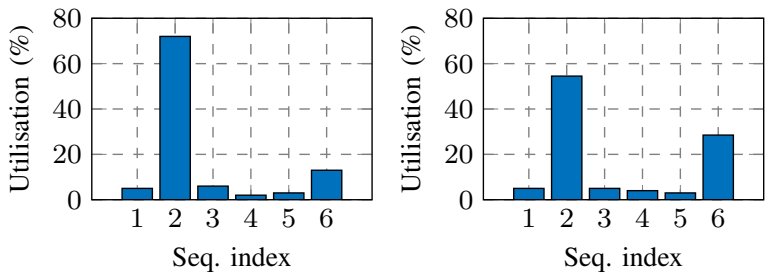


Figure 5.10: Sequences during 1 s observation at steady-state and no load (simulation with motor SyR_1, $SW_1 = 0$).

tive current control methods are compared: first the one proposed in [20], then the one presented in Sect. 6.2. The quantization of the current measurements was also implemented. The results are reported in Fig. 5.11. The former method suffers of small stagnation effects, which results in straight segments in Fig. 5.11 (top figure). Actually, it was found out that Seq. 6 was applied during those intervals. Under the same operating conditions, the simulation was repeated by implementing the proposed model-free algorithm that includes the sequence 6 and the rotation compensation. The results are reported at the bottom of Fig. 5.11, where the stagnation effect is effectively reduced and an example of LUT content is also highlighted. In the same figure, one can note the presence of small bumps, for example at $\vartheta_e \approx \pi/2$. It has been found that they depend on measurement noise in the currents. They may affect one or more $\Delta\mathbf{i}^z$ of the triplet that are used to estimate the current variations due to the remaining voltage vectors (not included in the triplet), according to Table 4. The linear combinations of Table 4 can either emphasize or reduce the effects of the current errors (originally present in the measurements) and associated to a specific voltage vector in Figure5.11. These bumps are transient situations that are readily fixed by the self-repairing feature of the model-free technique, within one electrical period of the rotor position. In other respects, the effect of bumps itself can be limited by both an higher switching and a more accurate current sensing.

5.3.2 *Prediction error maps*

This sub-section discusses the current prediction error in different working conditions. Several simulations are carried out and the current prediction of the model-based and model-free schemes are obtained for both motors of Table 5. The prediction error is calculated as follows:

$$\varepsilon\% = \left\| \frac{\hat{\mathbf{i}}(k+1) - \mathbf{i}(k+1)}{I_N} \right\| \cdot 100 \tag{5.14}$$

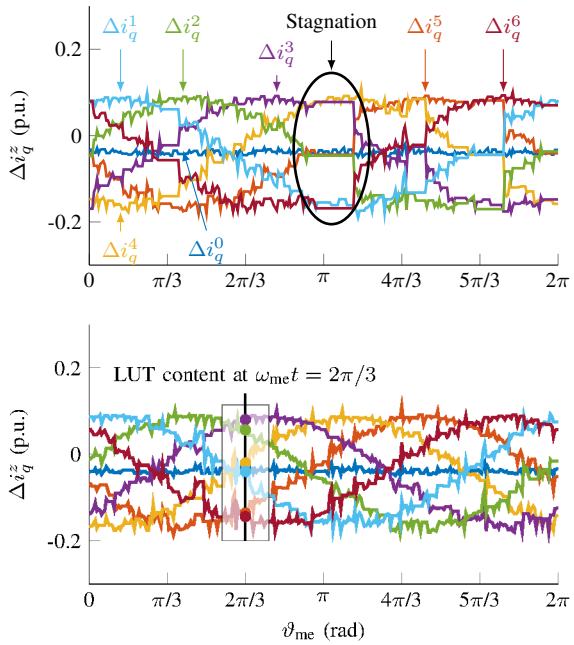


Figure 5.11: Time evolution at constant speed of the LUT for the q-axis current without (top) and with (bottom) considering Seq. 6 (SyR_2, simulation, $SW_1 = 0$).

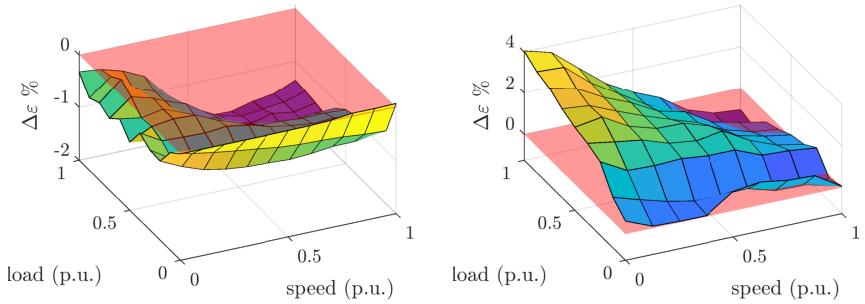
Fig. 5.12 shows the normalized difference defined as:

$$\Delta\varepsilon\% = \varepsilon\%_{\text{model-based}} - \varepsilon\%_{\text{model-free}} \tag{5.15}$$

That is, a positive value (e.g. $\Delta\varepsilon = 2\%$) in a given normalized working point means that the prediction obtained by model-based control is worse of 2% with respect to the prediction obtained by model-free control. The comparisons were carried out for both SyR_1 (Fig. 5.12a) and SyR_2 (Fig. 5.12b). The results of Fig. 5.12a establishes that for SyR_1 the model-based prediction is always better than the model-free one. Conversely, the result of Fig. 5.12b relative to SyR_2 exhibits large portions of speed-torque plane (i.e. some working conditions) in which the model-free has a prediction error fairly lower than the model-based solution. This does not establish any superiority of a method over the other, also because a rather simple model with unsaturated inductances was used. But it leaves the way open to further investigations and improvements on both sides.

For a fair comparison, the model-based predictive current control should be supported by proper motor parameters self-commissioning procedures [5, 35, 59, 65] or online parameter estimator algorithms [4, 27, 68].

As regards the model-free technique, the major weak point is that the prediction relies on past measurement, and strongly depends on rotor position. At higher speed, the past measurements become quickly obsolete. The compensation described in Sect. 5.2.3 aims at reducing this flaw. Of course, the prediction error can be mitigated



(a) SyR$_1$ (b) SyR$_2$

Figure 5.12: Normalized difference of the prediction error between model-based and model-free (simulation, $SW_1 = 1$).

by increasing the switching frequency of the inverter, as proven in [8], by accepting increasing switching losses.

5.3.3 *Results of steady-state tests*

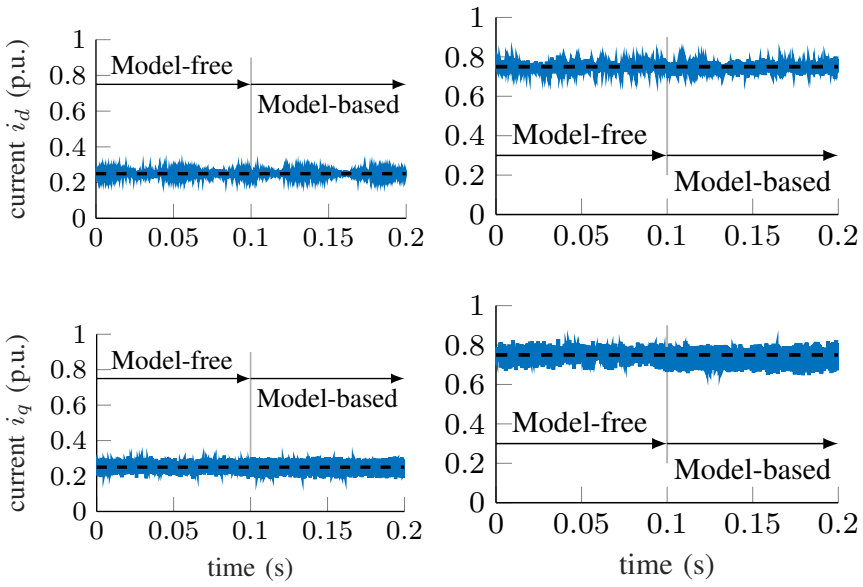
In this sub-section, a MTPA control strategy is selected [47]. Neglecting on purpose any motor saturation, an approximated MTPA strategy yields equal current references, i.e. $i_d^* = i_q^*$. In polar coordinates, the MTPA trajectory has a constant angle equal to 45°. Other choices are of course available, and in particular the one proposed in [60]. It is worth pointing out that there are MTPA techniques that do not require motor parameters knowledge, thus in favor of a full model-free electric drive. The resulting complete drive scheme is reported in Fig. 5.9.

Two different steady-state tests are carried out at various load levels and two speeds, namely 25% and 75% of the rated speed. The results are reported in Fig. 5.13 and 5.14, respectively.

During the tests, a switch from model-free and model-based predictive current control is performed. The operation at light load are reported in Fig. 5.13c and 5.14c: the transitions are smooth and there are almost no differences between the two control schemes. The obtained results are still comparable even at high load torque, as shown in Fig. 5.13d and 5.14d. On one hand, it can be seen that the model-free control exhibits a slightly higher current ripple at high speed. On the other hand, the model-based control suffers a dependence on the load torque, as expected.

5.3.4 *Results of dynamic tests*

The tests results reported in this section are obtained with the speed reference maintained by the load motor, while the motor under test is set to current control mode only ($SW_1 = 1$). A current step is imposed to one of the dq-axes. Therefore, no torque is produced and the speed



(c) $\tau_m^* = 25\%$ rated load (d) $\tau_m^* = 25\%$ rated load

Figure 5.13: Comparison between model-based and model-free predictive current control at different load values and at $\omega_m = 25\%\,\omega_N$ (SyR_1, experimental, $SW_1 = 1$).

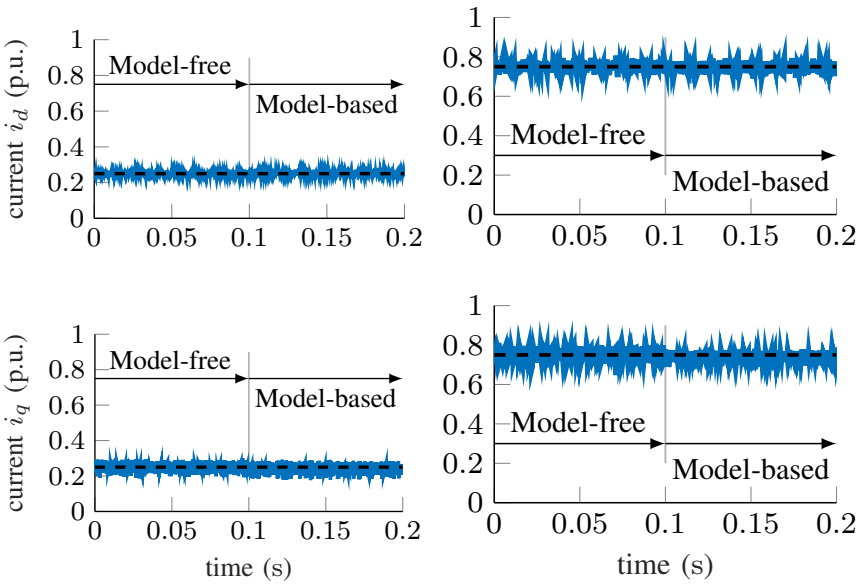
is not affected, allowing a fair comparison of the current responses between model-free and model-based approaches.

At the startup, the information stored in the current variations LUTs suffer of stagnation since no voltage vectors have been applied yet. However, as soon as one (random) voltage vector is applied on the motor, the LUTs information is updated as inherent feature of the proposed technique, just after 3 period T_c. A stable behavior of the currents is then guaranteed even at the startup of the motor drive.

The current measurements obtained for SyR_1 are reported in Fig. 5.15 and 5.16. The results of the model-based predictive current control are obtained by keeping the inductances in (2.27) at constant values reported in each figure. It is evident how the model-based predictive current control transients are influenced by parameter mismatches, whereas the model-free scheme simply has no parameters to tune. And despite its simplicity, the model-free control guarantees benchmarking dynamic performances.

To test the generality of the conclusions, the same experiment is carried out with motor SyR_2, which presents a rather different motor parameters set (see Table 5). The results are reported in Fig. 5.17 and 5.18. As the model-based control, it was verified that the proposed model-free control allows operations in the full speed and torque ranges.

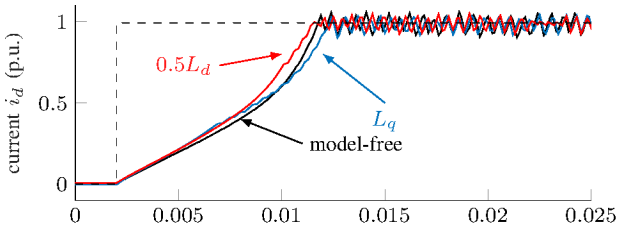
The results confirm that almost identical behavior in the dynamic is obtained by model-free and model based predictive controls. The



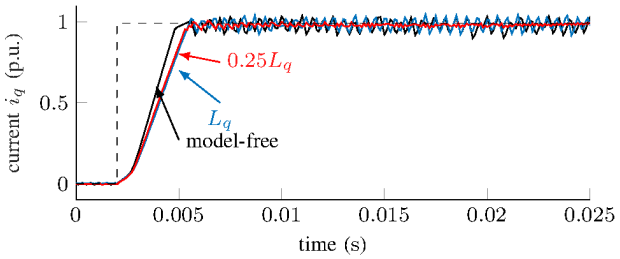
(c) $\tau_m^* = 25\%$ rated load (d) $\tau_m^* = 25\%$ rated load

Figure 5.14: Comparison between model-based and model-free predictive current control at different load values and at $\omega_m = 75\%\,\omega_N$ (SyR_1, experimental, $SW_1 = 1$).

steady state current ripple is worst in the model-free scheme compared to the case when the model-based predictive current control adopts the correct value for the inductances. However, this aspect of the model-free approach could be improved by proper design of a specific cost function in (5.1).

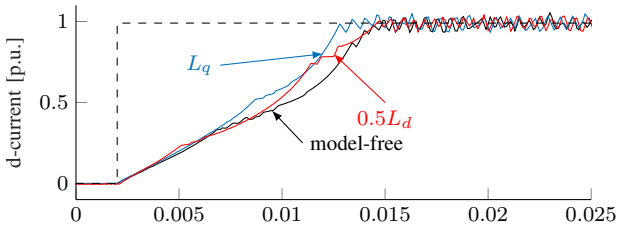


(a) Step on d-axis, $i_q^* = 0$

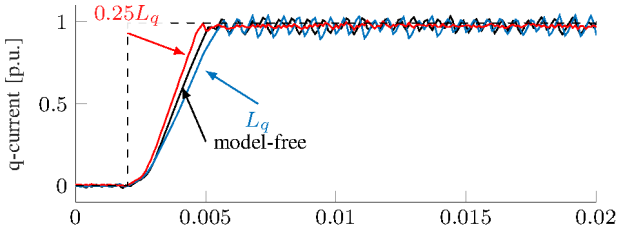


(b) Step on q-axis, $i_d^* = 0$

Figure 5.15: Current measurements with different motor parameters (SyR_1, experimental, $SW_1 = 1$, $\omega_m = 0.25\,\omega_N$).

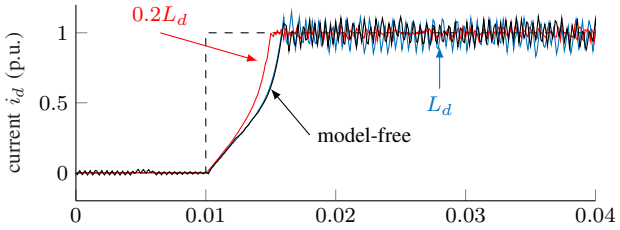


(a) Step on d-axis, $i_q^* = 0$

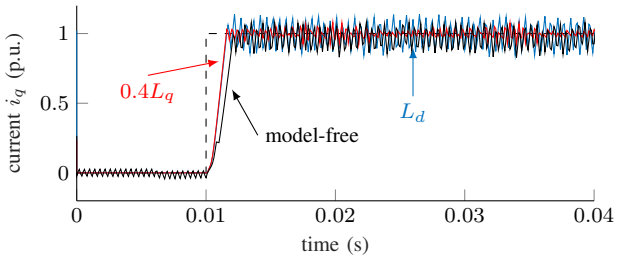


(b) Step on q-axis, $i_d^* = 0$

Figure 5.16: Current measurements with different motor parameters (SyR_1, experimental, $SW_1 = 1$, $\omega_m = 0.75\,\omega_N$).

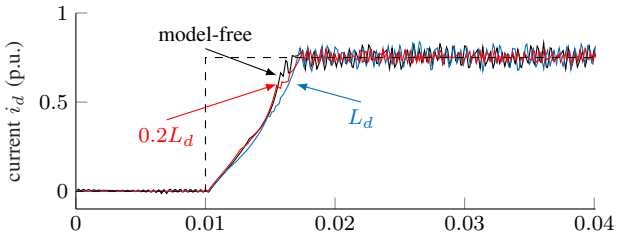


(a) Step on d-axis, $i_q^* = 0$

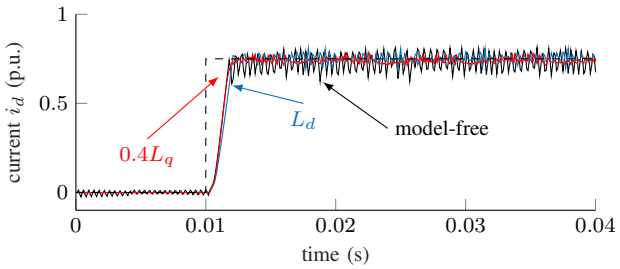


(b) Step on q-axis, $i_d^* = 0$

Figure 5.17: Current measurements with different motor parameters (SyR_2, experimental, $SW_1 = 1$, $\omega_m = 0.25\,\omega_N$).



(a) Step on d-axis, $i_q^* = 0$



(b) Step on q-axis, $i_d^* = 0$

Figure 5.18: Current measurements with different motor parameters (SyR_2, experimental, $SW_1 = 1$, $\omega_m = 0.75\,\omega_N$).

6

RECURSIVE LEAST-SQUARE ESTIMATION FOR PARAMETER-FREE CONTROL

Parameter-free predictive current control is generally based on a finite-set of voltage vectors. The first work in this field, [49], adopted LUTs for storing the current variations related to each of the eight base voltage vectors. This information is used to predict the best voltage vector to be applied, according to a predetermined cost function. As discussed in the previous chapter, the method suffered of a *stagnation* problem, due to the fact that if a voltage vector is not applied for many consecutive time steps, the stored information regarding the related current variation becomes obsolete. Two methods have been proposed to partially solve the stagnation problem so far. A *direct* method, proposed in [50], consists in modifying the cost function to force arbitrarily the application of voltage vectors not applied for long time. Of course, the arbitrary modification of the cost function acts as a disturbance in pursuing the desired control target. Conversely, *indirect* method have been proposed in the previous chapter, based on the publications [8, 12], to avoid the modification of the cost function. The LUTs update is carried out by reconstructing the current variations by specific mathematical relationships among the inverter voltage vectors at the cost of an increased complexity of the control algorithm. The information obsolescence is emphasized by both the fast changing of the operating point and the motor speed.

The research behind the solution described in this chapter is still around a parameter-free finite-set MPC current controller, but the problem of stagnation is definitely overcome by a new approach to the current prediction. To this aim, the model for the current prediction is derived by a particular recasting of the standard dq voltage equation.

The parameter-free method requires four coefficients, that are estimated by taking advantage of the information nested in the past measured current variations induced by known voltage vectors. A Recursive Least Squares (RLS) algorithm carries out the task of adapting the model parameters runtime. The theoretical background behind this solution refers to grey box modeling AppendixB. The sensitivity of the current prediction on both the operating point and the speed is greatly reduced.

6.1 SET-UP OF THE NON-PARAMETRIC MODEL

A finite-set MPC current controller requires a model of the current behavior to predict synchronous motors dynamic. In order to design

a finite-set MPC scheme that does not need any motor information, all the parameters used in the prediction phase have to be on-line identified.

To ease the mathematical representation, only the d-axis equation of a synchronous motor is considered. Equation of the q-axis dynamic is identical and lead to similar results, as summarized at the end of this Section.

$$\begin{aligned} u_d &= Ri_d + \frac{d\lambda_d(i_d, i_q)}{dt} - \omega_e \lambda_q(i_d, i_q) \\ &= Ri_d + l_d(i_d, i_q)\frac{di_d}{dt} - \omega_e \lambda_q(i_d, i_q) \end{aligned} \tag{6.1}$$

The magnetic cross-coupling between d and q axes is neglected.

The motor current dynamic is commonly described on the basis of the discretized voltage balance in the dq synchronous reference frame. In turn, the discretization of the current derivative yield $di_d/dt \approx \Delta i_d(k)/T_c = (i_d(k) - i_d(k-1))/T_c$, where T_c is the control sampling time. The last of (6.1) can be discretised at time kT_c as follows:

$$\frac{\Delta i_d(k)}{T_c} = -\frac{Ri_d(k) - \lambda_q(k)\omega_e(k)}{l_d(k)} + \frac{u_d(k)}{l_d(k)} \tag{6.2}$$

where $\lambda_q(k) = \lambda_q(i_d(k), i_q(k))$ and $l_d(k) = l_d(i_d(k), i_q(k))$. Equivalent equations can be written also for the q-axis. To ease the mathematical representation the time step dependence (k) is omitted in the rest of the chapter, whereas preference is given to highlight the dependence of the motor parameters and quantities on the dq currents.

The d-axis current increment (6.2) can be represented as the sum of two terms. The first one is obtained by applying a null voltage to the system:

$$\delta i_d^0(i_d, i_q, \omega_e) = -\frac{RT_c}{l_d(i_d, i_q)} i_d + T_c \omega_e \frac{\lambda_q(i_d, i_q)}{l_d(i_d, i_q)} \tag{6.3}$$

The second term is obtained by the application of one of the other six (active) voltage vectors of the inverter:

$$\delta i_d^f(i_d, i_q, \vartheta_e, \nu) = \frac{T_c}{l_d(i_d, i_q)} \frac{2U_{dc}}{3} \cos\left((\nu - 1)\frac{\pi}{3} - \vartheta_e\right) \tag{6.4}$$

where ϑ_e is the electric angle, U_{dc} is the DC-bus voltage and $\nu \in \mathcal{U} \triangleq [1 \dots 6]$ is the index of the applied voltage vector. The overall current variation in (6.2) can be expressed as:

$$\Delta i_d(k) = \delta i_d^0(i_d, i_q, \omega_e) + \delta i_d^f(i_d, i_q, \vartheta_e, \nu) \tag{6.5}$$

By putting

$$p_{1,d} \triangleq \delta i_d^0(i_d, i_q, \omega_e); \qquad p_{2,d} \triangleq \frac{T_c}{l_d(i_d, i_q)} \frac{2U_{dc}}{3} \tag{6.6}$$

the d-axis current variation (6.5) can be rearranged as follows:

$$\Delta i_d^\nu = \left[1,\, \cos\left((\nu-1)\frac{\pi}{3} - \vartheta_e\right)\right] \left[p_{1,d},\, p_{2,d}\right]^T = \boldsymbol{\phi}^\nu\, \mathbf{p}_d \tag{6.7}$$

The vector of coefficients $\mathbf{p}_d$ is defined as $[p_{1,d},\ p_{2,d}]^T$. It is worth pointing out that the regressors $\boldsymbol{\phi}^\nu$ change with the considered voltage vector ν. Finally, the current variations can be calculated by means of (6.7) without prior knowledge of the motor parameters, provided that the elements of $\mathbf{p}_d$ are known.

Similarly, the q-axis current variations are calculated as:

$$\delta i_q^0(i_d, i_q, \omega_e) = -\frac{RT_c}{l_q(i_d, i_q)} i_q - T_c \omega_e \frac{\lambda_d(i_d, i_q)}{l_q(i_d, i_q)} \tag{6.8}$$

$$\delta i_q^f(i_d, i_q, \vartheta_e, \nu) = \frac{T_c}{l_q(i_d, i_q)} \frac{2U_{dc}}{3} \sin\left((\nu-1)\frac{\pi}{3} - \vartheta_e\right) \tag{6.9}$$

An identical approach is used to obtain the $\mathbf{p}_q$ coefficients that describe the q-axis current variations, i.e. $\Delta i_q^\nu = \delta i_q^0 + \delta i_q^f$, leading to:

$$\Delta i_q^\nu = \left[1,\, \sin\left((\nu-1)\frac{\pi}{3} - \vartheta_e\right)\right] \left[p_{1,q},\, p_{2,q}\right]^T = \boldsymbol{\phi}^\nu\, \mathbf{p}_q \tag{6.10}$$

6.1.1 *The Recursive Least Square Estimator*

The elements of $\mathbf{p}_d$ and $\mathbf{p}_q$ have to be estimated online. For the sake of duality, only $\mathbf{p}_d$ estimation is discussed, bearing in mind that $\mathbf{p}_q$ can be estimated using the same approach. It is worth highlighting that $\mathbf{p}_d$ is time variant. In particular, $p_{1,d}$ depends both on the operating speed and magnetic iron saturation, see (6.3), whereas $p_{2,d}$ depends only on magnetic iron saturation, see (6.4).

The RLS technique is one of the most widespread method for estimating parameters during normal operations of a process [70]. It is particularly suited for applications that require adaptability to different working conditions. For these reasons it was adopted for estimating $p_{1,d}$ and $p_{2,d}$ that are the working point dependent coefficients.

The standard algorithm consists of a set of equations that can be solved recursively:

$$\begin{aligned} \mathbf{G}(k) &= \mathbf{Q}(k-1)\boldsymbol{\Phi}^T(k)(\boldsymbol{\Phi}\mathbf{Q}(k-1)\boldsymbol{\Phi}^T(k) + f\mathbf{I})^{-1} \\ \widehat{\mathbf{p}}_d(k) &= \widehat{\mathbf{p}}_d(k-1) + \mathbf{G}(k)\ (\mathbf{y}(k) - \boldsymbol{\Phi}(k)\widehat{\mathbf{p}}_d(k-1)) \\ \mathbf{Q}(k) &= \frac{1}{f}(\mathbf{Q}(k-1) - \mathbf{G}(k)\boldsymbol{\Phi}(k)\mathbf{Q}(k-1)) \end{aligned} \tag{6.11}$$

The estimated parameter vector $\widehat{\mathbf{p}}_d(k)$ is computed recursively minimizing the error between measured current variation stored in $\mathbf{y}$ and the model presented in (6.7). The same set of equations apply for the estimation of $\widehat{\mathbf{p}}_q(k)$, too. Matrix $\mathbf{G}(k)$ is the gain matrix and it

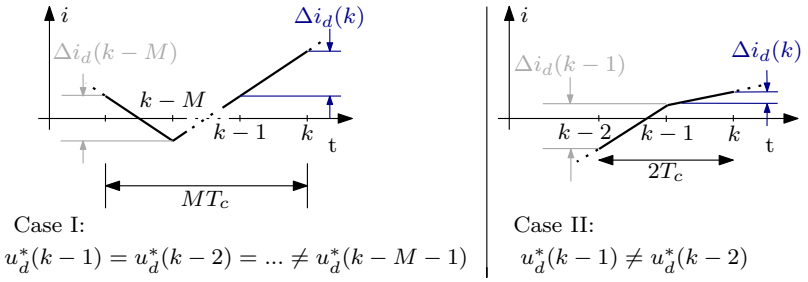


Figure 6.1: Different cases evaluation for the regressors vector ϕ calculation.

weights the error between measurements and estimations in the coefficients vector updates. The regressors matrix $\mathbf{\Phi}(k) = [\boldsymbol{\phi}^v(k), \boldsymbol{\phi}^v(k-1), \dots]^T$ includes all the regressors vectors $\boldsymbol{\phi}^v$ related to the measured current variations involved in the $\widehat{\mathbf{p}}_d(k)$ estimation, i.e. the ones related to the measurements $\mathbf{y}(k) = [\Delta i_d^v(k), \Delta i_d^v(k-1), \dots]^T$. Finally, $\mathbf{Q}(k)$ is the estimation error covariance matrix. A forgetting factor f is also introduced in (6.11) to properly weight the old data in the estimation, and its value determination is discussed in Sect. 6.3.2.

