



Technische Universität München  
TUM School of Engineering and Design

# Learning-Based Nonlinear Model Predictive Control for Electrical Drives

Issa Mohammad Issa Hammoud

Vollständiger Abdruck der von der TUM School of Engineering and Design der  
Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Hon.-Prof. Dr.-Ing. Martin Otter

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Dr. h.c. Ralph Kennel
2. Prof. Silverio Bolognani
3. Prof. Dr.-Ing. Mario Pacas

Die Dissertation wurde am 16.10.2023 bei der Technischen Universität München  
eingereicht und durch die TUM School of Engineering and Design am 30.04.2024  
angenommen.



*To my immensely loving parents.*

---

*Issa*



## Acknowledgment

---

The thesis at hand was written during my time as a doctoral candidate at the Chair of High-Power Converter Systems (previously the Institute for Electrical Drive Systems and Power Electronics) at the Technical University of Munich (TUM), Munich, Germany, where the research associated with it has been concurrently carried out in the automotive industry at the Powertrain Mechatronics Control Engineering Excellence Cluster at IAV GmbH in Gifhorn, Germany.

*“Instruction is good for a child; but example is worth more.”*

Alexandre Dumas | Twenty Years After

It is no overstatement that my foremost thanks go to my doctoral thesis supervisor Prof. Dr.-Ing. Dr. h.c. Ralph Kennel who has whetted my appetite generally to become an electrical drive engineer already in the first lecture of my master studies at TUM, and has boosted my interest towards predictive control. My deepest sense of gratitude to Prof. Kennel for giving me the freedom to learn and do research in the field of study that I loved. I have highly appreciated each minute with Prof. Kennel, as after each time I met him I was left with new learning and inspiration. Moreover, I would like to extend my appreciation to Prof. Silverio Bolognani and Prof. Dr.-Ing. Mario Pacas for agreeing to co-examine my dissertation and for their interest in my work, and Hon.-Prof. Dr.-Ing. Martin Otter for taking over the chairmanship of the examination committee.

*“Mentoring is a brain to pick, an ear to listen, and a push in the right direction.”*

John Crosby

I am very grateful for my team leader at IAV Dr.-Ing Thimo Oehlschlägel and my direct supervisor Dr.-Ing Sebastian Hentzelt. You were not only the best supervisors I could wish for, but even mentors to me during this journey. You were always there to listen, guide, and support. I learned a lot from your remarkable resourcefulness. You very much shaped the professional who I am today and I was very blessed to have worked with such beautiful minds as yours.

*“We didn’t realize we were making memories, we just knew we were having fun.”*

Winnie the Pooh

I would like to thank my colleagues at IAV: Dr.-Ing Manus Thiel, Dr.-Ing Erik Schulz, M.Sc. Horacio Calderon, Dr.-Ing Peter Kotzyba, Dr.-Ing Emre Arslantas, Dr.-Ing Felix Nolteernsting, M.Sc. Tim Zieger, Dr.-Ing Marc Wissing, Dr.-Ing Christopher Schindlbeck, Dr.-Ing Thomas Orlik, and Dr.-Ing Michael Homann for the very positive and supportive work environment, and a special thank you for M.Sc. Ke Xu for his sense of humour and for making the working times very joyful.

*“When you are grateful, fear disappears and abundance appears.”*

Tony Robbins

I would also like to acknowledge the following persons without whom this work wouldn't have been accomplished in its current form: Dr.-Ing Christian Wesemeyer for his immense trust and belief on me since my first months in Germany back in 2015, Mrs. Marita Behrendt (may her soul rest in peace) for hosting me in her house for the most of my Ph.D. duration in Gifhorn and for being as a grandmother to me, Prof. Dr.-Ing. habil. Christoph M. Hackl for allowing me to use his laboratory equipment and for taking the time to give me constructive feedback on part of my work, all the students whom I worked with on their internships and master theses and especially Mr. Hang Yu for his excellent work regarding machine modelling and the synchronization of the measurements at the test bench, M.Sc. Martin Riedel for being a role model in the extremely impressive efficient and consistent hard work in the Hyperloop project during my master studies, for my best friends M.Sc. Abdelrahman Budair, M.Sc. Rabee Jibrin and Omar Alaref for the amazingly supportive friendship and for challenging me to eventually become a marathon finisher during my doctoral thesis times, my beloved sister Faten and her husband Haitham and my brothers Ahmad and Abdulfattah for covering my absence over the years.

*“The price of anything is the amount of life you exchange for it.”*

Henry David Thoreau

Once I finished high-school, I faced the cruel fact that following my dream education and passion would inherently mean living far away from the most precious persons in my life, my parents, for quiet some time. This time has ended up to be almost 12 years by now, in which my parents have always pushed me to go further to the best for my own future, despite that this was definitely the extreme opposite on them. I hope that what I did can put a smile on their beautiful faces and make them proud of me, despite that nothing can compensate all the nights I spent away.

Finally, special thank you goes to my wife Lujain for the unconditional love and the immense support, for following me with your bike in all of the long training runs for the marathon, for understanding that I had many personal deadlines already in our first year of marriage, for having my back in the hard times. You played a great deal of difference.

Issa Hammoud  
Gihorn, Germany

## Abstract

---

The thesis at hand deals with advanced model predictive control (MPC) schemes for electrical drive systems, which is gaining an ever-increasing genuine interest within the power electronics and electrical drives community. MPC is an uprising control method that has been shown to exhibit superior dynamic performance in comparison to the incumbent classical state-of-the-art control techniques typically used in electrical drive applications. Nevertheless, MPC possesses its own set of challenges, mainly: identifying the accurate mathematical models that accurately represent the nonlinear plants under control, and tackling its computational complexity.

As the name implies, the *control* performance of MPC schemes is highly dependant on the mathematical *model* used to *predict* the system's response to a given input. The content of this dissertation is twofold, including modelling and control. On the one hand, it starts with deriving a thorough nonlinear model of the electrical drive system including the inverter and the electrical machine based on the conventional approaches and physical laws. This is followed by presenting data-driven modelling techniques as an alternative, however, purely from measurements and without bringing physical knowledge. The latter proposal is mainly motivated by two factors: its advantageous less dependency on a prior knowledge, which makes it useful in capturing the hard-to-model or even unmodelled dynamics by first-principles and physical laws, as well as the hope that it yields a better prediction accuracy in comparison with the conventional modelling approaches.

On the other hand, the obtained models are incorporated to design predictive controllers. This part starts with illustrating the concept of predictive control in the context of electrical drive applications and its direct and indirect variants (i.e. without or with a modulator, respectively), with developing appropriate computationally-efficient solvers to allow real-time implementation of the different predictive control schemes. Throughout this part, the characteristics, benefits and challenges of each variant are highlighted. Moreover, the effects of modelling mismatches in the closed-loop MPC performance are pinpointed with a comprehensive literature survey regarding the most known methods to mitigate these effects in the control of electrical drive systems. Moreover, an offset-free MPC scheme is presented by using the incremental state-space formulation of the constrained optimal control problem (OCP), which embeds an error integrating functionality to the control scheme. Furthermore, a novel learning-based MPC scheme that incorporates a data-driven model is proposed. Finally, the use of artificial intelligence (AI) algorithms to mimic the beneficial, but computationally complex MPC polices in a computationally-implementable manner is proposed. Whether the computational complexity

is caused by the nature of the optimization problem as in the case of the mixed-integer long-horizon direct MPC problem or caused by the use of complex nonlinear models as the case with the learning-based indirect MPC scheme, computationally-implementable neural network (NN)s proved to capture these control policies in real-time-capable manner. By this, the use of AI algorithms to alleviate some of the MPC's shortcomings for electrical drive applications is outlined.



---

# Table of Contents

---

<b>Acknowledgment</b>	<b>I</b>
<b>1 Introduction</b>	<b>1</b>
1.1 <i>What is MPC? definition and a historical overview.</i>	2
1.2 <i>What makes MPC fit power electronic systems more than conventional methods?</i>	4
1.2.1 <i>The characteristics and limitations of the conventional control methods</i>	4
1.2.2 <i>The advantageous characteristics of MPC</i>	5
1.3 <i>Who did MPC first in the power electronics community?</i>	7
1.4 <i>Challenges and active research areas</i>	8
1.5 <i>Motivation and contributions</i>	10
<b>2 Conventional modelling of an electrical drive</b>	<b>13</b>
2.1 <i>Components of an electrical drive system</i>	13
2.2 <i>Coordinate transformation</i>	14
2.3 <i>Generic model of a 2-levels inverter</i>	16
2.4 <i>Pre-modelling</i>	18
2.4.1 <i>Inverter nonlinearity compensation: conventional method</i>	19
2.4.1.1 <i>Problem statement</i>	19
2.4.1.2 <i>Experimental investigation of the VSI nonlinear effects</i>	19
2.4.1.3 <i>Feedforward inverter nonlinearity compensation</i>	19
2.4.1.4 <i>Validation of the obtained compensation curves</i>	21
2.4.2 <i>Angle delay compensation</i>	22
2.4.3 <i>Measurement synchronization</i>	23
2.5 <i>Generic model of PMSMs based on physical laws</i>	25
2.5.1 <i>Model derivation</i>	25
2.5.2 <i>Validation of the nonlinear PMSM model based on the physical laws</i>	28
<b>3 Data-driven modelling</b>	<b>31</b>
3.1 <i>Motivation and related work</i>	31
3.2 <i>Data-driven inverter nonlinearity compensation</i>	33
3.3 <i>Problem statement of data-driven model identification</i>	37
3.4 <i>Design of identification experiment</i>	38

3.5	Data-driven modelling of PMSMs via LSTM NNs . . . . .	40
3.5.1	Validation of the LSTM neural model . . . . .	41
3.6	Other potential data-driven modelling techniques . . . . .	43
3.6.1	Data-driven modelling of PMSMs via GPs . . . . .	43
3.6.2	Data-driven modelling of PMSMs using KO theory . . . . .	46
3.7	Conclusion . . . . .	47
<b>4</b>	<b>First-principles model predictive control</b>	<b>49</b>
4.1	Direct model predictive control . . . . .	50
4.1.1	One-step FCS-MPC . . . . .	55
4.1.1.1	Conventional one-step FCS-MPC . . . . .	55
4.1.1.2	Computationally-efficient one-step FCS-MPC . . . . .	56
4.1.1.3	Experimental results and observations . . . . .	58
4.1.1.4	Multiple-vector one-step FCS-MPC . . . . .	60
4.1.2	Long-horizon FCS-MPC . . . . .	61
4.1.2.1	Benefits of long-horizon FCS-MPC and the computational barrier . . . . .	61
4.1.2.2	Computationally-implementable approximated long-horizon FCS-MPC based on NNs . . . . .	63
4.2	Indirect MPC . . . . .	70
4.2.1	Indirect MPC: conceptual differences to direct MPC . . . . .	70
4.2.2	First-principles CCS-MPC . . . . .	72
4.2.2.1	Problem formulation and model-based controller architecture . . . . .	73
4.2.2.2	Numerical solver . . . . .	74
4.2.2.3	L-FP-CCS-MPC for an IPMSM . . . . .	76
4.2.2.4	L-FP-CCS-MPC for a SMPMSM . . . . .	76
4.2.2.5	Mitigation of modelling mismatch via integral action and parameters estimation: a review . . . . .	78
4.2.2.6	Offset-free linear CCS-MPC . . . . .	81
4.2.2.7	NL-FP-CCS-MPC for an IPMSM . . . . .	84
4.2.3	First-principles DB-MPC . . . . .	87
4.2.3.1	Linear DB-MPC . . . . .	87
4.2.3.2	Nonlinear DB-MPC . . . . .	88
<b>5</b>	<b>Learning-based model predictive control</b>	<b>91</b>
5.1	LB-CCS-MPC using the LSTM model . . . . .	91
5.1.1	Controller design and implementation . . . . .	91
5.1.2	Experimental results . . . . .	93
5.1.3	Benchmarking against the first-principles MPC . . . . .	96
5.1.3.1	Steady-state operation . . . . .	96
5.1.3.2	Transient operation . . . . .	97
<b>6</b>	<b>Conclusion and outlook</b>	<b>99</b>
6.1	Conclusion . . . . .	99
6.2	Potential future research directions . . . . .	102

---

<b>A</b>	<b>List of publications</b>	<b>105</b>
A.1	Journal papers . . . . .	105
A.2	Conference papers . . . . .	105
A.3	Awards . . . . .	106
A.4	Supervised and co-supervised theses . . . . .	106
A.5	List of symbols . . . . .	107
A.6	List of abbreviations . . . . .	114
<b>B</b>	<b>Test benches</b>	<b>117</b>
B.1	Test bench of the IPMSM . . . . .	117
B.2	Test bench of the SMPMSM . . . . .	118
<b>C</b>	<b>Classical PI-FOC</b>	<b>121</b>
	<b>List of Figures</b>	<b>124</b>
	<b>List of Tables</b>	<b>131</b>



# CHAPTER 1

---

## Introduction

---

*“The beginning is the most important part of the work.”*

Plato

This chapter is meant to establish the base of this dissertation from both the method as well as the application side. In a first step, an overview about the basic theoretical concept behind model predictive control (MPC), and a run-through regarding its development over the past four decades are given. After that, an insight about the nature and characteristics of power electronic systems, their needs and requirements which have to be fulfilled by the chosen control strategy are provided. Subsequently, the reasons that make MPC an appealing control strategy for power electronic and electrical drive applications are addressed, and the benefits that MPC could bring in contrast to the state-of-the-art methods are highlighted. Moreover, the most recent active research areas associated with MPC in the field of power electronics and electrical drives are reviewed, and the contributions of this dissertation are pointed out. In a nutshell, this chapter deals with the following points:

- *What is MPC? definition and a historical overview*
- *What makes MPC fit power electronic systems more than conventional methods?*
- *Who did MPC first in the power electronics community?*
- *What are the challenges associated with MPC in power electronic applications, and what are the current active research areas?*

Once having these foundations established, the contributions throughout this thesis to advance the existing body of knowledge are summarized.

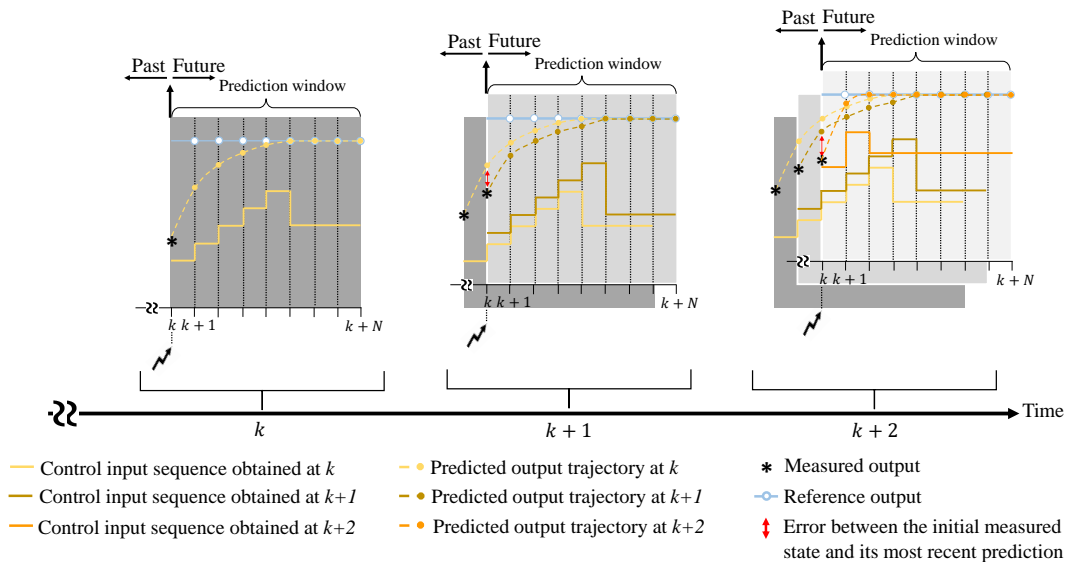
## 1.1 What is MPC? definition and a historical overview.

“When you make predictions, either you are lucky or wrong.”

Niels Bohr

MPC makes an explicit use of a mathematical *model* that represents the plant under control in order to predict the states of the system as a response to a given input trajectory. In MPC, a finite-horizon open-loop optimal control problem (OCP) subject to the corresponding system dynamics, constraints, and initial conditions is formulated. Typically, the solution of the OCP is determined in real-time at each sampling interval using advanced numerical optimization methods. The solution yields the predicted parameterized optimal control input sequence. Applying only the first control input from that sequence to the plant, and repeating the optimization with the new obtained measurements in a receding horizon manner, assures a feedback loop to adapt to changes in the process or system behaviour [1, 2]. To this end, MPC combines features from the state feedback regulators, advanced numerical optimization methods, and optimal control theory. From the first, it takes the advantage of having a feedback loop, and hence, the ability to counteract disturbances. From the latter, it takes the ability to foresee, consider, and consequently optimize the system response in advance.

The general concept of discrete-time MPC is depicted in Fig. 1.1 at three consecutive discrete time instants  $k$ ,  $k + 1$ , and  $k + 2$ . Due to the inevitable modelling errors, perturbations, and disturbances that act on the dynamical system, a discrepancy between the most recent system output at the beginning of each sampling interval  $k$ , and its most recent prediction made at  $k - 1$ , occurs.



**Figure 1.1:** The general concept of discrete-time model predictive control with a prediction horizon length  $N$ .

Consider  $\mathbf{u} \in \mathbb{R}^{N_u}$  representing the inputs of the system,  $\mathbf{x} \in \mathbb{R}^{N_x}$ , and  $\mathbf{y} \in \mathbb{R}^{N_y}$  denoting the states and outputs, respectively, with  $N_u$ ,  $N_x$ , and  $N_y$  as the number of inputs, states, and

outputs, respectively. The nonlinear continuous-time state-space model can be expressed in the general form as

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}_c(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \tilde{\mathbf{x}}_0 \quad (1.1a)$$

$$\mathbf{y}(t) = \mathbf{h}_c(\mathbf{x}(t)), \quad (1.1b)$$

where  $\tilde{\mathbf{x}}_0$  is the initial state,  $\mathbf{f}_c(\cdot)$  and  $\mathbf{h}_c(\cdot)$  are the continuous-time system dynamics and output functions, respectively, and  $t$  denotes the time. For digital implementation of the control schemes in power electronic and electrical drive applications, a discretized version of the model is to be used. By far the mostly used discretization method in these fields is the explicit Euler forward discretization method, in which for sufficiently small sampling period  $T_s$ , the time-derivative of the system dynamics is approximated as

$$\frac{d\mathbf{x}(t)}{dt} = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{T_s}, \quad (1.2)$$

with the discrete time index  $k$  representing the current discrete time instant as  $t_k = kT_s$ . The use of this simple discretization method in the community is driven by its acceptable accuracy with the beneficial low computational complexity that suit this class of applications, even though that more complex discretization methods do provide better accuracy [3, 4]. To obtain the discrete-time state-space model, this discretization method is simply applied by integrating the system dynamics from  $t = kT_s$  to  $t = (k + 1)T_s$ , and by keeping the continuous-time input value  $\mathbf{u}(t)$  constant over the sampling period and equals to its value at the beginning of the sampling interval  $\mathbf{u}_k$ . The general nonlinear discrete-time state-space model can be expressed as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \tilde{\mathbf{x}}_0, \quad (1.3a)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k), \quad (1.3b)$$

with  $\mathbf{f}(\cdot)$  and  $\mathbf{h}(\cdot)$  as the discrete-time system dynamics and output functions, respectively. In MPC, the following general dynamic optimization problem is to be solved

$$\min_{\mathbf{U}_k} J(\mathbf{U}) = \frac{1}{2} \sum_{i=0}^{N-1} (\mathbf{y}_{k+1+i} - \mathbf{r}_{k+1+i})^\top \mathbf{Q} (\mathbf{y}_{k+1+i} - \mathbf{r}_{k+1+i}) \quad (1.4a)$$

$$+ (\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1})^\top \mathbf{R} (\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1}), \quad (1.4b)$$

$$\text{s.t.} \quad 1.3, \quad (1.4b)$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{k+1+i} \leq \mathbf{x}_{max}, \quad \forall i \in \{0, \dots, N-1\}, \quad (1.4c)$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{k+i} \leq \mathbf{u}_{max}, \quad \forall i \in \{0, \dots, N-1\}, \quad (1.4d)$$

where  $\mathbf{U}_k = (\mathbf{u}_k^\top, \dots, \mathbf{u}_{k+N-1}^\top)^\top$ ,  $\mathbf{Q} = \mathbf{Q}^\top \succeq 0$  is the weighting matrix to penalize the deviation of the outputs from their references,  $\mathbf{R} = \mathbf{R}^\top \succ 0$  is the weighting matrix to penalize the rate of change of the control inputs,  $\mathbf{r}_k$  is the vector that contains the reference outputs, and  $N$  represents the prediction horizon length.

In industrial applications, the very first implementations of MPC have emerged in the refining and chemical process industries [5, 6]. In 1979, engineers from Shell Oil Company have

published their unconstrained multi-variable control algorithm, which they called at that time dynamic matrix control (DMC) as proposed in [7, 8]. Work regarding the internal model control followed in 1982 [9]. In late 1980s, MPC has been successfully used in many other different industrial applications as in [10, 11], where it has been applied to a cement mill, a spray drying tower, and a compliant robotic arm (with the shortest sampling frequency in the latter ranging from 30 to 100 Hz). In [11], even though sample intervals of the different applications had a range factor of 10,000:1, the performance was consistently good. The application area has been drastically expanded to include many other applications, such as: robotics [12], regulating patient's anesthetic state [13], and many other industrial applications [14]. Here, it is worth mentioning that a common property of these applications is the relatively long sampling times when compared to those needed in power electronic applications.

In the last 50 years, a steep improvement of the computational abilities of the digital microprocessors has occurred. Consequently, advanced control techniques has piqued the interest to be applied in the field of power electronics and electrical drives, as the new microprocessors and field programmable gate arrays (FPGAs) can provide the needed computational power to solve an OCP within some tens and even sub-ten of microseconds. On the methodological side, MPC has undergone extensive investigation and gradual improvements to be considered as a very convenient alternative to the state-of-the-art field-oriented control based on proportional-integral controllers (PI-FOC) for electrical drive applications [15–21].

## 1.2 What makes MPC fit power electronic systems more than conventional methods?

The degree of matching between what the method can provide, and what the application requires and needs is of an essential importance in engineering research. This becomes even more crucial when proposing an alternative to a well-functioning state-of-the-art method, such as the classical proportional-integral (PI) regulators to control power electronic systems and electrical machines.

The intention behind this section is to point out the motivation of using MPC for electrical drive applications, and to point out plainly where it can outperform the conventional methods such as PI-FOC and direct torque control (DTC).

### 1.2.1 The characteristics and limitations of the conventional control methods

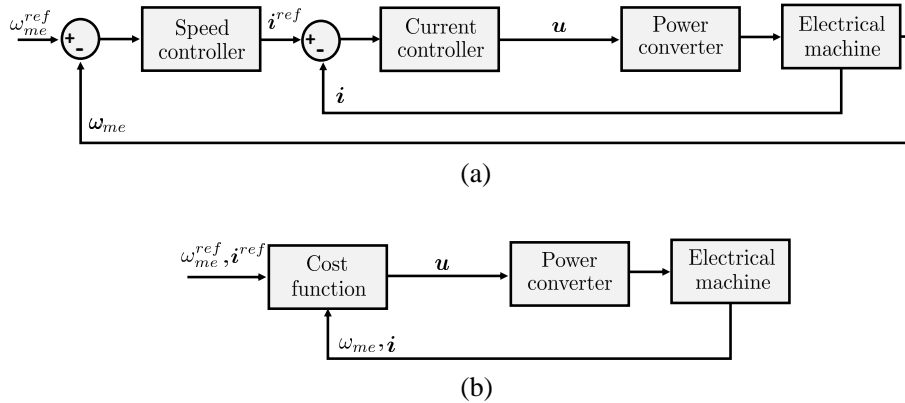
*“You can disagree without being disagreeable.”*

Ruth Bader Ginsburg

PI-FOC has been considered as the leading state-of-the-art control method for electrical drive systems till more advanced and convenient control approaches appeared to compete and even outperform it in some aspects, among others are the DTC [22, 23] and MPC [15, 16, 18]. The PI-FOC control is characterized as a simple and linear control, however; applied on nonlinear systems. Taking the simultaneous speed and current control of an electrical machine as



an example, the controller structure must be in a cascaded form when using the conventional single-input single-output (SISO) PI controllers. The cascaded structure obligates the inner control loop to be much faster than the outer loop in order to achieve an acceptable dynamic behaviour, where this is physically not required. A typical cascaded structure for the speed and current control of an *ac* machine is depicted in Fig. 1.2(a). Furthermore, it necessitates the linearization of the system model, and it needs to be combined with anti-windup schemes.



