LUDOVICO ORTOMBINA

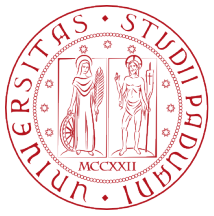
INNOVATIVE SOLUTIONS FOR CONVERTERS AND MOTOR DRIVES ORIENTED TO SMART CITIES AND COMMUNITIES

Ludovico Ortombina:
Innovative Solutions For Converters and Motor Drives Oriented to Smart Cities and Communities,

SUPERVISORS:
Mauro Zigliotto

LOCATION:
Vicenza

TIME FRAME:
March 2019



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede amministrativa: Università degli Studi di Padova

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

SCUOLA DI DOTTORATO IN INGEGNERIA MECCATRONICA E DELL'INNOVAZIONE MECCANICA DEL PRODOTTO

CICLO XXXI

INNOVATIVE SOLUTIONS FOR CONVERTERS AND MOTOR DRIVES ORIENTED TO SMART CITIES AND COMMUNITIES

Direttore della scuola: Ch.ma Prof.ssa Daria Battini

Relatore: Ch.mo Prof. Mauro Zigliotto

Dottorando: Ludovico Ortombina

To her for always being in my thoughts

ABSTRACT

Smart cities and communities are conjugated by European Union in different areas, including energy efficiency, low carbon technologies and mobility which are deeply merged with electric motors. Electric machines are ubiquitous in industry for a wide range of applications, consuming between 43 % and 46 % of all electricity that is generated in the world. Although some machines are used for high–performance applications, such as robots and machine tools, the majority are used in industrial processes for pumps, compressors, fans, conveyors, and other slower–dynamic applications. It is estimated that 92 % – 95 % of the life cycle costs of electric motors are associated with the energy they consume, leading to typical payback periods of < 2 years for the installation of an adjustable–speed drive. It is rather surprising to learn that, despite overwhelming evidence of the attainable savings, only 10 % – 15 % of all industrial motors presently use electronic adjustable speed drives. On the motor side, synchronous reluctance (SynR) motors are gaining lots of attention from industrial researchers and academics, due to their inherent characteristics like the high efficiency, the low cost and the low environmental footprint. Their characteristics fully meet the requirements imposed by smart cities and communities and the aforementioned low–dynamics applications, so they could be the heart of the revamping of those plants. There is wide agreement that the potential for future growth in the sales of industrial drives and SynR motors is still very substantial.

SynR motors are prone to magnetic saturation, making the classic model with lumped parameters unsuitable. The main part of this thesis concerns the development of a new magnetic model for anisotropic motors, especially for SynR motors. It is based on a special kind of neural network (NN), called radial basis function (RBF) NN, which is particularly advisable for an online updating due to its local property. A complete training procedure is proposed in which some considerations are done to define several NN parameters and to convert the nonlinear training problem into a linear one. Two different training algorithms are presented, the former one is fast but computationally cumbersome then suitable for an offline training while the latter one is lighter then proper for an online training. In order to complete the online parameters identification, a scheme based on a DC current injection is developed to estimate the stator resistance. An exhaustive analysis is carried out to disclose that the proposed method is independent from other motor parameters which is a strength asset in a saturable motor. An accurate stator resistance value improves in turn of the magnetic model.

The second part of this dissertation deals with how to exploit an accurate magnetic model to enhance the motor control. In order to improve the efficiency of the motor, exploiting the RBF NN model and the online training algorithm, the maximum torque per ampere (MTPA) curve is found. Starting from a blank NN, it is continuously online trained and a proper algorithm understands where the MTPA curve is respect to the current working point. Afterwards, the drive moves itself towards the actual MTPA. Finally, three different current control schemes tailored for anisotropic motors are presented, all based on the available NN–based magnetic model. The first one is a gain–scheduling proportional–integral (PI) control where the control gains are accordingly tuned to the working point to keep constant the control bandwidth. The second one is based on a classical PI regulator with a feed–forward (FF) action to compensate for all the nonlinearity of magnetic maps. The third one is a constrained direct model predictive control (MPC) where a long prediction horizon is achieved. In order to accomplish a long prediction horizon, the Sphere Decoding algorithm (SDA) is properly modified to make it suitable for a nonlinear system.

The whole thesis was fully validated through an intensive simulation and experimental stage, except the long–horizon MPC which was tested only by simulation.

This thesis was typeset adopting the typographical *classicthesis* style developed by André Miede.

SOMMARIO

Alcune aree definite dall'Unione Europea nel contesto delle *smart cities and communities* si fondono pienamente con i motori elettrici come, per esempio, l'efficienza energetica, le tecnologie a basse emissioni di carbonio e la mobilità. I motori elettrici sono utilizzati in molteplici applicazioni industriali e non, consumando tra il 43 % e il 46 % dell'energia elettrica prodotta su scala mondiale. Nonostante alcune applicazioni siano contraddistinte da dinamiche elevate, come manipolatori o macchine utensili, la maggior parte di esse sono caratterizzate da basse dinamiche in quanto facenti parte di processi industriali, per esempio pompe, compressori, ventilatori o nastri trasportatori. Si è stimato che il costo dell'intero ciclo di vita di un motore elettrico è ascrivibile per il 92 % – 95 % all'energia consumata, il che indurrebbe un tempo di ritorno dall'investimento per installazione di un azionamento elettrico minore di due anni. Nonostante il notevole risparmio economico e ambientale ottenibile, è piuttosto sorprendente apprendere che solo il 10 % – 15 % di tutti i motori industriali siano controllati da azionamenti elettrici.

Per quanto riguarda le diverse tecnologie di motori elettrici, i motori sincroni a riluttanza stanno ricevendo una notevole attenzione sia da ricercatori industriali che accademici. Il crescente interesse è principalmente motivato dalle loro intrinseche caratteristiche quali l'alta efficienza, il basso costo e il basso impatto ambientale dovuto alla mancanza di magneti permanenti. Per di più, le loro caratteristiche soddisfano appieno i requisiti imposti dalle *smart cities and communities* e sono adatti per tutte le applicazione, caratterizzate da una bassa dinamica, viste sopra. Per questi motivi, questa tecnologia di motori può essere posta al centro dei processi di rinnovamento di quelle applicazioni. Vi è ampio consenso sul potenziale incremento delle vendite sia di azionamenti elettrici che di motori sincroni a riluttanza.

I motori sincroni a riluttanza sono soggetti a una marcata saturazione magnetica, rendendo i classici modelli a parametri concentrati poco adatti. La prima parte di questa tesi riguarda lo sviluppo di un innovativo modello magnetico per motori anisotropi. Si basa su una rete neurale non tradizionale, chiamata *Radial Basis Function*. La sua proprietà locale rende questo tipo di rete neurale particolarmente adatta ad un addestramento durante il normale funzionamento del motore. Si propone una completa procedura di design e addestramento della stessa. In particolare vengono fatte alcune considerazione le quali permettono di definire a priori alcuni parametri della rete neurale rendendo il problema di addestramento lineare. Si descrivono due algoritmi di addestramento, il primo veloce ma computazionalmente

dispendioso perciò adatto per un'implementazione offline mentre il secondo idoneo ad un addestramento online. Infine, per concludere l'identificazione parametrica del motore, si propone uno schema basato sull'iniezione di una corrente continua il quale permette di stimare la resistenza di statore indipendentemente da tutti gli altri parametri della macchina. L'indipendenza parametrica permette un notevolmente miglioramento nell'accuratezza di stima del modello magnetico ottenuto con la rete neurale.

La seconda parte di questa tesi, invece, tratta il controllo del motore e come sia possibile migliorarne le performance utilizzando il modello identificato. Innanzitutto, per incrementarne l'efficienza si presenta un innovativo metodo per trovare la curva a massima coppia per corrente. La tecnica proposta lavora in stretta simbiosi con l'identificazione del modello magnetico in quanto è in grado di capire dove si trova la curva cercata rispetto all'attuale punto di lavoro sfruttando la stima locale dei flussi magnetici. Identificata la direzione di movimento, l'azionamento continuamente muove il punto di lavoro coerentemente. Infine, si propongono tre diversi controlli di corrente pensati per gestire un motore fortemente non lineare, tutti basati sul modello stimato. Il primo è un controllore proporzionale–integrale nel quale i parametri vengono modificati al variare del punto di lavoro con lo scopo di mantenere la dinamica della corrente di motore costante. Il secondo è anch'esso basato su un controllore proporzionale–integrale ma a guadagni costanti accoppiato ad un'azione di *feed–forward* la quale compensa tutte le non linearità presenti nella mappa magnetica. Infine, il terzo è un controllo predittivo il quale determina direttamente la posizione degli *switch* tali per cui la funzione di costo è minimizzata. All'interno del controllo, è inserito un vincolo sulla corrente massima e si utilizza un particolare algoritmo per ottenere un lungo orizzonte di predizione.

Tutti i metodi presentati nella tesi sono stata verificati attraverso dettagliate simulazioni e prove sperimentali, eccezione fatta per il controllo predittivo il quale è stato testato attraverso simulazioni.

PUBLICATIONS

[1] 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 (July 2018), pp. 3405–3414. ISSN: 0093-9994. DOI: 10.1109/TIA.2018.2819961.

[2] R. Antonello, L. Ortombina, F. Tinazzi, and M. Zigliotto. "Enhanced Low-Speed Operations for Sensorless Anisotropic PM Synchronous Motor Drives by a Modified Back-EMF Observer." In: *IEEE Transactions on Industrial Electronics* 65.4 (Apr. 2018), pp. 3069–3076. ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2748042.

[3] 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 (Feb. 2018), pp. 1140–1148. ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2733502.

[4] G. A. Longo, L. Ortombina, and M. Zigliotto. "Application of Artificial Neural Network (ANN) for modelling H2O/KCOOH (potassium formate) dynamic viscosity." In: *International Journal of Refrigeration* 86 (2018), pp. 435–440. ISSN: 0140-7007. DOI: 10.1016/j.ijrefrig.2017.11.033.

[5] S. Bolognani, L. Ortombina, F. Tinazzi, and M. Zigliotto. "Model sensitivity of fundamental-frequency-based position estimators for sensorless PM and reluctance synchronous motor drives." In: *IEEE Transactions on Industrial Electronics* 65.1 (Jan. 2018), pp. 77–85. ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2716902.

[6] G. A. Longo, C. Zilio, L. Ortombina, and M. Zigliotto. "Application of Artificial Neural Network (ANN) for modeling oxide-based nanofluids dynamic viscosity." In: *International Communications in Heat and Mass Transfer* 83 (2017), pp. 8–14. ISSN: 0735-1933. DOI: 10.1016/j.icheatmasstransfer.2017.03.003.

[7] 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* (Early Access 2018). ISSN: 2168-6777. DOI: 10.1109/JESTPE.2018.2858842.

[8] L. Ortombina, E. Liegmann, P. Karamanakos, F. Tinazzi, M. Zigliotto, and R. Kennel. "Constrained Long-Horizon Direct Model Predictive Control for Synchronous Reluctance Motor Drives." In: *2018 IEEE 19th Workshop on Control and Modeling for*

Power Electronics (COMPEL). June 2018, pp. 1–8. DOI: 10.1109/COMPEL.2018.8460173.

[9] L. Ortombina, F. Tinazzi, and M. Zigliotto. "Energy-efficient stand-alone solar water-pumping system for synchronous reluctance motor." In: *2017 IEEE 12th International Conference on Power Electronics and Drive Systems (PEDS)*. Dec. 2017, pp. 1, 049–1, 054. DOI: 10.1109/PEDS.2017.8289164.

[10] R. Antonello, L. Ortombina, F. Tinazzi, and M. Zigliotto. "Advanced current control of synchronous reluctance motors." In: *2017 IEEE 12th International Conference on Power Electronics and Drive Systems (PEDS)*. Dec. 2017, pp. 1, 037–1, 042. DOI: 10.1109/PEDS.2017.8289150.

[11] L. Ortombina, F. Tinazzi, and M. Zigliotto. "An effective start-up algorithm for sensorless synchronous reluctance and IPM motor drives." In: *2017 IEEE 12th International Conference on Power Electronics and Drive Systems (PEDS)*. Dec. 2017, pp. 1, 062–1, 067. DOI: 10.1109/PEDS.2017.8289167.

[12] R. Antonello, L. Ortombina, F. Tinazzi, and M. Zigliotto. "Online stator resistance tracking for reluctance and interior permanent magnet synchronous motors." In: *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*. Oct. 2017, pp. 5861–5868. DOI: 10.1109/ECCE.2017.8096970.

[13] L. Ortombina, F. Tinazzi, and M. Zigliotto. "Comprehensive magnetic modelling of internal PM synchronous motors through radial basis function networks." In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. Oct. 2016, pp. 4319–4324. DOI: 10.1109/IECON.2016.7793898.

[14] S. Bolognani, L. Ortombina, F. Tinazzi, and M. Zigliotto. "Model sensitivity assessment for sensorless PM and reluctance motor drives." In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. Oct. 2016, pp. 2851–2856. DOI: 10.1109/IECON.2016.7793841.

Io stimo più il trovar un vero,
benché di cosa leggiera,
ché il disputar lungamente delle massime questioni,
senza conseguir verità nissuna.

Galileo Galilei

Sometimes it is the people who no one imagines anything
of who do the things that no one can imagine.

Alan Turing

ACKNOWLEDGMENTS

For the third time, it is time to write thanks for my dissertation.

First of all, I would like to thank my family for always being close to me, for encouraging and supporting me in my choices. I will always be grateful to my mother and my father for all the goals I have accomplished so far and for those I will achieve soon.

I desire to thank prof. Mauro Zigliotto, who guided me during the last years with wisdom, tireless guidance and patience. A special thank goes to Fabio, my predecessor and now colleague, for his advice, support and endless patience and to Riccardo whom showed me a meticulous research approach. I would like to express my gratitude to all professors who taught me very interesting and useful subjects. Many thanks to all friends I met during my Ph.D. in the Mechatronics Lab with whom I have shared a lot of laughter and damages, in particular Riccardo, Luca, l'idoneo–ma–non–sano Francesco, Andrea e Tommaso. All I can say: *non è come pensi* or *amazing*.

I would like to thank all the friends whom I have shared pleasant time outside the lab. I cannot forget all the friends with whom I organised the camps. I spent some of the funniest and edifying moments of my life with them. Finally, a thank to Soraya, another *Valpolicellese* brought over in Vicenza, with whom I joked and spent some enjoyable free time.

This research has been supported by
FSU (Fondazioni Studi Universitari) of Vicenza.
I would like to express my sincere gratitude to the Foundation
for having sponsored my Ph.D. throughout these three years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

3L-NPC three–level neural point clamped

AC alternating current

BEMF back electromotive force

DC direct current

DSP digital signal processor

FCS finite control set

FEA finite element analysis

FF feed–forward

FPGA field programmable gate array

FW flux weaking

GNA Gauss–Newton algorithm

IGBT insulated gate bipolar transistor

ILS integer least–squares

IM induction motor

IPM interior permanent magnet (PM)

LMA Levenberg–Marquardt algorithm

LQR linear–quadratic regulator

LUT look-up table

MIMO multiple–input multiple–output

MMF magnetomotive force

MPC model predictive control

MRAS model reference adaptive system

MTPA maximum torque per ampere

MTPV maximum torque per volt

MUT motor under test

NN neural network

PI proportional–integral

PM permanent magnet

PMASRM permanent magnet assisted synchronous reluctance motor

PMSM permanent magnet synchronous motor

PWM pulse width modulation

Nd Neodymium

Pr Praseodymium

RBF radial basis function

RLS recursive least square

SISO single-input single–output

SDA Sphere Decoding algorithm

SynR synchronous reluctance

THD total harmonic distortion

VSI voltage source inverter

SVM space vector modulation

NOMENCLATURE

$\mathbf{A}^k$	Time–varying state matrix
a^k	Output of the k–th Gaussian function
B	Viscous friction coefficient of the system
β^*	Current angle in MTPA tracking procedure
b^k	Parameter related to the width of the k–th Gaussian function
$\mathbf{B}^k, \widetilde{\mathbf{B}}$	Input matrices
$\mathbf{C}^k$	Output matrix
$\delta\mathbf{i}_{\alpha\beta}$	Actually injected current in $\alpha\beta$ reference frame
$\delta\mathbf{i}^*_{\alpha\beta}$	Injected current reference in $\alpha\beta$ reference frame
$\delta\mathbf{i}_{dq}$	Actually injected current in dq reference frame
$\delta\mathbf{i}^*_{dq}$	Injected current reference in dq reference frame
e_a, e_b, e_c	Back electromotive force of each phase
E_d, E_q	Quadratic cost functions for neural network training
$\boldsymbol{\epsilon}_d, \boldsymbol{\epsilon}_q$	Estimation errors
F_s	PWM sampling frequency
$\mathbf{G}_{\alpha\beta}$	Base–band dynamical response of a modulated system
γ_d, γ_q	Coefficients that linearise the MTPA curve
$\mathbf{G}_{dq}$	Transfer matrix between $\delta\mathbf{i}^*_{dq}$ and $\delta\mathbf{i}_{dq}$
i_a, i_b, i_c	Phase currents
i^{bnd}	Stator current constraint
i_d, i_q	Motor currents in dq reference frame
J	Moment of inertia of the system
$\mathbf{J}_d, \mathbf{J}_q$	Jacobian matrix of the Levenberg–Marquardt algorithm
J^k	Model predictive cost function
k_i	Integral coefficient of PI regulator

k_p	Proportinal coefficient of PI regulator
L_a, L_b, L_c	Auto–inductances of each motor phase
$\lambda_a, \lambda_b, \lambda_c$	Flux linkages of each phase
λ_d, λ_q	Flux linkages in dq reference frame
Λ_{mg}	Permanent magnet flux linkage
L_d, L_q	dq axis apparent inductances
L_d^{diff}	d–axis differential inductance
L_{dq}^{diff}	dq cross–differential inductance
L_{qd}^{diff}	qd cross–differential inductance
L_q^{diff}	q–axis differential inductance
M_{ab}, M_{bc}, M_{ca}	Mutual inductances
n^k	Exponent of the k–th Gaussian function
ω_m	Mechanical angular speed
ω_{me}	Electromechanical angular speed
p	Pole pairs
$\mathbf{Q}$	Weighting matrix in model predictive control
ρ^k	Initial radius of the hypersphere
R	Stator resistance
s_a, s_b, s_c	Single–phase switches position
$\mathbf{T}_{abc/\alpha\beta o}$	Matrix transformation from abc to $\alpha\beta$o reference frame
$\mathbf{T}_{\alpha\beta/dq}$	Matrix transormation from $\alpha\beta$ to dq reference frame
τ_L	Load torque
τ	Electromagnetic torque
T_d	Overall control delay
ϑ_m	Mechanical position
ϑ_{me}	Electromechanical position
T_s	PWM sampling time
u_a, u_b, u_c	Phase line voltages
u_d, u_q	Motor voltages in dq reference frame

u_d^{FF}, u_q^{FF}	Feed–forward terms in dq reference frame
ε^N	Normalised error
V_{dc}	DC–link voltage
$\mathbf{J}$	Matrix representation of complex numbers
w_d^k	Second layer weigth related to the k–th Gaussian function which computes the d–flux linkage estimate
$\hat{\cdot}$	Estimated quantity of $\cdot$ variable
w_q^k	Second layer weigth related to the k–th Gaussian function which computes the q–flux linkage estimate
ξ	Generic dummy variable
ξ_d, ξ_q	Coefficients that linearise the torque equation
$\mathbf{x}^k$	Centre of the k–th Gaussian function in the dq plane
$\mathbf{x}_{dq}^k, \mathbf{u}_{dq}^k$	State and input vector
$\mathbf{y}^k$	Output vector
i_α, i_β	Motor currents in $\alpha\beta$ reference frame
K	Number of Gaussian functions
M	Total number of steady–state training point
u_α, u_β	Motor voltages in $\alpha\beta$ reference frame

1 INTRODUCTION

European Union is increasingly influencing individual states in different socio–economic spheres. It defines the guidelines that should be followed through the issuing of Community directives, e. g. the Ecodesign Directive 2009/125/EN, and the definition of projects, e. g. *Horizon 2020*. It is the biggest EU Research and Innovation programme ever with nearly €80 billion of funding available over 7 years (2014 to 2020). It aims at a smart, sustainable and inclusive growth with greater coordination of National and European policy. A branch of this enormous programme is about the city of tomorrow and the project is usually stated as *smart cities and communities*. Europe conjugates smart cities and communities in different fields and areas of interests and some of them are:

- smart health,
- smart education,
- cloud computing technologies for smart government,
- smart culture and tourism,
- renewable energy and smart grid,
- energy efficiency and low carbon technologies,
- smart mobility and last–mile logistic,
- sustainable natural resources (waste, water, urban biodiversity, etc).

Some of these branches are deeply merged with the electric motor field since mobility, energy efficiency, low carbon technologies are some of the main research threads.

Efficiency and performance depict the essential and unavoidable targets in up-to-date projects. Simultaneously, they symbolise two sides of the same coin, indeed engineers have usually to find the best trade-off among these two requirements. In the last years, the emphasis on the efficiency has been due to many factors, in particular the increasing energy costs, a growing environmental consciousness and the national and European regulations, both in industrial and civil society. The greater efficiency demanded by the customers has to be achieved without sacrificing the performances of the final product. The businessman wants the machinery with the highest productivity and the private citizen wants a car with good performances.

Both requirements can be achieved using permanent magnet synchronous motor (PMSM). They have been adopted in applications characterised by high dynamic demand since the 1980s when rare–earth magnetic materials and low cost electronics devices were available. The main strengths of PMSM are the high torque and power density, the high efficiency and the linearity of the torque respect to the current, which simplifies the control. Their relatively high production cost is one of the reason that limited their diffusion. Nevertheless, nowadays their strengths are overcoming their limitations and they are gaining new attentions. In particular, car manufacturers are choosing PMSM for their electric and hybrid cars. In 2017, *Tesla* announced to have adopted for their *Model 3 Long Range* car a 192 kW PMSM instead of their patented induction motor (IM) [1]. In order to get an idea, the production of the 500000 already ordered Model 3 will consume minimum 600 tonnes incrementing new demand of Neodymium (Nd)/Praseodymium (Pr) oxide per year. This is equivalent to 2 % of the world's global legal annual produced Nd/Pr oxide. In addition, according to *Bloomberg*, more than 120 additional new electric cars are in the pipe to get launched during the next 2,5 years [2].

The growing role of rare earth materials in a low carbon future is described in [3]. They are playing an important match under a technological point of view and political as well, because the reserves are located only in few states. In particular, China accounts for over 80 % of the world's production of rare earths. The reason is the immense impact on the environment and on the human health due to the mining operations. Therefore, having control of these elements pits China at a powerful position. In the fall of 2010, China showed his power in this field. China temporarily cut off supplies of rare earths to the Japan in retaliation for a maritime incident. Additionally, China reduced exports by a 40 % and it would further reduced exports by another 30 % the following year. The growing demand of rare earth materials, both to build the electric motor and the batteries, and the tightening on exports fell back on the price in a remarkable way. In some months, the prices skyrocketed over factor 20 times. After hitting their peak, prices of the materials dropped almost as rapidly and have settled to a reasonable price [4]. Without international crisis as in 2010, the analysts expect a generalised growing of prices in next years. In particular, the forecast about Nd's price shows an increment of about 50 % in the next 8 years [5]. A possible shuffling of the cards in play comes from Japan, where an enormous rare earth vein, containing 16 million tons of valuable metals, was discovered in 2012. The main issue is that the mineral are buried 5000 m deep the Pacific ocean, near the Minami Torishima island. Only in April 2018, the scientists published on the journal *Scientific Reports* an efficient method to mine the minerals [6].

Figure 1.1: SynR motor with a view of the cross–section rotor.

The exorbitant price reached by rare earth and high dependence from one state had an important effect on PM motor manufacturers. Another paramount element that has only been mentioned so far is the huge pollution created during the mining operation. The production of one ton of rare earth metal typically produces 2000 tons of toxic waste, as the ore is usually laced with radioactive material [7]. In order to reach a more cleaned and sustainable world, the environmental pollution created during all life's products have to be taken into account. It is essential to find a right equilibrium among price, environmental impact, efficiency and performance.

Based of what has been said, there has been a marked research activities to investigate the viability to develop alternative kinds of motor with comparable performances but less needy of strong permanent magnets. A viable solution can be identified in the SynR motors. They have been proposed several years ago, but only in the last years they are becoming more and more attractive. The reasons are the robustness, the overload capability, the wide operating range and their low cost. Furthermore, the absence of rotor currents leads SynR motors to a higher efficiency than IMs. The standard IEC $60034-30-1$ issued by the International Electrotechnical Commission in March 2014 defined four efficiency classes for electric motors designed for operation on sinusoidal voltage, i. e. directly connected to the main. The IE4 Super–Premium efficiency class can be easily achieved by SynR motors while some expedients have to be taken into account for IMs.

SynR motors do not use PM then torque is generated accordingly to the reluctance principle only. The design of SynR motor is challenging because high nominal torque and efficiency are achievable only with an accurate design. Usually, several flux–barriers per pole

are required. The essential anisotropic rotor to exploit the reluctance torque generation principle yields these motors extremely suitable for sensorless technique based on injection. As every synchronous motors, the rotor position is demanded for an appropriate control but the position sensor is generally an issues because it is expensive and it reduces the reliability of the whole system. The low speed sensorless aptitude of SynR motor is an important advantage respect to PMSM.

SynR motors are characterised by some limitations, e. g. high torque ripple and a low power factor. In order to solve these restrictions and to improve the performances, some magnets can be put inside the flux barriers. The obtained motor is called permanent magnet assisted synchronous reluctance motor (PMASRM). Generally, when the PM flux tends to be the dominant component of the machine flux, these motors are referred to as interior PM (IPM) motor. To preserve the main advantage of the SynR motor, the added PM is minimum. Moreover, in order to be independent from rare earth material, Ferrite magnets are usually adopted. The remanent flux density is typically of 0.4 T, less than half the corresponding value of NdFeB permanent magnets. Nevertheless, they are widely used in PMASRM due to the small flux required.

From the perspective of machine design, SynR motor and PMASRM are a remarkable challenge but they are a not trivial system to be controlled as well. These motors are prone to magnetic saturation, therefore the current control becomes nonlinear. Furthermore, torque nonlinearly depends on the stator current then torque and/or speed control becomes nonlinear. The magnetic model has to be identified accurately in order to develop and to employ suitable nonlinear control strategies. In addition, a precise knowledge of the magnetic model allows to exploit all the intrinsic features of these motors. The reluctance torque production lets an additional degree–of–freedom in the choice of the stator currents. The same torque level can be generated with different stator currents then the MTPA trajectory is usually implemented. It is an inherent feature of each single motor since it relies on the motor parameters. Moreover, the reduced amount or the complete absence of PM flux allows the motor to operate at a speed higher than the base one. To exploit this possibility, flux weaking (FW) and maximum torque per volt (MTPV) control strategies can be employed.

Notwithstanding the worldwide awareness about environmental issues supported by many high–profile projects, there are still people who label these issues as fake news. They are a small percentage of the world's population, although in some cases they have great responsibilities. Academics and researchers need to face new challenges with the aim of increasing wealth, reducing the environmental footprint and boosting the knowledge. They can follow some marked paths highlighted by European Union, international organisations, top–level academics. This dissertation wants to give its humble contribute into

the smart cities and communities context, hoping to provide interesting insights into future research.

1.1 INVESTIGATED ASPECTS AND CONTRIBUTION

Within the scenario outlined above, the dissertation investigates some critical aspects related to the identification and the control of nonlinear motors. The challenges on which the thesis is focused on and the proposed solutions are summarised in the following subsections.

1.1.1 *Model identification*

Challenge. High–end electric drives have to be able to identify the whole model of the connected motor. In particular, the magnetic model has to include nonlinearity and cross–coupling with a good accuracy. The full exploitation of the motor features depends on the knowledge of the model, for instance in sensorless, in current and in maximum efficiency controls. Furthermore, the capability of the electric drive to observe and track parameters variation plays a crucial rule for adaptive controller and for fault monitoring as well.
Contribution. A method for estimating the stator resistance independently from the other motor parameters was proposed. It is based on direct current injection in the stationary reference frame by exploiting the current controller in the rotating reference frame. The study was completed by a method to design the additional regulator requested by the injection. The identification stage was completed with a special kind of NN able to identify the whole magnetic model of a synchronous motor. Some rules were developed to define some NN parameters. The structure of the chosen NN is described. Two different training algorithms were proposed, the former one is based on a classical algorithm used in NN training while the latter one is suitable for an online application. Both the stator resistance and the magnetic maps can be estimated during the normal motor operation. Finally, experimental tests were carried to prove the effectiveness of whole identification stage.

1.1.2 *Motor control*

Challenge. Reluctance–based motors are characterised by a nonlinear magnetic model which poses new challenges compared to the PMSM. The current control has to handle the magnetic model properly, in order to guarantee the demanded dynamics and the stability of the controlled system. Furthermore, both stator currents generate electromagnetic torque. An additional degree–of–freedom characterises the control. In order to increase the system efficiency, it should be used to drive the system towards minimising copper losses.

Contribution. By exploiting the continuity of the NN–based model and the training algorithm able to continuously update the NN, a new strategy to find the MTPA was proposed. It is not based on any signal injection but just on the observation of the local estimated flux linkages. Furthermore, two different current control schemes were presented. They are tailored for handling the nonlinear model and the cross–coupling. The two regulators are easy to implement and they exhibit constant dynamic behaviour over the entire input current region. The former is mainly based on gain–scheduling PI controller while the latter has a FF action to compensate the nonlinearity in the magnetic model. Both the MTPA tracking algorithm and the controls were experimentally tested.

1.1.3 *Novel model predictive current control*

Challenge. Usually, nonlinear motors are controlled by linear PI regulators, due to their inherent simplicity. They can achieve good performances but linear regulators cannot exploit all the system characteristics. Furthermore, it is worth remembering that an electric drive is composed by switches which have a non–continuous behaviour. Usually, the averaging is used to conceal the switching behaviour but it is not always effective in particular for low switching frequencies. For those conditions, the switching behaviour should be addressed by the regulator properly.
Contribution. Direct MPC was proposed as a current control for SynR motors. It can manage quite easily the nonlinear magnetic model of SynR motor and it computes the optimal switching pattern that the electric drive has to apply. The main contributions is the range extension of the prediction horizon. A long horizon was achieved since the performances of the system can be enhanced respect to more common control paradigms, specially when the switching frequency is low as in high power application. Furthermore, hard current constraint was added to guarantee a stator current smaller than a predetermined threshold, since the current ripple is not negligible for low switching frequencies. The control performances were verified through several simulations.

1.2 DISSERTATION OUTLINE

The dissertation consists of nine chapters. The current chapter, Chapter 1, introduces the considered research topic, highlighting the main aspects on which the research activity reported herein is focused. Chapter 2 derives the motor model using a physical approach, starting from the windings distributions into the slots to get the voltage equation in the rotating reference frame. The central part of the thesis can be subdivided in two parts. The first one is composed by Chapters 3, 4, 5

where the methods to identify the model of a synchronous motor are depicted. In particular, Chapter 3 describes how to identify the stator resistance independently from all the other motor parameters during normal motor operations. The design for the injection regulator is also derived. An innovative technique to estimate the magnetic model based on a special kind of NN is described in Chapter 4. The selected NN is fully illustrated along with the proposed training algorithm, revealing pros and cons. Chapter 5 introduces a new training algorithm specially developed for the implementation in electric drives during the normal motor operation likewise the resistance identification technique. Chapters 6, 7, 8 deal with the control of a synchronous motor. All control paradigms are based on the accurate model developed in the first part of the dissertation. In detail, a method for minimising the copper losses, i. e. it finds the maximum torque–per–ampere curve, is described in Chapter 6. It is deeply merged with the magnetic model identification and it relies on the continuity of the NN model. Then, the current control of the converter is presented. Chapter 7 depicts two current control schemes able to handle the nonlinear magnetic model properly. They are PI regulators specially modified to attain a stable system which exhibit constant performance dynamics over the whole input current region. A more sophisticated current control, based on long–horizon model predictive algorithm is introduced in Chapter 8. All the proposed techniques, both for the identification and for the control, are experimentally verified with the exception of the last one where only simulations results are carried out. Finally, Chapter 9 reports the conclusions drawn from the research work.

Part I

PARAMETERS IDENTIFICATION

2 SYNCHRONOUS RELUCTANCE MOTORS MODEL

In control theory, the model of the controlled plant plays a key role. The model describes the behaviour of the system, relating the input signals with the output ones without feedback. It is commonly determined by physical properties of the system. An electric motor is composed by two subsystems: the electrical system and the mechanical one. In this chapter, the derivation of both models for a SynR motor is carried out [8]. It will be used throughout all this thesis.

2.1 ELECTRICAL MODEL

The mathematical derivation of the SynR motor model is quite challenging because it is a nonlinear system. The model of synchronous motor without saturation, i. e. a linear motor, is first derived and then some consideration will be carried out to obtain the complete model. Furthermore, the study of the linear case allows to deeply understand the fundamental behaviour of the machine, that it would be extremely burdensome for the nonlinear case. The time dependency will be omitted throughout the thesis to simplify the notation, except where demanded for clarity.

2.1.1 *Windings and magnetomotive force*

The mechanical structure of a synchronous motor is composed by a stator, where the windings are housed, and a rotor. Different synchronous motors are characterised by different rotors while the stator is generally the same for all kind of machines. The windings are embedded in slots around the inside circumference of the stator and each phase winding of a 3–phase winding is displaced 120° (electromechanical degree) with respect to each other. Generally, the coils of each phase are distributed and, if necessary, in the same slot there could be coils of different windings. In some cases, the coils may not be distributed uniformly over the part of circumference that they occupy. The described winding arrangement produces an air–gap magnetomotive force (MMF) which more closely approximates a sinusoidal air–gap

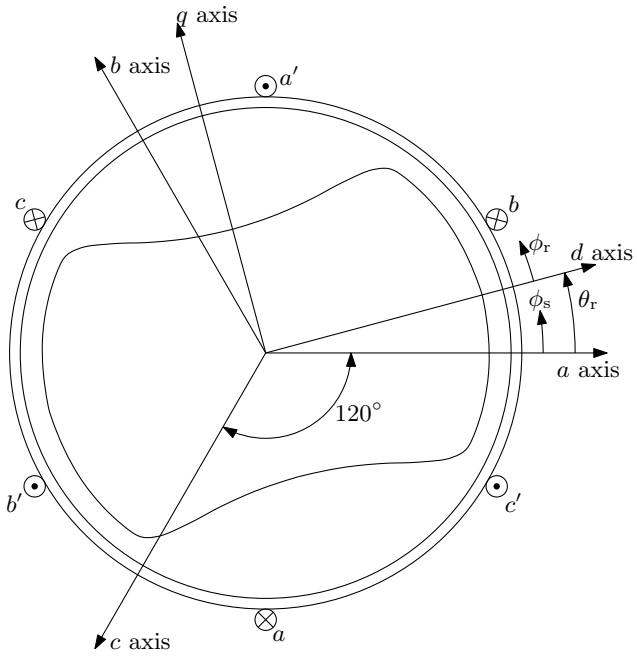


Figure 2.1: Elementary 2–pole, 3–phase synchronous machine.

MMF with respect to the angular position[1]. For one pole pair motor, the winding distribution N_a for the first phase a can be written as:

$$N_a = \frac{N_s}{2} |\sin(\phi_s)|, \qquad 0 < \phi_s \leqslant 2\pi \tag{2.1}$$

where N_s represents the number of turns of the equivalent sinusoidally distributed winding and the ϕ_s is the angular displacement along the stator circumference respect to the axis of the phase a[2]. The other windings are shifted by $2/3\pi$ and $4/3\pi$ radians. Finally, all the windings are identical in that each winding has the same resistance and the same number of turns. When a machine has three identical stator windings is often referred to as a symmetrical stator windings. A two poles machine is depicted in Figure 2.1.

Due to the high permeability of the stator and the rotor steel, the magnetic fields essentially exist only in the air–gap and tend to have radial direction due to the short length of the air–gap relative to the inside stator diameter. Applying the Ampere's law, the magnetomotive force of the first phase MMF_a is:

$$MMF_a = \frac{N_s}{2} i_a \cos(\phi_s) \tag{2.2}$$

1 A different winding arrangement is possible. Since the late of 1990s, researchers started to focus on concentrated windings, in which every stator winding encircles a single stator tooth. This kind of arrangement is frequently combined with a fractional–slot winding configuration in which the number of slots–per–phase–per–pole is a fraction that can be expressed as a non integer number [4, 9].

2 N_s is an equivalent number of turns of a sinusoidally distributed winding which would give rise to the same fundamental component as the actual winding distribution. N_s is not the total number of turns of the winding.

and the MMF of the other phases result:

$$\begin{aligned} MMF_b &= \frac{N_s}{2} i_b \cos\left(\phi_s - \frac{2\pi}{3}\right) \\ MMF_c &= \frac{N_s}{2} i_c \cos\left(\phi_s - \frac{4\pi}{3}\right). \end{aligned} \tag{2.3}$$

The total air–gap MMF produced by the stator currents can be written as:

$$\begin{aligned} MMF = \frac{N_s}{2}\Bigg(i_a \cos(\phi_s) + i_b \cos\left(\phi_s - \frac{2\pi}{3}\right) + \\ + i_c \cos\left(\phi_s - \frac{4\pi}{3}\right)\Bigg). \end{aligned} \tag{2.4}$$

With a balanced and symmetric stator currents:

$$\begin{aligned} i_a &= I\cos(\omega_{me} t) \\ i_b &= I\cos\left(\omega_{me} t - \frac{2\pi}{3}\right) \\ i_c &= I\cos\left(\omega_{me} t - \frac{4\pi}{3}\right) \end{aligned} \tag{2.5}$$

where I is the amplitude of each stator current. Substituting (2.5) into (2.4) and applying trigonometric functions, the total MMF becomes:

$$MMF = \frac{N_s}{2} I \frac{3}{2} \cos(\omega_{me} t - \phi_s). \tag{2.6}$$

The windings of a motor with p pole pairs consists of p series connected windings which are considered as sinusoidally distributed windings. The air–gap MMF generated by each phase can be written as:

$$\begin{aligned} MMF_a &= \frac{N_s}{2p} i_a \cos(p\phi_s) \\ MMF_b &= \frac{N_s}{2p} i_b \cos\left(p\phi_s - \frac{2\pi}{3}\right) \\ MMF_c &= \frac{N_s}{2p} i_c \cos\left(p\phi_s - \frac{4\pi}{3}\right) \end{aligned} \tag{2.7}$$

and the total MMF as:

$$MMF = \frac{N_s}{2p} I \frac{3}{2} \cos(\omega_{me} t - p\phi_s). \tag{2.8}$$

2.1.2 *Motor inductances*

In a magnetic linear system the self–inductance of a winding is the ratio between the flux linked by the winding and the current flowing in the winding with all other winding currents equal to zero. Mutual

inductance is the ratio between the flux linked by one winding due to the current flowing in a second winding with all other currents equal to zero. Let ϕ_r is the angular displacement along the rotor circumference and θ_r is the rotor position, which is also called electromechanical position ϑ_{me}. For a given angular position relative to the a–axis, the following relation holds:

$$\phi_s = \phi_r + \theta_r. \tag{2.9}$$

In order to compute the self– and mutual– inductances of a one pole pair synchronous motor, it is assumed that the air–gap length g can be approximated as:

$$g(\phi_r) = \frac{1}{\alpha_1 - \alpha_2 \cos(2\phi_r)} \tag{2.10}$$

or

$$g(\phi_s - \theta_r) = \frac{1}{\alpha_1 - \alpha_2 \cos(2(\phi_s - \theta_r))} \tag{2.11}$$

where α_1 and α_2 depend on the rotor geometries and the minimum air–gap length is $(\alpha_1 + \alpha_2)^{-1}$ and the maximum is $(\alpha_1 - \alpha_2)^{-1}$. The air–gap flux density B_r with all currents equal to zero excepts i_a is:

$$\begin{aligned} B_r(\phi_s, \theta_r) &= \mu_0 \frac{MMF_a(\phi_s)}{g(\phi_s - \theta_r)} \\ &= \mu_0 \frac{N_s}{2} i_a \cos(\phi_s) (\alpha_1 - \alpha_2 \cos(2(\phi_s - \theta_r))) \end{aligned} \tag{2.12}$$

where μ_0 is the air permeability. Similarly, the flux density with all currents zero except i_b is:

$$B_r(\phi_s, \theta_r) = \mu_0 \frac{N_s}{2} i_b \cos\left(\phi_s - \frac{2\pi}{3}\right) (\alpha_1 - \alpha_2 \cos(2(\phi_s - \theta_r))) \tag{2.13}$$

and with all current zero except i_c is:

$$B_r(\phi_s, \theta_r) = \mu_0 \frac{N_s}{2} i_c \cos\left(\phi_s - \frac{4\pi}{3}\right) (\alpha_1 - \alpha_2 \cos(2(\phi_s - \theta_r))). \tag{2.14}$$

In order to determine the self–inductance is necessary to compute the single winding flux linkage due to its own current. Let consider the flux linkage of a single turn of a stator winding which spans π radians and located at an angle ϕ_s, the flux can be determined by performing a surface integral over the open surface of the single turn, namely:

$$\Phi(\phi_s, \theta_r) = \int_{\phi_s}^{\phi_s + \pi} B_r(\xi, \theta_r) r l \, d\xi \tag{2.15}$$

where Φ is the flux linking a single turn oriented ϕ_s from the a–axis, l is the axial length of the air–gap of the motor, r is the mean radius

of the air–gap and ξ is dummy variable of integration. To obtain the flux linkages of whole winding, the flux linked by each turn must be summed. Considering the windings sinusoidally distributed and the magnetic material to be linear, this summations can be accomplished by integrating over all coil sides carrying current in the same direction. The total flux linkages of the a–winding due to current flowing in the same winding corresponds to:

$$\begin{aligned}\lambda_a =& L_\sigma i_a + \int N_a(\phi_s)\Phi(\phi_s,\theta_r)d\phi_s = \\ & L_\sigma i_a + \int N_a(\phi_s)\int_{\phi_s}^{\phi_s+\pi} B_r(\xi,\theta_r) r l \, d\xi \, d\phi_s\end{aligned} \tag{2.16}$$

where L_σ is the leakage inductance related to the stator flux that does not link the rotor. Substituting (2.1) and (2.12) into (2.16) yields:

$$\begin{aligned}\lambda_a = L_\sigma i_a - \int_\pi^{2\pi} \frac{N_s}{2}\sin\phi_s \int_{\phi_s}^{\phi_s+\pi} \mu_0 \frac{N_s}{2} i_a \cos\xi \\ (\alpha_1 - \alpha_2\cos(2(\xi-\theta_r)))\, r l \, d\xi \, d\phi_s = \\ L_\sigma i_a + \left(\frac{N_s}{2}\right)^2 \pi\mu_0 r l \left(\alpha_1 - \frac{\alpha_2}{2}\cos(2\theta_r)\right) i_a\end{aligned} \tag{2.17}$$

where the interval of integration is taken from π to 2π so as to comply with the convention that positive flux linkages are obtained in the direction of the positive a–axis by circulation of the assumed positive current in the clockwise direction about the coil (right–hand rule). The self–inductance of the a–winding is obtained by dividing (2.17) by i_a, thus:

$$L_a = L_\sigma + \left(\frac{N_s}{2}\right)^2 \pi\mu_0 r l \left(\alpha_1 - \frac{\alpha_2}{2}\cos(2\theta_r)\right). \tag{2.18}$$

In order to determine the mutual inductance it is necessary to compute the flux linking one winding due to the current flowing in another winding. The mutual inductance between the a– and b–windings can be determined by computing the flux linking the a–winding due to the current flowing only in the b–winding, thus:

$$\lambda_a = \int N_a(\phi_s)\int_{\phi_s}^{\phi_s+\pi} B_r(\xi,\theta_r) r l \, d\xi \, d\phi_s \tag{2.19}$$

where the magnetic coupling that might occur at the end turns of the windings is neglected. Substituting (2.1) and (2.13) in (2.19) yields:

$$\begin{aligned}\lambda_a = -\int_\pi^{2\pi} \frac{N_s}{2}\sin\phi_s \int_{\phi_s}^{\phi_s+\pi} \mu_0 \frac{N_s}{2} i_b \cos\left(\xi - \frac{2\pi}{3}\right) \\ (\alpha_1 - \alpha_2\cos(2(\xi-\theta_r)))\, r l \, d\xi \, d\phi_s.\end{aligned} \tag{2.20}$$

The mutual inductance between the a– and b–windings is obtained by dividing (2.20) by i_b thus:

$$M_{ab} = -\left(\frac{N_s}{2}\right)^2 \frac{\pi}{2}\mu_0 r l \left(\alpha_1 + \alpha_2\cos\left(2\left(\theta_r - \frac{\pi}{3}\right)\right)\right). \tag{2.21}$$

The remaining self– and mutual–inductances can be calculated using the same procedure.

The motor self–inductances can be expressed as:

$$\begin{aligned} L_a &= L_\sigma + L_0 - L_2 \cos(2\theta_r) \\ L_b &= L_\sigma + L_0 - L_2 \cos\left(2\left(\theta_r - \frac{2\pi}{3}\right)\right) \\ L_c &= L_\sigma + L_0 - L_2 \cos\left(2\left(\theta_r - \frac{4\pi}{3}\right)\right) \end{aligned} \tag{2.22}$$

while the mutual inductances can be expressed as:

$$\begin{aligned} M_{ab} &= -\frac{1}{2}L_0 - L_2 \cos\left(2\left(\theta_r - \frac{\pi}{3}\right)\right) \\ M_{bc} &= -\frac{1}{2}L_0 - L_2 \cos(2(\theta_r + \pi)) \\ M_{ca} &= -\frac{1}{2}L_0 - L_2 \cos\left(2\left(\theta_r + \frac{\pi}{3}\right)\right) \end{aligned} \tag{2.23}$$

where the inductances L_0 and L_2 are defined as follow to obtain a more compact notation:

$$\begin{aligned} L_0 &= \left(\frac{N_s}{2}\right)^2 \pi \mu_0 r l \alpha_1 \\ L_2 &= \frac{1}{2}\left(\frac{N_s}{2}\right)^2 \pi \mu_0 r l \alpha_2. \end{aligned} \tag{2.24}$$

In order to generalise the motor inductances calculated for a 2–pole machine to a p pole pairs motor, it is sufficient to substitute in (2.22) and (2.23) the rotor position θ_r with the electromechanical position ϑ_{me}/p and scale L_0 and L_2 by a factor $1/p^3$ [8]. The mathematical proof is reported in Appendix B.