Being two the number of elements in $\widehat{\mathbf{p}}_d(k)$, the vector $\mathbf{y}(k)$ should have at least size 2×1 (and $\mathbf{\Phi}(k)$, accordingly, at least size 2×2). In other words, at least two current variations measurements are requested in (6.11).

There are several possibility to select the measurements $\mathbf{y}(k)$. A first solution consists of using the last two measured current variations, i.e. $\Delta i_d^v(k-1)$ and $\Delta i_d^v(k)$, independently from the voltages applied in the considered control periods. A situation that may occur is when the same voltage vector is applied in two (or more) consecutive time steps. The RLS might turn out to be an ill-conditioned problem, and unexpected results could be obtained. In turn, it is difficult to estimate two different parameters from very similar measurements.

Another solution is to exploit the last measured current variation $\Delta i_d^v(k)$, whereas the second variation $\Delta i_d^v(k-M)$ is measured M time instants before. Therefore, the vector of measurements in (6.11) is defined as:

$$\mathbf{y}(k) = [\Delta i_d^v(k), \Delta i_d^v(k-M)] \tag{6.12}$$

The value of M is equal to the number of time steps from the application of a voltage vector different from the one applied at the beginning of $(k-1)$. This corresponds to Case I of Fig. 6.1. For instance, the same voltage vector can be applied for M consecutive time steps. Thus, the previous voltage vector different respect to the one applied at $(k-1)$ can be found at $(k-M-1)$.

Adopting the second strategy, the RLS algorithm works within a variable length time window. The minimum length is equal to two, as shown in the Case II of Figure 6.1. It happens when two different

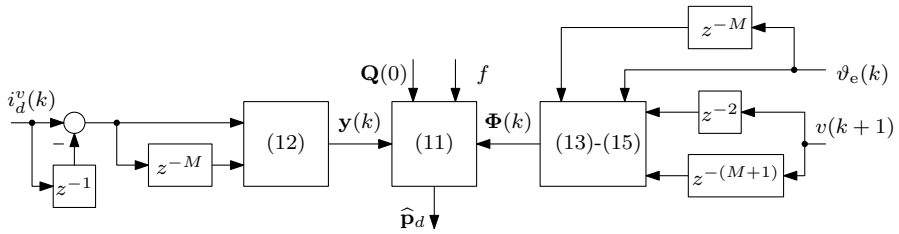


Figure 6.2: Implementation of the RLS algorithm for estimating $\widehat{\mathbf{p}}_d$. The same applies for $\widehat{\mathbf{p}}_q$ estimation.

voltages are applied in two consecutive time steps. The last applied voltage vector gives:

$$\boldsymbol{\phi}^{v}(k) = \left[1,\ \cos\left((v(k-1)-1)\frac{\pi}{3} - \vartheta_e(k)\right)\right] \tag{6.13}$$

whereas the last different voltage vector $v(k-M-1)$ returns:

$$\boldsymbol{\phi}^{v}(k-M) = \left[1,\ \cos\left((v(k-M-1)-1)\frac{\pi}{3} - \vartheta_e(k-M)\right)\right] \tag{6.14}$$

so that the regressors vector actually used in (6.11) is:

$$\mathbf{\Phi}(k) = [\phi^{v}(k),\ \phi^{v}(k-M)] \tag{6.15}$$

The outlined algorithm is summarised in the block schematic of Fig. 6.2.

In steady state conditions, the estimated $\widehat{\mathbf{p}}_d(k)$ values are not influenced by the time window length. The free response (6.3) and the amplitude of(6.4) are constant when i_d, i_q and ω_e are constant.

The estimation of $\widehat{\mathbf{p}}_d(k)$ is critical during transients, in particular when Case I of Figure6.1 occurs. The old measurement $\Delta i_d(k-M)$ carries information of the system in a previous operating point. Therefore, an error on the coefficients $\widehat{\mathbf{p}}_d$ estimation occurs. It is hard to draw a theoretical analysis on this side effect, since many different cases can happen. Therefore, this aspect is discussed by means of simulations and experiments in Sec. 6.3. Furthermore, it is expected that the choice of the forgetting factor f affects the transient behavior of the RLS algorithm.

6.2 MOTOR PARAMETER-FREE PREDICTIVE CONTROL ALGORITHM

The aim of this section is to show how the current predictions can be calculated without the knowledge of the motor parameters. The d-axis current variation can be obtained by (6.7) using $\widehat{\mathbf{p}}_d$, which can be calculated as in (6.11), instead of $\mathbf{p}_d$. The q-axis current variation can be obtained using the coefficients $\widehat{\mathbf{p}}_q$ in (6.10). The optimal voltage vector $\mathbf{u}$ to be applied at time $k+1$ is obtained by minimising the error

$$\boldsymbol{e}(k) = \mathbf{i}^{*} - \widehat{\mathbf{i}}(k+2) \tag{6.16}$$

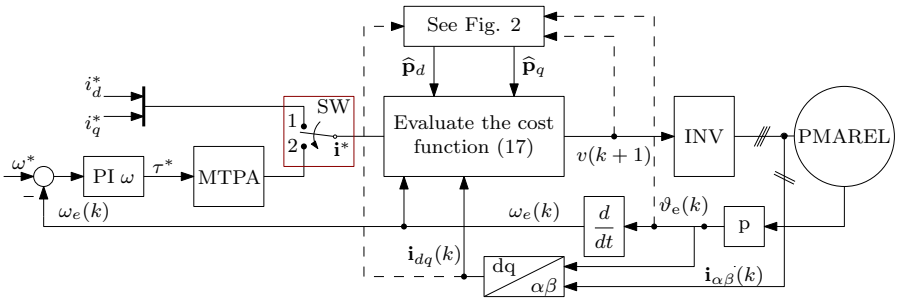


Figure 6.3: Scheme of the parameter-free predictive current control algorithm. The state of the switch *SW* determines either current (*SW* = 1) or speed (*SW* = 2) control mode.

which corresponds to the following minimisation problem

$$v(k+1) = \min_{v \in \mathcal{U}} \boldsymbol{e}^T(k) \cdot \boldsymbol{e}(k) \tag{6.17}$$

where $\mathbf{i}^*$ is the vector current references at time step $(k+1)$ and $\widehat{\mathbf{i}}(k+2)$ is the vector of the current estimates. In this work, the prediction horizon is set at 2 to compensate for the computation delay in the digital implementation of the finite-set model predictive control algorithm.

The first step to solve the problem in (6.17) is computing the current estimate at time $(k+1)$. The voltage vector $\mathbf{u}(k)$ is known and $\widehat{\mathbf{i}}(k+1)$ can be calculated applying (6.7) just once for each of the dq currents.

The second step is to calculate the current estimates at time $(k+2)$. Since the voltage vector has not been decided yet, it is necessary to evaluate the current prediction for all the voltage vectors in the set $\mathcal{U}$. The current estimates $\widehat{\mathbf{i}}(k+2)$ can be calculated by applying (6.7) for each of the voltage vectors in $\mathcal{U}$. The solution that satisfies the problem in (6.17) is thus selected and the voltage vector reference $\mathbf{u}(k+1)$ is obtained accordingly. In other words, the output of the problem (6.17) is the index $v(k+1)$, i.e. the index of the voltage vector to be applied at time instant $(k+1)$.

It is worth noting that (6.7) requires the electrical position ϑ_e. Since the voltage reference is actually applied at time $(k+1)$, it is correct to use the electrical position at time $(k+1)$. However, $\vartheta_e(k+1)$ is unknown. It is possible to estimate its value by imposing that the speed variation during one time step T_c is negligible. Therefore, the position $\vartheta_e(k+1)$ can be estimated assuming a constant speed within the prediction horizon, i.e. using a linear extrapolation:

$$\widehat{\vartheta}_e(k+1) = \vartheta_e(k) + \omega_e(k) T_c \tag{6.18}$$

The scheme of the proposed motor parameter-free predictive current control is summarised in Fig. 6.3 . The updating logic for the value of M is quite simple: the condition $v(k-1) \neq v(k-M)$ must always be satisfied and M must be as low as possible. It is straightforward that the minimum value of M is 2.

Table 6: Parameters of the motor under test

Motor Data	Symbol	Values
Pole pairs	p	2
Phase resistance	R	4.6 Ω
Direct inductance	L_d	0.16 H
Quadrature inductance	L_q	0.45 H
Permanent Magnet flux-linkage	Λ_{pm}	0.12 V s
Nominal current	I_N	6 A
Nominal speed	ω_N	700 rpm

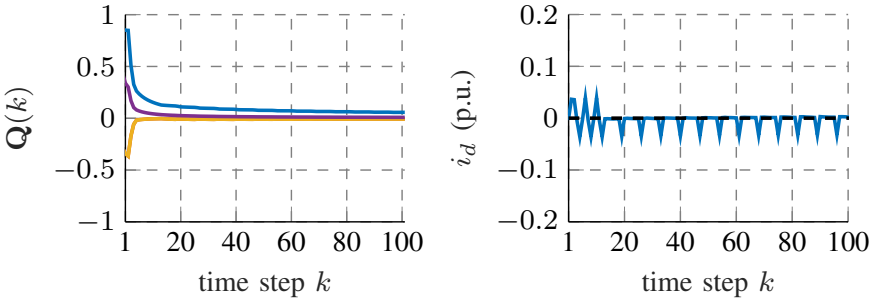


Figure 6.4: d-axis covariance $\mathbf{Q}$ and current at the startup.

6.3 PARAMETER-FREE MPC: EXPERIMENTAL RESULTS AND DISCUSSION

The proposed control scheme has been implemented by means of a dSPACE MicroLabBox board. The rig layout consists of two motors mechanically coupled. The motor under test is PMAREL synchronous motor. A second motor is mechanically coupled to the PMAREL motor and acts as a programmable load. Motor parameters are provided in Table 6 All the tests were carried out with a constant DC bus of 300 V.

The currents responses in the following results are reported in p.u.. Where it is not specified, the normalization quantities correspond to the MTPA operating point at nominal current, i.e. $(i_d, i_q) = [-4.42\,\text{A}, 4.05\,\text{A}]$. The speed results are also normalized with respect to the nominal speed value.

6.3.1 *Initialization of the RLS algorithm*

At the startup, the RLS algorithm in (6.11) is initialized with a null estimated coefficient vector, i.e. $\widehat{\mathbf{p}}_d(0) = [0; 0]$, and an identity covariance matrix $\mathbf{Q}(0)$. This is a common solution when *a priori* information are not available [70]. Motor nameplate parameters could be exploited to initialize the estimators, if available. However, this would go slightly in contrast with a complete parameter-free approach.

Figure 6.4 (left) reports the evolution of the covariance matrix $\mathbf{Q}$ elements at the startup. The random initialization of the covariance matrix does not influence significantly the current transients, as shown in Figure 6.4 (right). It is worth noting that the application of an inappropriate voltage vector following a wrong current prediction has limited effects on the current transient, both in time and amplitude. Actually, the RLS algorithm modifies the covariance matrix to correct the prediction. The availability in few control periods T_c of a reliable prediction assures quite limited effects on the controlled current at startup. Anyway, the amplitude on the current transients is strictly related to the ratio between the bus voltage and the inductances of the motor, as inferable from (6.4). The higher the ratio, the smaller the control period required to get satisfactory performances.

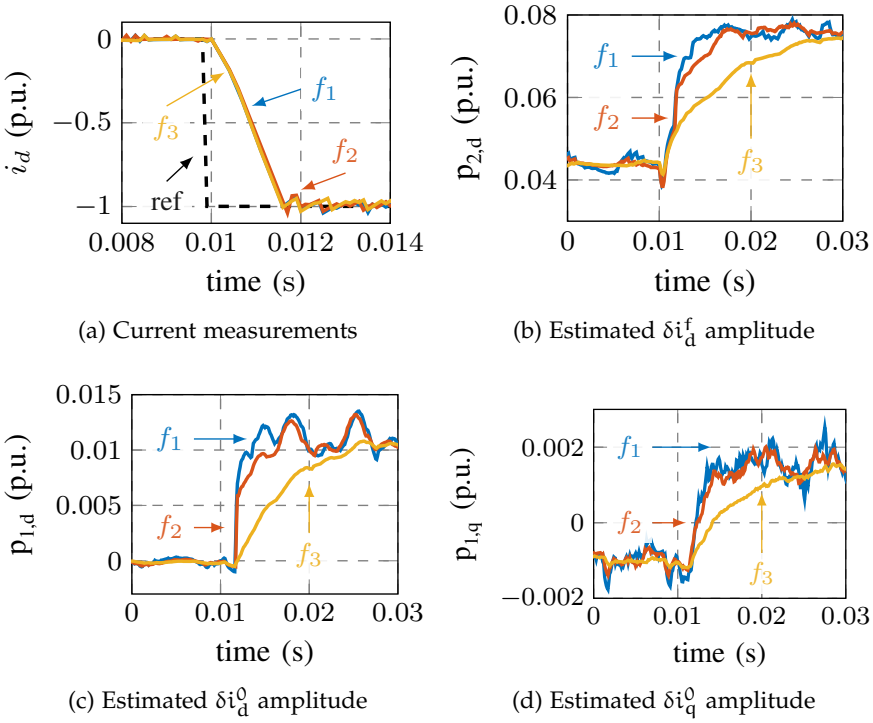
As a general remark, large inductances are quite normal in synchronous motors with dominant reluctance torque component as in the present case. Nevertheless, there can be a large variability in the inductance value for other motor topologies. As one can observe from (6.11), the updating of the covariance matrix $\mathbf{Q}(k)$ is influenced only by the regressor vector $\mathbf{\Phi}$ and the forgetting factor f and not by motor parameters. After few control periods one can assume that the prediction algorithm works correctly. The dynamics of the current control are still influenced by the motor inductances, but their values are not relevant for a successful motor startup.

6.3.2 *Tuning of the forgetting factor*

The only parameter that has to be tuned in the proposed scheme is the forgetting factor f in (6.11). The PMAREL motor is dragged at its nominal speed Ω_N by the load motor. The switch *SW* in Fig. 6.3 is in the state 1. Three current steps, each with a different value of forgetting factor f, are performed on the d-axis. The results are reported in Fig. 6.5a. Similar results can be obtained on the q-axis.

The coefficients evolution during the current steps are reported in Fig. 6.5b, Fig. 6.5c and Fig. 6.5d. The coefficients values were normalized with respect to the nominal current I_N. It is worth noting that in Fig. 6.5 the coefficients do not start from zero since the recording started when the motor was already in the steady state condition. For the sake of showing the complete transient of the coefficients, a longer time window was used compared to the one in Fig. 6.5a.

The forgetting factor value influences directly the promptness of the coefficients $\mathbf{p}_d$ and $\mathbf{p}_q$ estimation. In fact, it can be noticed from in Fig. 6.5b that the higher the value of f, the slower the estimation of $p_{2,d}$. This is an intrinsic feature of the RLS algorithm in (6.11): a value of f close to one means that the oldest measurements are equally weighted to estimate the actual value of the coefficients. On the contrary, smaller values of f force the RLS algorithm to consider only the



(a) Current measurements

(b) Estimated δi_d^f amplitude

(c) Estimated δi_d^0 amplitude

(d) Estimated δi_q^0 amplitude

Figure 6.5: Tuning of the forgetting factor f: d-axis current step with $f_1 = 0.9$, $f_2 = 0.94$, $f_3 = 0.98$.

latest measurements, but increasing the value of the covariance $\mathbf{Q}$ as inferable from (6.11).

The delay in the case of $f = 0.98$ does not affect significantly the d-axis current dynamics reported in 6.5a. This is due to the fact that the differential inductance l_d does not change considerably during the current step (see (6.4)).

An interesting result of the test in Fig. 6.5 is obtained by observing $\hat{p}_{1,q}$ in Fig. 6.5d. Before the current step, i.e. at zero dq-axes currents, $p_{1,d}$ is null, while $p_{1,q}$ is not null. This is due to the back-electromotive force induced by the permanent magnets as inferable from (6.8), since $\lambda_d(0,0) = \Lambda_{pm}$. As a final remark, the data collected during the current step have shown that the current prediction error is not significantly affected by the choice of the forgetting factor, at least within the normal range adopted in RLS algorithms.

6.3.3 *Influence of speed on the estimation*

It is worth recalling that parameters $p_{1,q}$ and $p_{1,d}$ are also strongly influenced by the operating speed as inferable from (6.3) and (6.8), respectively. The PMAREL motor speed is controlled by means of a PI speed regulator, which corresponds to the switch position $SW = 2$ in Fig. 6.3.

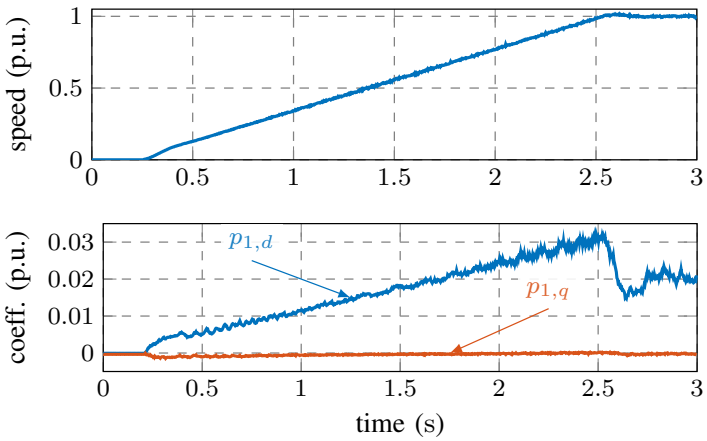


Figure 6.6: Estimation of $p_{1,d}$ and $p_{1,q}$ during a speed ramp.

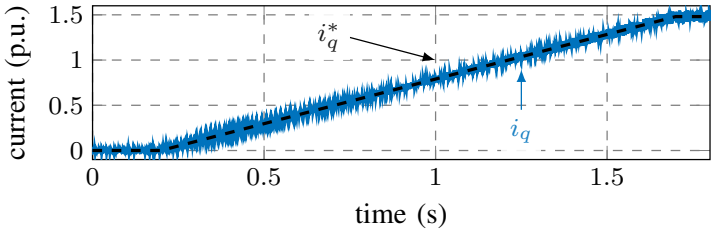
A speed ramp transient is commanded at no load condition from zero till the nominal speed. The results are reported in Fig. 6.6. The d-axis coefficient $p_{1,d}$ increases linearly with the speed because of the motional term in (6.3). The coefficient $p_{1,q}$ increases as well, but with a rather different slope. It is quite straight from (6.8) that the current variation δi_q^0 value depends on the differential inductance l_q and the d-axis flux λ_d. However, l_q is considerably higher than l_d and thus the slope of the current variation is smaller. A higher slope of the coefficient $p_{1,q}$ is obtained by increasing the permanent magnet flux linkage, such as the case of interior permanent magnet motors. The experiment has also revealed that a speed variation increases the noise of the current prediction error, but the compensation given by (6.18) assures that the error remains within a range of $\pm 1\%$ of the rated current.

6.3.4 *Influence of current on the estimation*

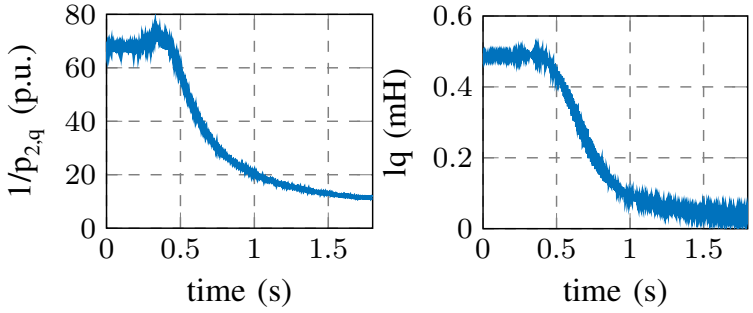
In order to show the load influence on the coefficients $\hat{\mathbf{p}}_d$ and $\hat{\mathbf{p}}_q$ estimation, a q-axis current ramp test was carried out. The result is reported in Fig. 6.7. The speed was maintained by the load motor, and the motor under test was control in torque mode by selecting $SW = 1$ in Fig. 6.3.

The coefficient $\hat{p}_{2,q}$ estimation is reported in Fig. 6.7b and it allows to draw interesting considerations. Its value changes with the current, that is the terms in (6.9) are changing with the load. Therefore, the value of $\hat{p}_{2,q}$ is strongly related to the value of the differential inductance l_q. A visual comparison between the coefficient $\hat{p}_{2,q}$ and the parameter l_q in Fig. 6.7b and 6.7c, respectively, confirms this relationship.

In case of unknown motor applied to the drive, it is possible to find out a magnetic anisotropy by simply observing the values of $\hat{p}_{2,d}$ and



(a) q-current tracking



(b) Estimated δi_q^f amplitude (c) q-axis differential inductance

Figure 6.7: Estimation of δi_q^f amplitude during a q current ramp: $\Omega = \Omega_N$, $i_d^* = 0$ p.u..

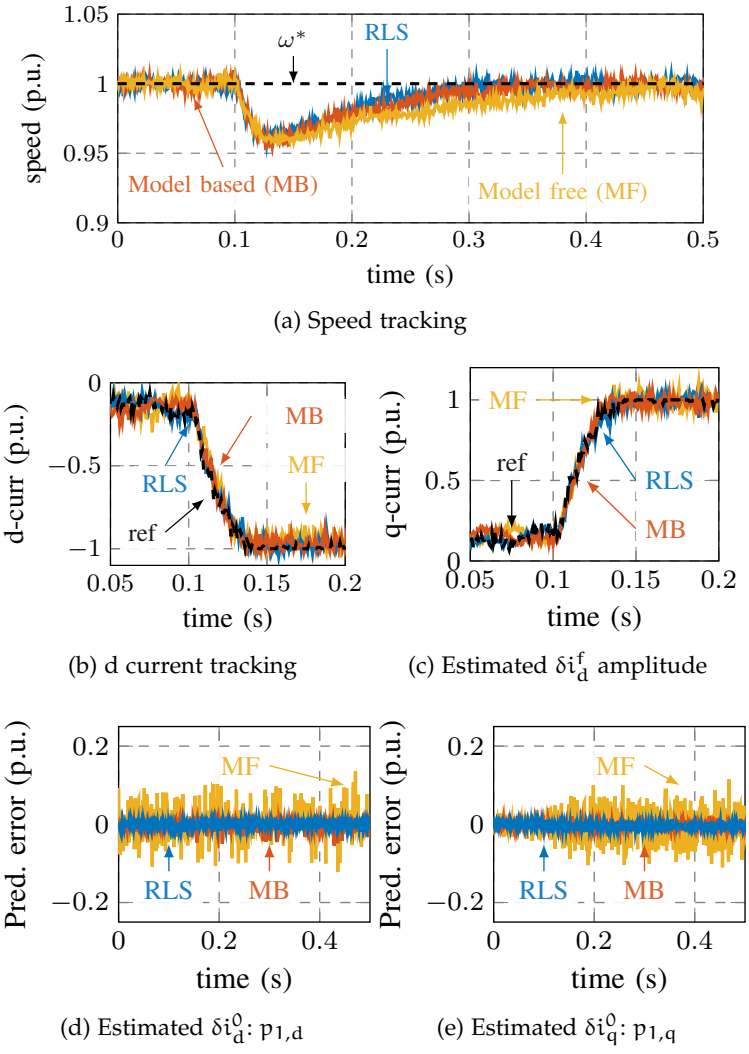
$\widehat{p}_{2,q}$. Different magnetic characteristics imply different inductances and thus different coefficients. This feature of the proposed algorithm goes toward the evolution of plug-and-play drives.

The experiment led also to the conclusion that the current prediction errors remain negligible because the RLS algorithm is able to track efficiently the variation of the differential inductance due to the moving working point (see Fig. 6.7).

6.3.5 *Comparison between the control algorithms*

For the sake of generality, the presented algorithm is compared with other two different predictive control algorithms. It was recently proposed a new control paradigm including the *model-free* concept [8, 12, 49, 50]. The parameter-free technique is inherently close to the model-free paradigm and the comparison between them is proposed. In particular, the technique proposed in [8] was used as term of comparison. Furthermore, the results of the same tests obtained with a model-based MPC controller are reported, too. Differential inductances and motor fluxes are not constant in the model, such as in (6.2). The implemented model-based MPC featured an interpolation algorithm to obtain the correct magnetic fluxes based on the dq-axes currents. The stator resistance variation is neglected in the implementation.

A load step test was carried out at different speeds. The motor under test was controlled in speed control mode, i.e. $SW = 2$ in Fig. 6.3. The load reference is thus set by the PI speed regulator and the cur-



(a) Speed tracking

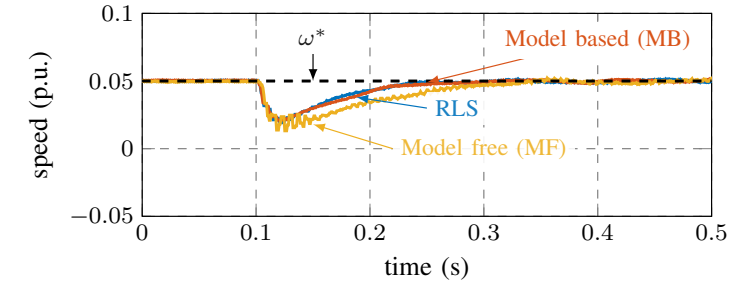
(b) d current tracking

(c) Estimated δi_d^f amplitude

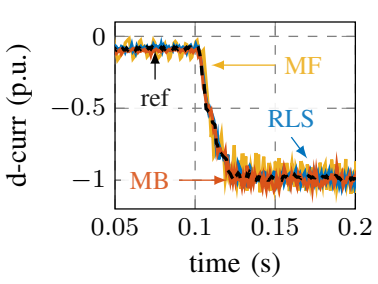
(d) Estimated δi_d^0: $p_{1,d}$

(e) Estimated δi_q^0: $p_{1,q}$

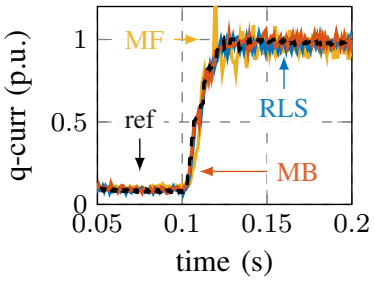
Figure 6.8: Comparison between the finite-set current controllers at the nominal point.



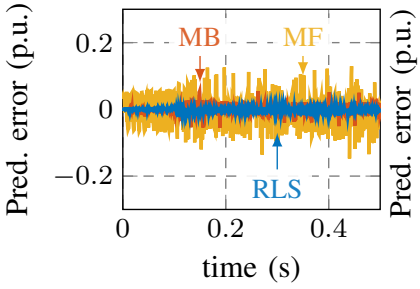
(a) Speed tracking



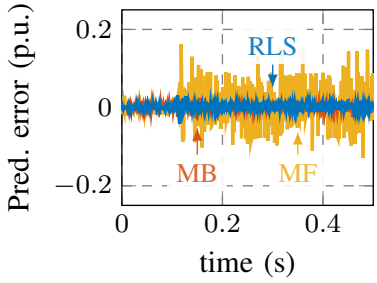
(b) d current tracking



(c) Estimated δi_d^f amplitude



(d) Estimated δi_d^0: $p_{1,d}$



(e) Estimated δi_q^0: $p_{1,q}$

Figure 6.9: Comparison between the finite-set current controllers at the $5\%\Omega_N$.

rent references are obtained by means of the MTPA curve. Two different speed values were considered aiming at showing the performances of the proposed algorithm along the whole nominal speed range of the motor under test.