**Figure 1.2:** (a) Typical cascaded control structure of an electrical machine (b) cascade-free alternative using model-based predictive control.

## 1.2.2 The advantageous characteristics of MPC

Taking a high-level look to the basic characteristics of MPC with respect to the needs and requirements while controlling power electronic systems, and by considering the progress made in academia as well as in industry in this regard, the main arguments to use MPC in the field of power electronics and electrical drives are the following:

1. The straight forward consideration of any foreknowledge or nonlinearities, and their direct inclusion within the control design. Here one needs to distinguish where do these relatively complex models are to be included:

- *In the cost function:* as in the case of the nonlinear models that are used in prediction because they better represent the plant behaviour. Whether a direct<sup>1</sup> or an indirect MPC scheme is used, the use of more complex nonlinear models yield an increase in the computational demand, however, with different extents. In direct MPC, the optimization problem is solved by an enumeration approach, in which the number of iterations of the solver is directly correlated to the number of possible control inputs (i.e. voltage vector (VV)s) for a given prediction horizon length. Hereof, once parallel computing is used, the impact of using nonlinear models is mild. For the indirect deadbeat model predictive control (DB-MPC), using a complex nonlinear

<sup>1</sup>The term *direct* control is used in the power electronics literature to describe control methods in which the semi-conducting devices are being directly controlled with the absence of a modulator. In contrary, the term *indirect* control is used for methods in which the control input is found by means of continuous control and then is realized by a modulation scheme.

prediction model does not significantly affect the computational complexity as the deadbeat MPC solution is computed analytically. In contrary, incorporating complex nonlinear models within the numerical constrained indirect MPC has a direct impact on increasing the computational complexity, and thus, it directly affects the length of the achievable prediction horizon for a given sampling time and computational power.

- ❑ *Outside the cost function*: which may come in the form of compensation lookup table (LUT)s such as the inverter phase-voltage drop or any other nonlinearity that can be compensated for in a feedforward manner, or dealing with delays (i.e. due to the digital implementation) within the control design. This can be included in all classes of MPC schemes with the same degree of simplicity.
2. The use of a multi-objective cost function, which facilitates reaching multi control objectives simultaneously. In the realm of power electronics and electrical drives, several innovative and advantageous applications of this property have come to light, such as:
- ❑ The simultaneous control of the current and speed of an electrical machine. This, as previously mentioned, makes the cascaded control structure not a must as illustrated in Fig. 1.2(b), in contrary to the traditional PI-FOC.
  - ❑ An improved acoustic behaviour of the drive system by directly penalizing acoustic properties in the cost function [24].
  - ❑ Obtaining less switching losses of the inverter at same total harmonic distortion (THD) content of the machine's stator current, in comparison with the state-of-the-art PI-FOC [25].
  - ❑ Implementing sensorless control by adding a high frequency voltage reference in the cost functional [26].
  - ❑ Achieving an active balancing of the neutral point voltage of a 3-levels neutral point clamped (NPC) inverter [27].

Nevertheless, the use of a multi-objective cost function yields one of the still unsolved issues of MPC, which is the choice of weighting factors [28].

3. The direct fulfilment of the states and inputs constraints, which allows to operate the system at its physical limit while protecting the different hardware components, such as the motor windings, the battery, and the semi-conducting devices. Depending on the nature of the MPC variant, the complexity of fulfilling the constraints is different, such as
- ❑ In direct MPC: the input constraints are fulfilled automatically by limiting the optimal solution to one out of the discrete VVs of the inverter. The common way to ensure that states' constraints are satisfied is to directly penalize violations of the constraints in the cost function by a barrier function approach.
  - ❑ In indirect MPC: for the DB-MPC variant, typically the inputs and states constraints are fulfilled using the projection method, which is known to be a sub-optimal method especially in field-weakening (FW) operation. For the continuous control set model predictive control (CCS-MPC) variant, the inputs and states constraints are typically

formulated as inequality constraints while formulating the OCP, which is typically solved by a standard approach such as interior-point (IP) or sequential quadratic programming (SQP) methods [29].

4. High-dynamic performance in transients without or with minimal over- or under-shootings, which makes MPC advantageous in applications where high dynamic performance is needed. However, to achieve this, the availability of an accurate mathematical model that represents the plant is essential.
5. Better reaction against load variation.
6. Intuitiveness of tuning, which enhances its acceptance.
7. Unified framework: which makes MPC with its direct and indirect variants an attractive control strategy to a wide range of applications with different needs and requirements. This makes it easier to standardize.

Combining some of these advantageous characteristics can lead to a more optimized design. For instance, operating the drive system at its maximum physical capacity can enable downsizing of the drive system or some of its components. Another example is the even distribution of switching events across the different switches in a power converter, resulting in a more evenly distributed thermal stress on each individual switch. These are direct measurable enhancements on important key performance indices (KPIs), such as: cost reduction, reliability and extended life span.

### 1.3 Who did MPC first in the power electronics community?

In this section, a historical overview of the development of predictive controllers specifically in the power electronics community is given in a chronological order. Afterwards, a conceptual classification of these predictive controllers is made.

In the year 1983, the first successful implementation of a controller that uses a mathematical prediction model for power converters was reported in [30,31]. The aim of the presented method was to minimize the switching frequency, or to take the steepest current gradient in transients, if needed. This was done by maintaining the current switching state of the converter until a predicted *hysteresis* is crossed, in which so called hysteresis-based predictive control. The optimization criteria was the maximum on-time until the error boundary is being touched, and this inherently ensures the lowest possible switching frequency.

Few months later, parallel work concerning predictive control in the field of power electronics got published [32–34], where a microcomputer-based predictive current control strategy with the corresponding theoretical and implementation aspects was presented. The potential of predictive control in this field, i.e. fast transient behaviour and the possible inclusion of the system nonlinearities was pointed out. The proposals in [30–34] eliminate the need of space vector modulators as well as dynamic decoupling systems, as the switching devices are being *directly* controlled.

In the following five years, another class of predictive controllers in the field of electrical drives has appeared, which is the trajectory-based predictive controllers. Despite the absence

of the word *predictive* in this class of controllers, in fact, they do rely on *pre-calculating* trajectories of the controlled variables. In this class, the VVs of the inverter are classified in a way such that, whether they will increase or decrease the value of the controlled variables. The power switches are directly switched based on the digital signal coming out of hysteresis controllers (comparators, e.g. Schmitt Trigger), which compare the actual values of the controlled variables to their reference values. The very first successful validations of the idea of trajectory-based predictive controllers are the direct self control (DSC) [35], and the DTC [22, 23].

After that, the number of contributions regarding predictive control within the power electronics community in the following decade was very limited. Mainly, because of the high computational burden at short sampling periods in order to gain performance at relatively high switching frequencies.

Due to the significant advancements of the computational resources and microprocessors, a new re-call of predictive control within the power electronics community was made in the beginning of the new millennium [17–19]. This class of predictive controllers is known in the literature as generalized predictive control (GPC) or model-based predictive control (MBPC), and is inherently different from the previous hysteresis-based and trajectory-based predictive controllers, as the system response is predicted not only over the next switching cycle, but over a finite prediction horizon. Based on this prediction, the control input that will force the system output toward its reference is chosen, normally by optimizing a quadratic cost function. Ever since, this class of MPC schemes has considerably piqued a gradual interest within the community [15, 16, 21, 36–42].

For a historical family tree of predictive control proposals within the power electronics and electrical drives community, the reader is referred to [17], and for an overview about the most recent sub-categories of MBPC, the reader is referred to [43].

## 1.4 Challenges and active research areas

Generally, MPC is facing the following main challenges in its way to be the future state-of-the-art in the field of electrical drives and power electronic systems:

- *Computational expense*: was the main challenge facing MPC in the field of power electronic systems since its very early proposals [31–34] until the re-call made at the beginning of the new millennium, once significant developments of the computational power have been made [17, 18]. Developing computationally efficient algorithms was and still is an active research direction in the community, with the following directions:
  - One-step finite control set model predictive control (FCS-MPC): faster methods were proposed to find, and if necessary, to project the unconstrained solution of the OCP back to the voltage constraints, then to evaluate only the adjacent VVs instead of all possible ones [44–46]. Currently, the required computations can already be executed with the traditional FCS-MPC approach that evaluates the cost function for all possible voltage vectors (VVs) of the power converter for short prediction horizons. Even though the execution time reduction using this computationally efficient approach might seem minor (i.e. 1–2  $\mu\text{s}$  for a 2-levels voltage source inverter

(VSI) fed machines) at the first glance, the inherent importance of this approach can be found in two applications:

- For power converters with high number of VVs as in the case for multi-level inverters, this approach makes the implementation of FCS-MPC possible, where evaluating all possible VVs in the traditional method is exhaustive.
  - For power converters with low number of VVs as in the case of 2-levels VSI, executing the algorithm in  $3 \mu\text{s}$  instead of  $4.5 \mu\text{s}$  allows going for 50% higher switching frequency, and hence, superior steady-state performance.
  - Long-horizon FCS-MPC: in which the exponential increase of the computational expense makes the optimization problem intractable [47]. Branch-and-bound methods (e.g. sphere decoding) have proved their effectiveness with successful real-time implementations [25, 48–52].
  - Constrained CCS-MPC: for which different numerical solvers have been proposed and achieved successful real-time realization with prediction horizons from one up to four steps [53–57].
- *Model dependency*: the accuracy of predictions is directly dependent on the used mathematical model of the plant, thus, model uncertainties as well as mismatches will negatively affect the control performance in both steady-state as well as in transients. The negative effect can be in a form of offsets, higher ripples, and over or under-shootings in transients. In severe cases, this may lead to control deterioration and instability. Different approaches have been proposed in the literature to counteract these effects, such as online parameter estimation, embedding discrete time integrators within the MPC control design, using offline-obtained nonlinear models either based on physical laws or obtained purely from data in a black box identification fashion.
  - *Weighting factors*: finding the optimal choice of weighting factors in multi-objective MPC is a challenging task due to the different dynamics of the terms present in the cost function, their varying and different control priority, scales, and units. For MPC in power electronic applications, and in particular, for FCS-MPC, various approaches have been proposed to ease the choice of the weighting factors, such as: following empirical design guidelines [58], using a ranking-based strategy [59], relying on algebraic design guidelines [60], enumerating the cost function sequentially in sequential MPC as proposed in [61] and in Even-handed sequential MPC [62], or by the use of NNs [63, 64]. In [65], a recent review of the different methods proposed in the power electronics literature to overcome the challenge of weighting factors design was made. Insightful counter-intuitive observations about the choices of the weighting factors have been reported in [28] (e.g. increasing the weighting factor of one term does not necessarily mean a minimization of its cost).
  - *Stability*: it has been proved when assuming linear time-invariant (LTI) systems [66, 67]. However, it is not straight forward to prove it considering nonlinear models, and any proof needs to refer to the cost function.

## 1.5 Motivation and contributions

The contributions made throughout this dissertation are mainly motivated by the aim to contribute into alleviating some of the previously reviewed MPC challenges. These are listed in the following:

- Simulative and experimental quantification of the closed-loop control performance loss resulting from the model mismatch between the real plant and the model used in MPC → partially published in [46] and [56].
- Detailed and comprehensive nonlinear modelling of the permanent magnet synchronous motor (PMSM) drive system is presented, which is essential for optimal performance of model-based control. This includes the parameters identification of the nonlinear electrical machine model based on the physical laws, the VSI nonlinearity compensation, the inevitable angle delay compensation, and the voltage and current measurement synchronization → partially published in [68] and [57].
- The proposal of using data-driven nonlinear modelling approaches to model the 2-levels VSI nonlinearity and the current dynamics of the PMSMs. Among other methods, NNs proved to be the most convenient choice as they do not require foreknowledge of the plant as priory, provided excellent prediction accuracy, and they do scale manageably with the data set size → partially published in [68] and [69].
- Comprehensive literature survey of integral action approaches with MPC schemes for the control of electrical drives → partially published in [57].
- Proposing a computationally-efficient one-step FCS-MPC algorithm that yields similar performance of the conventional method with one third less computational demand applied on a PMSM drive fed by a 2-levels VSI → partially published in [46].
- Proposing the use of NNs to mimic the long-horizon beneficial FCS-MPC policy in a real-time-capable manner for PMSM drives → partially published in [47].
- Developing an efficient numerical solver based on the primal-dual IP method for the use in real-time implementation of the constrained linear and nonlinear CCS-MPC schemes. It has been tested in real-time on a surface-mounted permanent magnet synchronous motor (SMPMSM) and an interior permanent magnet synchronous motor (IPMSM), and it can be adjusted to other constrained optimization problems for electrical drive applications → partially published in [56] and [57].
- Proposing an offset-free linear CCS-MPC scheme for PMSM drives based on the  $\delta u$  formulation, that uses the input increments as decision variables instead of the relative inputs, which has proved to ensure robust current control performance → partially published in [56].
- Experimental quantification of the performance of nonlinear CCS-MPC for an IPMSM drive with optimal dynamical response and constraints fulfilment in real-time → partially published in [57].

- 
- Proposing an explicit-like learning-based framework to incorporate generic data-driven models with MPC for electrical drive applications. The proposed algorithm is tested on an IPMSM drive with a neural data-driven model as an example and is benchmarked experimentally with the physics-based MPC → partially published in [69].





## CHAPTER 2

---

### Conventional modelling of an electrical drive

---

*“All models are wrong, but some are useful.”*

George Box

In this chapter, an overview about the well-known physics-based first-principles linear and non-linear models<sup>1</sup> of a PMSM drive, as a base example, particularly for the use in model-based control is given. The modelling section is preceded by a brief introduction that provides the primary components of an electrical drive, the fundamental coordinate transformation, and essential pre-knowledge needed to be incorporated in the control design of electrical drive systems. This includes: the VSI nonlinearity, the angle delay due to the digital implementation of control schemes, as well as the synchronization of the current and voltage measurements. To this end, the physics-based model is derived and the parameters of a real IPMSM are obtained at the test bench. This chapter is intended to give the reader a comprehensive illustration of the conventional modelling procedure to do model-base control, to be benchmarked with the proposed data-driven modelling techniques in the next chapter in terms of model complexity and accuracy, the needed computational time, and the needed effort to obtain.

#### 2.1 Components of an electrical drive system

An electrical drive system is a system that uses power electronics components and control techniques to convert electrical energy into mechanical energy. Electrical drive systems typically contain the following components:

**1- Power supply:** that provides the input electrical energy in the form required by the used electrical machine, whether *ac* or *dc*.

---

<sup>1</sup>The term first-principles model is used throughout the thesis to call the electrical machine model derived from the physical laws governing the functionality of the synchronous motor, whether it is the mostly used LTI model or the nonlinear model with the inductance/flux current-dependent maps.

**2- Inverter/converter:** that outputs the energy in the needed form to drive the electrical machine with respect to the power supply. Commonly, VSIs are used, in which the number of levels reflects the number of possible output VVs that the inverter can provide. Based on the topology and the number of semi-conducting modules used, different number of output VVs such as 8, 27, 64 or many others can be achieved.

**3- Electric motor:** whether it is a brushed or a brushless *dc* motor, a synchronous or an asynchronous *ac* motor, the electric motor is the main component of an electrical drive system that converts the electrical energy into mechanical energy to drive the load.

**4- Sensors:** are needed to convey the needed real-time information to control the states of the drive system at which the energy is being stored, which are the current, the position and the speed. For closed-loop control, measurement of phase-voltages, phase-currents, rotor position and angular speed are needed. To minimize costs, enough information could be derived by having only one input voltage sensor from which the phase-voltages can be derived by knowing the inverter switching state, two current sensors from which the third phase current can be computed assuming balanced three-phase system, and a position sensor from which the speed can be derived. The latter can be dispensable when a sensorless control algorithm is employed.

**5- Controller:** to process the measurements, to compute the control law, and to generate the control signals. These include microcontrollers (MCUs) and FPGAs.

**6- Mechanical load:** that is to be mechanically driven by the electric motor. For characterizing the motor under control, the mechanical load is typically emulated by coupling with another electrical machine. In the following sections, the basic coordinate transformation, the modelling of the 2-levels VSI and the three-phase PMSM, which are used in the base example throughout this thesis, are presented.

## 2.2 Coordinate transformation

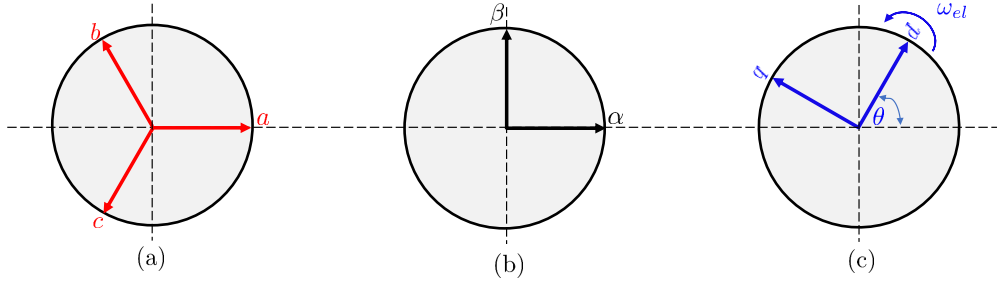
Typically, the three-phase quantities of an electrical drive system are mathematically transformed into the *stationary*  $\alpha - \beta$  plane, in which a classical control task is to control a rotating vector that represents the steady-state operation. To further simplify the control of *ac* machines, the quantities are typically further transformed into the *rotating*  $d - q$  frame, in which the  $d - axis$  is aligned with the rotor's axis of symmetry in which the magnetic field rotates, and the  $q - axis$  is perpendicular to it. This allows the instantaneous and separate control of the machine's flux and torque.

In order to convert any arbitrary three phase quantity from the stator-fixed *abc* reference frame to the stationary  $\alpha - \beta$  frame, amplitude-invariant Clarke transformation is used as in Eq. 2.1(a), where  $\zeta_{abc} = (\zeta_a \ \zeta_b \ \zeta_c)^\top$  is an arbitrary variable vector in the *abc* frame. To further transform the quantities to the rotating  $d - q$  frame, Park transformation is employed as in Eq. 2.1(b), in

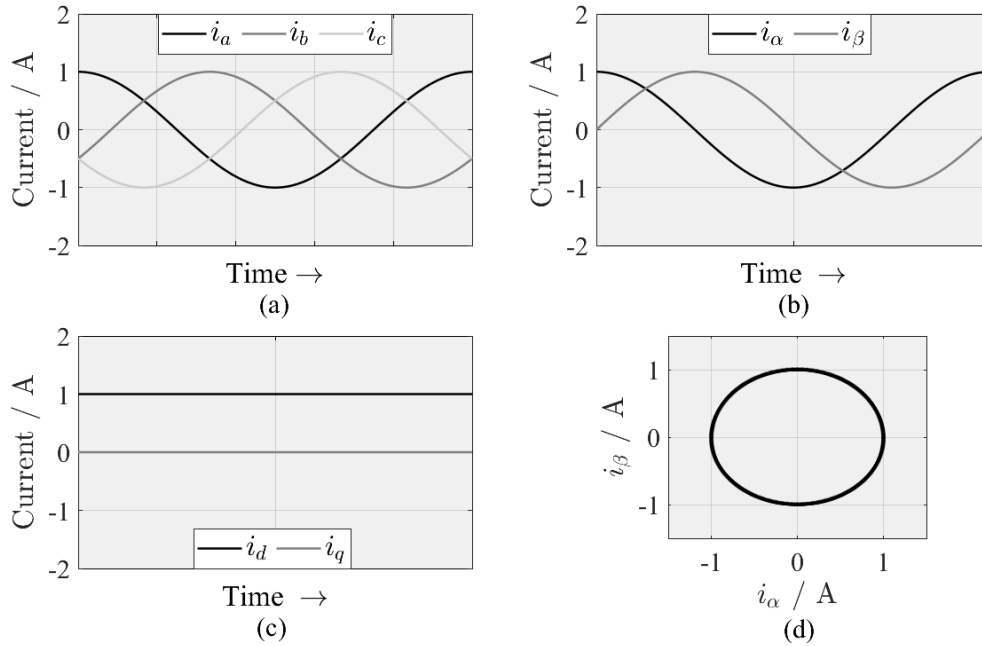
which the reference frame itself is rotating with  $\theta$  as the electrical angle of rotation [70].

$$\begin{pmatrix} \zeta_\alpha \\ \zeta_\beta \end{pmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{pmatrix} \zeta_a \\ \zeta_b \\ \zeta_c \end{pmatrix}, \quad (2.1a) \quad \begin{pmatrix} \zeta_d \\ \zeta_q \end{pmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{pmatrix} \zeta_\alpha \\ \zeta_\beta \end{pmatrix}. \quad (2.1b)$$

The three coordinate systems are depicted in Fig. 2.1, with an example of a steady-state current vector that has an amplitude of 1 A being depicted in the reference frames in Fig. 2.2.



**Figure 2.1:** Coordinate systems: (a) the  $abc$  reference frame, (b) the stationary  $\alpha - \beta$  frame, (c) the rotating  $d - q$  frame.



**Figure 2.2:** An example of a current vector in a typical steady-state operation: (a) in the  $abc$  frame, (b) the transformed current in the stationary  $\alpha - \beta$  frame, (c) the equivalent current in the rotating  $d - q$  frame, (d) one cycle of the rotating vector in the  $\alpha - \beta$  plane.

In a standard steady-state operation of an electrical drive, the machine has to be fed with an alternating three-phase voltage that rotates at a constant frequency. This is equivalent to a rotating VV in the  $\alpha - \beta$  frame, and is equivalent to a constant vector in the rotating  $d - q$  frame. Such a VV can be generated by the interplay of the *discrete* VVs of the power converter, while its maximum possible amplitude is defined with respect to the input side  $dc$  voltage.