Finally, isotropic uniform air–gap motors, e. g. PMSM, are characterised by constant inductances then the $2\theta_r$ variation is not present. The winding inductance my be determined from the above relationship by simply setting $\alpha_2 = 0$ in (2.24).

2.1.3 *Voltage balance*

The motor voltage equations in the stationary reference frame [10, 11] can be written as follow:

$$\begin{aligned}
u_a &= Ri_a + \frac{d\lambda_a}{dt} = Ri_a + L_a\frac{di_a}{dt} + M_{ab}\frac{di_b}{dt} + M_{ca}\frac{di_c}{dt} + \\
&\qquad \frac{dL_a}{dt}i_a + \frac{dM_{ab}}{dt}i_b + \frac{dM_{ca}}{dt}i_c + e_a \\
u_b &= Ri_b + \frac{d\lambda_b}{dt} = Ri_b + L_b\frac{di_b}{dt} + M_{ab}\frac{di_a}{dt} + M_{bc}\frac{di_c}{dt} + \\
&\qquad \frac{dM_{ab}}{dt}i_a + \frac{dL_b}{dt}i_b + \frac{dM_{bc}}{dt}i_c + e_b \\
u_c &= Ri_c + \frac{d\lambda_c}{dt} = Ri_c + L_c\frac{di_c}{dt} + M_{ca}\frac{di_a}{dt} + M_{bc}\frac{di_b}{dt} + \\
&\qquad \frac{dM_{ca}}{dt}i_a + \frac{dM_{bc}}{dt}i_b + \frac{dL_c}{dt}i_c + e_c
\end{aligned} \tag{2.25}$$

where the stator resistance R is considered identical for each phase. The back electromotive force (BEMF) e_a, e_b and e_c are defined as follow:

$$\begin{aligned}
e_a &= \frac{d\boldsymbol{\lambda}_{mg,a}}{dt} = -\Lambda_{mg}\omega_{me}\sin(\vartheta_{me}) = \\
&\qquad = -\Lambda_{mg}\omega_{me}\cos\left(\vartheta_{me} - \frac{\pi}{2}\right) \\
e_b &= \frac{d\boldsymbol{\lambda}_{mg,b}}{dt} = -\Lambda_{mg}\omega_{me}\cos\left(\vartheta_{me} - \frac{\pi}{2} - \frac{2\pi}{3}\right) \\
e_c &= \frac{d\boldsymbol{\lambda}_{mg,c}}{dt} = -\Lambda_{mg}\omega_{me}\cos\left(\vartheta_{me} - \frac{\pi}{2} - \frac{4\pi}{3}\right)
\end{aligned} \tag{2.26}$$

where $\boldsymbol{\lambda}_{mg,a}$, $\boldsymbol{\lambda}_{mg,b}$ and $\boldsymbol{\lambda}_{mg,c}$ are the flux linkages due to the PM along the three stationary axes. The voltages equation can be written in a matrix form as follow:

$$\mathbf{u}_{abc} = \mathbf{R}\mathbf{i}_{abc} + \frac{d\boldsymbol{\lambda}_{abc}}{dt} = \mathbf{R}\mathbf{i}_{abc} + \frac{d\mathbf{L}_{abc}\mathbf{i}_{abc}}{dt} + \mathbf{e}_{abc} \tag{2.27}$$

where:

$$\mathbf{u}_{abc} = \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}, \quad \mathbf{i}_{abc} = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}, \quad \boldsymbol{\lambda}_{abc} = \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix}, \quad \mathbf{e}_{abc} = \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \tag{2.28}$$

and $\mathbf{R}$ and $\mathbf{L}_{abc}$ are the resistance and inductance matrices, respectively:

$$\mathbf{R} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix}, \tag{2.29a}$$

$$\mathbf{L}_{abc} = \begin{bmatrix} L_a & M_{ab} & M_{ca} \\ M_{ab} & L_b & M_{bc} \\ M_{ca} & M_{bc} & L_c \end{bmatrix} \tag{2.29b}$$

In a three–phase system, the $\alpha\beta o$[3] reference frame is often used, for its compactness representation and properties even in case of unbalanced systems [12]. The matrix transformation $\mathbf{T}_{abc/\alpha\beta o}$ is defined as follows:

$$\mathbf{T}_{abc/\alpha\beta o} = \frac{2}{3}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \tag{2.30}$$

In compact notation, it stands:

$$\boldsymbol{\xi}_{\alpha\beta o} = \mathbf{T}_{abc/\alpha\beta o}\boldsymbol{\xi}_{abc}. \tag{2.31}$$

It is worth highlighting that the transformation (2.31) ensures that the amplitudes of the (balanced) three–phase signals are preserved but it is not power invariant. By applying (2.31) on (2.27), the voltages equation in the $\alpha\beta o$ reference frame is:

$$\mathbf{u}_{\alpha\beta o} = \mathbf{R}\mathbf{i}_{\alpha\beta o} + \frac{d\mathbf{L}_{\alpha\beta o}\mathbf{i}_{\alpha\beta o}}{dt} + \mathbf{e}_{\alpha\beta o} \tag{2.32}$$

where $\mathbf{L}_{\alpha\beta o} = \mathbf{T}_{abc/\alpha\beta o}\mathbf{L}_{abc}\mathbf{T}^{-1}_{abc/\alpha\beta o}$[4].

Generally, electric motors can be considered as a balanced three–phase system. The homopolar component ξ_0 can be considered null in both currents and voltages due to the balanced BEMF and the absence of the neutral wire. However, the homopolar component has no effect on the torque production. Therefore, (2.32) can be simplified by neglecting the third equation labelled with "o". Equation (2.32) without the homopolar component can be written as:

$$\mathbf{u}_{\alpha\beta} = \mathbf{R}\mathbf{i}_{\alpha\beta} + \frac{d\mathbf{L}_{\alpha\beta}\mathbf{i}_{\alpha\beta}}{dt} + \mathbf{e}_{\alpha\beta} \tag{2.33}$$

where all quantities have a suitable dimension. The resistance matrix is still equal to $\mathbf{R} = R\mathbf{I}$ but the identity matrix becomes $\mathbf{I} \in \mathbb{R}^{2\times 2}$ respect to the one defined in (2.29a). Throughout the thesis, the same symbol $\mathbf{R}$ will be used to indicate both $\mathbb{R}^{2\times 2}$ and $\mathbb{R}^{3\times 3}$ (see (2.29a)) resistance matrices.

Consequently, the Clarke matrix (2.30) can be revised by removing the last row when only the $\alpha\beta$ components are required. It is worth noting that, generally, both $\mathbf{L}_{\alpha\beta o}$ and $\mathbf{L}_{abc}$ are functions of the position then time–varying.

The $\alpha\beta o$ transformation can be generalised to the rotating orthogonal reference frame dqo with the direct d–, quadrature q– and homopolar o–axes. The angular position of the dqo reference frame is

3 The $\alpha\beta o$ transformation is also called *Clarke transformation* in honour of Edith Clarke, the first woman to earn an M.S. in electrical engineering from Massachusetts Institute of Technology MIT.

4 The matrix transformation between $\alpha\beta o$ reference frame to the abc stationary reference frame is usually stated as $\mathbf{T}_{\alpha\beta o/abc} = \mathbf{T}^{-1}_{abc/\alpha\beta o}$.

defined by ϑ_{me}, which is the angle between the d–axis of the rotating reference frame and the a–axis of the three–phase system. Neglecting the homopolar component, the transformation from the stationary reference frame to the rotating reference one can be defined as:

$$\mathbf{T}_{abc/dq} = \frac{2}{3}\begin{bmatrix} \cos(\vartheta_{me}) & \cos\left(\vartheta_{me} - \frac{2\pi}{3}\right) & \cos\left(\vartheta_{me} - \frac{4\pi}{3}\right) \\ -\sin(\vartheta_{me}) & -\sin\left(\vartheta_{me} - \frac{2\pi}{3}\right) & -\sin\left(\vartheta_{me} - \frac{4\pi}{3}\right) \end{bmatrix} \tag{2.34}$$

where $\mathbf{T}_{abc/dq} = \mathbf{T}_{abc/\alpha\beta}\mathbf{T}_{\alpha\beta/dq}$ and $\mathbf{T}_{\alpha\beta/dq}$ is the matrix transformation between the orthogonal reference frame and the rotating reference one defined as:

$$\mathbf{T}_{\alpha\beta/dq} = \begin{bmatrix} \cos(\vartheta_{me}) & \sin(\vartheta_{me}) \\ -\sin(\vartheta_{me}) & \cos(\vartheta_{me}) \end{bmatrix} \tag{2.35}$$

The transformation from the $\alpha\beta$ frame to the synchronous one is:

$$\boldsymbol{\xi}_{dq} = \mathbf{T}_{\alpha\beta/dq}\boldsymbol{\xi}_{\alpha\beta} \tag{2.36}$$

which is also called *Park transformation*[5].

By applying the (2.36) to the (2.33) the mathematical model of a linear synchronous motor in the dq reference frame is obtained as follow:

$$\mathbf{u}_{dq} = \mathbf{R}\mathbf{i}_{dq} + \mathbf{L}_{dq}\frac{d\mathbf{i}_{dq}}{dt} + \omega_{me}\mathbf{J}\mathbf{L}_{dq}\mathbf{i}_{dq} + \mathbf{e}_{dq} \tag{2.37}$$

where:

$$\mathbf{u}_{dq} = \begin{bmatrix} u_d \\ u_q \end{bmatrix}, \qquad \mathbf{i}_{dq} = \begin{bmatrix} i_d \\ i_q \end{bmatrix}, \qquad \mathbf{e}_{dq} = \begin{bmatrix} 0 \\ \omega_{me}\Lambda_{mg} \end{bmatrix} \tag{2.38}$$

and $\omega_{me} = p\omega_m$. The inductances matrix becomes constant due to the new reference frame which rotates integral with the rotor. The inductances matrix loses the position dependency then they become time invariant:

$$\mathbf{L}_{dq} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \tag{2.39}$$

where the d– and q–axis inductances are equal to:

$$\begin{aligned} L_d &= L_\sigma + \frac{3}{2}(L_0 - L_2) \\ L_q &= L_\sigma + \frac{3}{2}(L_0 + L_2) \end{aligned} \tag{2.40}$$

5 The trasformation between the dq rotating reference frame to the orthogonal reference one is $\mathbf{T}_{dq/\alpha\beta} = \mathbf{T}^{-1}_{\alpha\beta/dq}$

and the matrix $\mathbf{J}$ is defined as follows:

$$\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.41}$$

It is worth noting that the d– and q–axis inductances are equal if L_2 is equal to zero, i. e. with isotropic motor (see (2.24)). The matrix equation (2.37) can be split into the dq axes:

$$\begin{aligned} u_d &= Ri_d + L_d \frac{di_d}{dt} - \omega_{me} L_q i_q \\ u_q &= Ri_q + L_q \frac{di_q}{dt} + \omega_{me} L_d i_d + \omega_{me} \Lambda_{mg}. \end{aligned} \tag{2.42}$$

The derived electrical model is valid for both isotropic and anisotropic model with PM. It is advisable and suitable when the motor is not prone to magnetic saturation, as PMSMs, or to describe the d–axis of IPM motor which exhibits a quite linear behaviour.

The assumption about the linearity of the magnetic path is a strong limitation in PMASRM and SynR motor modelling, where the inductances change with the current magnitude. To overcome this restriction, it is possible to derive a more general electrical model which returns in (2.37) when the flux linkages are considered as a linear function of the stator currents.

2.1.4 *Electrical model including nonlinearity*

A general electrical model for a synchronous motor considering the magnetic saturation is:

$$\mathbf{u}_{abc} = \mathbf{R}\mathbf{i}_{abc} + \frac{d\boldsymbol{\lambda}_{abc}\left(\mathbf{i}_{abc}\right)}{dt} \tag{2.43}$$

where the flux linkage $\boldsymbol{\lambda}_{abc} = [\lambda_a\, \lambda_b\, \lambda_c]^T$ is considered instead of the product $\mathbf{L}_{abc}\mathbf{i}_{abc}$ as in (2.27). It is worth reminding that the flux linkage $\boldsymbol{\lambda}_{abc}$ is composed by the induced flux linkage generated by the stator currents and the one produced by the PM. The BEMF are included in the time derivative of the flux components.

The three–phase system can be represented in the $\alpha\beta o$ reference frame by using the transformation $\mathbf{T}_{abc/\alpha\beta o}$. With respect to the linear case, the homopolar component of the voltages can not null despite the absence of the neutral wire, while it is still zero in the currents. Anyway, the homopolar component of the flux linkages does not take part into the torque generation and therefore it can be neglected in (2.43). Applying the (2.31) to the (2.43), it yields:

$$\mathbf{u}_{\alpha\beta} = \mathbf{R}\mathbf{i}_{\alpha\beta} + \frac{d\boldsymbol{\lambda}_{\alpha\beta}\left(\mathbf{i}_{\alpha\beta}\right)}{dt}. \tag{2.44}$$

In order to formulate the model in the rotating reference frame, the (2.36) can be applied to (2.44) and yields:

$$\begin{aligned}\mathbf{u}_{dq} &= \mathbf{R}\mathbf{i}_{dq} + \mathbf{T}_{\alpha\beta/dq}\frac{d\left(\mathbf{T}_{dq/\alpha\beta}\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right)\right)}{dt} \\ &= \mathbf{R}\mathbf{i}_{dq} + \frac{d\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right)}{dt} + \omega_{me}\mathbf{J}\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right)\end{aligned} \tag{2.45}$$

where:

$$\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right) = \begin{bmatrix}\lambda_d\left(\mathbf{i}_{dq}\right) \\ \lambda_q\left(\mathbf{i}_{dq}\right)\end{bmatrix} \tag{2.46}$$

and $\lambda_d\left(\mathbf{i}_{dq}\right)$ and $\lambda_q\left(\mathbf{i}_{dq}\right)$ are the flux linkages on the d– and q–axis. It is worth noting that both flux linkages depend on both dq currents and $\lambda_d\left(\mathbf{i}_{dq}\right)$ takes into account the PM flux linkage Λ_{mg} as well. The currents dependency of the dq fluxes linkage will tacitly understood throughout the thesis to simplify the notation, unless it is required for clarity.

The model represented by (2.45) can be split into the dq axes as follows:

$$\begin{aligned}u_d &= Ri_d + \frac{d\lambda_d}{dt} - \omega_{me}\lambda_q \\ u_q &= Ri_q + \frac{d\lambda_q}{dt} + \omega_{me}\lambda_d\end{aligned} \tag{2.47}$$

where the flux linkages $\boldsymbol{\lambda}_{dq}$ can be represented as a product among the apparent inductances and the dq currents as:

$$\begin{aligned}\lambda_d &= L_d(\mathbf{i}_{dq})i_d + \Lambda_{mg} \\ \lambda_q &= L_q(\mathbf{i}_{dq})i_q.\end{aligned} \tag{2.48}$$

The equations (2.47) describe completely the non linear synchronous motor. Nevertheless, they are not usually useful to design the motor controller because most of the control techniques are based on the linearised equations of the motor. In order to linearise the (2.47), the chain rule can be used:

$$\frac{\partial\xi(x,y)}{\partial t} = \frac{\partial\xi(x,y)}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial\xi(x,y)}{\partial y}\frac{\partial y}{\partial t} \tag{2.49}$$

which allows to compute the derivative of the composition of two functions. Applying the (2.49) to the (2.47) yields:

$$\begin{aligned}u_d &= Ri_d + \frac{\partial\lambda_d}{\partial i_d}\frac{\partial i_d}{\partial t} + \frac{\partial\lambda_d}{\partial i_q}\frac{\partial i_q}{\partial t} - \omega_{me}\lambda_q \\ u_q &= Ri_q + \frac{\partial\lambda_q}{\partial i_d}\frac{\partial i_d}{\partial t} + \frac{\partial\lambda_q}{\partial i_q}\frac{\partial i_q}{\partial t} + \omega_{me}\lambda_d\end{aligned} \tag{2.50}$$

where the differential and cross–differential inductances are defined as follows:

$$\begin{aligned} L_d^{diff}(\mathbf{i}_{dq}) &= \frac{\partial \lambda_d}{\partial i_d}, \qquad L_q^{diff}(\mathbf{i}_{dq}) = \frac{\partial \lambda_q}{\partial i_q} \\ L_{dq}^{diff}(\mathbf{i}_{dq}) &= \frac{\partial \lambda_d}{\partial i_q}, \qquad L_{qd}^{diff}(\mathbf{i}_{dq}) = \frac{\partial \lambda_q}{\partial i_d}. \end{aligned} \tag{2.51}$$

The linearised model (2.50) can be compactly stated as:

$$\mathbf{u}_{dq} = \mathbf{R}\mathbf{i}_{dq} + \mathbf{L}_{dq}^{diff}(\mathbf{i}_{dq})\frac{d\mathbf{i}_{dq}}{dt} + \omega_{me}\mathbf{J}\mathbf{L}_{dq}(\mathbf{i}_{dq})\mathbf{i}_{dq} + \mathbf{e}_{dq} \tag{2.52}$$

where the vector $\mathbf{u}_{dq}$, $\mathbf{i}_{dq}$ and $\mathbf{e}_{dq}$ are defined in (2.38), the matrix $\mathbf{R}$ is the resistance matrix, the matrix $\mathbf{J}$ is defined in (2.41) while the inductance matrices are defined as follows:

$$\begin{aligned} \mathbf{L}_{dq}^{diff}(\mathbf{i}_{dq}) &= \begin{bmatrix} L_d^{diff}(\mathbf{i}_{dq}) & L_{dq}^{diff}(\mathbf{i}_{dq}) \\ L_{qd}^{diff}(\mathbf{i}_{dq}) & L_q^{diff}(\mathbf{i}_{dq}) \end{bmatrix} \\ \mathbf{L}_{dq}(\mathbf{i}_{dq}) &= \begin{bmatrix} L_d(\mathbf{i}_{dq}) & 0 \\ 0 & L_q(\mathbf{i}_{dq}) \end{bmatrix} \end{aligned} \tag{2.53}$$

It is worth reminding that apparent and differential inductances are correlated. Let a generic saturable inductor, the voltage equation is:

$$u(t) = \frac{d\lambda(i(t))}{dt} = \frac{d\lambda(i)}{di}\frac{di(t)}{dt} = L^{diff}(i(t))\frac{di(t)}{dt} \tag{2.54}$$

where $u(t)$, $i(t)$ are the voltage and the current respectively and $\lambda(i(t))$ is the flux linkage and it depends on the current. Furthermore, all the quantities are function of the time. By introducing the relation $\lambda(i(t)) = L^{app}(i(t))i(t)$ and assuming the apparent inductance $L^{app}(i(t))$ functions of the current only, follows:

$$\begin{aligned} u(t) &= \frac{d\lambda(i(t))}{dt} = \frac{d(L^{app}(i(t))i(t))}{dt} \\ &= L^{app}(i(t))\frac{di(t)}{dt} + i(t)\frac{dL^{app}(i(t))}{di(t)}\frac{di(t)}{dt} \\ &= \left(L^{app}(i(t)) + i(t)\frac{dL^{app}(i(t))}{di}\right)\frac{di(t)}{dt} \\ &= L^{diff}(i(t))\frac{di(t)}{dt} \end{aligned} \tag{2.55}$$

where the last equality derives from the (2.54). A graphic interpretation is sketch in Figure 2.2. Finally, it is worth highlighting that both L^{app} and L^{diff} are defined for a working point P which is described by the couple (i_{dc}, λ_{dc}).

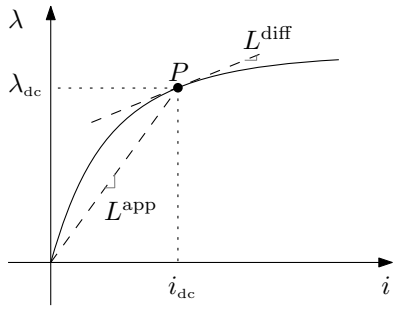


Figure 2.2: Illustration of the apparent and incremental inductances.

2.1.4.1 *Cross–differential inductances properties*

Cross–differential inductances hold some properties which are derived from the general laws of the electromagnetic field theory and from the geometrical symmetries of the rotor lamination [13]. One of them is called *reciprocity theorem* which is based on the energy conservation principle and yields the equality of the cross–differential inductances defined in (2.51). At locked rotor, the electric work dW due to infinitesimal flux linkage variation is given by:

$$\begin{aligned} dW &= \frac{3}{2}\left(i_d d\lambda_d + i_q d\lambda_q\right) = \\ &= \frac{3}{2}\left(i_d\left(\frac{\partial\lambda_d}{\partial i_d}di_d + \frac{\partial\lambda_d}{\partial i_q}di_q\right) + i_q\left(\frac{\partial\lambda_q}{\partial i_d}di_d + \frac{\partial\lambda_q}{\partial i_q}di_q\right)\right) \end{aligned} \tag{2.56}$$

and it is equal to the magnetic energy variation. The magnetic energy is a state function of the stator currents i_d and i_q then:

$$dW = \frac{\partial W\left(i_d, i_q\right)}{\partial i_d}di_d + \frac{\partial W\left(i_d, i_q\right)}{\partial i_q}di_q. \tag{2.57}$$

Comparing (2.56) and (2.57) results:

$$\begin{aligned} \frac{\partial W}{\partial i_d} &= \frac{3}{2}\left(i_d\frac{\partial\lambda_d}{\partial i_d} + i_q\frac{\partial\lambda_q}{\partial i_d}\right) \\ \frac{\partial W}{\partial i_q} &= \frac{3}{2}\left(i_d\frac{\partial\lambda_d}{\partial i_q} + i_q\frac{\partial\lambda_q}{\partial i_q}\right) \end{aligned} \tag{2.58}$$

and the two second mixed derivatives are expressed as:

$$\begin{aligned} \frac{\partial^2 W}{\partial i_d\partial i_q} &= \frac{3}{2}\left(i_d\frac{\partial^2\lambda_d}{\partial i_d\partial i_q} + \frac{\partial\lambda_q}{\partial i_d} + i_q\frac{\partial^2\lambda_q}{\partial i_d\partial i_q}\right) \\ \frac{\partial^2 W}{\partial i_q\partial i_d} &= \frac{3}{2}\left(i_d\frac{\partial^2\lambda_d}{\partial i_q\partial i_d} + \frac{\partial\lambda_d}{\partial i_q} + i_q\frac{\partial^2\lambda_q}{\partial i_q\partial i_d}\right). \end{aligned} \tag{2.59}$$

Schwarz's theorem for two variable functions yields:

$$\frac{\partial^2 W}{\partial i_d\partial i_q} = \frac{\partial^2 W}{\partial i_q\partial i_d} \tag{2.60}$$

then substituting (2.59) into (2.60) and simplifying equal terms on both sides equation, the reciprocity property is proven, then:

$$L_{dq}^{diff}\left(\mathbf{i}_{dq}\right) = L_{qd}^{diff}\left(\mathbf{i}_{dq}\right). \tag{2.61}$$

For the sake of completeness, the symmetries of the magnetic map are the follows:

$$\begin{cases} \lambda_d(i_d, i_q) = \lambda_d(i_d, -i_q) \\ \lambda_q(i_d, i_q) = -\lambda_q(i_d, -i_q) \\ \lambda_q(i_d, i_q) = \lambda_q(-i_d, i_q). \end{cases} \tag{2.62}$$

2.2 ELECTROMAGNETIC TORQUE AND MECHANICAL MODEL

The electromagnetic torque expression of a synchronous motor expressed in the dq rotating reference frame is equal to [14, 15]:

$$\tau = \frac{3}{2} p \left(\lambda_d i_q - \lambda_q i_d\right) + \frac{dW'_{fld}}{d\vartheta_m} \tag{2.63}$$

where W'_{fld} is the *magnetic coenergy* and it must be expressed as a state function of the state variables i_d, i_q and ϑ_m. It takes into account for all those effects which are usually neglected, such as the cogging torque. The magnetic coenergy is defined as:

$$W'_{fld}(i, \vartheta_m) = i\lambda - W_{fld}(\lambda, \vartheta_m) \tag{2.64}$$

where W_{fld} is the magnetic energy. The first term of the second member of (2.63) returns the average electromechanical torque generated by the motor, if an integral–slot winding motor is considered [14][6]. Finally, it is worth noting that (2.63) describes the torque production even in presence of the magnetic saturation and, in case of a linear motor, it can be simplified using the motor inductances as:

$$\tau = \frac{3}{2} p \left(\Lambda_{mg} i_q + (L_d - L_q) i_d i_q\right). \tag{2.65}$$

Two different contributions can be recognised in equation (2.65) and they result from different principles. The first one represents the torque related to the interaction among the rotor PM flux and the stator current while the second one is the reluctance torque which implies only stator–generated field. SynR motors are characterised by the lack of PM then the generated torque is based exclusively on the reluctance principle.

Throughout the present dissertation, only the average value of the torque (2.63) is considered and the variation of the magnetic coenergy

6 Only integral–slot windings motor are considered in this thesis.

is neglected. Therefore, the electromechanical torque expression is reduced to:

$$\tau = \frac{3}{2} p \left(\lambda_d i_q - \lambda_q i_d \right) \tag{2.66}$$

where the currents dependency is omitted.

Finally, the mechanical expression of the motor, including the load, is:

$$\tau = \tau_L + B\omega_m + J\frac{d\omega_m}{dt} \tag{2.67}$$

where B and J are the viscous friction coefficient of the system and the moment of inertia of the system, respectively. The load torque τ_L is independent from position and motor speed. It is worth highlighting that a proper mechanical parameters estimation is still challenging [16–18] and it plays a key role in high performances electric drives.

3

STATOR RESISTANCE IDENTIFICATION

The stator resistance knowledge is crucial in many cases, for example in fundamental–frequency–based sensorless control at low speed [19, 20], in flux–linkage estimation based on the integration of the BEMF [21, 22] or in thermal monitoring [23, 24]. A mismatch may be detrimental in control, parameters identification or monitoring of the electric drive. Nevertheless, the exact computation of the stator resistance is not as banal as it appears. Quite frequently the knowledge of the stator resistance is taken for granted and its estimation is even omitted in the study of the model.

Focusing on methods to estimate the stator resistance of a synchronous motor, and especially on techniques tailored for anisotropic motors, the already proposed methods can be subdivided in two categories. A first category is based on observers and recursive least square (RLS) algorithms [25–27] while the second group is constituted by injected–based techniques [28–30], implemented in open or closed loop way. Usually, the techniques of the first group assume other motor parameters constant and/or known. This assumption is not satisfactory with SynR motors due to magnetic saturation, that leads to the definition of many inductances, apparent and differential ones, which vary with the operation point. Quite often, the estimated stator resistance relies on another set of known or constant parameters. A further flaw is intrinsic in the RLS method. The convergence and the accuracy are subjected to several factors where the main one is the closeness of the initial guess to the real values. This key element is often neglected and poorly deepened.

On the other hand, in order to avoid the issues correlated to the methods of the former group, the latter ones excite the system injecting a direct current (DC) quantity. These methods can be implemented with two opposite strategies, i. e. injecting either DC current or voltage. Both are characterised by specific features, which make them suitable for different applications. The injection allows to become free–standing from other motor parameters but it generates a torque ripple in anisotropic motors. Despite the undesirable torque ripple, the parametric independence makes these techniques well tailored for SynR motor and PMASRM or, in general terms, for anisotropic machines. It is worth noting how an estimated parameter completely self–reliant from knowledge of the others allows to have a cornerstone in the process of identification as well as in the control.

3.1 RESISTANCE ESTIMATION PRINCIPLE

The voltage balance equation in the $\alpha\beta$ reference frame was obtained in (2.44) and recalled here for sake of clarity:

$$\mathbf{u}_{\alpha\beta} = R\mathbf{i}_{\alpha\beta} + \frac{d\boldsymbol{\lambda}_{\alpha\beta}\left(\mathbf{i}_{\alpha\beta}\right)}{dt}. \tag{3.1}$$

For a constant DC current vector $\mathbf{i}_{\alpha\beta} = \mathbf{I}_{\alpha\beta}$ or for a constant DC voltage vector $\mathbf{u}_{\alpha\beta} = \mathbf{U}_{\alpha\beta}$, the stator resistance can be estimated as:

$$R = \frac{|\mathbf{U}_{\alpha\beta}|}{|\mathbf{I}_{\alpha\beta}|} \tag{3.2}$$

regardless of the flux linkage knowledge. The conventional strategy to inject an additional signal into the motor consists in injecting a suitable voltage component, either an oscillating signal or a DC one, on the references generated by the current controller. In particular, the injection technique is widely study in sensorless control at low speed and standstill [31–33]. Once the DC voltage component is injected, the stator resistance can be estimated by measuring the corresponding DC current. The main drawback is posed by the current control, which rejects the injected voltage considering it as a disturbance. Actually, a constant DC voltage in $\alpha\beta$ is transformed into a sinusoidal voltage disturbance at the electromechanical frequency in the rotating reference frame. Usually, the frequency falls below the controller bandwidth, so that the DC voltage really applied for resistance sensing is far less than the imposed reference. If the resistance estimation is carried out on the voltage reference, the estimation will get inaccurate. It is worth highlighting how the mentioned issue is mitigated in a sensorless scheme since the frequency of the injected signal can be set high enough to reduce the attenuation of the current control. In order to overcome the drawback of the classical injection method, in this thesis the above procedure was reversed and a DC current is impressed into the motor. Exploiting the current control loop, the DC current is injected and the corresponding voltage is measured to estimate the stator resistance. The current control bandwidth in modern alternating current (AC) drives is usually broad enough to include the electromechanical frequency up to the nominal motor speed and more. In a practical implementation, the DC current in the $\alpha\beta$ reference frame can be produced by adding a sinusoidal signal $\delta\mathbf{i}^*_{dq} = [\delta i^*_d\ \delta i^*_q]^T$ (at the electromechanical frequency) to the reference of the dq current controllers as depicted in Figure 3.1. It induces a small current vector perturbation $\delta\mathbf{i}_{dq} = [\delta i_d\ \delta i_q]^T$ around the actual steady state working point (I_d, I_q).

Let:

$$\mathbf{G}_{dq}(s) = \begin{bmatrix} G_{dd}(s) & G_{dq}(s) \\ G_{qd}(s) & G_{qq}(s) \end{bmatrix} \tag{3.3}$$

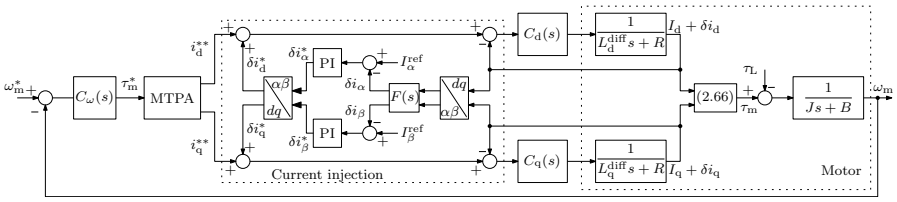


Figure 3.1: IPM motor and SynR motor drive block schematic with injection scheme.

be the transfer matrix between the reference $\delta\mathbf{i}^*_{dq}$ and the current vector $\delta\mathbf{i}_{dq}$ actually injected. The system introduces a certain attenuation and phase–lag at the frequency ω_{me} of the injected sinusoidal signal. Moreover, since the off–diagonal terms are not null, it also introduces a cross–coupling between the d– and q–axes, as Figure 3.2 shows. In the stationary reference frame, the actual injection current vector is obtained by the inverse Park's transformation as:

$$\delta\mathbf{i}_{\alpha\beta} = \mathbf{T}_{dq/\alpha\beta}(\vartheta_{me})\delta\mathbf{i}_{dq}. \tag{3.4}$$

The actual signal $\delta\mathbf{i}_{\alpha\beta}$ is generated according to the following compact expression:

$$\delta\mathbf{i}_{\alpha\beta} = \mathbf{G}_{\alpha\beta}(s)\delta\mathbf{i}^*_{\alpha\beta} \tag{3.5}$$

where

$$\mathbf{G}_{\alpha\beta}(s) = \mathbf{T}_{dq/\alpha\beta}(\vartheta_{me}) \times \mathbf{G}_{dq}(s) \times \mathbf{T}_{\alpha\beta/dq}(\vartheta_{me}). \tag{3.6}$$

Actually, (3.6) represents the *base–band* dynamical response of a modulated system, consisting of the following three parts: a modulation at the system input, performed by the transformation $\mathbf{T}_{\alpha\beta/dq}(\vartheta_{me})$, followed by a filtering in the *pass–band* (i.e. around the modulation frequency ω_{me}), and then finally a demodulation to recover the base–band signal, operated by the inverse transformation $\mathbf{T}_{dq/\alpha\beta}(\vartheta_{me})$. In Sec. 3.2 the linear, time–invariant (LTI) approximation of the time–variant system (3.6) was obtained using the method explained in [34]. Because of the modulation, the output of (3.6) contains, in addition to the base–band signals, sinusoidal terms at twice the modulation frequency, which can be removed by low–pass filtering. Therefore, in a practical implementation, the system (3.5) is replaced by:

$$\delta\mathbf{i}_{\alpha\beta} = \mathbf{F}(s)\mathbf{G}_{\alpha\beta}(s)\delta\mathbf{i}^*_{\alpha\beta} \tag{3.7}$$

where $\mathbf{F}(s) = \mathrm{diag}(F(s), F(s))$, being $F(s)$ the transfer function of a suitable low–pass filter which design is done in Sec. 3.3. In stationary reference frame, it is worth remembering that the measured currents and voltages are composed by the components due to the modulation superimposed on the fundamental ones. The low pass filter $F(s)$ has to be able to filter all the not constant components, not only the sinusoidal terms due to the modulation.

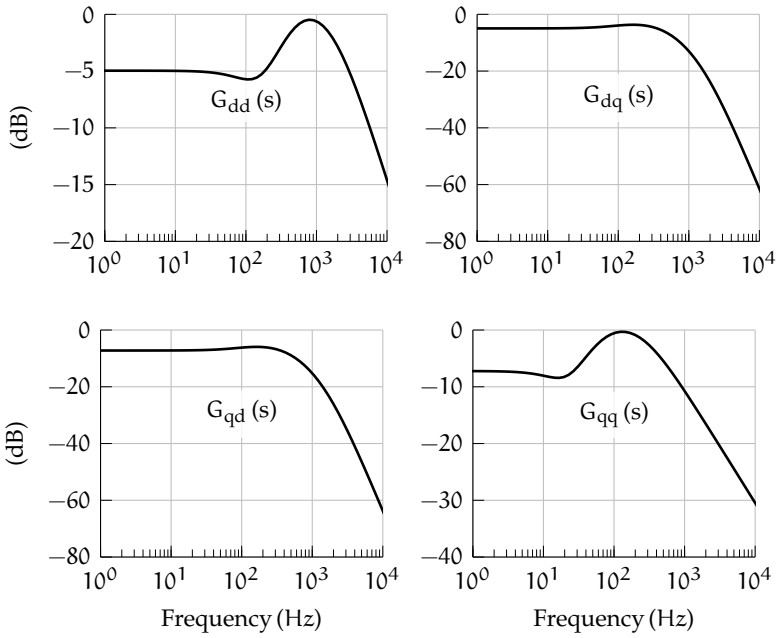


Figure 3.2: Magnitude of $\mathbf{G}_{dq}(s)$ elements at nominal torque, MTPA condition.

In order to keep the implementation of the resistance estimation scheme as simple as possible, it is convenient to inject only on a single axis, so that for the same amplitude of the current injection vector the measured scalar current is maximum. In doing this, (3.2) reduces to a ratio of two scalar quantities. For example, if the DC current is injected along the α–axis, U_α is the only voltage component to be measured, then the stator resistance estimation becomes $R = U_\alpha / I_\alpha$. Unfortunately, forcing a reference $\mathbf{I}^*_{\alpha\beta}$ directed only along the α–axis is not sufficient for the purpose, since the attenuation, phase–lag and cross–coupling effects introduced by (3.5) inevitably produce a current component also along the β–axis, as shown in Figure 3.3. Therefore, the resistance estimation carried out in an open loop manner could induce a low accuracy estimation or a DC current bigger than the reference. In order to highlight the issue, the Table 3.1 can be referred which was obtained on an experimental setup.

The tests were performed in two different and particular working points and they can help to stress out the concept outlined above. In both tests, the reference current vector was set $I^*_{\alpha\beta} = [0.05\,0]^T$ A. The first test was performed at 280 rpm in open loop. The influence of the outer speed control almost nullify the actual $\alpha\beta$ currents (and voltages) so that the imprecision in the measurements (even after a mean value post–processing) is quite inaccurate (error of -18.66%). Another critical situation is shown with speed reference of 420 rpm, where the open loop current δi_α is close to the reference but the cross–coupling also produces a non–negligible δi_β. In this case, while the resistance estimation is still correct and robust, the amplitude of the

Speed	280 rpm	420 rpm
δi_α	0.047 A	0.043 A
δi_β	0 A	−0.085 A
$\sqrt{\delta i_\alpha^2 + \delta i_\beta^2}$	0.047 A	0.095 A
u_α^{DC}	0.0177 V	0.205 V
R	3.76 Ω	4.78 Ω
Error	−18.66%	2.12%

Table 3.1: Estimation resistance obtained in open–loop injection.

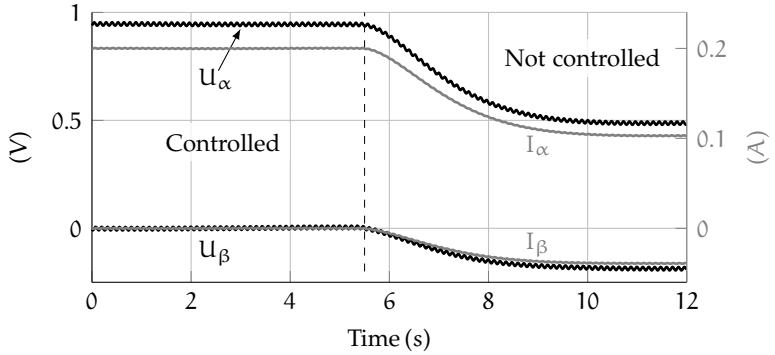


Figure 3.3: DC voltages and currents under closed–loop control ($0 \leqslant t \leqslant 5.5s$) and at open–loop ($t > 5.5s$). The detrimental cross–coupling effect is evident.

injected current vector is 2 times the closed loop case. To overcome this issue, the actual reference $\delta \mathbf{i}^*_{\alpha\beta}$ must be generated by a closed loop current control for the DC current levels (Figure 3.1), as detailed in Sec. 3.3.

3.2 TRANSFER FUNCTION ANALYSIS

Accepting some simplifying assumptions, the transfer function elements of (3.3) can be derived with a relatively modest effort. To this purpose, one can assume that dq decoupling and BEMF compensations are accomplished in the current control loop. For the derivation of (3.3), the whole control system of Figure 3.1 has to be linearised around a specified working point. This, in turn, requires the linearisation of the block responsible of the generation of the current references $\mathbf{i}^{**}_{dq} = [i^{**}_d \; i^{**}_q]^T$ and the block representing the mechanism of motor torque generation. The former is marked as MTPA in the block diagram of Figure 3.1, since it is assumed that the current references are generated to attain the maximum torque–per–ampere condition. The latter

is instead related to the expression for torque generation obtained in (2.66) and recalled here:

$$\tau = \frac{3}{2} p \left(\lambda_d i_q - \lambda_q i_d \right). \tag{3.8}$$

In a steady–state operating condition with constant motor speed, the linearisation of the MTPA block is performed around the constant torque reference command T^* imposed by the speed controller. The following two linear expressions result for a small perturbation of the current references:

$$\delta i_d^{**} = \gamma_d \delta\tau^*, \qquad \delta i_q^{**} = \gamma_q \delta\tau^* \tag{3.9}$$

where $\delta\tau^*$ is a small perturbation of the torque reference provided to the MTPA block by the speed controller, while γ_d and γ_q are two coefficients that depend on the steady–state torque T^*. For the linearisation of the motor torque, after replacing:

$$i_d = I_d + \delta i_d, \qquad i_q = I_q + \delta i_q \tag{3.10}$$

within (3.8), and linearising the expressions of the flux linkages $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ for small current perturbations, the expression for the torque perturbation around T^* is:

$$\delta\tau = \zeta_d \delta i_d + \zeta_q \delta i_q \tag{3.11}$$

with

$$\begin{aligned} \zeta_d &= -\frac{3}{2} p \left(\Lambda_q + L_{dq}^{diff} I_d - L_d^{diff} I_q \right) \\ \zeta_q &= \frac{3}{2} p \left(\Lambda_d + L_{dq}^{diff} I_q - L_q^{diff} I_d \right) \end{aligned} \tag{3.12}$$

where Λ_d and Λ_q are the dq flux linkages in the operation point.

By this linearisation, after some tedious algebraic computations, it is possible to derive the following expressions for the transfer function elements of (3.3):

$$\begin{aligned} G_{dd}(s) &= \frac{\delta i_d(s)}{\delta i_d^*(s)} = H_d(s) \frac{1 + \gamma_q \zeta_q H_q(s) P_m(s) C_\omega(s)}{D(s)} \\ G_{dq}(s) &= \frac{\delta i_d(s)}{\delta i_q^*(s)} = -H_d(s) \frac{\gamma_d \zeta_q H_q(s) P_m(s) C_\omega(s)}{D(s)} \\ G_{qd}(s) &= \frac{\delta i_q(s)}{\delta i_d^*(s)} = -H_q(s) \frac{\gamma_q \zeta_d H_d(s) P_m(s) C_\omega(s)}{D(s)} \\ G_{qq}(s) &= \frac{\delta i_q(s)}{\delta i_q^*(s)} = H_q(s) \frac{1 + \gamma_d \zeta_d H_d(s) P_m(s) C_\omega(s)}{D(s)} \end{aligned} \tag{3.13}$$

where $H_{d,q}(s)$ are the closed–loop transfer functions of the dq current control loops (from references $\delta \mathbf{i}_{dq}^{**}$ to measures $\delta \mathbf{i}_{dq}$), $P_m(s) = 1/(Js +$

B) is the motor load transfer function, $C_\omega(s)$ is the transfer function of the PI speed controller, while the common denominator is:

$$D(s) = 1 + \left(\gamma_d \zeta_d H_d(s) + \gamma_q \zeta_q H_q(s)\right) P_m(s) C_\omega(s). \tag{3.14}$$

The expressions (3.13) can be slightly simplified by assuming that the two PI current controllers are designed to cancel the electrical poles of the dq axes (at the specified working point, at least for the nominal case), i. e. by assuming that their transfer functions are set as follows:

$$C_d(s) = \omega_c \frac{L_d^{diff} s + R}{s}, \quad C_q(s) = \omega_c \frac{L_q^{diff} s + R}{s} \tag{3.15}$$

where ω_c is the desired closed–loop bandwidth of the current control loops. In fact, provided that the dq voltage–to–current transfer functions are approximated by:

$$P_d(s) = \frac{1}{L_d^{diff} s + R}, \quad P_q(s) = \frac{1}{L_q^{diff} s + R} \tag{3.16}$$

the closed–loop transfer functions of the dq current loops reduce to:

$$H_d(s) = H_q(s) = H(s) \triangleq \frac{\omega_c}{s + \omega_c}. \tag{3.17}$$

Moreover, (3.14) reduces to:

$$\begin{aligned} D(s) &= 1 + (\gamma_d \zeta_d + \gamma_q \zeta_q) H(s) P_m(s) C_\omega(s) \\ &= 1 + k_\tau H(s) P_m(s) C_\omega(s) \end{aligned} \tag{3.18}$$

where $k_\tau = \gamma_d \zeta_d + \gamma_q \zeta_q$. In order to achieve an invariant bandwidth current control over the whole operating region with an anisotropic motors, like SynR motors, special control schemes are demanded. A simple but effective solution is reported in Chapter 7.