A high-speed test was carried out by maintaining the nominal speed. The nominal load was applied step like and the results of speed and currents measurements are reported in Fig. 6.8a and 6.8b, respectively. It is known from literature [8] that the model-free solution reduces its performances for increasing speed and load. The prediction error is calculated by the difference between the predicted and measured currents and the results are reported in Fig. 6.8d and Fig. 6.8e for d- and q-axes, respectively. The improvements in the current prediction of the parameter-free technique respect to the model-free ones are quite relevant. Finally, the prediction errors of the proposed algorithm are comparable with the ones obtained by the model-based MPC solution. The results are surprisingly quite similar, even though the latter solution remains slightly superior.

Comparable results are obtained also at low speed (Fig. 6.9), where the model-free solution of [8] is less affected by the speed-dependent terms, but it is still dependent on the load. The prediction error results of Fig. 6.9d and 6.9e confirms the better accuracy of the proposed solution. Furthermore, the results of the proposed solution are very close to the model-based MPC ones.

In general, the proposed solution allows to sensibly improve the performances of the model-free solution. This is due to the filtering effect of the RLS algorithm in (6.11) which permits a rather better current variation estimation. Under the light of the results in Fig. 6.8 and 6.9, it is reasonable to say that a more sophisticated estimation scheme, such as the RLS, justifies the efforts of constructing a proper model of the system under consideration. The key-point of the proposed technique is to consider only the current variations and describe them by means of an equivalent model using a description similar to the free and forced response of the motor currents.

7 CURRENT RIPPLE MINIMIZATION

One of the main limitations of the parameter-free algorithm derives from the FS nature of the controller, for instance the control frequency could be kept low due to computational power limitation. The control frequency is often chosen to be equal to the frequency of switches state updating. The cardinality of the voltage vector set is thus limited to the eight base voltage vector Figure 2.8, and the problem of finding the optimum solution is reduced. However, if the control frequency is not high enough, the currents harmonic distortion could be significant. Of course, increasing the switches state adaptation frequency, thus the control frequency, results in a higher switching frequency. In addition, a high current ripple induces torque oscillation, which result in acoustic noise and vibrations in the system.

In order to improve the current harmonic content, the switches state updating frequency should be increased. However, the control frequency should be decoupled from the switches state updating one to keep the computational effort at bay. This is the underlying concept of the DSVM introduced in [14]. According to the DSVM, the control period is divided in a discrete number of sub-periods. Any of the eight inverter base voltage vectors can be applied in each sub-period. According to a benchmark DSVM, the voltage vectors synthesized by the modulation are mapped in LUT, which are usually built offline [95].

In the MPC framework, the DSVM is treated as a predictive controller with prediction horizon equal to the number of sub-periods. The discrete nature of the inverter switches positions implies that the MPC problem is non-deterministic Polynomial-time hard (NP-hard), i.e. it scales exponentially with the prediction horizon length. Proper solvers have been presented in literature to tackle this issue, such as the *branch and bound* and the *sphere decoding* methods [74]. The adoption of proper solvers is crucial to limit the computational effort at an affordable amounts. As an alternative, the selection of the optimal inverter switches position vector to be applied to the motor windings can be performed by exploiting both offline considerations and online optimization, such as in [62].

Finding a proper tuning of the MPC for accomplishing both a reduced harmonic content and a low switching frequency could be a quite troublesome aspect [44]. Thus, many FS-MPC solutions in literature implement a *deadbeat* version of the algorithm, [82]. Deadbeat algorithms are also very popular in the area of model-based predictive controllers [82, 94]. Parameter-free MPC algorithms massively adopt the deadbeat approach, even if not straightforwardly indicated in lit-

erature. This allows to minimize the tuning effort. For instance, the forgetting factor of two RLS estimators is the online design parameter of the controller proposed in [12].

7.1 MOTOR PARAMETER-FREE CURRENTS PREDICTION

The synchronous motor currents dynamics in (2.10) can be rewritten as follows:

$$\mathbf{i}_{dq}(k+1) = \mathbf{i}_{dq}(k) + (\mathbf{A}-\mathbf{I})\mathbf{i}_{dq}(k) + \mathbf{B}\mathbf{h}(k) + \mathbf{B}\mathbf{u}_{dq}(k) \tag{7.1}$$

where $\mathbf{I}$ is the 2×2 identity matrix. The equivalent grey-box model, introduced in the previous chapter, is derived from (7.1) suitable for online adaptation of the working point-dependent parameters. The method adopts the following adaptive model:

$$\begin{aligned} &\mathbf{i}_{dq}(k+1) = \mathbf{i}_{dq}(k) + \widehat{\boldsymbol{w}}(k) + \widehat{\boldsymbol{W}}(k)\mathbf{u}_{dq}(k) \\ &\widehat{\boldsymbol{w}}(k) = \begin{bmatrix} p_{1,d} \\ p_{1,q} \end{bmatrix}, \ \widehat{\boldsymbol{W}}(k) = \begin{bmatrix} p_{2,d} & 0 \\ 0 & p_{2,q} \end{bmatrix} \end{aligned} \tag{7.2}$$

where $p_{1,d}$, $p_{1,q}$, $p_{2,d}$ and $p_{2,q}$ are generic coefficients, not to be intended equivalent to the motor parameters of (2.10). The coefficients in (7.2) are estimated online by means of a recursive least square algorithm. The standard algorithm consists of a set of equations that can be solved recursively:

$$\begin{aligned} \mathbf{G}(k) &= \mathbf{Q}(k-1)\boldsymbol{\Phi}^T(k)(\boldsymbol{\Phi}(k)\mathbf{Q}(k-1)\boldsymbol{\Phi}^T(k) + f\mathbf{I})^{-1} \\ \widehat{\mathbf{p}}(k) &= \widehat{\mathbf{p}}(k-1) + \mathbf{G}(k)\ (\mathbf{y}(k) - \boldsymbol{\Phi}(k)\widehat{\mathbf{p}}(k-1)) \\ \mathbf{Q}(k) &= \frac{1}{f}(\mathbf{Q}(k-1) - \mathbf{G}(k)\boldsymbol{\Phi}(k)\mathbf{Q}(k-1)). \end{aligned} \tag{7.3}$$

The vector of measurements can be

$$\mathbf{y}(k) = \begin{bmatrix} i_d(k) - i_d(k-1) \\ i_d(k-1) - i_d(k-2) \end{bmatrix} \text{ when } \widehat{\mathbf{p}}(k) = \begin{bmatrix} p_{1,d} \\ p_{2,d} \end{bmatrix} \tag{7.4}$$

because the identification of two coefficients, e.g. $[p_{1,d};\ p_{2,d}]$, requires two independent set of measurements. A vector of measurements for the q-axis coefficients estimation can be obtained in the same fashion of (7.4) by replacing:

$$\mathbf{y}(k) = \begin{bmatrix} i_q(k) - i_q(k-1) \\ i_q(k-1) - i_q(k-2) \end{bmatrix} \text{ when } \widehat{\mathbf{p}}(k) = \begin{bmatrix} p_{1,q} \\ p_{2,q} \end{bmatrix}. \tag{7.5}$$

Therefore, past currents and switches configurations are exploited to estimate all the coefficients, without resorting to motor parameter information. The regressors vector $\boldsymbol{\Phi}$ is $[1;\ u_d]$ when the d-axis coefficients are estimated by means of (**??**), or $[1;\ u_q]$ alternatively. The

forgetting factor f can be chosen as reported in [73]. The gain matrix $\mathbf{G}(k)$ and the covariance matrix $\mathbf{Q}(k)$ are calculated at each time step (k) by the recursive least square algorithm. Finally, the vector $\widehat{\boldsymbol{w}}(k)$ is obtained by considering the first element of vector $\widehat{\mathbf{p}}(k)$ for each axis, whereas the two non-zero elements of matrix $\widehat{\boldsymbol{W}}(k)$ are obtained from the second element of vector $\widehat{\mathbf{p}}(k)$, again one for each axis. In order to avoid the resolution of a rank deficient least-square problem, the recursive least square algorithms work within a variable length time window (see [73] Sect. II.A).

The advantages that the proposed adaptive model brings along are manifold. First of all, the model (7.2) is quite general and it can be adopted for several synchronous motor topologies without additional commissioning effort. In other words, either permanent magnet and reluctance synchronous motors can be described by (7.2). The adaptive model modifies the coefficients values of $\widehat{\boldsymbol{W}}$ and $\widehat{\boldsymbol{w}}$ in (7.2) to track the synchronous motor model parameters variations. Thus, stator resistance and inductances variations are both taken into account. The adaptive model can replace, for instance, the use of LUTs for the inductive parameters, that are often included in the model-based MPC algorithm [77].

The current prediction at future time $k+2$ can be calculated as follows

$$\widehat{\mathbf{i}}_{dq}(k+2) = \mathbf{i}_{dq}(k+1) + \widehat{\boldsymbol{w}}(k) + \widehat{\boldsymbol{W}}(k)\mathbf{u}_{dq}(k+1). \tag{7.6}$$

Equation (7.6) can be used to predict current values at $k+2+n$, with $n \geqslant 1$, at the price of lower accuracy due to the constant value assumption of the coefficients value in $\widehat{\boldsymbol{w}}(k)$ and $\widehat{\boldsymbol{W}}(k)$. The adaptive model (7.2) relies on the coefficients adaptation to account for the innovation brought by the currents evolution.

7.2 DEADBEAT PREDICTIVE CURRENT CONTROL DESIGN

The main goal of the chapter is to propose a deadbeat predictive current control algorithm performing a reduced current ripple compared to existing motor parameter-free algorithms. The control frequency should be kept at a low value for the sake of computational burden reduction. The switching frequency is a consequence of the adopted switches state updating frequency, which is selected to be a fraction of the control frequency. Therefore, two different time scales are introduced to distinguish between *control* and *switches* state updating frequency. The former is denoted by the time index k_c, whereas the latter by k_s. The equivalent control period T_c is selected as an integer multiple of the switches state update period T_s. An example of the different time variables is reported in Fig. 7.1, where a ratio $T_c/T_s = N = 3$ was selected for the sake of a clearer representation. The choice of N will be discussed in Sect. 7.3.1.

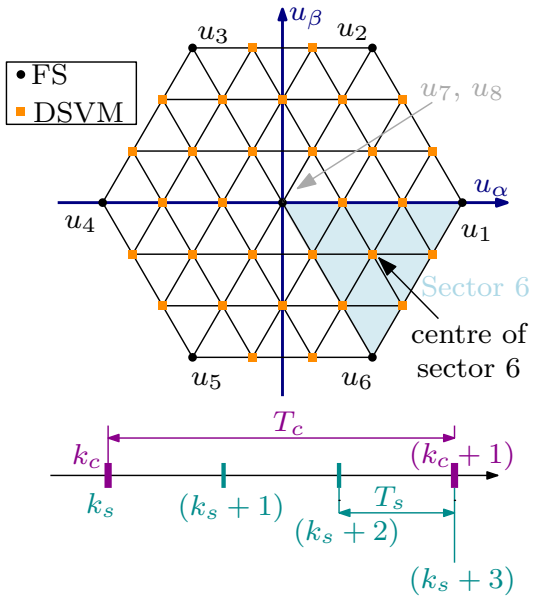


Figure 7.1: Set of αβ voltage vectors that can be synthesised using a standard FS controller and the DSVM method.

In order to minimize the tracking error, a quadratic cost function is adopted in this paper. A detailed overview of alternative cost functions formulation are reported in [44]. A prediction horizon N is used to exploit the adaptive model (7.2)

$$\begin{aligned} \min_{\mathbf{u}(\cdot)} \quad & \sum_{z=1}^{N} \|\mathbf{i}^*_{dq} - \mathbf{i}_{dq}(k_s+z)\|^2 \\ \text{s.t.} \quad & \mathbf{i}_{dq}(k_s+1) = \mathbf{i}_{dq}(k_s) + \widehat{\boldsymbol{W}}\mathbf{u}_{dq}(k_s) + \widehat{\boldsymbol{w}}(k_s) \end{aligned} \quad . \tag{7.7}$$

In turn, the solution of (7.7) consists of a sequence of switches configurations. In the original parameter-free controller [12], the receding horizon policy was adopted. Thus, the optimization problem is solved for every sampling period T_s. It is worth reminding that a digital delay of one control step occurs between the computation of the optimal control action and its application. The delay is compensated by means of an open loop prediction, as described in [74].

7.2.1 *DSVM application to the motor parameter-free controller*

A three-phase two level inverter can generate only eight base voltage vectors in the αβ stator reference frame, corresponding to the eight possible configurations of inverter switch positions. Hereinafter these vectors are denoted as: $u_1 = [1\,0\,0]^T$, $u_2 = [1\,1\,0]^T$, $u_3 = [0\,1\,0]^T$, $u_4 = [0\,1\,1]^T$, $u_5 = [0\,0\,1]^T$, $u_6 = [1\,0\,1]^T$, $u_7 = [0\,0\,0]^T$ and $u_8 = [1\,1\,1]^T$. The fundamental vectors are reported in Fig. 7.1 by means of black dots. It is worth noticing that a zero voltage vector can be generated by two

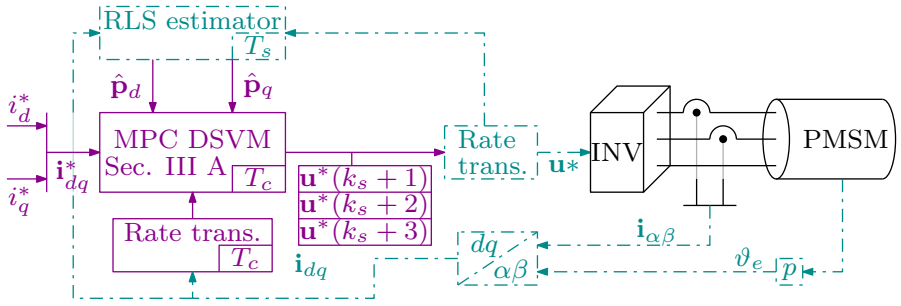


Figure 7.2: Overview of the control scheme architecture. Continuous-line (magenta) and dashed-line (cyan) indicate the T_c and T_s samples times, respectively.

switches configurations, i.e u_7 and u_8: the choice between the two of them is driven by the minimization of switching events.

According to the DSVM, the control period T_c is divided in an integer number of shorter sub-period T_s. Different base voltage vectors can be potentially applied at each one of these T_s-long sub-periods. From a predictive controller point of view, the MPC problem solution is calculated at every T_c. The RLS algorithm described in Sect. 7.1 is run with a period T_s to better track the coefficients $\hat{\mathbf{p}}$ evolution. In other words, the time variable k from (7.1) to (7.6) should be rewritten by using the time variable k_s. Therefore, the validity of equations (7.1)-(7.6) is general for any value of N.

For the sake of simplicity, it is convenient to set the prediction horizon length N equal to the number of sub-periods. The different sample rates of the main blocks that compose the proposed control scheme are sketched in Fig. 7.2. As a result, the DSVM enlarges the set of equivalent voltages that can be synthesized within a control period T_c. The augmented set of voltages is represented by orange squares in Fig. 7.1, where a number of sub-periods $N = 3$ was chosen.

As the number of sub-periods N increases, problem (7.7) becomes more and more computationally expensive. In fact, the number of equivalent vectors grows in an exponential rate, and so does the number of cost function evaluations. Moreover, each equivalent voltage vector can be obtained with different combinations of the base vectors $u_1, \ldots, u_8$. The proposed deadbeat motor parameter-free predictive controller adopts one of the most up-to-dated methods to solve the control problem, i.e. the one proposed in [62]. The idea behind this method is to separate the current tracking problem (7.7) from the selection of the optimal combination of base voltages.

The tracking problem is solved very efficiently in terms of the equivalent voltage vector. In fact, the cost function is first evaluated for the six central voltages of the hexagon sectors. An example of sector and its center is reported in Fig. 7.1, in particular *sector 6*. Once the voltage vector returning the lowest cost is found, all other vectors that belong to the same sector are evaluated. Considering 3 sub-

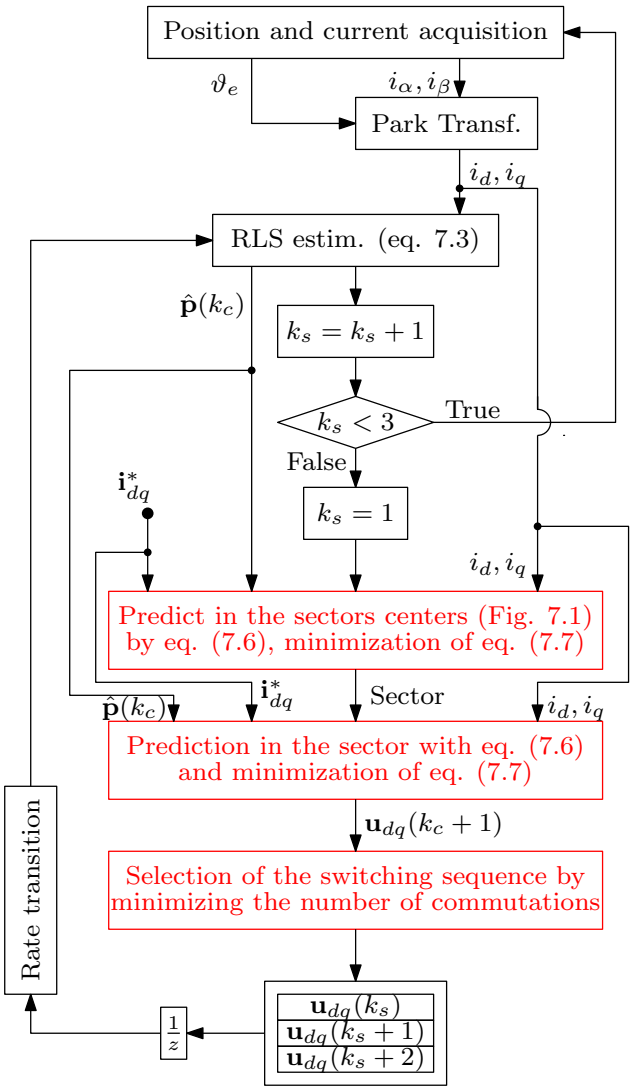


Figure 7.3: Flow chart of the DSVM parameter-free MPC (N = 3).

periods, 15 cost evaluations are computed, instead of evaluating all the 37 equivalent vectors. This number is obtained by considering the sectors centers, i.e. 6, the number of voltage vectors in the sector, i.e. 8, and only one zero voltage vector. Then, the voltage vector of the remaining set returning the lowest cost is considered and it is synthesized in such a way that the number of inverter switches commutations is minimized. The minimization of the switching events is computationally cheap due to offline considerations (maximum 4 cases need to be considered). Extensive considerations and detailed discussions are reported in [62]. The idea of separating the tracking problem from the switching minimization can be extended to a different number of sub-periods. A flow chart of the overall deadbeat parameter-free algorithm is reported in Fig. 7.3. The distinctive features with respect to previous model-free schemes discussed int this thesis are highlighted.

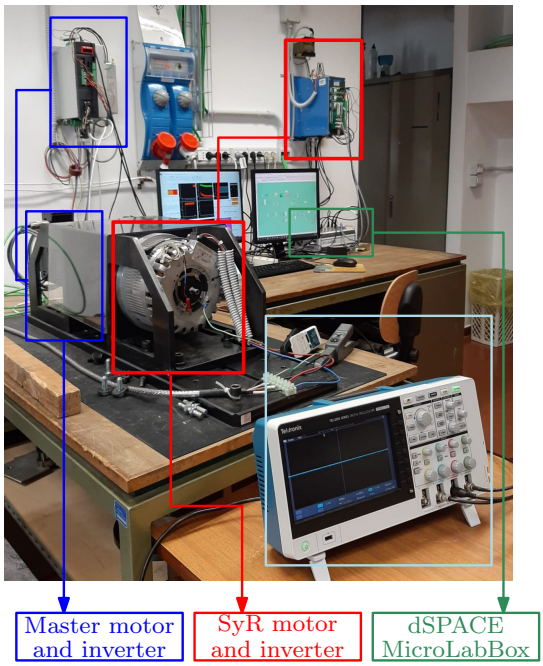


Figure 7.4: Test rig layout.

Table 7: Parameters of the prototype SyR motor under test

Motor Data	Symbol	Values
Pole pairs	p	2
Phase resistance	R	4.6 Ω
Direct inductance	L_d	0.25 H
Quadrature inductance	L_q	0.08 H
Nominal current	I_N	8.5 A
Nominal d-axis current	$I_{d,N}$	3.6 A
Nominal q-axis current	$I_{q,N}$	7.7 A
Nominal speed	ω_N	500 rpm

7.3 EXPERIMENTAL RESULTS AND DISCUSSION

The motor under test was a SyR motor, whose plate parameters are reported in Table 7. A dSPACE MicroLabBox hardware was used for the real-time implementation, featuring a 2 GHz NXP QorlQ P5020 microprocessor. The test rig is reported in Fig. 7.4. The inverter bus voltage was set at $u_{bus} = 300$ V. The SyR motor under test was dragged by a speed-controlled PMSM load motor during all tests reported in this Section.

Six control schemes were compared, three FS-MPC algorithms and three DSVM based predictive control algorithms. Model-based control schemes, i.e. the ones that adopt the parametric model of the PMSM (2.10), are denoted with the acronym Model-Based (MB). The sampling frequency of the finite-set algorithms was $f_s = 10$ kHz and

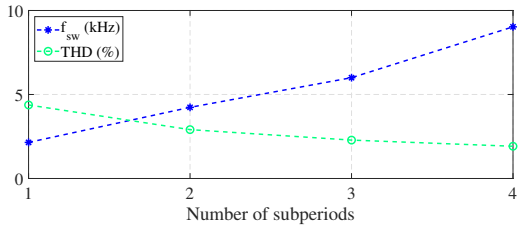


Figure 7.5: Switching frequency and current THD as function of the number of sub-periods of the DSVM.

it coincided with the control frequency. The DSVM schemes shared the same control frequency of the FS ones. However, they differed in terms of sampling frequency, which is three times higher ($N = 3$), if not specified. It is worth recalling that the oversampling was exploited by the RLS estimators to update the parameter-free adaptive model. No tuning parameters are requested in (7.7) for the predictive controllers. A forgetting factor $f = 0.98$ was selected in (6.11), following the design guidelines discussed in [73]. In conclusion, the only remaining parameter that has to be selected is the number of sub-periods, as discussed in Sect. 7.3.1.

On the one hand, two model-based MPC scheme were implemented. One exploited the nominal motor parameters for the current prediction, whereas the other a more accurate model. The model was designed to take into account the effects of the iron-saturation. Moreover, a third FS controller was implemented, i.e. the deadbeat version of the parameter-free method proposed in [73]. On the other hand, two model-based MPC schemes adopting a DSVM architecture were implemented for the sake of comparison. As for the FS case, one scheme implemented the nominal motor model, whereas the other one the more accurate model. Finally, the novel DSVM controller based on the parameter-free algorithm proposed in this paper was tested and compared to the previous alternatives.

All the quantities reported in the results are normalized with respect to their nominal value, reported in Table 7.

7.3.1 *Selection of the number of sub-periods*

The only parameter that needs to be selected when implementing the parameter-free DSVM algorithm is the number of sub-periods within one control period. This parameter was selected by means of a commissioning test of the SyR motor at its nominal operating point (Table 7). In particular, the SyR motor drive was in current control mode, while the opposite load motor was keeping the speed constant at the rated value. Four different number of sub-periods were analyzed.

During the commissioning, the average switching frequency of the inverter and the Total Harmonic Distortion (THD) of the phase cur-

rents were monitored. On the one hand, the average switching frequency reflects the losses due to the switching. The average switching frequency was measured starting from the switches states generated by the controller implemented in the dSPACE hardware. On the other hand, currents THD is selected as performance index to prove and quantify the current ripple reduction. Indeed, THD is the most adopted index by practitioners and the most widespread in literature. A lower THD results in reduced joule losses in the stator windings and a lower torque ripple. Phase currents THD was measured by means of current probes and an oscilloscope (Fig. 7.4).All the results are reported in Fig. 7.5.

The two considered performance indexes conflicts when the number of sub-periods increases. In particular, the switching frequency grows with the number of sub-periods, while the current THD has an opposite trend. After the commissioning test, the design of the controller can be carried out harmonizing it with the requirements of the application. It is selected a number of sub-periods equal to three, which i the best trade-off to minimize the current THD. Indeed, choosing four sub-periods allows a small reduction of the ripple at a price of a relevant increase of the switching. It is worth highlighting that the model-free algorithm is capable of self-adapting the coefficients $\widehat{\mathbf{p}}$ for any number of sub-periods. Therefore, the number of sub-periods can be varied even during online operations thanks to the adaptability characteristics of the proposed model-free solution, provided that sufficient computational power is available. All the results reported hereinafter regarding DSVM based algorithms adopts three sub-periods, which results in a sampling frequency of 30 kHz.

From practitioners point of view, it is of primary interest to know whether the computational burden of the designed deadbeat controller is higher than the one of a benchmark FS controller with receding horizon running at 30 kHz. On the one hand, a FS controller would solve $N = 3$ times the MPC optimization problem by evaluating the 8 base voltages reported in Fig. 7.1. It results cheaper to evaluate 7 base voltages, thus considering the two zero vectors u_7 and u_8 together. Then, if a zero vector is optimal, the one that involves the minimum switching effort is applied. In conclusion, the cost function is evaluated 21 times every T_c. On the other hand, the proposed deadbeat parameter-free method requires 15 cost evaluations every T_c, as already discussed in Sect. 7.2.1. Thus, a lower number of cost evaluations is achieved with respect to a conventional FS controller. Nevertheless, the computation time required to identify the optimal switching pattern increases by the number of sub-periods.

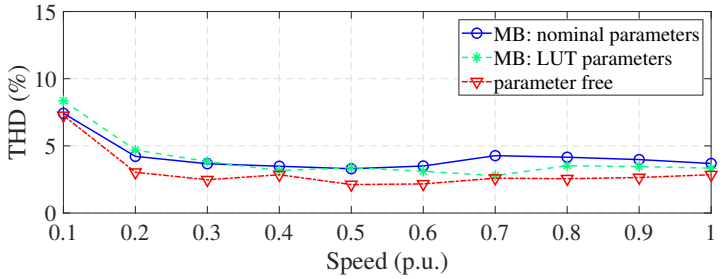
In general, it is difficult to state which solution is computationally cheaper, since it depends on the adopted hardware and on the code generation procedure, too. Therefore, the average turn-around-time of the controller for several number of sub-periods is reported

Table 8: Average turn-around-time of the considered controllers for different number of sub-periods N.

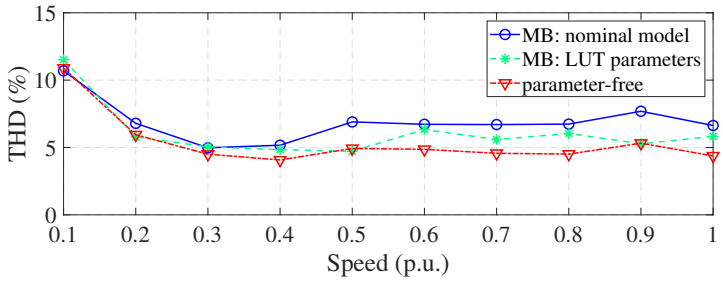
N	MB	MB+LUT parameters	parameter-free
1	10 µs	13.6 µs	13.7 µs
2	10.6 µs	14.3 µs	14.3 µs
3	12 µs	15.5 µs	15.5 µs
4	12.6 µs	16.4 µs	16.5 µs

in Table 8, as comprehensive computational burden index. This is a common choice for evaluating the computational complexity of a predictive control algorithm, such as in [1, 91]. It is worth noting that the proposed parameter-free scheme guarantees a computation time similar to the one of a model-based MPC which adopts an accurate motor model.

7.3.2 *Analysis of the phase current distortion*



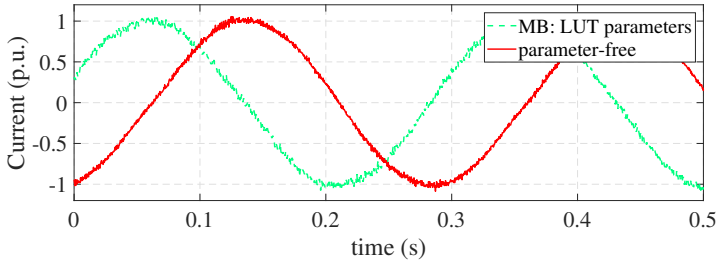
(a) DSVM controllers (N = 3).