### 2.3 Generic model of a 2-levels inverter

The most commonly used *dc* to *ac* inverter especially in low-to-medium power applications is the 2-levels inverter, which consists of six switches with parallel free wheeling diodes [71]. The input side consists of the *dc*-link, which has a specific *dc* input voltage that will be modulated to form the output three-phase *ac* voltage to drive the electric machine. The gates of the upper and lower switches of each leg of the inverter must have complementary driving signals. Their transition states are shifted with appropriate dead times to avoid short circuit on the input side [71]. The instantaneous phase-to-phase voltage  $\mathbf{u}_{ptp}^f$  on the output of the inverter is expressed in terms of *dc*-link voltage  $u_{dc}(t)$  and the switching states vector:

$$\mathbf{s}_{abc} = \begin{pmatrix} s_a \\ s_b \\ s_c \end{pmatrix} \in \sim := \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right\} \text{ as}$$

$$\mathbf{u}_{ptp}^f(t) = \begin{pmatrix} u_{ab}^f(t) \\ u_{bc}^f(t) \\ u_{ca}^f(t) \end{pmatrix} = u_{dc}(t) \mathbf{T}_{ptp} \mathbf{s}_{abc}(t), \quad (2.2)$$

and the three-phase stator voltages are defined as

$$\mathbf{u}_{abc}^f(t) = \begin{pmatrix} u_a^f(t) \\ u_b^f(t) \\ u_c^f(t) \end{pmatrix} = \frac{u_{dc}(t)}{3} \mathbf{T}_{abc} \mathbf{s}_{abc}(t), \quad (2.3)$$

where  $\mathbf{T}_{ptp}$  and  $\mathbf{T}_{abc}$  are coefficient matrices, such as

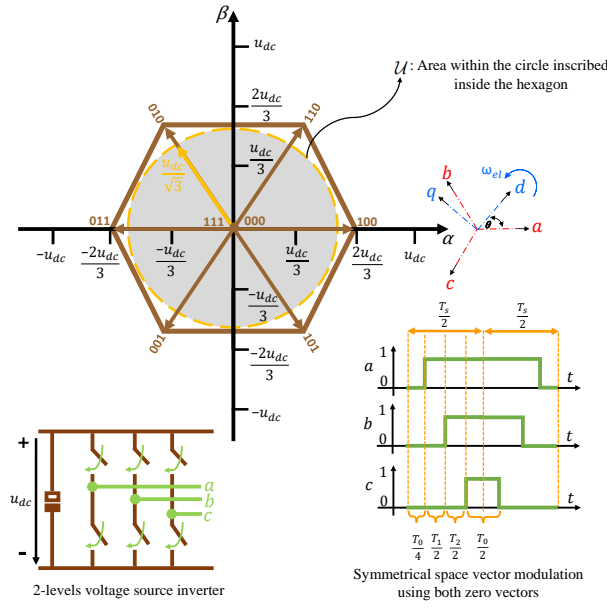
$$\mathbf{T}_{ptp} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}, \quad (2.4a) \quad \mathbf{T}_{abc} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \quad (2.4b)$$

and hence,  $\mathbf{u}_{(\cdot)}^f(t) \in \{0, \frac{u_{dc}(t)}{3}, \frac{2u_{dc}(t)}{3}, \frac{-u_{dc}(t)}{3}, \frac{-2u_{dc}(t)}{3}\}$ , for  $(\cdot) \in \{a, b, c\}$ .

The finite output VVs of a 2-levels VSI with an illustration of the symmetrical space vector modulation (SVM) using both zero vectors that is typically used in indirect control of electrical drives are depicted in Fig. 2.3. In case that the continuous-time VV is not identical to one of the base vectors, and as the base vectors have fixed magnitude with respect to the *dc*-link voltage level, any continuous-time VV that lies within the hexagon needs to be masked as an average by using the adjacent vectors in the sector where it lies in combination with the zero vectors with corresponding timings in each sampling period, in which so-called SVM. In case the hexagonal constraints are considered, a higher utilization of the *dc*-link voltage is achieved, however, at the expense of distorting the phase currents while operating in the over-modulation region. The control input vector elements are determined from a *finite-set* of possible output VVs

$$\mathbf{u}_k \in \{\mathbf{u}_1, \dots, \mathbf{u}_{N_{VV_s}}\}, \quad (2.5)$$

where  $N_{VV_s}$  is the number of VVs of the power converter, and is  $= 8$  for 2-levels inverters. To this end, the output VVs of the 2-levels inverter are summarized in Table 2.1 in the *abc* reference frame, and in Table 2.2 in the  $\alpha - \beta$  reference frame.



**Figure 2.3:** 2-levels VSI and its output VVs in the stationary  $\alpha - \beta$  plane with an illustration of the symmetrical SVM.

**Table 2.1:** Output VVs of a 2-level VSI in the three-phase  $abc$  reference frame.

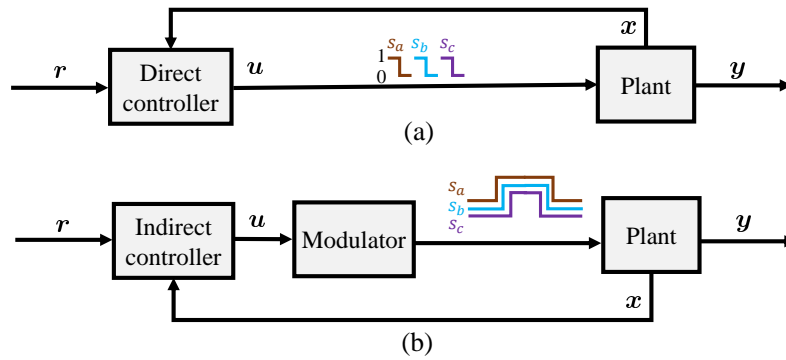
$s_{abc}$	$u_a^f$	$u_b^f$	$u_c^f$
000	0	0	0
100	$\frac{2}{3}u_{dc}$	$-\frac{1}{3}u_{dc}$	$-\frac{1}{3}u_{dc}$
110	$\frac{1}{3}u_{dc}$	$\frac{1}{3}u_{dc}$	$-\frac{2}{3}u_{dc}$
010	$-\frac{1}{3}u_{dc}$	$\frac{2}{3}u_{dc}$	$-\frac{1}{3}u_{dc}$
011	$-\frac{1}{3}u_{dc}$	$\frac{1}{3}u_{dc}$	$\frac{2}{3}u_{dc}$
001	$-\frac{1}{3}u_{dc}$	$-\frac{1}{3}u_{dc}$	$\frac{2}{3}u_{dc}$
101	$\frac{1}{3}u_{dc}$	$-\frac{2}{3}u_{dc}$	$\frac{1}{3}u_{dc}$
111	0	0	0

**Table 2.2:** Output VVs of a 2-level inverter in the stationary  $\alpha - \beta$  reference frame.

$s_{abc}$	$u_\alpha^f$	$u_\beta^f$
000	0	0
100	$\frac{2}{3}u_{dc}$	0
110	$\frac{1}{3}u_{dc}$	$\frac{\sqrt{3}}{3}u_{dc}$
010	$-\frac{1}{3}u_{dc}$	$\frac{\sqrt{3}}{3}u_{dc}$
011	$-\frac{1}{3}u_{dc}$	0
001	$-\frac{1}{3}u_{dc}$	$-\frac{\sqrt{3}}{3}u_{dc}$
101	$\frac{1}{3}u_{dc}$	$-\frac{\sqrt{3}}{3}u_{dc}$
111	0	0

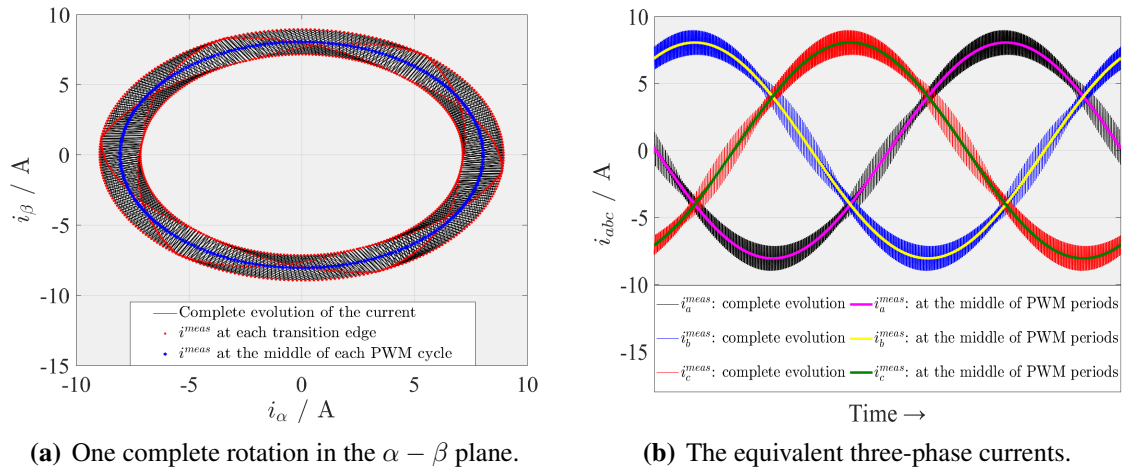
In general, the control schemes of electrical drives are classified into *direct* and *indirect* control. In direct methods, the switching states of each phase of the power converter are switched once and are fixed over the sampling period, where indirect methods incorporate the use of a modulator to the real-valued control inputs, and hence, allowing each phase to flip the complementary gating signals at dedicated timing defined by the corresponding duty cycles in order to realize, as an average, the intended continuous-time VV. The general structure of direct and indirect controllers is depicted in Fig. 2.4.

For simulation purposes, a crucial aspect that must be taken into consideration is the modelling of the SVM scheme in the simulation environment. The sampling time needs to be small enough to detect the complete current evolution at each transition edge within each pulse width modulation (PWM)/sampling period. This is illustrated for an arbitrary current tracking exam-



**Figure 2.4:** Direct and indirect control structure for electrical drives: (a) direct control, (b) indirect control, with  $r$  as the reference vector.

ple in the  $\alpha - \beta$  plane in Fig. 2.5(a), and the corresponding three-phase currents in the  $abc$  frame are shown in Fig. 2.5(b) for an ideal inverter. Considering this ripple is essential, especially to make a fair performance comparison between direct and indirect control methods in terms of ripple amplitudes and THD content.



**Figure 2.5:** Example illustrating the difference between measuring the current only once at the middle of each symmetrical pulsation SVM period and at each switching transition within the sampling period assuming an ideal inverter.

## 2.4 Pre-modelling

In this section, essential pre-modelling knowledge for the design of model-based control is discussed. Firstly, the mismatch effect between the commanded and the real phase voltages is addressed, and the conventional compensation method is implemented. Secondly, the inevitable angle delay caused by the needed time to execute the mathematical operations on digital platforms is presented with the standard method that is typically used to counteract this effect. Finally, the section is concluded by the needed synchronization between the different time instants at which the voltages are being applied and the currents are being measured.

## 2.4.1 Inverter nonlinearity compensation: conventional method

### 2.4.1.1 Problem statement

Due to various sources of output voltage deterioration of VSIs, such as: safety dead times of the switches on the same leg to prevent short-circuiting of the voltage source, varying turn on and turn off times of each semiconductor as a function of the phase current and temperature, voltage drop during the conducting state of the semi conductive device, zero-current clamping phenomena, and others, the output voltage of the VSI is not identical to the commanded voltage. For model-based control, accurate knowledge of the phase voltages of the electrical machine is essential. Typically, only current sensors and a *dc*-link voltage sensor are equipped to minimize the overall cost of the drive system. Therefore, an appropriate compensation method is needed to match the real output voltage with the commanded one.

### 2.4.1.2 Experimental investigation of the VSI nonlinear effects

The state-of-the-art procedure to observe as well as to compensate for the 3-phase VSI non-linearity is to lock the rotor shaft, and to inject phase current gradually from the maximum negative to positive polarity in an ascending profile for one phase, while the second phase takes the same amplitude of the first phase current with an opposite polarity, and the third phase is kept on zero current, in which so called the two-phase configuration as described in [72]. In this setup, with the shaft being locked, the machine model simplifies to

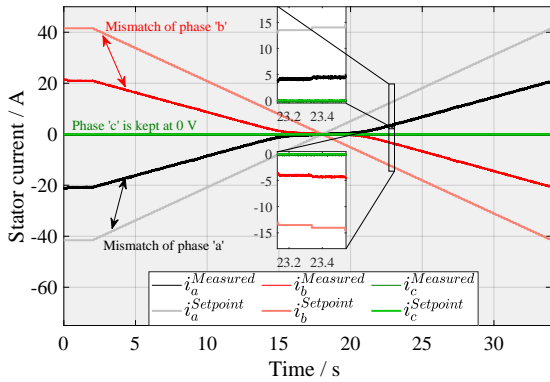
$$u_{(\cdot)} = R_s i_{(\cdot)} \quad (2.6)$$

$\forall(\cdot) \in \{a, b, c\}$ . The phase resistance  $R_s$  is measured via an ohmmeter before performing the test, and it has a value of  $38.5 \text{ m}\Omega$  for the motor in the test bench shown in Fig. B.1 in Appendix B. The phase current is measured, and hence the phase voltage can be calculated accurately. By comparing the calculated phase voltages with the commanded ones, a mismatch is observed due to the effects mentioned at the beginning of the subsection. This standstill test was repeated at the three possible combinations by keeping one phase at  $0 \text{ V}$  while exciting the other two phases, and the results are shown in Fig. 2.6(a)-(c). To illustrate the effect in the complete  $\alpha - \beta$  plane, a fine grid of commanded VVs are given to the inverter, and their corresponding reference currents are calculated and compared with the measured ones as illustrated in Fig. 2.6(d).

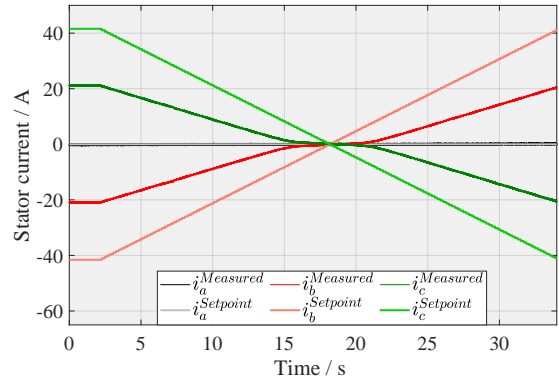
### 2.4.1.3 Feedforward inverter nonlinearity compensation

The state-of-the-art procedure to compensate for the non-ideal behaviour of the VSI in a feedforward approach is done by measuring the difference between the commanded voltage and the measured<sup>2</sup> one as described in [72]. The obtained current-dependent phase voltage curves of the three phases are shown in Fig. 2.7.

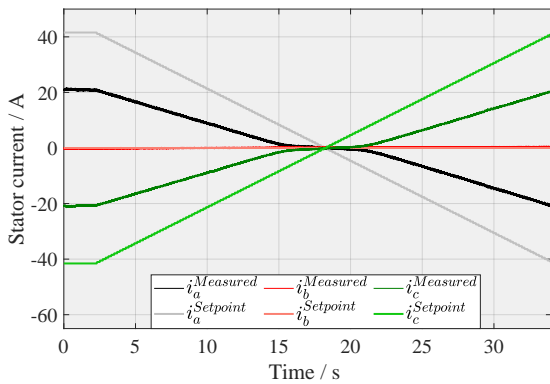
<sup>2</sup>In this test procedure, the voltage is not measured by a phase voltage sensor but computed from the measured phase currents and the known resistance.



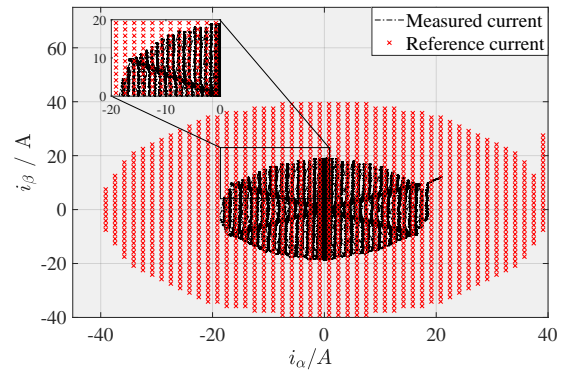
(a) Ascending setpoint steps for phase 'a', descending steps for phase 'b', and zero for phase 'c'.



(b) Ascending setpoint steps for phase 'b', descending steps for phase 'c', and zero for phase 'a'.

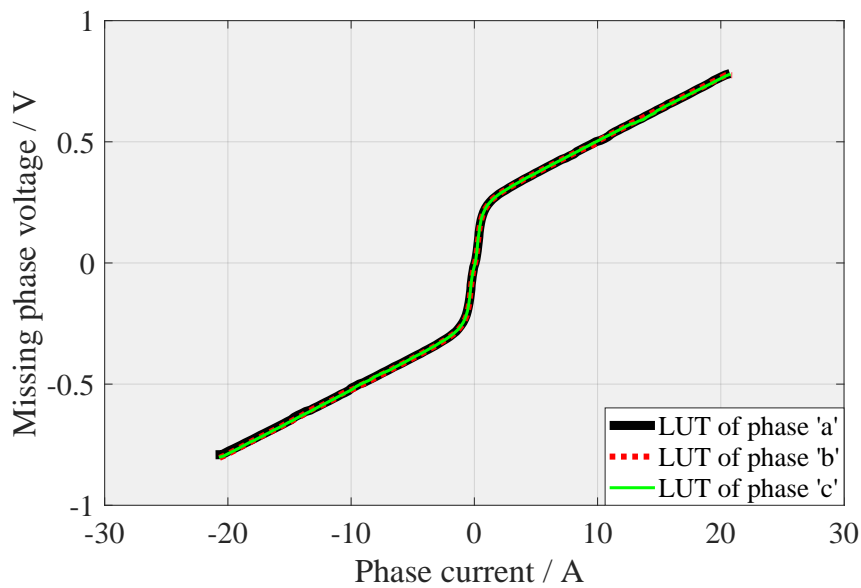


(c) Ascending setpoint steps for phase 'c', descending steps for phase 'a', and zero for phase 'b'.



(d) Grid of reference current vectors in the  $\alpha - \beta$  plane and their corresponding measured ones.

**Figure 2.6:** Standstill test results to investigate the VSI nonlinearity effect: (a)-(c) ascending reference profile to one phase, descending reference profile to the other while keeping the third phase at 0 V, (d) fine grid of current vectors in the complete  $\alpha - \beta$  plane.

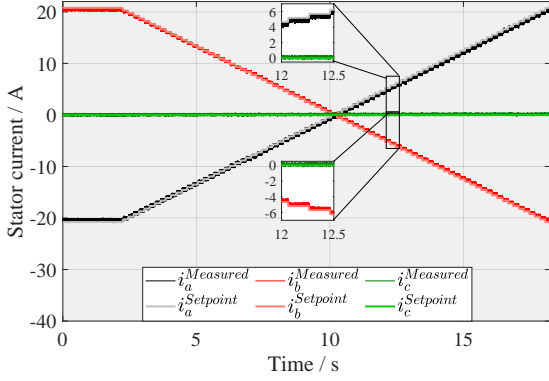


**Figure 2.7:** VSI missing voltage curves as a function of the phase currents.

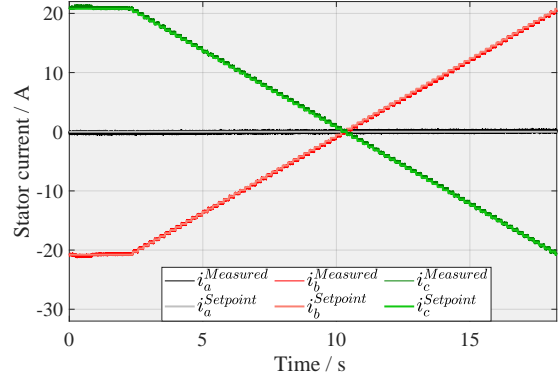


### 2.4.1.4 Validation of the obtained compensation curves

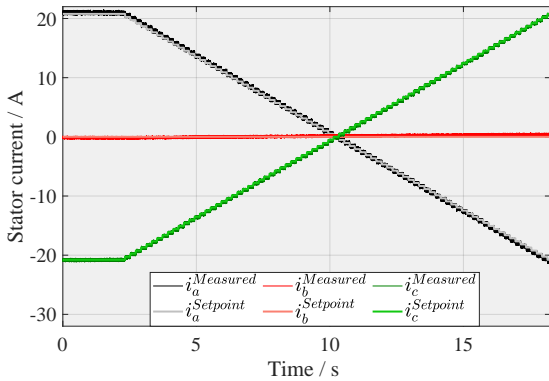
In this subsection, the same two-phase configuration test is repeated by sending the same commanded reference voltages to the inverter while adding to it the feedforward missing voltage shown in Fig. 2.7. The measured currents are compared with their references in Fig. 2.8(a)-(c), and the fine grid test results covering the complete  $\alpha - \beta$  plane with 928 reference current vectors are shown in Fig. 2.8(d).



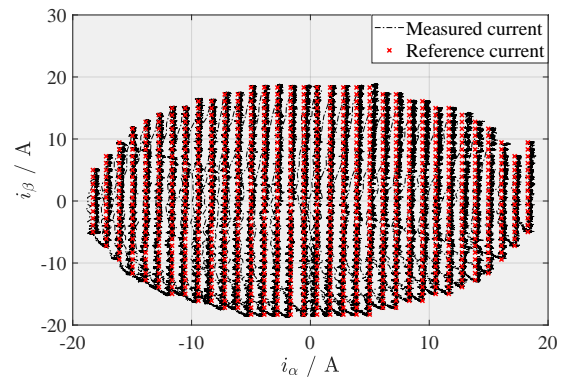
(a) Ascending setpoint steps for phase 'a', descending setpoint steps for phase 'b', and zero for phase 'c'.



(b) Ascending setpoint steps for phase 'b', descending setpoint steps for phase 'c', and zero for phase 'a'.



(c) Ascending setpoint steps for phase 'c', descending setpoint steps for phase 'a', and zero for phase 'b'.

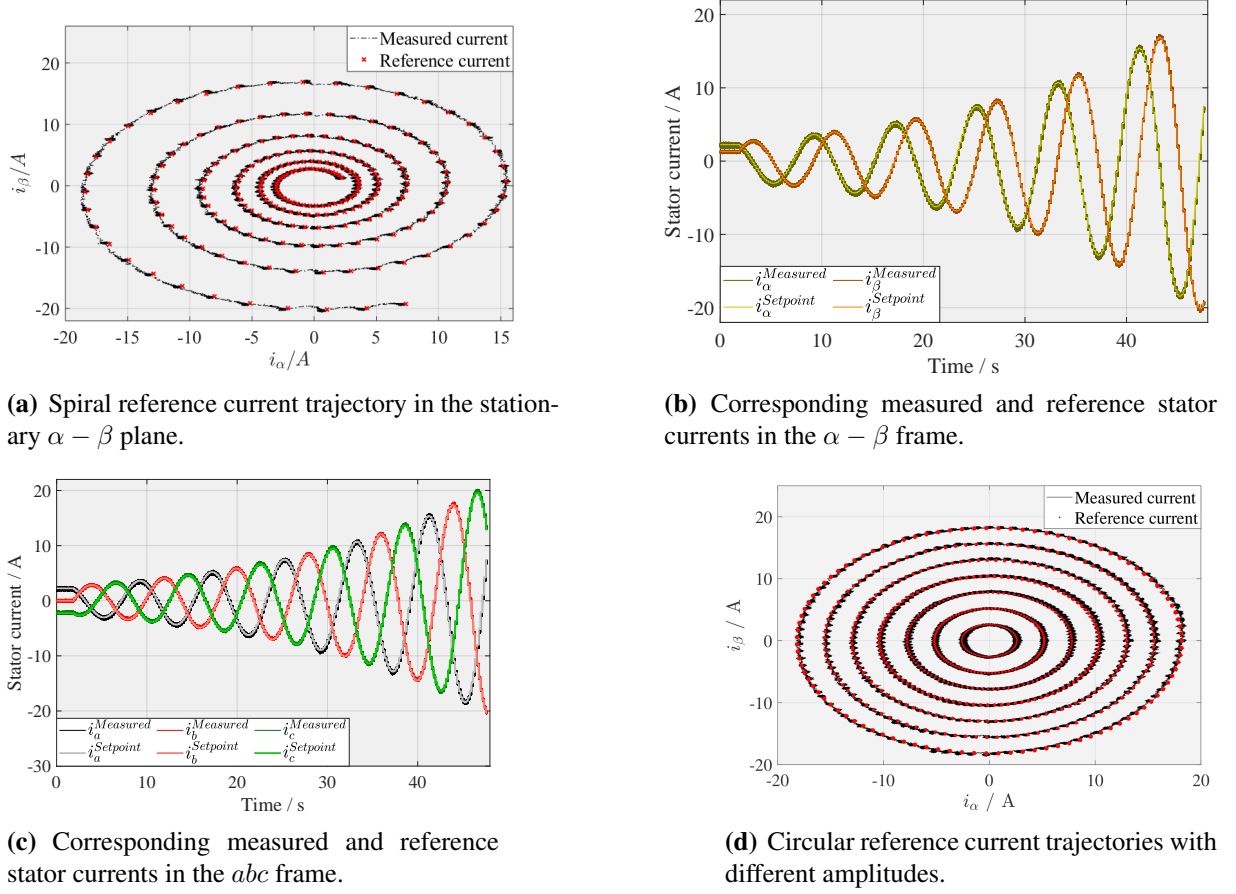


(d) Fine grid of reference current vectors in the  $\alpha - \beta$  plane and their corresponding resultant measured vectors.