After linearising the block diagram of Figure 3.1 around a specific working point, it follows quite easily that the closed–loop transfer function $T_\omega(s)$ from the speed reference $\delta\omega_m^*$ (small–signal deviation around the working point value) to the actual speed $\delta\omega_m$ is:

$$T_\omega(s) = \frac{\delta\omega_m(s)}{\delta\omega_m^*(s)} = \frac{k_\tau H(s) P_m(s) C_\omega(s)}{D(s)}. \tag{3.19}$$

Assuming that the speed controller $C_\omega(s)$ is designed to achieve a certain phase margin φ_m^* at a specified gain crossover frequency ω_n^* and according to the *dominant–pole approximation method* [35], the transfer function $T_\omega(s)$ can be approximated as follows (*dominant–pole approximation*):

$$T_\omega(s) \approx \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{3.20}$$

with $\omega_n = \omega_n^*$ and $\xi = \sin(\varphi_m^*/2)$. From (3.13), after some mathematical manipulations it follows that:

$$G_{dd}(s) = H(s)\,[1 - K_{dd}T_\omega(s)] \quad \text{and} \quad K_{dd} \triangleq \frac{\gamma_d \zeta_d}{k_\tau}. \tag{3.21}$$

After replacing (3.20) in (3.21) and by assuming that $H(s) \approx 1$ (since $\omega_n \ll \omega_c$), (3.21) reduces to:

$$G_{dd}(s) \approx \frac{s^2 + 2\xi\omega_n s + \omega_n^2(1 - K_{dd})}{s^2 + 2\xi\omega_n s + \omega_n^2}. \tag{3.22}$$

Similarly, the following dominant–pole approximations can be derived for the remaining transfer functions in (3.13):

$$G_{dq} \approx -\frac{K_{dq}\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}, \quad G_{qd} \approx -\frac{K_{qd}\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

$$G_{qq} \approx \frac{s^2 + 2\xi\omega_n s + \omega_n^2(1 - K_{qq})}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{3.23}$$

where

$$K_{dq} = \frac{\gamma_d \zeta_q}{k_\tau}, \quad K_{qd} = \frac{\gamma_q \zeta_d}{k_\tau}, \quad K_{qq} = \frac{\gamma_q \zeta_q}{k_\tau}. \tag{3.24}$$

The approximations (3.22)–(3.23) are used next to derive the base–band approximations of the transfer matrix $\mathbf{G}_{\alpha\beta}(s)$. The right–hand side of (3.6) can be expanded to obtain the expressions of each element $G_{\alpha\alpha}(s)$, $G_{\alpha\beta}(s)$, $G_{\beta\alpha}(s)$ and $G_{\beta\beta}(s)$. For sake of clarity, the transfer function $G_{\alpha\alpha}(s)$ (element in position $(1,1)$ of the transfer matrix $\mathbf{G}_{\alpha\beta}(s)$) is obtained as:

$$\begin{aligned} G_{\alpha\alpha}(s) = {} & \cos\vartheta_{me} \times G_{dd}(s) \times \cos\vartheta_{me} - \cdots \\ & \cdots - \cos\vartheta_{me} \times G_{dq}(s) \times \sin\vartheta_{me} - \cdots \\ & \cdots - \sin\vartheta_{me} \times G_{qd}(s) \times \cos\vartheta_{me} + \cdots \\ & \cdots + \sin\vartheta_{me} \times G_{qq}(s) \times \sin\vartheta_{me}. \end{aligned} \tag{3.25}$$

Similar expressions can be obtained for the remaining transfer function elements $G_{\alpha\beta}(s)$, $G_{\beta\alpha}(s)$ and $G_{\beta\beta}(s)$. That is, each element of the matrices is composed by addends that represents the sequence of a modulation (with carrier frequency ω_{me}), a band–pass filtering, and a demodulation.

For example, the first term coincides with the modulated system shown in Figure 3.4, where $G(s) = G_{dd}(s)$, up to the demodulated output y. The final low–pass filter $F(s)$ is used to remove harmonics at twice the modulation frequency, produced by the demodulation process – see (3.7).

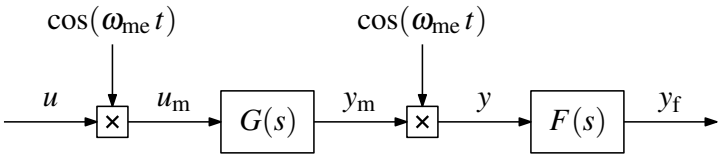


Figure 3.4: Modulated system.

As shown in [35], the Laplace transform of the modulated signal $\sigma(t) = \cos(\omega_{me} t + \phi)\, x(t)$ is:

$$\mathcal{L}(\sigma) = \frac{\left[e^{j\phi} X(s - j\omega_{me}) + e^{-j\phi} X(s + j\omega_{me})\right]}{2}. \tag{3.26}$$

This equality can be effectively exploited to obtain an approximated transfer function of the modulated system depicted in Figure 3.4. According to (3.26), the transfer function of the modulated input u_m is:

$$U_m(s) = \frac{1}{2}\left[U(s - j\omega_{me}) + U(s + j\omega_{me})\right]. \tag{3.27}$$

The modulated signal is then filtered by the pass–band filter $G(s)$ to get:

$$Y_m(s) = G(s) U_m(s). \tag{3.28}$$

The property (3.26) can be further applied to obtain the Laplace transform of the demodulated output, namely:

$$Y(s) = \frac{1}{2}\left[Y_m(s - j\omega_{me}) + Y_m(s + j\omega_{me})\right]. \tag{3.29}$$

Expression (3.29) can be expanded by using (3.27) and (3.28) as follow:

$$\begin{aligned} Y(s) = \frac{1}{4}&\left[G(s - j\omega_{me}) + G(s + j\omega_{me})\right] U(s) + \cdots \\ \cdots + \frac{1}{4}&\left[G(s - j\omega_{me}) U(s - 2j\omega_{me}) + \cdots\right. \\ &\left. \cdots + G(s + j\omega_{me}) U(s + 2j\omega_{me})\right]. \end{aligned} \tag{3.30}$$

After filtering with the low–pass filter $F(s)$, the terms at twice the modulation frequency in (3.30) are suppressed, and the Laplace transform of the demodulated output becomes:

$$Y_f(s) \approx F(s) G_{m,cc}(s, \omega_{me}) U(s) \tag{3.31}$$

where

$$G_{m,cc}(s, \omega_{me}) = \frac{1}{4}\left[G(s - j\omega_{me}) + G(s + j\omega_{me})\right]. \tag{3.32}$$

It is therefore possible to conclude that the original modulated system can be represented with the following base–band approximation:

$$F(s)\left[\cos\vartheta_{me} \times G(s) \times \cos\vartheta_{me}\right] \approx F(s) G_{m,cc}(s, \omega_{me}). \tag{3.33}$$

A similar procedure can be adopted to obtain the base–band approximation of a modulated system with any other combination of modulation and demodulation carriers (corresponding to the other terms in $G_{\alpha\alpha}(s)$, as well as in all the other elements of $\mathbf{G}_{\alpha\beta}(s)$). Equivalently to (3.33), the remaining terms are:

$$\begin{aligned} F(s)\left[\sin\vartheta_{me} \times G(s) \times \cos\vartheta_{me}\right] &\approx F(s)G_{m,sc}(s,\omega_{me}) \\ F(s)\left[\cos\vartheta_{me} \times G(s) \times \sin\vartheta_{me}\right] &\approx F(s)G_{m,cs}(s,\omega_{me}) \\ F(s)\left[\sin\vartheta_{me} \times G(s) \times \sin\vartheta_{me}\right] &\approx F(s)G_{m,ss}(s,\omega_{me}) \end{aligned} \tag{3.34}$$

with

$$\begin{aligned} G_{m,sc}(s,\omega_{me}) &= \frac{1}{4j}\left[G(s-j\omega_{me}) - G(s+j\omega_{me})\right] \\ G_{m,cs}(s,\omega_{me}) &= -\frac{1}{4j}\left[G(s-j\omega_{me}) - G(s+j\omega_{me})\right] \\ G_{m,ss}(s,\omega_{me}) &= \frac{1}{4}\left[G(s-j\omega_{me}) + G(s+j\omega_{me})\right]. \end{aligned} \tag{3.35}$$

Therefore, the final expression of (3.25) is:

$$\begin{aligned} G_{\alpha\alpha} = F(s)\big(&G_{dd,cc}(s,\omega_{me}) - G_{dq,sc}(s,\omega_{me}) - \cdots \\ &\cdots - G_{qd,cs}(s,\omega_{me}) + G_{qq,ss}(s,\omega_{me})\big) \end{aligned} \tag{3.36}$$

where the subscript m is substituted by the identifier of the modulated transfer function, e.g.:

$$G_{dd,cc}(s,\omega_{me}) = \frac{1}{4}\left[G_{dd}(s-j\omega_{me}) + G_{dd}(s+j\omega_{me})\right] \tag{3.37}$$

The subscript cc, cs, sc and ss indicates the modulating terms, for instance cc means that the transfer function is pre and post–multiplied by $\cos(\vartheta_{me})$.

The base–band expression of (3.36) can be accordingly further approximated as:

$$G_{\alpha\alpha}(s) \approx F(s)G'_{\alpha\alpha}(s) \tag{3.38}$$

where

$$G'_{\alpha\alpha}(s) = \frac{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{3.39}$$

with

$$\begin{aligned} b_0 &= 2\omega_n^4 - K_{dd}\omega_n^4 - K_{qq}\omega_n^4 + 2K_{dq}\xi\omega_n^3\omega_{me} - \cdots \\ &\quad \cdots - 2K_{qd}\xi\omega_n^3\omega_{me} - 4\omega_n^2\omega_{me}^2 + K_{dd}\omega_n^2\omega_{me}^2 + \cdots \\ &\quad \cdots + K_{qq}\omega_n^2\omega_{me}^2 + 8\xi^2\omega_n^2\omega_{me}^2 + 2\omega_{me}^4 \\ b_1 &= 8\xi\omega_n^3 - 2K_{dd}\xi\omega_n^3 - 2K_{qq}\xi\omega_n^3 + 2K_{dq}\omega_n^2\omega_{me} - \cdots \\ &\quad \cdots - 2K_{qd}\omega_n^2\omega_{me} + 8\xi\omega_n\omega_{me}^2 \\ b_2 &= 4\omega_n^2 - K_{dd}\omega_n^2 - K_{qq}\omega_n^2 + 8\xi^2\omega_n^2 + 4\omega_{me}^2 \\ b_3 &= 8\xi\omega_n, \qquad b_4 = 2 \end{aligned} \tag{3.40}$$

and

$$\begin{aligned} a_0 &= 2\left(\omega_n^4 - 2\omega_n^2\omega_{me}^2 + 4\xi^2\omega_n^2\omega_{me}^2 + \omega_{me}^4\right) \\ a_1 &= 2\left(4\xi\omega_n^3 + 4\xi\omega_n\omega_{me}^2\right) \\ a_2 &= 2\left(2\omega_n^2 + 4\xi^2\omega_n^2 + 2\omega_{me}^2\right) \\ a_3 &= 8\xi\omega_n, \quad a_4 = 2. \end{aligned} \tag{3.41}$$

It can be verified that the four poles of (3.39) are equal to:

$$\begin{aligned} s_{1,2} &= -\xi\omega_n \pm j\omega_n\left(r_\omega + \sqrt{1-\xi^2}\right) \\ s_{3,4} &= -\xi\omega_n \pm j\omega_n\left(r_\omega - \sqrt{1-\xi^2}\right) \end{aligned} \tag{3.42}$$

where $r_\omega = \omega_{me}/\omega_n$. The frequencies of the two pole pairs are obtained by evaluating their magnitudes:

$$\begin{aligned} |s_{1,2}| &= \omega_n\sqrt{1 + r_\omega^2 + 2r_\omega\sqrt{1-\xi^2}} \\ |s_{3,4}| &= \omega_n\sqrt{1 + r_\omega^2 - 2r_\omega\sqrt{1-\xi^2}}. \end{aligned} \tag{3.43}$$

The dominant poles are those with lower frequency, namely $s_{3,4}$. Therefore, the dominant–pole approximation of (3.39) is equal to:

$$\begin{aligned} G'_{\alpha\alpha}(s) &\approx G'_{\alpha\alpha,0}\frac{s_3 s_4}{(s-s_3)(s-s_4)} \\ &= G'_{\alpha\alpha,0}\frac{\omega_d^2}{s^2 + 2\xi_d\omega_d s + \omega_d^2} \end{aligned} \tag{3.44}$$

where

$$\omega_d = |s_{3,4}|, \qquad \xi_d = \frac{\xi\omega_n}{\omega_d} \tag{3.45}$$

and $G'_{\alpha\alpha,0}$ is the DC gain of (3.39), namely:

$$G'_{\alpha\alpha,0} = G'_{\alpha\alpha}(0) = \frac{1}{2}\cdot\frac{1 + \left(8\xi^2 - 3\right)r_\omega^2 + 2r_\omega^4}{1 + 2\left(2\xi^2 - 1\right)r_\omega^2 + r_\omega^4}. \tag{3.46}$$

The procedure outlined above can be repeated to determine the base–band approximations of the other transfer function elements in $\mathbf{G}_{\alpha\beta}(s)$. For example, it can be verified that:

$$G_{\alpha\beta}(s) \approx F(s)G'_{\alpha\beta}(s) \tag{3.47}$$

where

$$G'_{\alpha\beta}(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{3.48}$$

with

$$\begin{aligned} b_0 &= 2\xi K_{dd}\omega_n^3\omega_{me} - K_{dq}\omega_n^4 + K_{dq}\omega_n^2\omega_{me}^2 + \cdots \\ &\quad \cdots + K_{qd}\omega_n^4 - K_{qd}\omega_n^2\omega_{me}^2 + 2\xi K_{qq}\omega_n^3\omega_{me} \\ b_1 &= 2K_{dd}\omega_n^2\omega_{me} - 2\xi K_{dq}\omega_n^3 + 2\xi K_{qd}\omega_n^3 + 2K_{qq}\omega_n^2\omega_{me} \\ b_2 &= K_{qd}\omega_n^2 - K_{dq}\omega_n^2 \end{aligned} \tag{3.49}$$

while $a_0, \ldots, a_4$ are identical to (3.41). Since $G'_{\alpha\beta}(s)$ and $G'_{\alpha\alpha}(s)$ share the same denominator, the dominant–pole approximation of $G'_{\alpha\beta}(s)$ is equal to:

$$G'_{\alpha\beta}(s) \approx G'_{\alpha\beta,0} \frac{\omega_d^2}{s^2 + 2\xi_d \omega_d s + \omega_d^2} \tag{3.50}$$

where ξ_d and ω_d are defined as in (3.45), while:

$$G'_{\alpha\beta,0} = G'_{\alpha\beta}(0) = \frac{\xi r_\omega}{1 + 2\left(2\xi^2 - 1\right) r_\omega^2 + r_\omega^4}. \tag{3.51}$$

For the remaining transfer functions, it can be verified that:

$$G_{\beta\alpha}(s) \approx F(s) G'_{\beta\alpha}(s), \quad G_{\beta\beta}(s) \approx F(s) G'_{\beta\beta}(s) \tag{3.52}$$

where $G'_{\beta\alpha}(s) = -G'_{\alpha\beta}(s)$ and $G'_{\beta\beta}(s) = G'_{\alpha\alpha}(s)$.

The expressions of the DC gains (3.46) and (3.51) are obtained by observing that under MTPA condition it holds:

$$\gamma_d \zeta_q - \gamma_q \zeta_d = 0. \tag{3.53}$$

The expression (3.53) can be proven observing that the MTPA curve is the locus of the operating point in the dq plane obtained by solving the following constrained maximisation problem:

$$\max_{i_d, i_q} \ \tau\left(i_d, i_q\right) \qquad \text{subject to} \qquad i_d^2 + i_q^2 = |i|^2 \tag{3.54}$$

where $|i|$ is a certain constant current level. The problem can be solved by resorting to the method of the Lagrange's multipliers. To this aim, let us to define the Lagrangian function:

$$\mathcal{L}\left(i_d, i_q, \lambda\right) = \tau - \lambda\left(i_d^2 + i_q^2 - |i|^2\right) \tag{3.55}$$

where λ is the Lagrange's multiplier. The solution of the original constrained maximisation problem corresponds to a stationary point of the Lagrangian function, i. e. a solution of the following set of equations:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial i_d} = \dfrac{\partial \tau}{\partial i_d} - 2\lambda i_d = \xi_d - 2\lambda i_d = 0 & \text{(3.56a)} \\ \dfrac{\partial \mathcal{L}}{\partial i_q} = \dfrac{\partial \tau}{\partial i_q} - 2\lambda i_q = \xi_q - 2\lambda i_q = 0 & \text{(3.56b)} \\ \dfrac{\partial \mathcal{L}}{\partial \lambda} = i_d^2 + i_q^2 - |i|^2 = 0 & \text{(3.56c)} \end{cases}$$

Solving (3.56a) and (3.56b) for ξ_d and ξ_q and exploiting the (3.56c) yields:

$$\lambda = \frac{\sqrt{\xi_d^2 + \xi_q^2}}{2|i|} \tag{3.57}$$

that can be replaced into (3.56a) and (3.56b) to get the following solutions:

$$i_d = \frac{\xi_d}{\sqrt{\xi_d^2 + \xi_q^2}}|i|, \qquad i_q = \frac{\xi_q}{\sqrt{\xi_d^2 + \xi_q^2}}|i| \tag{3.58}$$

which is the MTPA curve parametrised according to the variable current vector magnitude. Therefore, on the MTPA curve it holds that:

$$\begin{aligned} \gamma_d &= \frac{\partial i_d}{\partial \tau} = \frac{\partial i_d}{\partial |i|}\frac{\partial |i|}{\partial \tau} = \frac{\xi_d}{\sqrt{\xi_d^2 + \xi_q^2}}\frac{\partial |i|}{\partial \tau} \\ \gamma_q &= \frac{\partial i_q}{\partial \tau} = \frac{\partial i_q}{\partial |i|}\frac{\partial |i|}{\partial \tau} = \frac{\xi_q}{\sqrt{\xi_d^2 + \xi_q^2}}\frac{\partial |i|}{\partial \tau} \end{aligned} \tag{3.59}$$

which implies:

$$\gamma_d \xi_q - \gamma_q \xi_d = 0. \tag{3.60}$$

In brief, the base–band dynamical response of the modulated system (3.7) can be approximated with the following dominant–pole approximated model:

$$\mathbf{F}(s)\mathbf{G}_{\alpha\beta}(s) \approx \begin{bmatrix} G'_{\alpha\alpha}(0) & G'_{\alpha\beta}(0) \\ -G'_{\alpha\beta}(0) & G'_{\alpha\alpha}(0) \end{bmatrix} F(s)G_d(s) \tag{3.61}$$

where $G'_{\alpha\alpha}(0)$ and $G'_{\alpha\beta}(0)$ are defined in (3.46) and (3.51), $F(s)$ is a suitable low–pass filter and $G_d(s)$ represents the approximated dynamic of the system (as obtained in (3.44)) and it is defined as follows:

$$G_d(s) = \frac{\omega_d^2}{s^2 + 2\xi_d\omega_d s + \omega_d^2}. \tag{3.62}$$

It is worth noting that the base–band approximation (3.61) can be further simplified if the cut–off frequency of the low–pass filter $F(s)$ is chosen sufficiently smaller than the frequency ω_d of the poles of $G_d(s)$. Therefore, the base–band response is dominated by the low–pass filter dynamics and (3.61) can be approximated as follows:

$$\mathbf{F}(s)\mathbf{G}_{\alpha\beta} \approx \begin{bmatrix} G'_{\alpha\alpha}(0) & G'_{\alpha\beta}(0) \\ -G'_{\alpha\beta}(0) & G'_{\alpha\alpha}(0) \end{bmatrix} F(s). \tag{3.63}$$

Equation (3.63) can be effectively exploited to design the current–injection PI controllers in the $\alpha\beta$ frame, as reported in Sec. 3.3.

3.2.1 *Base–band dynamical approximation analysis*

The study of the base–band dynamical approximation (3.63) is of paramount importance. It is not only crucial to tune the current–injection PI controllers but it highlights some potential issues in the current–injection scheme. For extremely low frequencies (as assumed to get (3.63)), the system behaviour can be approximated with a function which depends on two variables that characterise the speed control, i. e. phase margin φ_m and crossover frequency ω_n, and the current rotating speed ω_{me}. It is worth noting how the obtained approximation is analytic then the meaning of the terms that compose it is self–explicative. The expression (3.63), without the filter $F(s)$, represents the DC gains of the system. From the control point of view, their values are crucial and, in particular, the matrix should be a constant not–null identity matrix to obtain a decoupled and steady behaviour of the current–injection PI controllers. Actually, the aforementioned condition is extremely complicated to be obtained and a critical issue could appear. In some condition, the main diagonal entries $G'_{\alpha\alpha}(0)$ can become null. From control theory, a system characterised by a null gain is not controllable. Therefore, the DC current injection becomes impossible and this entails the impossibility to estimate the stator resistance.

In order to understand how to avoid critical conditions, the expression (3.46) has to be analysed. It is quite evident that the crossover frequency ω_n of the speed control does not play a crucial role since only shifts the ratio among the electromechanical frequency and the crossover one. On the other hand, the speed regulator phase margin deeply influences the shape of $G'_{\alpha\alpha}(0)$, as shown in Figure 3.5. If the phase margin is small, the shape of $G'_{\alpha\beta}(0)$ is characterised by a sort of oscillation. The smaller the phase margin, the deeper the valley is. There is a limit value of phase margin such that $G'_{\alpha\beta}(0)$ will be zero or even negative. With a DC gain close to zero, the current injection becomes not trivial. Therefore, the stator resistance cannot be properly estimated with a small phase margin of the speed controller. It is worth noting how the aforementioned issue should not appear in a standard electric drive since the speed controller is properly tuned, with a phase margin of 50° or more degrees. Nevertheless, if the load inertia changes a lot during the normal operation of the drive, some issues could appear. The purpose of this section is to highlight the issue associated to the current injection which could affect some sensorless schemes as well.

3.3 CLOSED–LOOP INJECTION CONTROL

Two independent PI controllers are designed to regulate the DC current levels in the $\alpha\beta$ frame. The approximated dynamics to be controlled

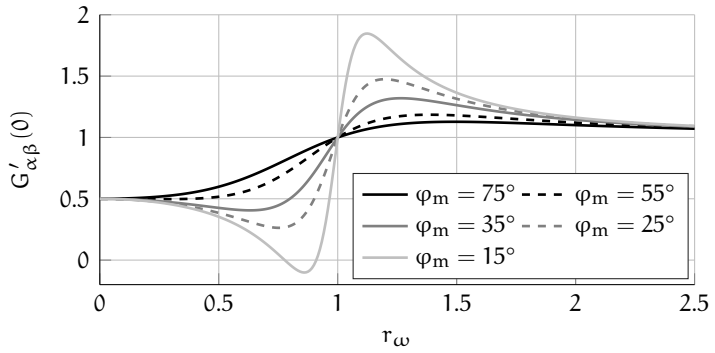


Figure 3.5: $G'_{\alpha\alpha}(0)$ magnitude with different phase margins of the speed control and a constant crossover frequency equal to 15 rad/s.

(plant dynamics) is that reported in (3.63). Although a coupling is present between the two axes because of the off–diagonal components in $\mathbf{G}_{\alpha\beta}(s)$, the cross–coupling terms can be regarded as two independent exogenous disturbances, and the two PI controllers can be separately designed by taking the elements on the main diagonal of (3.63) as the actual plant dynamics (*decoupled* design). This is the same for both axes, so that:

$$\frac{\delta i^*_\alpha(s)}{\delta i_\alpha(s)} = \frac{\delta i^*_\beta(s)}{\delta i_\beta(s)} = G'_{\alpha\alpha}(0)F(s). \tag{3.64}$$

The PI design can be performed with conventional methods, e. g. frequency–domain methods aimed to achieve a certain phase margin $\varphi_{m,\alpha\beta}$ at a desired control bandwidth $\omega_{n,\alpha\beta}$ (approximately equal to the control bandwidth). Since the gain $G'_{\alpha\alpha}(0)$ is slightly dependent on the working point, i.e. the motor speed, the design has to be carried out with a certain nominal value of such gain, obtained for a selected motor speed. For a robust design, for which the control performance is practically insensitive on the actual working point, it is preferable to select the control bandwidth smaller or equal to the filter cut–off frequency ω_d, and a sufficiently high phase margin. In this work, the bandwidth of the PI controller was set equal to ω_d and the phase margin was 70°. It is immediate to verify that the PI gains required to achieve the desired specifications in terms of control bandwidth and phase margin are (see [35]):

$$k_p = \kappa\cos(\sigma), \qquad k_i = -\omega_{n,\alpha\beta}\kappa\sin(\sigma) \tag{3.65}$$

where $\kappa = \left(G'_{\alpha\alpha}(0)|F(j\omega_{n,\alpha\beta})|\right)^{-1}$ and $\sigma = -\pi + \varphi_{m,\alpha\beta} - \angle F(j\omega_{n,\alpha\beta})$ (with $\angle$ denoting the argument of the complex number). The design is simplified by the fact that the low–frequency response of $\mathbf{G}_{\alpha\beta}(s)$ is dominated by the low–pass filter $F(s)$ used for the demodulation (see (3.63)). Therefore, by choosing a sufficiently small control bandwidth, the closed–loop response becomes almost insensitive to variations of both the motor parameters and the system working point as well.

For the experiments, the speed controller has been designed to have a control bandwidth $\omega_n = 220\,rad/s$ (i.e. 35 Hz) and a phase margin $\varphi_m^* = 75°$. Given ω_n and φ_m^*, the poles ω_d are described by a parabola that is function of ω_{me}. Then, it is possible to compute the minimum of ω_d in the operation range and to choose a suitable cut–off frequency. With the above values, the minimum correspond to $\omega_d \approx 130\,rad/s$ therefore the choice:

$$F(s) = \frac{1}{s^2 + 1.732s + 1} \tag{3.66}$$

for the low–pass filter is largely adequate to guarantee that the approximation (3.63) is valid. In fact, the cut–off frequency of the low pass filter is $1\,rad/s$, well below the frequency ω_d.

3.4 EXPERIMENTAL RESULTS

The test bench is described in Appendix A. In order to fully validate the proposed method, the estimated stator resistance was compared with that obtained with a four–wire resistance measurement performed by an Agilent 33410A multimeter. Two kinds of motors were considered, namely a SynR motor and IPM motors, whose parameters are reported in Table A.1. For the sake of generality, two different IPM motors, namely IPM_1 and IPM_2, with different saliencies and stator resistances were considered. It is worth noting that IPM_2 motor resistance is well below $1\,\Omega$. For completeness it is worth remembering how the estimated resistance also includes the resistance of the cables that connect the inverter to the motor. Therefore, for a fair comparison, the resistance measurement carried out with the multimeter was performed on the inverter terminals. After the estimation of the resistance with the proposed method, the drive was switched off and the motor was physically separated from the inverter by means of a very low resistance switch ($< 2\,m\Omega$). The dq current control bandwidth was set equal to 350 Hz, while the speed control bandwidth was set approximatively equal to 35 Hz. The resistance estimation was performed at a constant motor speed of one third and at full rated speed. The injected DC current magnitude was $I_\alpha^* = 0.15\,A$ for the SynR motor and $I_\alpha^* = 0.2\,A$ for both the IPM motors, while $I_\beta^* = 0\,A$ for all the motors.

In order to test the performances of the proposed stator resistance estimation technique, two different tests were carried out. In the former, the stator resistance estimation is enabled by starting from a predefined value. In this way, the ability of the estimation technique to converge to the actual value is validated. In the second test, the stator resistance estimation is always active, but the applied load was changed. The goal of this test was to check the ability of the proposed method to continuously track the stator resistance variations.

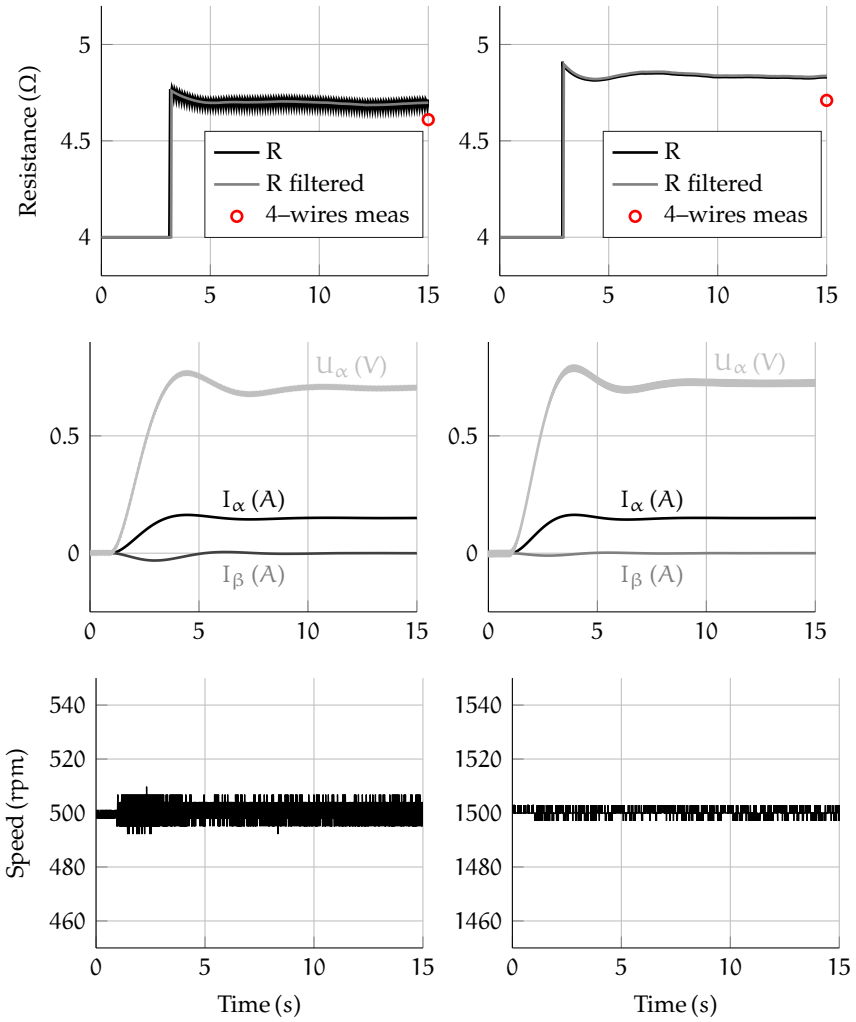


Figure 3.6: Relevant variables dynamics during the estimation of the SynR motor resistance.

3.4.1 *Convergence of the resistance estimate*

The result of the resistance estimation using the SynR motor are reported in Figure 3.6, while the ones obtained using the IPM motors are reported in Figure 3.7 and Figure 3.8. The DC currents δi_α and δi_β reach their set points in about 2 s, while the resistance estimate reaches a constant value after 3 s.

The value of estimated R is affected by a small ripple, due to the filter (3.66) that is unable to completely remove the no constant terms (at the electromechanical frequency ω_{me} and at twice of frequency due to the modulation). Note that with a constant rotation speed of 500 rpm, the electromechanical speed was equal to 16.67 Hz. However, the filter cut–off frequency was set to 1 rad/s. To reduce the oscillation, the estimated resistance can be further filtered by adding a second low–pass filter.

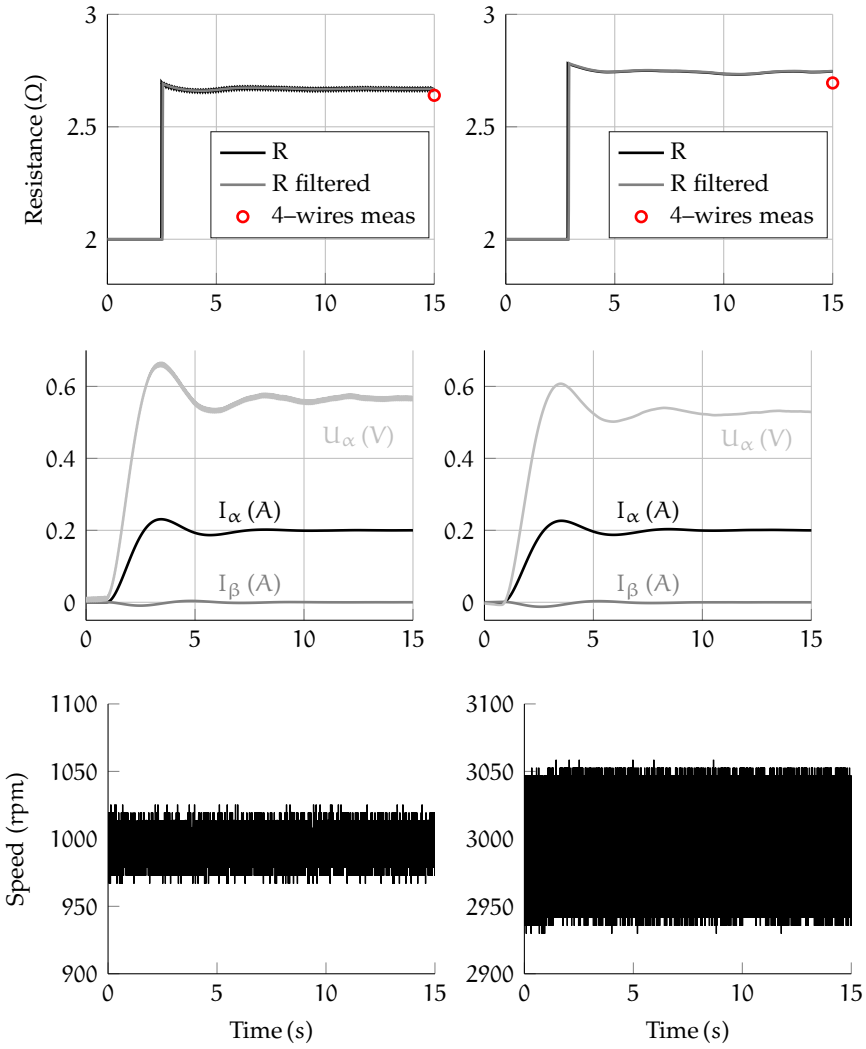


Figure 3.7: Relevant variables dynamics during the estimation of the IPM$_1$ motor resistance.

For validation purposes, the result of the 4–wire measurement was also included in the figures. The worst resistance estimation error was of 2.25% for the SynR motor and of 1.7% for the IPM$_1$ motor.

The waveforms of the injected current and voltage signals in the $\alpha\beta$ frame are also reported in Figure 3.6, 3.7, and 3.8 where the DC injection starts at $t = 1\,s$. As expected, the current δi_β goes to zero after a short initial transient. This assures the best signal–to–noise ratio in the computation of (3.2), for the same amplitude of the injected vector $\mathbf{I}^*_{\alpha\beta}$.

In principle, the dynamics of the DC currents should be function of the motor speed, as shown by (3.63). In practice, in all the tests, the injected currents response is the same, proving that the low pass filter $F(s)$ is dominant in the dynamics of the resistance estimation. Therefore, the technique is quite insensitive to motor parameters.

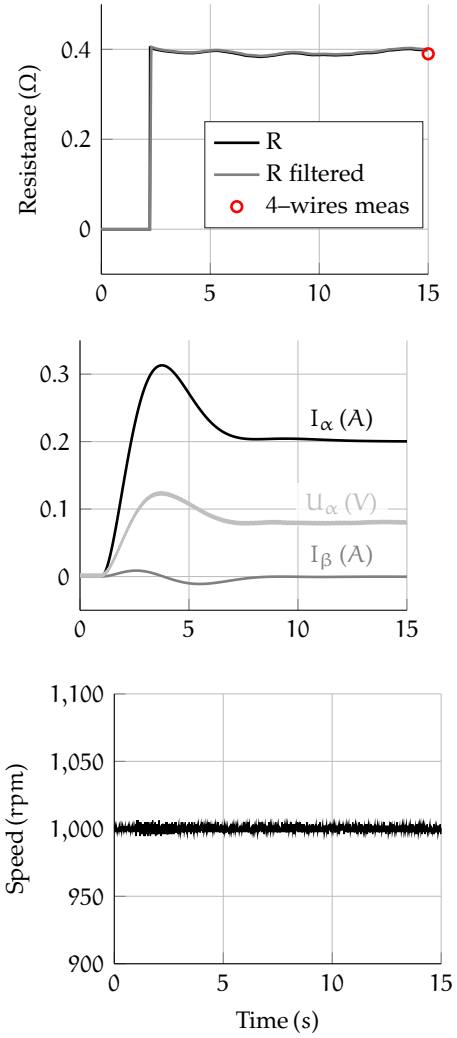


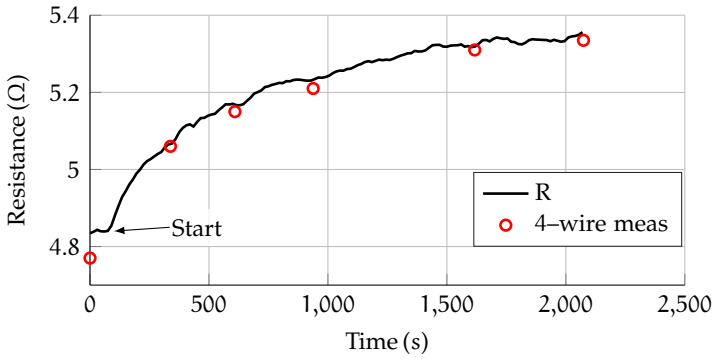
Figure 3.8: Relevant variables dynamics during the estimation of the IPM2 motor resistance.

The motors speed before and after the application of the DC current perturbation is reported at the bottom of the same figures. The speed ripple is less than 2% of the steady–state speed value for the SynR motor at 500 rpm. The higher the speed, the lower the ripple, due to the filtering action of the load.

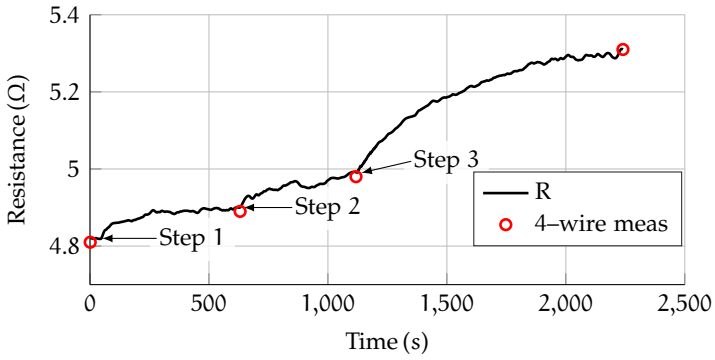
3.4.2 *Tracking of the resistance at variable load*

Since the resistance is a slow–varying quantity in time, it is possible to filter the resistance estimate with a 0.1 rad/s cut–off frequency filter. This allows to get rid of the ripple highlighted in Figure. 3.6 and 3.7, without consequences on the usability of the resistance estimates.

Two different load tests were carried out on each motor. The first test was performed by applying the nominal load to the motor. The resistance estimation was enabled at the beginning of the experiment,

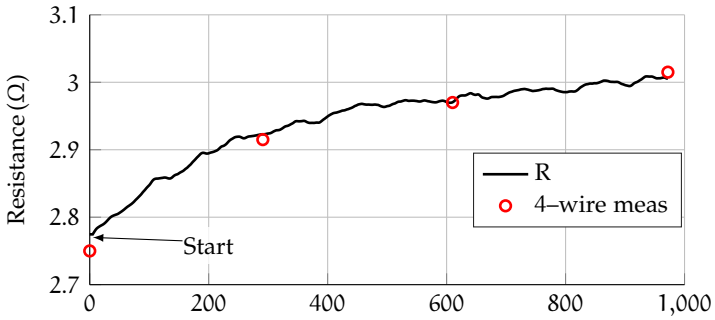


(a) With nominal current applied.

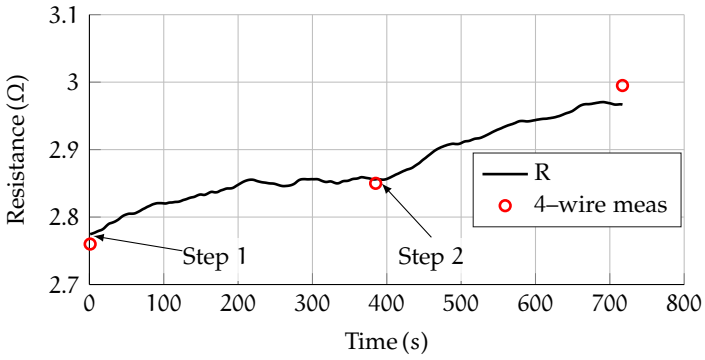


(b) With different load steps.

Figure 3.9: Resistance tracking with different applied loads. The motor under test is a SynR motor.



(a) With nominal current applied.



(b) With different load steps.

Figure 3.10: Resistance tracking with different applied loads. The motor under test is IPM_1 motor.

in order to continuously track the resistance variation. The results are reported in Figure 3.9a and 3.10a for the SynR and IPM_1 motor, respectively. The estimated resistance value (continuous line) is compared with the measurements obtained with the 4–wire method (red circles).

A second test was carried out on both motors, by applying three load steps, corresponding to the 25%, 50% and 100% of the rated load. The resistance estimation technique was always active. The results are reported in Figure 3.9b and 3.10b for the SynR and IPM_1 motor, respectively. As before, the estimate is compared with the measurement obtained with the 4–wire method, showing an impressive matching.

4

OFFLINE MAGNETIC MODELLING USING NEURAL NETWORKS

The rediscovery of SynR motor and PMASRM have encouraged the researchers to develop new approaches able to estimate the magnetic model of a motor accurately. Motors in which torque generation is mainly based on the reluctance principle are characterised by a strongly nonlinear magnetic model. Therefore, the usual magnetic description based on two constant inductances in the rotating reference frame is no longer satisfactory. As described in Sec. 2.1.4, the general electrical model becomes more complex. It can be represented by several lumped parameters in a linearised manner, as (2.52), or with the nonlinear flux–based representation (2.45).

The proper way for modelling the magnetic model of a synchronous motor is the flux–based representation because no linearisation is required. Several methods have already been proposed in literature and they can be classified into different categories, sometimes overlapping. A first important partition is related to the electromechanical speed at which the identification is carried out, either standstill or not. The former group is composed by [36–38] where the flux linkages are computed by voltage integration, taking advantage of the zeroing of the cross–coupling terms. These methods are particularly suitable for an initial identification of the magnetic model in a commercial drive, since they do not require the motion of the shaft. Nevertheless, it is quite challenging the whole magnetic maps identification because it is extremely complicated to explore the whole input region without moving the rotor. As a general remark, the common flaw of standstill methods is that they are well suited for SynR motors, whereas they fail in estimating the PM flux linkage, when present. It is worth noting that in both the cases, the injection of a voltage excitation forces the use of a reduced dc bus voltage, which is not an industry standard. On the other hand, methods belonging to the latter group estimate the magnetic model by evaluating the cross–coupling voltage terms in electrical steady–state condition, namely when the rate of change of flux linkages in (2.45) is zero [39–41]. The estimation carried out with these methods are accurate but they require a test bench or free shaft rotation then they are not recommended for an industrial drive. It is worth remembering how [39] is often taken as a benchmark for comparison in magnetic model identification. The well–known influence of the phase resistance is smoothed by exploiting the flux linkage maps symmetry.

A second subdivision can be carried out on the mathematical model that the methods return. Most of the techniques in literature give a look-up table (LUT), [37, 39, 40]. Therefore, some problems of interpolation and differentiability are posed. Recently, in order to solve the aforementioned problems and to make available continuous models, some novel methods are proposed [36, 38]. The first one suggests a new flux saturation approximating function, which requires the identification of a multiple sets of constants to account for cross–coupling effect. The method works well for self–axis identification, but the whole model is troublesome for its high number of parameters and its moderate accuracy. The latter one proposed a mathematical function for the inverse of the magnetic model, i. e. flux linkages to stator currents, where some exponents of the model have to be empirically chosen.

4.1 THEORETICAL BACKGROUND

The voltage balance equation of a synchronous motor in the rotating reference frame was obtained in (2.45) and reported hereafter for clarity:

$$\mathbf{u}_{dq} = R\mathbf{i}_{dq} + \frac{d\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right)}{dt} + \omega_{me}\mathbf{J}\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right) \tag{4.1}$$

where for this time the dependence of both flux linkages on currents is made explicit. In steady state condition, the equation (4.1) loses its derivative term as follow:

$$\mathbf{u}_{dq} = R\mathbf{i}_{dq} + \omega_{me}\mathbf{J}\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right). \tag{4.2}$$

The proposed identification method recalls a model reference adaptive system (MRAS) in which the adaptive model is represented by a combination of a motor model and a NN. The block schematic of the training procedure is reported in Figure 4.1. The NN is chosen to approximate the inherent complexity of the interaction between motor structure, currents and magnetic fluxes exploiting its universal approximator property. The steady–state measured voltages are compared with the estimated ones in order to obtain two performance indexes which will use to train the NN. The estimated voltages $\widehat{\mathbf{u}}_{dq}$ are obtained as:

$$\widehat{\mathbf{u}}_{dq} = R\mathbf{i}_{dq} + \omega_{me}\mathbf{J}\widehat{\boldsymbol{\lambda}}_{dq}(\mathbf{i}_{dq}) \tag{4.3}$$

where

$$\widehat{\boldsymbol{\lambda}}_{dq}(\mathbf{i}_{dq}) = \begin{bmatrix}\widehat{\lambda}_{d}(\mathbf{i}_{dq}) \\ \widehat{\lambda}_{q}(\mathbf{i}_{dq})\end{bmatrix}. \tag{4.4}$$

is the estimated flux linkages vector that represents the output of the NN. The error $\boldsymbol{\epsilon}_{dq}$, which will lead the training of the NN, is calculated as:

$$\boldsymbol{\epsilon}_{dq} = \mathbf{u}_{dq} - \widehat{\mathbf{u}}_{dq} = \mathbf{u}_{dq} - R\mathbf{i}_{dq} - \omega_{me}\mathbf{J}\widehat{\boldsymbol{\lambda}}_{dq}(\mathbf{i}_{dq}). \tag{4.5}$$

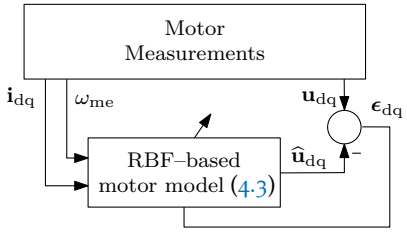


Figure 4.1: RBF network training scheme.

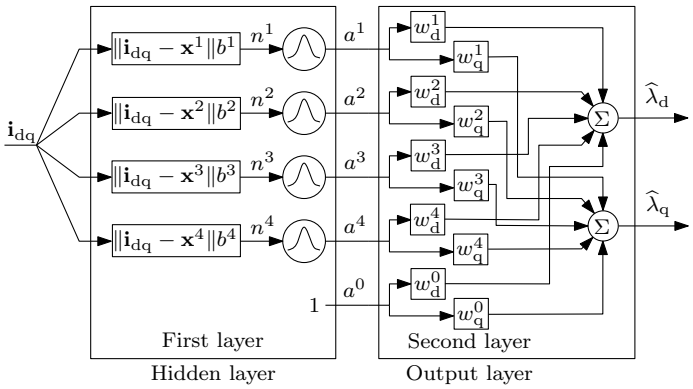


Figure 4.2: RBF network.

Provided that voltages, currents and resistance are known, the zeroing of error $\boldsymbol{\epsilon}_{dq}$ will result in the fulfilment of the equalities:

$$\lambda_d(\mathbf{i}_{dq}) = \widehat{\lambda}_d(\mathbf{i}_{dq}), \qquad \lambda_q(\mathbf{i}_{dq}) = \widehat{\lambda}_q(\mathbf{i}_{dq}) \tag{4.6}$$

namely the estimated linkage fluxes are equal to the actual ones.