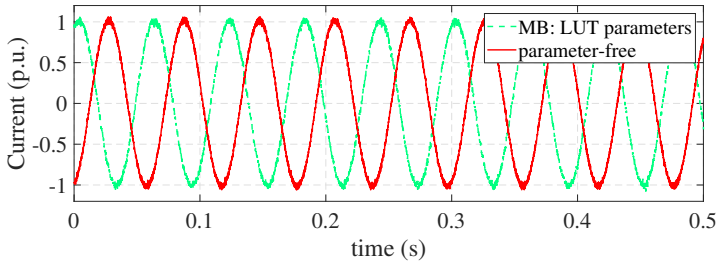
(b) FS controllers.

Figure 7.6: Comparison of the current THD of the different controllers for different operating speeds.

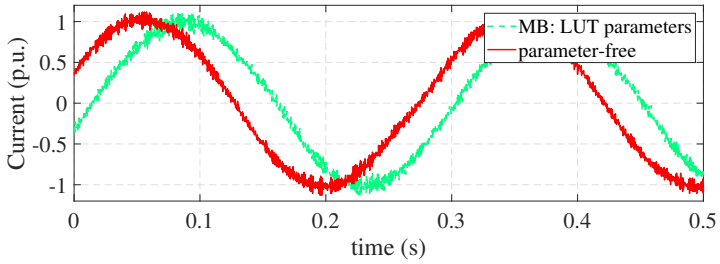
The main motivation for combining the DSVM method with the parameter-free scheme is to reduce the ripple in the motor currents. Therefore, several tests were performed in order to analyze the ripple in the entire operating speed range of the SyR motor, considering all the six predictive controllers implemented in the paper. The results



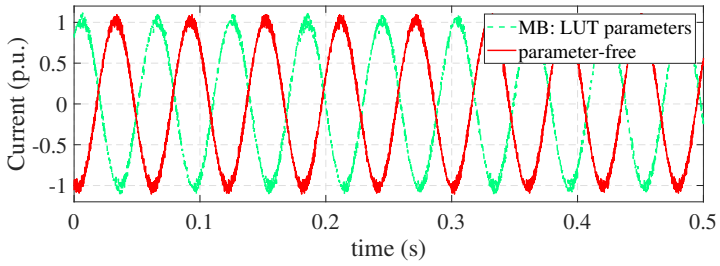
(a) DSVM (N = 3): 20% of the nominal speed.



(b) DSVM (N = 3): nominal speed.



(c) FS: 20% of the nominal speed.



(d) FS: nominal speed.

Figure 7.7: Motor phase current at steady state for two operating speed and considering different control strategies.

are reported in Fig. 7.6, categorized according to the type of modulation technique used (FS or DSVM).

As a first consideration, the current THD obtained using the DSVM technique (Fig. 7.6a) is always lower than the one obtained with the FS method (Fig. 7.6b), regardless of the working speed and the predictive control algorithm. Indeed, this is a well-known result in literature [82]. Moreover, the model-based algorithms that exploit a more accurate model outperform the ones that implement just the nominal model, regardless of the adopted modulation technique.

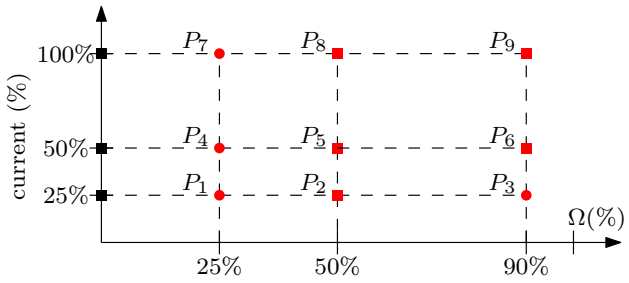


Figure 7.8: Map of the analyzed points. Squares denote the points suggested by the EN 50598-2 standard. The considered points are red-colored.

The current THD analysis for both model-based and parameter-free algorithms was performed. The parameter-free schemes achieve a current THD which is equal or lower than their model-based counterparts within the whole considered speed range. This may be justified by the fact that even the SyR motor model that accounts for the iron saturation does not fully describe the dynamics of the currents. It is worth highlighting, indeed, that the proposed accurate model could neglect some non-linear phenomena. The authors are aware that these phenomena may be included in the motor model. However, an even more accurate model requires additional commissioning effort. Moreover, the current predictions evaluations become more computationally expensive. Therefore, the simple parameter-free adaptive model appears to be a reasonable compromise between having good performances in terms of currents THD and maintaining a relatively cheap prediction model.

The phase currents are reported for two particular operating speed in Fig. 7.7, for sake of completeness. The parameter-free schemes are compared to the model-based algorithms that use the more accurate motor model. The phase shift between the current measurements is introduced on purpose for the sake of clarity. The current ripple reduction obtained thanks to the DSVM methods Fig. 7.7a-7.7b with respect to the FS ones Fig. 7.7c-7.7d is evident. The improvement in the waveform is achieved especially in the peaks of the sinusoids, where low current derivatives are required. In fact, the DSVM unlocks the possibility of synthesizing voltage vectors of lower magnitude (see Fig. 7.1). Thus, the current variations become smoother.

In order to provide quantitative results, some relevant motor operating points are considered, as shown by the grid reported in Fig. 7.8. The selected points are similar to the ones suggested by the EN50598-2 standard, which regards the efficiency evaluation of the drive system. The obtained THD results are reported in Table 9 for each point of the grid. The proposed parameter-free method exhibits promising results with respect to the model-based MPC. The percentage improve-

Table 9: THD (%) comparison between the three controllers, considering the red points of Fig. 7.8.

Point	parameter-free	MB	η	MB+LUT	η
P_1	5.7	8.2	-31	6.9	-18.1
P_2	3.6	5.8	-37	3.9	-5.7
P_3	3.0	4.9	-39	3.2	-5.9
P_4	6.0	8.7	-31	6.9	-13.4
P_5	3.8	6.0	-37	3.8	-0.5
P_6	2.7	4.6	-42	2.9	-7.0
P_7	6.9	8.9	-22	7.2	-3.8
P_8	3.8	6.1	-37	4.1	-5.7
P_9	2.6	4.4	-41	2.6	-1.9

ments of the current THD achieved by the proposed technique can be calculated as

$$\eta = \left(1 - \frac{THD_{MB} - THD_{MF}}{THD_{MB}}\right) \cdot 100 \tag{7.8}$$

The results in terms of THD differences are reported in Table 9. From the efficiency point of view, the reduction of current THD is proportional to the reduction of Joule losses cause by the additional harmonics components of the motor currents. The results of Table 9 confirms that the proposed model-free algorithm consists a remarkable improvement of Joule losses reduction compared to a model-based solution with constant parameters.

7.3.3 *Analysis of the switching frequency*

The DSVM technique permits a reduction of the current THD at a price of an increase of the average inverter switching frequency. For this reason, the current THD analysis performed in the section above is supported by an analysis of the average inverter switching frequency f_{sw}, reported in Fig. 7.9.

All the DSVM based control algorithms (Fig. 7.9a) produce a switching frequency on average higher with respect to the FS ones (Fig. 7.9b). This is, in general, a good initial results. The implemented DSVM algorithms, characterised by 3 sub-periods, allow a maximum switching frequency three times higher than the FS ones. Thus, the frequency is increased less than linearly. Moreover, it is noticed that the switching frequency of the FS algorithms always decreases when increasing operating speeds. This is mainly due to the fact that the motional cross coupling term in (2.10) is proportional to the speed. Thus, in steady state condition, the current derivatives are lower at high speed, when

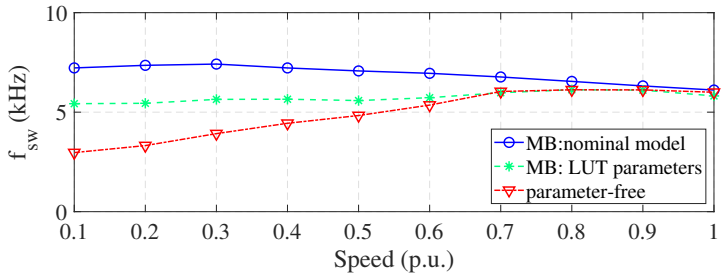
Table 10: Switching frequency (kHz) comparison between the three controllers, considering the red points of Fig. 7.8.

Point	parameter-free	MB	η	MB+LUT	η
P_1	6	6.7	-11	6.1	-1.6
P_2	2.7	7.4	-64	4.0	-34
P_3	3.5	7.1	-51	4.9	-28.4
P_4	4.7	7.2	-35	5.8	-19.2
P_5	4.6	7.3	-38	5.1	-11.4
P_6	4.4	6.8	-35	5.0	-12.7
P_7	6.2	7.1	-13	6.2	-0.2
P_8	5.2	5.9	-12	5.5	-5.1
P_9	5.8	6.1	-5	5.8	-0.2

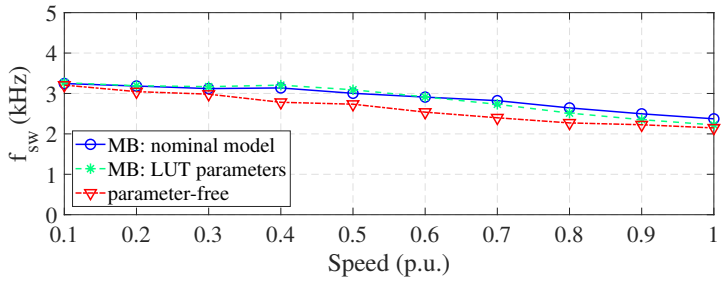
the cross coupling counterbalances the input voltage term, reducing the overall available voltage.

As for the THD analysis, the model-based controller that takes into account the saturation in the iron results as the best between the model-based solutions. However, the parameter-free paradigm grants a slightly lower switching effort than the model-based one. The most relevant improvements are achieved by the DSVM parameter-free algorithm at low speed (Fig. 7.9a). In conclusion, the currents THD and switching frequency analysis of the controllers highlights some beneficial effects of the DSVM parameter-free approach. In fact, it permits a significant reduction of the phase current harmonic content and, at the same time, the lower increase of switching frequency between the DSVM based controllers.

Quantitative results are reported in Table 10. They were obtained by considering the working points defined in Fig. 7.8. The switching frequency plays an important role in the drive efficiency, which may depend on the size of the drive and on the technology of the inverter switches. Anyway, switching losses are always proportional to the average switching frequency of the converter. The results reported in Table 10 suggest that the proposed solution allows to obtain lower switching frequencies than the model-based solution with constant parameters. Actually, the switching frequencies of the proposed model-free solution are very similar to the ones of the model-based algorithm with LUT parameters, i.e. with the correct model for each working point. Lower losses in the converter improve both the design of the cooling system and the life cycles of the switches.



(a) DSVM controllers (N = 3).



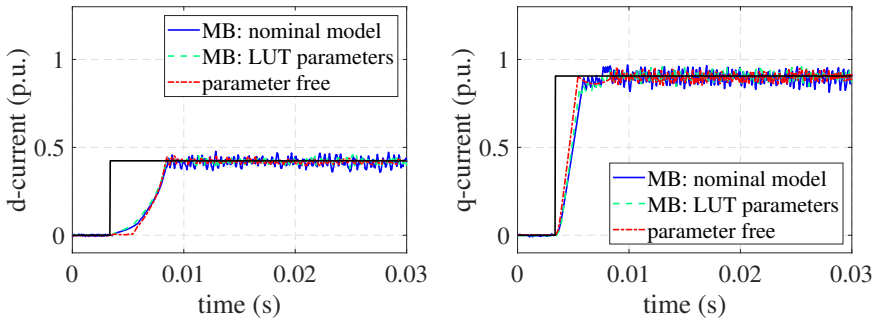
(b) FS controllers.

Figure 7.9: Comparison of the switching frequency among the different controllers for different operating speed.

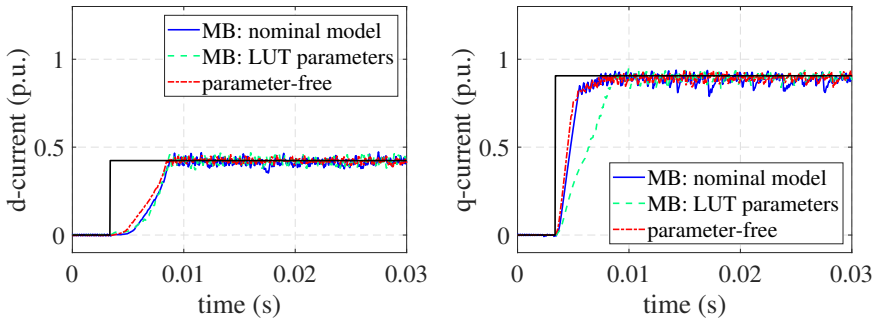
7.3.4 *Step response analysis*

The currents reference step responses are considered here to analyze the dynamic performances of the proposed DSVM parameter-free controller. All the tests are performed changing the current references step-wise from zero to the nominal point Fig. 7.10-7.12. The dynamics are repeated for two operating speed, namely at standstill and at nominal speed, covering the entire working range of the motor. The currents shape characteristics of Fig. 7.10 and 7.12 resemble the magnetic curve relationships. As a distinctive feature of every finite-set algorithm, the transient responses are as fast as possible subject to the available voltage.

From the rise time and overshoot point of view, all the algorithms assure very good performances and are very similar between all the schemes. This is, in fact, a distinguished feature of the MPC paradigm. It is worth noticing that the parameter-free exhibits a similar dynamic behavior of a model-based scheme, without any a-priori information of the motor parameter. However, the model-based controllers that adopt an accurate motor model tend to behave slightly better. In case of the parameter-free algorithms, in fact, the model learning rate is determined by the recursive least square estimator dynamics. The dynamics of the estimated coefficients during the step responses in Fig. 7.10 are reported in Fig. 7.11. The currents responses are faster than the estimators ones. Thus, the parameter-free controller is adapting the model during the transients, while the accurate model-based



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



(c) d-axis current: nominal speed. (d) q-axis current: nominal speed.

Figure 7.10: Comparison of the step responses among the DSVM model-based and parameter-free controllers at different speed.

controller exploits previously measured motor-parameters. This fact justifies the minimal reduction of performances of the parameter-free controller in terms of step response when compared to accurate model-based schemes. Nevertheless, the steady-state current ripple improvement achieved with the parameter-free approach counterbalance the limited reduction of dynamic performances. The beneficial effects of the adapted coefficients on the current ripple is thus a reduction of the THD as discussed in Sect. 7.3.2, which are comparable to a model-based predictive current control algorithm with LUTs based parameters.

7.4 SOLVERS FOR THE MPC PROBLEM

The MPC problem presented in (7.7), in the general case, may includes a penalty on the input:

$$\begin{aligned} \min_{\mathbf{u}(\cdot)} \quad & \sum_{z=1}^{N} \left|\left|\mathbf{i}^*_{dq} - \mathbf{i}_{dq}(k_s+z)\right|\right|^2 + \lambda_U \left|\left|\Delta\mathbf{u}_{dq}(k_s+z-1)\right|\right|^2 \\ \text{s.t.} \quad & \Delta\mathbf{u}_{dq}(k_s+z-1) = \mathbf{u}_{dq}(k_s+z-1) - \mathbf{u}_{dq}(k_s+z-2) \end{aligned} \tag{7.9}$$

which results in a penalization of the switching frequency of a factor λ_U. In fact, each voltage vector corresponds to a specific power switches configuration (Table 1).

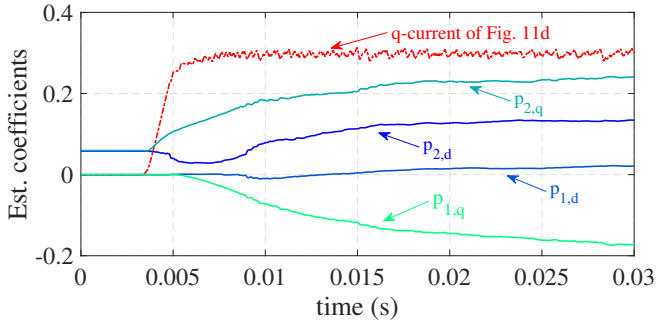


Figure 7.11: Estimated coefficients dynamics vs current dynamic.

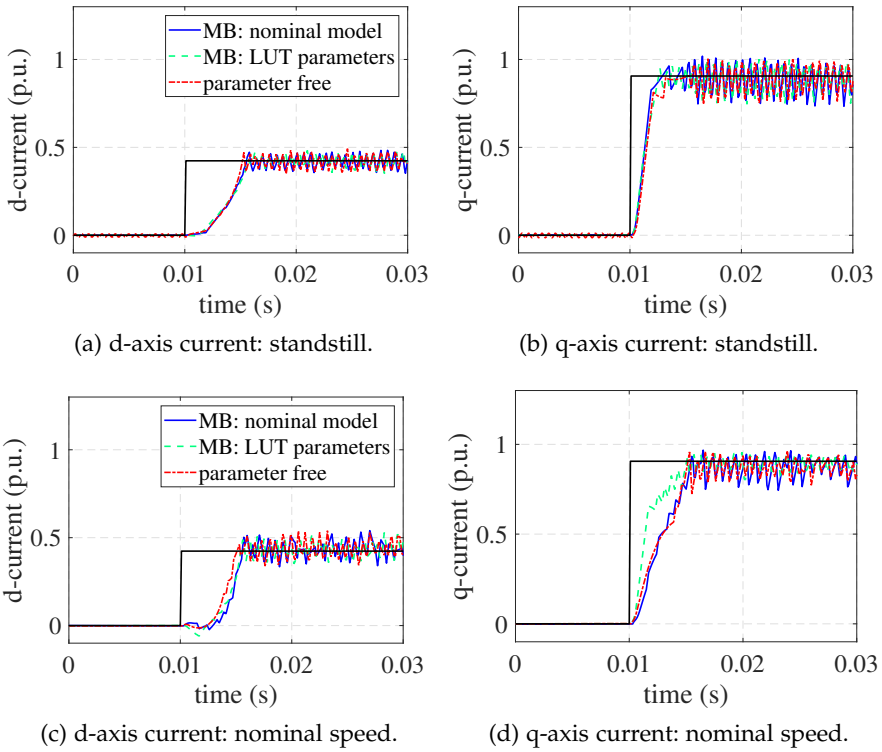
This problem, as already stated, is an NP-hard one, i. e. it is computationally expensive. A bulky and trivial solution to solve (7.9) is to use the *enumeration method*, which turns out to be too time demanding for the real-time constraints of the considered application. In fact, the cost function is evaluated for all the possible switches configurations to get the one that give the lowest cost. In details, the problem scales exponentially with the value of N. One of the eight fundamental voltages can be applied every T_s (see Fig. 7.1). Since a prediction horizon equal to 3 is used in the implemented controller, the number candidates is $8^3 = 512$.

Many general-purpose efficient algorithms are available to reduce the computational burden of the MPC scheme, see [74]. In this paper, the implementation of two solvers are analyzed, i.e. the *branch and bound* and the *sphere decoding* methods.

7.4.1 *The "branch and bound" solver*

The branch and bound technique is a popular solver for integer optimization problems. Let a *candidate* be a combination of voltage vectors to be applied during one T_c. Let ub and lb be the upper and lower bounds of an optimal cost $J(\mathbf{u}_{opt})$, respectively. The optimal solution is the one having the lowest cost among all possible solutions. The branch and bound technique allows to minimize the number of candidate solutions to be evaluated compared to the enumeration method. Upper bounds of the cost function are obtained by evaluating the cost expression for one of the voltage candidates in the set Fig. 7.1. Typically, *heuristics* are used to decide which is the first candidate to be evaluated, as it will be discussed in Sect. 7.4.3. The result of (7.9) evaluated for the chosen first candidate sets the first value of ub.

Bearing in mind that in (7.9) the cost function is quadratic, a positive contribution to the cost value is calculated for each element of the candidate under consideration. Let's consider, for instance, a solution that contains u_1 as first vector (see Fig. 7.13, i.e. u_1 is considered for the first T_s out of three. A lower bound lb_1 is calculated for such volt-



(a) d-axis current: standstill.

(b) q-axis current: standstill.

(c) d-axis current: nominal speed.

(d) q-axis current: nominal speed.

Figure 7.12: Comparison of the step responses among the FS model-based and parameter-free controllers at different speed.

age vector. If $lb_1 > ub$, all the leaves child of u_1 can be removed from further considerations. This operation is called *pruning*. On the other hand, if the cost value obtained by considering one candidate, i.e. a combination of three voltage vector for the entire T_c, is lower than the u_b, the upper bound is updated by the cost value of the same candidate. Therefore, once all the possible candidates have been evaluated, the optimal solution is found to be the one returning the lowest cost function. The pruning actions allow to reduce considerably the amount of calculations to be carried out.

7.4.2 *The "sphere decoding" solver*

The sphere decoding solver works in a slightly different way. First, the solution of problem (7.9) is computed removing the integer constraint that the solution must be one of the eight fundamental voltage vectors. Since no other constraints are considered, the solutions is unconstrained (3.10), which is much cheaper to compute. It is reminded that it is simply $\mathbf{u}_{unc}^{opt} = \mathbf{H}^{-1}\mathbf{d}$ where $\mathbf{H}$ and $\mathbf{d}$ are the hessian matrix and the linear term of the problem, respectively. More details about how the matrices are computed can be found in [74]. Second, the Cholesky decomposition of $\mathbf{H}$ is computed, i.e. $\mathbf{H} = \mathbf{V}^T\mathbf{V}$, where $\mathbf{V}$ is a lower triangular matrix. The result allows to define $\bar{\mathbf{u}} = \mathbf{V}\mathbf{u}_{unc}^{opt}$.

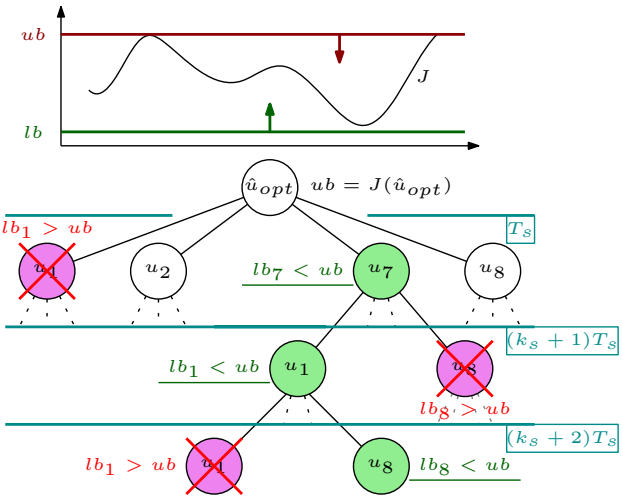


Figure 7.13: Branch and bound method principle.

The key idea behind the sphere decoding solver is that, given the $\mathbf{u}_{\text{unc}}^{\text{opt}}$, the integer solution of (7.9) $\mathbf{u}_{\text{int}}^{\text{opt}}$ is the one closest to $\bar{\mathbf{u}}$, in terms of the euclidean distance $\rho = \|\bar{\mathbf{u}} - \mathbf{V}\mathbf{u}_{\text{int}}^{\text{opt}}\|_2$. This distance, named *radius*, is used to run a branch and bound search, which works component-wise on the distance between vector $\bar{\mathbf{u}}$ and $\mathbf{u}_{\text{int}}^{\text{opt}}$. The extended description of the algorithm can be found in [74].

The main difference from the branch and bound solver is the pruning technique in the search. One can consider again Fig. 7.13, and the voltage u_1 in the first layer of branching. Its pruning implies that 64 candidate solutions are not evaluated as candidate optimum. On the other hand, let's consider the case when $\rho_1 = \|\bar{\mathbf{u}}(1) - \mathbf{V}\mathbf{u}(1)\|_2 > \rho_{\text{best}}, \mathbf{u}(1) = 0$, where ρ_{best} is the radius of the integer candidate solution closer to $\bar{\mathbf{u}}$ available so far in the search process. Since $\mathbf{u}_1 = \{0, 1\}$, the pruning of the candidate solutions starting with $\mathbf{u}(1) = 0$ exclude half of set of candidates.

7.4.3 *Heuristic methods*

Even if the branch and bound or the sphere decoding solver are implemented, a consistent number of cost evaluations could be required to find the optimum. Also, the branch and bound technique is effective provided that an initial ub value is good enough.

Heuristic techniques are considered in this paper, which still identify the optimum of problem 7.9. It is noticed that starting with a tight upper bound in the branch and bound or with small radius in the sphere decoding algorithm speed-up the pruning process, i.e. less candidate solutions are evaluated before finding the optimum. In other words, it is convenient to initialize the search process with a good candidate. In this work three heuristics are studied, as initial guess of the optimum:

Table 11: Parameters of the prototype IPM motor under test

Motor Data	Symbol	Values
Pole pairs	p	4
Phase resistance	R	$1.5\,\Omega$
d-axis inductance (unsaturated)	L_d	$0.036\,H$
q-axis inductance (unsaturated)	L_q	$0.084\,H$
Permanent magnet flux-linkage	Λ_{pm}	$0.18\,V\,s$
Nominal current	I_N	$6.6\,A$
Nominal speed	Ω_N	650 rpm

H1 the two sequences of three null voltage vectors, i.e. $(\mathbf{u}_7, \mathbf{u}_7, \mathbf{u}_7)$ and $(\mathbf{u}_0, \mathbf{u}_0, \mathbf{u}_0)$;

H2 the solution computed at previous control period, which is the most wide-spread one in case of receding horizon controllers;

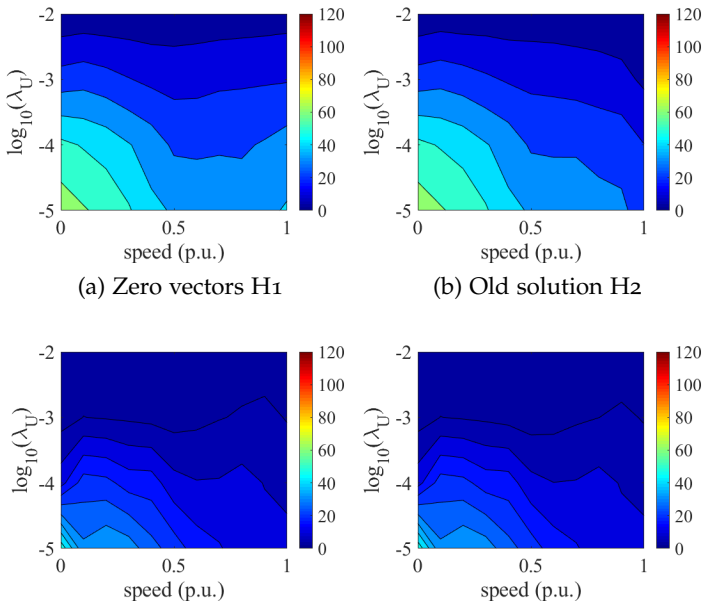
H3 the projected unconstrained solution of the MPC problem, i.e. all the switches positions are rounded to 0 or 1.

7.4.4 *Solver performances*

The solver performances are here analyzed, considering the prototype IPM motor, whose plate parameters are reported in Table 11. The sampling frequency was set at $f_s = 30\,kHz$, whereas the inverter bus voltage was $u_{bus} = 300\,V$. For the sake of simplicity, the iron saturation is neglected in the IPM motor model used in the simulations. The operating speed in the simulation results is normalised with respect to the nominal value, reported in Table 11.

Branch and bound and sphere decoding solvers applied to the proposed motor parameter-free controller

The two solvers are compared on the basis of the average number of candidate solutions evaluated before identifying the optimum of the MPC problem (7.9). Results are reported in Fig. 7.15 and Fig. 7.16, considering the IPM motor working at nominal current rate, but at different operating speed and with different weights value of the switching effort λ_U. A decrease of evaluated candidates with respect to the enumeration method is obtained by both algorithms, bearing in mind that 512 candidates should be tested with the enumeration methods. The sphere decoding results to be more efficient than the branch and bound solver. This is due to the fact that the sphere decoding method is more efficient in the pruning process.