**Figure 2.8:** Validation of the feedforward compensation of the VSI non-ideal behaviour at standstill: (a)-(c) two-phase configuration test results, (d) fine grid test results in the  $\alpha - \beta$  plane.

In order to validate the above mentioned compensation method in the  $\alpha - \beta$  plane in a similar scenario to the normal operation while driving an electrical machine, a spiral reference current trajectory<sup>3</sup> is sent to the inverter, and it is compared to the measured current in Fig. 2.9(a)-(c). Lastly, rotating circular reference trajectories with different amplitudes in the  $\alpha - \beta$  plane were tested and the results are shown in Fig. 2.9(d). These results prove that this simple compensation method is effective in mitigating the voltage drop effect of the VSI.

<sup>3</sup>By means of reference voltage and known resistance.



**Figure 2.9:** Spiral and circular validation tests of the feedforward state-of-the-art compensation method of a 2-level VSI: (a)-(c) spiral reference trajectory, (d) circular reference current trajectories with different amplitudes.

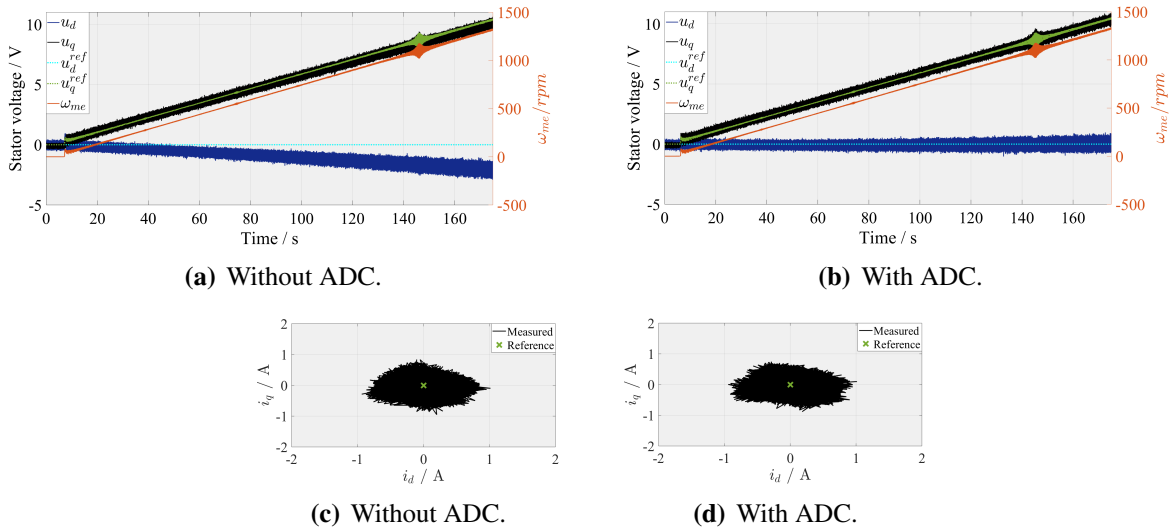
## 2.4.2 Angle delay compensation

The inevitable time delay between signal processing and execution in digital implementation of current regulation schemes yields an undesirable output voltage mismatch [73]. This mismatch is translated into a rotating speed-dependent error  $VV$  in the rotating  $d - q$  frame. To investigate this phenomena, the reference current of the main motor is set to 0 A, where the rotor speed is increased linearly via the load machine from standstill to 1400 rpm, and the needed voltage of a PI current regulator to maintain 0 current of the main motor is recorded. The control inputs are compared with the needed theoretical voltage to keep zero current, which can be computed by setting the currents and their derivatives in the model that governs the current dynamics of a synchronous motor to zero, such as

$$u_d^{ref} = R_s \dot{i}_d^0 + L_d \frac{di_d^0}{dt} - \omega_{el} L_q \dot{i}_q^0 = 0, \quad (2.7a)$$

$$u_q^{ref} = R_s \dot{i}_q^0 + L_q \frac{di_q^0}{dt} + \omega_{el} L_d \dot{i}_d^0 + \omega_{el} \psi_p = \omega_{el} \psi_p. \quad (2.7b)$$

This effect is more significant in the  $d$  axis as can be seen in Fig. 2.10(a), with the stator current corresponding to this test shown in Fig. 2.10(c). The conventional approach to implement the angle delay compensation (ADC) is to add a speed-dependent angle compensation term of  $1.5T_s\omega_{el}$  while converting the commanded VV from the control scheme to the inverter duty cycles. This approach is known in the literature as phase advancing compensation [73]. The results of the same test scenario while using this additive compensation term where the computed voltages got aligned with their references is shown in Fig. 2.10(b), and the measured stator current is shown in Fig. 2.10(d).



**Figure 2.10:** Injecting zero current while increasing the shaft speed via the load machine: (a) stator voltages and their references and the measured speed without ADC, (b) stator voltages and their references and the measured speed with ADC, (c) and (d) contains the stator current without and with ADC, respectively.

### 2.4.3 Measurement synchronization

The three-phase current measurements are being collected at the middle of each sampling period  $T_s$ , while the control inputs (i.e. the voltages) are being applied at the end of one sampling period/beginning of the upcoming one. This misalignment between the time instant at which the computed voltage is applied and the instant at which the current is measured must be compensated for, in prediction for control as well as in parameters estimation and model identification. For the first case of prediction in control, this comes in the form of initial condition correction of the states. This is done by predicting the values of the states half a cycle ahead, such as

$$\mathbf{x}_k = \tilde{\mathbf{x}}_k + \frac{T_s}{2} \dot{\mathbf{x}}, \quad (2.8)$$

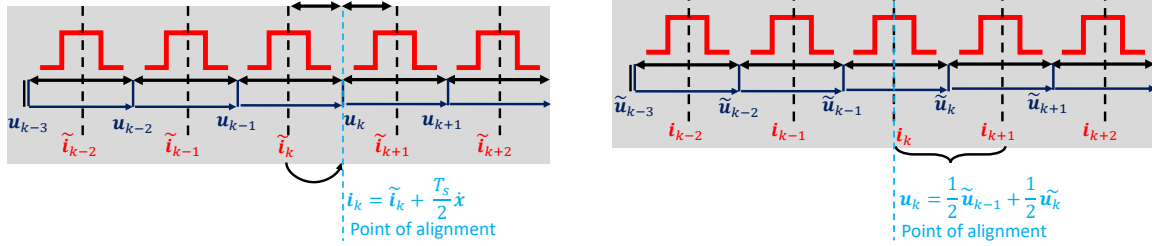
with the system dynamics vector  $\dot{\mathbf{x}}$  defined as

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \begin{pmatrix} \frac{-R_s i_d}{L_d} + \frac{L_q \omega_{el} i_q}{L_d} + \frac{u_d}{L_d} \\ \frac{-R_s i_q}{L_q} - \frac{L_d \omega_{el} i_d}{L_q} + \frac{u_q}{L_q} - \frac{\omega_{el} \psi_p}{L_q} \end{pmatrix}. \quad (2.9)$$

By this, the effect of applying  $\mathbf{u}_{k-1}$  for the upcoming half sampling period since the current was measured is compensated for. Such a compensation for real-time MPC applications was proposed in [74]. This is illustrated in Fig. 2.11(a). This method is used in the MPC schemes proposed in this dissertation. In the other case when the system dynamics are to be learned solely from collected data, a simple correction of the control inputs is sufficient to yield aligned quantities, such as

$$\mathbf{u}_k = \frac{1}{2}\tilde{\mathbf{u}}_{k-1} + \frac{1}{2}\tilde{\mathbf{u}}_k, \quad (2.10)$$

and as depicted in Fig. 2.11(b).



(a) Aligning the current measurement with the computed voltage by predicting its value half a cycle ahead, and using the predicted value as an initial state in the MPC scheme.

(b) Aligning the computed voltages with the measured current by averaging the voltages between two consecutive sampling periods. This synchronization is essential while *learning* the system dynamics.

**Figure 2.11:** Voltage and current measurement synchronization: (a) correction of the initial states for control, (b) correction of the control inputs, which is useful in online parameters estimation or while learning the system dynamics.

An alternative method was proposed and used by computing the continuous-time Koopman operator (KO) from the obtained discrete-time KO from measurements sampled at equidistant intervals (i.e. fixed sampling time), and then re-computing the discrete-time KO at the desired sampling (i.e. in this case  $\frac{T_s}{2}$ ) to be used for the initial conditions correction [75].

## 2.5 Generic model of PMSMs based on physical laws

In this section, the nonlinear mathematical model of PMSMs based on the well-known physical laws is derived, and the model parameters are obtained for the motor shown in Fig. B.1 in Appendix B. Moreover, the linearized model with constant parameters (i.e. an LTI model) that is mostly used with MPC schemes within the power electronics community is presented. These models are to be used in the following chapter to design different direct and indirect model-based predictive controllers (i.e. FCS-MPC, DB-MPC and CCS-MPC schemes).

### 2.5.1 Model derivation

The mathematical model of a three-phase PMSM in the rotating  $d - q$  reference frame is expressed as [76]

$$u_d = R_s i_d + \frac{d\psi_d^{(i_d, i_q)}}{dt} - \omega_{el} \psi_q^{(i_d, i_q)}, \quad (2.11a)$$

$$u_q = R_s i_q + \frac{d\psi_q^{(i_d, i_q)}}{dt} + \omega_{el} \psi_d^{(i_d, i_q)}, \quad (2.11b)$$

with

$$\psi_d^{(i_d, i_q)} = \psi_p^{(i_q)} + L_d^{(i_d, i_q)} i_d, \quad (2.12a)$$

$$\psi_q^{(i_d, i_q)} = L_q^{(i_d, i_q)} i_q, \quad (2.12b)$$

$$\frac{d\psi_d}{dt} = \frac{d\psi_p}{dt} + L_{dd}^{(i_d, i_q)} \frac{di_d}{dt} + L_{dq}^{(i_d, i_q)} \frac{di_q}{dt}, \quad (2.12c)$$

$$\frac{d\psi_q}{dt} = L_{qq}^{(i_d, i_q)} \frac{di_q}{dt} + L_{qd}^{(i_d, i_q)} \frac{di_d}{dt}, \quad (2.12d)$$

where  $i_d$  and  $i_q$  are the stator currents (in A),  $u_d$  and  $u_q$  are the stator voltages (in V),  $L_{dd}$  and  $L_{qq}$  are the self differential inductances,  $L_{dq}$  and  $L_{qd}$  are the mutual differential inductances,  $L_d$  and  $L_q$  are the absolute inductances (all inductances are in H),  $R_s$  is the stator resistance (in  $\Omega$ ),  $\psi_d$  and  $\psi_q$  are the flux components,  $\psi_p$  equals to  $\psi_d^{(i_d=0, i_q)}$  is the permanent-magnet flux linkage (all fluxes are in Wb),  $\omega_{el}$  is the electrical angular speed (in  $\text{rad s}^{-1}$ ), and  $\omega_{el} = n_p \omega_{me}$ , where  $n_p$  is the number of pole pairs, and  $\omega_{me}$  is the mechanical rotor speed (in  $\text{rad s}^{-1}$ ).

The absolute and differential inductances are current-dependent, and therefore, they must be identified in the whole current operating range. The absolute inductances are defined as

$$L_d^{(i_d, i_q)} = \frac{\psi_d^{(i_d, i_q)} - \psi_p^{(i_q)}}{i_d}, \quad (2.13a) \quad L_q^{(i_d, i_q)} = \frac{\psi_q^{(i_d, i_q)}}{i_q}, \quad (2.13b)$$

where the differential inductances are defined as the partial derivatives of the fluxes along the currents

$$L_{dd}^{(i_d, i_q)} = \frac{\partial \psi_d}{\partial i_d}, \quad (2.14a) \quad L_{dq}^{(i_d, i_q)} = \frac{\partial \psi_d}{\partial i_q}, \quad (2.14b)$$

$$L_{qd}^{(i_d, i_q)} = \frac{\partial \psi_q}{\partial i_d}, \quad (2.14c) \quad L_{qq}^{(i_d, i_q)} = \frac{\partial \psi_q}{\partial i_q}. \quad (2.14d)$$

To this end, substituting Eq. 2.12 in Eq. 2.11 and discretizing the system model using the explicit Euler discretization method with the sampling time  $T_s$  yields the nonlinear current prediction model that can be used within the different MPC schemes, such as:

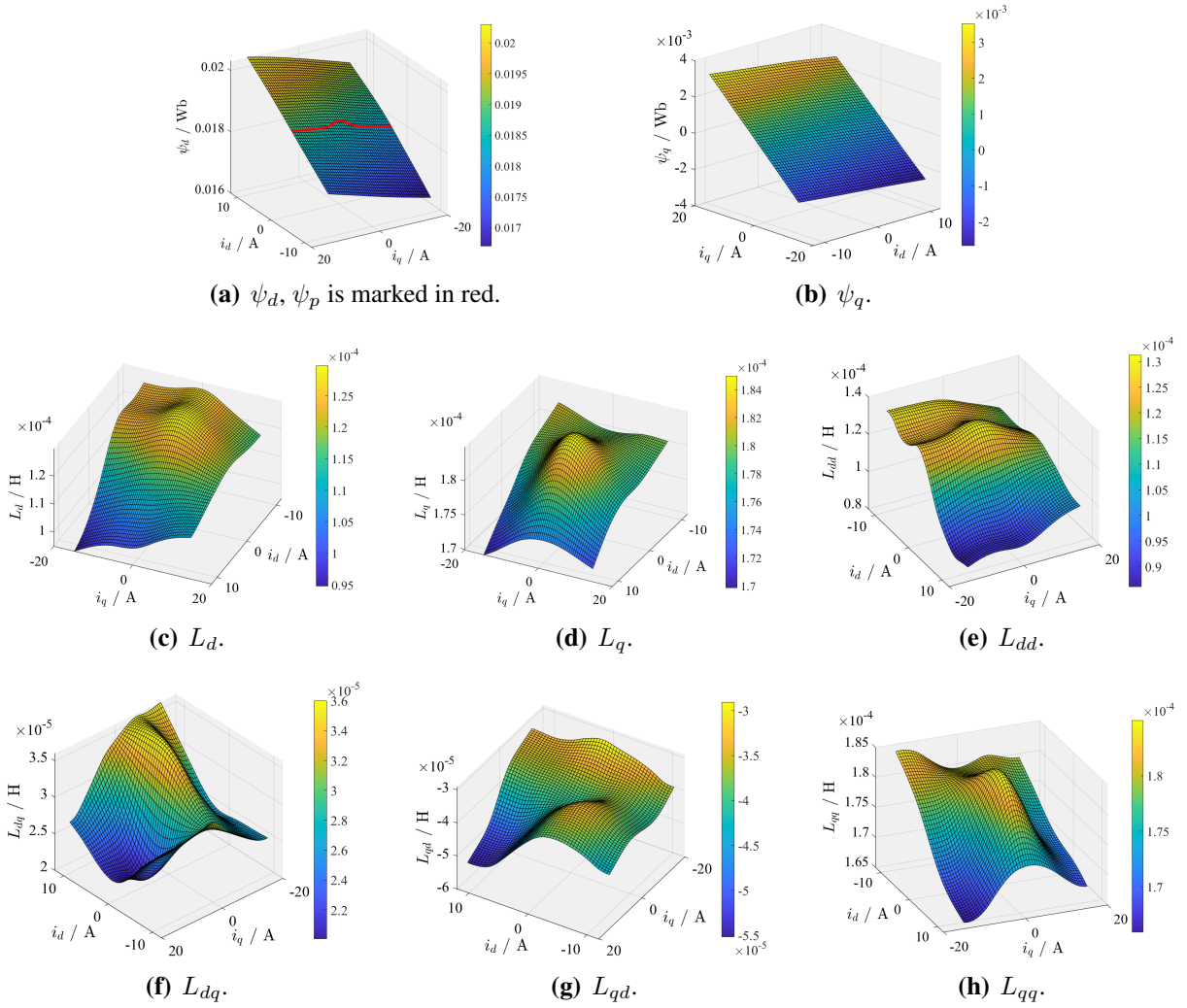
$$\begin{aligned} \begin{pmatrix} i_{d,k+1} \\ i_{q,k+1} \end{pmatrix} &= \begin{pmatrix} i_{d,k} \\ i_{q,k} \end{pmatrix} + T_s \begin{bmatrix} L_{dd,k}^{(i_d,i_q)} & L_{dq,k}^{(i_d,i_q)} \\ L_{qd,k}^{(i_d,i_q)} & L_{qq,k}^{(i_d,i_q)} \end{bmatrix}^{-1} \mathbf{j} \\ \text{with} & \\ \mathbf{j} &= \left( \begin{pmatrix} u_{d,k} \\ u_{q,k} \end{pmatrix} - \begin{bmatrix} R_s & -\omega_{el,k} L_{dq,k}^{(i_d,i_q)} \\ \omega_{el,k} L_{qd,k}^{(i_d,i_q)} & R_s \end{bmatrix} \begin{pmatrix} i_{d,k} \\ i_{q,k} \end{pmatrix} - \begin{pmatrix} 0 \\ \omega_{el,k} \psi_p^{(i_d,i_q)} \end{pmatrix} \right). \end{aligned} \quad (2.15)$$

The standard procedure to identify the current-dependent electrical parameters in the model in Eq. 2.15 is to control the motor shaft speed via a load motor<sup>4</sup>, and to inject reference steady-state currents in the complete  $i_d - i_q$  plane with an appropriate grid resolution. A simple PI controller is used to drive the stator currents to each steady-state setpoint, and the stator resistance is to be measured and assumed to be constant throughout the parameters identification procedure. Then, the fluxes can be computed at each steady-state point as

$$\psi_d = \frac{u_q - R_s i_q}{\omega_{el}}, \quad (2.16a) \quad \psi_q = -\frac{u_d - R_s i_d}{\omega_{el}}, \quad (2.16b)$$

and then the absolute inductances can be calculated as in Eq. 2.13. For the IPMSM used in the experimental validation throughout this dissertation, shown in Appendix B, the parameters are identified using the aforementioned procedure at  $\omega_{me} = 500 \text{ rpm}$  with a grid resolution of 0.5 A. The obtained parameters maps are smoothed with the splines interpolation method [77], and are depicted in Fig. 2.12. From a computational perspective, utilizing the nonlinear model in Eq. 2.15 in the different MPC schemes differs as following: in DB-MPC, it does not complicate the real-time implementation as the solution is to be found analytically. In FCS-MPC and CCS-MPC, computing the inverse of the differential inductances matrix at each iteration of the solver makes utilizing this model challenging in the submillisecond sampling time range. In FCS-MPC, the problem is solved by enumeration, and thus, the number of iterations is *fixed* based on three factors: the number of VVs, the prediction horizon length, and the number of the states. In CCS-MPC, the number of iterations is *correlated* to the prediction horizon length and the number of the states, but it is not affected by the number of VVs of the power converter. Particularly with respect to the increase in the length of the prediction horizon, CCS-MPC scales better than FCS-MPC due to the different nature of the optimization problem.

<sup>4</sup>This procedure is typically to be done at a speed that lies in the base operating speed range of the motor under consideration. For machines that operate at a very wide speed range, the parameters can be obtained at different speeds, and then a 3-dimensional LUT is to be used, i.e. letting the parameters to be functions of  $i_d$ ,  $i_q$ , and  $\omega_{me}$ .



**Figure 2.12:** The identified and smoothed nonlinear model parameters.

Therefore, for the nonlinear CCS-MPC scheme that is proposed in the upcoming chapter, the following approximations are made in order to simplify the model<sup>5</sup>:

- The time derivative of the permanent magnet flux linkage  $\frac{d\psi_p}{dt}$  is set to zero [76].
- The cross-saturation is to be neglected. Hence, the mutual differential inductances  $L_{dq}$  and  $L_{qd}$  tend to zero [78]. This has a minor effect in prediction as their values are typically one order of magnitude less than the self differential inductances  $L_{dd}$  and  $L_{qq}$  [79].
- The self differential inductances  $L_{dd}$  and  $L_{qq}$  are highly correlated to the absolute inductances  $L_d$  and  $L_q$  [76]. Slight differences occur only at high motor speeds. Thus, it is assumed that  $L_{dd} \approx L_d$  and  $L_{qq} \approx L_q$  [54].

<sup>5</sup>To validate these assumptions for the IPMSM used in the experimental validation in this dissertation, a nonlinear analytical DB-MPC was designed and tested while incorporating both the complete nonlinear model shown in Eq. 2.15, and the simplified nonlinear model in Eq. 2.17, and only minor closed-loop performance enhancement was observed in transients when using the detailed one, therefore, the simplified nonlinear model is further considered in this work.

With these three simplifications, the simplified nonlinear dynamic model of PMSMs can be stated in the form

$$u_d = R_s i_d + L_d^{(i_d, i_q)} \frac{di_d}{dt} - \omega_{el} L_q^{(i_d, i_q)} i_q, \quad (2.17a)$$

$$u_q = R_s i_q + L_q^{(i_d, i_q)} \frac{di_q}{dt} + \omega_{el} L_d^{(i_d, i_q)} i_d + \omega_{el} \psi_p^{(i_q)}, \quad (2.17b)$$

which can be efficiently used in CCS-MPC schemes as will be shown in the following chapter.

On the contrary to this, the vast majority of the published predictive control schemes in the literature, whether its direct or indirect control for PMSMs, consider the use of simple LTI models (i.e. the model in Eq. 2.17 with constant parameters) [56, 80–84]. This choice is typically based on the abstract fact that such models are easy to obtain, and cheap to be used online from a computational perspective. This choice necessitates the use of online parameter estimators, disturbance observers, or the use of an integral loop with MPC schemes to compensate for the effects that yield from the discrepancy between this linear model and the real nonlinear plant. Despite the fact that such complementary choices enhance the steady-state performance of the controller and may achieve an offset-free tracking, the use of nonlinear models is essential for the optimal MPC transient performance, as long as such a model is available and implementable from computational point of view.

One of the main objectives of this dissertation is to motivate the use of nonlinear models as well as any knowledge available about the drive system as explained in the section of the inverter nonlinear effects, as well as the voltage and current measurement synchronization. Doing this in practice is a necessity to show the full potential of MPC schemes.

Nevertheless, it has to be noted that the use of error integrating functionalities will still be needed even when using nonlinear models. This assures an offset-free tracking performance in case of disturbances or parameters change over time due to various reasons such as components' aging or temperature dependencies.

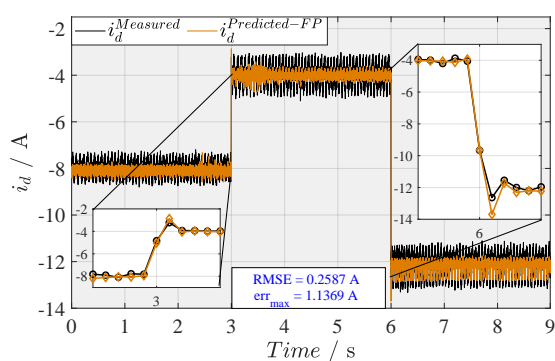
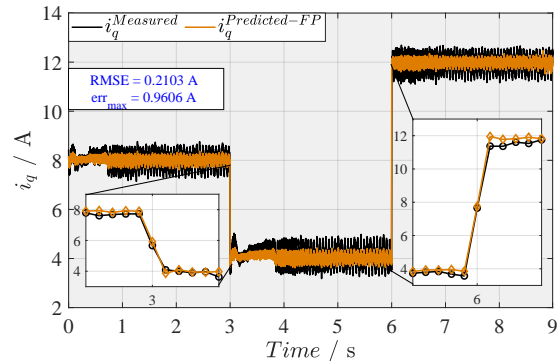
## 2.5.2 Validation of the nonlinear PMSM model based on the physical laws

In order to validate the model accuracy, the prediction function based on the model in Eq. 2.15 was deployed on the dSPACE to run in parallel with the current controller<sup>6</sup>. Steps of the reference  $i_d$  current from  $-8$  A to  $-4$  A and then to  $-12$  A, and of the reference  $i_q$  current from  $8$  A to  $4$  A, and then to  $12$  A were made at three different motor speeds of 250 rpm, 500 rpm, and 1000 rpm, and the results are depicted in Fig. 2.13, Fig. 2.14, and Fig. 2.15 for the three speeds, respectively.