4.2 RBF NETWORK FUNDAMENTALS AND DESIGN

For the first time, NNs were proposed in electric drives in the 80s, especially in the area of motor control. They did not draw and convince the academic and industry staff due to training issue, outstanding computational burden and poor results. In particular, the computational power demanded for the training and for the execution was prohibitive for the electronic devices available in those years. For these reasons, the research on NNs has cooled to a halt. In last years, the coming of field programmable gate array (FPGA)s in the electric drive field have pushed a second youth for NNs. FPGAs are suitable for NNs implementation since their parallel architecture fits well with the highly parallel structure of the NNs. The weakening of the technical limits allows the study of new NNs for motor control and other purposes. It is worth highlighting how the studies and the applications based on NNs are having an important growth in other fields as well, like speech recognition, artificial intelligence, image analysis and so on. In electric drives and particularly in model estimation, the NNs have some advantages respect to classical approaches based on LUT,

for instance they give a continuous approximation of the nonlinear magnetic relation including the cross–coupling and they can be easily derived to compute the motor inductances.

In principle, artificial NNs can be considered as universal function approximators. There are several types, that differ by topology and learning process. The choice must be application–oriented. The common NNs are composed by many layers in which the activation functions are usually made with a sigmoid function [42]. It is a *global function* in the sense that its output is not zero even when the net input tends to infinity. The sigmoid functions in the hidden layer create a distributed representation for each input, with the hidden neuron functions collaborating and overlapping. The training and the design of this kind of NN is not trivial due to its inherent structure. The design is usually made with a trial–and–error approach while the training is based on an iterative back–propagation algorithm. Furthermore, the global property of a sigmoid function makes the common NNs unsuitable for a efficient online training and updating. For these reasons, RBF NNs were chosen to estimate the magnetic model of a synchronous motor. The selected NN is always composed by only two layers (more the input layer which connect the input to the hidden layer). The activation functions in the hidden layer are Gaussian functions, in place of the sigmoids ones. These functions present a *local* characteristic, which means that the output of each neuron is close to zero if the inputs are conveniently far from the centre of the Gaussian. Since in a RBF network each Gaussian function is active only around its centre, for any given input only few Gaussian functions will be active at a time. Such local property is interesting in perspective, because it facilitates the development of the online RBF updating algorithm described in Chapter 5. It is worth noting how the local property of the RBF NN helps to understand the basic working principle, aiding the development of the training rules. The two–layer structure of the RBF network can be described with the aid of Figure 4.2. Only four Gaussian functions are drawn in Figure 4.2, but actually many others are present and cover with the due resolution the whole input range.

4.2.1 *First layer – Hidden layer*

As indicated in Figure 4.2, the input to the RBF network is the phase current vector $\mathbf{i}_{dq}$, defined within a circular region in the $\mathbb{R}^2$ plane (Figure 4.3). For an easier implementation, the RBF network training region was extended to the square that circumscribes the circumference. Each dot in Figure 4.3 represents the centre of a Gaussian function. The layout of the centres is a degree of freedom in the design of the RBF network. The local property of the RBF NN entails that the hidden neurons have to be spread over the whole input region to get a proper flux linkages estimation wherever. A comprehensive analysis is given

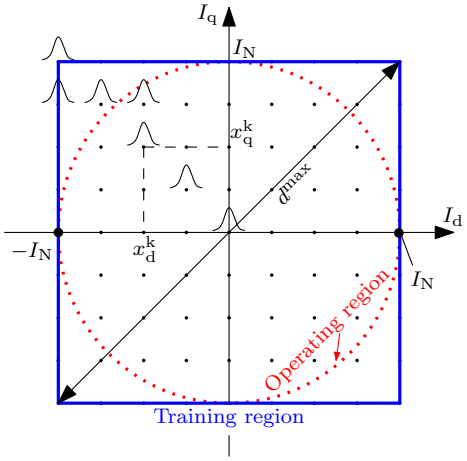


Figure 4.3: Gaussian centres (dots) in the $\mathbb{R}^2$ input plane.

in [43], that proposes either a fixed layout, with random positioning of the centres, or adaptive RBF centres. The conclusion is that the nonlinear optimisation of the first layer parameters is beneficial only when a minimal network is required to solve a given problem. This is achieved at the cost of an overall increased complexity of the system. Furthermore, it is worth highlighting that if the positioning of the Gaussian functions is let to the training algorithm, the training procedure becomes a nonlinear problem, i. e. complex to be solved.

In order to simplify the training and to solve the positioning of the Gaussian function, it has been found that a regular spacing of the RBF centres is a satisfactory solution. The lower the number of Gaussian functions, the lighter the implementation and less the estimation accuracy. A good trade–off consists in spacing the Gaussian centres of about the 25% of the nominal current, in both axes directions. The resulting number of Gaussian function is $K = N_g \times N_g$ functions, with $N_g = 9$. Once the centres of each Gaussian function are defined, the wideness of them have to be determined. It is a crucial parameter because the Gaussian functions has to be partially overlapped by allowing a proper flux linkage estimation but not excessively to preserve the local property. The width of each Gaussian function is inversely proportional to the parameter b^k (Figure 4.2) and a good guess for it is suggested in [42]:

$$b^k = \frac{\sqrt{K}}{2d^{max}} \tag{4.7}$$

where d^{max} is the maximum distance as defined in Figure 4.3.

The hidden layer computes the Euclidean distance between each Gaussian centre $\mathbf{x}^k = [x_d^k\, x_q^k]^T$ and the input vector $\mathbf{i}_{dq}$. The result is then multiplied by the coefficients b^k and sent as input to the Gaussian functions:

$$n^k = \left\| \mathbf{i}_{dq} - \mathbf{x}^k \right\| b^k \qquad k = 1 \dots K. \tag{4.8}$$

The hidden layer Gaussian outputs are:

$$a^k = e^{-(n^k)^2} \qquad k = 1 \dots K. \tag{4.9}$$

It is worth remembering that the first layer is completed by an unitary neuron. It acts as a bias in the NN and it is modulated by weights in the output layer. Being the only first layer neuron without the local property, the associated second layer weights are set to zero throughout all the thesis.

4.2.2 *Second layer – Output Layer*

The second layer is the conventional one adopted in most of NNs, i. e. a linear one. It is composed by two neurons, since two flux linkages have to be estimated. The estimated flux linkages vector $\widehat{\boldsymbol{\lambda}}_{dq}$ can be expressed as linear combinations of the first layer outputs, weighted by the coefficients $w^k_{d,q}$ (Figure 4.2):

$$\begin{aligned} \widehat{\lambda}_d &= \sum_{k=1}^{K} w^k_d a^k = \mathbf{w}^T_d \mathbf{a} \\ \widehat{\lambda}_q &= \sum_{k=1}^{K} w^k_q a^k = \mathbf{w}^T_q \mathbf{a} \end{aligned} \tag{4.10}$$

where $\mathbf{w}_d = [w^1_d\, w^2_d \dots w^K_d]^T$ and $\mathbf{w}_q = [w^1_q\, w^2_q \dots w^K_q]^T$ are the second layer weights vectors and $\mathbf{a} = [a^1\, a^2 \dots a^K]^T$ is the vector of Gaussian outputs. The RBF weights $w^k_{d,q}$ will be the object of the training procedure described in Sec. 4.3.

4.3 RBF NETWORK TRAINING

The RBF training is a linear problem since all first layer parameters are already set, i. e. the centre and the width of each Gaussian function. Although the training problem is linear, the proposed NN is still able to approximate nonlinear functions because it is still composed by a nonlinear layer. The proposed RBF training procedure consists of two steps. It starts by the acquisition of voltages, currents and speed measurements in multiple steady–state conditions, followed by an offline training. A lighter procedure suitable for an online implementation will be developed in Chapter 5.

4.3.1 *Data acquisition*

In order to minimise the errors due to temperature variations, the motor is first warmed up to the thermal equilibrium. Two current references, namely i^*_d and i^*_q, are imposed to the motor under test (MUT), which is driven at fixed speed by a second independent motor

unit. The speed reference comes out as a design trade–off. On one hand, the speed should be low to reduce the influence of iron losses, which drain part of the stator current deputed to the flux production [44]. On the other hand, there is the request of good signal–to–noise ratio for the voltage measurement, which gets better at higher speed. The experimental RBF training was performed at 100 rpm, as described in Sec. 4.4.

It is worth mentioning that the acquisition phase is crucial from the point of view of the estimation accuracy and training on NN. In order to get a well conditioned problem, acquired data have to be spread over the whole dq plane and in an appropriate quantity, i. e. in comparable or greater number respect to the Gaussian functions N_g. Therefore, the training region in Figure 4.3 was gridded to get current references equally spaced–out from each other. The total number of successive steady–state training points is:

$$M = \left(\frac{2I_N}{\Delta I} + 1\right)^2 \tag{4.11}$$

where $\Delta I = I_N/10$ is the result of a design trade–off between resolution and data storage space. For each of the M steady–state working points, the voltages, currents and speed were sampled every T_s and averaged over a complete mechanical revolution. This smooths possible disturbances occurring at either the electrical or mechanical frequency, caused by mechanical and winding asymmetries.

At the end of the first step, M reliable samples of the vectors $\mathbf{u}_{dq}, \mathbf{i}_{dq}$ and the speed ω_{me} are available for the subsequent training of the RBF network.

4.3.2 *Offline network training*

Two different sets of weights $\mathbf{w}_d$ and $\mathbf{w}_q$ are used for the flux linkages estimate, as shown in Figure 4.2. During the training, they were iteratively adjusted by means of the Levenberg–Marquardt algorithm (LMA), chosen for its documented property of fast convergence, even from a rather wrong initial guess [42]. It is worth noting that different algorithms can be chosen since the training problem is linear. Essentially, LMA is a damped least–squares method that calculates the weights with the goal of minimising two quadratic cost functions defined as:

$$\begin{aligned} E_d(\mathbf{w}_q) &= \frac{1}{2}\sum_{i=1}^{M}(\epsilon_d^i)^2 = \frac{1}{2}\boldsymbol{\epsilon}_d^T\boldsymbol{\epsilon}_d \\ E_q(\mathbf{w}_d) &= \frac{1}{2}\sum_{i=1}^{M}(\epsilon_q^i)^2 = \frac{1}{2}\boldsymbol{\epsilon}_q^T\boldsymbol{\epsilon}_q. \end{aligned} \tag{4.12}$$

As shown in Figure 4.1, the inputs to the LMA are the voltage estimation errors $\boldsymbol{\epsilon}_d = [\epsilon_d^1\ \epsilon_d^2 \dots \epsilon_d^M]^T$ and $\boldsymbol{\epsilon}_q = [\epsilon_q^1\ \epsilon_q^2 \dots \epsilon_q^M]^T$. These errors

were obtained from the measurements, according to (4.5). It is worth noting that the cost function E_d depends on $\widehat{\lambda}_q$, due to the motional cross–coupling terms in (4.5). Therefore, E_d is function of $\mathbf{w}_q$ and vice versa for E_q. This cross–dependence has been made explicit in (4.12).

Assuming that $\mathbf{w}_d^h$ and $\mathbf{w}_q^h$ are the weight sets that minimise the cost functions E_d and E_q at the h–th iteration of the LMA, let the Jacobian matrix $\mathbf{J}_d(\mathbf{w}_d^h)$ of the vector–valued function $\boldsymbol{\epsilon}_q(\mathbf{w}_d^h)$ be defined as:

$$\mathbf{J}_d(\mathbf{w}_d^h) = \begin{bmatrix} \dfrac{\partial \epsilon_q^1(\mathbf{w}_d^h)}{\partial w_d^1} & \dfrac{\partial \epsilon_q^1(\mathbf{w}_d^h)}{\partial w_d^2} & \cdots & \dfrac{\partial \epsilon_q^1(\mathbf{w}_d^h)}{\partial w_d^K} \\ \dfrac{\partial \epsilon_q^2(\mathbf{w}_d^h)}{\partial w_d^1} & \dfrac{\partial \epsilon_q^2(\mathbf{w}_d^h)}{\partial w_d^2} & \cdots & \dfrac{\partial \epsilon_q^2(\mathbf{w}_d^h)}{\partial w_d^K} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial \epsilon_q^M(\mathbf{w}_d^h)}{\partial w_d^1} & \dfrac{\partial \epsilon_q^M(\mathbf{w}_d^h)}{\partial w_d^2} & \cdots & \dfrac{\partial \epsilon_q^M(\mathbf{w}_d^h)}{\partial w_d^K} \end{bmatrix} \tag{4.13}$$

A similar definition holds for $\mathbf{J}_q(\mathbf{w}_q^h)$. The M current vector measurements are proposed in sequence to the network input, and the related estimation errors are computed accordingly. Then the LMA weights updating laws are applied:

$$\begin{aligned} \mathbf{w}_d^{h+1} &= \mathbf{w}_d^h - \left[\mathbf{J}_d^T \mathbf{J}_d + \mu^h \mathbf{I}\right]^{-1} \mathbf{J}_d^T \boldsymbol{\epsilon}_q^h \\ \mathbf{w}_q^{h+1} &= \mathbf{w}_q^h - \left[\mathbf{J}_q^T \mathbf{J}_q + \mu^h \mathbf{I}\right]^{-1} \mathbf{J}_q^T \boldsymbol{\epsilon}_d^h. \end{aligned} \tag{4.14}$$

As a distinctive feature of the LMA, the coefficient μ^h is added to make the two matrices $[\mathbf{J}_{d,q}^T \mathbf{J}_{d,q} + \mu^h \mathbf{I}]$ certainly invertible. Provided that the data set is well conditioned (i. e. the measurements are well spaced in the training region), the coefficient μ^h can be set very close to zero, speeding up the training process. At least, when $\mu^h = 0$, the search of the optimal weights vectors is performed in a single iteration only.

Each element of the matrix (4.13) can be made explicit by using (4.5). For example, for the first element it holds:

$$\frac{\partial \epsilon_q^1(\mathbf{w}_d^h)}{\partial w_d^{h,1}} = \frac{\partial (u_q^1 - R i_q^1 - \omega_{me}^1 \widehat{\lambda}_d^1(\mathbf{w}_d^h))}{\partial w_d^{h,1}}. \tag{4.15}$$

It is worth noting that the stator resistance R is supposed to be known, and u_q^1, i_q^1 and ω_{me}^1 are measurements, then independent from $w_d^{h,1}$. With reference to (4.10) and the symbols used in Figure 4.2, equation (4.15) is simplified as follows:

$$\frac{\partial \epsilon_q^1(\mathbf{w}_d^h)}{\partial w_d^{h,1}} = -\omega_{me}^1 \frac{\partial \widehat{\lambda}_d^1(\mathbf{w}_d^h)}{\partial w_d^{h,1}} = -\omega_{me}^1 \underbrace{\frac{\partial \left(\sum_{k=1}^{K} a^k w_d^{h,k} \right)}{\partial w_d^{h,1}}}_{a^1} \tag{4.16}$$

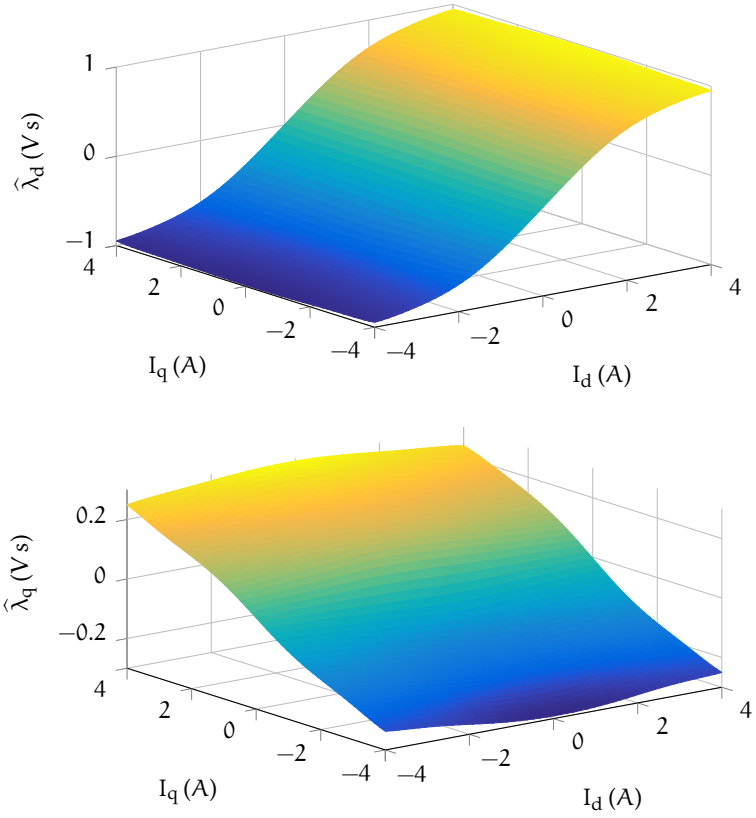


Figure 4.4: 3D magnetic maps of the SynR motor.

where the indexes k and h indicate the evaluated Gaussian function and the evaluated performance index, respectively. In the light of (4.16), the computation of $\mathbf{J}_d(\mathbf{w}_d^h)$ and $\mathbf{J}_q(\mathbf{w}_q^h)$ is trivial and does not require the calculation of new variables, since the a^k terms are already available (because they are used to compute the voltage estimation errors ϵ_{dq}).

4.4 EXPERIMENTAL RESULTS

The experiments were performed on both motors, IPM_1 and SynR, whose parameters are reported in Table A.1. The test bench is described in Appendix A. In order to prove the RBF NN, the MUT was current–controlled and the current references were generated as explained in Sec. 4.3, i. e. forming an equispaced grid in the dq reference plane. The virtual load was speed–controlled at 100 rpm, as motivated in Sec. 4.3.1.

The stator resistance R and the identification of the steady–state condition play a key role to estimate an accurate magnetic model, since they are the fundamental assumptions in (4.12). As mentioned in Chapter 3, the estimation of the stator resistance is not trivial. The use of its nominal value would lead to an imprecise stator flux linkages estimation. In order to prove the proposed magnetic model, two

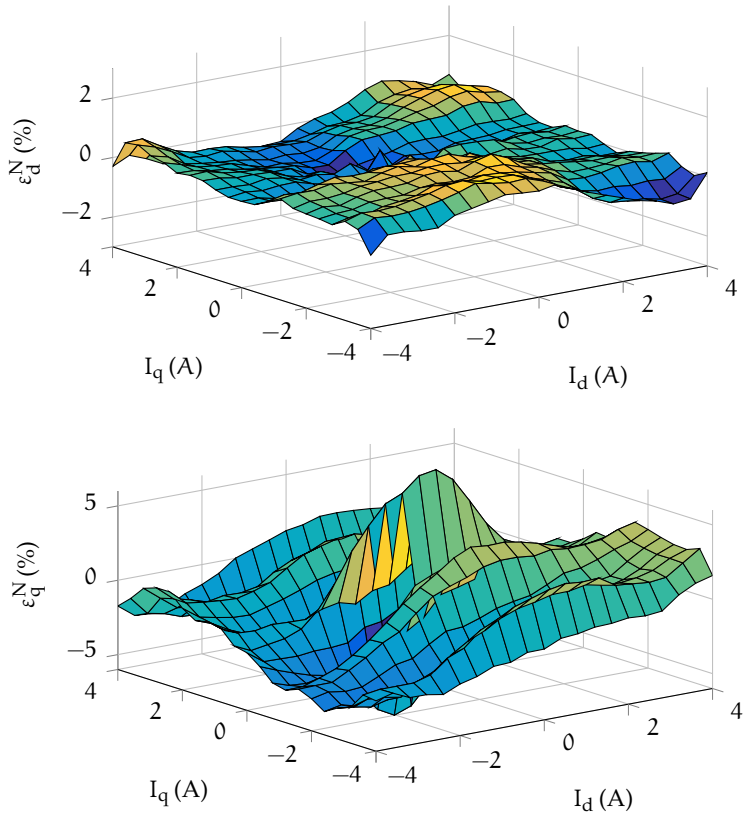


Figure 4.5: SynR motor flux linkages mismatch, proposed method vs. estimate obtained as in [39].

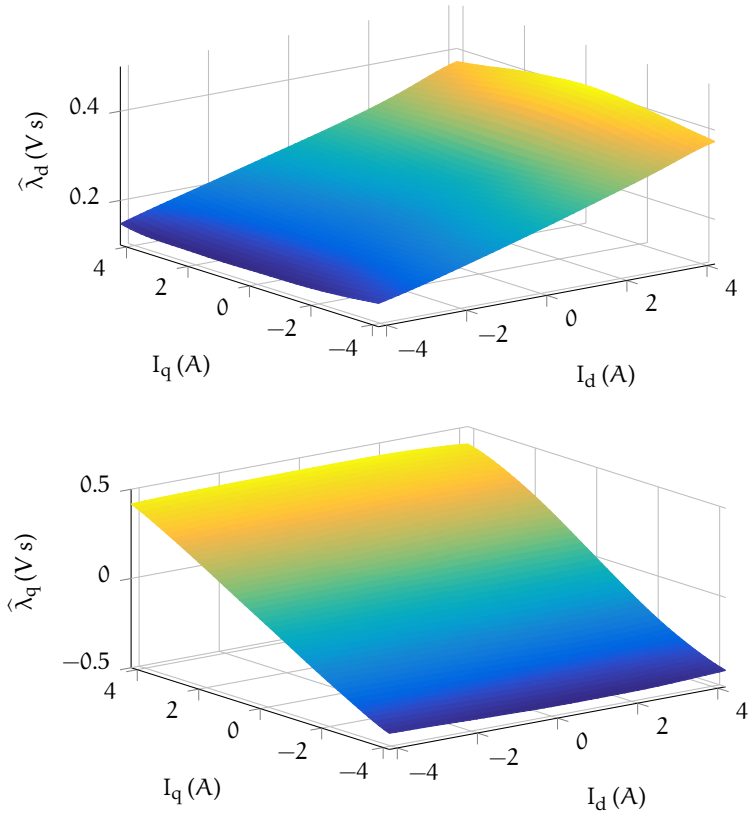


Figure 4.6: 3D magnetic maps of the IPM$_1$ motor.

different approaches can be exploited to estimate the stator resistance. The first one is explained in Chapter 3 while the second one takes advantage of how the test was carried out. Current references were spread over an equispaced grid in dq plane and some of them have the q–current equal to zero. At those references, the resulting steady–state d–voltage equation (4.2) is further simplified, i.e.:

$$u_d = R i_d \tag{4.17}$$

and the resistance can be easily estimated. Finally, the resistance value was estimated before and after the application of some current references and the intermediate values were calculated by means of standard linear interpolation. It is worth remembering that the described method is suitable exclusively for an offline training algorithm. To obtain the resistance estimate it is necessary to have already completed the whole phase of measures. The aforementioned method can be extended to the other axis if the motor lacks of PM.

In order to evaluate the proposed magnetic mapping technique, the results were compared with those obtained by the offline benchmark method [39]. Let the normalised error be defined as:

$$\varepsilon_{dq}^{N} = \frac{\widehat{\boldsymbol{\lambda}}_{dq} - \boldsymbol{\lambda}_{dq}}{\max(\boldsymbol{\lambda}_{dq})} \cdot 100 \tag{4.18}$$

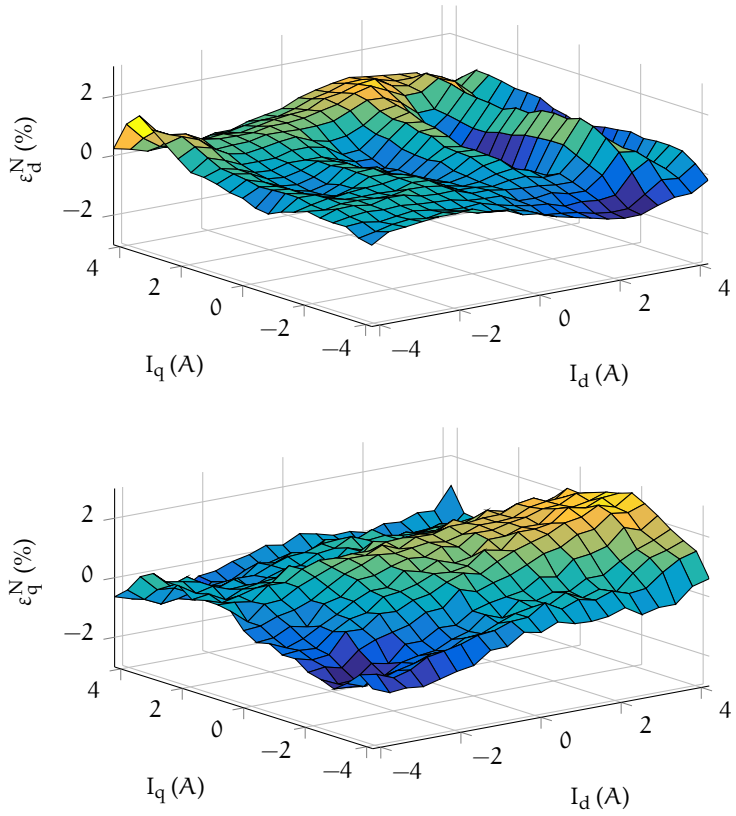


Figure 4.7: IPM$_1$ motor flux linkages mismatch, proposed method vs. estimate obtained as in [39].

where $\widehat{\boldsymbol{\lambda}}_{dq}$ are the RBF network estimates, $\boldsymbol{\lambda}_{dq}$ are the "true" values obtained with the method proposed in [39] and $\max(\boldsymbol{\lambda}_{dq})$ are the two maximum values of flux linkages along each axis.

The magnetic maps of the SynR motor, generated by the first trained RBF network, are reported in Figure 4.4. The normalised errors with respect to the cited benchmark are reported in Figure 4.5. Similarly, the magnetic maps of the IPM_1 motor and the normalised errors are shown in Figure 4.6 and Figure 4.7, respectively. The error magnitude remains almost within $\pm 3\%$ in both cases. In particular, the normalised d–axis error is always within $\pm 1\%$. At very low currents, the q–axis error of the SynR motor increases up to 5%. The higher error ε_q^N around zero q–current is probably due to the inability of the NN to properly estimate the magnetic map. It is not a significant issue since the flux linkages around zero currents are usually poorly exploited. As a countermeasure, a more accurate result could be obtained by increasing the number of neurons (Gaussian functions) in the region of interest. This can always be done, as soon as the regular distribution proposed in Figure 4.3 proves inadequate. Of course, any increase of K or M brings along a not negligible memory consumption and computation time.

For the sake of comparison, the curves obtained with the two extreme cross–coupling conditions, in which the other current is either null, or close to its nominal value, are reported in Figure 4.8. The match between the two models is very accurate in any operating condition. Being linear combinations of Gaussian functions, flux linkages are continuous function of the currents (black lines), while the magnetic maps obtained by the reference method [39] are discrete points (red circles).

For safety reasons, the RBF network was trained up to the nominal motor currents only. To investigate the behaviour of the RBF network outside the training region, a simulation with currents up to twice the nominal value was performed. The results related to the SynR motor are reported in Figure 4.9. As expected, the estimated flux linkages drop quite rapidly as the current exceeds the boundary of the training region Figure 4.3. This tendency is motivated by the local property of the network (Sec. 4.2.1), which ultimately prevents each Gaussian function to contribute to the output only when the input is outside the working region. It means that the proposed RBF NN has no extrapolation skill but it always ensures convergence to zero outside the working region. Of course, the working region (and the training one) can be extended beyond the nominal current values, if requested by the application. In LUT–based models, this is equivalent to enlarge the table size.

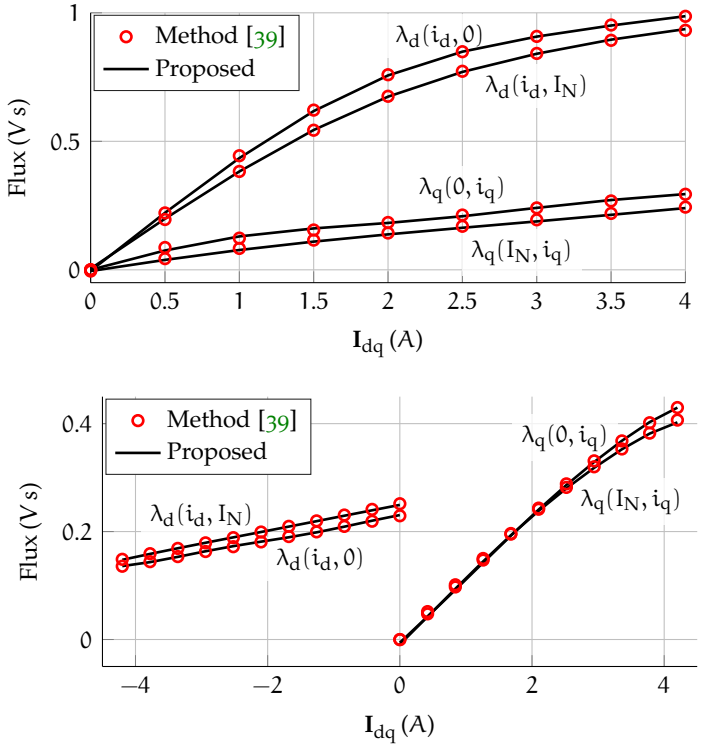


Figure 4.8: 2D magnetic maps (upper: SynR lower: IPM_1). For each axis, the flux linkages are computed either with maximum or without cross–coupling effect.

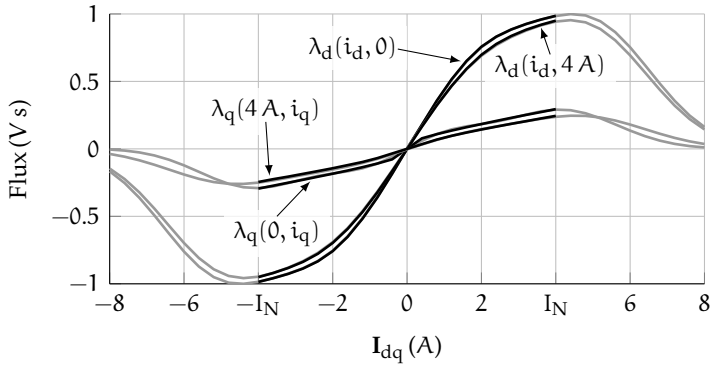


Figure 4.9: Flux linkages estimation up to the nominal current (black) and up to twice the nominal current (grey). The RBF network was trained up to the nominal current value only.

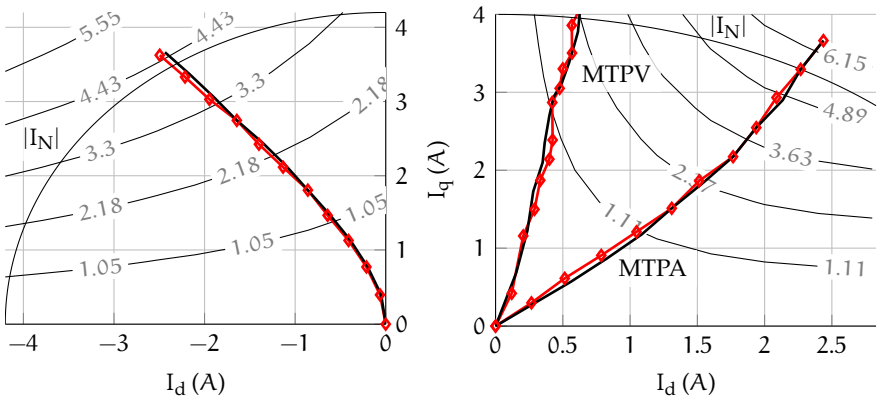
4.4.1 *Assessment of model accuracy through MTPA curves*

A further assessment of the precision of the RBF–based model can be obtained comparing the MTPA and MTPV curves computed from the proposed model and the ones determined with the direct torque measure. The comparison is performed for both motors and the results are reported in Figure 4.10a and Figure 4.10b, respectively. First, the real MTPA curve was obtained by measurements. In particular, the synchronous motor was speed–controlled at constant speed against a variable load torque. For different torque levels ranging from 0 to T_N the phase of the reference current vector was swept to seek the one relative to the minimum–amplitude vector. The result was a collection of MTPA points that can be connected to form the "measured" MTPA curve in Figure 4.10. In the experiment, the shaft torque was measured by a torquemeter. Second, the estimated MTPA curve was computed by exploiting the electromagnetic torque (2.66) and the magnetic model (4.10) obtained by the RBF–based model (Figure 4.4 and Figure 4.6). In order to obtain the estimated MTPA curve from (2.66), two different approaches are feasible. The former is akin to the one used to obtain the curve from the torque measurements while the latter is based on the continuity of the NN and it will be reported in Chapter 6 and both return the same result. The same assessment methodology were used for the comparison of the MTPV curves, Figure 4.10b. It is worth noting that since $\Lambda_{mg}/L_d > I_N$, it was not possible to calculate the MTPV curve for the IPM_1 motor.

Actually, the superposition is almost perfect, testifying the high accuracy of the proposed technique. The slight mismatch between the IPM_1 motor curves at higher currents can be ascribed to the flatness of the torque vs. current surface for that motor. It reduces the sensitivity of MTPA algorithm and may induce some imprecision in the measured curve. Nevertheless, the possible error has little influence on the efficiency since the amplitude difference among the current vector in MTPA and the one actually applied is very small.

4.4.2 *Motor inductances*

The continuity and the differentiability of the flux linkages respect to stator currents is one of the main features of the proposed technique. The computation of the differential inductances is a significant exam-



(a) IPM$_1$ motor (b) SynR motor

Figure 4.10: Assessment of accuracy through the comparison of MTPA and MTPV curves (for the SynR motor, right). Since $\Lambda_{mg}/L_d > I_N$, for the IPM$_1$ motor only the MTPA curve is reported. The —◆— shows the estimated curves and —— for the measured ones.

ple in this sense. They can be obtained by differentiating (4.10) with respect to both i_d and i_q and yield:

$$\begin{aligned} L_d^{diff} &= -2\sum_{k=1}^{K} a^k w_d^k (b^k)^2 (i_d - x_d^k) \\ L_q^{diff} &= -2\sum_{k=1}^{K} a^k w_q^k (b^k)^2 (i_q - x_q^k) \\ L_{dq}^{diff} &= -2\sum_{k=1}^{K} a^k w_d^k (b^k)^2 (i_q - x_q^k) \\ &= -2\sum_{k=1}^{K} a^k w_q^k (b^k)^2 (i_d - x_d^k). \end{aligned} \tag{4.19}$$

The obtained differential inductances of the SynR motor are reported in Figure 4.11. The smoothness of the computed inductances is quite evident. Furthermore, it is noticeable that they change considerably over the whole current input region, due to saturation and cross–coupling of the magnetic flux paths. The inductance maps show the double symmetries along the two dq axes and the experimental cross–differential inductances verify the properties reported in Sec. 2.1.4.1. Conversely, the discrete nature of LUT–based model forces the computation of the difference quotient, which returns the piecewise constant inductances. The results for both motors, IPM$_1$ and SynR, are reported in Figure 4.12. The L_{dq}^{diff} were obtained keeping the current in the other axis equal to zero. The inductances calculated by both methods are consistent with each other, although the NN–based inductances look more real.

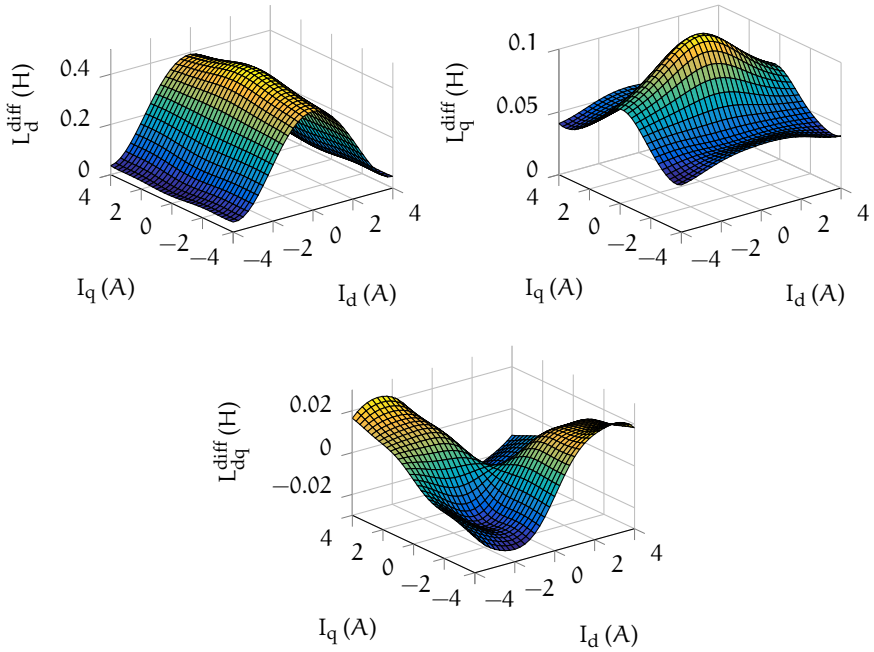
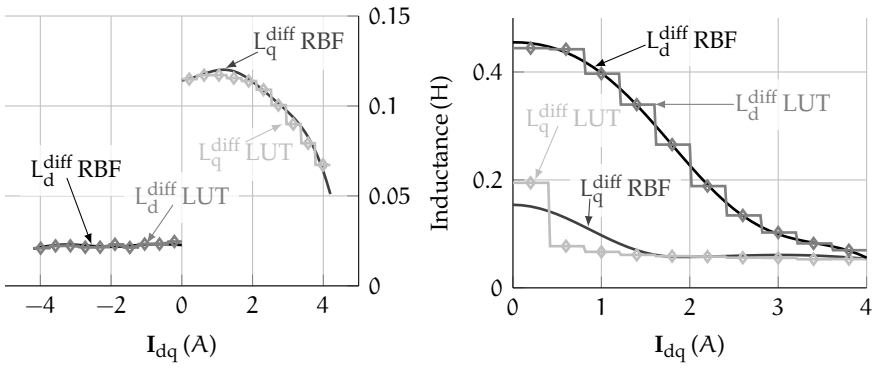


Figure 4.11: Differential and cross–differential inductances of the SynR motor under test.



(a) IPM_1 motor. (b) SynR motor.

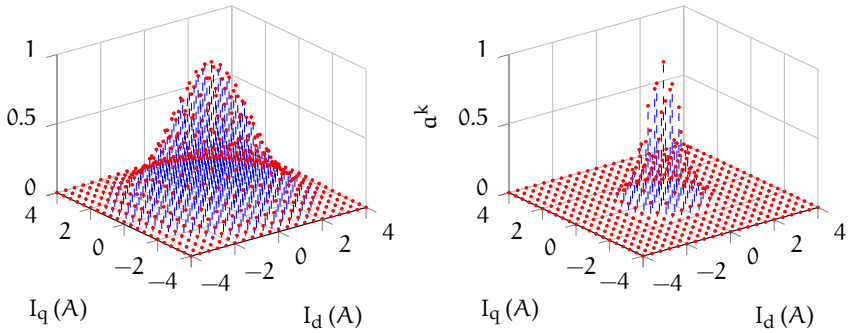
Figure 4.12: Differential inductances as function of their respective axis current (while the current in the other axis is taken equal to zero).

5

ONLINE MAGNETIC MODELLING USING NEURAL NETWORKS

Neural networks have proven to be an excellent tool for identifying the magnetic model of a synchronous motor, with or without PM. The nonlinear behaviour and cross–coupling are properly taken into account by the model introduced in the Chapter 4. In order to train the RBF NN, a two steps offline procedure has to be carried out, as described in Sec. 4.3. It consists in the acquisition of M steady–state samples of voltages, currents and speed and consequently in the training of the network through the LMA. The LMA is computationally cumbersome since multiplications and inversions among matrices are demanded. Therefore, the training approach proposed in Sec. 4.3 can be considered implementable only offline. It is worth reminding how the LMA is a damped least–squares method and it works correctly only if all the data for the model fitting are knowable a–priori. This condition is fulfilled in the approach introduced in Chapter 4 but it never occurs during an online training, where data are gathered instant by instant. The incapability of the LMA to train NNs online establishes a strong limitation since some NN features cannot be exploited. It is worth remembering how RBF NN was chosen due to its local property which makes this kind of NN notably tailored for an online application. Notwithstanding its inherent feature, it is useless without an adequate algorithm. The continuous learning ability is an indispensable feature that high–end intelligent electric drives must have. To be convinced of this, just think about the new services offered by tech giants who learn the habits of users to offer more and more appropriate services and suggestions.

The key features delineation is a fundamental process in the design of new high–end electric drives. Fault detection, self–commissioning, parametric tracking and advanced controls are credible features which have to be developed to be embedded in future electric drives. Learning ability could be as a milestone in the implementation of these aspects, perhaps combined with NNs. In order to make the already proposed RBF NN able to carry out some of the aforementioned features, a new training algorithm has to be developed. It has to demand a low computational effort combined with the ability of working with few data not evenly distributed over the entire input current plane.



(a) RBF with $b^k = 0.5$.

(b) RBF with $b^k = 1$.

Figure 5.1: Local property of the RBF network. First layer output components a^k with $\mathbf{i}_{dq} = \mathbf{0}$ and b^k equal to 0.5 and 1, respectively.

5.1 NEURAL NETWORK STRUCTURE ADJUSTMENTS

The NN structure was described in Chapter 4 with a comprehensive mathematical dissertation. Despite the general behaviour was already depicted, the local property of the RBF NN has to be deeply understood since it will be used in different manners to develop the new algorithm. The local property makes the output of each first layer neuron a^k close to zero whenever the input is far enough from the centre $\mathbf{x}^k$ of the related neuron. Local property can be strengthened or weakened by changing the wideness of the Gaussian function, which depends on the parameter b^k in (4.8). In order to visualise this important property, the Gaussian outputs a^k obtained with two different values of b^k are reported in Figure 5.1. The current input vector $\mathbf{i}_{dq} = \mathbf{0}$ is the same for both figures. Figure 5.1a is computed by a smaller b^k than Figure 5.1b therefore it is characterised by a weakened local property. It is worth highlighting that only the Gaussian functions, namely the first layer neurons, close to the input return a non–zero output. This peculiarity is very useful during the online training, which is performed in a certain steady–state condition and where data of the whole input range are not available. Thanks to the local property, the update action remains limited to the surrounding of the new measured point, without (badly) influencing the other input region. A sort of memory is given to the RBF NN since the update procedure does no alter the weights related to the far neurons. For the sake of completeness, the tighter the Gaussian functions, the higher the number of hidden neurons.

5.1.1 *Approximation of the Gaussian function*

The online implementation of the RBF NN requires the computation of the exponential function (4.9) several times and the task may exceed the computational power of the drive. Therefore, an appropriate numeric approximation was considered.

c_5	$1.06 \cdot 10^{-3}$	c_2	$459.3 \cdot 10^{-3}$
c_4	$17.64 \cdot 10^{-3}$	c_1	$985.9 \cdot 10^{-3}$
c_3	$122.1 \cdot 10^{-3}$	c_0	$999.2 \cdot 10^{-3}$

Table 5.1: Approximation coefficients.

First of all, it is worth noting that the exponents n^k of a^k in (4.8) ideally ranges from 0 (when an input exactly matches a Gaussian centre) to minus infinity, when the actual current measurement is infinitely far from the Gaussian centre. Accordingly, the exponentials a^k range from 1 to 0. In practice, it is reasonable to fix a lower limit ξ:

$$\xi \leqslant a^k \leqslant 1 \;\rightarrow\; \ln(\xi) \leqslant -(\|\mathbf{i}_{dq} - \mathbf{x}^k\| b^k)^2 \leqslant 0 \tag{5.1}$$

so that only the exponentials whose exponent is in the range indicated by (5.1) will be considered for approximation, while the others will be forced to zero. By assuming that the Gaussian functions that are worth less than one hundredth of their maximum value can be neglected, a good value was $\xi = 0.01$ then $\ln(0.01) = -4.61$. Finally, the flux linkages can be estimated with (4.10), reported hereafter for simplicity:

$$\begin{aligned} \widehat{\lambda}_d &= \sum_{k=1}^{K} w_d^k a^k = \mathbf{w}_d^T \mathbf{a} \\ \widehat{\lambda}_q &= \sum_{k=1}^{K} w_q^k a^k = \mathbf{w}_q^T \mathbf{a}. \end{aligned} \tag{5.2}$$

Flux linkages are computed as a summation of several terms and only the Gaussian functions which return a marginal contribution are completely neglected. The overall estimation error on the flux linkages due to the above approximation is less than 0.01, i. e. ξ, since the Gaussian functions which give a remarkable contribution in (5.2) are fully evaluated.

The generic exponential function was approximated by a 5-th order polynomial function. The coefficients where computed offline by imposing the least mean square error in the range specified by (5.1):

$$e^x = c_5 x^5 + c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0, \qquad (-4.61 \leqslant x \leqslant 0). \tag{5.3}$$

The values of the coefficients are reported in Table 5.1. The choice of the order of the polynomial was a trade–off between the required computational power and the mean square approximation error.

5.1.2 *Design the RBF hidden layer*

In Sec. 4.2.1, the Gaussian functions are uniformly distributed across the entire dq input plane, with a regular spacing. The Gaussian number on one axis N_g was set to 9, as a good trade–off among the density

of neurons and the algorithm complexity. By defining the total number of the first layer neurons $K = N_g \times N_g$, the parameter b^k was imposed equal for all the neurons using (4.7). In the offline version of the training algorithm, the local property was not crucial since the whole magnetic maps was estimated through a minimisation of a cost function in few iterations of the training algorithm. In addition, the complete set of data was available at the beginning of the training procedure. It is worth noting that if only the estimation of the magnetic maps is required, other NN structures are more suitable since better performances can be achieved with less parameters.