(a) Zero vectors H1 (b) Old solution H2

(c) Unconstrained solution H3 (d) All heuristics H1+H2+H3

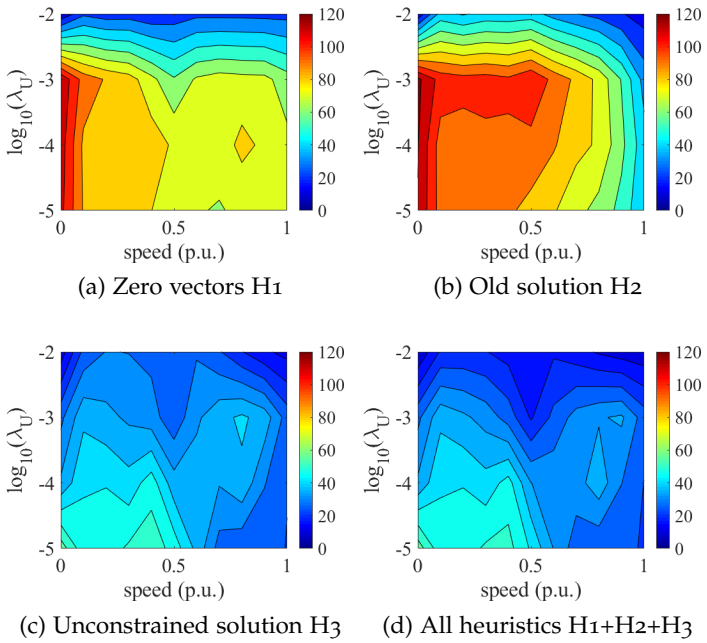
Figure 7.14: Number of candidate solutions evaluated at nominal current: sphere decoding solver, MB controller.

The switching effort weight λ_U is of paramount importance in the problem solution. When the weight value is too low, many candidates need to be explored, as inferable from Fig. 7.1. If the switching effort is not relevant, the equivalent voltages can be synthesised by many different combinations of the eight base voltages. The current reference tracking error, in fact, would dominate in the cost value calculation. It is worth mentioning that, FS or DSVM schemes are usually attractive for applications where the switching effort is relevant for the converter, i.e. high λ_U values are expected. Thus, both the solvers, in particular the sphere decoding, appear to be effective for these applications.

Effectiveness of the heuristics

The first heuristic to be considered was the application of the null voltage vectors, i.e. $\mathbf{u}_7 = [1, 1, 1]$ and $\mathbf{u}_0 = [0, 0, 0]$. This heuristic method resulted to be the worst one in terms of average evaluated candidates, see Fig. 7.16a and Fig. 7.15a. When the speed increases, the number of candidates increases since a higher equivalent voltage vector has to be synthesised.

The solution of the MPC problem obtained for the previous control period is, at steady state, a quite effective heuristic. The heuristic H2 returns a number of candidate evaluation that is not as good as expected if compared to the heuristic H1. The heuristic H1 is effective at low speed with respect to heuristic H2, see for instance Fig. 7.15a and



(a) Zero vectors H1 (b) Old solution H2

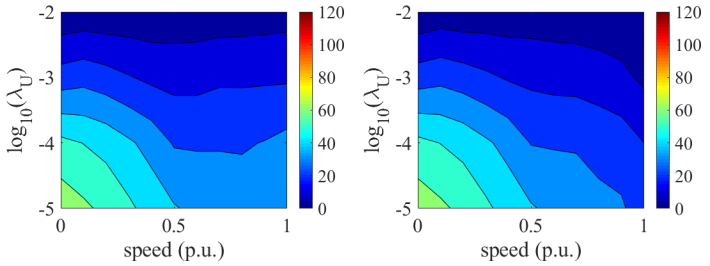
(c) Unconstrained solution H3 (d) All heuristics H1+H2+H3

Figure 7.15: Number of candidate solutions evaluated at nominal current operation: branch and bound solver, RLS controller.

Fig. 7.15b. The reason of the poor performances of heuristic H2 was found in the non-use of the receding horizon policy. On the contrary, heuristic H2 would be often the best in case the receding horizon is applied. This is due to the fact that there is a partial overlapping between the prediction horizons of two consecutive control periods. In other words, the part of the old solution related to the overlapping time window represents a good guess for the next MPC problem, provided that the prediction model is accurate. Anyway, a general drawback of this heuristic was found in dynamic conditions.

The rounding of the unconstrained solution to the closest feasible switch position is another interesting alternative. The heuristic H3 is quite convenient to implement for the sphere decoding solver, since the unconstrained solution is strictly required by the algorithm. Rounding the switching positions to zero or one is quite accurate approximation of the solution, as proved by Fig. 7.16c and Fig. 7.15c. This heuristic method appears to be the most effective one, reminding that the receding horizon policy is not used.

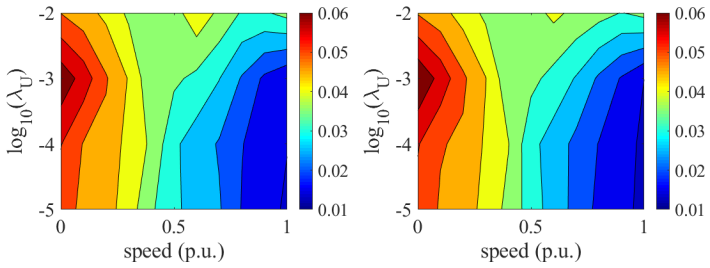
For sake of completeness, the effect of all the heuristic methods to calculate the initial value of the cost for the optimum solution are reported in Fig. 7.16d and Fig. 7.15d. The obtained results benefit, as expected, of the effects of all the described methods.



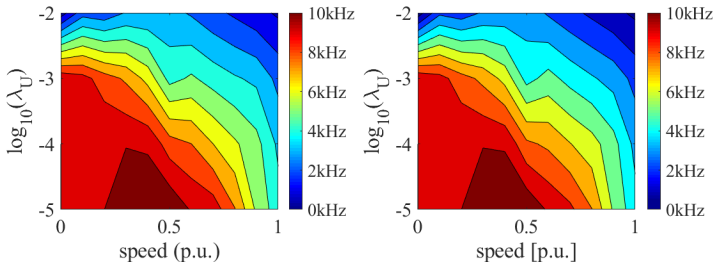
(a) Zero vectors H1 (b) Old solution H2

(c) Unconstrained solution H3 (d) All heuristics H1+H2+H3

Figure 7.16: Number of candidate solutions evaluated at nominal current operation: sphere decoding solver, RLS controller.



(a) Current TDD: model-based (b) Current TDD: parameter-free



(c) $\langle f_{sw} \rangle$: model-based (d) $\langle f_{sw} \rangle$: parameter-free

Figure 7.17: Comparison of the model-based and the parameter-free FS predictive controllers: ideal motor model.

Sphere decoding performances for the model-based and the motor parameter-free controllers

The same simulation tests of Sect. 7.4.4 were performed by means of a conventional model-based predictive controller, i.e. (2.23). The com-

parison of Fig. 7.16 and Fig. 7.14 confirms that the RLS adaptive model is as accurate as the model-based one. This fact is further confirmed by Fig. 7.17, showing the controllers performances in terms of average switching frequency ($\langle f_{sw} \rangle$) and current Total Demand Distortion (TDD). These are the two performance indices of greatest interest from the industrial applications point of view [44]. Both the frequency and the current distortion are almost the same for different operating speeds and tuning. In simulations, the parameter-free controller could reach, at maximum, the same performances of the model-based one. On the other hand, for the experiments, the parameter-free algorithm is expected to be even more accurate. The RLS estimators capture also inverter non-linarities and motor time varying parameters, that are neglected in the state space model adopted for the model-based prediction (2.23).

Part III

DATA-DRIVEN CONTROL

Data-driven control techniques have become increasingly popular in recent years in the control community, due to the availability of massive amounts of data and several advances in data science. These control design methods bypass the system identification step and directly exploit collected data to construct the controller. Two of the most promising data-driven algorithms are presented, namely the Subspace Predictive Control algorithm and the Data-Enabled Predictive Control algorithm. These methods are real-time implemented for the current control of many synchronous motor drives. The same mathematical framework is exploited also for the design of data-driven estimators. The estimation of a load torque disturbance is presented, as an applicative example.

8

DATA-DRIVEN CURRENT CONTROL

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 [37]. 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 MPC, which has been applied to power electronics control tasks for two decades, reaching an industrial and commercial level [77].

CS MPC methods for PMSM current control, which is the focus of this chapter, rely on a state-space model of the motor to build the predictive controller [9, 25, 34]. 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 [81] or online [6] procedures. Self-commissioning and auto tuning techniques are also consolidated strategies. In [59], 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 [21, 52, 80, 86, 92], although most of these strategies are implemented for FS-MPCs.

The key idea behind data-driven predictive controllers is to avoid the model identification stage entirely, and design the controller directly from collected 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 [87]. However, there are just a few examples of data-driven control applications for electric motor drives. In [10], 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 [2, 73, 93] or parameter-free [53] algorithms. In particular, [93] and [73] 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.

A transition from model-based to data-driven control design is described, 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. Optimization-based control schemes are considered, i.e. MPC-type solutions. First a state-of-the-art CS-MPC si presented, whose model is obtained through a previous motor characterization. Then, a further step towards data-driven control designs is performed, 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 [42]. A further step is represented by the SPC [26], where the collected data are processed offline by means of a least-square program, and the resulting Auto Regressive with eXogenous inputs (ARX) predictor is de-noised by singular value thresolding. 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 [18, 19], 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 [11, 39, 40].

The novel contributions of this work are manifold:

- the perspective of data-driven control design is illustrated, using a predictive control framework;
- 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;
- the computational aspects of the presented control strategies are compared;
- as a technical contribution, the problem of the offset-free tracking for the SPC and DeePC methods is addressed;
- guidelines are provided for the choice of the control parameters and excitation input signals for this application.

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.

8.1 MODEL-BASED CONTINUOUS-SET MPC OF PMSM CURRENTS

According to the CS-MPC paradigm (Section 3.1), the future control input sequence $\mathbf{u}_{dq} = [\mathbf{u}_{dq}(k), \mathbf{u}_{dq}(k+1), \dots \mathbf{u}_{dq}(k+N-1)]^T$ is optimized in order to steer the predicted future currents $\mathbf{i}_{dq} = [\mathbf{i}_{dq}(k+1), \mathbf{i}_{dq}(k+2), \dots \mathbf{i}_{dq}(k+N)]^T$ to a desired reference $\mathbf{i}^*_{dq} = [\mathbf{i}^*_{dq}(k+1), \mathbf{i}^*_{dq}(k+2), \dots \mathbf{i}^*_{dq}(k+N)]^T$. Only the first optimal input of the sequence $\mathbf{u}_{dq}(k)$ is applied to the plant (receding horizon principle). Thus, the following optimization problem is solved at each control period:

$$\begin{aligned} \min_{\mathbf{u}_{dq}, \mathbf{i}_{dq}} \quad & \left(\|\mathbf{i}_{dq} - \mathbf{i}^*_{dq}\|^2_{\mathbf{Q}} + \|\mathbf{u}_{dq}\|^2_{\mathbf{R}} \right) \\ \text{subject to} \quad & \mathbf{i}_{dq}(k+1) = \mathbf{A}\mathbf{i}_{dq}(k) + \mathbf{B}\mathbf{u}_{dq}(k), \\ & \mathbf{u}_{dq}(k) \in \mathcal{U}, \; k = 0, \dots, N-1 \end{aligned} \tag{8.1}$$

where, accordingly to the notation presented in Section 3, N is the prediction horizon, $\mathbf{Q} \succcurlyeq 0$ and $\mathbf{R} > 0$ are two weighting matrices, $\mathbf{A}$ and $\mathbf{B}$ represent the state space model used to predict the future currents 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 QP which requires an online solver like qpOASES, as in [75], but it is still easily solvable in real-time. The weighting matrices tuning could be challenging. In case of LTI plants, the tuning can be performed offline granting asymptotic stability. In real life, a semi-manual tuning is often preferred based on experiments.

In the context of PMSM currents control, future currents are estimated by exploiting the parametric model, based on the PMSM voltage balance equations, represented in the dq reference frame, presented in (2.27).

Table 12: 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

In the considered application, the selected model neglects the cross-saturation phenomena, as well as iron-saturation (2.20) and back-EMF harmonics effects. Thus, the model can result as oversimplified for some PMSM topologies, such as pure reluctance motors. However, many CS-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 [17]. It is worth noticing that the data-driven paradigm overcomes the issues coming from assumptions on the model structure.

An integral action is included in the MPC formulation by means of the velocity form of the MPC problem (8.1) [63], in order to achieve an unbiased current reference tracking. The discussion about the offset-free data-driven control is given in Sec. 8.2.3.

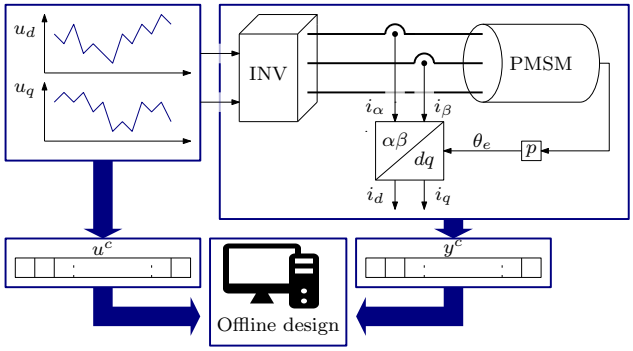
8.2 TOWARDS DATA-DRIVEN CONTROL OF PMSM DRIVES

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. In contrast to the parametric model (2.27) 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:

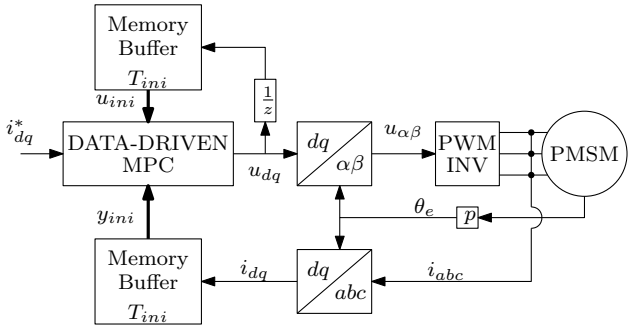
- A *data collection step*, followed by offline rearrangement of the voltages/currents samples into proper matrices;
- An *online program*, when the tracking problem is solved, with the voltages/currents samples matrices acting as a constraint. In this online step, the controller has access to the latest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

8.2.1 *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 8.1a). The sequence $u^c = [u_1^c; u_2^c; ...; u_T^c] \in \mathbb{R}^{2T}$ contains the



(a) Scheme of the data collection process.



(b) Block scheme of the online controller.

Figure 8.1: Overview of the data collection step and the online program.

inverter reference voltages and it fulfills the *persistency of excitation* requirement [85, Corollary 2], that is the Hankel matrix of inputs in (8.3) needs to have full raw rank. The selection of the input signal is further discussed in Sec. 8.4.1. The resulting output sequence contains the dq currents $y^c = [y_1^c; y_2^c; \dots; y_T^c] \in \mathbb{R}^{2T}$.

PEM-MPC

In the PEM-MPC method, the coefficients of the state-space matrices $\mathbf{A}$ and $\mathbf{B}$ used in (2.28) are inferred from data by means of an ordinary least-square problem[1] that involves the sequence u_c and y_c:

$$\min_{A,B} \sum_{k=1}^{T-1} \|x_c(k+1) - Ax_c(k) - Bu_c(k)\|^2 \tag{8.2}$$

The main difference between the resulting model and the parametric voltage balance equation (2.27) 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 (2.10).

1 It is referred to [39] for a discussion on how to solve this problem numerically.

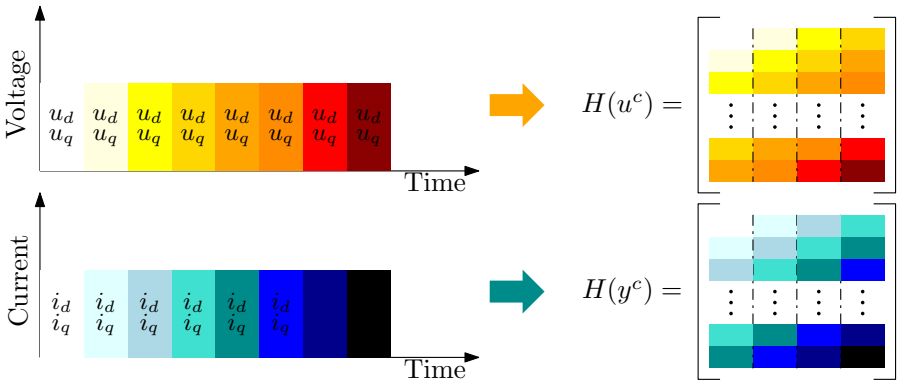


Figure 8.2: Hankel matrix construction.

SPC

In the SPC algorithm, the state-space model (2.23) is replaced by a 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 (Figure 8.2):

$$\mathscr{H}(u_c) := \begin{bmatrix} u_1 & u_2 & \cdots & u_{T-T_{ini}-N+1} \\ u_2 & u_3 & \cdots & u_{T-T_{ini}-N+2} \\ \vdots & \vdots & & \vdots \\ u_{T_{ini}+N} & u_{T_{ini}+N+1} & \cdots & u_T \end{bmatrix}. \tag{8.3}$$

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:

$$\begin{bmatrix} U_P \\ U_F \end{bmatrix} := \mathscr{H}(u^c), \qquad \begin{bmatrix} Y_P \\ Y_F \end{bmatrix} := \mathscr{H}(y^c), \tag{8.4}$$

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 12 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 ARX model as predictor [38]:

$$y = P_w \begin{pmatrix} u_{ini} \\ y_{ini} \end{pmatrix} + P_u u, \tag{8.5}$$

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 solving the least-square problem

$$\min_{P_w, P_u} \left\| Y_F - \left[\begin{array}{c|c} P_w & P_u \end{array}\right] \left[\begin{array}{c} U_P \\ Y_P \\ \hline U_F \end{array}\right] \right\|^2, \tag{8.6}$$

where P_w multiplies the two blocks $\begin{bmatrix} U_P \\ Y_P \end{bmatrix}$ and P_u multiplies the block U_F. 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 [26]. In a nutshell, the matrix is decomposed as:

$$P_w = U\Sigma V^T = U \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & \epsilon_1 & & \\ & & & & \ddots & \\ & & & & & \epsilon_n \end{bmatrix} V^T \tag{8.7}$$

where U and V are two orthogonal matrices, whereas Σ is a diagonal matrix. Moreover, $\sigma_1, \ldots, \sigma_k$ are the dominant singular values, while $\epsilon_1, \ldots, \epsilon_n$ are the negligible ones. The denoising is performed by reconstructing the matrix P_w using matrices U and V, and a simplified version of Σ which contains only the dominant singular values.

DeePC

The design of a DeePC controller is purely data-driven, as the data block Hankel matrices defined in (8.4) are used in their raw form in the controller. This method is based on the so called *Fundamental Lemma* of *behavioral system theory* [85], 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

$$\begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}. \tag{8.8}$$

Implicitly, (8.8) 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 (u, y) are considered as free optimization variables,

the vector g that satisfies the first two block-equations of (8.8) can be expressed explicitly as

$$g = \begin{bmatrix} U_P \\ Y_P \end{bmatrix}^{\dagger} \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \Phi z = M \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \Phi z, \tag{8.9}$$

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.

8.2.2 *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 (2.23) 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.

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 (8.1). If the problem is unconstrained ((8.1) is absent), the PEM-MPC yields a linear feedback controller [25] of the form $u = K^r r + K^x x(k)$. On the other hand, the QP problem requires an iterative solver as in [75], 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 dimension 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.

SPC

The SPC algorithm solves the same tracking problem (8.1) as in MPC or PEM-MPC, but with the state-space model (2.23) replaced by the predictor (8.5):

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

$$\text{subject to} \quad y = P_w \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + P_u u \tag{8.10b}$$

$$u(k) \in \mathcal{U}, \; k = 0, \dots, N-1 \tag{8.10c}$$

Similarly to the PEM-MPC, if the constraints (8.10c) are not present, then the problem can be solved in closed-form by substituting the predictor equation (8.10b) into (8.10a) 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 (8.10c) 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.

DeePC

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

$$\min_{g,u,y} \quad \|y - r\|_Q^2 + \|u\|_R^2 + \lambda_g \|g\|^2 \tag{8.11a}$$

$$\text{s.t.} \begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}, \; u(k) \in \mathcal{U}, \; k = 0, ..., N-1 \tag{8.11b}$$

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 (8.11a) could be different from the predicted one. Thus, the term $\lambda_g \|g\|^2$ helps to robustify the control problem [23, Section III.C]. The absolute-value norm is an interesting alternative which is studied in literature, in order to promotes the sparsity of g. In other words, the optimal sequence should be generated by combining a few trajectories of the data Hankel matrices. However, this solution is too computationally demanding for real-time applications.

In the unconstrained case, the problem can be solved directly using the null-space representation presented in (8.9). The future currents

and voltages sequences u and y are replaced in (8.11a) with $U_F g$ and $Y_F g$, respectively, obtaining

$$\min_z \|Y_F(M\begin{bmatrix}u_{ini}\\ y_{ini}\end{bmatrix} + \Phi z) - r\|_Q^2 + \|U_F(M\begin{bmatrix}u_{ini}\\ y_{ini}\end{bmatrix} + \Phi z)\|_R^2 \quad (8.12)$$

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:

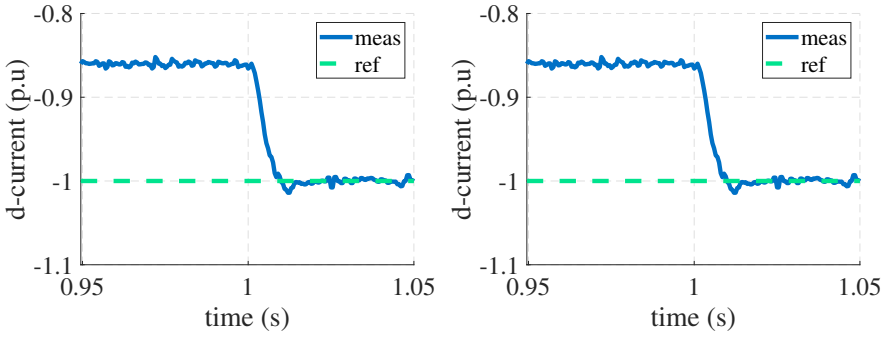
$$\begin{aligned} H &:= \Phi^T Y_F^T Q Y_F \Phi + \Phi^T U_F^T R U_F \Phi \\ d &:= \left(r - Y_F M \begin{bmatrix}u_{ini}\\ y_{ini}\end{bmatrix}\right)^T Q\Phi - \left(U_F M \begin{bmatrix}u_{ini}\\ y_{ini}\end{bmatrix}\right)^T R\Phi. \end{aligned} \quad (8.13)$$

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 [39]. Starting from the optimal value of z^{opt}, (8.9) 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 (8.12) 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 (8.8). 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 [23]: 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 LTI system model, whereas the DeePC does not. 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 non-linear system, e.g. the grid connected inverter application shown in [39]. Finally, SPC solves the tracking problem in the input u, whereas the DeePC in g.

8.2.3 *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 this principle, this framework is introduced also for data-driven controllers. For the MPC and PEM-MPC algorithms, the integral can be included by formulating the optimization problem in its velocity-form [63]. The idea is to perform the data collection stage



(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 $t = 1s$.

filling the matrices with incremental data, e.g. $\Delta y = y(k) - y(k-1)$. For instance, the DeePC problem in (8.11) is written as follows:

$$\begin{aligned} \min_{g,\Delta u,\Delta y} \quad & \|\Delta y - r'\|_Q^2 + \|\Delta u\|_R^2 + \lambda_g \|g\|^2 \\ \text{subject to} \quad & \begin{bmatrix} U'_P \\ Y'_P \\ U'_F \\ Y'_F \end{bmatrix} g = \begin{bmatrix} \Delta u_{ini} \\ \Delta y_{ini} \\ \Delta u \\ \Delta y \end{bmatrix} \\ & u(k) = u(k-1) + \Delta u(k) \in \mathcal{U}, \ k = 0, 1..., N-1 \\ & r(k)' = r(k) - y(k), \ k = 1, ..., N \end{aligned} \tag{8.14}$$

$[U'_P, Y'_P, U'_F, Y'_F]^T$ are the Hankel matrices filled with incremental data. The optimization problem (8.14) 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.

8.3 SIMULATION RESULTS

Three different PMSM topologies are considered in the simulations in order to prove the flexibility of the data-driven approach: a SPM motor, an IPM motor and a SyR motor. The nominal parameters of the three machines are reported in Table 13. Some non-linear phenomena, such as iron saturation, are neglected.

Table 13: Overview of the main nominal parameters of the three synchronous motors used for the simulations

Motor Data	Symbol	SPM	IPM	SyR
Pole pairs	p	2	2	
Phase resistance	R	1.5 Ω	2.7 Ω	4.7 Ω
d-axis inductance	L_d	0.003 H	0.02 H	0.4 H
q-axis inductance	L_q	0.003 H	0.110 H	0.08 H
PM flux-linkage	Λ_{pm}	0.25 V s	0.22 V s	0 V s
Nominal current	I_N	10 A	4.2 A	5.6 A
Nominal d current	$I_{N,d}$	0 A	2.8 A	4 A
Nominal q current	$I_{N,q}$	10 A	3.1 A	4 A
Nominal speed	Ω_N	3000 rpm	2000 rpm	500 rpm

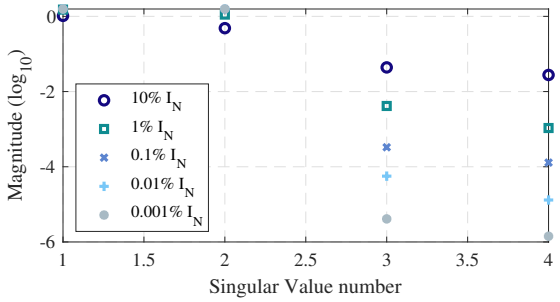


Figure 8.4: Effect of the current measurement quantization in the singular values magnitudes (SPM motor case).

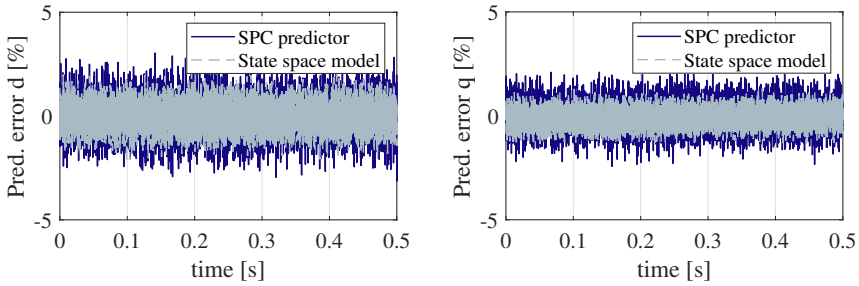
8.3.1 *Data acquisition step*

In the data acquisition step a training sequence of random dq voltage vectors is applied to the motor (see Sec. 8.2.1). The SPM machine is considered for the following analysis.

The SPM motor is maintained at standstill by a second machine while a zero-mean sequence of voltages is applied. The amplitude of the training voltage vectors is limited to 10% the nominal voltage, i.e. 17V. In order to obtain a data-driven controller with integral action, we adopt an incremental approach, as the one presented in [25]. In a nutshell, we collect the current increments induced by the application of a voltage increment within one sample period $T_s = 100\mu s$.