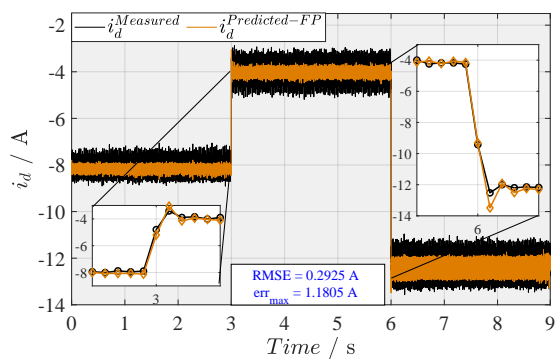
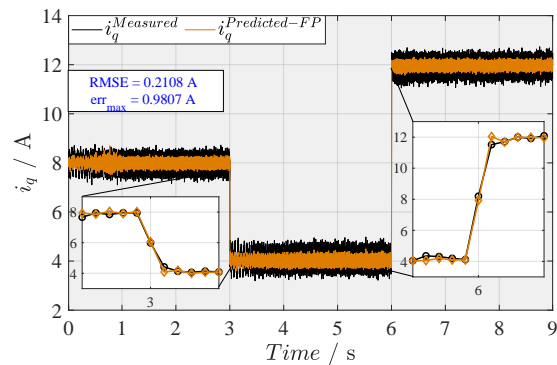
<sup>6</sup>In the results used here, the machine was being controlled via a DB-MPC controller. However, it can be validated with any other controller, for example, the model was tested to predict in parallel with a PI controller, and same conclusions were observed.



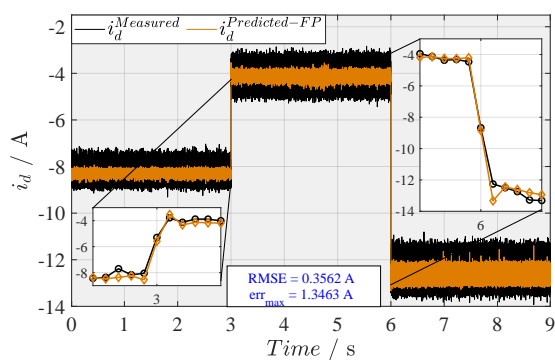
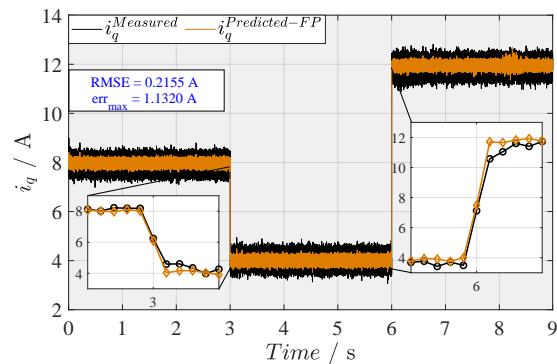
From these experimental results, it is clearly observed that the model can accurately predict the change of the currents for a given control input, and hence, it can be used for the purpose of designing a model-based current controller.

(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.

**Figure 2.13:** Experimental validation results of the nonlinear physics-based model at 250 rpm.

(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.

**Figure 2.14:** Experimental validation results of the nonlinear physics-based model at 500 rpm.

(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.

**Figure 2.15:** Experimental validation results of the nonlinear physics-based model at 1000 rpm.



## CHAPTER 3

---

### Data-driven modelling

---

*“What gets measured, gets managed.”*

Peter Drucker

In this chapter, the use of data-driven methods is proposed for the modelling task of a PMSM drive system as an alternative to the conventional modelling techniques introduced in the previous chapter. The chapter starts with a brief overview of the main motives behind the ongoing increasing interest in industry and academia in this regard, and a brief review regarding the most used data-driven modelling techniques specially for the use with MPC schemes. Following, an alternative data-driven inverter nonlinearity compensation approach is proposed by using a simple feedforward neural network (FNN) that learns the missing voltage as a function of the phase current for a given dead-time setup. The proposed data-driven inverter nonlinearity compensation approach has successfully mitigated these effects with less identification effort in comparison with the state-of-the-art method. After that, the rest of this chapter deals with the data-driven modelling of the current dynamics of synchronous machines, including a brief statement of the problem formulation and the design of the experiment to generate a proper training data set that is to be used for the data-driven modelling techniques. In particular, the use of long-short term memory (LSTM) NNs is proposed to purely model the current dynamics of the IPMSM from collected measurements and without prior knowledge, and it yields prediction results which have significantly outperformed the conventional LTI physics-based model and are in line with the prediction results from the nonlinear physics-based conventional model. For completeness, other data-driven modelling methods are reviewed for this application and their pros and cons in comparison with the chosen approach by using LSTM NNs are highlighted.

#### 3.1 Motivation and related work

Recently, genuine interest within the control and systems theory community has developed toward the use of data-driven methods in combination with MPC schemes [85], as an alter-

native to the conventional modelling approaches that is typically based on physical laws and first-principles. This interest is appealing and of high importance as these approaches address important issues in many industrial and real-life applications, some of which are:

- Saving time and exhausting modelling efforts, and in some cases costs.
- Having a better representation than the conventional models of real plants prone to noise, and hard-to-model- or even unmodelled dynamics by first-principles and physical laws.
- Requiring less domain expertise and pre-knowledge.
- Aiming for better scalability.
- Improving pre-defined performance indices in repetitive tasks by using a stochastic approaches to handle uncertainty with MPC schemes [86].

Motivated by one or more of the aforementioned motives, the following methods have been receiving increasing attention to be used for the modelling task purely from measured data while being applied to a wide set of applications:

- AI and machine learning (ML): approaches have been used for modelling complex dynamics in different domains and applications due to their characteristics as universal functions approximators. The range of applications in which these methods have been used for modelling include thermal modelling [87], modelling of quadrotor dynamics [88], modelling of vehicle dynamics for automated driving [89] as well as modelling the tire dynamics for the use in MPC [90], and as a generic black-box identification method [91–95].
- Gaussian process (GP)s: as an augmentation of a nominal dynamical model that aims to model the residual model mismatch/uncertainty applied to a racing car [96, 97], to model the aerodynamic effects for the control of quadrotors as presented in [98], and in [99] a GP model incorporating expert knowledge was proposed to design an air pressure controller of a diesel engine.
- KO: is a method in which the nonlinear dynamics are lifted into a higher dimensional space, where the states evolution is approximated linearly, which allows the use of the well-established linear model-based control methods [100]. This identification method was applied to soft robots as presented in [101], to electric power systems in [102], and was applied on nonlinear flows in [103].

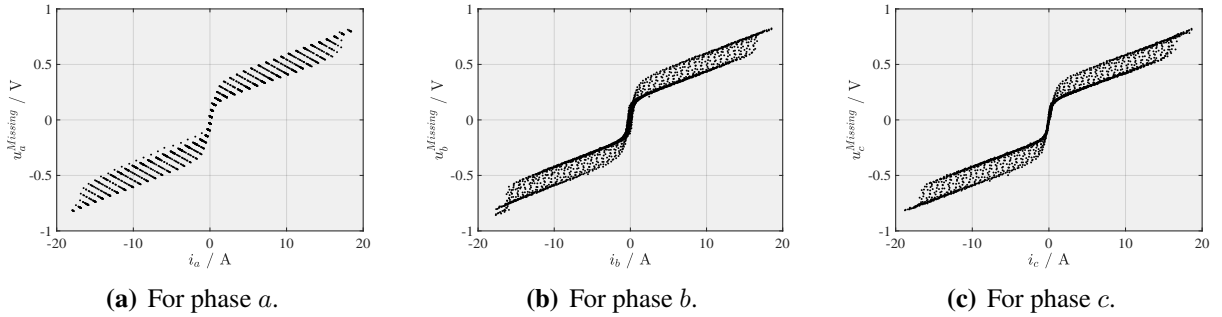
These methods differ from each other conceptually, and therefore, a careful investigation with respect to the needs, requirements, and characteristics of the application have to be carried out in order to check suitability. Factors such as training feasibility on large data sets, the degree of matching between the needed and available amount of prior knowledge about the system under identification, the computational complexity and model quality are essential in tipping the balance in favour of one method over the others. For the base data-driven modelling example considered in this dissertation that deals with learning the current dynamics of an IPMSM, these methods are considered. In the upcoming sections, the details regarding the challenges and benefits of using each of these methods are identified.

## 3.2 Data-driven inverter nonlinearity compensation

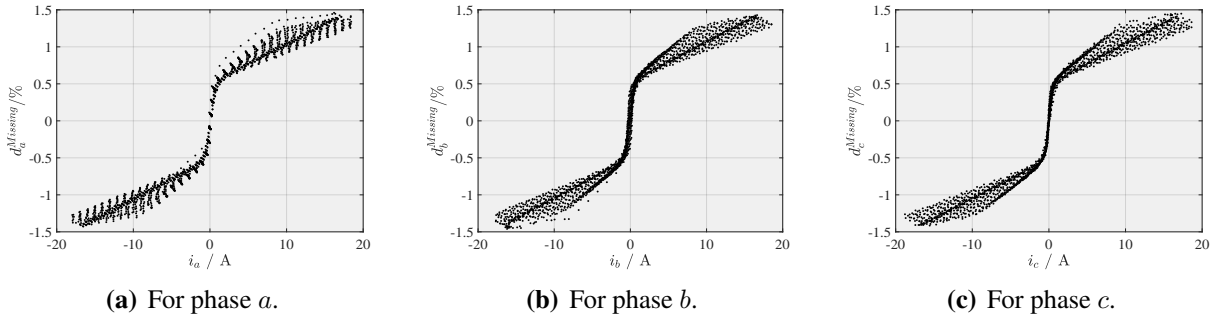
NNs are inspired with analogy to the biological neural system [104]. They are composed of connected artificial neurons that form a *layer*, where the stacked layers form together a NN. The artificial neuron is represented as a function that takes an arbitrary number of weighted and biased inputs, and its output can be mathematically expressed as

$$y_i = \sigma(a_i) = \sigma\left(\sum_{j=1}^n W_{ij}x_j + b_i\right) \quad (3.1)$$

with an  $i^{th}$  neuron in a layer and a  $j^{th}$  feature input to that specific neuron,  $\sigma$  denotes any arbitrary activation function, and  $n$  is the number of neurons in the previous layer (i.e. the number of input features in case of the input layer). In this section, a FNN approach is presented to approximate a function that compensates for the non-ideal behaviour of the 2-levels VSI. The motivation behind this choice is to have less modelling effort with the hope that only one test is needed to generate enough data to approximate the function, and hopefully, to take advantage of the ever-increasing development of efficient real-time execution of NNs [105]. To decide the inputs and outputs of the NN and in which reference frame, different representations for the same objective are to be visualized. In Fig. 3.1, the missing phase voltage is plotted against its phase current for all the points from the test in Fig. 2.6(d), and in Fig. 3.2 the missing phase duty cycle is plotted instead for the same measurements. From these two visualizations, it is observed that the functions are not unique with respect to the corresponding phase current over the whole  $\alpha - \beta$  plane as shown in the test in Fig. 2.6(d).

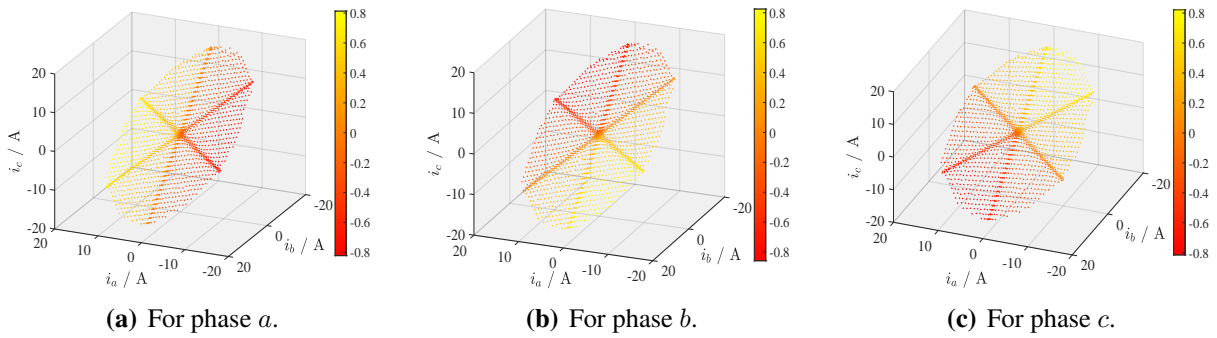


**Figure 3.1:** Correction phase voltages per the individual phase current.

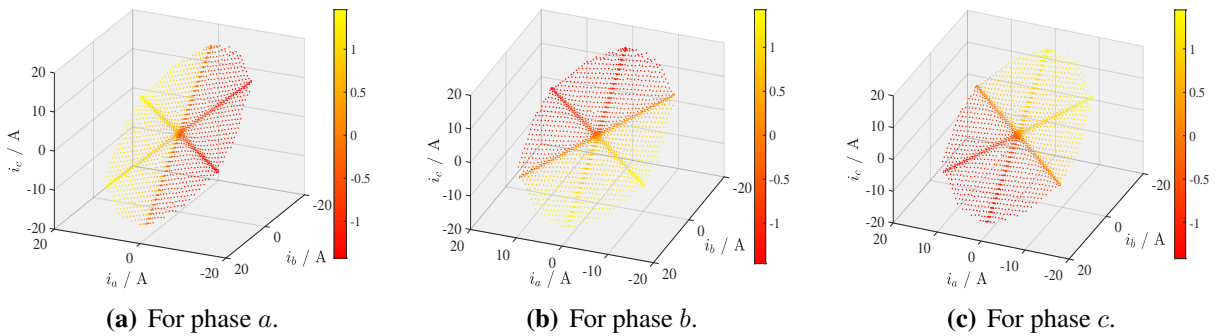


**Figure 3.2:** Correction duty cycles per the individual phase current.

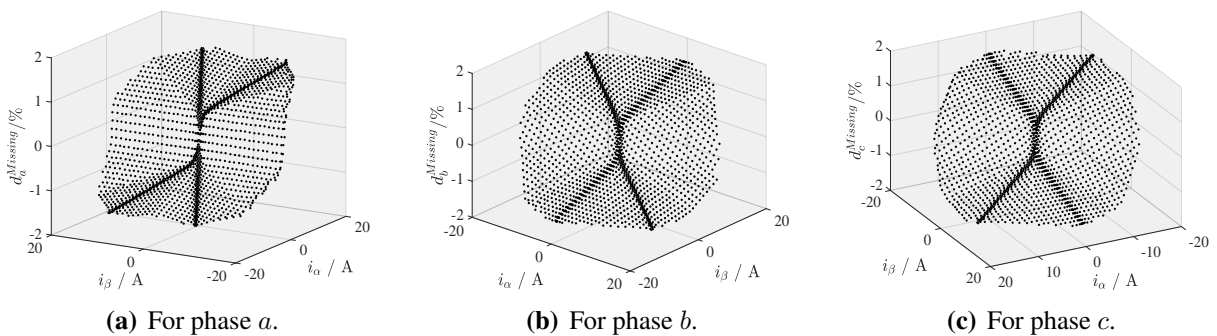
For this reason, a four-dimensional visualization for the correction phase voltage as a function of the three-phase currents is shown in Fig. 3.3, and for the correction duty cycle as a function of the three-phase currents in Fig. 3.4. Here it becomes clear that the functions are unique, and hence, these measurements can theoretically be used to train a FNN to approximate it. However, in the  $\alpha - \beta$  plane, the number of inputs can be reduced from three to two. A visualization of the correction duty cycles as a function of the phase currents in the  $\alpha - \beta$  plane is shown in Fig. 3.5. Taking this choice, the NN will have three outputs (i.e. the three correction duty cycles). However, the most optimal choice from computational perspective would be to have the phase currents as inputs and the correction phase voltages as outputs, all in the  $\alpha - \beta$  plane.



**Figure 3.3:** Correction phase voltages as functions of the three-phase currents.

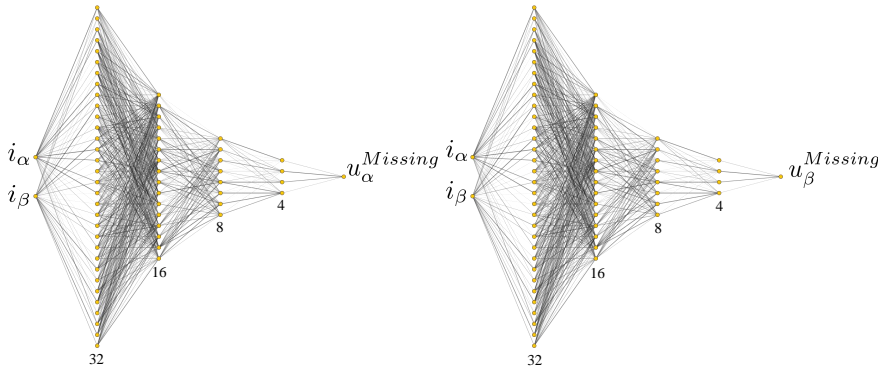


**Figure 3.4:** Correction phase duty cycles as functions of the three-phase currents.

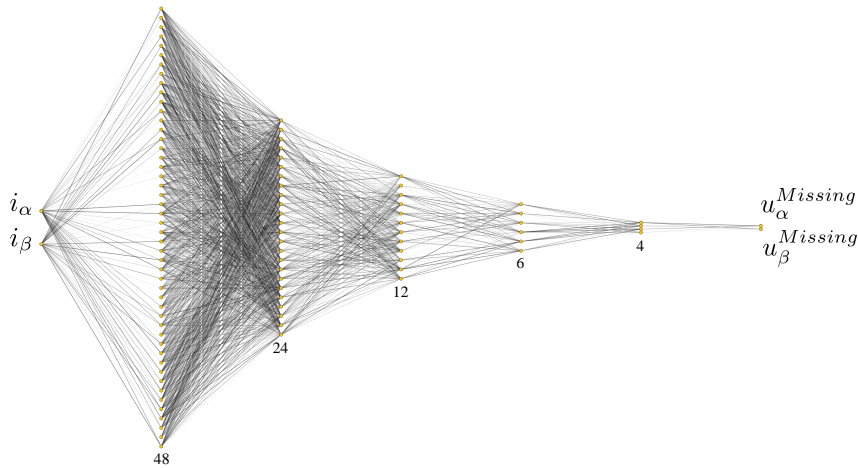


**Figure 3.5:** Correction phase duty cycles as functions of the  $\alpha - \beta$  currents.

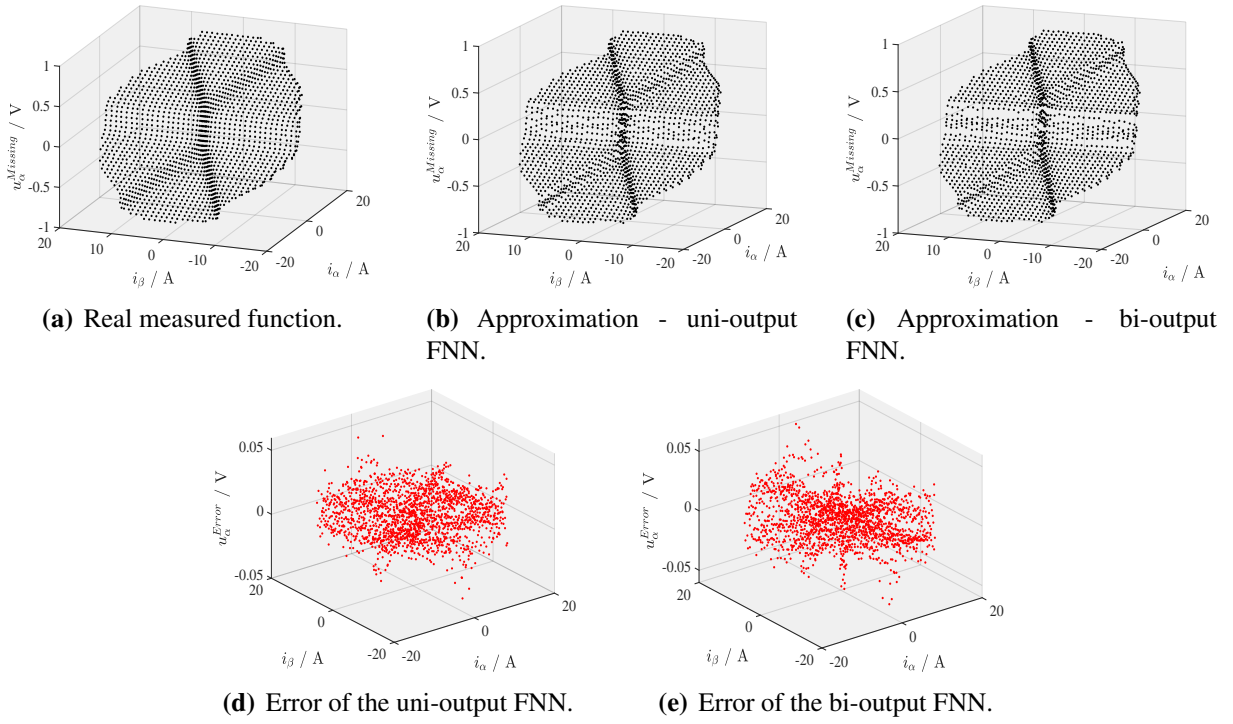
Two trials to approximate the  $\alpha$  and  $\beta$  correction voltages as functions of the phase current  $i_{\alpha\beta}$  were investigated: the first is to approximate them individually by two separate uni-output NNs, and the second is to use a bi-output NN to approximate both functions simultaneously. The chosen architecture of the uni-output NNs was identical to each other and consists of 2-32-16-8-4-1 neurons, and is depicted in Fig. 3.6, where the bi-output NN has an architecture of 2-48-24-12-6-4-2 neurons, and is depicted in Fig. 3.7. The functions to be approximated are shown in Fig. 3.8(a) and Fig. 3.9(a), respectively. The FNNs are trained using KERAS [106] in Python with TensorFlow backend [107]. The used loss function while training the regressor was mean squared error (MSE), and the optimizer is *adam*. The used activation function in all layers is rectified linear unit (ReLU). After the training, the outputs of both NNs are shown in Fig. 3.8(b)-(c) for the  $\alpha$ -axis correction voltage, and in Fig. 3.9(b)-(c) for the  $\beta$ -axis. In Fig. 3.8(d)-(e) and Fig. 3.9(d)-(e), the errors between the NNs outputs and the real measured functions are depicted. In both cases, the maximum approximation error was within 4%, which proves the effectiveness of the use of FNNs to approximate the correction voltage of the 2-levels VSI that is supposed to counteract the effects of its non-ideal switching behaviour. The proposed method might be beneficial in applications where an automatic correction method is needed, where the voltage deterioration effects do change over time (e.g. due to aging), and hence, repeating the training process with new collected measurements can be automated.