In the new version of the training procedure, data were collected in any given steady state working point therefore the NN was iteratively trained till the whole magnetic map is proper identified. Given the iterative nature of the new algorithm, the concept of memory associated to the local property of the NN is essential. Therefore, the parameter b^k becomes the first element to be defined. The choice of b^k can be made so that a Gaussian function gives a negligible contribution (a^k in (4.9)) for current vector whose Euclidean distance from the Gaussian centre exceeds a given constant r, as Figure 5.2 shows. A small value of r accentuates the local property but increases the number of Gaussian functions. It was set to half of the rated current $r = I_N/2$. According to (5.1), the following expression holds:

$$b^k = \frac{\sqrt{-\ln(\xi)}}{r} = \frac{2\sqrt{-\ln(\xi)}}{I_N}. \tag{5.4}$$

Given b^k, the number of Gaussian functions can be computed by exploiting (4.7), where the number K of hidden neurons is linked to b^k and to the side I_N of the squared training region [42], i.e.:

$$\sqrt{K} = 4\sqrt{2} I_N b^k \;\rightarrow\; K = -128 \ln(\xi) \tag{5.5}$$

where the right–hand equality is obtained using (5.4).

In general, the value returned by (5.5) is a real number. It is necessary to approximate it to the nearest perfect square, so that the centre of the Gaussian functions will be disposed evenly and symmetrically over the training region, which is a square with side equal to $2I_N$. When the final number of Gaussian functions is determined, the exact b^k has to be recomputed with (4.7).

By arranging the hidden neurons as described in Sec. 4.2.1 leads to a non–homogeneity in the Gaussian functions distribution. The Gaussian function with its centre in $\mathbf{x}^k = \mathbf{0}$ is surrounded on all sides by other neurons. This is not verified for the Gaussian function centred in a corner of the training region since three of the four encircling quadrants are blank. This difference involves some disequilibriums in flux linkages estimation (4.10) where the summations have different number of elements. Therefore, the training algorithm will suffer of some disequilibrium issues as well. In order to reduce this non–homogeneity, the region where the Gaussian functions are arranged is

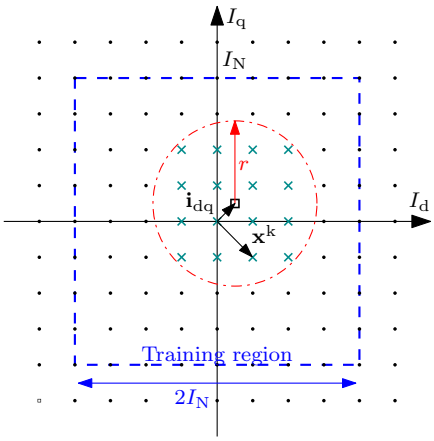


Figure 5.2: Example of mask (dotted–dashed line) in the training region (dashed).

extended outside the training region, as Figure 5.2 depicts. In order to guarantee the same numbers of Gaussian function in (4.10) when the current input vector $\mathbf{i}_{dq}$ changes, each side of the training region is extended of r.

The extension of the region covered by Gaussian functions leads to a considerable boosts in the total number of Gaussian functions. Only the hidden neurons within the training region are 576 against the 81 of the shown case in Chapter 4. The demanded memory is surely increased but the required computational resources can be saved due to the application of the criterion (5.1), as specified hereafter. The Gaussian function approximation described in Sec. 5.1.1 can be further exploited. The inequality (5.1) is a constraint which can be represented as a circle in the dq plane centred in the measured current vector $\mathbf{i}_{dq}$ Figure 5.2. The Gaussian functions characterised by a centre within the circumference return a value greater than ξ, vice–versa the other ones. By exploiting (5.1), the Gaussian outputs smaller than ξ are set to zero therefore they do not influence the flux linkages estimates (5.2). Consequently, the Gaussian functions whose centre is outside by the circle can be neglected from the beginning. It is worth noting how the evaluated hidden neurons are a small part of the total amount, indeed the evaluated Gaussian functions are approximately one hundred.

5.2 THE NEW TRAINING ALGORITHM

The new training algorithm is based on the same scheme described in Chapter 4, namely Figure 4.1. Steady–state condition is still demanded and same performance index vector $\boldsymbol{\epsilon}_{dq}$ (4.5) leads the training of the NN. The zeroing of the performance index entails that the network is properly trained. It is worth pointing out that all the parameters of the first layer, i. e. the position of the Gaussian functions and their wideness, are chosen during the design phase while the RBF weights that

constitute the hearth of the second layer are obtained by the training algorithm. Bearing in mind the reduced computational resources in industrial drives, the training should be selected as computationally light as possible. The new algorithm has to able to handle few data and the ability of zeroing the error vector in one iteration. Although the LMA is the fastest algorithm, it brings along heavy computations and the inability of working with few data then it was discarded.

In order to design a computationally efficient learning algorithm it is worthwhile to note an important consideration which is inferable from (4.9) and (4.5). Any measured current vector $\mathbf{i}_{dq}$ produces a vector $\mathbf{a} = [a^1 \ \dots \ a^K]^T$ of real numbers. Once substituted in (4.5), together with the related voltages, currents and speed measurements, it yields an error function $\boldsymbol{\epsilon}_{dq}$ whose components are a linear combination of the RBF weights $\mathbf{w}_{dq}$. If the NN is properly trained the error vector $\boldsymbol{\epsilon}_{dq}$ would be zero but it is non–zero during the update process. It is possible to define the optimal RBF weights $\widetilde{\mathbf{w}}_{dq}$ which bring to zero the error vector:

$$\mathbf{u}_{dq} - R\mathbf{i}_{dq} - \omega_{me}\mathbf{J}\left(\widetilde{\mathbf{w}}_{dq}^T\mathbf{a}\right) = 0 \tag{5.6}$$

where $\widetilde{\mathbf{w}}_{dq}$ is the weights vector which the training procedure has to be determined. The expression (5.6) is clearly underdetermined, since there are 2K variables (the second layer weights) and just two equations (one for each axis). In order to transform an underdetermined system into a determined one, additional constraints have to be added. Exploiting the local property of the RBF NN, the weights set should be updated aiming at improving the flux linkage estimation in the neighbourhood of the considered input only. In other words, it is more meaningful to modify the weights linked to the Gaussians closest to the input current vector $\mathbf{i}_{dq}$ since they are the ones that contribute mostly in the flux linkages estimation. In addition, by modifying more the functions closest to the input and less the other ones, the concept of memory is created. Without loss of generality, starting from the existing weight vector $\mathbf{w}_{dq}$, one can write:

$$\widetilde{\mathbf{w}}_{dq} = \mathbf{w}_{dq} + \Delta\mathbf{w}_{dq} \tag{5.7}$$

where $\Delta\mathbf{w}_{dq}$ is the unknown vector to be found. In order to implement the concept of memory, the RBF weights have to be modified proportionally to the Gaussian outputs $\mathbf{a}$. Therefore, by imposing:

$$\Delta\mathbf{w}_{dq} = \mathbf{a}\mathbf{W}_{dq}^T = \mathbf{a}\begin{bmatrix} W_d & W_q \end{bmatrix} \tag{5.8}$$

where the two unknown real constants W_d and W_q are the new variables that have to be determined. The lacking constraints in (5.6) are created with (5.8). Substituting (5.7) and (5.8) into (5.6) and by solving the equation for $\mathbf{W}_{dq}$ yields:

$$\mathbf{W}_{dq} = \frac{\mathbf{J}^{-1}\boldsymbol{\epsilon}_{dq}}{\omega_{me}\mathbf{a}^T\mathbf{a}}. \tag{5.9}$$

Replacing (5.9) and (5.8) into (5.7) finalises the proposed training rule for the RBF NN:

$$\widetilde{\mathbf{w}}_{dq} = \mathbf{w}_{dq} + \mathbf{a}\left(\frac{\mathbf{J}^{-1}\boldsymbol{\epsilon}_{dq}}{\omega_{me}\mathbf{a}^{T}\mathbf{a}}\right)^{T}. \tag{5.10}$$

Every time the AC drive enters in a steady–state condition, the voltage error $\boldsymbol{\epsilon}_{dq}$ is averaged on a suitable number of measurements to reduce the possible errors due to spikes and noise, and then the weights of the second layer are updated according to (5.10). It is worth noting that the new training algorithm is capable of working with one measurement at time then more algorithm iterations are needed to cover the whole input region and get the complete magnetic maps.

5.2.1 *Time complexity*

The time complexity is the amount of time it takes to run an algorithm. It is commonly estimated by counting the number of elementary operation performed by the algorithm, supposing that each elementary operation takes a fixed amount of time to perform. Thus, the amount of time taken and the number of elementary operation performed by the algorithm may differ by at most a constant factor [45]. The time complexity is commonly expressed using big O notation. Since this notation excludes coefficients and minor order terms, it is said that temporal complexity is asymptotically described. For example, if the required time for the execution of an algorithm with inputs of dimension n is $5n^3 + 3n$, the time complexity is $O(n^3)$.

The time complexity is a suitable tool to compare computation burden required by training algorithms, the one based on LMA and the newest one. In order to compute the time complexity of both algorithms it is necessary to observe their updating rule (4.14) and (5.10) and calculate the number of elementary operations. From the computation theory, time complexity of matrix algebra is already known, for example:

- matrix multiplication among one $n \times m$ matrix and one $m \times p$ with a resulting $n \times p$ matrix, the complexity is $O(nmp)$;
- matrix inversion of one $n \times n$ matrix, the complexity is $O(n^3)$ with the Gauss–Jordan elimination algorithm while $O(n^{2.373})$ with the optimised CW–like algorithm.

where the assumption that arithmetic with individual elements has complexity O(1) was made. To compute the complexity of the algorithms, the dimensions of vectors and matrices related to the d–axis are reported:

$$\mathbf{J}_d \in \mathbb{R}^{M\times K}, \qquad \boldsymbol{\epsilon}_q \in \mathbb{R}^{M\times 1}, \qquad \mathbf{a} \in \mathbb{R}^{K\times 1}. \tag{5.11}$$

The LMA is composed by the multiplication $\mathbf{J}_d^T \mathbf{J}_d$, an addition among a full matrix and an identity one of dimension K, a K–matrix inversion, a multiplication among a $K \times K$ and $K \times M$ matrices and, finally, a multiplication among a matrix $K \times M$ and the vector $\boldsymbol{\epsilon}_q$. Since the number of collected measurements M is greater than the total number of Gaussians K, the worst operation is the first multiplication $\mathbf{J}_d^T \mathbf{J}_d$ which has a time complexity of $O(K^2M)$. The other operations have a less burdensome complexity then they constitute minor order terms in the time complexity. Since the notation big O neglects them, time complexity of the LMA is $O(K^2M)$.

As regards the new algorithm, the computation of the time complexity is easier since it is mainly composed by 3K operations to compute $\mathbf{a}^T\mathbf{a}$, one multiplication and one division. The total operations number is $3K + 2$, therefore the time complexity is O(K). Even if the evaluated Gaussians with the new algorithm are slightly greater respect to the ones with LMA, the time complexity is substantially smaller.

5.3 EXPERIMENTAL RESULTS

In order to validate the lighter algorithm, two tests were carried out. Analogously to Sec. 4.4, stator resistance has to be known and the use of the nominal value would lead to an imprecise stator flux linkages estimation. In the first test, the resistance was estimated as in Sec. 4.4 while, in the second test, it was estimated with the method proposed in Sec. 3. In both tests, the second layer weights $\mathbf{w}_d$ and $\mathbf{w}_q$ were initialised to zero. This choice represents the worst case scenario and it allows to show that the proposed technique is insensible to the initialised weights values. In order to decrease the estimation time it is possible to initialise them so that the relation among stator currents and estimated flux linkages would be the no–saturated inductances.

Depending on the application, the AC drive may rest in a steady state condition once in a while. In order to enhance the training performances, it is possible to further train the network during transients. The last V steady–state voltages, currents and speed measurements can be saved and kept updated adding the newest ones and discarding the oldest ones. During transients and acquisition time, the NN can be further trained using the stored measurements. In this way, the portions of the sampling times which would not be used due to the lack of new measurements are fully exploited. In addition, since the stored measurements correspond to the last V steady–state points, they were likely collected in several dq working points. An ongoing training performed over V different points which represent a wide dq input area allows to reach better and faster training. In the experimental stage, the buffer length V was of 50 elements.

5.3.1 *Validation of the RBF magnetic model*

The aim of this test was the evaluation of the capability of the proposed method to model the flux linkage map, exactly as in Sec. 4.4. The MUT was current–controlled while the load motor was speed–driven. Only the SynR motor was used as MUT. The test was carried out at a speed of 100 rpm to reduce the possible influences of iron losses, which drain part of the stator current deputed to the flux production [44]. The current references i_d^* and i_q^* were generated as described in Sec. 4.3.1, i. e. they were spread over a grid that covers the whole dq current plane. The grid resolution was fixed to $I_N/10$. Once the motor was at steady–state, the measurements of voltages, currents and speed were collected and averaged over an entire mechanical revolution. This allows the elimination of any signal with electrical or mechanical periodicity. When reliable data were available, the (4.5) was computed and RBF weights were updated with (5.10). It is worth remembering that this test was done only to validate the online algorithm since if a test–bench is available and the whole dq plane can be explored, the best algorithm to be used is the LMA.

The estimated magnetic maps were compared with the estimates made with the benchmark method proposed in [39] which is necessarily an offline, discrete method. As an advantage, the flux estimates in the exact test points are quite reliable and fit for acting as a benchmark. The normalised error defined in (4.18) was used. The estimated magnetic model is reported in Figure 5.3 and the normalised errors are shown in Figure 5.4. It is worth noting that the q–normalised error obtained with the new algorithm and shown in Figure 5.4b is extremely akin to the one computed with the first method and depicted in Figure 4.5b. The plots of both flux linkages, in case of either null or maximum cross–coupling are reported in Figure 5.5, along with normalised errors computed according to (4.18). The error magnitude remains mostly within ±4% for both flux linkages. At low currents, the q–axis error shows values that reach some percent points more. Anyway, the electric drive rarely works at low currents because, even if at no–load, it needs power to keep the motor rotating therefore it rarely needs a flux estimates. Furthermore, at low current the motor is usually not saturated then the MTPA curve is close to the sub–optimal strategy $i_d = i_q$. The error can be still considered quite negligible as can be seen in Figure 5.5 where the absolute values of the flux linkages errors are shown.

Figure 5.6 depicts the measured MTPA curve within the estimated one from the neural network. The estimated curve is close to the one considered as benchmark [39], proving the accuracy of the flux linkages estimates.

It has been proved that the proposed light training algorithm exhibits good performance similar to those of that presented in Chapter 4.

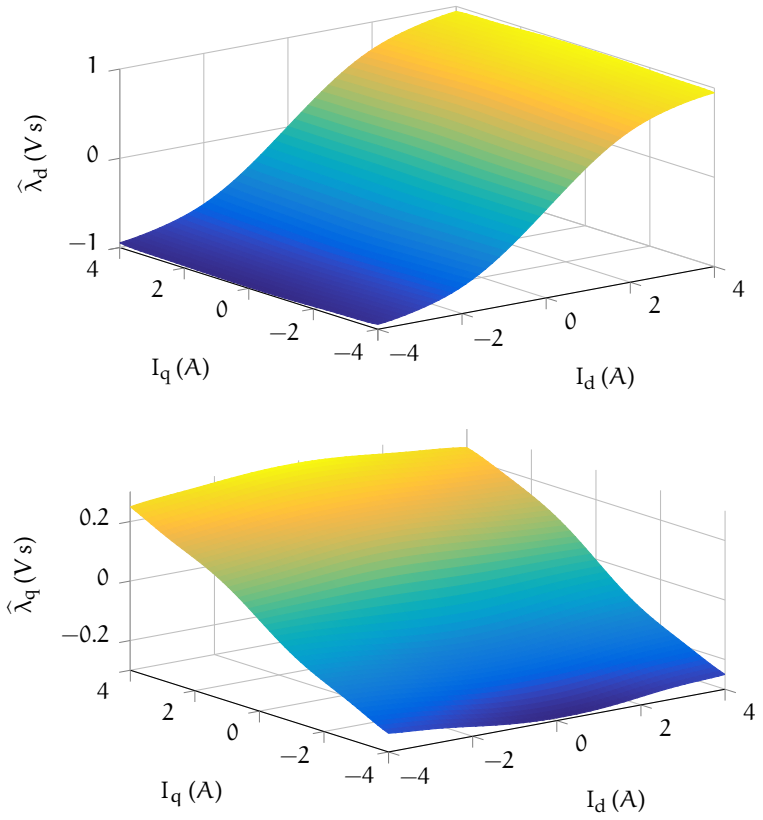


Figure 5.3: Flux linkages estimate with the proposed method for a SynR motor.

The slight deterioration in the accuracy of the online method respect to the LMA is due to the LMA computes a minimisation over the entire dq input plane, and the result is the one that overall minimises the performance index. The online technique minimises the cost function only locally then it loses the optimality.

In order to accelerate the construction of the magnetic maps, one can exploit symmetries and anti–symmetries proper of the maps, see (2.62). The SynR motor magnetic fluxes are anti–symmetric functions of the currents. Therefore, the third quadrant of the dq current plane is actually the same of the first quadrant, but with opposite sign.

5.3.2 *Online RBF training with suboptimal MTPA control*

The previous sections have demonstrated the ability of RBF network coupled with the new learning algorithm to cope with the complexity of the nonlinear magnetic model of SynR motors. The training of the network was performed online, exploiting every single steady–state condition during the normal drive operation.

In this section, the effectiveness of the online algorithm with the RBF NN is tested during the normal working condition of a speed–controlled drive. The MUT was speed–controlled at a constant speed of 350 rpm and a simple sub–optimal ($i_d = i_q$) current control strategy

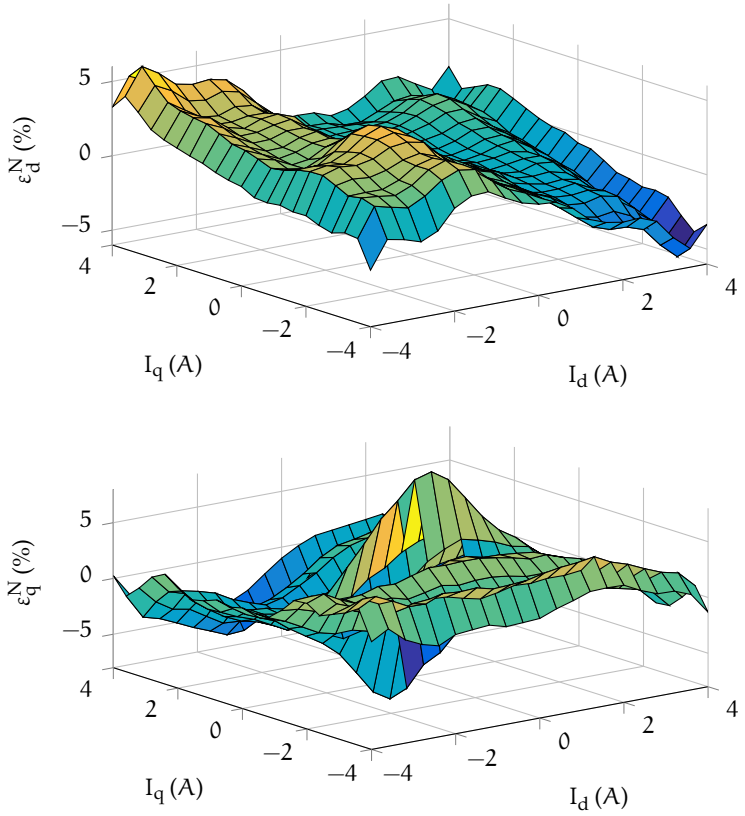


Figure 5.4: Normalised errors between the proposed flux estimation and the compared method.

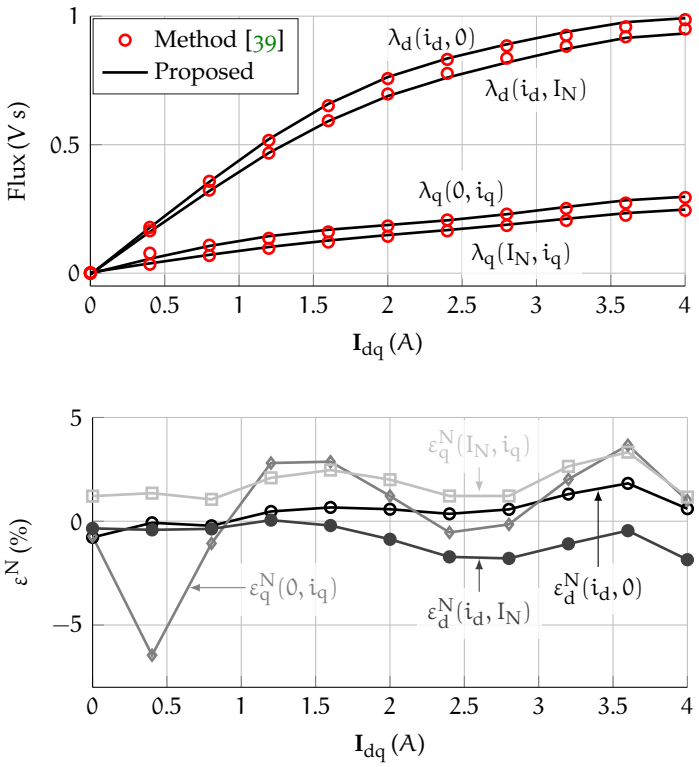


Figure 5.5: 2D magnetic maps and relative errors, using [39] as benchmark.

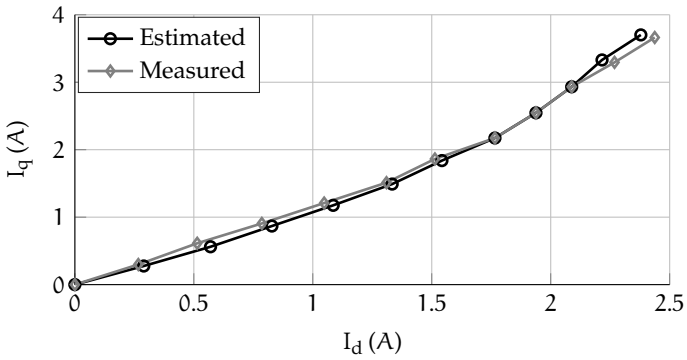


Figure 5.6: MTPA curve.

was selected. The load motor was torque–controlled and it applied six load steps with the same magnitude from 0 to the nominal torque T_N. The experiment procedure consists in:

- the virtual load imposes a load torque step,
- as soon as the system reaches the steady–state, voltages, currents and speed are measured and averaged over an entire mechanical revolution,
- the NN is trained,
- the virtual load applies the following load level or it re–imposes the first one.

The procedure is repeated several time to simulate a real working cycle.

Since both currents are needed to produce torque in a SynR motor, it is not possible to implement the technique previously used to estimate the stator resistance. In order to overcome this issue, the method described in Chapter 3 was exploited. The parameters independence of the resistance identification allows to break the errors chain among different parameters estimation which occurs when the resistance identification is based on knowledge of magnetic maps and vice–versa.

Figure 5.7 shows the magnetic maps of λ_{dq} after the six–step procedure. The local property of the network implies that only the weights close to the input $\mathbf{i}_{dq}$ points are actually updated. As shown in Figure 5.7, the estimated flux linkages are not null closely around the $i_d = i_q$ line. However, the correctness of the estimation rapidly decrease as one deviates from the $i_d = i_q$ line, due to incompleteness of the training for those points. The flux linkages estimation has not any meaning outside the training trajectory. The effectiveness of the magnetic model obtained online was verified by comparing the estimated flux linkages of Figure 5.7 with those obtained by the benchmark method [39] under the same operating conditions, i. e. along the line $0 \leqslant i_d = i_q \leqslant I_N$. The results are reported in Figure 5.8. The errors still

remain within a narrow band of 2%. The normalised errors along the trained trajectory are smaller than the ones obtained in Sec. 5.3.1 and in Chapter 4. The smaller errors can be ascribed to the local property of the algorithm indeed it nullifies the performance index point by point. The Gaussian functions centred outside the $i_d = i_q$ curve contribute to the flux linkages estimation only along the working trajectory without further constraints. This can be well understood in Figure 5.7a where third quarter is negative even if no data were available in that region, the NN was initialised blank and the flux linkages along the explored curve are positive. The algorithm modified the Gaussian functions in the third quarter to better identify the flux linkages along the trained curve, and this explains the setting of those neurons with negative weights.

It is worth pointing out once more that this test was aimed at evaluating the capability of the network to self–adjust online from a completely blank condition. Therefore, only a section of the first quadrant in the dq current plane was explored and properly trained. The remaining domain remains blank as it was initialised, i. e. at zero. Finally, it is worth noting that the RBF NN is able to estimate properly the flux linkages even where no measures are available, as among two load levels.

5.4 FUTURE WORKS

This chapter and the previous one demonstrate the capability of RBF NN to estimate the magnetic map of a synchronous motor. Both the proposed training algorithms assume that the system is in steady state and that the resistance is properly tracked. Both hypotheses should be overcome in order to create an effective and accurate estimation system. The exceeding of the first assumption leads to a cross–coupling among the voltage equations then the identification cannot be carried out for the two axes separately, as it was hitherto. On the other hand, by removing the second assumption a complete different algorithm has to be developed since the error vector ϵ_{dq} cannot be defined anymore. The fall of the two hypotheses represents a major challenge that must be addressed.

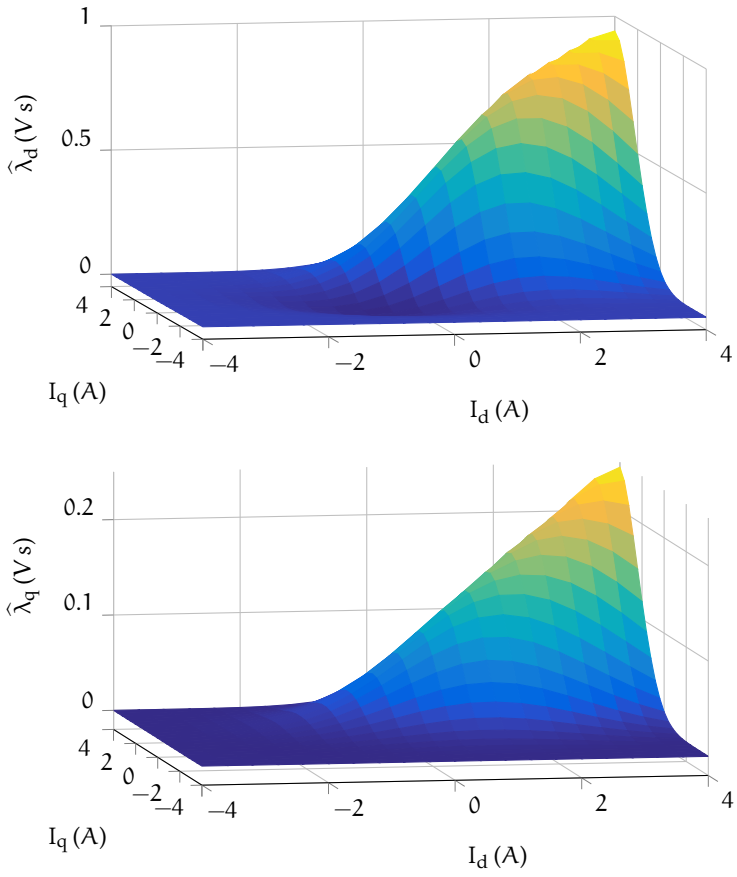


Figure 5.7: RBF–based 3D magnetic maps, obtained by online training from blank conditions and sub–optimal MTPA strategy ($i_d = i_q$).

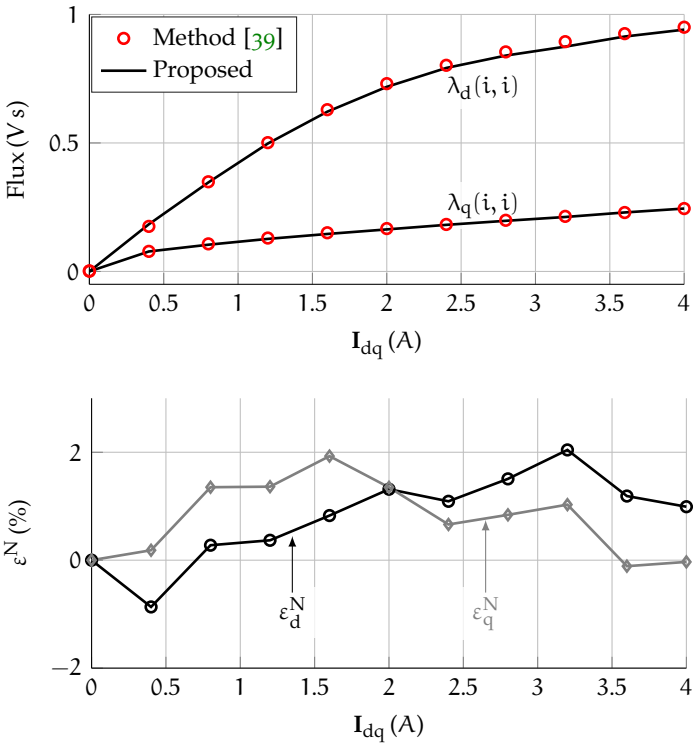


Figure 5.8: Measured and estimated flux linkages along the 45° MTPA trajectory in the i_d–i_q plane. The normalised errors are shown in Figure 5.8b.

Part II

MOTOR CONTROL

6 MTPA TRACKING CONTROL

Modern AC drives have to accomplish two main tasks, i. e. the identification of the motor and its efficient control. An original and innovative identification scheme has been proposed in Part i, able to identify during the normal working conditions, both the stator resistance and the magnetic map of a synchronous motor. The technique for estimating the flux linkages based on a special kind of NN is particularly innovative. Its local property joined a computationally efficient training algorithm enables new control paradigms, based on the availability of a precise motor model.

SynR motors are prone to magnetic saturation and their torque generation, as function of currents, is nonlinear. Both stator currents are demanded to produce torque then the control has an additional degree–of–freedom respect to PMSMs, where the direct current is held to zero to maximise the efficiency. This additional control target can be handled in different ways. One could set both currents equal or leave the direct current to control the flux linkage in the motor and the torque generation entrusted to the quadrature current. Each of these control strategies have pros and cons but they are not efficient, for sure. The preferable control strategy should be the MTPA. It minimises the copper losses for any given torque level but, as shown in [46], the maximum efficiency is not guaranteed.

The are several ways to obtain the MTPA curve for a synchronous motor. The main ones can be partitioned in:

- parameter based,
- finite element analysis (FEA) based,
- model–free based.

The first group relies on a precise knowledge of all machine parameters, usually obtained by several offline tests on a laboratory test rig. The FEA based techniques require the design parameters of the machine, which are not usually available. The latter group is the most interesting one, since it does not depend on the motor parameters. Different techniques have been already proposed, mostly based on perturb–and–observe method [47–49]. The main disadvantage is the detrimental injection which generates an undesired torque ripple.

6.1 MTPA TRACKING ALGORITHM

The MTPA curve can be computed from the complete magnetic model. Unfortunately, different motor and electric drive manufacturers usu-

ally prevent an a–priori tuning. During the commissioning procedure, the electric drive should identify most of the motor parameters to allow it to run. The magnetic model, then the MTPA curve, is one of the most complicated parameters to estimate at standstill, i. e. the preferred condition [21, 22, 36]. For this reason, it is allowed to initialise the electric drive with sub–optimal control strategy and during its operation the control is modified to achieve the desired control targets.

Chapter 5 presented a novel method able to identify online the magnetic map of a synchronous motor. It is based on RBF NN and through a light training algorithm it can be implemented on an industrial drive and can locally estimate flux linkages. Sec. 5.3.2 showed the identified flux linkage of a SynR motor with the proposed method, envisaging the electric drive initialised with a sub–optimal[1] ($i_d = i_q$) control strategy.

In order to achieve a better efficiency of the whole system and having a technique able to estimate the flux linkages along the current control trajectory, a method able to find and track the real MTPA curve can be inferred. For sake of clarity, the main equations of the NN–based model which allow the computation of the flux linkages are reported hereafter from Chapter 4:

$$a^k = e^{-(\left\|\mathbf{i}_{dq}-\mathbf{x}^k\right\| b^k)^2} \qquad k = 1 \dots K \tag{6.1}$$

and

$$\widehat{\lambda}_d = \sum_{k=1}^{K} w_d^k a^k \qquad \widehat{\lambda}_q = \sum_{k=1}^{K} w_q^k a^k. \tag{6.2}$$

The expressions (6.1) and (6.2) clarify that the flux linkages estimates are continuous and derivable functions of the stator currents i_d and i_q. This is peculiar to the NN–based technique and it can be exploited to develop a MTPA tracking algorithm.

For a given reference current vector $\mathbf{i}_{dq}^* = I^* e^{j\beta^*}$ of a fixed amplitude I^* and a variable phase β^*, the maximum torque–per–ampere condition is obtained imposing a null derivative of the torque equation (2.66) with respect to the current angle β^*:

$$\frac{\partial \tau}{\partial \beta^*} = \frac{3}{2} p \frac{\partial \left(\widehat{\lambda}_d^* i_q^* - \widehat{\lambda}_q^* i_d^* \right)}{\partial \beta^*} \stackrel{!}{=} 0. \tag{6.3}$$

1 Assuming a constant inductances SynR motor, the torque equation is $\tau = 3/2p\left((L_d - L_q) i_d i_q\right)$ then the MTPA curve is characterised by ($i_d = i_q$). Having stated this, the aforementioned condition is usually considered as the sub–optimal MTPA control for a SynR motor.

Replacing $i_d^* = I^* \cos(\beta^*)$ and $i_q^* = I^* \sin(\beta^*)$ and by means of the explicit expression of the flux linkages estimation (6.2), the condition (6.3) becomes:

$$\frac{\partial \tau}{\partial \beta^*} = \frac{3}{2} p I^* \left(\widehat{\lambda}_d^* \cos(\beta^*) + \widehat{\lambda}_q^* \sin(\beta^*) + \sum_{k=1}^{K} \left(w_d^k \sin(\beta^*) - w_q^k \cos(\beta^*) \right) \frac{\partial a^k}{\partial \beta^*} \right) \stackrel{!}{=} 0. \tag{6.4}$$

Actually, the a^k terms, defined by (6.1), depend on the stator currents. Using the polar notation, the derivative of a^k respects to the current phase angle is:

$$\frac{\partial a^k}{\partial \beta^*} = 2 a^k (b^k)^2 I^* (x_d^k \sin(\beta^*) - x_q^k \cos(\beta^*)) \tag{6.5}$$

where $\mathbf{x}^k = [x_d^k \, x_q^k]^T$ is centre of the k–th Gaussian function in the (i_d–i_q) plane, see Sec. 4.2.1. The (6.4) is equal to zero when the current vector is along the MTPA curve while it is greater than zero when the MTPA curve has a greater current angle respect to the actual one and vice–versa when (6.4) is negative.

Anyway, the straightforward computation of β^* might be burdensome. The solution of (6.4) is the MTPA angle for a given current module I^* only if the whole magnetic map is identified. The MTPA curve shown in Figure 4.10 can be obtained calculating the (6.4) for different current module I^* but only when the magnetic model is completely known.

Usually, in a real applications the electric drive is speed–controlled and it is initially set with a sub–optimal strategy. With this background, the flux linkages can be estimated using the proposed method only along the control trajectory, as shown in Sec. 5.3.2. Therefore, the (6.4) does not return the correct MTPA angle, due to the limited extent of the magnetic model. Nevertheless, the (6.4) can be used to find the direction of the MTPA curve respect to the actual working point instead of the exact angle. The left–hand term of (6.4) is greater than zero for current phase angles below the correct one, and vice–versa. In addition, the greater the derivative of the torque respect to the current angle, the greater the MTPA angle respect to the current working point.

6.1.1 *MTPA control implementation*

The MTPA tracking algorithm is deeply merged with the flux linkages estimate method, presented in Chapter 5. It returns a meaningful direction toward the real MTPA curve only if the flux linkages are already estimated properly, almost locally. Therefore, while the identification algorithm runs continuously and only when new voltages, currents and speed measurements are collected and the RBF NN is

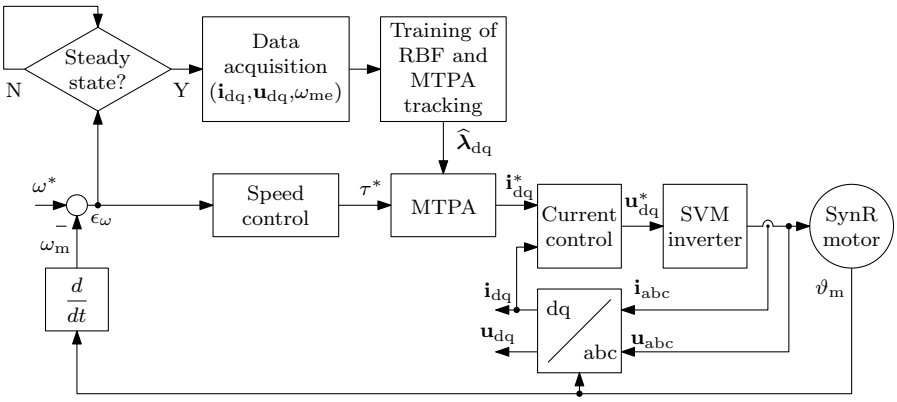


Figure 6.1: Adaptive MTPA control of SynR motor by RBF network.

properly trained, the (6.4) can be evaluated. The procedure can be explained with the aid of Figure 6.1. Without loss of generality, an outer speed control loop is usually present and it is supposed to produce the torque reference τ^*. It is worth noting that (6.4) returns only the direction of the real MTPA respect to the current working point. Therefore, the MTPA curve has to be implemented in a classical manner, i. e. through a LUT or a polynomial function. Once the (6.4) is evaluated, the implemented MTPA has to be updated accordingly. It is worth noting that the online MTPA condition is obtained without any disturbing signal injection, as a key–feature of the proposed method.

6.2 EXPERIMENTAL RESULTS

In order to prove the MTPA tracking algorithm, two tests were carried out. The NN architecture was the same as described in Chapter 5 and the second layer weights $\mathbf{w}_d$ and $\mathbf{w}_q$ were initialised to zero. Furthermore, the electric drive was set with the sub–optimal MTPA, i. e. ($i_d = i_q$). It allows the running of any kind of synchronous motor without any specific knowledge, i. e. the ideal choice for a general purpose electric drive.

6.2.1 *Local convergence*

The first experiment was designed to prove the effectiveness of the proposed strategy and to evaluate the time it takes to find a true MTPA point including the contemporary training of the RBF NN. Any time the MTPA tracking moves the working point closer to the real MTPA curve, a new set of measurements enable an updating of the magnetic model by the RBF NN. It is worth reminding how the MTPA tracker depends on the NN training. The time to find the true MTPA curve strictly depends on the capability of the NN training algorithm to update. The proposed strategy trusts on the motor steady–state conditions then the training time could be not negligible. The test was carried out by applying a

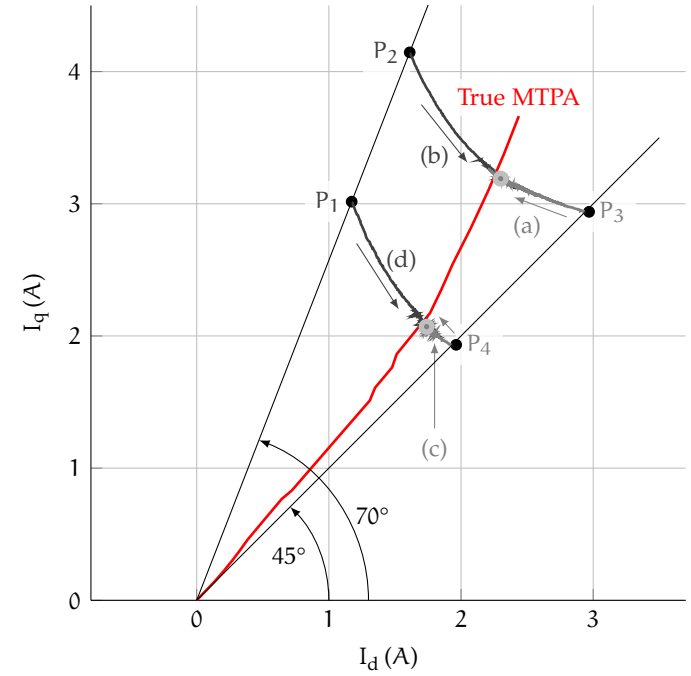
constant load. Then, the adaptive MTPA algorithm was started and the drive reached the true MTPA point.

Figure 6.2a depicts the current path in dq plane while Figure 6.2b shows the current vector magnitude dynamic over the time. The test was repeated four times, with different initial working points ($P_{1,\dots,4}$), changing the load level and the initial MTPA angle. It is evident that the MTPA search is precise and effective, regardless of the initial guess. The stator current $\mathbf{i}_{dq}$ starts from the initialised working point and sweeps the iso–torque curve till to reach the true MTPA point. The operating point moves towards its target without ever exceeding it, as it can be seen in Figure 6.2b since the current vector magnitude is a monotonous decreasing function. Furthermore, the algorithm takes the current vector amplitude to its minimum. In Figure 6.2b, the initial values of the currents corresponding to curves (a) and (b) are slightly above the rated value. However, the final value of the current magnitude, i. e. the MTPA point, remains below the rated value. The current limitation was relaxed during the initial training of the NN to allow the correct identification of the MTPA point at rated load. The convergence time depends on how far the MTPA point is from the initial guess, indeed the convergence time of test P_4 results smaller than the others. Finally, the convergence time can be further reduced by optimising the C–code. This was not implemented because out of the scope of this work and it concerns only the final implementation.

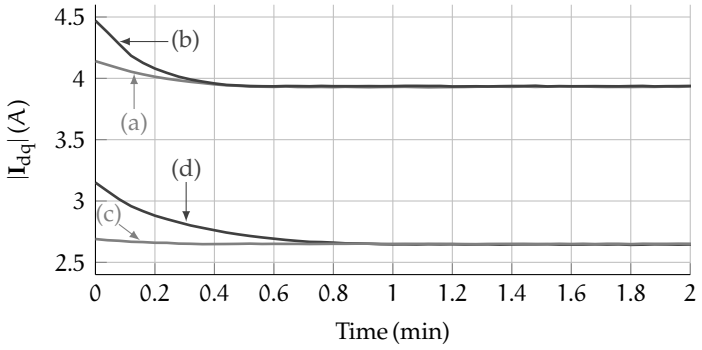
To fully understand the MTPA tracking algorithm, the Figure 6.3 reports the MTPA curve at the end of test P_4. The MTPA curve is as initialised, i. e. a straight line at 45°, except around the point P, where it was modified by the tracking algorithm to get the MTPA condition. Furthermore, at the end of the test the expression (6.4) was evaluated along the whole MTPA curve. It is equal to zero only around the point P and it is greater than zero elsewhere, meaning that the true MTPA locus is characterised by a bigger current angle.

6.2.2 *Global convergence*

The second experiment was to prove the global effectiveness of the (6.4). It was performed by applying five different load torque levels to the SynR motor drive. Whenever the electric drive reached a steady–state condition, voltages, currents and speed were collected. Then, the flux linkages were estimated, the RBF NN was trained and the MTPA curve was modified accordingly with (6.4). The load levels were repeated many times until (6.4) was zero everywhere. The five load points reached by the adaptive algorithm are denoted by diamond–shape marks in Figure 6.4, which reports as a solid line the real MTPA curve as well. The true MTPA was obtained offline by using a torque–meter. The dashed line refers to the initial sub–optimal ($i_d = i_q$) condition. The accuracy is rather good and the estimated MTPA curve



(a) MTPA search trajectories.



(b) Current vector magnitude.

Figure 6.2: Adaptive MTPA transient behaviour.

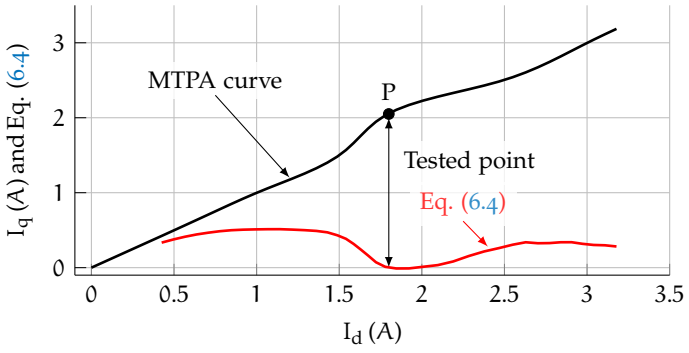


Figure 6.3: Single MTPA point detection and output of (6.4) after the RBF training.

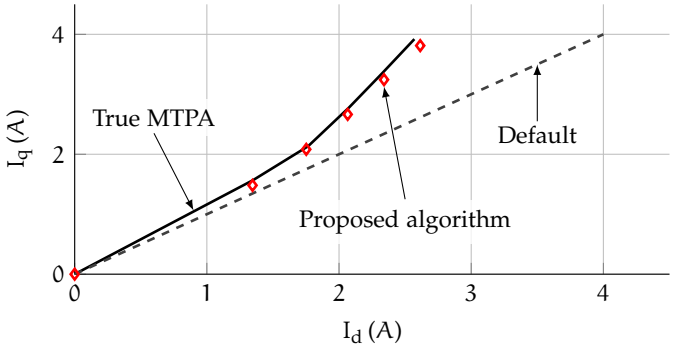


Figure 6.4: Estimated MTPA points. The true MTPA was obtained offline by using a torque–meter.

represented by the diamond marks is close to the real one. Figure 6.5 shows the complete magnetic map at the end of the experiment. It is similar to Figure 5.7 but the ridge of the curve is no longer along $i_d = i_q$ but along another trajectory characterised by a greater current angle.

As a final remark, it is worth noting that the NN training linked to the adaptive MTPA algorithm takes just a small portion of the time in the lifespan of a electric drive. It can be carried out during the very first hours of the entire lifespan. In this scenario, the RBF NN training is a rather cheap price to pay in terms of time. It is worth reminding that the trained second layer weights $\mathbf{w}_d$ and $\mathbf{w}_q$ are always saved and they are reloaded every time the power is turned on. Whatever the NN has learnt, it is not forgotten but it is used as a starting point during the next training. In this, the proposed technique is different from an online MTPA which has to be continuously active to track the MTPA curve.