8.3.1.1 *Effect of the current sensing in the design procedure*

The differential current measurement could compromise the controller design because of the resolution of the current sensing. It is worth mentioning that very noisy measurement badly affect the Hankel ma-



(a) d axis error (b) q axis error

Figure 8.5: Experimental test: comparison between state space model and data-driven predictor in terms of prediction error.

trices [18]. For this purpose, the authors analyze the effect of the noise induced by different resolutions of the current sensing on the singular values of P_w (Sec. 8.2.2). Results are reported in Figure 8.4 for a SPC controller with $T_{ini} = 1$ and $N = 3$.

The electric motor model (2.10) has a dimension equal to 2. Thus, it is expected to find 2 dominant singular value among the 4 that can be obtained with the selected T_{ini} and N. This is confirmed by Figure 8.4, where at least two order of magnitude separate the first two singular values from the others, excluding the case of an unrealistic quantization of $10\%I_N$.

8.3.1.2 *Accuracy of the data-driven predictor*

The training test has been performed also on a test bed, considering the SPM motor. The SPC matrices are derived. In order to check the accuracy of the data-driven predictor stated in the equality constraint of (8.10), the latter is compared with the voltage balance model (2.10). The state space is indeed more accurate, as proved by Figure 8.5. However, the data-driven predictor grants a similar accuracy without requiring any parameter identification procedure on the motor. The data-driven design procedure can be adopted as a tool to verify the accuracy of the nominal motor model. Manufacturing tolerances, in fact, could cause small discrepancies between nominal parameters and the actual ones.

8.4 EXPERIMENTAL VALIDATION

The authors propose the experimental validation on an interior permanent magnet motor. The nominal parameters of the considered machine are reported in Table 14, while the test-bench layout is similar to the ones reported in previous sections. All the algorithms, i.e. the MPC, PEM-MPC, SPC and DeePC, are real-time implemented on the dSPACE MicroLabBox at a sample rate of $T_s = 100\mu s$.

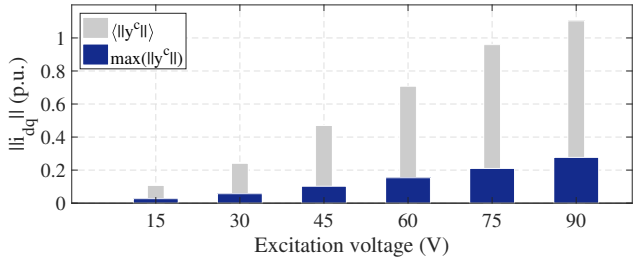
Table 14: Overview of the motor parameters.

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	$1000\,rpm$

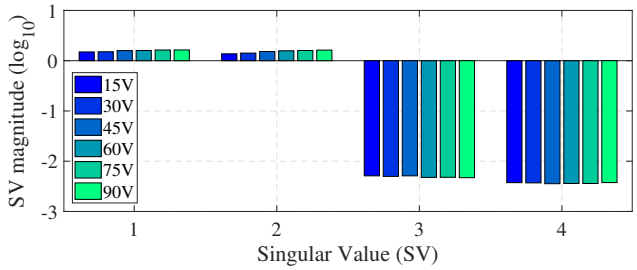
8.4.1 *Data Acquisition Step*

The test designed to collect I/O data from the IPM motor consists of excitation with a random (detailed below) dq voltage vectors sequence u^c and the measurement of the dq currents via LEM sensors. 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 remains 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 8.6a 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, other problems need to be considered, i.e. if the information carried by the data is rich enough to describe the current dynamics. The PWM synthesis of low voltages could emphasize some inverter non-linearities, e.g. not properly com-



(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 8.6: Overview of some key parameters of the data collection test.

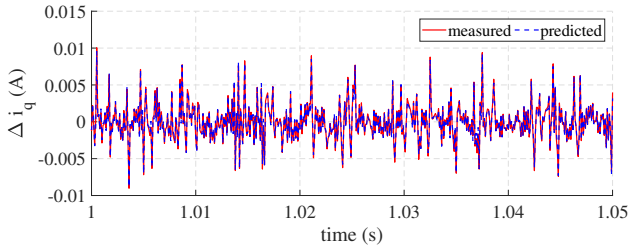
pensated dead-times, that are not of interest of our identification. In order to evaluate if the data are collected properly, the dominant singular values of the matrix P_w are analyzed (see the logarithmic plot in Figure 8.6b). The number of dominant values should be coherent with the anticipated dimension of the state, see Table 12. Two dominant values characterize the considered dynamic, as expected.

8.4.2 *Parameters Selection*

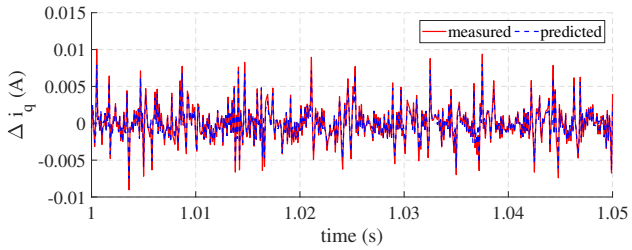
In this section the problem of parameters selection is addressed for designing the data-driven controls. The prediction horizon length N is chosen according to the MPC framework, i.e. $N = 3$. This value is a good trade-off between accuracy and computational effort for this application [17]. Moreover, 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 (8.11) 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 T_{ini} provides a complexity for the model. In [85], the system lag[2] l is used to find a lower bound for T_{ini}. In particular, if $T_{ini} \geqslant l$ the system

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.

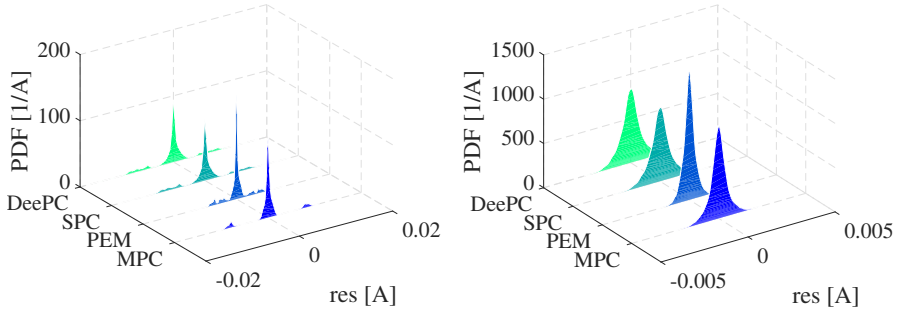


(a) PEM-MPC



(b) DeePC

Figure 8.7: Accuracy of the data-driven predictors in the estimation of the q-axis current variation.



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

Figure 8.8: Residual analysis of the prediction error: probability distribution function of the residuals for the presented predictors.

prediction is uniquely determined. Thank to this criterion, 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 [85] gives a lower bound for T, whose value for the considered application is $T \geqslant 3(T_{ini} + N + 2) - 1$. $T = 100$ samples are selected, which satisfies the inequality. All dimensions of the matrices can be computed using Table 12.

8.4.3 *Accuracy of the Data-Driven Predictor*

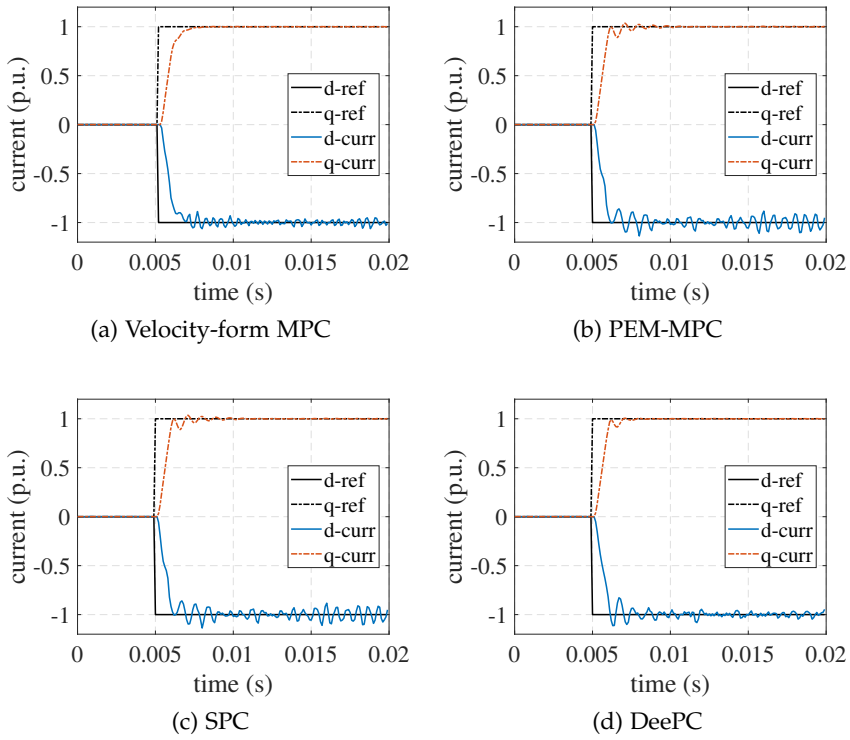
The accuracy of the data-driven predictors is investigated in this subsection, taking the model-based MPC as benchmark. This analysis is performed during steady state operation, when the motor is working at the nominal MTPA current point (see Table 14) at standstill. During the tests the currents are regulated by standard PI controllers. The open-loop prediction accuracy of the methods is analyzed. This means that the predictors are fed by current measurements and the reference voltages computed by the PIs. A first qualitative information on the accuracy is provided by Figure 8.7a and Figure 8.7b. The figures show the comparison between the measured q-axis current increments and the predicted ones, using respectively the predictor obtained with (8.2) and (8.12). A good correspondence between measurements and predictions is observed for both the controllers.

The residuals between measured and estimated currents are considered as performance index, as suggested in [10]. The results of this analysis are reported in Figure 8.8. The figures show the estimated probability density function of the d and q residuals for all the described predictors. From literature [10], we expect a zero mean normal distribution of the residuals, 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 8.7a, since the predicted currents almost overlap the measured ones.

8.4.4 *Online Unconstrained Controller*

In this subsection the authors provide a comparison between model-based and data-driven designed controllers in terms of step current reference response. In particular, the reference r is changed from zero to the nominal maximum-torque-per-ampere current in a step-wise manner. The model-based MPC adopts the motor parameters which were previously obtained by means of characterization procedures (see Table 14). All the data-driven controllers are designed from the same data recording, in particular the one defined by a $u_{exc} = 50\,V$.

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 [59]. Among data driven controllers, the DeePC is considered the most data-oriented algorithm, because it uses raw data without any pre-processing. De-

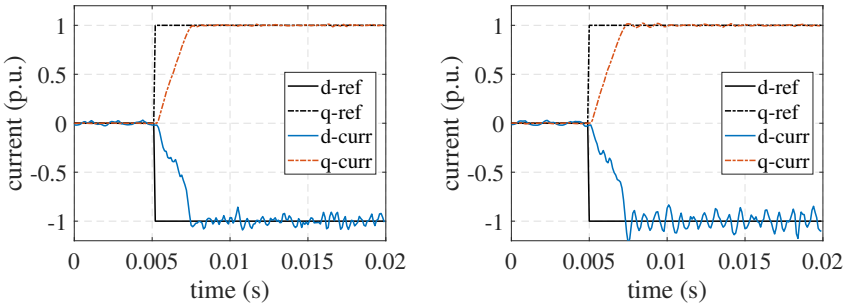


(a) Velocity-form MPC (b) PEM-MPC

(c) SPC (d) DeePC

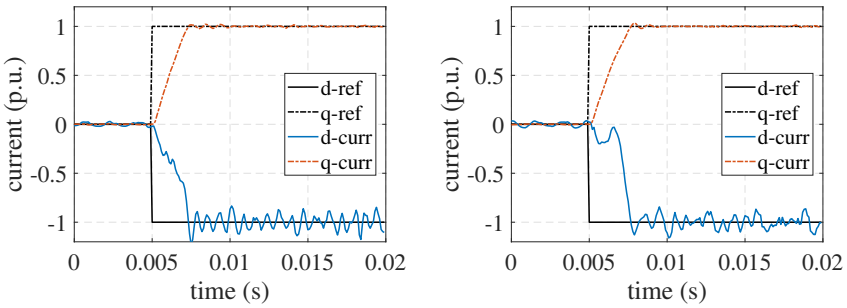
Figure 8.9: Comparison of the step responses of model-based and data-driven current controllers at standstill.

spite the direct exploitation of raw data, it has almost the same performance as the others. We reported the same step response analysis also at nominal speed rate in Figure 8.10. 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 (2.10) 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, we underline 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 [18]). In addition, accordingly to [43], a feedforward term can be nested in the controller to improve disturbance rejection performances, without penalizing the overshoot in the dynamics. We therefore believe that there is much unexplored potential to improve the performance to data-driven controllers.



(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.

9 DATA-DRIVEN MOVING HORIZON ESTIMATION

In this chapter it is reported an original contribution of this thesis regarding the input disturbance estimation problem, which is summarized in Figure 9.1. In the scheme it is represented a standard output tracking control problem for a LTI system P. In a nutshell, it is desired that the process outputs $y \in \mathbb{R}^p$ track their reference $r \in \mathbb{R}^p$ through the controller C that impose the input $u \in \mathbb{R}^m$. However, the control input applied to the plant is affected by a disturbance $w \in \mathbb{R}^m$. Thus, the actual input that determines the dynamic of the system is $u' = u + w$. In a first moment, it is assumed that the output noise n is negligible, then the assumption is relaxed.

The first goal to achieve consists into estimating the disturbance w without performing an identification of the process model. It is straightforward that the main challenge in the identification is the absence of an available model of the process However, according to the behavioral system theory Section 4, it is assumed to have a N_s-long sequence of applied inputs $u^d \in \mathbb{R}^{m N_s}$ and the related sequence of outputs $y^d \in \mathbb{R}^{p N_s}$, i. e. same data samples of the process. In order to solve the estimation problem it is required the *persistency of excitation* property for the input sequence u^d, as defined in Section 4.1.

In the following analysis, it is assumed that the input disturbance w is always null when the input-output sequence (u^d, y^d) is collected from the plant. Thanks to this assumption, the sequence of actual process inputs $u' = u + w$ is persistently excited, too. In fact, it is remarkable that the persistency of excitation of u' could be lost for certain sequence of disturbances, e.g. $w^d \equiv -u^d \rightarrow u'^d \equiv 0$.

9.1 DATA DRIVEN INPUT DISTURBANCE ESTIMATION

Let's consider at first a controllable process P, which has input u' and output y. A sequence of input-output trajectories (u^d, y^d) has been collected from the plant. It is reminded that $u^d \equiv u'^d$. Since u'^d is persistently excited, the *Fundamental Lemma* by Willems is applied. The lemma states that any trajectory of a LTI system can be obtained by computing a linear combination of the input-output Hankel matrix columns. Thus, these matrices can be exploited to predict future behavior of the system and design optimal estimators.

Let's focus on the discrete nature of the input disturbance estimation, as shown in Figure 9.2. Moreover, it is assumed to have an estimation horizon length equal to T_{ini} steps, which coincides with the rows number of $\mathscr{H}_{T_{ini}}(u^d) \equiv \mathscr{H}_{T_{ini}}(u'^d)$.

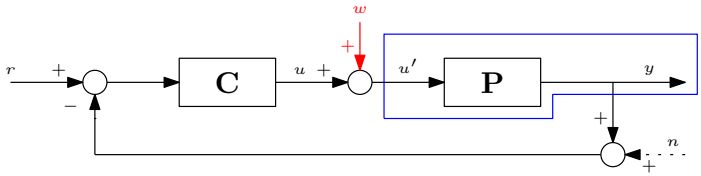


Figure 9.1: Input disturbance estimation problem.

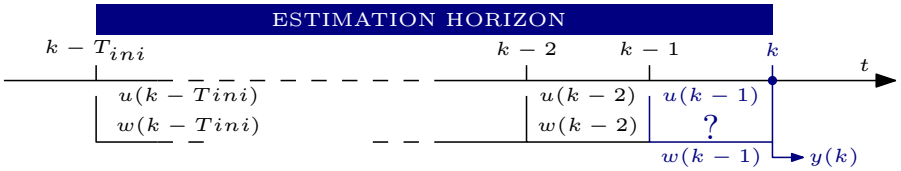


Figure 9.2: Overview of the estimation problem in a discrete time domain.

Considering the k-th step, the last measured output is $y(k)$, induced by the input $u'(k-1) = u(k-1) + w(k-1)$. The simplest input disturbance estimation problem that it is possible to build up consists into estimating $w(k-1)$ given:

- a past sequence of inputs applied to the plant $u_{ini} = [u(k - T_{ini}), u(k - T_{ini} + 1), \dots, u(k-2)]^T \in \mathbb{R}^{m(T_{ini}-1)}$;
- a past sequence od measured output $y_{ini} = [y(k - T_{ini} + 1), y(k - T_{ini} + 1), \dots, y(k-1)]^T \in \mathbb{R}^{p(T_{ini}-1)}$;
- the last applied control input $u(k-1) \in \mathbb{R}^m$;
- the last measured output $y(k) \in \mathbb{R}^p$;

The idea behind a data-driven disturbance estimation consists into computing which fictitious input $\hat{u}(k-1)$ let the output evolve from the initial trajectory (u_{ini}, y_{ini}) to the output $y(k)$. The difference between the actual input $u(k-1)$ and fictitious input $\hat{u}(k-1)$ gives the disturbance $w(k-1)$ affecting the process.

In order to treat the estimation problem in a data-driven framework, the input-output Hankel matrices $\mathscr{H}_{T_{ini}}(u^d)$ and $\mathscr{H}_{T_{ini}}(y^d)$ are decomposed in two blocks:

$$\begin{pmatrix} U_P \\ U_F \end{pmatrix} := \mathscr{H}_{T_{ini}}(u^d) = \mathscr{H}_{T_{ini}}(u'^d), \qquad \begin{pmatrix} Y_P \\ Y_F \end{pmatrix} := \mathscr{H}_{T_{ini}}(y^d), \quad (9.1)$$

where the matrices dimensions are respectively $U_P \in \mathbb{R}^{m(T_{ini}-1)x(N_s-L)}$, $Y_P \in \mathbb{R}^{p(T_{ini}-1)x(N_s-L)}$, $U_F \in \mathbb{R}^{m\,x(N_s-L)}$ and $Y_F \in \mathbb{R}^{p\,x(N_s-L)}$.

Thanks to the fundamental lemma, any trajectory (u', y) of the system can be written as:

$$\begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u'_{ini} \\ y_{ini} \\ u'(k-1) \\ y(k) \end{bmatrix} = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u(k-1) \\ y(k) \end{bmatrix} + \begin{bmatrix} w_{ini} \\ 0^{p(T_{ini}-1)} \\ w(k-1) \\ 0 \end{bmatrix}, \; g \in \mathbb{R}^{N_s - T_{ini}} \tag{9.2}$$

where $w_{ini} = [w(k-T_{ini}), w(k-T_{ini}+1), \dots, w(k-2)]^T \in \mathbb{R}^{m(T_{ini}-1)}$. The disturbance plays in the system in the same way as the input.

Given u_{ini}, y_{ini} and $u(k-1)$, there is not a unique solution to the problem (9.2), in general. However, uniqueness is reached if it holds that $T_{ini} - 1 \geqslant \ell(\mathfrak{B})$, where $\ell(\mathfrak{B})$ is the lag of the system imposes uniqueness. The hypothesis needed to guarantee the uniqueness are formalized by the *Initial condition* lemma. The idea behind this lemma is the following. If a sufficiently long past trajectory of a system is observed, then it is possible to identify the unique initial state from which the system started to evolve.

9.1.1 *Data driven simulation algorithm*

Let's assume that the process P is running with an initial null disturbance. Considering (9.2), this means that $w_{ini} \equiv 0$. At the k-th step an input disturbance $w(k) \neq 0$ occurs on the plant.

If a past estimation window of length T_{ini} is observed, such that $T_{ini} - 1 \geqslant \ell(\mathfrak{B})$, it holds that:

$$\exists! g \in \mathbb{R}^{N_s - T_{ini}} \text{ s.t. } \begin{bmatrix} U_P \\ Y_P \\ Y_F \end{bmatrix} g = \begin{bmatrix} u'_{ini} \\ y_{ini} \\ y(k) \end{bmatrix} = \begin{bmatrix} u_{ini} \\ y_{ini} \\ y(k) \end{bmatrix} + \begin{bmatrix} w_{ini} \\ 0^{p(T_{ini}-1)} \\ 0 \end{bmatrix}. \tag{9.3}$$

Once solved the problem (9.3) in g, an estimated input for the (k-1)st step $\hat{u}(k-1) = U_F g$ is computed. Since the true control input $u(k-1)$ is known, the disturbance $w(k-1)$ acting on the system is easily derived as:

$$w(k-1) = \hat{u}(k-1) - u(k-1) = U_F g. \tag{9.4}$$

The estimation is repeated every time step, meaning that the vector of initial disturbances w_{ini} is updated and the problem is solved with the upcoming $y(k)$ and $u(k-1)$.

This first approach to estimate the input disturbance appears quite rude. Each time step, in fact, the disturbance is recomputed without any regularization. Thus, it is expected quite a noisy estimation in presence of output noise.

Algorithm 1 Data driven simulation for input disturbance estimation

Input: $[u^d, y^d]^T$ input-ouput data pre-collected from the plant, the last T_{ini} input-output measurements, split in $[u_{ini}, y_{ini}, y(k)]^T$ and $u(k-1)$.

Solve (9.3) for g.
Estimate the (k-1)st input $\hat{u}(k-1) = U_F g$.
Compute the input disturbance $w(k-1)$ as in (9.4).
Update the initial disturbances vector $w_{ini} = [w_{ini}(m : m(T_{ini-1})); w(k-1)]$

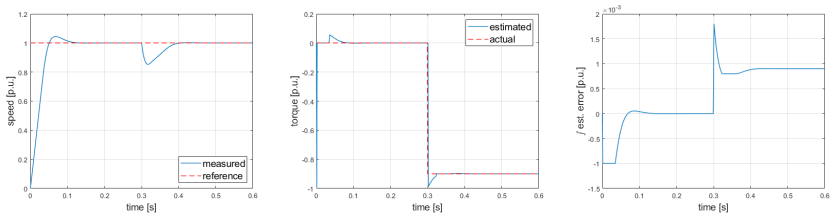


Figure 9.3: Estimation of a constant load torque at nominal speed operation.

9.1.1.1 *A first application: Load torque estimation for the speed control loop of a motor drive*

A simple example is presented to ease the comprehension of the estimator and to show its performances. Let's consider the torque balance equation on a motor shaft, described by the following state-space model, based on (2.32) and (2.33):

$$x(k+1) = (1 - T_s \frac{B}{J})x(k) + \frac{T_s}{J}u(k) + \frac{T_s}{J}w(k), \quad y(k) = x(k); \quad (9.5)$$

The output $y(k)$ of the system is the angular speed and it coincides with the state. The control input $u(k)$ is the motor torque, but a disturbance torque $w(k)$ affects the process. The main model parameters are: the rotating inertia $J = 53.56 Kg/cm^2$, the viscous friction coefficient $B = 10^{-5} Kg\, m^2/s$, the nominal power $P_N = 3kW$ and the nominal speed $\Omega_N = 1500rpm$. The complete list of model parameters can be found at https://docs.rs-online.com/520d/0900766b814d71de.pdf. A sampling time T_s equal to 1ms is selected.

A PI controller is implemented to accomplish the speed control task. In details, the proportional gain is 0.5 and the integral gain is 20. Both integrator output and PI output are saturated if they overcome the nominal torque.

A first test is performed to collect input-output matrices to design the observer. In details, $N_s = 100$ input-output trajectories are collected from the motor, to build an observer with an estimation window equal to $T_{ini} = 10$ time intervals.

The estimation of a constant load torque is pretty accurate, as proved by Figure 9.3. On the left side, the speed reference tracking is re-

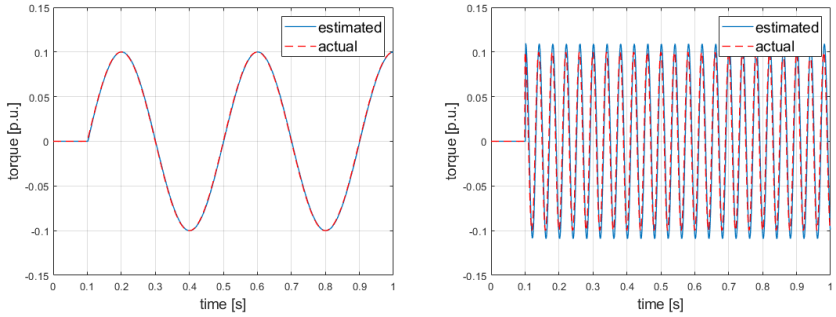


Figure 9.4: Estimation of a periodic load torque at standstill.

ported to visualize the test performed. A step change of the load occurs during the simulation. In the center the disturbance estimation is reported. As already observed, the estimation is good, except close to the transients.

On the right side, the integral of the error between estimated and actual disturbance is shown. In particular, the estimation is unbiased, since the integral seems not to grow.

On the other hand, the estimation of a periodic disturbance is quite poor, see Figure 9.7. On the left, a torque disturbance of angular frequency $0.1\Omega_N$ is pretty well tracked. However, as far as the frequency increases, the estimators performances drop. In fact, the right figure reporting a disturbance of frequency Ω_N reveals an error of magnitude and phase of the estimated signal.

A more realistic control scheme is considered for sake of completeness. In details, it is removed the hypothesis of having ideal torque actuators, both from the motor and the load side. The torque actuator dynamic is described by the following equation:

$$\tau^*(k) = \tau^*(k-1) + \frac{1}{1 + T_s/(2\pi BW_{cl})}(u(k) - \tau^*(k-1)) \tag{9.6}$$

where $u(k)$ is the output of the speed controller, whereas $\tau^*(\cdot)$ is the actual torque applied by the motor. BW_{cl} represent the bandwidth of torque actuator, i.e. the current control loop bandwidth. BW_{cl} is equal to 200Hz in the considered example. In a nutshell, the current loop dynamic is represented as a low pass filter.

In a realistic application, it would be required also a very low overshoot in the speed dynamic Figure 9.3. A filter is often designed for the reference speed, to fulfill this requirement. The filter has the same form of (9.6), with a different bandwidth BW_{ref}. A bandwidth of $BW_{ref} = 20$Hz is used in the simulations. This permits us to avoid saturation of the speed controller integrator. Thus, the estimator should not be affected by non linearities due to input saturation.

The simulation reported in Figure 9.3 is repeated with the new control architecture and results are resumed in Figure 9.5. On the left, a significant improvement of the speed dynamic is noticed, both in

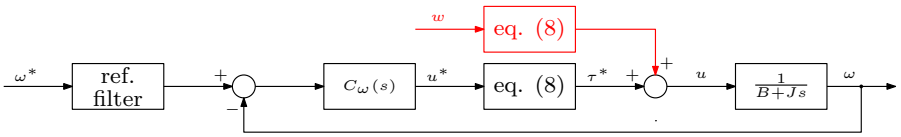


Figure 9.5: Overall control architecture of the speed loop, considering non-ideal torque actuators and a filter on the reference speed.

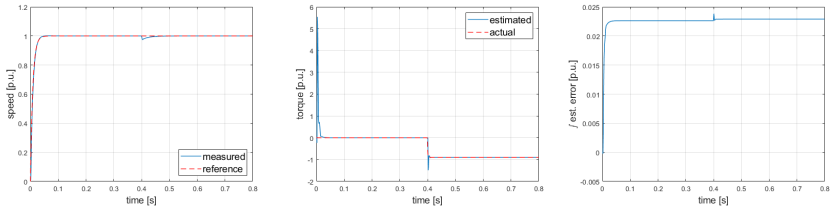


Figure 9.6: Estimation of a constant load torque at nominal speed operation, non-ideal torque actuator case.

terms of step response and step load torque transients. The observer still tracks the disturbance load torque accurately. However, poorer dynamic performances are obtained, with respect to the ideal torque actuator case. This is due to the fact that the actual torque input τ^* is no more accessible. u^*, in fact, is a slightly filtered version of τ^*.

The estimation of a periodic disturbance is considered, too (as in Figure 9.7). Low frequency disturbances are still well tracked, as in the case of ideal torque actuator. Quite similar performances are observed also at high frequency load torques (Figure 9.6, on the right). However, initial dynamic still needs some regularization.