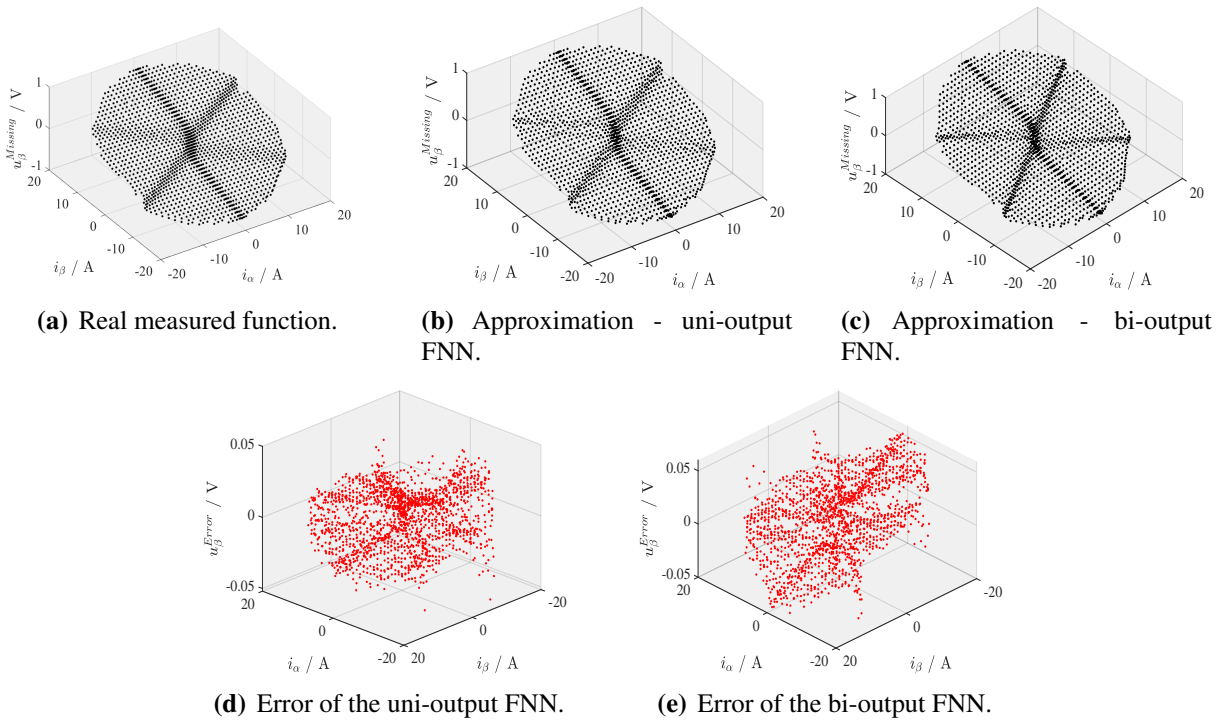
**Figure 3.6:** Uni-output NN architecture to fit the function of the inverter nonlinearity compensation.



**Figure 3.7:** Bi-output NN architecture to fit the function of the inverter nonlinearity compensation.



**Figure 3.8:** Missing  $\alpha$  – axis voltage as a function of  $i_{\alpha\beta}$ .



**Figure 3.9:** Missing  $\beta$  – axis voltage as a function of  $i_{\alpha\beta}$ .



**Conclusion:** The proposed data-driven method to learn the compensation of the non-ideal switching behaviour of a 2-levels VSI via a simple FNN with ReLU activation functions has shown excellent function approximation accuracy of  $> 96\%$  with respect to the conventional compensation method. The proposed data-driven method requires only one test to generate the training data set that covers the whole operation range of the currents in the  $\alpha - \beta$  plane, where the conventional method requires three tests to best capture the average compensation curve. Future investigations may include:

- The dependency on the  $dc$ -link voltage variation.
- Applying the proposed method to multi-level inverters with different topologies.
- Applying the proposed method to inverters with wide-bandgap semiconducting devices.

### 3.3 Problem statement of data-driven model identification

The model identification problem of discrete-time nonlinear dynamic systems can be stated to learn the dynamics in the general discrete-time state-space form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \underbrace{\mathbf{f}(\mathbf{u}_k, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_{u,max}}, \mathbf{x}_k, \mathbf{x}_{k-1}, \dots, \mathbf{x}_{k-n_{x,max}})}_{\delta \mathbf{x}_{k+1}: \text{ to be learned from data}}, \quad (3.2a)$$

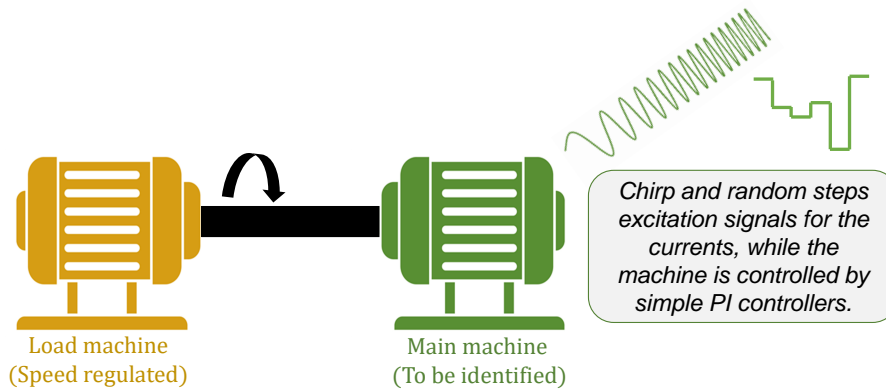
$$\mathbf{y}_k = \mathbf{h}(\mathbf{u}_k, \mathbf{x}_k), \quad (3.2b)$$

from measured data at dedicated persistent excitations, where  $\mathbf{f}(\cdot)$  represents the system dynamics function,  $\mathbf{h}(\cdot)$  is the output function,  $\mathbf{x}_k$  is the states vector,  $\mathbf{y}_k$  is the outputs vector, and  $\mathbf{u}_k$  is the vector of the control inputs, with  $n_{u,max}$  and  $n_{x,max}$  as the maximum number of lagged inputs and states, respectively. For the example considered in this chapter by learning the current dynamics of synchronous machines, the currents and the speed (i.e. the states) are measured via the corresponding sensors, and the commanded voltages from the controller represent the real phase voltages (i.e. the control inputs) once the inverter nonlinearity is compensated properly. Therefore, all of the variables here are known.

Once a data-driven model is obtained/learned, it can then be used within an MPC framework. Typically, a trade-off between model complexity and its prediction accuracy is to be made. Whenever the available sampling time is long enough, an approximate solution of the formulated constrained optimization problem is to be found online. Nevertheless, this becomes challenging when the demand on high switching frequencies (i.e. short sampling times) is imposed concurrently with the need to incorporate complex nonlinear models. In such cases, it is proposed in this dissertation to solve the optimization problem offline while incorporating an accurate, but computationally-expensive nonlinear model in order to generate a data set that represents the mapping between corresponding initial system states and the optimal control input that drives the system to a desired end state, and then to learn this mapping via a computationally implementable NN in a similar fashion as in explicit MPC [108, 109].

### 3.4 Design of identification experiment

The objective of this section is to present the used procedure to generate a training data set that represents the PMSM current dynamics in the rotating  $d - q$  reference frame to be used to train data-driven models from real measurements taken from the test bench shown in Fig. B.1 in Appendix B. The motor to be identified is excited by chirp and random steps reference signals using a PI current controller without the use of prior physical knowledge at a sampling frequency of 10 kHz, while the shaft speed was regulated via the load motor. The setup of the excitation test is sketched in Fig. 3.10.



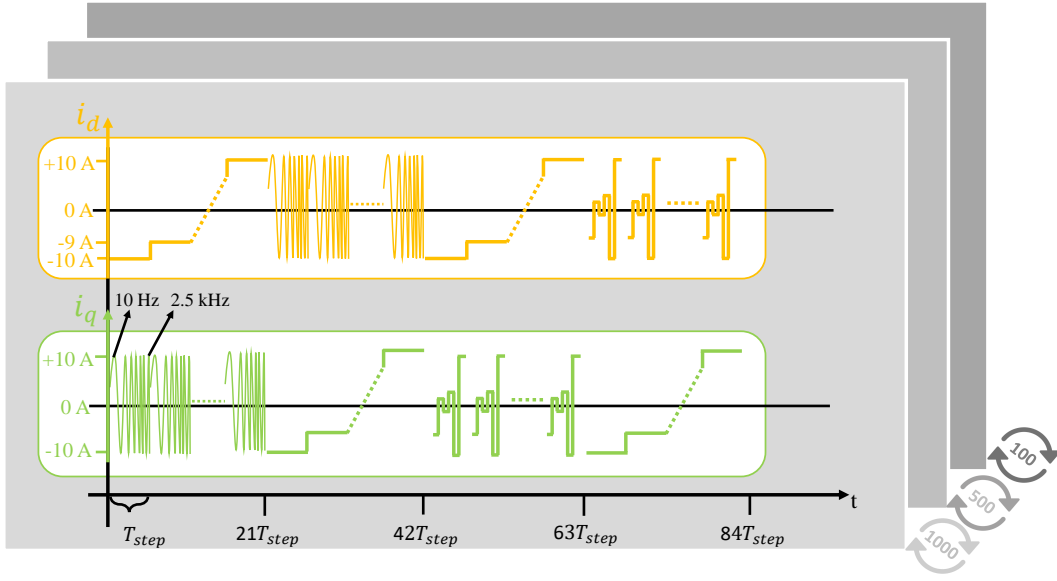
**Figure 3.10:** The setup of generating the excitation data set for the model identification task.

The excitation consists of the following four sections:

1. Firstly, the  $d - axis$  is excited with step reference values in the range from  $-10$  A to  $+10$  A with 1 A increments, while each step has a time duration of  $T_{step} = 5$  s. During each step duration, the  $q - axis$  is excited with a chirp reference with an amplitude of 10 A and frequency range from 2.5 kHz going down exponentially to 10 Hz.
2. In the second section, the excitation references between the  $d$  and  $q$  axes are swapped.
3. Thirdly, with the same step reference profile given in point (1) for the  $d - axis$ , the  $q - axis$  is excited with limited random amplitude references instead of the chirp reference.
4. In the last section, the profile illustrated in (3) is swapped between the two axes again.

The above mentioned procedure is repeated at three motor speeds of 1000 rpm, 500 rpm, and 100 rpm. In total, this yields an excitation data set that consists of 12.6 million sample points, which is equivalent to 21 min of excitation sampled at 10 kHz. This excitation profile is graphically illustrated in Fig. 3.11.

After the above illustrated identification test is conducted on the test bench, the measurements are collected and processed. The first step in processing the data is to align the measurements of the currents with the commanded voltages at each time step as explained in Sec. 2.4.3 in Fig. 2.11(b). Once this compensation is performed, the data is to be formed in an input-output



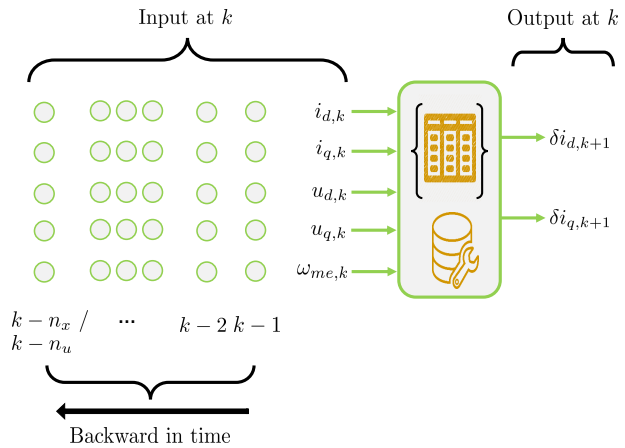
**Figure 3.11:** Sketch to depict the reference excitation signals of the motor to be identified.

form that represents the correct discrete-time current dynamics of PMSMs. The outputs are the measured changes of the stator current between each two consecutive discrete-time instants (i.e.  $i_{dq,k+1} - i_{dq,k}$ ), where the inputs are the stator currents and voltages as well as the shaft mechanical speed at the current discrete time instant  $k$  backward in time till  $k - n$ , with  $n$  as the number of the maximum needed discrete-time delayed steps. As the different inputs and outputs consists of different physical quantities and they range on different scales, normalization of all inputs and outputs helps for a faster convergence while training the data-driven model. Therefore, the inputs and outputs are normalized using the min-max normalization, such as

$$\xi' = l_l + \frac{(\xi - \min(\xi))(l_u - l_l)}{\max(\xi) - \min(\xi)}, \quad (3.3)$$

where  $\xi$  represents the original measured value to be normalized,  $\xi'$  is the scaled value,  $\min(\xi)$  and  $\max(\xi)$  are the minimum and maximum values of any arbitrary vector  $\xi$  to be normalized,  $l_l$  and  $l_u$  are the scaling lower and upper bounds, and are equal to -1 and 1, respectively.

By this, the processed data is ready to be used for the different data-driven modelling techniques, such as: NNs, GPs, and KO theory. A sketch that represents the training data for the learning task of PMSM current dynamics is depicted in Fig. 3.12.



**Figure 3.12:** Sketch to illustrate the resulting processed data set in an input-output form, to be used for the data-driven modelling.

### 3.5 Data-driven modelling of PMSMs via LSTM NNs

In this section, the use of LSTM<sup>1</sup> NNs to model the current dynamics of three-phase PMSMs is proposed. The model is to be obtained purely from real collected measurements from the test bench without prior physical knowledge. The LSTM NNs have been broadly used in different applications, such as speech recognition [111], handwriting recognition [112], and learning dynamical systems [93,94]. For the latter, once an LSTM model is trained, the learning abilities of the network structure is utilized to make system output predictions from given states and control inputs. An LSTM unit consists of three gates that govern the flow of information to the next unit in a stacked LSTM architecture, namely an input, a forget, and an output gate. Mathematically, the output of a stacked LSTM network is obtained by the manipulation of the inputs with the weights and biases obtained after the training process, such as

$$\mathbf{i}_k = \sigma_g(\mathbf{W}_i \boldsymbol{\chi}_k + \mathbf{Z}_i \mathbf{h}_{k-1} + \mathbf{b}_i), \quad (3.4a)$$

$$\mathbf{f}_k = \sigma_g(\mathbf{W}_f \boldsymbol{\chi}_k + \mathbf{Z}_f \mathbf{h}_{k-1} + \mathbf{b}_f), \quad (3.4b)$$

$$\mathbf{o}_k = \sigma_g(\mathbf{W}_o \boldsymbol{\chi}_k + \mathbf{Z}_o \mathbf{h}_{k-1} + \mathbf{b}_o), \quad (3.4c)$$

$$\tilde{\mathbf{c}}_k = \sigma_h(\mathbf{W}_c \boldsymbol{\chi}_k + \mathbf{Z}_c \mathbf{h}_{k-1} + \mathbf{b}_c), \quad (3.4d)$$

$$\mathbf{c}_k = \mathbf{f}_k \circ \mathbf{c}_{k-1} + \mathbf{i}_k \circ \tilde{\mathbf{c}}_k, \quad (3.4e)$$

$$\mathbf{h}_k = \mathbf{o}_k \circ \sigma_h(\mathbf{c}_k), \quad (3.4f)$$

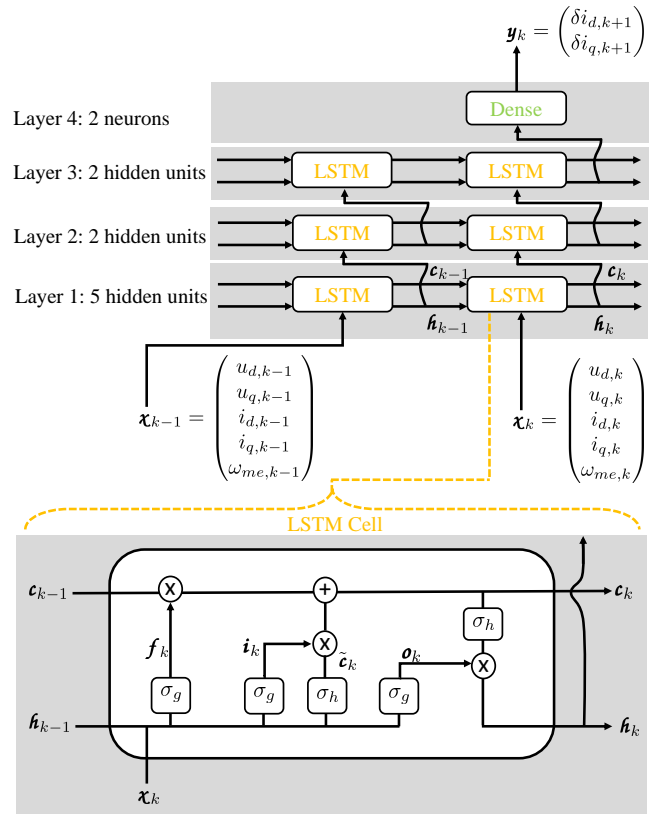
with  $\boldsymbol{\chi}$  representing the input vector to the network,  $\mathbf{h}$  as the hidden state/LSTM cell output vector (which goes as an input to the next layer), and  $\mathbf{c}$  representing the cell state vector. The vectors  $\mathbf{i}$ ,  $\mathbf{f}$ , and  $\mathbf{o}$  are the input gate, forget gate, and output gate activation vectors, respectively, and the cell input activation vector is  $\tilde{\mathbf{c}}$ . The functions  $\sigma_g$  and  $\sigma_h$  represent a sigmoid and a tangent hyperbolic activation functions, respectively. The operator  $\circ$  means element-wise multiplication. Once the model training process is performed, the optimized NN parameters are fixed. These parameters are all the weighting matrices  $\mathbf{Z}_*$  and  $\mathbf{W}_*$ , and the bias vectors  $\mathbf{b}_*$  for  $* \in \{i, f, c, o\}$  corresponding to input, forget, cell, and output, respectively.

Using real measurements obtained from the test bench and formed in an input-output data set as depicted in Fig. 3.12, an LSTM neural model is trained in KERAS [106] with TensorFlow backend [107] and with *adam* [113] as an optimizer. The training data set was divided into training, validation, and testing sets with percentages of 60%, 20%, and 20%, respectively, and a batch size of 56 was used for the training. Different models with different architectures were trained and tested, and have yielded excellent current prediction accuracy in both steady-state and transients. However, to facilitate the real-time implementation of the model to make predictions online, the most compact architecture found that yielded excellent prediction accuracy is selected. The chosen model consists of 3 LSTM layers with 5, 2, and 2 hidden units, respectively, followed by a dense layer with a linear activation function. In addition, the input features at the current and only one previous discrete time step (i.e.  $k$  and

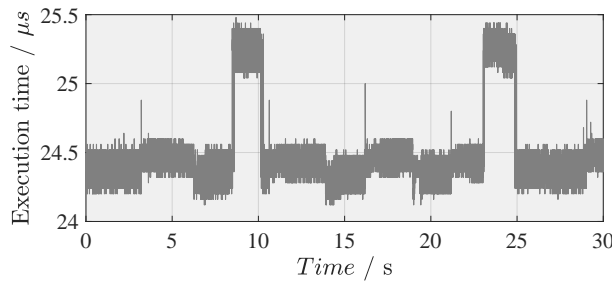
<sup>1</sup>In general, other types of NNs can also be used for the modelling task of nonlinear dynamical systems, such as multilayer perceptron (MLP), where the time delayed inputs are fed as additional inputs to the network. This will result in a different architecture, and consequently, different computational complexity of the obtained model. However, for applications with noisy measurements this choice did not show a superior performance [110].

$k - 1$ ) are fed to the network to minimize the execution time, as it is proportional to the number of previous discrete time steps to be fed to the network. An illustration of the used NN architecture is depicted in Fig. 3.13, with an illustration of all corresponding inputs and outputs, and depicting the mathematical operations within an LSTM unit.

For the real-time implementation of the LSTM NN on the dSPACE MicroAutobox II platform in order to make one-step currents' prediction, all the weights and biases are exported from python to MATLAB, and a self-written MATLAB function that emulates the LSTM network with the imported weights and biases is deployed on the dSPACE platform. This function runs in parallel to the current controller and its current predictions are recorded and compared with the real measured currents at the next sampling period  $k + 1$ . The execution time of the self-written LSTM MATLAB function on the dSPACE is recorded over time period of 30 s and it ranges from  $24.2 \mu\text{s}$  to  $25.5 \mu\text{s}$  as shown in Fig. 3.14. This is sufficient for using the neural model to make one-step current predictions in real-time within the submillisecond sampling time range with the needed time for sensors reading, processing, and controller execution. However, it is computationally expensive to be used as a model for *online* numerical optimization (i.e. online CCS-MPC schemes).



**Figure 3.13:** The architecture of the LSTM neural model.

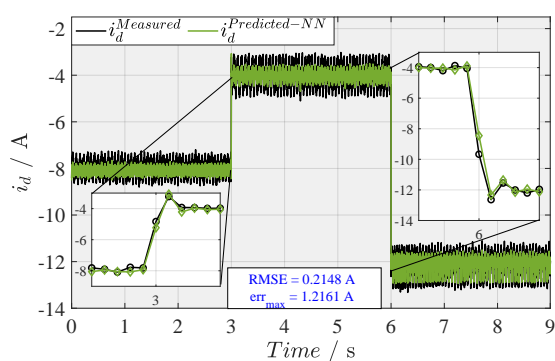


**Figure 3.14:** The execution time of the LSTM NN on the dSPACE MicroAutobox II platform.

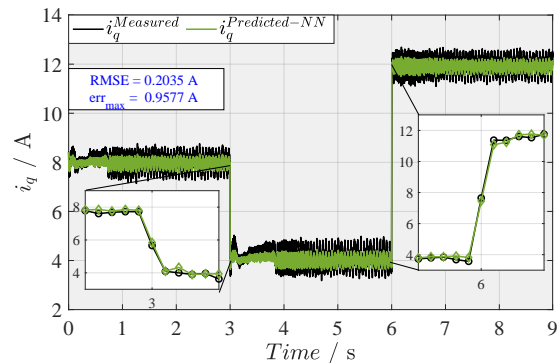
### 3.5.1 Validation of the LSTM neural model

Similar to the validation presented in Sec. 2.5.2, the aforementioned LSTM neural model is similarly tested and the results are shown in Fig. 3.15, Fig. 3.16, and Fig. 3.17.

Similar prediction accuracy on both axes of the current and at the three different tested motor speeds is obtained. This is verified through the maximum error and the root-mean-squared error (RMSE) values in each set of results as shown in the figures.

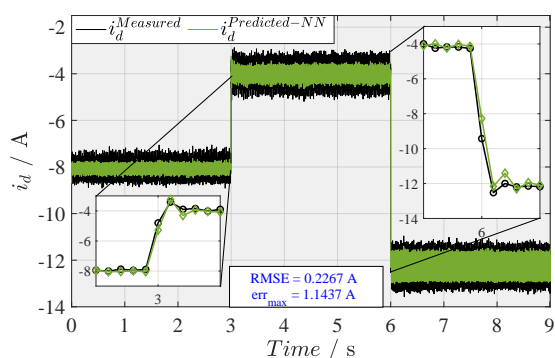


(a) Step changes of the  $d$  – axis current.

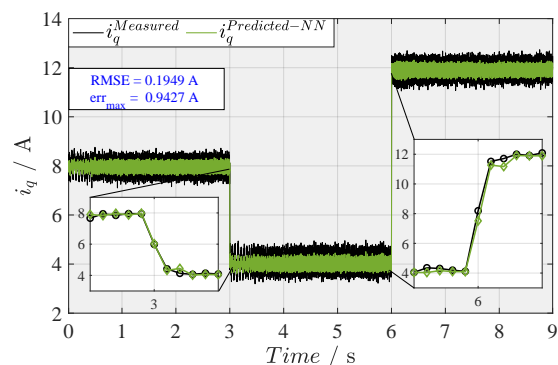


(b) Step changes of the  $q$  – axis current.