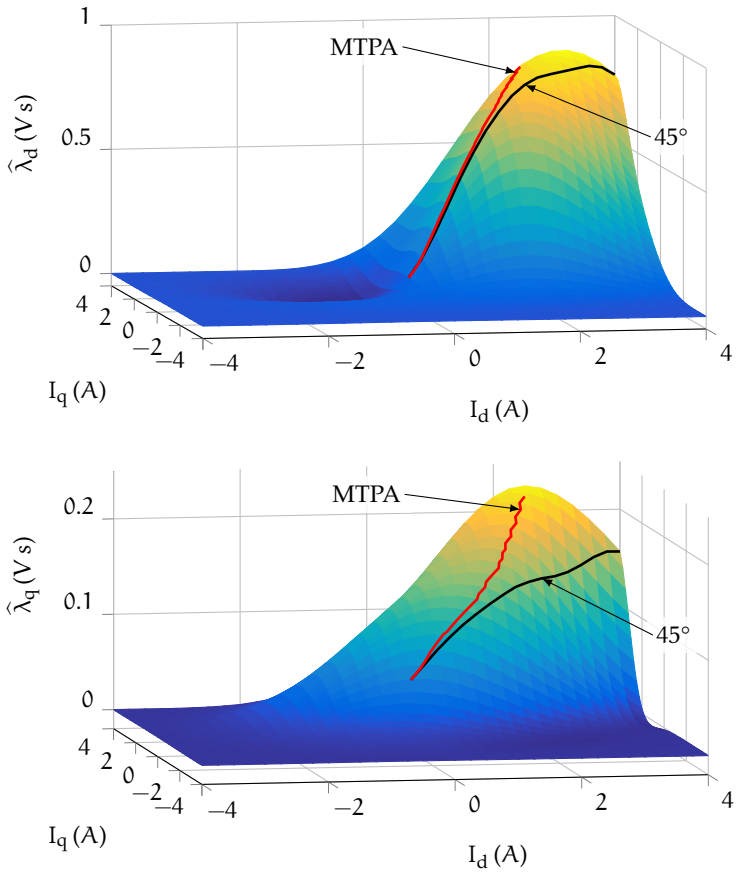


Figure 6.5: 3D flux linkages maps after the convergence of the adaptive MTPA tracking procedure on five different working points. The flux linkages along the $i_d = i_q$ and MTPA trajectories are highlighted.

7 ADVANCED CURRENT CONTROL

Current control is the innermost loop and the heart of high performance electric drives and converters [50–53]. Due to the tight link between current and electromagnetic torque, the current control is responsible for torque ripple and the dynamic response to load changes, both of extreme interest in most of drive applications. It should be able to carry out the following goals:

- handle the magnetic saturation and cross–coupling that characterise the motor,
- compensate the semiconductor voltage drop and dead times of the converter,
- stabilise the controlled system satisfying the given specification.

In the last decades, the research efforts about current control has been quite considerable. Many techniques have been presented with the purpose of increasing the dynamic performances, with particular attention to PMSMs. The proposed control structures can be subdivided in three main categories: predictive, hysteresis and PI control. Every proposed control technique has strengths and week points. For instance, model–based methods can achieve high steady–state and dynamic performances but they are sensible to the model inaccuracy. On the other hand, model–free schemes, like the hysteresis one, suffers of the well–known problem of chattering even at steady–state.

The stability is an essential element for a control scheme, more than the maximum achievable performances. The performances of a system that is weakly or not stable cannot even be calculated. Stability is deeply studied for PI controllers [35] while there is a limited literature for other techniques. PI controls take the lion's share in industrial electric drives due to the broad literature about stability jointly with an (apparent) ease of calibration and the good achievable performances. In addition, their implementation is quite trivial and well–known.

PI controller is a linear control and it works properly with linear system, e. g. with PMSMs. The recent interest in anisotropic motors, e. g. IPM motors, PMASRMs and SynR motors, has posed the issue of the development of control structures able to manage nonlinear system and to obtain good performances, as PIs do with PMSMs. Since nonlinear motors are already an industrial reality[1], it is fundamental to fill this

1 In 1998 an important patent about SynR motor design was published [54]. It has partially limited the development and spread of these motors as companies that wanted to market would have to pay royalties. In the last years, only few companies

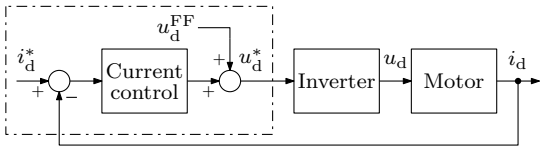


Figure 7.1: Closed–loop system of the d–axis current control.

control gap. Researchers have to quickly develop effective and ease control schemes for rapid technology transfer to the companies. At the same time, more tailored control paradigms have to be examined in depth for electric drives of tomorrow.

7.1 MODEL OF THE SYSTEM AND PI DESIGN

The controlled plant consists in the series of the inverter and electric motor, as sketched in Figure 7.1.

7.1.1 *Motor model*

The mathematical model of a generic synchronous motor, i. e. taking into account magnetic saturation and cross–coupling, in the rotating reference frame was derived in Sec. 2.1.4 and hereafter reported for sake of clarity:

$$\begin{aligned} u_d &= Ri_d + \frac{d\lambda_d\left(\mathbf{i}_{dq}\right)}{dt} - \omega_{me}\lambda_q\left(\mathbf{i}_{dq}\right) \\ u_q &= Ri_q + \frac{d\lambda_q\left(\mathbf{i}_{dq}\right)}{dt} + \omega_{me}\lambda_d\left(\mathbf{i}_{dq}\right). \end{aligned} \tag{7.1}$$

The motional terms $\omega_{me}\lambda_q\left(\mathbf{i}_{dq}\right)$ and $\omega_{me}\lambda_d\left(\mathbf{i}_{dq}\right)$ in (7.1) are considered as a disturbance in the current control loop and they are compensated by means of two FF terms. This is known as *decoupling* and it is almost universally applied in electric drives. Therefore, the motional terms will be omitted and the expression (7.1) reduces to the following:

$$\begin{aligned} u_d' &= Ri_d + \frac{d\lambda_d\left(\mathbf{i}_{dq}\right)}{dt} \\ u_q' &= Ri_q + \frac{d\lambda_q\left(\mathbf{i}_{dq}\right)}{dt}. \end{aligned} \tag{7.2}$$

The symbols u_d' and u_q' are used to emphasise the absence of the motional terms with respect to (7.1).

have sold SynR motors. Nowadays, the patent is expired and the commercial offer of reluctance–based motor is expanding considerably.

In order to highlight the cross–coupling and to derive the control strategies, the linearised version of (7.2) can be derived by recalling (2.50):

$$\begin{aligned} u'_d &= Ri_d + \frac{\partial \lambda_d\left(\mathbf{i}_{dq}\right)}{\partial i_d}\frac{\partial i_d}{\partial t} + \frac{\partial \lambda_d\left(\mathbf{i}_{dq}\right)}{\partial i_q}\frac{\partial i_q}{\partial t} \\ u'_q &= Ri_q + \frac{\partial \lambda_q\left(\mathbf{i}_{dq}\right)}{\partial i_d}\frac{\partial i_d}{\partial t} + \frac{\partial \lambda_q\left(\mathbf{i}_{dq}\right)}{\partial i_q}\frac{\partial i_q}{\partial t} \end{aligned} \tag{7.3}$$

where the flux linkages derivatives over the stator current are the differential inductances defined in (2.51). The structure of equations (7.3) is akin to those of (7.1). In particular, cross–coupling terms are present which consist in motional terms in (7.1) and in voltage drop due to the cross–differential inductances in (7.3). Since they have the same structure, they can be treated analogously. The already introduced FF action is integrated with the cross–differential inductances contribution. The overall FF terms are the following:

$$\begin{aligned} u_d^{FF} &= -\omega_{me}\lambda_q\left(\mathbf{i}_{dq}\right) + \frac{\partial \lambda_d\left(\mathbf{i}_{dq}\right)}{\partial i_q}\frac{\partial i_q}{\partial t} \\ u_q^{FF} &= \omega_{me}\lambda_d\left(\mathbf{i}_{dq}\right) + \frac{\partial \lambda_q\left(\mathbf{i}_{dq}\right)}{\partial i_d}\frac{\partial i_d}{\partial t} \end{aligned} \tag{7.4}$$

Figure 7.1 shows the closed–loop system for the d–axis where the FF action is highlighted. The voltage components in (7.4) related to the cross–differential inductances are functions of the current time derivative which is notoriously difficult to calculate in discrete time system. In order to overcome this issue, the derivative was low–pass filtered to limit the computational noise. However, the filtered derivative introduces an undesired time delay. The effect of the delay jointly to the parametric uncertainties can be partially smoothed by an attenuation gain on the second terms of the right members in the FF action (7.4). It is worth noting that the whole or partial FF action can be omitted, by leaving to the PI controller the effort of compensating the disturbances. Of course, this leads to a detrimental control action during transients.

In order to understand the effect of the cross–differential inductances on the current behaviour, some simulations were carried out with the SynR motor and they are reported in Figure 7.2. Tests were obtained by imposing a current step variation on the q–axis, and observing the behaviour of the measured current on the d–axis. The proposed tests were carried out with the rotor locked to completely disregard the motional terms and applying a q–axis current step from 0 to I_N, i. e. 4 A. The d–axis current reference was set at 4 A. As a consequence of the cross–differential inductances effect, an overshoot of amplitude 250 mA, equal to 6.25%I_N, appeared on the d–axis current. By applying the proposed FF terms, the current overshoot is 50% smaller than of case with the standard PI. The i_d does not exactly follow the reference

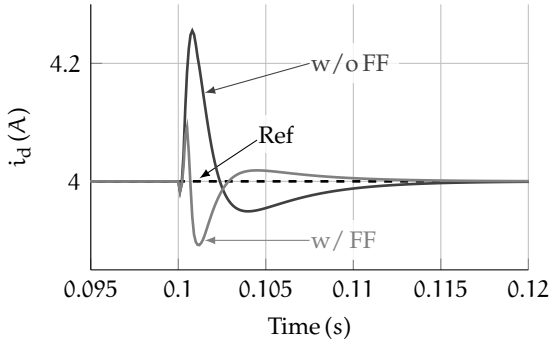


Figure 7.2: d–axis current response with a q–axis current step with and without the proposed FF action. The cross–differential inductances coupling on the d–axis is shown.

probably due to small errors in the magnetic model and the attenuation gain introduced in the filtered current derivative to mitigate the delay. Nevertheless, the FF action improved the overall current behaviour.

By implementing the FF actions, the electric motor can be modelled as a first order system, where the electrical time constant is $\tau_e \triangleq L/R$ and a constant gain equal to $1/R$. Actually, the conventional (and constant) inductance L has to be replaced by L_d^{diff} or L_q^{diff} for the design of the current control along each synchronous coordinate (d,q). In Laplace domain, the transfer function of the electric motor becomes:

$$G_m(s) = \frac{1}{Ls + R} = \frac{1}{R}\frac{1}{1 + s\tau_e} \tag{7.5}$$

7.1.2 *Inverter and controller model*

The digital control and the voltage source inverter (VSI) introduce delays in the control loop. They can be modelled as an unique time delay T_d. The digital control takes some time to process the voltage reference and a time delay elapses between when the voltage reference is given to the VSI and when it is actually applied. Typically, the former is negligible or it is a fraction of T_s when a FPGA with a fast control paradigm is implemented [50] while it is equal to T_s using a digital signal processor (DSP). The latter delay is not trivial to be computed and it is strongly affected by the modulation technique [55]. Using a symmetric pulse width modulation (PWM), the delay is exactly half of T_s. Therefore, the overall control delay is equal to $T_d = 3/2T_s$, since the test bench is based on a DSP with a symmetric PWM, as described in Appendix A. In Laplace domain, the delay can be approximated by a first–order lag system:

$$\mathbf{u}_{\alpha\beta}(s) = e^{-sT_d}\mathbf{u}^*_{\alpha\beta}(s) \approx \frac{1}{1 + sT_d}\mathbf{u}^*_{\alpha\beta}(s). \tag{7.6}$$

Since the control algorithm is implemented in the rotating reference frame, it is worth representing the delay accordingly. As detailed in Appendix D, the equation (7.6) becomes:

$$\mathbf{u}_{dq}(s) = e^{-sT_d}\mathbf{D}\mathbf{u}^*_{dq}(s) \tag{7.7}$$

where $\mathbf{D}$ is a rotating matrix defined as follow:

$$\mathbf{D} = \begin{bmatrix} \cos(\omega_{me}T_d) & \sin(\omega_{me}T_d) \\ -\sin(\omega_{me}T_d) & \cos(\omega_{me}T_d) \end{bmatrix} \tag{7.8}$$

The delay T_d introduces a coupling among the axes and an attenuation, as equation (7.7) shows. These effects should be compensated by the controller since they have a detrimental effects on the current behaviour. In particular, the higher the motor speed, the greater the distortion. In the present chapter, all the simulations and experimental tests were carried out at standstill, so that the matrix $\mathbf{D}$ becomes the identity one and it has no effect.

7.1.3 *Constant PI design*

Engineers are familiars with PI controllers since they are used in a wide range of applications. They are effective and good performances can be attained if the system is linear. It is worth highlighting that the linearity of the model is only a necessary condition, not sufficient, since the system parameters have to be known for an analytical tuning. The PI design can be carried out by the pole–placement method, the magnitude optimum or the root locus and the frequency response criterion [35, 50, 56, 57]. Nevertheless, the design of a stable PI controller in the whole operating region can be challenging when the system is nonlinear. SynR motors are highly nonlinear systems and their differential inductances are prone a factor 10 variation in dependence of the working point, as shown in Figure 4.11. The first, simple solution to control SynR motors is to design the controller by considering the worst case scenario. It is usually called *robust control* and assures the stability of the system. The worst condition has to be found by the designer and it may not be trivial.

Usually, the minimum control bandwidth is one of the control requirements with the stability. In order to design a current regulator able to guarantee a minimum control bandwidth, the tuning would be done with the biggest motor inductance. By decreasing the inductance due to the magnetic saturation, the electric pole shifts rightward in the Bode plane by increasing the system bandwidth so the requirement would always be satisfied. Nevertheless, the phase margin could excessively decrease due to the system delay and stability could be affected.

A robust PI design can be carried out using the smallest differential inductances of the motor. At lighter loads, when the effects of satura-

tion are gradually reduced, the electrical pole shifts leftwards in the Bode plane, by keeping a sufficient phase margin while reducing the cut–off frequency. Therefore, the bandwidth decreases but the stability is preserved.

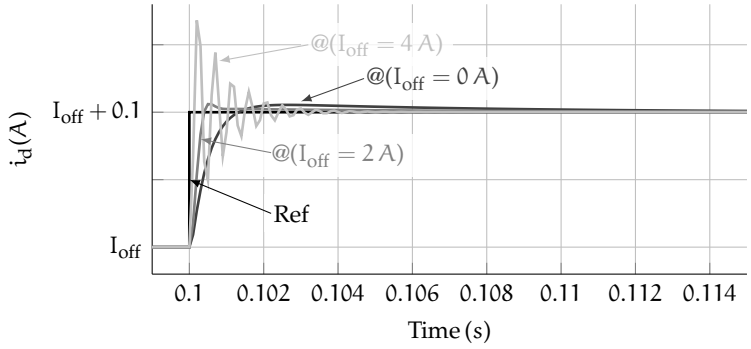
In order to prove the aforementioned claims, several simulation tests were carried out. The current control specifications were a bandwidth of 300 Hz and 70° phase margin. All the tests were carried out at standstill in order to isolate the effects on stator currents due to the PI design from the effects due to motional terms. Many current steps were carried out around different working points. A first group of tests was done with a constant–gain conventional PI controller where it was design by taking the no–load values of the stator inductances. The results are reported in Figure 7.3a. Only the curve with $I_{off} = 0$ A has the desired profile, which is indeed the case where the actual inductance is the same used in the PI design. A completely different situation appears at $I_{off} = 4$ A. The real value of the differential inductance is very different to the one used during the tuning. The electrical pole of the system is placed quite far from the one used in the design. The real pole is at higher frequency, due to the inductance saturation, and the system undergoes an oscillatory response meaning that the controlled system was close to the instability.

The second group of tests is reported in Figure 7.3b where a robust PI regulator was used. In this case, the curves have no oscillatory behaviours, i. e. the stability is always preserved, but only the step at $I_{off} = 4$ A responded with the imposed dynamic. The remaining cases exhibited slower dynamics performances. For instance, the current response step with $I_{off} = 0$ A shows a 70 Hz bandwidth, quite far from the design value.

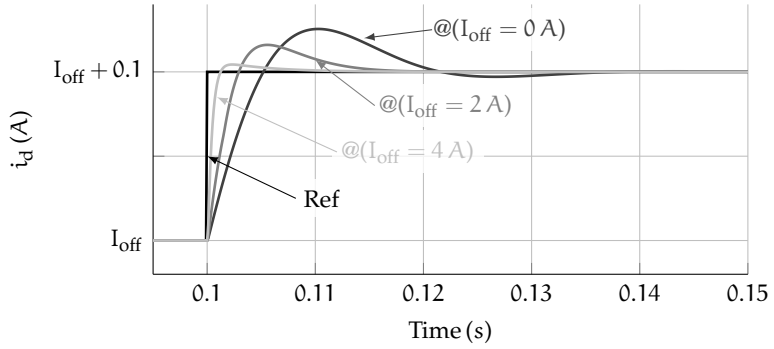
7.2 MODEL–BASED PI STRUCTURES

The proposed current controls are both model–based since a novel and accurate technique to estimate stator resistance and magnetic map is available and it was proposed in Part i. The presented methods rely on classical PI scheme and the frequency response criterion is chosen to design the current PIs. Furthermore, the aim was to develop control schemes which can accomplish the aforementioned described goals in a simple manner to encourage a fast and ease industrial implementation.

They are based on the availability of a precise magnetic model then differential, cross–differential and apparent inductances are known. The already presented FF terms in Sec. 7.1.1 are included in both techniques. Finally, only the d–axis current controller is considered in the following. This is not a limitation, since the same techniques also apply for the q–axis current control, with identical considerations.



(a) Conventional PI



(b) Robust design

Figure 7.3: d–axis current responses with a conventional PI.

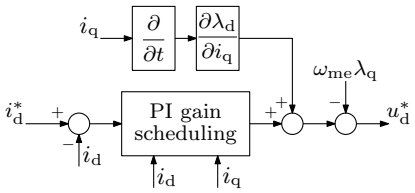


Figure 7.4: Gain–scheduling current control with FF.

7.2.1 *Gain–scheduling PI control*

The PI current control with variable parameters can be designed by exploiting the availability of the differential inductances. The block diagram of the proposed gain–scheduling PI current regulator is reported in Figure 7.4, which corresponds to the dot–dashed area of Figure 7.1. In order to design a gain–scheduling PI controllers, the system sketched in Figure 7.1 can be described by the product of expressions (7.5) and (7.7). The PI transfer function is:

$$C(s) = k_p(i_d, i_q) + \frac{k_i(i_d, i_q)}{s} = k_i(i_d, i_q)\frac{1 + s\tau_c(i_d, i_q)}{s} \tag{7.9}$$

where $k_p(i_d, i_q)$ is the proportional gain, $k_i(i_d, i_q)$ is the integral gain and $\tau_c(i_d, i_q) = k_p(i_d, i_q)/k_p(i_d, i_q)$ is the time constant of the controller. To ease the mathematical notation, the explicit dependence of

the PI parameters from the stator currents will be omitted from now on.

By imposing the desired phase margin φ_m at the chosen cut–off frequency ω_c, the following equation can be obtained:

$$\tau_c = \frac{\tan\left(\varphi_m - \frac{\pi}{2} + \tan^{-1}\left(\frac{\omega_c}{R} L_d^{diff}\right) + \tan^{-1}(\omega_c T_d)\right)}{\omega_c}. \tag{7.10}$$

Finally, the variable parameters of the PI controller are calculated as:

$$\begin{aligned} k_i &= \frac{\omega_c R \sqrt{1 + \left(\frac{\omega_c}{R}\right)^2 (L_d^{diff})^2}\sqrt{1 + \omega_c^2 T_d^2}}{\sqrt{1 + \omega_c^2 \tau_c^2}} \\ k_p &= \tau_c k_i. \end{aligned} \tag{7.11}$$

The differential inductance in (7.10) and (7.11) is computed using (4.19) at each sampling time T_s. The clear advantage is that both k_p and k_i depend on the operating point, guaranteeing the same dynamic response over the whole dq operating region.

7.2.2 *Constant gain PI regulator with additional feed–forward*

The second method consists in a constant gain PI controller with an additional FF term able to compensate all nonlinearities. It is based on the similarities present in equations (7.3), where there are two similar nonlinear terms composed by differential inductances and currents derivatives. In method presented in Sec. 7.2.1, the voltage term related to the cross–differential inductance was compensated with a FF term while the main differential inductance effect was managed with a gain–scheduling controller. In theory, both voltage terms related to differential and cross–differential inductances can be handled in the same manner, without the use of a PI. Nevertheless, an integral action is usually wished to have a zero tracking error to a constant reference. The solution proposed in this section consists in compensating the nonlinear effects in both derivative terms by two FF actions.

Let defined a new flux linkage $\lambda'_d(i_d, i_q)$ as:

$$\lambda'_d(i_d, i_q) \triangleq \lambda_d(i_d, i_q) - L_{dm} i_d \tag{7.12}$$

where L_{dm} is the smallest differential inductance on the d–axis. The voltage equation (7.2) can be rewritten as follow:

$$\begin{aligned} u'_d &= R i_d + \frac{\partial \lambda_d}{\partial t} = R i_d + \frac{\partial\left(\lambda'_d + L_{dm} i_d\right)}{\partial t} \\ &= R i_d + \frac{\partial\left(\lambda'_d + L_{dm} i_d\right)}{\partial i_d}\frac{\partial i_d}{\partial t} + \frac{\partial\left(\lambda'_d + L_{dm} i_d\right)}{\partial i_q}\frac{\partial i_q}{\partial t} \\ &= R i_d + L_{dm}\frac{\partial i_d}{\partial t} + \frac{\partial \lambda'_d}{\partial i_d}\frac{\partial i_d}{\partial t} + \frac{\partial \lambda'_d}{\partial i_q}\frac{\partial i_q}{\partial t}. \end{aligned} \tag{7.13}$$

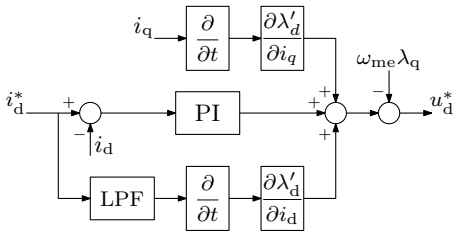


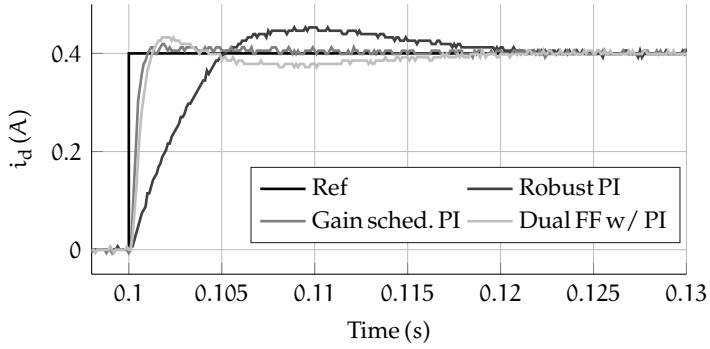
Figure 7.5: Current control with double FF. One from the reference.

The first two terms on the right–hand side of (7.13) constitute a linear first–order system, for which a standard PI regulator can be properly designed. It is worth remembering that the obtained PI controller is always stable since it is designed using the smallest inductance of the d–axis, as described in Sec. 7.1.3. The last term in (7.13) is the cross–differential inductance voltage drop since the following equalities hold:

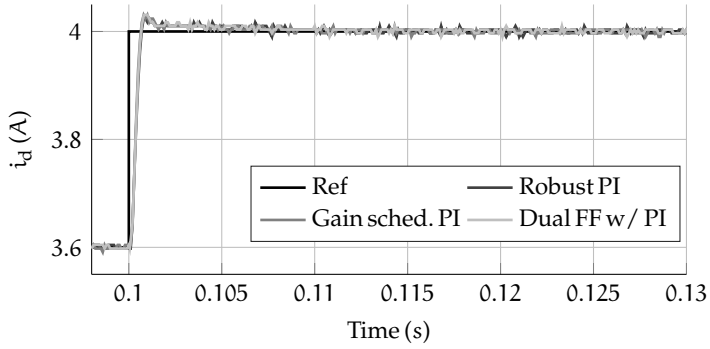
$$\frac{\partial \lambda_d}{\partial i_q} = \frac{\partial \lambda'_d}{\partial i_q} = \frac{\partial \lambda_q}{\partial i_d} = \frac{\partial \lambda'_q}{\partial i_d} = L_{dq}^{diff}. \tag{7.14}$$

Being the same described in Sec. 7.1.1, it is already compensated by applying the FF terms (7.4). Finally, the (7.13) is composed by two terms which are similar, where one is balanced by a FF action as aforementioned said. Therefore, the third term in the right member of (7.13) would be compensated with a FF term, as well. Actually, it cannot be implemented as a simple FF since it is directly function of the controlled variable i_d. Its implementation would lead to an uncontrollable modification of the loop transfer function, with detrimental effects on stability and performances. In order to solve this issue, the real current i_d is replaced by the current reference in the computation of the derivative. The control scheme is shown in Figure 7.5, which still corresponds to the dot–dashed area of Figure 7.1. To increase the similarity with the actual current, the reference is processed by a low–pass filter, whose bandwidth is equal to that fixed during the design of the PI regulator. As a positive aspect, the current reference contains less noise than the measured current, which yields a smoother derivative computation.

In Sec. 7.1.3 several tests were carried out to show the issues related to a constant–gain PI regulator. Some experimental tests were executed to prove the effectiveness of the proposed control techniques. Same requirements were used to do a fair comparison, i. e. a control bandwidth of 300 Hz, a phase margin of 70° and all the tests were carried out at standstill. The results are reported in Figure 7.6. The current transients from 0 A to 0.4 A are reported in Figure 7.6a. It is evident how the proposed techniques allow very fast responses, conversely to the standard robust PI controller. The dual–FF PI exhibits a little overshoot and undershoot, probably due to the approximation of considering the reference current instead of the measured one. When the transient



(a) Low–current step.



(b) High–current step.

Figure 7.6: Experimental d–axis current response with proposed methods.

due to the low–pass filter on the current reference is over, the FF action is zero and the current control is based only on the constant–gain PI regulator. It is tuned with the smallest inductance L_{dm} of the d–axis then the system has a small bandwidth at low current. Therefore, the quite long settling time is motivated. The d–axis current responses to a step from 3.6 A to 4 A are shown in Figure 7.6b. The three current responses have similar behaviour because (at nominal current) the robust PI, gain–scheduling and the dual–FF are all designed with the same saturated motor inductances.

7.3 FUTURE WORKS

The proposed current regulators are tailored for nonlinear motors, such as SynR motors. They were all tested in laboratory and the results met the expectation. Both are stable and they maintain same dynamic performances over the whole operating region. It is worth noting that the proposed techniques handle the nonlinearities differently. The former adapts the regulator's gain over the working point sampling time by sampling time then the transfer function of the control loop varies continuously. The latter one has a fixed regulator while the nonlinearity is managed by the FF terms. Intuitively, the stability analysis of the latter method should demand less effort since the loop

transfer function is already stable. A comprehensive analytical study of the stability is in progress, as conclusion of this peculiar current control activity.

8

DIRECT MODEL PREDICTIVE CONTROL

The MPC is an advanced method that is used to control a process while satisfying a set of constraints. The first uses were in the refining, petrochemicals and chemical industries since 1980s while it received little attention from power electronics community [58]. The reasons for the late adoption of MPC by power electronics community include the limited processing power that was available in the last century, which was a strong limitation for NNs too, and the very short time constants of the power electronic systems. The former led to limitations in solving the control problem in real–time, while the latter imposed the use of a short sampling interval. Nevertheless, some initial investigations in MPC–related concepts for power converters were accomplished in the 1980s [59]. In the recent years, MPC has rapidly emerged in power electronics [60]. The tremendous increase of the computational power available in the controller hardware has facilitated its progress but it was accompanied by a significant speed–up of the solvers that compute the solution to the underlying optimisation problem.

In motor control field, MPC has been applied to several variables, for instance to the electromechanical torque [61–64], the mechanical speed [65–67] or the stator currents [68–70], and to different kind of motors like IMs, PMSMs and SynR motors. Furthermore, different MPC schemes have been proposed, some called indirect MPC where a modulator is compulsory and others called direct MPC or finite control set (FCS) MPC able to directly manipulate the switching pattern. Since the discrete nature of a VSI, direct MPCs are playing the lion's share. Its inherent ability to find the optimal switching pattern, according to a cost function, allows to fully exploit the potentialities of the converter. This feature is interesting in high power application, where the hardware oversizing is remarkably expensive. Alternatively, MPC allows to reduce the hardware requirements, for example to reduce or remove harmonic filters and to reduce DC–link capacitors [60].

The main advantage of MPC is the time domain formulation rather than the frequency one. Generally, it enables MPC to address nonlinear systems in a systematic way. It is worthwhile noting how SynR motors are nonlinear systems and they are driven by VSIs which are switched nonlinear systems. Moreover, MPC is the only controller able to systematically cope with hard constraints on manipulated variables, states and controlled variable. This feature is suited for SynR motors control since they can work in different conditions, as MTPA, flux–weakening or MTPV, which are characterised by different active constraints. The possibility to include these constraints directly in the controller could

be extremely convenient for an effective control strategy. MPC consists in shifting a major part of the control effort from the design stage to the computational one. Therefore, in order to implement a comprehensive MPC, a powerful hardware is mandatory.

A fundamental parameter in MPC is the prediction horizon since performances, stability, parameters mismatch sensitivity, computational burden depend on it. Long prediction horizons yields a better closed–loop performances than a short ones. In particular, the infinite horizon case often ensures closed–loop stability, provided that a solution with a finite cost exists. Unfortunately, long horizons exacerbate the computational issue. FCS MPC provides a quite intuitive implementation but the computational complexity grows exponentially with the length of the prediction horizon, making the brute–force approach of exhaustive search rapidly unfeasible when long horizons are employed. In order to make the problem computational tractable, in [68] a heuristic branch–and–bound algorithm, namely SDA [71], used in e. g. communications and cryptography, is utilised for power electronic systems.

MPC is a promising control strategy for SynR motors, due to its ability to control nonlinear system. In addition, it is based on a cost function and it allows to address diverse and possibly conflicting control objectives. These objectives can be prioritised, by weighting them differently in the cost function. The cost function endues the MPC with the capability of directly handles multiple–input multiple–output (MIMO) plants without dividing them into multiple control loops with single-input single–output (SISO) controllers in a cascaded manner, unlike by PI controllers.

As introduced in Chapter 7, current control is the heart of high–end VSI then an extensive research has to be carry out to design the inverters of tomorrow. The proposed methods in Chapter 7 are feasible and easy to understand since PI regulators are well known in engineering. Nevertheless, they were not devised for controlling nonlinear and discontinuous systems such as SynR motors driven by a VSIs, indeed the concept of averaging is adopted. In some conditions, better performances can be achieved with a more sophisticated control scheme.

8.1 PHYSICAL MODEL OF THE SYSTEM

The analysed MPC belongs to the FCS MPC then the switching pattern is directly computed by the controller, i. e. the modulator is no longer required. The computational complexity of FCS MPC grows exponentially with the prediction horizon and with the number of levels that characterise the VSI taken into account. The studied system consists of a three–level neural point clamped (3L-NPC) VSI driving a SynR motor, as shown in Figure 8.1. It is assumed that the DC–link voltage V_{dc} is

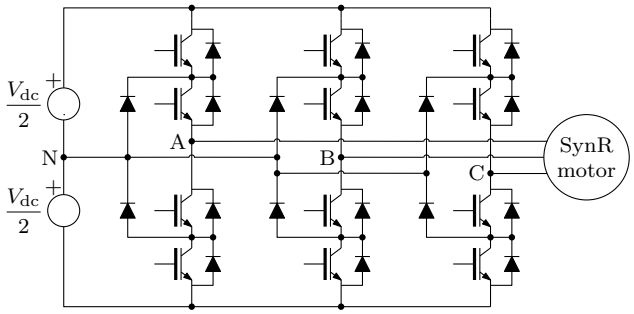


Figure 8.1: 3L-NPC VSI with a SynR motor. The inverter has a constant neutral point potential.

constant and the neural point potential v_N is fixed to zero. It is worth noting that 3L-NPC VSI is not the industrial standard but it is used for high power application, typically above the megawatt. Nevertheless, it was coupled with the SynR motor used throughout the thesis since the main aim of this work was to extend the method proposed in [68] to a nonlinear system. Of course, the DC–link voltage V_{dc} was reduced from the typically 5200 V for high power application to the industrial standard 560 V.

8.1.1 *Inverter model*

The single–phase switch position of a 3L-NPC is modelled with the integer variables s_a, s_b, $s_c \in \mathcal{S} \triangleq \{-1, 0, 1\}$. Each leg of the 3L-NPC can output a phase voltage of $-V_{dc}/2$, 0, $V_{dc}/2$. Introducing the vector $\mathbf{s}_{abc} = [s_a\ s_b\ s_c]^T \in \boldsymbol{\mathcal{S}} = \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ to denote the three–phase switch position, the output voltage of a three–level inverter in the three–phase abc frame can be expressed as:

$$\mathbf{u}_{abc} = \frac{V_{dc}}{2}\mathbf{s}_{abc}. \tag{8.1}$$

The voltage vectors of a 3L-NPC are depicted in Figure 8.2.

8.1.2 *Motor model discretisation*

The discretisation of the motor voltage equation is a fundamental step indeed it is crucial to find a trade–off among accuracy and complexity of the model, in order to make the state prediction feasible in real–time.

For the last time, the voltage equation of a SynR motor in the rotating reference frame is reported:

$$\frac{d\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right)}{dt} = \mathbf{u}_{dq} - R\mathbf{i}_{dq} - \omega_{me}\mathbf{J}\boldsymbol{\lambda}_{dq}\left(\mathbf{i}_{dq}\right). \tag{8.2}$$

Since the mechanical dynamics are much slower than the electrical ones, a constant motor speed is assumed during a control cycle.

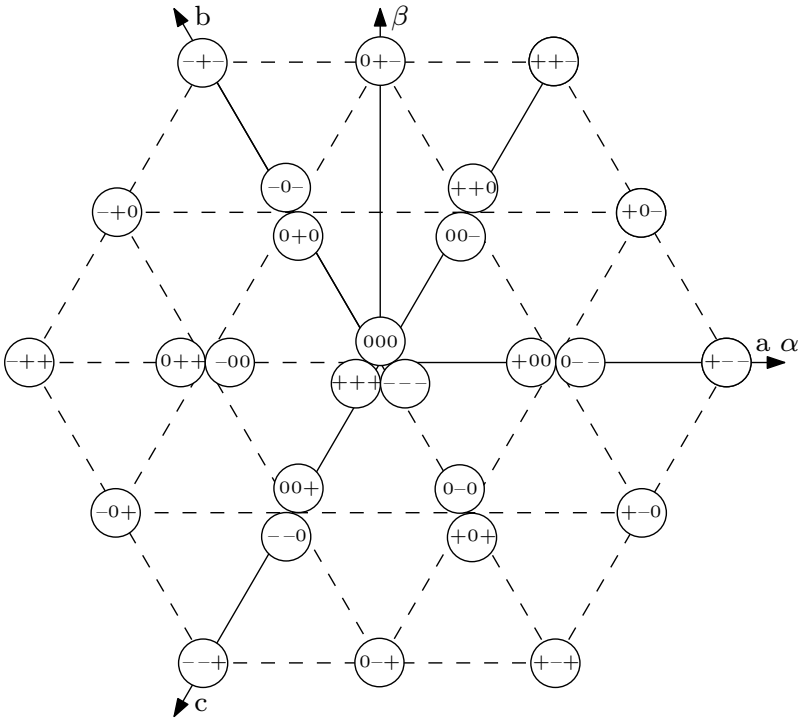


Figure 8.2: Voltage vectors produced by a three–level inverter shown in the $\alpha\beta$ plane with the corresponding values of the three–phase switch positions $\mathbf{s}_{abc}$ (where "+" refers to "+1' and "-" to "-1").

The implementation of the control strategy requires the discretisation of the continuous model. The method developed in [66] for a SynR motor is adopted, as a trade–off between accuracy and complexity. Let ξ^k indicate the sampled version of a generic continuous variable $\xi(t)$ at the sampling instant $t = kT_s$. Introducing the average operator:

$$\langle \xi \rangle^k \triangleq \frac{1}{T_s} \int_{kT_s}^{(k+1)T_s} \xi(t)\, dt \tag{8.3}$$

the integration of (8.2) over the interval $[kT_s, (k+1)T_s]$ yields:

$$\boldsymbol{\lambda}_{dq}^{k+1} = \boldsymbol{\lambda}_{dq}^{k} + T_s \langle \mathbf{u}_{dq} \rangle^k - RT_s \langle \mathbf{i}_{dq} \rangle^k - \mathbf{J}T_s \langle \omega_{me} \boldsymbol{\lambda}_{dq} \rangle^k . \tag{8.4}$$

Recalling the constant speed assumption over a sampling period, imposing a constant voltage vector $\mathbf{u}_{dq}$ over the whole sampling period and assuming that:

$$\langle \mathbf{i}_{dq} \rangle^k \approx \frac{\mathbf{i}_{dq}^{k+1} + \mathbf{i}_{dq}^{k}}{2}, \qquad \langle \omega_{me} \boldsymbol{\lambda}_{dq} \rangle^k \approx \omega_{me} \frac{\boldsymbol{\lambda}_{dq}^{k+1} + \boldsymbol{\lambda}_{dq}^{k}}{2} \tag{8.5}$$

the expression (8.4) can be further simplified to:

$$\boldsymbol{\lambda}_{dq}^{k+1} \approx \boldsymbol{\lambda}_{dq}^{k} + T_s \mathbf{u}_{dq}^{k} - RT_s \frac{\mathbf{i}_{dq}^{k+1} + \mathbf{i}_{dq}^{k}}{2} - \mathbf{J}\omega_{me} T_s \frac{\boldsymbol{\lambda}_{dq}^{k+1} + \boldsymbol{\lambda}_{dq}^{k}}{2}. \tag{8.6}$$

Given the nonlinear relation between the current vector $\mathbf{i}_{dq}^{k+1}$ and the flux linkage $\boldsymbol{\lambda}_{dq}^{k+1}$, (8.6) can be solved only numerically, which

is undesirable when real–time applications are of concern. However, accepting that the average current $\langle \mathbf{i}_{dq} \rangle^k$ in (8.6) is approximated by its instantaneous value $\mathbf{i}_{dq}^k$ (due to the small sampling interval), the following closed–form solution for $\boldsymbol{\lambda}_{dq}^{k+1}$ can be obtained:

$$\begin{aligned}\boldsymbol{\lambda}_{dq}^{k+1} \approx (\mathbf{I} - \mathbf{J}\omega_{me}T_s/2)\,(\mathbf{I} + \mathbf{J}\omega_{me}T_s/2)^{-1}\,\boldsymbol{\lambda}_{dq}^{k} \\ + T_s\left(\mathbf{u}_{dq}^k - R\mathbf{i}_{dq}^k\right)(\mathbf{I} + \mathbf{J}\omega_{me}T_s/2)^{-1}\,.\end{aligned} \tag{8.7}$$

The constant current assumption over a sampling period introduces a negligible error. It is worth noting that the voltage resistance drop is usually small respect to the motional terms. Therefore the error done by neglecting the current variation on the stator resistance is less than what would be obtained by keeping the flux linkages constant over a sampling period.

When devising MPC algorithms it is useful to have the model in its state–space representation [58]. Defining the state vector and the input one as:

$$\mathbf{x}_{dq}^k = \begin{bmatrix}\lambda_d^k \\ \lambda_q^k\end{bmatrix}, \qquad \mathbf{u}_{dq}^k = \begin{bmatrix}u_d^k \\ u_q^k\end{bmatrix} \tag{8.8}$$

the equation (8.7) can be rewritten as:

$$\mathbf{x}_{dq}^{k+1} = \mathbf{A}^k\mathbf{x}_{dq}^k + \widetilde{\mathbf{B}}\mathbf{u}_{dq}^k \tag{8.9}$$

where the time–varying state matrix is:

$$\mathbf{A}^k = \frac{1}{1+M^2}\begin{bmatrix}1 - M^2 - \frac{RT_s}{L_d^k} & 2M - \frac{RT_sM}{L_q^k} \\ -2M + \frac{RT_sM}{L_d^k} & 1 - M^2 - \frac{RT_s}{L_q^k}\end{bmatrix} \tag{8.10}$$

and the input matrix is:

$$\widetilde{\mathbf{B}} = \frac{1}{1+M^2}\begin{bmatrix}T_s & MT_s \\ -MT_s & T_s\end{bmatrix} \tag{8.11}$$

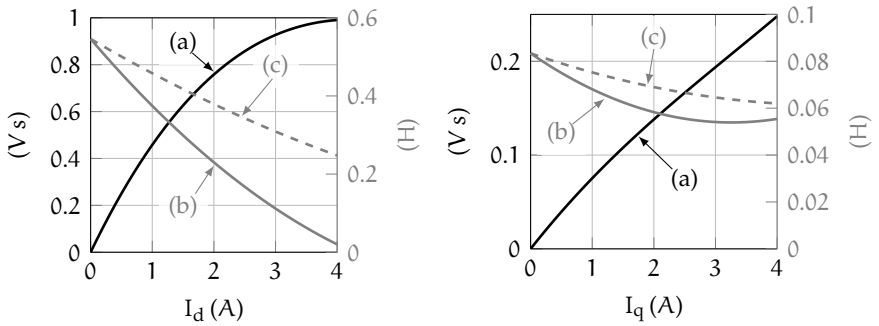
where $M = T_s\omega_{me}/2$. Moreover, the L_d^k and L_q^k are the d– and q–apparent inductances computed at the current $\mathbf{i}_{dq}^k$. The time–varying output matrix $\mathbf{C}^k$ is defined as:

$$\mathbf{C}^k = \begin{bmatrix}\frac{1}{L_d^k} & 0 \\ 0 & \frac{1}{L_q^k}\end{bmatrix} \tag{8.12}$$

and the output equation is:

$$\mathbf{y}^k = \mathbf{C}^k\mathbf{x}_{dq}^k \tag{8.13}$$

with the stator current in the dq frame forming the output vector, i. e. $\mathbf{y}^k = \mathbf{i}_{dq}^k$.



(a) d–axis. (b) q–axis.

Figure 8.3: Magnetic maps of a SynR motor: (a) flux linkages; (b) differential and (c) apparent inductances.

The aim of the chosen MPC is to find the optimal switch position to be applied to the inverter, i. e. $\mathbf{s}_{abc}^{opt}$. Therefore, the model of the drive needs to be revised to reflect this. To this aim, (8.9) is written as:

$$\mathbf{x}_{dq}^{k+1} = \mathbf{A}^k \mathbf{x}_{dq}^k + \widetilde{\mathbf{B}}\mathbf{T}_{abc/dq}\mathbf{u}_{abc}^k = \mathbf{A}^k \mathbf{x}_{dq}^k + \mathbf{B}^k \mathbf{s}_{abc}^k \tag{8.14}$$

where the input matrix $\mathbf{B}^k = \widetilde{\mathbf{B}}\mathbf{T}_{abc/dq} V_{dc}/2$ becomes time–varying. With the proposed modelling the features of the motor model in the dq frame, i. e. the magnetic maps independent of the position, are still present, while, at the same time, the three–phase switch position is modelled in a straightforward manner. Moreover, equations (8.13) and (8.14) represent a state model with the input expressed in the abc reference frame and state and output variables written in the rotating reference frame. This simplifies the derivation of the reference currents.

In order to simplify the analysis, but without any loss of generality, only the saturation effects are taken into account and the magnetic cross–coupling is neglected. The two flux components λ_d and λ_q are modelled as two nonlinear functions, each depending on the relative axis coordinate, i.e.:

$$\lambda_d = f_d(i_d), \qquad \lambda_q = f_q(i_q). \tag{8.15}$$

The two functions (8.15) and the associated differential and apparent inductances are shown in Figure 8.3. It was worth noting that the aim is to extend a linear MPC to a nonlinear system and it was carried out even if the cross–coupling is neglected. Probably, a real implementation of the proposed MPC control paradigm would lead lower performance than that achievable one with a MPC controller based on the entire magnetic maps.

8.2 DIRECT MODEL PREDICTIVE CONTROL

Direct MPC or FCS MPC is a control paradigm in which the switching signals are directly generated by the controller and fed into the inverter, namely without the interposition of a modulator. Each possible three–phase switch position is evaluated by the controller on the basis of the subsequent system performance, as quantified by a scalar performance index. The control action (i. e. the sequence of switch positions) that yields the minimum cost of the chosen performance index is selected as optimal. As can be understood, the definition of the performance index, also known as cost function, is one of the key–elements of an MPC algorithm and it is formulated based on the control objectives. For the system under investigation, the control tasks are the following:

- the flux linkage $\boldsymbol{\lambda}_{dq}$ should track its reference $\boldsymbol{\lambda}^{*}_{dq}$ accurately;
- the switching frequency has to be minimised;
- the inverter should be protected from overcurrents.

With regards to the first objective, by directly controlling the flux linkage the stator current is indirectly controlled, as can be seen from (8.15). It is worth remembering that the relation among flux linkages and stator currents is a static mapping in a synchronous motor, as shown in Chapter 4, and not a first–order dynamic relation as in IMs [72]. Therefore, the flux linkage is chosen to be controlled because this allows for the mathematical manipulations required when deriving the integer least–squares (ILS) problem in Sec. 8.3.