9.1.1.2 *A second example: Load torque estimation for the position control loop of a motor drive*

The same algorithm has been simulated for a slightly different case. In details, the position dynamic is included, adding the equation $\theta(k+1) = \theta(k) + T_s\omega(k)$ to the model defined in (9.5). Shortly, the following state-space model is adopted:

$$\begin{bmatrix} \omega(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} 1 - T_s\frac{B}{J} & 0 \\ T_s & 1 \end{bmatrix} \begin{bmatrix} \omega(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} -\frac{T_s}{J} \\ 0 \end{bmatrix} \tag{9.7}$$

The motor position is controlled by an outer PI controller which generates the speed reference, as shown in Figure 9.8. In this case, the output of the system is the position θ, while the control input is the motor torque, as in the previous example. The output Hankel matrices are built using the position of the motor, instead of the speed ω.

The estimator performances are investigated in case of constant load torque disturbance and during a ramp transient. The position reference is set at $\theta^* = 180$ř. Results are resumed in Figure 9.10 and

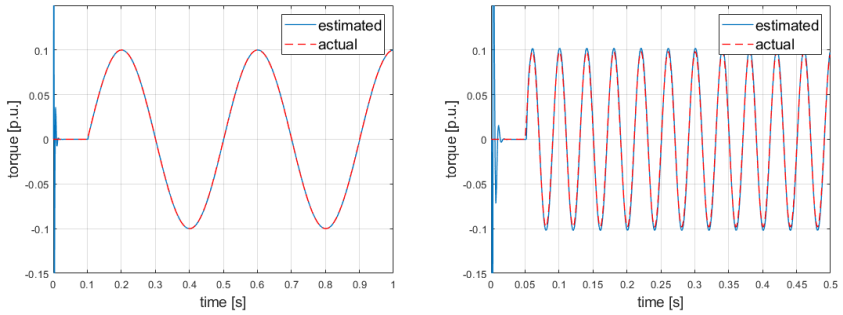


Figure 9.7: Estimation of a periodic load torque at standstill, non-ideal torque actuator.

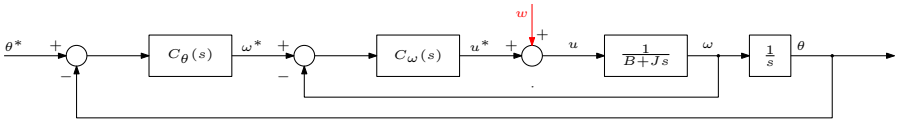


Figure 9.8: Position control problem: block diagram.

9.9. The overall performances of the estimator are good, when the torque disturbance is constant. On the opposite, when a load torque ramp occurs, Figure 9.9 highlights quite poor results. First, a wide spike is noticed in the estimation when the load changes in order to filter the estimation. Thus, a regularized version of the estimation problem is suggested to improve the dynamic. Moreover, during the ramp an increase of the integral estimation error is observed. This suggests that the estimator is not able to track a disturbance ramp with zero integral error.

9.1.2 *Regularized data driven estimator*

In this section the estimation algorithm presented in Sec. 9.1.2 is revised in order to regularize the estimation. As an extra benefit, a tuning parameter is available to regulate the promptness of the estimator.

The estimation problem is rewritten as a quadratic optimization problem. In details, the objectives to reach are a correct output estimation and a smooth disturbance estimation:

$$\min_{\hat{y}(k),w(k-1),g} \left|\left|y(k)-\hat{y}(k)\right|\right|_Q^2 + \left|\left|w(k-1)-w(k-2)\right|\right|_R^2 \tag{9.8}$$

$$\text{s.t.} \begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u(k-1) \\ \hat{y}(k) \end{bmatrix} + \begin{bmatrix} w_{ini} \\ 0^{p(T_{ini}-1)} \\ w(k-1) \\ 0 \end{bmatrix} \tag{9.9}$$

It is worth noticing that the weight matrix R permits to filter the estimated disturbance. The proposed estimation problem is solved in

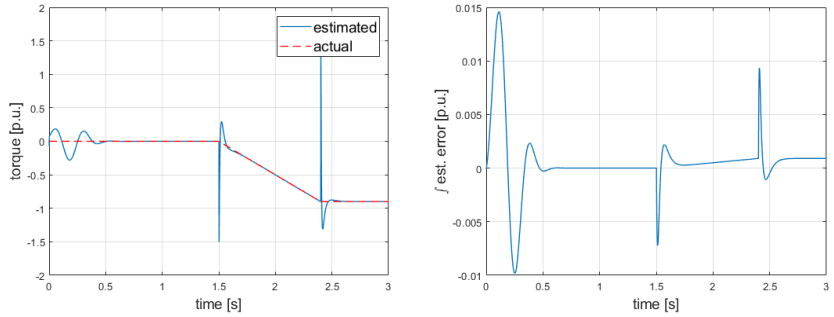


Figure 9.9: Disturbance estimator performances overview.

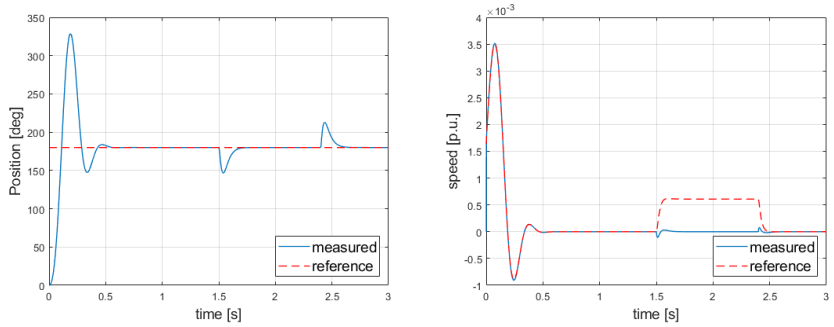


Figure 9.10: Position and speed reference tracking.

g. Thus, the estimated output $\hat{y}(k)$ and $w(k-1)$ are substituted in the optimization problem exploiting the available constraints:

$$\min_{g} \left|\left|y(k) - Y_F g\right|\right|_Q^2 + \left|\left|U_F g - u(k-1) - w(k-2)\right|\right|_R^2 \tag{9.10}$$

$$\text{s.t.} \begin{bmatrix} U_P \\ Y_P \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \begin{bmatrix} w_{ini} \\ 0^{p(T_{ini}-1)} \end{bmatrix} \tag{9.11}$$

The problem can be solved using null-space methods reducing online computations. A basis for the kernel of $[U_P; Y_P]$ is computed offline and a simpler optimization problem is solved online, characterized by a smaller decision variable g'.

9.1.2.1 *Regularized torque disturbance estimation for the speed loop.*

Let's first consider the case of the load torque estimation for the speed control loop, presented in Sec. 9.1.2.1. The estimator is simulated for the case of non-ideal torque actuator and with the filter on the reference speed, described in the just mentioned section.

The benefits of having a regularized estimation problem are evident in case of constant load torque estimation (Figure 9.6). The initial error in the estimation is drastically reduced with respect to the case reported in Figure 9.6. Even the response to a step change of the disturbance is improved.

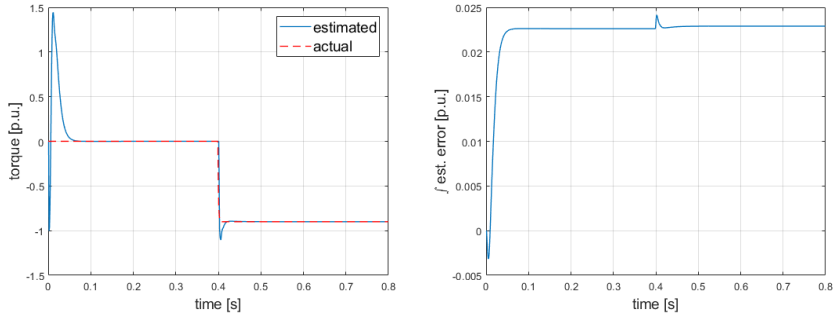


Figure 9.11: Disturbance estimator performances overview with regularization in case of non-ideal torque actuator.

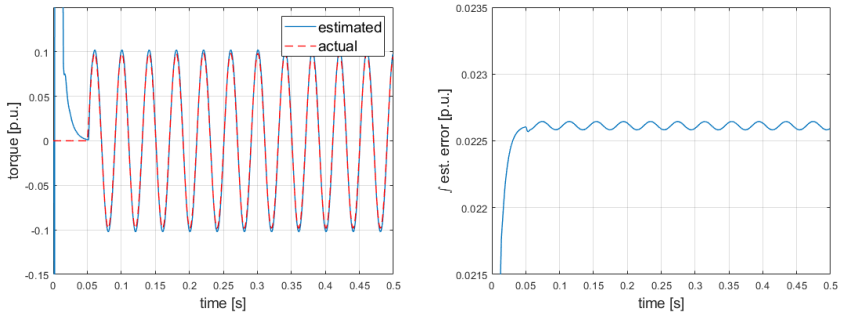


Figure 9.12: Disturbance estimator performances overview with regularization in case of non-ideal torque actuator. Periodic load torque disturbance.

The tuning of R plays a key role if it is desired to track high frequency disturbances, i.e. with an angular frequency close to Ω_N. In Figure 9.12 the same simulation presented in Figure 9.7 is repeated for the regularized algorithm. A lower weight ($1 - 10\%$ of the one used in the previous simulation) should be chosen to obtain a good tracking of the disturbance, keeping also a relatively small integral error. However, this regularized version of the estimator does not permit to perform a closed loop estimation.

9.1.2.2 *Regularized torque disturbance estimation for the position loop.*

The effects of the regularization in the input disturbance estimation is shown for the position control example reported in Sec. 9.1.1.2. The weight related to disturbance variation is set to 1% with respect to the weight on the output estimation.

At first, a simulation is run, with the same conditions of the one whose results are reported in Figure 9.9. The results obtained with the new algorithm are reported in Figure 9.13. The smoothness of the estimation is drastically improved with the new method. In particular, the spikes that were affecting the estimation are almost elided. As in

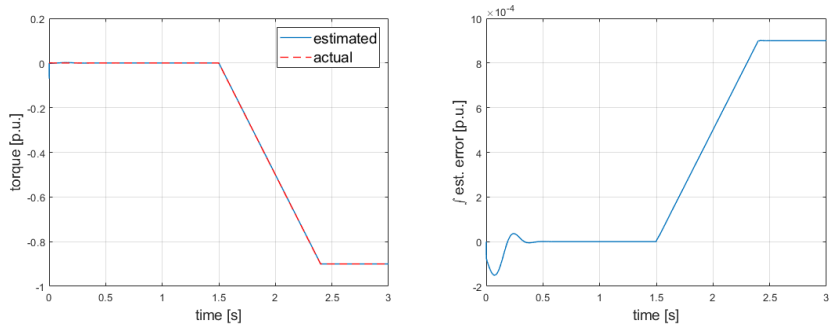


Figure 9.13: Disturbance estimator performances overview with regularization.

the previous case, a rise of the integral error is still noticed during the load torque ramp transient.

9.1.3 *Moving horizon data-driven estimation*

In this section the estimation algorithm presented in Sec. 9.1.2 is generalized for longer estimation horizons, trying to get good closed-loop results. Longer estimation horizons mean that the past input-output trajectories are split in two blocks:

- a $\hat{T}_{ini}$-long sequence, used to define an unique initial condition of the system
- a $\hat{N}$-long sequence, exploited in the estimation problem

This seems a natural generalization of the estimator, considering the dual Data EnablEd Predictive Controller (DeePC). On one hand, the algorithm presented in Sec. 9.1.2 corresponds to a DeePC scheme with a 1-step long prediction horizon. On the other hand, the algorithm presented in this section corresponds to a DeePC scheme with a generic $\hat{N}$-step long prediction horizon.

The estimation problem has to be slightly revised for the new algorithm. In details, a cost function with three different terms: is proposed

$$\min_{\hat{y},w,g} \left|\left|y-\hat{y}\right|\right|_Q^2 + \left|\left|E_1 w - E_2 w_{old}\right|\right|_R^2 + \left|\left|w(k-1) - w_{old}(k-1)\right|\right|_P^2 \tag{9.12}$$

$$\text{s.t.} \begin{bmatrix} U_P \\ Y_P \\ U_F \\ Y_F \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ \hat{y} \end{bmatrix} + \begin{bmatrix} w_{ini} \\ 0^{p\hat{T}_{ini}} \\ w \\ 0 \end{bmatrix} \tag{9.13}$$

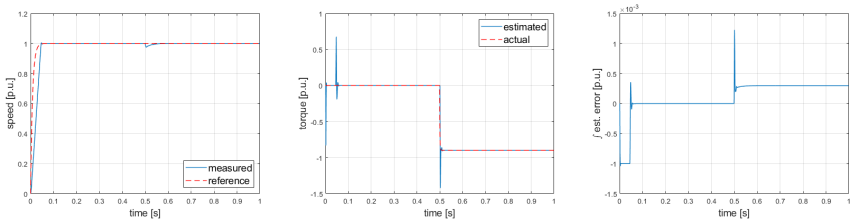


Figure 9.14: Overview of the estimation performances.

where y is the sequence of the last $\hat{N}$ output measurements, w and w_{old} are the sequence of the last $\hat{N}$ estimated disturbances at the k-th and $(k-1)$st step, respectively. Moreover, E_1 is a vector that extract all the element of w except the last one, whereas vector E_2 extracts all the element of w_{old} except the first one. Finally, $w_{old}(k-1)$ is the last estimation of the input disturbance, i.e. the last element of w_{old}.

The first term, as in (9.9), weights the tracking of the measured outputs. The second term weights the trust on the $(k-1)$st estimation. Each step the sequence $w = [w(k-\hat{N}), w(k-\hat{N}+1), \dots, w(k-1)]^T$ is predicted. However, at $(k-1)$st step an estimation for $[w(k-\hat{N}), w(k-\hat{N}+1), \dots, w(k-2)]$ is already available. If the data-driven performs good predictions, the two sequences should be quite close. Finally, the third term regularizes the last estimated disturbance, preventing wide variations.

It results convenient to put together the last two terms, i.e. penalizing $\left|\left|w - [w_{old}; w_{old}(k-1)]\right|\right|^2$ with a different weighting matrix R_P. Then, the output and disturbance expressions are substituted to obtain a function of the decision variable g:

$$\min_g \left|\left|y(k) - Y_F g\right|\right|_Q^2 + \left|\left|U_F g - u - w_{old}\right|\right|_{R_P}^2 \tag{9.14}$$

$$\text{s.t.} \begin{bmatrix} U_P \\ Y_P \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \end{bmatrix} + \begin{bmatrix} w_{ini} \\ 0^p \hat{T}_{ini} \end{bmatrix} \tag{9.15}$$

The obtained problem has the same form of (9.11), therefore it can be solved with the same methods. Thanks to the new formulation more interesting results are expected for the closed loop estimation, as proved by the following simulations.

9.1.3.1 *Load torque estimation for the speed control loop of a motor drive*

The just described algorithm is used for the load torque estimation example presented in Sec. 9.1.1.1. In particular, the more realistic case is considered, with non-ideal torque actuator and filter on the reference speed. A $\hat{T}_{ini} = 5$ and a $\hat{N} = 5$ are adopted.

First, the constant load torque disturbance case is considered. The overview of the simulation in terms of speed dynamic, disturbance estimation and integral of the estimation error are reported in Figure 9.14, as for the previous algorithms.

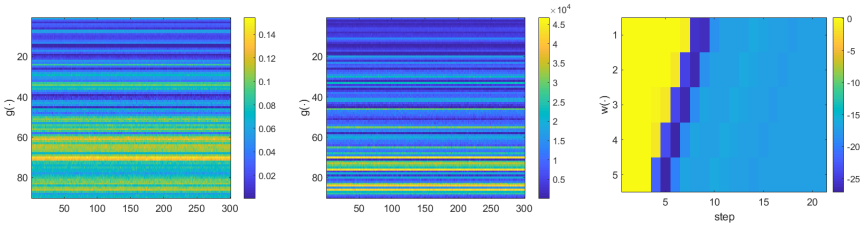


Figure 9.15: Left: vector g coefficients at no load. Center: vector g coefficients with a 90% load torque disturbance. Right: focus on the disturbance vector w estimation during the transients.

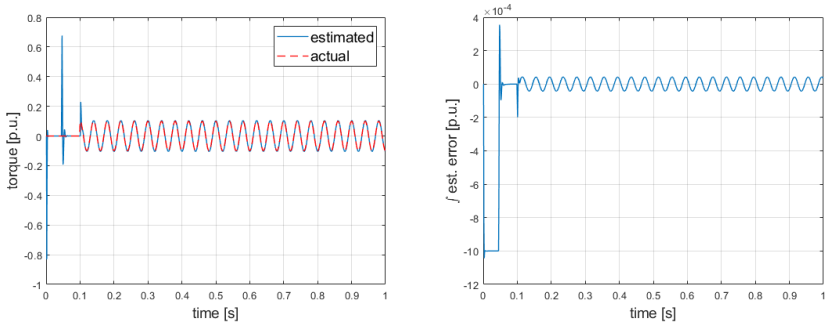


Figure 9.16: Overview of the estimation performances with sinusoidal load torque disturbance with angular frequency ω_N.

More interesting considerations can be done observing the coefficients of g and the estimated disturbance vector w, reported Figure 9.15. The g coefficients are quite constant before and after the load torque level is changed. However, in the loaded case (center) the coefficients magnitudes are much higher, as expected. The transients is analyzed in terms of the estimated disturbance (left). In this case, the estimated disturbance describes a stepped pattern. For instance, the fifth estimated w at the 5th time step is equal to the first estimated w at the 10th step. This confirms that the prediction within the past estimation horizon $\hat{N}$ is quite accurate.

In addition, a simulation regarding the estimation of a sinusoidal load torque of angular frequency Ω_N (Figure 9.16) is proposed, too. As in the previous case overall performances are good but the focus should be on the g coefficients and the estimated disturbance, which are reported in Figure 9.17. First, periodic values of g coefficients are obtained solving the problem (9.13). Since the disturbance is periodic, the decision variable g is periodic, too. Moreover, a nice stepped pattern is observed for the estimated w vector. As for the previous simulation, this means that the estimated disturbance within the past prediction horizon $\hat{N}$ is accurate.

Next step consists into plugging the estimated load torque in closed loop together with the speed controller. As a first step, the integral action is removed in the speed loop, believing that the data-driven estimator can provide this contribute.

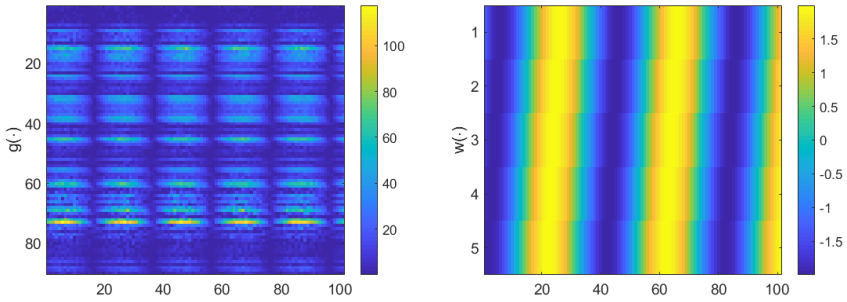


Figure 9.17: Left: vector g coefficients. Right: Estimated w.

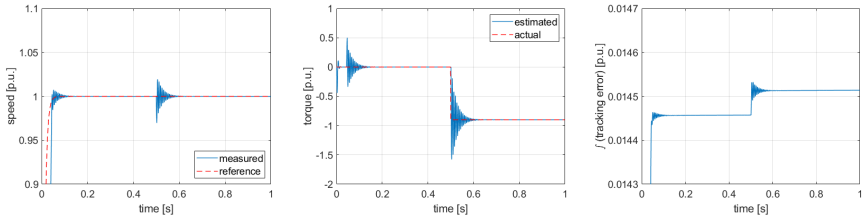


Figure 9.18: Load torque step from 0 to 90% T_N at constant speed operation. Left: speed dynamic. Center: load torque estimation. Right: integral of the speed tracking error.

In general, including the estimator information within the controller results much more tricky, as could be expected. It turns out that all the previous proposed algorithms give not interesting results. In details, the estimated torque has either wide oscillations or it simply blows up. In order to reduce the oscillations, a low pass filter could be implemented in cascade with the estimator, having the same bandwidth of the speed reference filter. However, it is desired to obtain a filtered estimation directly with the optimization problem statement. Using the algorithm presented in this section the filter is no more needed.

Thanks to this filter, it is possible to close the speed loop with the load torque information. In Figure 9.18 the results are summarized for constant torque disturbances and load torque step. The most interesting figure is the one on the right, reporting the speed tracking error integral. In details, the estimator replaces effectively the role of the integrator keeping, preventing the rise of the integral error.

g coefficients and the estimated disturbance vector w are reported in Figure 9.12. The pattern described by the coefficients is slightly different from the one obtained in open loop, shown in Figure 9.15. However, coefficients have similar magnitude. Moreover, coefficients are almost constant for a constant torque, as expected. Then, during the transient (right) the estimated disturbance vector is coherent.

The results related to the periodic load torque disturbance (torque harmonics in the system) are reported in Figure 9.20. The coupling of the estimator with the proportional controller is more aggressive

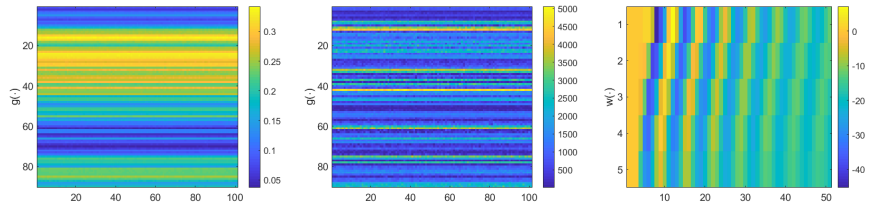


Figure 9.19: Left: vector g coefficients at no load. Center: vector g coefficients with a 90% load torque disturbance. Right: focus on the disturbance vector w estimation during the transients.

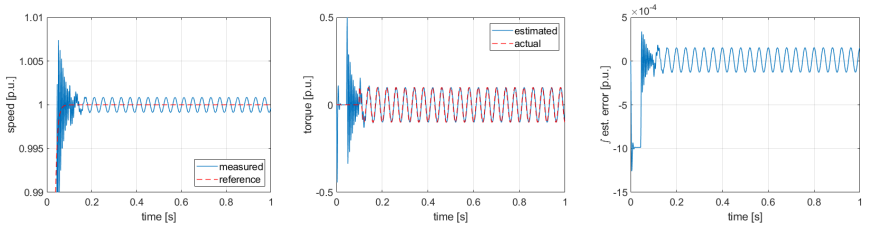


Figure 9.20: Sinusoidal load torque with angular frequency Ω_N and amplitude 10% T_N. Left: speed dynamic. Center: load torque estimation. Right: integral of the speed tracking error.

than the integral action granted by the previous PI controller. This is confirmed by smaller oscillations in the speed. Nevertheless, the torque disturbance information provided by the estimator is quite accurate, as proved for instance by the integral of the estimation error.

The maps of g and w are reported in Figure 9.21, for sake of completeness. In addition, on the right the 2-norm of g for two periods of the torque disturbance is reported. The norm is compared to the one obtained for the open loop estimation (whose results were previously reported in Figure 9.16). As for the open loop estimation of a sinusoidal load, a periodic pattern is observed in g coefficients, which are slightly different in the two cases. However, if the g norm is compared in the two cases, the values of open loop and closed loop estimation are very close (notice that the y-axis is logarithmic). The pattern described by w is quite similar to the one obtained for the open loop case.

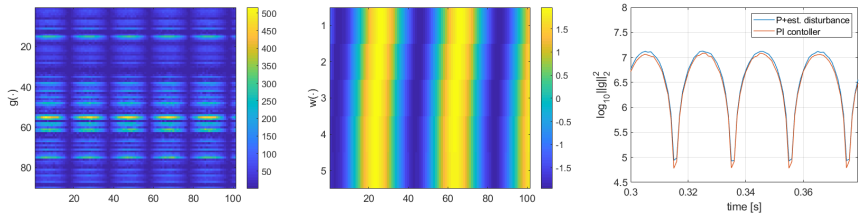


Figure 9.21: Sinusoidal load torque with angular frequency Ω_N and amplitude 10% T_N. Left: vector g coefficients. Center: estimated disturbance vector w. Right: Norm of g comparison between open and closed loop estimator.

10

CONCLUSIONS

The research presented in this work concerns new advances on model predictive control of synchronous motor drives. Particular attention is paid to the effective exploitation of data to possibly enhance the performances of both continuous-set and finite-set architectures.

First finite-set algorithms are considered, focusing on the model-free control branch. An improved finite-set model-free control is described, based on current differences look-up-tables. Its remarkable feature consists in the fact that no knowledge of the motor parameters is assumed. Thus, the scheme results robust against parameters mismatches. Steady state and dynamic performances are slightly inferior, but reasonably comparable, to those of a model-based controller. The developed solution may be convenient if motor manufacturers do not provide detailed machine data and self-commissioning procedures are hard or not convenient to realize. Main contributions with respect to already existent solutions include the reduction of computational cost and the compensation of the motor rotation in the look-up-tables of measured currents. Then, the model-free solution is significantly revisited by setting up a non-parametric model, whose parameters are estimated by recursive least square filters. Relationships between motor working conditions and estimated coefficients are demonstrated, focusing separately on the roles of the speed and the current. Design guidelines on the tuning of the forgetting factors in the recursive least square estimation are provided. Relevant improvements in terms of prediction error and current tracking performances are achieved by this second model-free solution, with respect to previous works presented in literature. Finally, the discrete space vector modulation is effectively coupled to the adaptive non-parametric model. The resulting control architecture allows for a reduction of the current ripple that affects the parameter-free algorithm, with respect to the finite-set version. This, in turn, allows to reduce Joule losses in the stator windings, increasing the efficiency of the drive. Thanks to the deadbeat implementation of the predictive control algorithm, the number of tuning parameters of the proposed algorithm is minimal. Only one commissioning test is needed to tune properly the number of sub-periods of the discrete space vector modulation. The total phase current distortion and the inverter average switching frequency are evaluated for the first time for model-free schemes. The proposed parameter-free discrete space vector modulation algorithm results having a lower current distortion with respect to the finite set version. The method outperforms even model based

ones in many operating points. At the end, this last developed solution emulates, sometimes outperforming, the behavior of model-based controllers, but dribbling all the overwhelming critical issues related to the model identification.

Secondly, continuous-set algorithms are considered, focusing on the new data-driven control branch. The real-time feasibility of some data-driven methods is proved, including the subspace predictive control and data-enabled predictive control. This is not trivial seems these methods are expensive in terms of computation and samples. For this reason, computational aspects regarding constrained and unconstrained predictive controllers are discussed. The problem of the offset-free tracking is addressed for the first time for such class of algorithms. Moreover, the data-collection test that is required by these controllers is studied. Technical suggestions about the excitation signal and mathematical properties are derived, roving that the data-driven procedure can be a systematic design tool also for permanent magnet synchronous motor controllers. Finally, the problem of data-driven estimation of disturbance is considered. A moving horizon estimator is designed, making several considerations on the terms that need to be included in the estimation problem. The estimation of a load torque disturbance is considered as case study for such method. Similar performances are found for model-based and data-driven moving horizon algorithms while estimating constant and sinsuoidal load torque disturbances.