**Figure 3.15:** Experimental validation results of the LSTM model at 250 rpm.

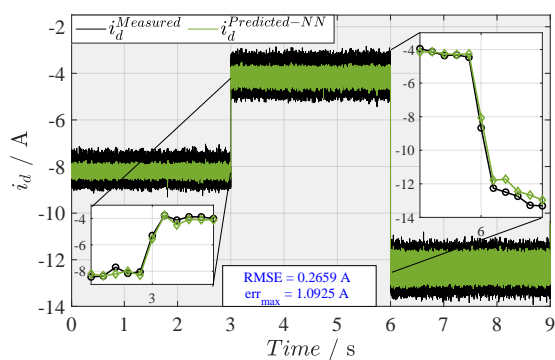


(a) Step changes of the  $d$  – axis current.

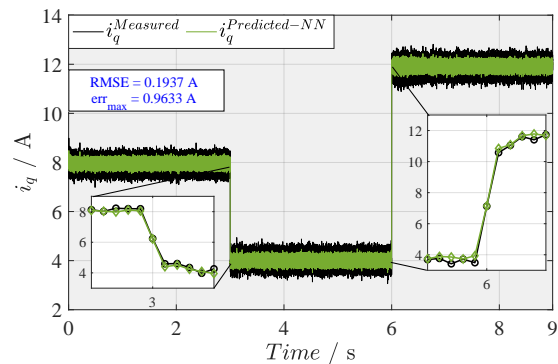


(b) Step changes of the  $q$  – axis current.

**Figure 3.16:** Experimental validation results of the LSTM model at 500 rpm.



(a) Step changes of the  $d$  – axis current.



(b) Step changes of the  $q$  – axis current.

**Figure 3.17:** Experimental validation results of the LSTM model at 1000 rpm.

## 3.6 Other potential data-driven modelling techniques

In this section, other data-driven modelling techniques for predictive control are briefly presented with application notes to power electronic and electrical drive systems. Among others, GPs and KO theory are investigated through the co-supervision of the Master Theses 1 and 2 in Appendix A.4, respectively, for the modelling task of the IPMSM.

### 3.6.1 Data-driven modelling of PMSMs via GPs

Considering a training data set  $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$  with the input  $\mathbf{X} \in \mathbb{R}^{L \times D}$ , and  $\mathbf{y} \in \mathbb{R}^{L \times 1}$  denoting its output, with the data set length  $L$  and input dimension  $D$ , a non-parametric GP model can be completely defined by a mean function  $\mu(\mathbf{x})$  calculated at an input  $\mathbf{x}$  and a covariance function  $k(\mathbf{x}_p, \mathbf{x}_q)$  between two arbitrary points  $\mathbf{x}_p$  and  $\mathbf{x}_q$  [114]. One of the most used covariance functions is the squared exponential kernel

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_s^2 e^{\frac{-1}{2\ell_D^2} \|\mathbf{x}_p - \mathbf{x}_q\|_2^2} + \sigma_n^2 \delta_{p,q}, \quad (3.5)$$

with the signal variance  $\sigma_s^2$ , the measurement noise variance  $\sigma_n^2$ , the length-scale  $\ell_D$ , and the Kronecker delta

$$\delta_{p,q} = \begin{cases} 0 & \text{if } p \neq q, \\ 1 & \text{if } p = q. \end{cases} \quad (3.6)$$

If the input space does not contain information about the uncertainty,  $\sigma_n^2$  becomes zero, and the signal variance  $\sigma_s^2$  does not affect the predictions [115]. Therefore, the squared exponential kernel simplifies to

$$k(\mathbf{x}_p, \mathbf{x}_q) = e^{\frac{-1}{2\ell_D^2} \|\mathbf{x}_p - \mathbf{x}_q\|_2^2}. \quad (3.7)$$

For more details regarding possible covariance functions, the reader is referred to [116].

To this end, in order to predict an output  $y \in \mathbb{R}$  at a given new input  $\mathbf{x} \in \mathbb{R}^D$ , the joint normal distribution  $\mathcal{N}(\cdot)$  between  $(\mathbf{X}, \mathbf{y})$  and the new input  $\mathbf{x}$

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N} \left( \mu(\mathbf{x}), \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) & k(\mathbf{X}, \mathbf{x}) \\ k(\mathbf{x}, \mathbf{X}) & k(\mathbf{x}, \mathbf{x}) \end{bmatrix} \right), \quad (3.8)$$

is used. A typical choice of the mean function  $\mu(\mathbf{x})$  is either zero or a polynomial that fits the data via linear regression. Alternatively, foreknowledge about the function to be learned can be included by a proper choice of  $\mu(\mathbf{x})$ . The covariance matrices are defined as

$$k(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_L) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_L) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_L, \mathbf{x}_1) & k(\mathbf{x}_L, \mathbf{x}_2) & \cdots & k(\mathbf{x}_L, \mathbf{x}_L) \end{bmatrix} \in \mathbb{R}^{L \times L}, \quad (3.9)$$

and

$$k(\mathbf{x}, \mathbf{X}) = (k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \dots, k(\mathbf{x}, \mathbf{x}_L)) \in \mathbb{R}^{1 \times L}, \quad (3.10)$$

and  $\mathbf{k}(\mathbf{X}, \mathbf{x}) = \mathbf{k}(\mathbf{x}, \mathbf{X})^\top$ . The output  $\mathbf{y} \sim \mathcal{N}(\mathbf{E}(\mathbf{x}), \text{Var}(\mathbf{x}))$  at a given arbitrary input  $\mathbf{x}$  can then be computed by the conditional distribution

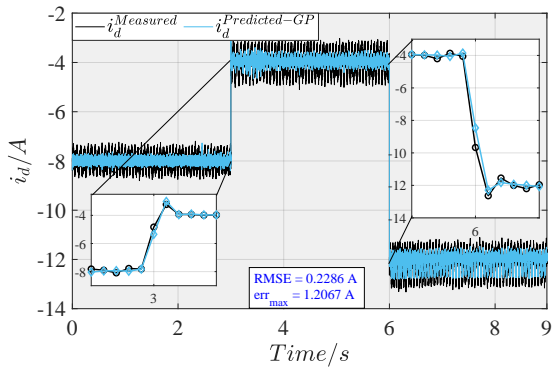
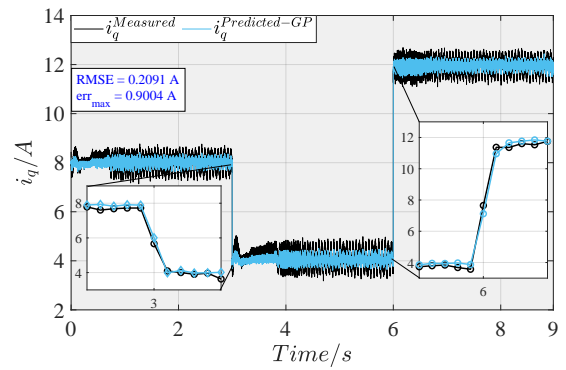
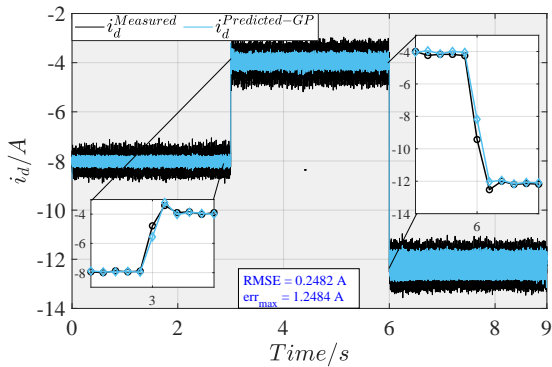
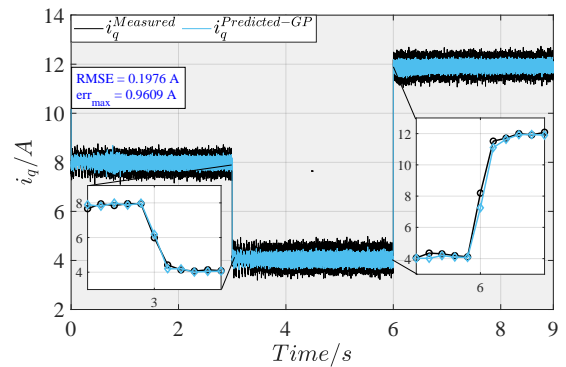
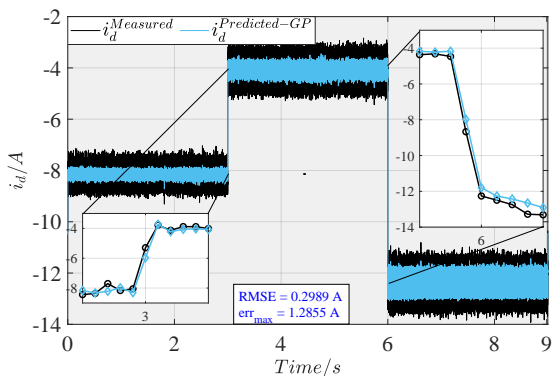
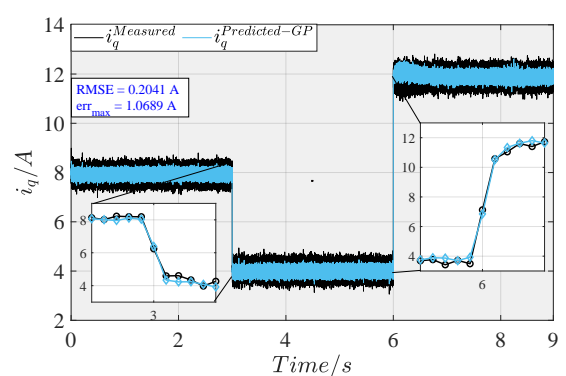
$$\mathbf{E}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) + \mathbf{k}(\mathbf{x}, \mathbf{X})(\mathbf{k}(\mathbf{X}, \mathbf{X}))^{-1}(\mathbf{y} - \boldsymbol{\mu}(\mathbf{X})), \quad (3.11a)$$

$$\text{Var}(\mathbf{x}) = \mathbf{k}(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x}, \mathbf{X})(\mathbf{k}(\mathbf{X}, \mathbf{X}))^{-1}\mathbf{k}(\mathbf{X}, \mathbf{x}). \quad (3.11b)$$

The expectation  $\mathbf{E}(\mathbf{x})$  (i.e. the predicted mean) is the predicted estimation of the output at the given arbitrary input  $\mathbf{x}$  with uncertainty represented by the variance  $\text{Var}(\mathbf{x})$ . For more details regarding the use of GPs for modelling, the reader is referred to [114].

From a computational perspective, training a standard GP model to optimize its hyper parameters requires the inversion and determinant of the covariance matrix of the training data set, which scales in  $\mathcal{O}(L^3)$  with the length of the data set  $L$ . Furthermore, computing predictions with the obtained GP model using Eq. 3.11 with pre-computing the inverse of the training data set covariance matrix  $(\mathbf{k}(\mathbf{X}, \mathbf{X}))^{-1}$  would still scale in  $\mathcal{O}(L)$  for the mean and in  $\mathcal{O}(L^3)$  for the variance [117]. This poor scalability with the length/size of the training data set, not to mention the needed high memory requirements, hinders the applicability of GPs in their standard form on large data sets, and makes it only applicable for data sets with size of up to  $10^4$  samples [117]. To overcome this shortcoming while retaining the desired model approximation accuracy, different approaches were proposed in the literature [118]. One of these methods is to obtain an approximate GP model with inducing points that reflect the original data via variational inference, where the size of the approximate GP model and the corresponding computational complexity are defined by the reduced number of the inducing points [119]. In the Master Thesis 1 in Appendix A.4, an approximate GP model of the training data used to model the current dynamics of the IPMSM is investigated, where a reduced number of representative data points from the original big data set is selected via a greedy algorithm, that aims to maximize the sampling coverage of the original data set with respect to size of the new data set. Once a subset of the original data set with the size  $\sim 10^4$  is chosen, an approximate GP model is trained using the software package GPyTorch with GPU acceleration [120]. The investigation made throughout this Master Thesis yielded that choosing  $10^4$  samples as a representative subset of the original training data set presented in Sec. 3.4, and then using the concept of inducing points to get an approximate GP model would require 0.63 ms for 1000 inducing points and 1.25 ms for 2000 inducing points to make one prediction once the approximate model is deployed on the dSpace MicroAutoBox II. Despite the significant approximations made, one evaluation of this data-driven model requires execution time in one order of magnitude higher than the available sampling time of 100  $\mu\text{s}$ , which hinders its use within a real-time capable numerical optimization framework. Nonetheless, such a model can still be used in an explicit-like MPC design by solving the MPC problem offline while incorporating this model, and learning the obtained control policy in an approximated computationally-implementable manner. The obtained approximate GP model is validated on the same data set on which the first-principles and the LSTM neural models were validated, and the results are shown in Fig. 3.18, Fig. 3.19, and Fig. 3.20. Despite the satisfactory prediction accuracy in terms of a maximum prediction error and the RMSE value, the approximate GP did not outperform neither the first-principles nor the data-driven model based on LSTM NN. In the light of this outcome while considering the computational challenges associated with this method, it is excluded in the further use of data-driven modelling for model-based control design in this dissertation.



(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure 3.18:** Validation results of the approximate GP model at 250 rpm.(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure 3.19:** Validation results of the approximate GP model at 500 rpm.(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure 3.20:** Validation results of the approximate GP model at 1000 rpm.

The following settings were chosen while training the approximate GP model in GPyTorch, with the voltages, currents, and rotor speed at the current discrete time instant  $k$  as inputs, and the change of the current (one model for  $\delta i_{d,k+1}$ , and one model for  $\delta i_{q,k+1}$ ) as an output:

- Mean function: *'ConstantMean'*
- Kernel: *'RBFKernel'*
- Variational strategy: *'IndependentMultitaskVariationalStrategy'*
- Variational distribution: *'CholeskyVariationalDistribution'*
- Optimizer: *'Adam'*
- Learn inducing points locations: *'True'*
- Learning rate: *'0.05'*
- Iterations: *'200'*

### 3.6.2 Data-driven modelling of PMSMs using KO theory

The basic concept behind using the KO theory as a modelling technique of nonlinear dynamical systems is to lift the nonlinear dynamics to a higher dimensional space, in which the states' evolution is approximately linear [121]. The KO  $\mathcal{K}$  is an infinite-dimensional linear operator corresponds to the dynamics of scalar observable functions (i.e. called observables) of the nonlinear dynamical system [122]. A finite-dimensional approximation of  $\mathcal{K}$  acting on finite-dimensional subspace of the observables can be used to predict the state variables which lie in the subspace of functions that the operator is truncated on [121].

Considering a classical state-space discrete-time dynamical system with  $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ , and a vector of scalar observables as nonlinear functions of its states and inputs, such as

$$\boldsymbol{\psi}(\mathbf{x}, \mathbf{u}) = (\psi_1(\mathbf{x}, \mathbf{u}), \psi_2(\mathbf{x}, \mathbf{u}), \dots, \psi_{n_\psi}(\mathbf{x}, \mathbf{u}))^\top, \quad (3.12)$$

the action of the KO  $\mathcal{K}$  on the vector of observables is given by

$$[\mathcal{K}\boldsymbol{\psi}](\mathbf{x}, \mathbf{u}) = \boldsymbol{\psi}(\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)) = \boldsymbol{\psi}(\mathbf{x}_{k+1}), \quad (3.13)$$

which propagates the observables vector, and hence, the system output forward [123]. To this end, having chosen a set of observables and having a training data set available, a finite-dimensional approximation of the infinite-dimensional KO  $\mathcal{K}$  can be computed by stating Eq. 3.13 as

$$\boldsymbol{\psi}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) = \boldsymbol{\mathcal{K}}\boldsymbol{\psi}(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\rho}_k \quad (3.14)$$

where the matrix  $\boldsymbol{\mathcal{K}}$  with the size of  $n_\psi \times n_\psi$  represents a finite-dimensional approximation of the KO, and the vector of residuals  $\boldsymbol{\rho}$  represents the approximation error. Common approaches to compute the matrix  $\boldsymbol{\mathcal{K}}$  include dynamic mode decomposition (DMD) [124] and extended dynamic mode decomposition (EDMD) [125]. In the EDMD, two snapshot matrices representing the vector of observables computed at consecutively equidistant discrete-time sampling with one sampling time step difference between the two matrices are defined, such as

$$\mathbf{\Pi} = \begin{bmatrix} | & | & | & | \\ \psi_1 & \psi_2 & \cdots & \psi_{L-1} \\ | & | & | & | \end{bmatrix}, \quad (3.15a)$$

$$\tilde{\mathbf{\Pi}} = \begin{bmatrix} | & | & | & | \\ \psi_2 & \psi_3 & \cdots & \psi_L \\ | & | & | & | \end{bmatrix}, \quad (3.15b)$$

and then  $\mathcal{K}$  can be approximated by solving a least-squares problem that minimizes the approximation error/residuals as

$$\min_{\mathcal{K}} \|\tilde{\mathbf{\Pi}} - \mathcal{K}\mathbf{\Pi}\|_F^2, \quad (3.16)$$

where  $\|\cdot\|_F^2$  denoting the Frobenius norm. The minimization problem has the analytical solution  $\mathcal{K} = \tilde{\mathbf{\Pi}}\mathbf{\Pi}^\dagger$ , in which the  $(\cdot)^\dagger$  denotes the Moore-Penrose pseudo-inverse [126].

This approach was investigated within the co-supervised Master Thesis 2 in Appendix A.4, and for more details, the reader is referred to the followed cooperative work in [75], where the KO theory was applied for the identification and real-time capable CCS-MPC design of PMSMs. It was shown in [75] that leveraging the physical knowledge when choosing the observables is advantageous in yielding a competitive approximation accuracy, and consequently, a good closed-loop MPC performance.

## 3.7 Conclusion

The investigation made in this chapter has highlighted the efficacy of data-driven modelling techniques in accurately capturing the current dynamics of IPMSMs. The results have demonstrated that these models can achieve prediction accuracy comparable to that of the nonlinear physics-based model. Among the investigated methods, LSTM NNs appear to be the most suitable approach for further consideration in designing model-based predictive controllers. This choice is based on the fact that the LSTM NN model did not require any prior physical knowledge, unlike the KO and GPs approaches where it is advantageous to incorporate physical knowledge. This suggests that the potential of LSTM NNs extends beyond IPMSMs, and can be utilized for modelling other power electronic and electrical drive applications where the dynamics are difficult to be captured using physical laws. Moreover, LSTM NNs exhibit scalability with large data sets from a computational perspective, which is a significant constraint for GPs. The obtained LSTM neural model will be leveraged to design a learning-based predictive current controller. The detailed discussion of this topic is presented in Ch. 5.



## CHAPTER 4

---

### First-principles model predictive control

---

*“The beauty and elegance of the physical laws themselves are only apparent when expressed in the appropriate mathematical framework.”*

Melvin Schwartz

In this chapter, conventional and advanced predictive control techniques based on the linear and the nonlinear physics-based models<sup>1</sup> for electrical drive applications are thoroughly investigated. The conceptual differences between predictive control and conventional control methods are highlighted. Moreover, methods to overcome the typical shortcomings of predictive controllers, such as model mismatch effects and the high computational demand are proposed.

In predictive control, the cost function is designed to simultaneously serve multi control objectives (normally tracking a given reference, but in principle it can include any other objective such as limiting the switching frequency, improving the acoustic behaviour, and/or any other). Once an accurate model that describes the dynamics of the plant is available, the parameterized optimal control sequence that yields from a predictive controller is computed with respect to the constraints of the system states and the control inputs. On the one hand, fulfilling the input constraints with a long-enough prediction horizon is beneficial in guaranteeing an optimal dynamic operation of electrical drive systems with respect to the available energy, and allows the operation on the physical limit (e.g. in field-weakening operation). On the other hand, electrical drive systems do impose states constraints which must be fulfilled. Typically, these come in a form of a maximum allowable current for safe operation of the different system components (i.e. batteries, motors, and active or passive components).

---

<sup>1</sup>For the direct MPC section, the linear model is used in all simulations and experiments, where in the indirect MPC section both the linear and the nonlinear models are used, and the results of both are benchmarked.

## 4.1 Direct model predictive control

*Direct* MPC schemes rely on the discrete nature of power converters/inverters. In this class of control schemes for electrical drive systems, the optimal control input to be found and applied to the plant is one out of the finite number of possible VVs of the inverter used to feed the plant, and is typically applied for a complete sampling interval  $T_s$ <sup>2</sup>, as the case with DSC, DTC, and hysteresis-based predictive controller. Therefore, the OCP is typically solved by a combinatorial approach.

Recalling that  $\mathbf{u} \in \mathbb{R}^{N_u}$ ,  $\mathbf{x} \in \mathbb{R}^{N_x}$ , and  $\mathbf{y} \in \mathbb{R}^{N_y}$  are considered to be the inputs, states and outputs of the system, respectively, with  $N_u$ ,  $N_x$ , and  $N_y$  as number of inputs, states, and outputs, respectively, the LTI continuous-time model is stated as

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t) + \mathbf{D}_c\mathbf{v}(t), \quad \mathbf{x}(0) = \tilde{\mathbf{x}}_0 \quad (4.1a)$$

$$\mathbf{y}(t) = \mathbf{C}_c\mathbf{x}(t), \quad (4.1b)$$

where  $\tilde{\mathbf{x}}_0$  is the initial state,  $\mathbf{A}_c \in \mathbb{R}^{N_x \times N_x}$ ,  $\mathbf{B}_c \in \mathbb{R}^{N_x \times N_u}$ , and  $\mathbf{C}_c \in \mathbb{R}^{N_y \times N_x}$  are the continuous-time state-space system matrices. The vector  $\mathbf{v}$  represents a measured disturbance with an additional matrix  $\mathbf{D}_c$  in order to incorporate the effect of this disturbance in the model. In discrete-time, the LTI model can be stated in the following representation

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{v}_k, \quad \mathbf{x}_0 = \tilde{\mathbf{x}}_0, \quad (4.2a)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k, \quad (4.2b)$$

where  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are the discrete-time system matrices, and are defined as

$$\mathbf{A} = \mathbf{I} + \mathbf{A}_c T_s, \quad (4.3a) \quad \mathbf{B} = \mathbf{B}_c T_s, \quad (4.3b)$$

$$\mathbf{C} = \mathbf{C}_c, \quad (4.3c) \quad \mathbf{D} = \mathbf{D}_c T_s, \quad (4.3d)$$

with  $\mathbf{I}$  as the identity matrix.

In FCS-MPC, all possible system transitions (i.e. based on all possible switching states) are evaluated over a prediction horizon in the cost function, and then the first switching state of the optimal sequence which minimizes the cost function is chosen to be applied at the following complete sampling period (at equidistant time intervals) [21, 127].