At time–step k, the first two control objectives are mapped into the cost function:

$$J^{k} = \sum_{l=k}^{k+N-1} \left\| \boldsymbol{\lambda}^{*,l+1}_{dq} - \boldsymbol{\lambda}^{l+1}_{dq} \right\|^{2}_{\mathbf{Q}} + \left\| \Delta \mathbf{s}^{l}_{abc} \right\|^{2}_{2} \tag{8.16}$$

which penalises the variables of interest over a finite prediction horizon of N time–steps[1]. The $\boldsymbol{\lambda}^{*}_{dq}$ is the flux linkage reference trajectory and the term $\Delta \mathbf{s}^{l}_{abc} = \mathbf{s}^{l}_{abc} - \mathbf{s}^{l-1}_{abc}$ accounts for the number of switching transitions involved between two consecutive time–steps. By minimising the number of transitions the switching losses (switching frequency) can be minimised as well. Matrix $\mathbf{Q} \in \mathbb{R}^{2\times 2}$ is a diagonal, positive definite weighting matrix which is used to compute the quadratic forms of the cost function. Note that $\|\xi\|^{2}_{\mathbf{Q}} \triangleq \xi^{T}\mathbf{Q}\xi$ denotes the squared norm of the vector ξ weighted by the matrix $\mathbf{Q}$. The choice of the diagonal entries of the weighting matrix sets the trade–off between the tracking accuracy of each flux linkage and the switching losses. In this thesis,

1 The 1–norm and the squared Euclidean norm of the switching effort yield the same cost $\|\Delta \mathbf{s}_{abc}\|_{1} = \|\Delta \mathbf{s}_{abc}\|^{2}_{2}$ since switching is possible only by one step up or down in each phase.

all the diagonal terms of $\mathbf{Q}$ are the same, meaning that the flux linkage tracking errors are equally penalised. In principle, it is possible to assign different weights. For instance, it could be interesting to track accurately the flux λ_d, as it is the main responsible for the torque response. It is worth noting how the most of the MPC schemes already presented weigh the term related to the switching effort, i. e. $\Delta\mathbf{s}_{abc}$, since the stator currents were tracked with the same relevance.

As listed above, the protection of the inverter from overcurrents has to be considered. This can be realised by imposing a constraint on the stator current as follows:

$$\left\| \mathbf{i}_{dq}^{k+1} \right\|_2 \leqslant i^{bnd} \tag{8.17}$$

where the positive scalar $i^{bnd} \in \mathbb{R}^+$ defines the boundary value of the stator current, as proposed in [73]. Generally, the control action of MPC guarantees that the cost function is always the smallest as possible. Nevertheless, the single controlled variables could assume unacceptable values, e. g. the current could overcome a safety limitation therefore an adequate constraint has to be added.

In order to find the optimal sequence of control actions at time–step k, the cost function (8.16) has to be minimised with respect to the so–called optimisation variable, namely the switch sequence $\mathbf{S}^k = \left[\mathbf{s}_{abc}^{k,T}\ \mathbf{s}_{abc}^{k+1,T}\ \dots\ \mathbf{s}_{abc}^{k+N-1,T}\right]^T$, over the horizon N. The following problem needs to be solved in real–time:

$$\mathbf{S}^{opt,k} = \arg\ \underset{S}{\text{minimise}}\ J^k \tag{8.18a}$$

$$\text{subject to: } \mathbf{S}^k \in \mathcal{S} \tag{8.18b}$$

$$\left\| \Delta\mathbf{s}_{abc}^{l} \right\|_\infty \leqslant 1,\ \forall l = k, \dots, k+N-1 \tag{8.18c}$$

$$\left\| \mathbf{i}_{dq}^{k+1} \right\|_2 \leqslant i^{bnd} \tag{8.18d}$$

with $\mathcal{S} = \mathcal{S} \times \cdots \times \mathcal{S}$ being the N–times Cartesian product of the set $\mathcal{S}$, representing the feasible input set[2]. Note that constraint (8.18c) allows to avoid a shoot–through in the inverter.

According to the receding horizon principle [58], only the first element $\mathbf{s}_{abc}^{k,T}$ of the sequence $\mathbf{S}^{opt,k}$ is applied to the system, whereas the rest elements are discarded. Following, at step $k+1$ the whole procedure is repeated over a shifted prediction horizon after acquiring new measurements and/or estimates. The receding horizon policy is fundamental to provide feed–back to the control action since the solution to the optimisation problem (8.18) yields at the time–step k an open–loop optimal sequence $\mathbf{S}^{opt,k}$ from the time–step k to $k+N-1$. Hence, MPC combines open–loop constrained optimal control with the receding horizon policy that provides feed–back and closes the control

2 The $\|\Delta\mathbf{s}_{abc}\|_\infty$ denotes the infinity–norm of the vector $\Delta\mathbf{s}_{abc}$, which is defined as the component of $\Delta\mathbf{s}_{abc}$ with the largest absolute value, that is, $\|\Delta\mathbf{s}_{abc}\|_\infty = \max(|\Delta s_a|, |\Delta s_b|, |\Delta s_c|)$.

loop. It is worth noting how the receding horizon policy is one of the main differences together the capability to manage constraints among the MPC and the linear–quadratic regulator (LQR).

8.3 INTEGER LEAST–SQUARES PROBLEM

The integer optimisation problem (8.18) is clearly nonlinear. However, since its form somewhat resembles the linear problem solved in [68], motivated by [74] where a nonlinear problem was transformed to a linear one, it was reformulated so as to solve it with the SDA. More details on the operation of the SDA can be found in [68] and [75]. This algorithm is a smart branch–and–bound method which can find the solution of the long–horizon direct MPC problem in a computationally efficient manner [75]. The underlying idea is that the integer solution of the direct MPC problem is in the hypersphere (n-dimensional sphere) of radius ρ^k centred at the unconstrained solution. This allows the a–priori exclusion of several candidate switch positions, thus, effectively reducing the number of possible solutions to be tested in real–time. Only these sequences the elements of which correspond to nodes that are inside the hypersphere are evaluated.

In brief, the main goal is to explore at least one complete branch, i. e. a complete switch sequence $\mathbf{S}^k$, from the root of the tree to the last node (leaf). Every time a complete branch is visited, the radius of the hypersphere gets smaller and the procedure is repeated as long as all possible paths to the leaf nodes are explored. By doing so, the optimal switch sequence that leads to the most desirable system behaviour is found. To speed up the search process, the initial radius ρ^k of the hypersphere should be carefully chosen. It should be large enough in order to contain at least one complete switch sequence, but sufficiently small to reduce the number of candidate solutions that need to be evaluated. Moreover, when computing the radius the current constraint (8.17) should be taken into account. This makes the initial choice of ρ^k more challenging since all switch positions that result in a current that exceeds its respective limit i^{bnd} should be discarded from the feasible set. To achieve this, the initial radius is defined according to [73].

The extension of the SDA to a nonlinear system can be done starting from the linear formulation [68]. Given the unconstrained solution $\mathbf{S}^{unc,k}$, which will be computed in Sec. 8.3.1, the cost function can be rewritten as follows:

$$J^k = \left(\mathbf{S}^k - \mathbf{S}^{unc,k}\right)^T \mathbf{H}^k \left(\mathbf{S}^k - \mathbf{S}^{unc,k}\right) \tag{8.19}$$

where $\mathbf{H}^k$ is symmetric, positive definite matrix defined in Sec. 8.3.1. A unique invertible and lower triangular matrix $\mathbf{V}^k \in \mathbb{R}^{3N\times 3N}$ exists

and satisfies $(\mathbf{V}^k)^T\mathbf{V}^k = \mathbf{H}^k$. The problem (8.18) can be rewritten as the equivalent constrained ILS problem:

$$\mathbf{S}^{opt,k} = \arg\underset{S}{\text{minimise}} \ \left\|\bar{\mathbf{S}}^{unc,k} - \mathbf{V}^k\mathbf{S}^k\right\|_2^2 \tag{8.20a}$$

$$\text{subject to } \mathbf{S}^k \in \mathbb{S} \tag{8.20b}$$

$$\left\|\Delta\mathbf{s}_{abc}^l\right\|_\infty \leqslant 1, \ \forall l = k, \cdots, k+N-1 \tag{8.20c}$$

$$\left\|\mathbf{i}_{dq}^{k+1}\right\|_2 \leqslant i^{bnd} \tag{8.20d}$$

where $\bar{\mathbf{S}}^{unc,k} = \mathbf{V}^k\mathbf{S}^{unc,k}$. Note that in order to extend the SDA to the nonlinear system under examination, a linearisation of the system along the unconstrained solution has been accomplished, see Sec. 8.3.1.

8.3.1 *Unconstrained solution*

As mentioned, (8.18) represents a nonlinear quadratic function and its minimum cannot be computed by a closed–form expression. Therefore, the Gauss–Newton algorithm (GNA) [76] was adopted to find the unconstrained solution of the cost function, by first ignoring the constraints (8.18b)–(8.18d). The GNA can only be used to minimise a sum of squared function values, but it has the advantage that the second derivatives, which can be challenging to compute, are not required. It is worth noting how the LMA, used to train the NN in Chapter 4, is an extension of GNA where the trust region approach is used. Both algorithms are obtained in Appendix C. The iterative algorithm is of the form:

$$\mathbf{S}^{unc,k}(z+1) = \mathbf{S}^{unc,k}(z) - \left[\mathbf{J}^k(z)^T\mathbf{J}^k(z)\right]^{-1}\mathbf{J}^k(z)^T\boldsymbol{\epsilon}(z) \qquad z \in \{0,\ldots,Z\} \tag{8.21}$$

where $Z \in \mathbb{N}$ is the number of iterations needed to find the global minimum with the given accuracy. $\mathbf{S}^{unc,k}(z)$ denotes the (real–valued) switch position at iteration z, while $\mathbf{J}^k(z)$ is the Jacobian matrix of the problem and $\boldsymbol{\epsilon}(z)$ is the error vector which has the same arguments as the cost function J^k, i. e. is a function of the flux linkages and the switching effort. It should be pointed out that both $\mathbf{J}^k(z)$ and $\boldsymbol{\epsilon}(z)$ are calculated with respect to $\mathbf{S}^{unc,k}(z)$.

Given the error vector, and by splitting the two terms in (8.16), the following expression is obtained:

$$J^k = \sum_{l=k}^{k+N-1}\left\|\boldsymbol{\lambda}_{dq}^{*,l+1} - \boldsymbol{\lambda}_{dq}^{l+1}\right\|_{\mathbf{Q}}^2 + \sum_{l=k}^{k+N-1}\left\|\Delta\mathbf{s}_{abc}^l\right\|_2^2 = \boldsymbol{\epsilon}_{dq}^T\boldsymbol{\epsilon}_{dq} + \boldsymbol{\epsilon}_{\mathbf{s}_{abc}}^T\boldsymbol{\epsilon}_{\mathbf{s}_{abc}} = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} \tag{8.22}$$

where $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_{dq}^T \ \boldsymbol{\epsilon}_{\mathbf{s}_{abc}}^T]^T \in \mathbb{R}^{5N\times 1}$ is the error vector. The structure of $\boldsymbol{\epsilon}$ facilitates the GNA execution.

The minimisation procedure (8.21) has to be computed at each time–step. Therefore, it needs to be as computationally efficient as possible. To this end, the previously computed unconstrained solution $\mathbf{S}^{\mathrm{unc},k-1}$ is utilised according to the receding horizon principle of MPC. It is reasonable to assume that the optimal solution at step k will be close to that at step $k-1$, at least at steady–state conditions. Therefore, at each step the GNA is initialised using the previous solution and the system is linearised along the same trajectory. The initial guess of $\mathbf{S}^{\mathrm{unc},k}(0)$ can be obtained by shifting $\mathbf{S}^{k-1}$ by one time–step and repeating the last switch position:

$$\mathbf{S}^{\mathrm{unc},k}(0) = \mathbf{R}\mathbf{S}^{k-1} \tag{8.23}$$

where the matrix $\mathbf{R}$ is defined as:

$$\mathbf{R} = \begin{bmatrix} 0_3 & \mathbf{I}_3 & 0_3 & \cdots & 0_3 \\ 0_3 & 0_3 & \mathbf{I}_3 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0_3 \\ 0_3 & \cdots & \cdots & 0_3 & \mathbf{I}_3 \\ 0_3 & \cdots & \cdots & 0_3 & \mathbf{I}_3 \end{bmatrix} \tag{8.24}$$

Given the guess for the initial input, it is possible to calculate the associated initial flux linkage trajectory $\boldsymbol{\lambda}_{dq}^{k}(0)$ using (8.14) and (8.23). The initial error vector $\boldsymbol{\epsilon}(0)$ can be computed as:

$$\boldsymbol{\epsilon}(0) = \begin{bmatrix} \sqrt{\widetilde{\mathbf{Q}}}\left(\boldsymbol{\lambda}_{dq}^{*} - \boldsymbol{\lambda}_{dq}^{k}(0)\right) \\ \mathbf{S}^{\mathrm{unc},k}(0) - \mathbf{S}^{k-1} \end{bmatrix} = \begin{bmatrix} \sqrt{\widetilde{\mathbf{Q}}}\left(\boldsymbol{\lambda}_{dq}^{*} - \boldsymbol{\lambda}_{dq}^{k}(0)\right) \\ -\mathbf{S}^{\mathsf{T}}\mathbf{S}^{k-1} \end{bmatrix} \tag{8.25}$$

where $\widetilde{\mathbf{Q}} \in \mathbb{R}^{2N\times 2N}$ is a block diagonal matrix $\widetilde{\mathbf{Q}} = \mathrm{diag}\,(\mathbf{Q},\ldots,\mathbf{Q})$ and the time–invariant matrix $\mathbf{S}$ is defined as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_3 & 0_3 & 0_3 & \cdots & 0_3 \\ -\mathbf{I}_3 & \mathbf{I}_3 & 0_3 & \ddots & \vdots \\ 0_3 & -\mathbf{I}_3 & \mathbf{I}_3 & \ddots & 0_3 \\ \vdots & \ddots & \ddots & \ddots & 0_3 \\ 0_3 & \cdots & 0_3 & -\mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix} \tag{8.26}$$

Matrix $\mathbf{J}^{k}(z) \in \mathbb{R}^{5N\times 3N}$ is the Jacobian matrix of function (8.16), calculated along the trajectory $\boldsymbol{\lambda}_{dq}^{k}(z)$. The entries of $\mathbf{J}^{k}(z)$ are defined as the derivative of each element of $\boldsymbol{\epsilon}(z)$ with respect to each element of $\mathbf{S}^{\mathrm{unc},k}(z)$. Thanks to the structure of the vector error $\boldsymbol{\epsilon}$, the Jacobian has the following advantageous structure:

$$\mathbf{J}^{k}(z) = \begin{bmatrix} \boldsymbol{\Upsilon}^{k}(z) \\ \mathbf{S} \end{bmatrix} \tag{8.27}$$

In addition, the time–varying matrix $\boldsymbol{\Upsilon}^k(z)$ is a lower triangular block matrix owing to the fact that the prediction error at step n is independent of the input at step m, with $m > n$. For this reason, $\boldsymbol{\Upsilon}^k(z)$ can be defined as:

$$\boldsymbol{\Upsilon}^k(z)\big(2(m-1)+1:2m, 3(n-1):3n\big) = \\ = \begin{cases} \sqrt{\mathbf{Q}} \begin{bmatrix} \frac{\partial \epsilon_d^m}{\partial s_a^n} & \frac{\partial \epsilon_d^m}{\partial s_b^n} & \frac{\partial \epsilon_d^m}{\partial s_c^n} \\ \frac{\partial \epsilon_q^m}{\partial s_a^n} & \frac{\partial \epsilon_q^m}{\partial s_b^n} & \frac{\partial \epsilon_q^m}{\partial s_c^n} \end{bmatrix} & \text{if } n \leqslant m \\ \mathbf{0}_{2,3} & \text{otherwise} \end{cases} \tag{8.28}$$

where $m, n = 1, \ldots, N$. It is worth remembering that matrix $\mathbf{J}^k(z)$ and the error vector $\boldsymbol{\epsilon}(z)$ need to be recalculated at each iteration z until the error vector becomes smaller than a predetermined threshold. The (real–valued) switch position that minimises the error vector is the so–called unconstrained solution $\mathbf{S}^{\text{unc},k}$.

After the preprocessing stage is over, i. e. matrix $\boldsymbol{\Upsilon}^k$ and the unconstrained solution $\mathbf{S}^{\text{unc},k}$ are computed, matrix $\mathbf{H}^k$ is computed according to the following expression:

$$\mathbf{H}^k = (\boldsymbol{\Upsilon}^k)^T \widetilde{\mathbf{Q}} \boldsymbol{\Upsilon}^k + \mathbf{S}^T \mathbf{S} \tag{8.29}$$

8.3.2 *Hypersphere radius computation*

The initial radius ρ^k of the hypersphere defines the upper bound of the search process. As mentioned above, a good upper bound should guarantee that problem (8.20) is feasible as well as that SDA can terminate (if possible) within the given time, i. e. the sampling interval T_s. Therefore, it should be as small as possible, consistently with the prerequisite that the hypersphere has to include almost one complete branch of the tree. To this aim, in [77], the initial radius is computed as:

$$\rho^k = \min\left\{\rho_1^k, \rho_2^k\right\} \tag{8.30}$$

where

$$\rho_1^k = \left\| \bar{\mathbf{S}}^{\text{unc},k} - \mathbf{V}^k \mathbf{S}^{\text{bab},k} \right\|_2 \tag{8.31}$$

and

$$\rho_2^k = \left\| \bar{\mathbf{S}}^{\text{unc},k} - \mathbf{V}^k \mathbf{U}^{\text{ed},k} \right\|_2 . \tag{8.32}$$

Radius ρ_1^k in (8.31) depends on $\mathbf{S}^{\text{bab},k}$ that corresponds to the rounded unconstrained solution to the closest integer vector which respects the constraints [78], i.e.:

$$\mathbf{S}^{\text{bab},k} = \lfloor \mathbf{S}^{\text{unc},k} \rceil \qquad \text{s.t. } \mathbf{S}^{\text{bab},k} \in \mathbb{S}. \tag{8.33}$$

Radius ρ_2^k, as can be seen in (8.32), is a function of $\mathbf{U}^{ed,k}$ which is the previously applied solution $\mathbf{S}^{k-1}$ shifted by one step [68], i. e. the initial guess $\mathbf{S}^{unc,k}(0)$ for the GNA.

However, the computation of the radius ρ^k is more challenging when (8.17) has to be considered. More specifically, the hypersphere needs to include at least one switch position that leads to a current which does not violate its respective bound i^{bnd}. Calculating the radius based on (8.30) does not guarantee that the problem is feasible since the unconstrained solution is computed while neglecting the current constraint. As a consequence, the procedure for the computation of the initial radius needs to be revised. To this end, the constraint on the current, which is an output constraint since $\mathbf{y}^k = \mathbf{i}_{dq}^k$, is translated into an input one, as follows. By using (8.13), (8.17) can be rewritten as:

$$\left\| \mathbf{C}^k \mathbf{A}^k \mathbf{x}_{dq}^k + \mathbf{C}^k \widetilde{\mathbf{B}} \frac{V_{dc}}{2} \mathbf{s}_{dq}^k \right\|_2 \leqslant i^{bnd} \tag{8.34}$$

that is an ellipse $\mathcal{E}$ in the rotating reference frame centred at:

$$\mathbf{s}_{dq}^{constr,k} = -\frac{V_{dc}}{2} (\widetilde{\mathbf{B}})^{-1} \mathbf{A}^k \mathbf{x}_{dq}^k \tag{8.35}$$

where $\mathbf{s}_{dq}^k$ is the switch position in the rotating reference frame. The ellipse $\mathcal{E}$ can be expressed in a matrix form as $\widetilde{\mathbf{p}}^T \widetilde{\mathbf{G}} \widetilde{\mathbf{p}} = 0$ where the vector $\widetilde{\mathbf{p}}$ is defined as $\widetilde{\mathbf{p}} = [(\mathbf{s}_{dq}^k)^T \, 1]^T$ and the matrix $\widetilde{\mathbf{G}}$ as:

$$\widetilde{\mathbf{G}} = \begin{bmatrix} \Theta & \Psi^T/2 \\ \Psi/2 & \Gamma \end{bmatrix} \tag{8.36}$$

where

$$\begin{aligned} \Theta &= \left(\mathbf{C}^k \widetilde{\mathbf{B}} \frac{V_{dc}}{2} \right)^T \left(\mathbf{C}^k \widetilde{\mathbf{B}} \frac{V_{dc}}{2} \right) \\ \Psi &= 2 (\mathbf{x}_{dq}^k)^T \left(\mathbf{C}^k \mathbf{A}^k \right)^T \left(\mathbf{C}^k \widetilde{\mathbf{B}} \frac{V_{dc}}{2} \right) \\ \Gamma &= (\mathbf{x}_{dq}^k)^T \left(\mathbf{C}^k \mathbf{A}^k \right)^T \left(\mathbf{C}^k \mathbf{A}^k \right) \mathbf{x}_{dq}^k - (i^{bnd})^2. \end{aligned} \tag{8.37}$$

The ellipse can be brought back to the abc three–phase frame $\mathbf{p}^T \mathbf{G} \mathbf{p} = 0$ by employing the transformation from the dq to abc, namely:

$$\mathbf{G} = \bar{\mathbf{T}}_{abc/dq} \widetilde{\mathbf{G}} \bar{\mathbf{T}}_{abc/dq}^T \tag{8.38}$$

being the vector $\mathbf{p}$ defined as $\mathbf{p} = \mathbf{s}_{abc}^k$ and the matrix $\bar{\mathbf{T}}_{abc/dq}$ as $\bar{\mathbf{T}}_{abc/dq} = [\mathbf{T}_{abc/dq}; 1/2 \; 1/2 \; 1/2]$.

In order to meet constraint (8.17) the control input has to lie within the ellipse $\mathcal{E}$, as shown in Figure 8.4 in the $\alpha\beta$ plane. It is worth remembering that there is a static relation between abc and $\alpha\beta$ frames, then they are equivalent. Formulating the output constraint (8.17) as an input one changes the feasible set of the integer–valued input

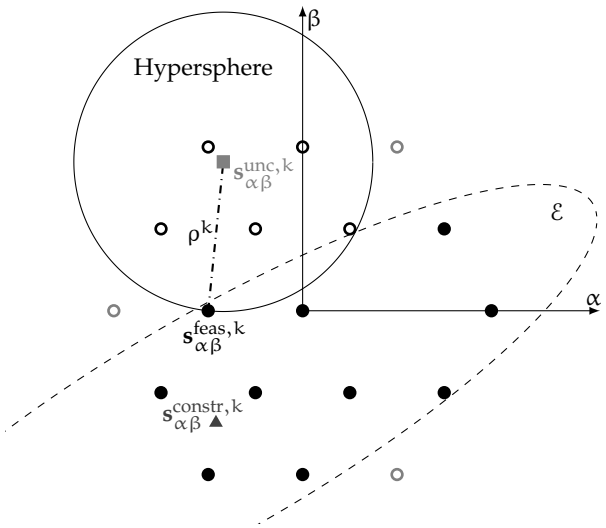


Figure 8.4: Feasible set in the $\alpha\beta$ plane when the current constraint (8.17) is active. The switches positions that satisfy the current constraint are the black solid circles. In this example, only one solid circle lies inside the hypersphere and meets all the constraint at the same time; this is the solution to the problem, and the corresponding three-phase switch position is applied to the converter.

$\mathcal{S}$ to $\mathcal{S}^{constr} = \mathcal{S}^{constr_1} \times \mathcal{S}^{constr_2} \times \cdots \times \mathcal{S}^{constr_N}$ with $\mathcal{S}^{constr_i} = \mathcal{S}$ for $i \in \{2, \ldots, N\}$ and

$$\mathcal{S}^{constr_1} = \left\{ \mathbf{s}_{abc}^k | \mathbf{s}_{abc}^k \in \mathcal{E} \ \wedge \ \mathbf{s}_{abc}^k \in \mathcal{S} \right\} \tag{8.39}$$

Having defined the feasible set, the computation of the initial radius ρ^k has to be refined accordingly. It has to respect the same requirement as in the unconstrained case, i.e. it should be smallest as possible, and the associate hypersphere has to include almost one feasible input. Therefore, the initial radius is computed as in [73].

In order to explain the concept of the refined feasible set and the revised computation of the radius, Figure 8.4 shows a case where the feasible set does not include all possible switch positions, but rather a subset of them, as limited because of the current constraint (8.17). For simplicity, the visualisation is done in the $\alpha\beta$ plane. As can be seen, the nearest point, i. e. switch position, to the unconstrained solution does not respect the current constraint. This implies that the hypersphere needs to be enlarged to include at least one feasible point. Hence, the radius ρ^k increases until it encloses until one feasible point, i. e. the point $\mathbf{s}_{\alpha\beta}^{feas,k}$, which corresponds to a switch position in the abc frame.

8.4 SIMULATION RESULTS

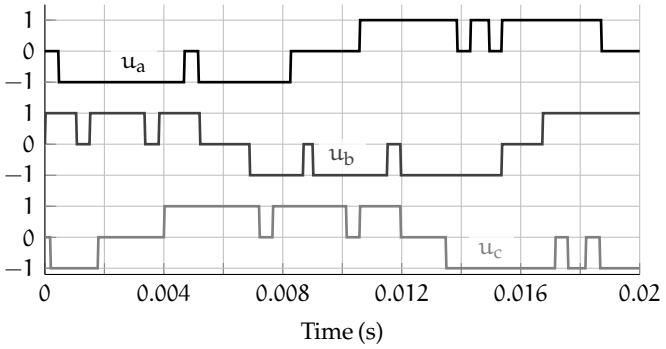
The proposed method was simulated for a 3L-NPC inverter with a constant DC bus voltage $V_{dc} = 560\,V$, driving the SynR motor with

parameters reported in Table A.1. All the simulations were carried out under the MTPA condition. Accordingly, the nominal torque is produced by i_d = 2.25 A and i_q = 3.30 A. The sampling interval was T_s = 25 µs. It is worth noting that the sampling interval plays a crucial role on the MPC performances. It should be as small as possible in order to have a high granularity in the control action but, at the same time, the prediction horizon should be extended to cover a proper time horizon. A trade–off among the sampling time and the prediction horizon has to be found [79]. Finally, the computational burden inherited by the prediction horizon has to taken into account for a real implementation which plays a strong restriction. The performance of direct MPC with long prediction horizon was investigated at steady–state, for a ten–step $N = 10$ prediction horizon. To minimise the nonlinear cost function J^k, only one iteration of the proposed algorithm was needed. A matrix $\mathbf{Q} = \mathrm{diag}\,(132, 132)$ was chosen by a trial–and–error procedure, to obtain a switching frequency of approximately 190 Hz.

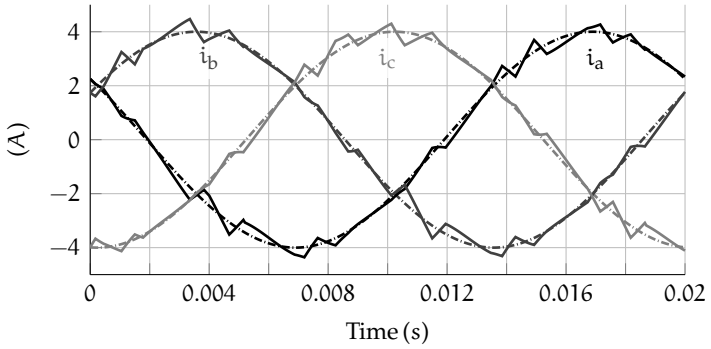
The results over one fundamental period are depicted in Figure 8.5. Figure 8.5a shows the three–phase switch positions, i. e. the outputs of direct MPC. The corresponding stator current $\mathbf{i}_{abc}$ and the flux linkages in the rotating reference frame along their references are depicted in Figure 8.5b and Figure 8.5c, respectively. Both flux linkages and stator currents follow accurately their reference values. For the sake of completeness, the harmonic content of the stator currents is shown in Figure 8.5d. A current total harmonic distortion (THD) of 5.01 % is achieved, a satisfactory result if compared with the THD=7.62 % of the conventional asymmetric space vector modulation (SVM) at the same switching frequency, see also Figure 8.6.

Figure 8.6 depicts the MPC performances obtained with different horizon lengths ($N = 1, 5, 10$) compared to ones obtained with carrier–based PWM and SVM. Specifically, a three–level, asymmetric, regularly sampled carrier–based PWM was implemented with two triangular carriers that are in phase, the so called phase disposition. In addition, a third harmonic component is added to the modulating reference signal. The carrier–based PWM with phase disposition is generally accepted for multilevel converter because a lowest harmonic distortion is achieved. Each dot corresponds to a single simulation carried out with a different value for the diagonal, non–zero entries of $\mathbf{Q}$, to test different switching frequencies. The fundamental frequency was 50 Hz.

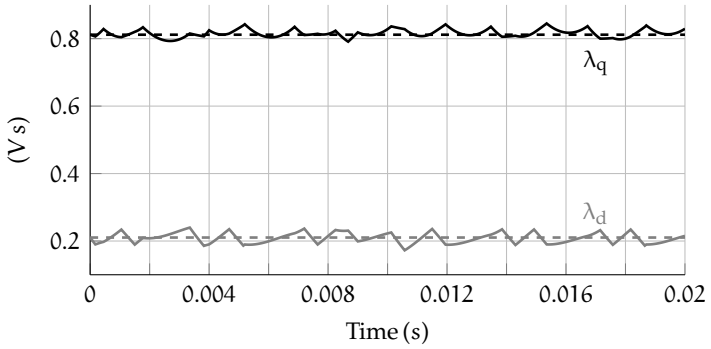
It is worth noting that the proposed direct MPC shows its advantages with respect to SVM at lower switching frequencies or, more generally, when the ratio between the switching frequency and the fundamental one is less than 10, Figure 8.6. One more thing observed in this figure is that the current THD reduces as the length of the prediction horizon increases. This is in line with the published literature, see i. e. [79] and [72]. However, it should be mentioned that the computational require-



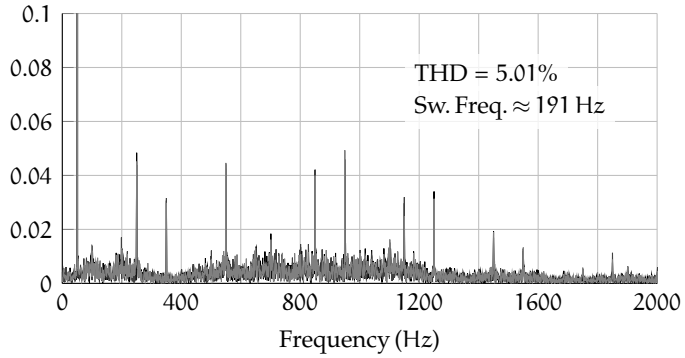
(a) Switch positions.



(b) Stator currents (solid lines) and their references (dashed lines).



(c) Flux linkages (solid lines) and their references (dashed lines).



(d) Current spectrum.

Figure 8.5: Direct MPC with flux linkages reference tracking at steady-state condition, at full speed, rated torque and in MTPA condition.

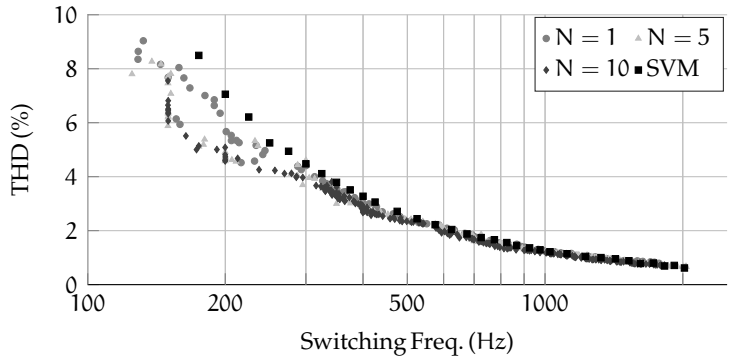


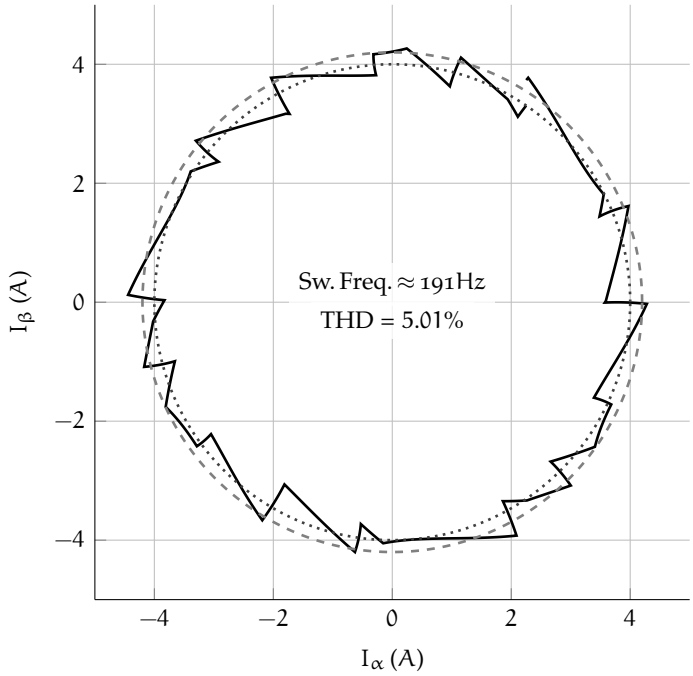
Figure 8.6: THD vs switching frequency obtained with the MPC and different prediction horizons. The THD obtained with the asymmetric SVM is added for comparison.

ments increase with the length of the prediction horizon, thus, there is a trade–off between system performance and prediction horizon that can be implemented [77]. Finally, for a complete analysis it is worth remembering that the SVM and the MPC cannot be implemented with the same hardware. The computational effort of the proposed MPC is orders of magnitude greater than that required by SVM, which is widespread implemented.

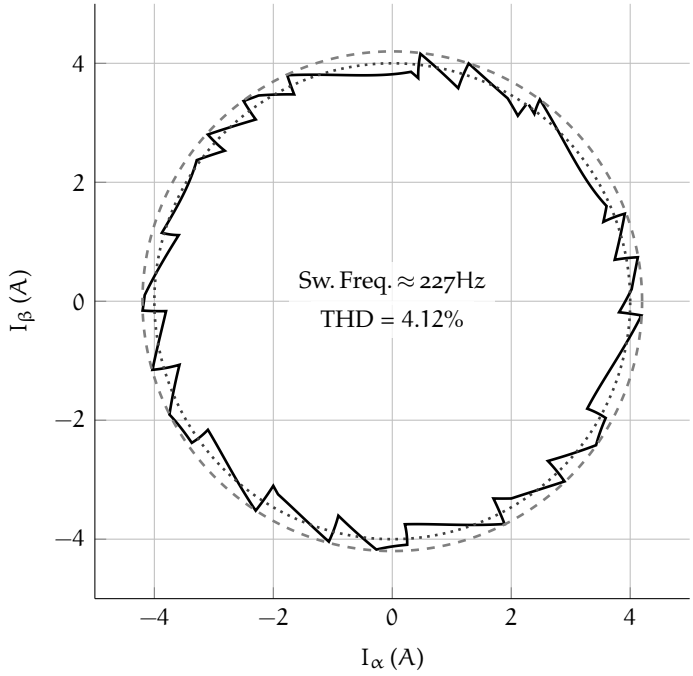
Figure 8.7 shows the effectiveness of the current limitation around $1.05\,I_N$, envisaging the unconstrained case in Figure 8.7a and the constrained one in Figure 8.7b. The stator current is depicted in the stationary $\alpha\beta$ plane, with the current reference and the bound. As can be seen in Figure 8.7a, where the current constraint (8.17) is inactive, such control actions, namely the switch position, are taken by the MPC algorithm that result, occasionally, in an instantaneous current that exceeds the bound, shown as a dashed circle. On the other hand, when the current constraint is taken into account by the optimisation problem, the current always remains within the current limitation, as depicted in Figure 8.7b. A side effect is that the MPC algorithm forces the converter to switch more frequently to keep the current within its bound, so that the switching frequency increases from 190 Hz to 227 Hz. The subsequent increase in the switching losses is balanced by a lower current distortion, which reduces from THD = 5.01 % to THD = 4.12 %.

8.5 FUTURE WORKS

The proposed MPC scheme was evaluated only through simulation since the real implementation demands a remarkable computational load. Therefore, an extensive work has to be carried out in order to implement it on a real test bench where a FPGA is likely mandatory. The optimisation stage has to be further developed in relation with the available hardware. At the same time, other aspects of MPC have



(a) Without any current constraint.



(b) With maximum current constraint.

Figure 8.7: Effects of the current limitation constraint. The solid lines —— represent the trajectory of the real current vector in the stationary ($\alpha\beta$) reference frame, whereas the dotted and dashed circles show its reference trajectory ····· and bound - - -, respectively.

to be deeply understood. In some working conditions, better THD can be achieve with MPC respect to the SVM and the prediction horizon influence was depicted. Notwithstanding, some open issues which would be interesting to analyse are the prediction horizon effect on parameters mismatch, MPC behaviour during transients, development of MPC speed control which takes into account additional constraints such as MTPA, flux–weakening and MTPV curves.

9 CONCLUSION

Horizon 2020 is pushing forward a comprehensive strategy to foster smart, inclusive and sustainable growth in Europe. Innovation was placed at the heart of the 2020 strategy since it is also the best means of successfully tackling major societal challenges, such as climate change and energy efficiency [80]. Smart urban technologies can provide an important contribution to the sustainable development of European cities. The smart cities and communities cover the areas of energy, transport, information and communication. The objective is to catalyse progress in those areas which offer new interdisciplinary opportunities to improve services while reducing energy and resource consumption and greenhouse gas and other polluting emissions. Electric motors represent one of the biggest electric energy consumers in the world, due to their wide utilisation both in industrial, transport and domestic applications.

In this thesis work, the aspects involving the development of AC high–end electric drives for SynR motors have been carefully studied and deepened. SynR motors fulfil the objectives posed by Horizon 2020 since they lack PM then they have smaller environmental footprint than PMSM, they may attain higher efficiency than IM and, last but not least, they are cheap. In order to fully exploit their features and to essentially increase the system efficiency, the first part of this dissertation introduced novel techniques to estimate the stator resistance and the highly nonlinear magnetic model of the connected motor. The main characteristic of the proposed methods is that they can work even without the need of specific tests, namely they can estimate the motor parameters during the normal working cycle of the electric drive.

The introduced magnetic model is based on RBF NN. A grey–box approach looked appropriate since the magnetic model of a reluctance–based motor is highly nonlinear and the NN returns a continuous function. It is an original contribution since it paves the way to many applications in which the flux linkages derivatives are needed. For instance, an innovative MTPA tracking method was presented. It is deeply merged with the magnetic model identification and it is able to understand where the real MTPA curve is respect to the current working point by exploiting the locally estimated flux linkages. The overall accuracy in the model estimation takes advantage of the precise stator resistance identification. It is completely independent from all the other motor parameters, providing a reliable and accurate estimation that is exploited for the NN training algorithm.

An accurate motor model shows its usefulness when used in conjunction with an advanced control algorithm. As a natural prosecution of the activities, the second part of the dissertation dealt with model–based control paradigms. The proposed PI–based current regulators exhibited clear improvements compared to classical control methods. Current dynamics met overall requirements in the entire input current range. All the proposed techniques, both identification and control algorithms, were deeply discussed and experimentally evaluated. The last control technique taken into consideration was the sophisticated MPC. It was proven to exhibit many advantages, but the implementation is still tricky. The study carried out in this work about it was devoted to the extension of the prediction horizon for nonlinear systems. The results were satisfactory and promising as well. A reduced harmonic content was attained respect to that achievable with SVM for low switching frequencies which can be benefited by high power converters. It opens up interesting researches perspectives.

The identification and control techniques presented in this dissertation can be implemented in general–purpose AC electric drive, especially if connected to a nonlinear motor as SynR one. Some of the depicted methods, i. e. the NN and the MPC, require a powerful hardware with a fast DSP and/or a FPGA while the other techniques can be directly applied on a standard drive. Future activities will focus on further optimisation and comprehension of the MPC potentialities, on development of a sensorless scheme for anisotropic motors based on the accurate available motor model and on overcoming of the assumptions made for the magnetic maps identification.

Part III

APPENDIX

A

TEST BENCH

The test bench was composed by two motors coupled in a back-to-back configuration with a torque transducer, as in Figure A.1. The load motor was a PMSM and it was driven by a standard AC drive which could be set in current or speed control, depending on the test that had to carried out. Different MUTs were used in the experimental stages and their parameters are shown in Table A.1. The MUT was fed by a two-level three-phase insulated gate bipolar transistor (IGBT) voltage inverter, connected to a variable voltage DC bus and controlled by a fast control prototyping system featuring a dSPACE DS1104 controlled board, programmed in C-language. The switching frequency F_s was set to 10 kHz. The inverter nonlinearities are compensated by a LUT based technique as described in [81], that is a common approach in standard AC drives. In order to improve the precision of the estimation techniques, a direct voltage measurement was also adopted. The voltage measurements was carried out by means of a custom FPGA–based digital voltage measurement system, based on a fast PWM signal oversampling. In fact, a direct voltage measurement is usually preferred over an estimation based on the use of the voltage references [24, 82–84]. It is worth noting how the benefits coming from the voltage measurements are manifolds. To mention some of them, advanced control strategies based on model based prediction, sensorless control and parameters estimation take great advantages from exact knowledge of the voltages.

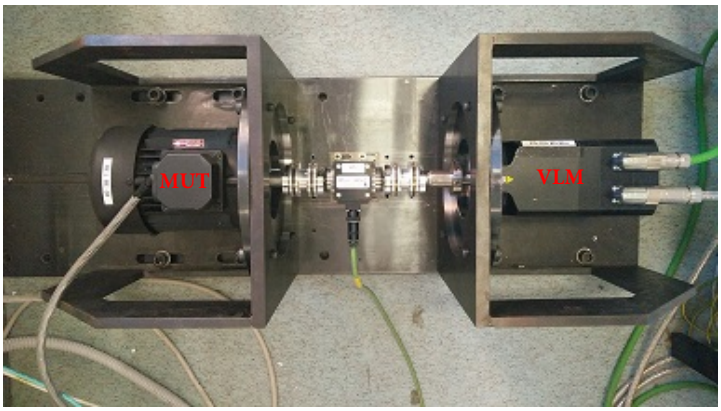


Figure A.1: Experimental rig.

	IPM$_1$	IPM$_2$	SynR
Nominal current (A)	4.2	12.1	4
Nominal speed (rpm)	3000	3000	1500
Nominal Torque (N m)	4.5	16	5.5
Pole pairs	2	3	2
Stator resistance (Ω)	2.73	0.367	4.76
d–inductance (mH)	21	8	380
q–inductance (mH)	114	15.7	85
Inertia (kg m^2)	$3 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	$3 \cdot 10^{-3}$
PM flux linkage (V s)	0.23	0.24	–

Table A.1: Synchronous motors nameplate data.

B

INDUCTANCES COMPUTATION

B.1 INDUCTANCES IN P POLE PAIRS MACHINE

The winding inductances of a p pole pairs machine may be determined by considering the elementary 4–pole, 3–phase synchronous machine shown in Figure B.1. In order to compute the self–inductance of the a winding, the air–gap flux density due to current only in the a winding can be obtained from (2.12) as:

$$B_r(\phi_s, \theta_{rm}) = \mu_0 \frac{MMF_a(\phi_s)}{g(\phi_s - \theta_{rm})} \tag{B.1}$$

where for a p pole pairs machine the MMF_a is given by (2.7) and θ_{rm} is defined in Figure B.1. From (2.11), the air–gap length for a p pole machine can be approximated as:

$$g(\phi_s - \theta_{rm}) = \frac{1}{\alpha_1 - \alpha_2 \cos\left(2p(\phi_s - \theta_{rm})\right)}. \tag{B.2}$$

Substituting (2.7) and (B.2) into (B.1), it yields:

$$B_r(\phi_s, \theta_{rm}) = \mu_0 \frac{N_s}{2p} i_a \cos(p\phi_s) \left[\alpha_1 - \alpha_2 \cos\left(2p\left(\phi_s - \theta_{rm}\right)\right)\right]. \tag{B.3}$$

Following (2.16), the flux linkage for a p pole pairs machine can be obtained as:

$$\lambda_a = L_\sigma i_a + p \int_{\frac{\pi}{p}}^{\frac{2\pi}{p}} N_a(\phi_s) \int_{\phi_s}^{\phi_s + \frac{\pi}{p}} B_r(\xi, \theta_{rm}) r l \, d\xi \, d\phi_s \tag{B.4}$$

where

$$N_a(\phi_s) = -\frac{N_s}{2p} \sin\left(p\phi_s\right), \qquad \frac{\pi}{p} \leqslant \phi_s \leqslant \frac{2\pi}{p}. \tag{B.5}$$

The double integral in (B.4) is multiplied by p to account for the flux linkages of the complete a winding. Evaluating (B.4) and dividing by i_a yields:

$$L_a = L_\sigma + \frac{N_s^2}{4p^3} \pi \mu_0 r l \left(\alpha_1 - \frac{\alpha_2}{2} \cos\left(2p\theta_{rm}\right)\right) \tag{B.6}$$

Finally, to link the p pole pairs machine inductances with the one pole pair motor describe in Sec. 2.1.2, it holds that:

$$\theta_r = p\theta_{rm}. \tag{B.7}$$

The same procedure can be followed to compute the others self– and mutual–inductances for a p pair poles machines.