Part IV

APPENDIX

A

CLARKE AND PARK TRANSFORMATION

A.1 CLARKE TRANSFORMATION

the winding of a three-phase motor is distributed in such a way that the spatial displacement of the windings of each phase are shifted from each other by $\frac{2}{3}\pi$ radiants. However, it is convenient to represent the three-phase circuit with an equivalent circuit, whose phase are shifted by $\frac{\pi}{2}$ radiants. The transformation between these two reference frames is known as Clarke transformation. Adopting a matrix representation, its expression is:

$$\begin{bmatrix} s_\alpha(t) \\ s_\beta(t) \\ s_0(t) \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} s_a(t) \\ s_b(t) \\ s_c(t) \end{bmatrix} \rightarrow \mathbf{s}_{\alpha\beta}(t) = \mathbf{T}_{\alpha\beta 0}\mathbf{s}_{abc}(t), \tag{A.1}$$

where $\mathbf{s}_{abc}(t)$ is a generic signal in the three-phase reference frame, e. g. voltages or currents, while $\mathbf{s}_{\alpha\beta}(t)$ is the signal in the new orthogonal reference frame. The reported transformation keeps all signals amplitudes. Thus, the transformation is not power invariant and a factor $\frac{3}{2}$ must be included to preserve the coherence.

The homopolar component $s_0(t)$ is often neglected when considering star-connected three-phase windings, since it holds:

$$s_a(t) + s_b(t) + s_c(t) = 0,$$

because of the Kirchhoff current law. Taking advantage of this further hypothesis, the transformation simplifies to:

$$\begin{bmatrix} s_\alpha(t) \\ s_\beta(t) \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} s_a(t) \\ s_b(t) \\ s_c(t) \end{bmatrix} \rightarrowtail \mathbf{s}_{\alpha\beta}(t) = \mathbf{T}_{\alpha\beta}\mathbf{s}_{abc}(t). \tag{A.2}$$

The simplified inverse transformation is:

$$\begin{bmatrix} s_a(t) \\ s_b(t) \\ s_c(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} s_\alpha(t) \\ s_\beta(t) \end{bmatrix} \rightarrow \mathbf{s}_{abc}(t) = \mathbf{T}_{\alpha\beta}^{\mathsf{T}}\mathbf{s}_{\alpha\beta}(t). \tag{A.3}$$

With some abuse of notation the inverse transformation is often denoted also as $\mathbf{T}_{\alpha\beta}^{-1}$, even if the inverse of a rectangular matrix is not defined.

A.2 PARK TRANSFORMATION

The Park transformation is a transformation introduced to report stator dependent signals to a rotating reference frame. Most of electric quantities in electric motors are position dependent, meaning that their periodicity is related to the electric position θ_e, defined as the product between the rotor position θ_m and the number of pole pairs of the electric machine. Thus, the Park transformation permits to report all these signals in a reference frame where they are constant. This results very useful from the point of view of the control design. The transformation from stator and rotor reference frames is defined as:

$$\begin{bmatrix} s_d(t) \\ s_q(t) \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} s_\alpha(t) \\ s_\beta(t) \end{bmatrix} \rightarrow \mathbf{s}_{dq}(t) = \mathbf{T}_{dq}\mathbf{s}_{\alpha\beta}(t), \quad \text{(A.4)}$$

where $s_d(t)$ and $s_q(t)$ are the signals in the new rotating reference frame, denoted as dq. When permanent magnets are mounted on the rotor, the d axis is often aligned with the permanent magnet flux. The inverse transformation is obtained with a simple matrix inversion, resulting:

$$\begin{bmatrix} s_d(t) \\ s_q(t) \end{bmatrix} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} s_\alpha(t) \\ s_\beta(t) \end{bmatrix} \rightarrow \mathbf{s}_{dq}(t) = \mathbf{T}_{dq}^{-1}\mathbf{s}_{\alpha\beta}(t), \quad \text{(A.5)}$$

It is worth noticing that, being the transformation a rotation, it is orthonormal. In fact, the determinant of $\mathbf{T}_{dq}$ is unitary and the column of the same matrix are orthogonal each others. As a consequence, it holds that $\mathbf{T}_{dq}^{-1} = \mathbf{T}_{dq}^{\top}$.

It could be of interest the computation of the time derivative of the Park transformation:

$$\begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} = \omega_e \begin{bmatrix} -\sin\theta_e & \cos\theta_e \\ -\cos\theta_e & -\sin\theta_e \end{bmatrix} = \omega_e \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{T}_{dq}, \quad \text{(A.6)}$$

which is the product of the electric speed ω_e, a skew-symmetric matrix and the Park transformation matrix. In other words, it can be interpreted as the product between the electric speed, a rotation of $\frac{\pi}{2}$ and $\mathbf{T}_{dq}$.

B

BLACK BOX AND GREY BOX MODELS

The model-free and the parameter-free control exploit model structures which are typical of the *system identification* framework [**Ljung**]. System identification is a procedure to build a mathematical model of the dynamics of a system from measured data. The identified models can be of different types, depending on how much information is exploited about the physics behind the dynamics. It is possible to distinguish between:

- *white box* models if the model structure is based on physics principles, whose parameters are estimated from measured data;
- *grey box* models, if the model is partially built from physics equation and the rest is inferred from data;
- *black box* models, if both the model structure and its parameters are unknown and they are directly estimated from I/O data.

The model adopted by the parameter-free control (6.7) is a customized grey-box model. Its expression is here derived starting from the general black box structure.

According to the black box structure, the output of a plant y at the k-th time instant can be estimated by means of the previous outputs and inputs $\mathbf{u}$ applied to the plant:

$$\begin{aligned} y(k) = &- a_1 y(k-1) - a_2 y(k-2) - \cdots - a_n y(k-n) \\ &+ b_1 u(k-1) + b_2 u(k-2) + \cdots + b_m u(k-m) + w(k) \end{aligned} \tag{B.1}$$

where $\boldsymbol{w}(k)$ denotes the output noise or, equivalently, the effect of a disturbance v acting on the plant dynamics, i.e. $w(k) = v(k) - a_1 v(k-1) - a_2 v(k-2) - \cdots - a_n \mathbf{y}(k-n)$. Considering the voltage balance equation of a PMSM (2.10), this disturbance comprehends at least the back-EMF induced by the permanent magnets. In particular this disturbance has a non-zero mean.

The more general class of black box model suitable to describe the currents dynamics of a synchronous motor is the Box-Jenkins one:

$$y(k) = \frac{B(z)}{A(z)} u(k) + \frac{C(z)}{D(z)} e(k) \tag{B.2}$$

where, adopting the z-transform notation,

$$\begin{aligned} A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n} \\ B(z) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-m} \\ C(z) &= 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_n z^{-q} \\ D(z) &= 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots + d_n z^{-r} \end{aligned} \tag{B.3}$$

and $e(k)$ is a zero mean white noise.

Assuming the polinomia $A(z)$ and $D(z)$ equal and $C(z)$ unitary, the ARX model is obtained:

$$A(z)y(z) = B(z)u(z) + e(z) \tag{B.4}$$

The grey-box model behind the model-free control is obtained by first fixing the degree of the non-zero polinomia in (B.3). In details, it is set $n = m = 1$, obtaining

$$y(k) = -a_1 y(k-1) + b_0 u(k) + b_1 u(k-1) + w(k) \tag{B.5}$$

Then, some information regarding the actual plant is exploited to fix two of the parameters of the just presented black box model. Since a digital delay of one control step occurs in the digital implementation of FS MPCs, the term b_1 is null. In other words, being at the k-th time step, the controller computes a control action based on the k-th current measurement, which is influenced by the previously applied voltage. Finally, a further simplification of the black box model is achieved by imposing $a_1 = 1$.

The equivalence between the obtained grey-box model and the model exploited for instance by the parameter-free controller is here clarified:

$$\begin{aligned} i_d(k) = y(k) &= i_d(k-1) + p_{2,d} u_d(k-1) + p_{1,d} \\ &= y(k-1) + b_1 u(k-1) + w(k) \end{aligned} \tag{B.6}$$

BIBLIOGRAPHY

[1] Pablo Acuna, Christian A. Rojas, Roky Baidya, Ricardo P. Aguilera, and John E. Fletcher. "On the Impact of Transients on Multistep Model Predictive Control for Medium-Voltage Drives." In: 34.9 (2019), pp. 8342–8355. DOI: 10.1109/TPEL.2018.2889565.

[2] 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.

[3] R. Antonello, M. Carraro, L. Peretti, and M. Zigliotto. "Hierarchical Scaled-States Direct Predictive Control of Synchronous Reluctance Motor Drives." In: *IEEE Transactions on Industrial Electronics* 63.8 (2016), pp. 5176–5185. ISSN: 0278-0046. DOI: 10.1109/TIE.2016.2536581.

[4] R. Antonello, L. Ortombina, F. Tinazzi, and M. Zigliotto. "Online Stator Resistance Tracking for Reluctance and Interior Permanent Magnet Synchronous Motors." In: *IEEE Transactions on Industry Applications* 54.4 (2018), pp. 3405–3414. ISSN: 0093-9994. DOI: 10.1109/TIA.2018.2819961.

[5] N. Bedetti, S. Calligaro, and R. Petrella. "Stand-Still Self-Identification of Flux Characteristics for Synchronous Reluctance Machines Using Novel Saturation Approximating Function and Multiple Linear Regression." In: *IEEE Transactions on Industry Applications* 52.4 (2016), pp. 3083–3092. ISSN: 0093-9994. DOI: 10.1109/TIA.2016.2535413.

[6] A. Boglietti, A. Cavagnino, and M. Lazzari. "Experimental High-Frequency Parameter Identification of AC Electrical Motors." In: *IEEE Transactions on Industry Applications* 43.1 (2007), pp. 23–29. DOI: 10.1109/TIA.2006.887313.

[7] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto. "Design and Implementation of Model Predictive Control for Electrical Motor Drives." In: *IEEE Transactions on Industrial Electronics* 56.6 (2009), pp. 1925–1936. DOI: 10.1109/TIE.2008.2007547.

[8] S. Bolognani, P. G. Carlet, F. Tinazzi, and M. Zigliotto. "Fast and Robust Model Free Predictive Current Control for SynREL Motor Drives." In: *2018 IEEE ECCE*. 2018, pp. 5466–5472. DOI: 10.1109/ECCE.2018.8558321.

[9] 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.

[10] 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 Transactions on Power Electronics* 36.2 (2021), pp. 2179–2190. DOI: 10.1109/TPEL.2020.3006779.

[11] 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.

[12] Paolo Gherardo Carlet, Fabio Tinazzi, Silverio Bolognani, and Mauro Zigliotto. "An Effective Model-Free Predictive Current Control for Synchronous Reluctance Motor Drives." In: *IEEE Transactions on Industry Applications* 55.4 (2019), pp. 3781–3790. DOI: 10.1109/TIA.2019.2910494.

[13] Paolo Gherardo Carlet, Francesco Toso, Andrea Favato, and Silverio 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.

[14] D. Casadei, G. Serra, and K. Tani. "Implementation of a direct control algorithm for induction motors based on discrete space vector modulation." In: *IEEE Transactions on Power Electronics* 15.4 (2000), pp. 769–777. DOI: 10.1109/63.849048.

[15] 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.

[16] 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.

[17] 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.

[18] Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-Enabled Predictive Control: In the Shallows of the DeePC." In: *18th Eur. Control Conf. (ECC)* (2018).

[19] Jeremy Coulson, John Lygeros, and Florian Dörfler. *Distributionally Robust Chance Constrained Data-enabled Predictive Control*. 2020. arXiv: 2006.01702 [math.OC].

[20] D. Da Rú, M. Polato, and S. Bolognani. "Model-free predictive current control for a SynRM drive based on an effective update of measured current responses." In: *2017 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*. 2017, pp. 119–124. DOI: 10.1109/PRECEDE.2017.8071279.

[21] 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 Transactions on Industry Applications* 54.2 (2018), pp. 1437–1447. DOI: 10.1109/TIA.2017.2772171.

[22] T. Dragičević and M. Novak. "Weighting Factor Design in Model Predictive Control of Power Electronic Converters: An Artificial Neural Network Approach." In: *IEEE Transactions on Industrial Electronics* 66.11 (2019), pp. 8870–8880.

[23] Florian Dörfler, Jeremy Coulson, and Ivan Markovsky. *Bridging direct & indirect data-driven control formulations via regularizations and relaxations*. 2021. arXiv: 2101.01273 [math.OC].

[24] J. Falck, G. Buticchi, and M. Liserre. "Thermal Stress Based Model Predictive Control of Electric Drives." In: *IEEE Transactions on Industry Applications* 54.2 (2018), pp. 1513–1522. ISSN: 0093-9994. DOI: 10.1109/TIA.2017.2772198.

[25] A. Favato, P. G. Carlet, F. Toso, and S. Bolognani. "A Model Predictive Control for Synchronous Motor Drive with Integral Action." In: *44th Annu. Conf. of the IEEE Ind. Electron. Soc. (IECON)*. 2018. DOI: 10.1109/IECON.2018.8591769.

[26] Wouter Favoreel, Bart De Moor, and Michel 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.

[27] G. Feng, C. Lai, and N. C. Kar. "A Novel Current Injection-Based Online Parameter Estimation Method for PMSMs Considering Magnetic Saturation." In: 52.7 (2016), pp. 1–4. ISSN: 0018-9464. DOI: 10.1109/TMAG.2016.2525805.

[28] H. J. Ferreau, H. G. Bock, and M. Diehl. "An online active set strategy to overcome the limitations of explicit MPC." In: *International Journal of Robust and Nonlinear Control* 18.8 (), pp. 816–830. DOI: 10.1002/rnc.1251. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rnc.1251. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.1251.

[29] Hans Joachim Ferreau, Christian Kirches, Andreas Potschka, Hans Georg Bock, and Moritz 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.

[30] Gianluca Frison and Moritz Diehl. *HPIPM: a high-performance quadratic programming framework for model predictive control*. 2020. arXiv: 2003.02547 [math.OC].

[31] Cristian Garcia, Cesar Silva, Jose Rodriguez, and Pericle Zanchetta. "Cascaded model predictive speed control of a permanent magnet synchronous machine." In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 2016, pp. 2714–2718. DOI: 10.1109/IECON.2016.7793151.

[32] T. Geyer. "Algebraic Tuning Guidelines for Model Predictive Torque and Flux Control." In: *IEEE Transactions on Industry Applications* 54.5 (2018), pp. 4464–4475. ISSN: 0093-9994. DOI: 10.1109/TIA.2018.2835375.

[33] L. Guo, N. Jin, C. Gan, L. Xu, and Q. Wang. "An Improved Model Predictive Control Strategy to Reduce Common-Mode Voltage for Two-Level Voltage Source Inverters Considering Dead-Time Effects." In: *IEEE Transactions on Industrial Electronics* 66.5 (2019), pp. 3561–3572. ISSN: 0278-0046. DOI: 10.1109/TIE.2018.2856194.

[34] Sören Hanke, Oliver Wallscheid, and Joachim Böcker. "Continuous-Control-Set Model Predictive Control with Integrated Modulator in Permanent Magnet Synchronous Motor Applications." In: *2019 IEEE International Electric Machines Drives Conference (IEMDC)*. 2019, pp. 2210–2216. DOI: 10.1109/IEMDC.2019.8785122.

[35] M. Hinkkanen, P. Pescetto, E. Mölsä, S. E. Saarakkala, G. Pellegrino, and R. Bojoi. "Sensorless Self-Commissioning of Synchronous Reluctance Motors at Standstill Without Rotor Locking." In: *IEEE Transactions on Industry Applications* 53.3 (2017), pp. 2120–2129. ISSN: 0093-9994. DOI: 10.1109/TIA.2016.2644624.

[36] Marko Hinkkanen, Zengcai Qu, Hafiz Asad Ali Awan, Toni Tuovinen, and Fernando Briz. "Current control for IPMSM drives: Direct discrete-time pole-placement design." In: *2015 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)*. 2015, pp. 156–164. DOI: 10.1109/WEMDCD.2015.7194524.

[37] Zhong Sheng Hou and Zhuo 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.

[38] Biao Huang and Ramesh 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.

[39] 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.

[40] Linbin Huang, Jianzhe Zhen, John Lygeros, and Florian Dörfler. *Quadratic Regularization of Data-Enabled Predictive Control: Theory and Application to Power Converter Experiments*. 2020. arXiv: 2012.04434 [eess.SY].

[41] M. Jofré, A. M. Llor, and C. A. Silva. "Sensorless Low Switching Frequency Explicit Model Predictive Control of Induction Machines Fed by Neutral Point Clamped Inverter." In: *IEEE Transactions on Industrial Electronics* 66.12 (2019), pp. 9122–9128. DOI: 10.1109/TIE.2018.2885717.

[42] 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.

[43] Ramesh Kadali, Biao Huang, and Anthony 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.

[44] P. Karamanakos and T. Geyer. "Guidelines for the Design of Finite Control Set Model Predictive Controllers." In: *IEEE Transactions on Power Electronics* 35.7 (2020), pp. 7434–7450.

[45] 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 Industry Applications* 1 (2020), pp. 95–114. DOI: 10.1109/OJIA.2020.3020184.

[46] Mohammad Khalilzadeh, Sadegh Vaez-Zadeh, Jose Rodriguez, and Rasool Heydari. "Model-Free Predictive Control of Motor Drives and Power Converters: A Review." In: *IEEE Access* 9 (2021), pp. 105733–105747. DOI: 10.1109/ACCESS.2021.3098946.

[47] S. Kim, Y. D. Yoon, S. K. Sul, and K. Ide. "Maximum Torque per Ampere (MTPA) Control of an IPM Machine Based on Signal Injection Considering Inductance Saturation." In: *IEEE Transactions on Power Electronics* 28.1 (2013), pp. 488–497. ISSN: 0885-8993. DOI: 10.1109/TPEL.2012.2195203.

[48] Werner Leonhard. *Control of Electrical Drives*. Springer, 2012. ISBN: 9783642976469.

[49] C. Lin, T. Liu, J. Yu, L. Fu, and C. Hsiao. "Model-Free Predictive Current Control for Interior Permanent-Magnet Synchronous Motor Drives Based on Current Difference Detection Technique." In: *IEEE Transactions on Industrial Electronics* 61.2 (2014), pp. 667–681. ISSN: 0278-0046. DOI: 10.1109/TIE.2013.2253065.

[50] C. Lin, J. Yu, Y. Lai, and H. Yu. "Improved Model-Free Predictive Current Control for Synchronous Reluctance Motor Drives." In: *IEEE Transactions on Industrial Electronics* 63.6 (2016), pp. 3942–3953.

[51] J. Liu, C. Gong, Z. Han, and H. Yu. "IPMSM Model Predictive Control in Flux-Weakening Operation Using an Improved Algorithm." In: *IEEE Transactions on Industrial Electronics* 65.12 (2018), pp. 9378–9387. ISSN: 0278-0046. DOI: 10.1109/TIE.2018.2818640.

[52] 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 Transactions on Power Electronics* 35.10 (2020), pp. 10778–10788. DOI: 10.1109/TPEL.2020.2980930.

[53] N. A. Losic and L. D. Varga. "A current-free and parameter-free control algorithm." In: *IEEE Transactions on Industry Applications* 30.2 (1994), pp. 324–332. DOI: 10.1109/28.287527.

[54] Y. Luo and C. Liu. "A Simplified Model Predictive Control for a Dual Three-Phase PMSM With Reduced Harmonic Currents." In: *IEEE Transactions on Industrial Electronics* 65.11 (2018), pp. 9079–9089. ISSN: 0278-0046. DOI: 10.1109/TIE.2018.2814013.

[55] 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.

[56] Ivan Markovsky and Florian Dörfler. *Identifiability in the behavioral setting*. 2020.

[57] Ivan Markovsky and Paolo Rapisarda. "Data-driven simulation and control." In: *Int. J. Control* 81.12 (2008), pp. 1946–1959. ISSN: 00207179. DOI: 10.1080/00207170801942170.

[58] S. Nalakath, M. Preindl, and A. Emadi. "Online multi-parameter estimation of interior permanent magnet motor drives with finite control set model predictive control." In: *IET Electric Power Appl.* 11.5 (2017), pp. 944–951. ISSN: 1751-8679. DOI: 10.1049/iet-epa.2016.0514.

[59] 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 Transactions on Power Electronics* 34.4 (2019), pp. 3603–3614. DOI: 10.1109/TPEL.2018.2856589.

[60] L. Ortombina, F. Tinazzi, and M. Zigliotto. "Adaptive Maximum Torque per Ampere Control of Synchronous Reluctance Motors by Radial Basis Function Networks." In: *IEEE Journal of Emerging and Selected Topics in Power Electronics* (2018), pp. 1–1. ISSN: 2168-6777. DOI: 10.1109/JESTPE.2018.2858842.

[61] L. Ortombina, F. Tinazzi, and M. Zigliotto. "Magnetic Modeling of Synchronous Reluctance and Internal Permanent Magnet Motors Using Radial Basis Function Networks." In: *IEEE Transactions on Industrial Electronics* 65.2 (2018), pp. 1140–1148. ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2733502.

[62] I. Osman, D. Xiao, M. F. Rahman, M. Norambuena, and J. Rodriguez. "Discrete Space Vector Modulation Based Model Predictive Flux Control with Reduced Switching Frequency for IM Drive." In: (2020), pp. 1–1. DOI: 10.1109/TEC.2020.3033356.

[63] G. Pannocchia, M. Gabiccini, and A. Artoni. "Offset-free MPC explained: novelties, subtleties, and applications." In: *IFAC Papers OnLine* 48.23 (2015), pp. 342 –351. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.11.304.

[64] Gianmario Pellegrino, Thomas M. Jahns, Nicola Bianchi, Wen Soong, and Francesco Cupertino. *The Rediscovery of Synchronous Reluctance and Ferrite Permanent Magnet Motors*. 1st ed. Springer International Publishing, 2016.

[65] L. Peretti, P. Sandulescu, and G. Zanuso. "Self-commissioning of flux linkage curves of synchronous reluctance machines in quasi-standstill condition." In: *IET Electric Power Applications* 9.9 (2015), pp. 642–651. ISSN: 1751-8660. DOI: 10.1049/iet-epa.2015.0070.

[66] M. Preindl and S. Bolognani. "Comparison of direct and PWM model predictive control for power electronic and drive systems." In: *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*. 2013, pp. 2526–2533. DOI: 10.1109/APEC.2013.6520651.

[67] 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.

[68] M. Pulvirenti, G. Scarcella, G. Scelba, A. Testa, and M. Harbaugh. "On-Line Stator Resistance and Permanent Magnet Flux Linkage Identification on Open-end Winding PMSM Drives." In: *IEEE Transactions on Industry Applications* (2018), pp. 1–1. ISSN: 0093-9994. DOI: 10.1109/TIA.2018.2869877.

[69] J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young, and C. A. Rojas. "State of the Art of Finite Control Set Model Predictive Control in Power Electronics." In: 9.2 (2013), pp. 1003–1016. ISSN: 1551-3203. DOI: 10.1109/TII.2012.2221469.

[70] Torsten Söderström and Petre Stoica. *System Identification*. 1st ed. Prentice Hall International, 1989.

[71] 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.

[72] Marius Stender, Oliver Wallscheid, and Joachim Böcker. "Accurate Torque Estimation for Induction Motors by Utilizing a Hybrid Machine Learning Approach." In: *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*. 2021, pp. 390–397. DOI: 10.1109/PEMC48073.2021.9432615.

[73] 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 Transactions on Industrial Electronics* (2019), pp. 1–1.

[74] Geyer Tobias. *Model Predictive Control of High Power Converters and Industrial Drives*. John Wiley & Sons, Ltd, Sept. 2016. ISBN: 978-1-119-01090-6. DOI: 10.1002/9781119010883.

[75] 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.

[76] Francesco Toso, Paolo Gherardo Carlet, Andrea Favato, and Silverio Bolognani. "On-line Continuous Control Set MPC for PMSM drives current loops at high sampling rate using qpOASES." In: *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2019, pp. 6615–6620. DOI: 10.1109/ECCE.2019.8912838.

[77] 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 Transactions on Industrial Electronics* 64.2 (2017), pp. 935–947. DOI: 10.1109/TIE.2016.2625238.

[78] Kim P. Wabersich and Melanie N. Zeilinger. *Performance and safety of Bayesian model predictive control: Scalable model-based RL with guarantees*. 2020. arXiv: 2006.03483 [eess.SY].

[79] F. Wang, X. Mei, J. Rodriguez, and R. Kennel. "Model predictive control for electrical drive systems-an overview." In: *CES Transactions on Electrical Machines and Systems* 1.3 (2017), pp. 219–230.

[80] 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 Transactions on Power Electronics* 35.12 (2020), pp. 13652–13662. DOI: 10.1109/TPEL.2020.2994948.

[81] 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 Transactions on Power Electronics* 34.11 (2019), pp. 11103–11115. DOI: 10.1109/TPEL.2019.2900559.

[82] Y. Wang, X. Wang, W. Xie, F. Wang, M. Dou, R. M. Kennel, R. D. Lorenz, and D. Gerling. "Deadbeat Model-Predictive Torque Control With Discrete Space-Vector Modulation for PMSM Drives." In: *IEEE Transactions on Industrial Electronics* 64.5 (2017), pp. 3537–3547. DOI: 10.1109/TIE.2017.2652338.

[83] Yawei Wang, Mattia Filippini, Giacomo Bacco, and Nicola Bianchi. "Parametric Design and Optimization of Magnetic Gears With Differential Evolution Method." In: *IEEE Transactions on Industry Applications* 55.4 (2019), pp. 3445–3452. DOI: 10.1109/TIA.2019.2901774.

[84] Sebastian Wendel, Barnabas Haucke-Korber, Armin Dietz, and Ralph Kennel. "Cascaded Continuous and Finite Model Predictive Speed Control for Electrical Drives." In: *2018 20th European Conference on Power Electronics and Applications (EPE'18 ECCE Europe)*. 2018, P.1–P.10.

[85] Jan Willems, Paolo Rapisarda, Ivan Markovsky, and Bart De Moor. "A note on persistency of excitation." In: *Syst. Control Lett.* 54 (Apr. 2005). DOI: 10.1016/j.sysconle.2004.09.003.

[86] 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 Transactions on Power Electronics* 32.12 (2017), pp. 9267–9279. DOI: 10.1109/TPEL.2017.2654540.

[87] S. Yin, H. Gao, and O. Kaynak. "Data-Driven Control and Process Monitoring for Industrial Applications - Part I." In: *IEEE Transactions on Industrial Electronics* 61.11 (2014), pp. 6356–6359.

[88] P. Zanchetta. "Heuristic multi-objective optimization for cost function weights selection in finite states model predictive control." In: *2011 Workshop on Predictive Control of Electrical Drives and Power Electronics*. 2011, pp. 70–75. DOI: 10.1109/PRECEDE.2011.6078690.

[89] 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.

[90] 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.

[91] Andrea Zanelli, Julian Kullick, Hisham M. Eldeeb, Gianluca Frison, Christoph M. Hackl, and Moritz Diehl. "Continuous Control Set Nonlinear Model Predictive Control of Reluctance Synchronous Machines." In: (2021), pp. 1–12. DOI: 10.1109/TCST.2020.3043956.

[92] X. Zhang, L. Zhang, and Y. Zhang. "Model Predictive Current Control for PMSM Drives With Parameter Robustness Improvement." In: *IEEE Transactions on Power Electronics* 34.2 (2019), pp. 1645–1657. DOI: 10.1109/TPEL.2018.2835835.

[93] 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 Transactions on Industrial Electronics* 68.2 (2021), pp. 993–1003. DOI: 10.1109/TIE.2020.2970660.

[94] Yongchang Zhang, Yuning Bai, and Haitao Yang. "A Universal Multiple-Vector-Based Model Predictive Control of Induction Motor Drives." In: *IEEE Transactions on Power Electronics* 33.8 (2018), pp. 6957–6969. DOI: 10.1109/TPEL.2017.2754324.

[95] Yongchang Zhang, Hao Jiang, and Haitao Yang. "A Universal Discrete Space Vector Modulation Based Model Predictive Control for PMSM Drives." In: *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*. 2019, pp. 1–6. DOI: 10.1109/ICEMS.2019.8921678.

[96] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi. "Willems' Fundamental Lemma for State-Space Systems and Its Extension to Multiple Datasets." In: *IEEE Control Systems Letters* 4.3 (2020), pp. 602–607. DOI: 10.1109/LCSYS.2020.2986991.

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 November 30, 2021 (`classicthesis` version 2.0).