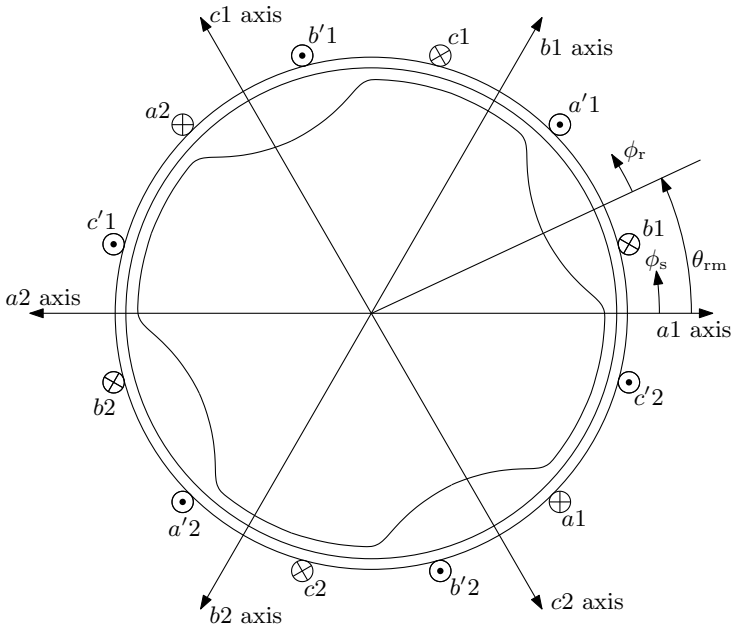


Figure B.1: Elementary 4–pole, 3–phase synchronous machine.

B.2 INDUCTANCES IN THE ROTATING REFERENCE FRAME

The inductances matrix in the rotating reference frame (2.39) can be computed by starting (2.27). Reminding the matrix transformation $\mathbf{T}_{abc/dqo}$ among the abc reference frame to the dqo rotating reference frame defined as:

$$\mathbf{T}_{abc/dqo} = \frac{2}{3}\begin{bmatrix} \cos(\vartheta_{me}) & \cos\left(\vartheta_{me}-\frac{2\pi}{3}\right) & \cos\left(\vartheta_{me}-\frac{4\pi}{3}\right) \\ -\sin(\vartheta_{me}) & -\sin\left(\vartheta_{me}-\frac{2\pi}{3}\right) & -\sin\left(\vartheta_{me}-\frac{4\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{B.8}$$

and in compact notation:

$$\boldsymbol{\xi}_{dqo} = \mathbf{T}_{abc/dqo}\boldsymbol{\xi}_{abc} \tag{B.9a}$$

$$\boldsymbol{\xi}_{abc} = \mathbf{T}^{-1}_{abc/dqo}\boldsymbol{\xi}_{dqo} \tag{B.9b}$$

where $\mathbf{T}_{dqo/abc} = \mathbf{T}^{-1}_{abc/dqo}$. By applying (B.9b) to (2.27), it yields:

$$\mathbf{T}^{-1}_{abc/dqo}\mathbf{u}_{dqo} = \mathbf{R}\mathbf{T}^{-1}_{abc/dqo}\mathbf{i}_{dqo} + \frac{d\mathbf{T}^{-1}_{abc/dqo}\boldsymbol{\lambda}_{dqo}}{dt} \tag{B.10}$$

or equivalently:

$$\mathbf{u}_{dqo} = \mathbf{R}\mathbf{i}_{dqo} + \mathbf{T}_{abc/dqo}\frac{d\mathbf{T}^{-1}_{abc/dqo}\boldsymbol{\lambda}_{dqo}}{dt} \tag{B.11}$$

In the linear case, the flux linkages can be computed by multiplying the motor inductances with the stator currents plus the flux linkages contribute due to the PM, as following:

$$\boldsymbol{\lambda}_{abc} = \mathbf{L}_{abc}\mathbf{i}_{abc} + \boldsymbol{\lambda}_{mg,abc} \tag{B.12}$$

and by applying (B.9b) to (B.12), it yields:

$$\mathbf{T}^{-1}_{abc/dqo}\boldsymbol{\lambda}_{dqo} = \mathbf{L}_{abc}\mathbf{T}^{-1}_{abc/dqo}\mathbf{i}_{dqo} + \mathbf{T}^{-1}_{abc/dqo}\boldsymbol{\lambda}_{mg,dqo} \tag{B.13}$$

therefore:

$$\begin{aligned}\boldsymbol{\lambda}_{dqo} &= \mathbf{T}_{abc/dqo}\mathbf{L}_{abc}\mathbf{T}^{-1}_{abc/dqo}\mathbf{i}_{dqo} + \boldsymbol{\lambda}_{mg,dqo} \\ &= \mathbf{L}_{dqo}\mathbf{i}_{dqo} + \boldsymbol{\lambda}_{mg,dqo}\end{aligned} \tag{B.14}$$

where $\mathbf{L}_{dqo} = \mathbf{T}_{abc/dqo}\mathbf{L}_{abc}\mathbf{T}^{-1}_{abc/dqo}$. The flux linkages due to the PM in the rotating reference frame $\boldsymbol{\lambda}_{mg,dqo}$ is a constant vector and equal to:

$$\boldsymbol{\lambda}_{mg,dqo} = \mathbf{T}_{abc/dqo}\boldsymbol{\lambda}_{mg,abc} = \begin{bmatrix}\Lambda_{mg}\\ 0\\ 0\end{bmatrix} \tag{B.15}$$

The inductances matrix in the rotating reference frame $\mathbf{L}_{dqo}$ can be obtained after many tedious algebraic manipulations which are hereafter omitted. The final result can be obtained by substituting (2.22) and (2.23) into (2.29b) and the resulting matrix computed with (B.9a) and (B.9b) in according to $\mathbf{L}_{dqo} = \mathbf{T}_{abc/dqo}\mathbf{L}_{abc}\mathbf{T}^{-1}_{abc/dqo}$. The matrix $\mathbf{L}_{dqo}$ is equal to:

$$\mathbf{L}_{dqo} = \begin{bmatrix}L_d & 0 & 0\\ 0 & L_q & 0\\ 0 & 0 & L_\sigma\end{bmatrix} \tag{B.16}$$

with

$$\begin{aligned}L_d &= L_\sigma + \frac{3}{2}(L_0 - L_2)\\ L_q &= L_\sigma + \frac{3}{2}(L_0 + L_2)\,.\end{aligned} \tag{B.17}$$

By replacing (B.14) into the (B.11), it gets:

$$\begin{aligned}\mathbf{u}_{dqo} &= \mathbf{R}\mathbf{i}_{dqo} + \mathbf{T}_{abc/dqo}\frac{d\mathbf{T}^{-1}_{abc/dqo}(\mathbf{L}_{dqo}\mathbf{i}_{dqo} + \boldsymbol{\lambda}_{mg,dqo})}{dt}\\ &= \mathbf{R}\mathbf{i}_{dqo} + \mathbf{T}_{abc/dqo}\frac{d\mathbf{T}^{-1}_{abc/dqo}}{dt}(\mathbf{L}_{dqo}\mathbf{i}_{dqo} + \boldsymbol{\lambda}_{mg,dqo})\\ &\quad + \mathbf{L}_{dqo}\frac{d\mathbf{i}_{dqo}}{dt}\end{aligned} \tag{B.18}$$

where the position independence of the matrix $\mathbf{L}_{dqo}$ and the vector $\boldsymbol{\lambda}_{dqo}$ is exploited. In addition, it holds:

$$\mathbf{T}_{abc/dqo}\frac{d\mathbf{T}_{abc/dqo}^{-1}}{dt} = \omega_{me}\begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} = \omega_{me}\bar{\mathbf{J}}. \tag{B.19}$$

Finally, the voltage balance in the rotating reference frame is:

$$\begin{aligned}\mathbf{u}_{dqo} &= \mathbf{R}\mathbf{i}_{dqo} + \mathbf{L}_{dqo}\frac{d\mathbf{i}_{dqo}}{dt} + \omega_{me}\bar{\mathbf{J}}(\mathbf{L}_{dqo}\mathbf{i}_{dqo} + \boldsymbol{\lambda}_{mg,dqo})\\ &= \mathbf{R}\mathbf{i}_{dqo} + \mathbf{L}_{dqo}\frac{d\mathbf{i}_{dqo}}{dt} + \omega_{me}\bar{\mathbf{J}}\mathbf{L}_{dqo}\mathbf{i}_{dqo} + \mathbf{e}_{dqo}\end{aligned} \tag{B.20}$$

where

$$\mathbf{e}_{dqo} = \omega_{me}\bar{\mathbf{J}}\boldsymbol{\lambda}_{dqo} = \begin{bmatrix}0\\ \omega_{me}\Lambda_{mg}\\ 0\end{bmatrix} \tag{B.21}$$

By disregarding the homopolar component in (B.20), the (2.37) is obtained.

C

GAUSS–NEWTON AND LEVENBERG–MARQUARDT ALGORITHM

The aim of GNA, LMA and other algorithms introduced in this appendix is to find the value of $\mathbf{x}$ which minimise the cost function $F(\mathbf{x})$ [42]. All the presented algorithms are iterative then from some initial guess $\mathbf{x}_0$, they are updated according to an equation of the form:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{p}^k \tag{C.1}$$

where the vector $\mathbf{p}^k$ represents a search direction and the positive scalar α^k is the learning rate which determines the length of the step.

Updating the initial guess, the cost function has to decrease at each iteration as follow:

$$F(\mathbf{x}^{k+1}) < F(\mathbf{x}^k) \tag{C.2}$$

therefore the direction $\mathbf{p}^k$ has to be chosen to go downhill for sufficiently small learning rate α^k. Considering the first–order Taylor series expansion of $F(\mathbf{x}^k)$ as:

$$F(\mathbf{x}^{k+1}) = F(\mathbf{x}^k + \Delta\mathbf{x}^k) \approx F(\mathbf{x}^k) + (\mathbf{g}^k)^T \Delta\mathbf{x}^k \tag{C.3}$$

where $\mathbf{g}^k$ is the gradient evaluated in $\mathbf{x}^k$:

$$\mathbf{g}^k \triangleq \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^k}. \tag{C.4}$$

For $F(\mathbf{x}^{k+1})$ to be less than $F(\mathbf{x}^k)$, the second terms on the right–hand side of (C.3) must be negative:

$$(\mathbf{g}^k)^T \Delta\mathbf{x}^k = \alpha^k (\mathbf{g}^k)^T \mathbf{p}^k < 0. \tag{C.5}$$

By selecting α^k small but greater than zero, this implies:

$$(\mathbf{g}^k)^T \mathbf{p}^k < 0. \tag{C.6}$$

Any vector $\mathbf{p}^k$ that satisfies this equation is called a descent direction and the function has to go down if a small enough step in this direction is taken. The direction in which the function $F(\mathbf{x}^k)$ decreases most rapidly corresponds when $(\mathbf{g}^k)^T \mathbf{p}^k$ is most negative. By assuming that the length of $\mathbf{p}^k$ does not change but only the direction, the inner product between the gradient and the direction vector is most negative when the direction vector has the same direction but opposite sign of the gradient. Therefore, a vector that points in the steepest descent direction is:

$$\mathbf{p}^k = -\mathbf{g}^k. \tag{C.7}$$

Replacing (C.7) in (C.1) yields:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \mathbf{g}^k \tag{C.8}$$

which is the method called *steepest descent*.

The derivation of the steepest descent algorithm is based on the first–order Taylor series expansion. Exploiting the second–order Taylor series:

$$F(\mathbf{x}^{k+1}) = F(\mathbf{x}^k + \Delta\mathbf{x}^k) \approx F(\mathbf{x}^k) + (\mathbf{g}^k)^T \Delta\mathbf{x}^k + \frac{1}{2}\Delta(\mathbf{x}^k)^T \mathbf{G}^k \Delta\mathbf{x}^k \tag{C.9}$$

where

$$\mathbf{G}^k = \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^k} \tag{C.10}$$

is the Hessian evaluated in $\mathbf{x}^k$, the Newton's method can be derived. The principle behind the Newton's method is to locate the stationary point of this quadratic approximation to $F(\mathbf{x})$. In order to find the minimum of (C.9), the gradient of $F(\mathbf{x}^{k+1})$ can be computed as:

$$\nabla F(\mathbf{x}^{k+1}) = \mathbf{G}^k \Delta\mathbf{x}^k + \mathbf{g}^k \tag{C.11}$$

and setting (C.11) equal to zero and solving for $\Delta\mathbf{x}^k$ yields:

$$\Delta\mathbf{x}^k = -(\mathbf{G}^k)^{-1}\mathbf{g}^k. \tag{C.12}$$

The *Newton's method* is then defined as:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{G}^k)^{-1}\mathbf{g}^k \tag{C.13}$$

This method always finds the minimum of a quadratic function in one step because it is designed to locate the stationary point of the quadratic function. If the cost function is quadratic, it is minimised in one step. If the cost function $F(\mathbf{x}^k)$ is not quadratic, the Newton's method does not generally converge in one step. Actually, the convergence is not guaranteed at all since it depends on the function and the initial guess.

In case of the cost function $F(\mathbf{x})$ is a sum of squares of other nonlinear function, GNA and LMA can be derived. By assuming that $F(\mathbf{x})$ is a sum of squares function:

$$F(\mathbf{x}) = \sum_{n=1}^{N} (v^n)^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x})\mathbf{v}(\mathbf{x}) \tag{C.14}$$

then the j–th element of the gradient is:

$$[\nabla F(\mathbf{x})]^j = \frac{\partial F(\mathbf{x})}{\partial x^j} = 2\sum_{n=1}^{N} v^n(\mathbf{x})\frac{\partial v^n(\mathbf{x})}{\partial x^j}. \tag{C.15}$$

The gradient can therefore be written in matrix form:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}) \tag{C.16}$$

where

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v^1(\mathbf{x})}{\partial x^1} & \frac{\partial v^1(\mathbf{x})}{\partial x^2} & \cdots & \frac{\partial v^1(\mathbf{x})}{\partial x^n} \\ \frac{\partial v^2(\mathbf{x})}{\partial x^1} & \frac{\partial v^2(\mathbf{x})}{\partial x^2} & \cdots & \frac{\partial v^2(\mathbf{x})}{\partial x^n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v^N(\mathbf{x})}{\partial x^1} & \frac{\partial v^N(\mathbf{x})}{\partial x^2} & \cdots & \frac{\partial v^N(\mathbf{x})}{\partial x^n} \end{bmatrix} \tag{C.17}$$

is the Jacobian matrix. In order to minimise (C.14) with the Newton's method (C.13), the Hessian matrix has to be computed as:

$$\left[\nabla^2 F(\mathbf{x})\right]^{k,j} = \frac{\partial F(\mathbf{x})}{\partial x^k \partial x^j} = 2\sum_{n=1}^{N}\left\{\frac{\partial v^n(\mathbf{x})}{\partial x^k}\frac{\partial v^n(\mathbf{x})}{\partial x^j} + v^n\frac{\partial v^n(\mathbf{x})}{\partial x^k \partial x^j}\right\} \tag{C.18}$$

which can be expressed in matrix form as following:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}) \tag{C.19}$$

where

$$\mathbf{S}(\mathbf{x}) = \sum_{n=1}^{N} v^n(\mathbf{x})\nabla^2 v^n(\mathbf{x}). \tag{C.20}$$

By assuming that $\mathbf{S}(\mathbf{x})$ is small then negligible, the Hessian matrix can be approximated as:

$$\nabla^2 F(\mathbf{x}) \approx 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) \tag{C.21}$$

Substituting (C.21) and (C.16) into (C.13), the GNA can be obtained:

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k - \left[2\mathbf{J}^T(\mathbf{x}^k)\mathbf{J}(\mathbf{x}^k)\right]^{-1} 2\mathbf{J}^T(\mathbf{x}^k)\mathbf{v}(\mathbf{x}^k) \\ &= \mathbf{x}^k - \left[\mathbf{J}^T(\mathbf{x}^k)\mathbf{J}(\mathbf{x}^k)\right]^{-1} \mathbf{J}^T(\mathbf{x}^k)\mathbf{v}(\mathbf{x}^k). \end{aligned} \tag{C.22}$$

It is worth noting how that the advantage of GNA over the standard Newton's method is that it does not require the calculation of second derivatives.

One problem with the GNA is that the matrix $\mathbf{H} = \mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x})$ may not be invertible. This can be overcome by using the following modification to the approximate Hessian matrix:

$$\widetilde{\mathbf{H}} = \mathbf{H} + \mu\mathbf{I} \tag{C.23}$$

This expedient makes the matrix $\widetilde{\mathbf{H}}$ certainly invertible. In order to demonstrate it, let suppose that the eigenvalues and eigenvectors of $\mathbf{H}$ are $\{\lambda^1, \lambda^2, \ldots, \lambda^n\}$ and $\{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^n\}$, respectively. Then:

$$\widetilde{\mathbf{H}}\mathbf{z}^i = [\mathbf{H} + \mu\mathbf{I}]\,\mathbf{z}^i = \mathbf{H}\mathbf{z}^i + \mu\mathbf{z}^i = \lambda^i\mathbf{z}^i + \mu\mathbf{z}^i = (\lambda^i + \mu)\mathbf{z}^i \tag{C.24}$$

therefore the eigenvectors of $\widetilde{\mathbf{H}}$ are the same as the eigenvectors of $\mathbf{H}$, and the eigenvalues of $\widetilde{\mathbf{H}}$ are $(\lambda^i + \mu)$. The matrix $\widetilde{\mathbf{H}}$ can be made

positive definite by increasing μ until $(\lambda^i + \mu) > 0$ for all i, then the matrix will be invertible.

Replacing the $\mathbf{H}$ with $\widetilde{\mathbf{H}}$ in (C.22), the LMA is derived:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left[\mathbf{J}^T(\mathbf{x}^k)\mathbf{J}(\mathbf{x}^k) + \mu^k\mathbf{I}\right]^{-1}\mathbf{J}^T(\mathbf{x}^k)\mathbf{v}(\mathbf{x}^k). \tag{C.25}$$

The LMA has the very useful feature that as μ^k is increased it approaches the steepest descent algorithm with small learning rate:

$$\mathbf{x}^{k+1} \approx \mathbf{x}^k \frac{1}{\mu^k}\mathbf{J}^T(\mathbf{x}^k)\mathbf{v}(\mathbf{x}^k) = \mathbf{x}^k - \frac{1}{2\mu^k}\nabla F(\mathbf{x}), \quad \text{for large } \mu^k \tag{C.26}$$

while as μ^k is decreased to zero the algorithm becomes the GNA.

The algorithm begins with a μ^k set to some small value. If a step does not yield a smaller value for $F(\mathbf{x})$, then the step is repeated with μ^k multiplied by some factor $\vartheta > 1$. The cost function should decrease since a small step in the direction of the steepest descent is taken. If the step produces a smaller value for $F(\mathbf{x})$, then μ^k is divided by ϑ for the next step, so that the algorithm will approach GNA, which should provide a faster convergence. The LMA provides a nice compromise between the speed of Newton's method and the guaranteed convergence of the steepest descent.

D

DELAY IN THE ROTATING REFERENCE FRAME

In order to describe the effect of the control and inverter delay T_d in the rotating reference frame, the method illustrated in Sec. 3.2 can be used. The relation among the actual voltages and the reference ones in the rotating reference frame, shown in Figure D.1, can be written as:

$$\mathbf{u}_{dq} = \mathbf{T}_{\alpha\beta/dq}(\omega_{me}t) \times \begin{bmatrix} e^{-sT_d} & 0 \\ 0 & e^{-sT_d} \end{bmatrix} \times \mathbf{T}_{dq/\alpha\beta}(\omega_{me}t)\mathbf{u}^*_{dq} \quad \text{(D.1)}$$

which represents the *base-band* dynamical response of a modulated system, i. e. the delay. Equation (D.1) is akin to (3.5) with (3.6). Both equations are a composition of time–domain and Laplace–domain matrices, so (D.1) can be rewritten in the Laplace–domain only by applying the technique presented in Sec. 3.2. In detail, $\mathbf{u}_{dq}$ can be expanded to obtain the expressions of u_d and u_q. For sake of simplicity, only u_d is obtained hereafter:

$$\begin{aligned} u_d = {} & \cos(\omega_{me}t) \times e^{-sT_d} \times \cos(\omega_{me}t)u^*_d - \cdots \\ & \cdots - \cos(\omega_{me}t) \times e^{-sT_d} \times \sin(\omega_{me}t)u^*_q + \cdots \\ & \cdots + \sin(\omega_{me}t) \times e^{-sT_d} \times \cos(\omega_{me}t)u^*_q + \cdots \\ & \cdots + \sin(\omega_{me}t) \times e^{-sT_d} \times \sin(\omega_{me}t)u^*_d. \end{aligned} \quad \text{(D.2)}$$

and, analogously, u_q can be obtained. It is worth noting that both voltages are composed by addends which represents the sequence of a modulation (with carrier frequency ω_{me}), a base–band filtering and a demodulation.

The first term of (D.2) coincided with the modulated system shown in Figure D.2. It is equal to Figure 3.4, with the exception of the final filter $F(s)$. Exploiting the Laplace transform of a modulated signal (see (3.26)), the transfer function of the modulated system (D.2) can be computed. For sake of simplicity, the transfer functions of the modulated system depicted in Figure D.2 are reported hereafter. The transfer function of the modulated input u_m is:

$$U_m(s) = \frac{1}{2}\left[U(s - j\omega_{me}) + U(s + j\omega_{me})\right]. \quad \text{(D.3)}$$

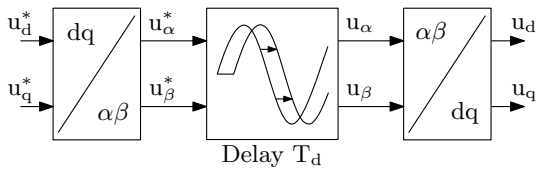


Figure D.1: Relation among the actual voltages and the reference ones in the rotating frame.

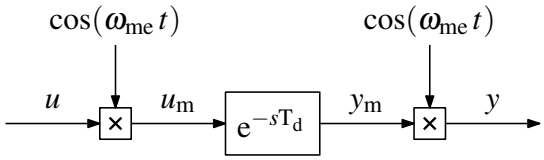


Figure D.2: Modulated system.

The modulated signal is then delayed to get:

$$Y_m(s) = e^{-sT_d} U_m(s). \tag{D.4}$$

The property (3.26) can be further applied to obtain the Laplace transform of the demodulated output, namely:

$$Y(s) = \frac{1}{2}\left[Y_m(s - j\omega_{me}) + Y_m(s + j\omega_{me})\right]. \tag{D.5}$$

Expression (D.5) can be expanded by using (D.3) and (D.4) as follow:

$$\begin{aligned} Y(s) = \frac{1}{4}\left[e^{-(s-j\omega_{me})T_d} + e^{-(s+j\omega_{me})T_d}\right] U(s) + \cdots \\ \cdots + \frac{1}{4}\left[e^{-(s-j\omega_{me})T_d} U(s - 2j\omega_{me}) + \cdots \right. \\ \left. \cdots + e^{-(s+j\omega_{me})T_d} U(s + 2j\omega_{me})\right]. \end{aligned} \tag{D.6}$$

By repeating the same procedure for all the modulated system with any other combination of modulation and demodulation carriers in (D.2) and u_q and recalling the *Euler's formula*, i.e.:

$$\cos(x) = \frac{e^{ix} + e^{-ix}}{2}, \qquad \sin(x) = \frac{e^{ix} - e^{-ix}}{2} \tag{D.7}$$

the transfer function of (D.1) can be calculated and it is equal to:

$$\mathbf{u}_{dq}(s) = e^{-sT_d} \mathbf{D} \mathbf{u}^*_{dq}(s) \tag{D.8}$$

where $\mathbf{D}$ is a rotating matrix defined as follow:

$$\mathbf{D} = \begin{bmatrix} \cos(\omega_{me} T_d) & \sin(\omega_{me} T_d) \\ -\sin(\omega_{me} T_d) & \cos(\omega_{me} T_d) \end{bmatrix} \tag{D.9}$$

BIBLIOGRAPHY

[1] United States Environmental Protection Agency. *Certification Summary Information Report*. 2017. URL: https://www3.epa.gov/otaq/datafiles/CSI-JTSLV00.0L13.PDF.

[2] Bloomberg. *The Electric-Car Boom Is So Real Even Oil Companies Say It's Coming*. 2017. URL: https://www.bloomberg.com/news/articles/2017-04-25/electric-car-boom-seen-triggering-peak-oil-demand-in-2030s.

[3] Daniele La Porta Arrobas, Kirsten Lori Hund, Michael Stephen Mccormick, Jagabanta Ningthoujam, and John Richard Drexhage. *The Growing Role of Minerals and Metals for a Low Carbon Future*. Washington, D.C., USA: World Bank Group, 2017.

[4] Gianmario Pellegrino, Thomas M. Jahns, Nicola Bianchi, Wen Soong, and Francesco Cupertino, eds. *The Rediscovery of Synchronous Reluctance and Ferrite Permanent Magnet Motors*. 1st ed. Springer International Publishing, 2016.

[5] Statista. *Neodymium oxide price worldwide from 2009 to 2025*. 2018. URL: https://www.statista.com/statistics/450152/global-reo-neodymium-oxide-price-forecast/.

[6] Y. Takaya et al. "The tremendous potential of deep-sea mud as a source of rare-earth elements." In: *Scientific Reports* 8 (Apr. 2018). DOI: 10.1038/s41598-018-23948-5.

[7] The Guardian. *Rare earth mining in China: the bleak social and environmental costs*. 2014. URL: https://www.theguardian.com/sustainable-business/rare-earth-mining-china-social-environmental-costs.

[8] Paul C. Krause, Oleg Wasynczuk, and Scott D. Sudhoff, eds. *Analysis of Electrical Machinery and Drive Systems*. 2nd ed. IEEE, 2002.

[9] M. Barcaro, N. Bianchi, and F. Magnussen. "Analysis and Tests of a Dual Three-Phase 12-Slot 10-Pole Permanent-Magnet Motor." In: *IEEE Transactions on Industry Applications* 46.6 (Nov. 2010), pp. 2355–2362. ISSN: 0093-9994. DOI: 10.1109/TIA.2010.2070784.

[10] A.E. Fitzgerald, A.E. Fitzgerald, C. Kingsley, and S.D. Umans. *Electric Machinery*. McGraw-Hill International Edition. McGraw-Hill, 2003. ISBN: 9780071230100.

[11] W. Leonhard. *Control of Electrical Drives*. Springer Berlin Heidelberg, 2012. ISBN: 9783642976469.

[12] W. C. Duesterhoeft, M. W. Schulz, and E. Clarke. "Determination of Instantaneous Currents and Voltages by Means of Alpha, Beta, and Zero Components." In: *Transactions of the American Institute of Electrical Engineers* 70.2 (July 1951), pp. 1248–1255. ISSN: 0096-3860. DOI: 10.1109/T-AIEE.1951.5060554.

[13] D. Mingardi, M. Morandin, S. Bolognani, and N. Bianchi. "On the Proprieties of the Differential Cross-Saturation Inductance in Synchronous Machines." In: *IEEE Transactions on Industry Applications* 53.2 (Mar. 2017), pp. 991–1000. ISSN: 0093-9994. DOI: 10.1109/TIA.2016.2622220.

[14] M. Barcaro, N. Bianchi, and F. Magnussen. "Remarks on Torque Estimation Accuracy in Fractional-Slot Permanent-Magnet Motors." In: *IEEE Transactions on Industrial Electronics* 59.6 (June 2012), pp. 2565–2572. ISSN: 0278-0046. DOI: 10.1109/TIE.2011.2160517.

[15] D.C. White and H.H. Woodson. *Electromechanical Energy Conversion*. The M.I.T. core curriculum program in electrical engineering. Wiley, 1959.

[16] F. Tinazzi and M. Zigliotto. "Torque Estimation in High-Efficency IPM Synchronous Motor Drives." In: *IEEE Transactions on Energy Conversion* 30.3 (Sept. 2015), pp. 983–990. ISSN: 0885-8969. DOI: 10.1109/TEC.2015.2408214.

[17] K. Liu and Z. Q. Zhu. "Mechanical Parameter Estimation of Permanent-Magnet Synchronous Machines With Aiding From Estimation of Rotor PM Flux Linkage." In: *IEEE Transactions on Industry Applications* 51.4 (July 2015), pp. 3115–3125. ISSN: 0093-9994. DOI: 10.1109/TIA.2015.2399615.

[18] X. Zhang and Z. Li. "Sliding-Mode Observer-Based Mechanical Parameter Estimation for Permanent Magnet Synchronous Motor." In: *IEEE Transactions on Power Electronics* 31.8 (Aug. 2016), pp. 5732–5745. ISSN: 0885-8993. DOI: 10.1109/TPEL.2015.2495183.

[19] S. Bolognani, L. Ortombina, F. Tinazzi, and M. Zigliotto. "Model sensitivity of fundamental-frequency-based position estimators for sensorless pm and reluctance synchronous motor drives." In: *IEEE Transactions on Industrial Electronics* 65.1 (Jan. 2018), pp. 77–85. ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2716902.

[20] T. Tuovinen, M. Hinkkanen, and J. Luomi. "Analysis and Design of a Position Observer With Resistance Adaptation for Synchronous Reluctance Motor Drives." In: *IEEE Transactions on Industry Applications* 49.1 (Jan. 2013), pp. 66–73. ISSN: 0093-9994. DOI: 10.1109/TIA.2012.2228616.

[21] 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.

[22] 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 (May 2017), pp. 2120–2129. ISSN: 0093-9994. DOI: 10.1109/TIA.2016.2644624.

[23] D. D. Reigosa, J. M. Guerrero, A. B. Diez, and F. Briz. "Rotor Temperature Estimation in Doubly-Fed Induction Machines Using Rotating High-Frequency Signal Injection." In: *IEEE Transactions on Industry Applications* 53.4 (July 2017), pp. 3652–3662. ISSN: 0093-9994. DOI: 10.1109/TIA.2017.2684742.

[24] F. Baneira, A. G. Yepes, O. Lopez, and J. Doval-Gandoy. "Estimation Method of Stator Winding Temperature for Dual Three-Phase Machines Based on DC-Signal Injection." In: *IEEE Transactions on Power Electronics* 31.7 (July 2016), pp. 5141–5148. ISSN: 0885-8993. DOI: 10.1109/TPEL.2015.2479410.

[25] Y. Inoue, Y. Kawaguchi, S. Morimoto, and M. Sanada. "Performance Improvement of Sensorless IPMSM Drives in a Low-Speed Region Using Online Parameter Identification." In: *IEEE Transactions on Industry Applications* 47.2 (Mar. 2011), pp. 798–804. ISSN: 0093-9994. DOI: 10.1109/TIA.2010.2101994.

[26] D. Q. Dang, M. S. Rafaq, H. H. Choi, and J. W. Jung. "Online Parameter Estimation Technique for Adaptive Control Applications of Interior PM Synchronous Motor Drives." In: *IEEE Transactions on Industrial Electronics* 63.3 (Mar. 2016), pp. 1438–1449. ISSN: 0278-0046. DOI: 10.1109/TIE.2015.2494534.

[27] M. S. Rafaq, F. Mwasilu, J. Kim, H. H. Choi, and J. W. Jung. "Online Parameter Identification for Model-Based Sensorless Control of Interior Permanent Magnet Synchronous Machine." In: *IEEE Transactions on Power Electronics* 32.6 (June 2017), pp. 4631–4643. ISSN: 0885-8993. DOI: 10.1109/TPEL.2016.2598731.

[28] G. Zanuso, L. Peretti, and P. Sandulescu. "Stator reference frame approach for DC injection-based stator resistance estimation in electric drives." In: *2015 IEEE 11th International Conference on Power Electronics and Drive Systems*. June 2015, pp. 867–872. DOI: 10.1109/PEDS.2015.7203391.

[29] G. Feng, C. Lai, K. Mukherjee, and N. C. Kar. "Current Injection-Based Online Parameter and VSI Nonlinearity Estimation for PMSM Drives Using Current and Voltage DC Components." In:

IEEE Trans. Transport. Electrific 2.2 (June 2016), pp. 119–128. DOI: 10.1109/TTE.2016.2538180.

[30] 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* (2018), pp. 1–1. ISSN: 0093-9994. DOI: 10.1109/TIA.2018.2819961.

[31] S. Bolognani, S. Calligaro, R. Petrella, and M. Tursini. "Sensorless Control of IPM Motors in the Low-Speed Range and at Standstill by HF Injection and DFT Processing." In: *IEEE Transactions on Industry Applications* 47.1 (Jan. 2011), pp. 96–104. ISSN: 0093-9994. DOI: 10.1109/TIA.2010.2090317.

[32] M. J. Corley and R. D. Lorenz. "Rotor position and velocity estimation for a salient-pole permanent magnet synchronous machine at standstill and high speeds." In: *IEEE Transactions on Industry Applications* 34.4 (July 1998), pp. 784–789. ISSN: 0093-9994. DOI: 10.1109/28.703973.

[33] S. Shinnaka. "A New Speed-Varying Ellipse Voltage Injection Method for Sensorless Drive of Permanent-Magnet Synchronous Motors With Pole Saliency - New PLL Method Using High-Frequency Current Component Multiplied Signal." In: *IEEE Transactions on Industry Applications* 44.3 (May 2008), pp. 777–788. ISSN: 0093-9994. DOI: 10.1109/TIA.2008.921446.

[34] K. Lau, G.C. Goodwin, and R.T. M'Closkey. "Properties of modulated and demodulated systems with implications to feedback limitations." In: *Automatica* 41.12 (2005), pp. 2123–2129. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2005.07.009.

[35] G.F. Franklin, J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson Education Limited, 2015. ISBN: 9781292068916.

[36] 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 (July 2016), pp. 3083–3092. ISSN: 0093-9994. DOI: 10.1109/TIA.2016.2535413.

[37] 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.

[38] 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 (May 2017), pp. 2120–2129. ISSN: 0093-9994.

[39] E. Armando, R.I. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli. "Experimental Identification of the Magnetic Model of Synchronous Machines." In: *IEEE Transactions on Industry Applications* 49.5 (Sept. 2013), pp. 2116–2125. ISSN: 0093-9994. DOI: 10.1109/TIA.2013.2258876.

[40] G. Pellegrino, B. Boazzo, and T. M. Jahns. "Magnetic Model Self-Identification for PM Synchronous Machine Drives." In: *IEEE Transactions on Industry Applications* 51.3 (May 2015), pp. 2246–2254. ISSN: 0093-9994. DOI: 10.1109/TIA.2014.2365627.

[41] S. Wiedemann, R. M. Kennel, S. Hall, and M. Alaküla. "Dynamic Testing characterization of a synchronous reluctance machine." In: *2016 Eleventh International Conference on Ecological Vehicles and Renewable Energies (EVER)*. Apr. 2016, pp. 1–7. DOI: 10.1109/EVER.2016.7476426.

[42] M.T. Hagan, H.B. Demuth, M.H. Beale, and O. De Jesús. *Neural Network Design*, 2nd. 978-0-9717321-1-7. 2014. URL: http://hagan.okstate.edu/nnd.html.

[43] D. Lowe. "Adaptive radial basis function nonlinearities, and the problem of generalisation." In: *Artificial Neural Networks, 1989., First IEE International Conference on (Conf. Publ. No. 313)*. Oct. 1989, pp. 171–175.

[44] N. Urasaki, T. Senjyu, and K. Uezato. "A novel calculation method for iron loss resistance suitable in modeling permanent-magnet synchronous motors." In: *IEEE Transactions on Energy Conversion* 18.1 (Mar. 2003), pp. 41–47. ISSN: 0885-8969. DOI: 10.1109/TEC.2002.808329.

[45] M. Sipser. *Introduction to the Theory of Computation*, 3rd. 978-1-133-18779-0. 2013.

[46] F. Tinazzi, P. Sandulescu, L. Peretti, and M. Zigliotto. "On the true maximum efficiency operations of synchronous motor drives." In: *2017 IEEE 12th International Conference on Power Electronics and Drive Systems (PEDS)*. Dec. 2017, pp. 1, 043–1, 048. DOI: 10.1109/PEDS.2017.8289154.

[47] R. Antonello, M. Carraro, and M. Zigliotto. "Maximum-Torque-Per-Ampere Operation of Anisotropic Synchronous Permanent-Magnet Motors Based on Extremum Seeking Control." In: *IEEE Transactions on Industrial Electronics* 61.9 (Sept. 2014), pp. 5086–5093. ISSN: 0278-0046. DOI: 10.1109/TIE.2013.2278518.

[48] S. Bolognani, R. Petrella, A. Prearo, and L. Sgarbossa. "Automatic Tracking of MTPA Trajectory in IPM Motor Drives Based on AC Current Injection." In: *IEEE Transactions on Industry Applications* 47.1 (Jan. 2011), pp. 105–114. ISSN: 0093-9994. DOI: 10.1109/TIA.2010.2090842.

[49] S. Kim, Y. Yoon, S. 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 (Jan. 2013), pp. 488–497. ISSN: 0885-8993. DOI: 10.1109/TPEL.2012.2195203.

[50] S. Buso, T. Caldognetto, and D. I. Brandao. "Comparison of oversampled current controllers for microgrid utility interface converters." In: *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*. Sept. 2015, pp. 6888–6895. DOI: 10.1109/ECCE.2015.7310625.

[51] J. Holtz. "Advanced PWM and Predictive Control - An Overview." In: *IEEE Transactions on Industrial Electronics* 63.6 (June 2016), pp. 3837–3844. ISSN: 0278-0046. DOI: 10.1109/TIE.2015.2504347.

[52] M. P. Kazmierkowski and L. Malesani. "Current control techniques for three-phase voltage-source PWM converters: a survey." In: *IEEE Transactions on Industrial Electronics* 45.5 (Oct. 1998), pp. 691–703. ISSN: 0278-0046. DOI: 10.1109/41.720325.

[53] L. R. Limongi, R. Bojoi, G. Griva, and A. Tenconi. "Digital current-control schemes." In: *IEEE Industrial Electronics Magazine* 3.1 (Mar. 2009), pp. 20–31. ISSN: 1932-4529. DOI: 10.1109/MIE.2009.931894.

[54] A. Vagati. "Synchronous reluctance electrical motor having a low torque-ripple design." Patent US 5818140A (US). Oct. 1998.

[55] L. Corradini, D. Maksimovic, P. Mattavelli, and R. Zane. *Digital Control of High-Frequency Switched-Mode Power Converters*. IEEE Press Series on Power Engineering. Wiley, 2015. ISBN: 9781118935101. URL: https://books.google.it/books?id=FMfCCQAAQBAJ.

[56] M. Hinkkanen, H. Asad Ali Awan, Z. Qu, T. Tuovinen, and F. Briz. "Current Control for Synchronous Motor Drives: Direct Discrete-Time Pole-Placement Design." In: *IEEE Transactions on Industrial Applications* 52.2 (Mar. 2016), pp. 1530–1541. ISSN: 0093-9994. DOI: 10.1109/TIA.2015.2495288.

[57] Kiam Heong Ang, G. Chong, and Yun Li. "PID control system analysis, design, and technology." In: *IEEE Transactions on Control Systems Technology* 13.4 (July 2005), pp. 559–576. ISSN: 1063-6536. DOI: 10.1109/TCST.2005.847331.

[58] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill, 2009.

[59] J. Holtz and S. Stadtfeld. "Field-Oriented Control by Forced Motor Currents in a Voltage Fed Inverter Drive." In: *Control in Power Electronics and Electrical Drives 1983*. Ed. by R. Zwicky. IFAC Symposia Series. Oxford: Pergamon, 1984, pp. 103–110. ISBN: 978-0-08-030536-3. DOI: 10.1016/B978-0-08-030536-3.50019-2. URL: http://www.sciencedirect.com/science/article/pii/B9780080305363500192.

[60] T. Geyer. *Model Predictive Control of High Power Converters and Industrial Drives*. Wiley, 2016. ISBN: 9781119010869. URL: https://books.google.it/books?id=KtgQDQAAQBAJ.

[61] T. Geyer, G. Papafotiou, and M. Morari. "Model Predictive Direct Torque Control—Part I: Concept, Algorithm, and Analysis." In: *IEEE Transactions on Industrial Electronics* 56.6 (June 2009), pp. 1894–1905. ISSN: 0278-0046. DOI: 10.1109/TIE.2008.2007030.

[62] H. Miranda, P. Cortes, J. I. Yuz, and J. Rodriguez. "Predictive Torque Control of Induction Machines Based on State-Space Models." In: *IEEE Transactions on Industrial Electronics* 56.6 (June 2009), pp. 1916–1924. ISSN: 0278-0046. DOI: 10.1109/TIE.2009.2014904.

[63] G. Papafotiou, J. Kley, K. G. Papadopoulos, P. Bohren, and M. Morari. "Model Predictive Direct Torque Control—Part II: Implementation and Experimental Evaluation." In: *IEEE Transactions on Industrial Electronics* 56.6 (June 2009), pp. 1906–1915. ISSN: 0278-0046. DOI: 10.1109/TIE.2008.2007032.

[64] Y. Zhang and H. Yang. "Model Predictive Torque Control of Induction Motor Drives With Optimal Duty Cycle Control." In: *IEEE Transactions on Power Electronics* 29.12 (Dec. 2014), pp. 6593–6603. ISSN: 0885-8993. DOI: 10.1109/TPEL.2014.2302838.

[65] 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 (June 2009), pp. 1925–1936. ISSN: 0278-0046. DOI: 10.1109/TIE.2008.2007547.

[66] 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 (Aug. 2016), pp. 5176–5185. ISSN: 0278-0046. DOI: 10.1109/TIE.2016.2536581.

[67] M. Preindl and S. Bolognani. "Model Predictive Direct Speed Control with Finite Control Set of PMSM Drive Systems." In: *IEEE Transactions on Power Electronics* 28.2 (Feb. 2013), pp. 1007–1015. ISSN: 0885-8993. DOI: 10.1109/TPEL.2012.2204277.

[68] T. Geyer and D. E. Quevedo. "Multistep Finite Control Set Model Predictive Control for Power Electronics." In: *IEEE Transactions on Power Electronics* 29.12 (Dec. 2014), pp. 6836–6846. ISSN: 0885-8993. DOI: 10.1109/TPEL.2014.2306939.

[69] D. Su, C. Zhang, and Y. Dong. "Finite-State Model Predictive Current Control for Surface-Mounted Permanent Magnet Synchronous Motors Based on Current Locus." In: *IEEE Access* 5 (2017), pp. 27366–27375. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2771418.

[70] F. Morel, X. Lin-Shi, J. Retif, B. Allard, and C. Buttay. "A Comparative Study of Predictive Current Control Schemes for a Permanent-Magnet Synchronous Machine Drive." In: *IEEE Transactions on Industrial Electronics* 56.7 (July 2009), pp. 2715–2728. ISSN: 0278-0046. DOI: 10.1109/TIE.2009.2018429.

[71] B. Hassibi and H. Vikalo. "On the sphere-decoding algorithm I. Expected complexity." In: *IEEE Transactions on Signal Processing* 53.8 (Aug. 2005), pp. 2806–2818.

[72] T. Geyer, P. Karamanakos, and R. Kennel. "On the benefit of long-horizon direct model predictive control for drives with LC filters." In: *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*. Sept. 2014, pp. 3520–3527. DOI: 10.1109/ECCE.2014.6953879.

[73] P. Karamanakos, T. Geyer, and R. Kennel. "Constrained long-horizon direct model predictive control for power electronics." In: *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*. Sept. 2016, pp. 1–8. DOI: 10.1109/ECCE.2016.7854957.

[74] E. Liegmann, P. Karamanakos, T. Geyer, T. Mouton, and R. Kennel. "Long-horizon direct model predictive control with active balancing of the neutral point potential." In: *2017 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*. Sept. 2017, pp. 89–94. DOI: 10.1109/PRECEDE.2017.8071274.

[75] P. Karamanakos, T. Geyer, and R. Kennel. "Reformulation of the long-horizon direct model predictive control problem to reduce the computational effort." In: *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*. Sept. 2014, pp. 3512–3519. DOI: 10.1109/ECCE.2014.6953878.

[76] Å. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996. DOI: 10.1137/1.9781611971484. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611971484. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611971484.

[77] P. Karamanakos, T. Geyer, and R. Kennel. "Suboptimal search strategies with bounded computational complexity to solve long-horizon direct model predictive control problems." In: *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*. Sept. 2015, pp. 334–341. DOI: 10.1109/ECCE.2015.7309707.

[78] L. Babai. "On Lovász' lattice reduction and the nearest lattice point problem." In: *Combinatorica* 6.1 (1986), pp. 1–13.

[79] T. Geyer and D. E. Quevedo. "Performance of multistep finite control set model predictive control for power electronics." In: *IEEE Transactions on Power Electronics* 30.3 (Mar. 2015), pp. 1633–1644.

[80] European Commission. *Smart cities and communities - European innovation partnership*. 2012. URL: https://www.smartcities.at/europe/eu-initiatives/eip-smart-cities-and-communities.

[81] N. Bedetti, S. Calligaro, and R. Petrella. "Self-Commissioning of Inverter Dead-Time Compensation by Multiple Linear Regression Based on a Physical Model." In: *IEEE Transactions on Industry Applications* 51.5 (Sept. 2015), pp. 3954–3964. ISSN: 0093-9994. DOI: 10.1109/TIA.2015.2436882.

[82] R. Antonello, F. Tinazzi, and M. Zigliotto. "Benefits of Direct Phase Voltage Measurement in the Rotor Initial Position Detection for Permanent-Magnet Motor Drives." In: *IEEE Transactions on Industrial Electronics* 62.11 (Nov. 2015), pp. 6719–6726. ISSN: 0278-0046. DOI: 10.1109/TIE.2015.2448514.

[83] M. Schubert and R. W. De Doncker. "Instantaneous Phase Voltage Sensing in PWM Voltage Source Inverters." In: *IEEE Transactions on Power Electronics* PP.99 (2017), pp. 1–1. ISSN: 0885-8993. DOI: 10.1109/TPEL.2017.2759662.

[84] Y. Wang, Y. Xu, N. Niimura, B. D. Rudolph, and R. D. Lorenz. "Using Volt-Second Sensing to Directly Improve Torque Accuracy and Self-Sensing at Low Speeds." In: *IEEE Transactions on Industry Applications* 53.5 (Sept. 2017), pp. 4472–4482. ISSN: 0093-9994. DOI: 10.1109/TIA.2017.2702752.