For the considered base example in this dissertation of controlling the current of synchronous machines in the  $d - q$  frame, the optimization problem of the traditional FCS-MPC is stated as

---

<sup>2</sup>To overcome the cons of having relatively high steady-state ripples on the tracked variables caused by the lack of a modulator (i.e. applying only one VV at a complete sampling interval  $T_s$ ), derivatives of direct MPC schemes were proposed in the literature by applying the optimal VV not at equidistant time intervals but at a variable point in time in which so called variable switching point model predictive control (VSP-MPC), or by applying more than one vector in which so called multiple vector model predictive control (MV-MPC).

$$\min_{\mathbf{U}_{abc,k}} J_1(\mathbf{U}_{abc,k}) \quad (4.4a)$$

$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{P}_k\mathbf{u}_{abc,k} + \mathbf{D}\mathbf{v}_k, \quad \mathbf{x}(0) = \tilde{\mathbf{x}}_0 \quad (4.4b)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k, \quad (4.4c)$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{k+i} \leq \mathbf{x}_{max}, \forall i \in \{1, \dots, N\}, \quad (4.4d)$$

$$\mathbf{u}_{k+i} \in \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \mathbf{u}_5, \mathbf{u}_6, \mathbf{u}_7, \mathbf{u}_8\} \forall i \in \{0, \dots, N-1\}, \quad (4.4e)$$

with the system vectors and matrices being defined as

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1 - \frac{T_s R_s}{L_d} & \frac{L_q T_s \omega_{el,k}}{L_d} \\ -\frac{L_d}{L_q} T_s \omega_{el,k} & 1 - \frac{T_s R_s}{L_q} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \frac{T_s}{L_d} & 0 \\ 0 & \frac{T_s}{L_q} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ \mathbf{D} &= \begin{bmatrix} 0 & 0 \\ 0 & -\frac{T_s \psi_p}{L_q} \end{bmatrix}, \mathbf{P}_k = \frac{u_{dc}}{3} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \end{bmatrix} \mathbf{T}_{abc}, \\ \mathbf{x}_k &= \begin{pmatrix} i_{d,k} \\ i_{q,k} \end{pmatrix}, \mathbf{u}_{abc,k} = \begin{pmatrix} s_{a,k} \\ s_{b,k} \\ s_{c,k} \end{pmatrix}, \mathbf{v}_k = \begin{pmatrix} 0 \\ \omega_{el,k} \end{pmatrix}. \end{aligned} \quad (4.5)$$

The cost function  $J_1$  is defined as

$$J_1 = \sum_{i=0}^{N-1} (1 - \lambda_u) \left\| \mathbf{y}_{k+1+i|k} - \mathbf{r}_{k+1+i|k} \right\|_2^2 + \lambda_u \left\| \Delta \mathbf{u}_{abc,k+i|k} \right\|_2^2 + \gamma_{k+1+i|k}, \quad (4.6)$$

where  $0 \leq \lambda_u \leq 1$  is a scalar weighting term to balance the priority between the tracking objective and limiting the energy/switching effort. The variable  $p \in \{1, 2\}$  is used to distinguish the possible use of either the  $\ell_1$  or the  $\ell_2$  norms, which are defined for any arbitrary vector  $\boldsymbol{\nu} = (\nu_1 \ \nu_2 \ \nu_3 \ \dots \ \nu_n)^\top$  as

$$\ell_1 : \|\boldsymbol{\nu}\|_1 = |\nu_1| + |\nu_2| + |\nu_3| + \dots + |\nu_n| \quad (4.7a)$$

$$\ell_2 : \|\boldsymbol{\nu}\|_2^2 = \boldsymbol{\nu}^\top \boldsymbol{\nu} = \nu_1^2 + \nu_2^2 + \nu_3^2 + \dots + \nu_n^2, \quad (4.7b)$$

where  $|\cdot|$  denotes the absolute value of a scalar. This penalty on the change of the control inputs  $\Delta \mathbf{u}_{abc}$  (i.e. in this case it is directly on the switching states transition), is used to limit/regularize the average switching frequency  $f_{sw,avg}$ . This is essential especially when prediction horizons longer than one are utilized. When having a sole tracking objective beside the regularization of the change of the control inputs, using a scalar weighting factor as introduced in Eq. 4.6 is typically sufficient.

It has been reported in the literature that the use of the  $\ell_2$  norm is recommended over the  $\ell_1$  norm despite the slight increase of the computational expense, as it yields a significantly better closed-loop control performance in terms of stability [128]. This is in particular recommended when operating at relatively low switching frequencies, where the  $\ell_1$  was reported to yield a deteriorated closed-loop control performance. Furthermore, the use of the  $\ell_2$  norm yields

significantly better THD at the same average switching frequency in comparison with the  $\ell_1$  norm as reported in [129], especially for high-order systems and systems that operate at low frequency.

The term  $\gamma$  in Eq. 4.6 represents the soft constraints, and is defined for the case of current tracking as:

$$\gamma_{k+1+i|k} = \begin{cases} 0, & \text{if } \sqrt{i_{d,k+1+i|k}^2 + i_{q,k+1+i|k}^2} \leq I_{max}, \\ \infty, & \text{otherwise,} \end{cases} \quad (4.8)$$

with  $\sqrt{i_{d,k+1+i|k}^2 + i_{q,k+1+i|k}^2}$  as the amplitude of the stator current at the corresponding future sampling period  $I_{k+1+i}$ , and  $I_{max}$  denotes the maximum allowed current amplitude value. Constraints of any other output (i.e. torque) depending on the control objective, can be formulated similar-wise.

Conceptually, FCS-MPC controllers in their simplest form consider the prediction for only one step ahead, and the objective of the cost function aims to minimize the tracking error of the system outputs to given references while simultaneously limiting the control effort without violating the states constraints. To this end, this describes the simplest form of direct predictive controllers in the field of electrical drives by controlling only one physical quantity (e.g. the current), and with/without penalizing the control effort while fulfilling the constraints. More advanced forms of direct predictive controllers that exploit more potential of the ability to forecast the system response are for example those which utilize long prediction horizons, while simultaneously penalizing the switching effort, hence, optimizing the current trajectory over the prediction horizon with respect to a given amount of switching effort/frequency. Obviously, it is by now clear that direct predictive controllers are fundamentally different from direct self/torque controllers, which aim to regulate the states to within pre-defined bounds in a hysteresis/bang-bang manner.

In order to visualize the concept of the simplest form of FCS-MPC controllers, a simulation of an FCS-MPC controller with the sole objective of tracking constant-amplitude stator current of a synchronous machine is carried out at a constant rotational speed and assuming an ideal inverter behaviour. The reference current (in black dashed red), the current constraint represented by the circle inscribed within the hexagon (in red), the possible current transitions (in green) with the optimal transition (i.e. closest to the reference) marked in dark orange, as well as the measured current trajectory (in blue dashed magenta) with the initial current measurement at each time step marked as  $\times$  are visualized step by step for 50 consecutive sampling intervals in Fig. 4.1 to illustrate the controller behaviour.

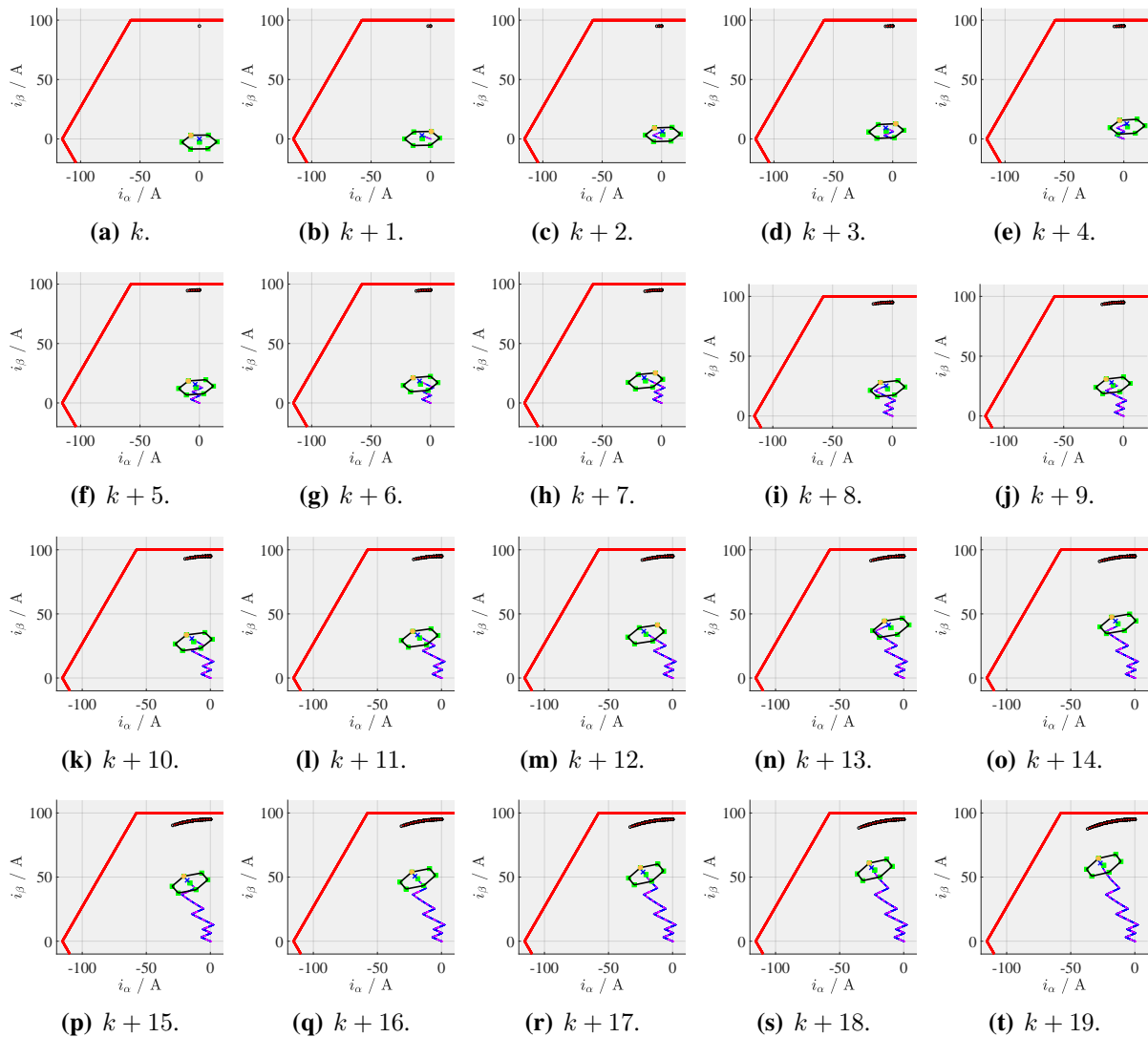
It worth to be mentioned that increasing the prediction horizon without adding a penalty on the control effort (i.e. limiting the switching frequency) will yield the exact same solution as the one-step FCS-MPC as reported in [129], and hence, does not improve the closed-loop control performance.

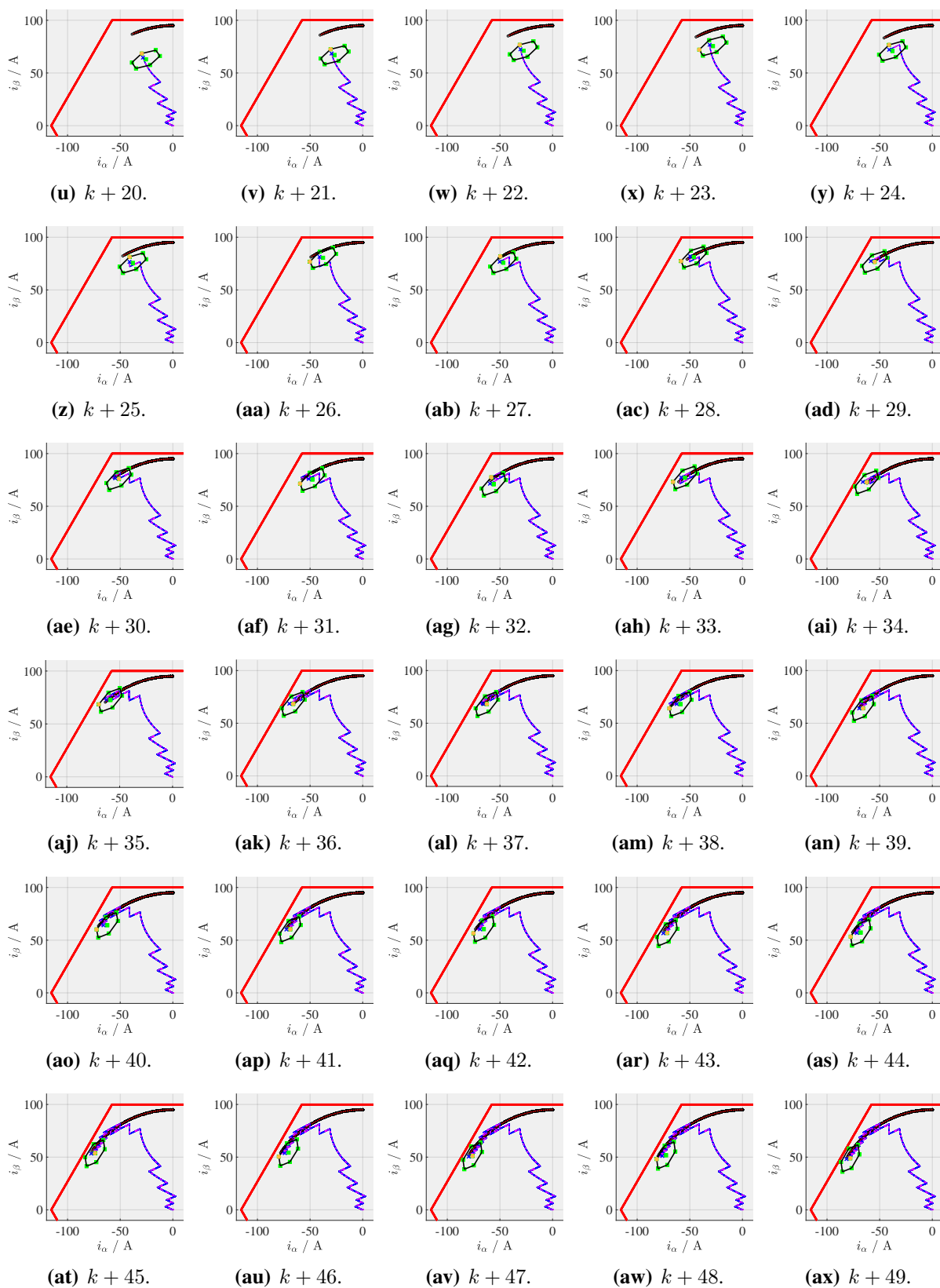
This visualization manifests the intrinsic characteristics of the direct variant of MPC in power electronic and electrical drive applications, such as

- The logic behind the controller aims to minimize the tracking error, and therefore, it takes the fastest possible way to the reference.



- The switching frequency is variable (unless measures are taken to fix its average by proper tuning of the corresponding weighting factor).
- The fulfilment of the input constraints is inherently included without the need of any extra limitation or projection.
- The states constraints can be easily included in the cost function, and therefore, can be straightforwardly fulfilled as long as an accurate states prediction model is available.
- The computational demand is directly proportional to the number of VVs of the inverter. Only the terms which are functions of the control inputs have to be computed for each VV, where all other terms in the prediction equation need to be computed only once. This makes this control strategy benefit from computing platforms in which parallel computing is used.
- The ripple is directly proportional to the sampling time and inversely proportional to the switching frequency.





**Figure 4.1:** Visualization of the operation concept of the one-step FCS-MPC for an example of current tracking of synchronous machines.

### 4.1.1 One-step FCS-MPC

In this subsection, the conventional FCS-MPC controller for the current control task of synchronous machines is briefly introduced, and a computationally-efficient algorithm that yields similar performance in one-third less computational time is proposed. Experimental results on the SMPMSM shown in Fig. B.2 in Appendix B are provided for validation of both algorithms under same test scenarios.

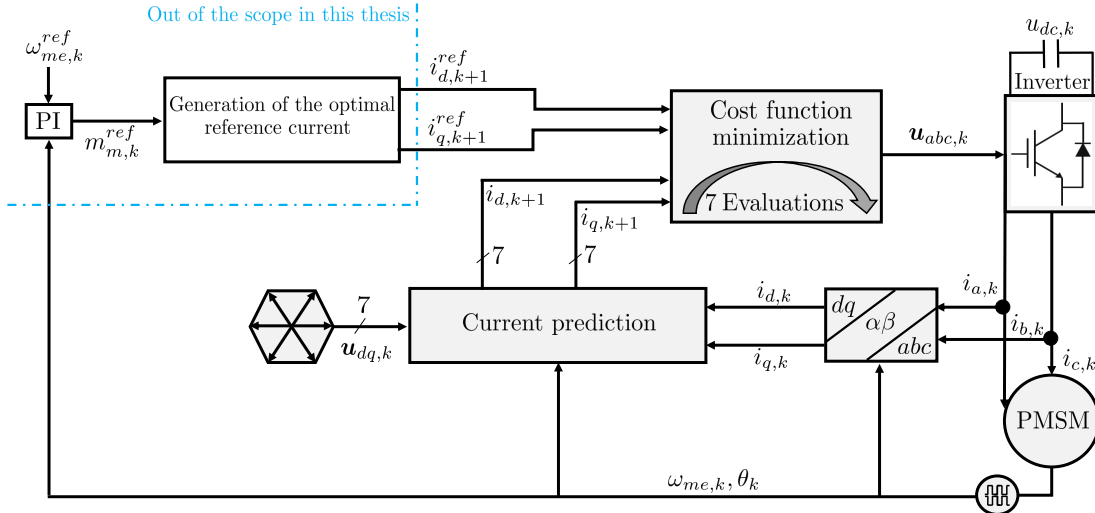
#### 4.1.1.1 Conventional one-step FCS-MPC

Conventionally, the discrete-time current prediction model is used to predict all possible current values at the upcoming sampling period corresponding to the finite-set of possible VVs. The LTI current prediction model of synchronous machines in the  $d - q$  reference frame is stated as

$$i_{d,k+1} = \left(1 - \frac{T_s R_s}{L_d}\right) i_{d,k} + \frac{L_q}{L_d} T_s \omega_{el,k} i_{q,k} + \frac{T_s}{L_d} u_{d,k}, \quad (4.9a)$$

$$i_{q,k+1} = \left(1 - \frac{T_s R_s}{L_q}\right) i_{q,k} - \frac{L_d}{L_q} T_s \omega_{el,k} i_{d,k} + \frac{T_s}{L_q} u_{q,k} - \frac{T_s \omega_{el,k} \psi_p}{L_q}. \quad (4.9b)$$

Using this model, seven predictions of each current are to be made corresponding to the seven unique switching states of the 2-levels VSI. For the current control task, the output vector is  $\mathbf{y} = (i_d \ i_q)^\top$ , and the reference vector is  $\mathbf{r} = (i_d^{ref} \ i_q^{ref})^\top$ <sup>3</sup>. By this, the cost function in Eq. 4.6 is to be evaluated seven times for all the predictions, and the VV that is associated to the prediction that yields the minimum cost is to be applied at the discrete time instant  $k$ .



**Figure 4.2:** Conventional FCS-MPC scheme for synchronous machines.

<sup>3</sup>As the focus throughout this dissertation concerns the current loop, the generation of the optimal reference current is out of the scope. For more details in that regard, the reader is referred to [130–132], and the references therein.

### 4.1.1.2 Computationally-efficient one-step FCS-MPC

Even though that with the available computational power nowadays most of the one-step FCS-MPC algorithms can be already executed in real-time, especially for inverters with relatively low number of VVs as the case with the 2-levels VSI, still, the reduction of the needed execution time can be beneficial. The main motivation behind this is that, the reduction of the needed execution time allows operating at higher sampling frequencies, and hence, getting higher average switching frequency that would yield less current ripple, and consequently, less THD values.

Using continuous formulation of the traditional FCS-MPC stated in Eq. 4.4 and subjected to the continuous input constraint

$$|\mathbf{u}_k| \leq u_{max} \quad (4.10)$$

with  $u_{max}$  being determined typically in the  $d - q$  as well as in the  $\alpha - \beta$  frames with respect to the dc-link voltage as:  $u_{max} = \frac{u_{dc}}{\sqrt{3}}$ , instead of the constraint in Eq. 4.4e, it is possible to determine a continuous-time optimal VV that is to be realized as an average over a sampling period via a modulation scheme. This continuous-time optimal reference VV is to be computed analytically by substituting the predicted currents in Eq. 4.9 with their reference values in a deadbeat fashion, and then re-arranging the equation to solve for the reference VV that will drive the currents to their references<sup>4</sup>

$$u_{d,k}^{ref} = R_s i_{d,k} + \frac{L_d}{T_s} (i_{d,k+1}^{ref} - i_{d,k}) - L_q \omega_{el,k} i_{q,k}, \quad (4.11a)$$

$$u_{q,k}^{ref} = R_s i_{q,k} + \frac{L_q}{T_s} (i_{q,k+1}^{ref} - i_{q,k}) + L_d \omega_{el,k} i_{d,k} + \omega_{el,k} \psi_p. \quad (4.11b)$$

In order to fulfill the constraints on the analytically-computed continuous-time optimal VV in Eq. 4.10, the computed VV in Eq. 4.11b is to be projected to within the circular constraints by scaling its magnitude, which is defined as

$$u_{mag,k} = \sqrt{(u_{d,k}^{ref})^2 + (u_{q,k}^{ref})^2}, \quad (4.12)$$

by the corresponding scaling [134], such as

$$u_{d,k}^{ref} = \begin{cases} u_{d,k}^{ref} & u_{mag,k} \leq u_{max} \\ \frac{u_{dc}}{\sqrt{3}} \frac{u_{d,k}^{ref}}{\|u_{dq,k}^{ref}\|} & u_{mag,k} > u_{max} \end{cases}, \quad u_{q,k}^{ref} = \begin{cases} u_{q,k}^{ref} & u_{mag,k} \leq u_{max} \\ \frac{u_{dc}}{\sqrt{3}} \frac{u_{q,k}^{ref}}{\|u_{dq,k}^{ref}\|} & u_{mag,k} > u_{max} \end{cases}. \quad (4.13)$$

One possibility here that has been proposed in literature [44, 45, 135, 136] is to locate the sector in which  $u_{dq,k}^{ref}$  lies in the stationary  $\alpha - \beta$  frame after applying inverse park transformation, after computing its angle as

$$\phi_k^{ref} = \tan^{-1} \left( \frac{u_{\beta,k}^{ref}}{u_{\alpha,k}^{ref}} \right). \quad (4.14)$$

<sup>4</sup>An alternative expression that considers the one-step delay compensation as in [133], and an incremental state-space model is proposed in details in [46].

However, the computation of  $\tan^{-1}$  is demanding in real-time, and therefore, this contradicts with the main motivation and intention behind reducing the computational complexity of the traditional one-step FCS-MPC schemes, especially for 2-levels power converters. In the following, a simpler rule-based geometrical algorithm is proposed to locate the sector in which the analytically-computed reference VV lies without the need to compute  $\tan^{-1}$ . The sector allocation algorithm is presented in Algorithm 1.

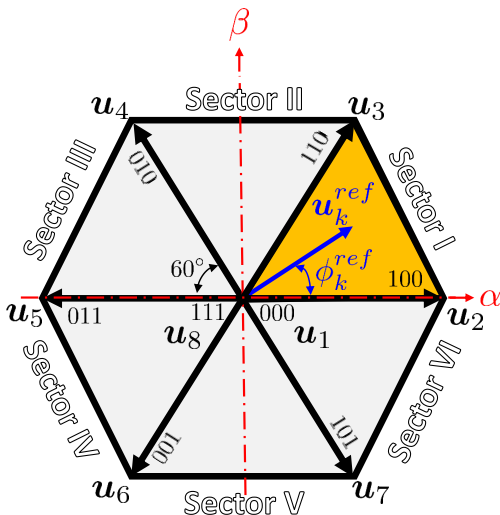
Once the sector is determined, only three evaluations of a modified cost function

$$J_2 = \frac{1}{2}(u_{\alpha,k}^{ref} - u_{\alpha,k}^{r\iota})^2 + \frac{1}{2}(u_{\beta,k}^{ref} - u_{\beta,k}^{r\iota})^2 + \gamma_{k+1}, \quad (4.15)$$

with  $\iota \in \{1, 2, 3\}$ , and subjected to the reduced set of VVs

$$\mathbf{u}_k \in \{\mathbf{u}^{r1}, \mathbf{u}^{r2}, \mathbf{u}^{r3}\}. \quad (4.16)$$

are required. The vectors  $\mathbf{u}^{r1}$ ,  $\mathbf{u}^{r2}$ , and  $\mathbf{u}^{r3}$  contain always the two active VVs and either one of the zero vectors, that together form one out of the six sectors in the  $\alpha - \beta$  frame, and they are dynamically defined based on the location of the analytically-computed reference



**Figure 4.3:** Example of locating the reference VV in the  $\alpha - \beta$  frame.

VV. As an example, if  $\mathbf{u}_{\alpha\beta,k}^{ref}$  is found to be located as shown in Fig. 4.3, it lies in Sector I that is defined by the switching state vectors: 100, 110, and either one of the zero vectors. The proposed computationally efficient finite control set model predictive current control (FCS-MPCC) scheme is illustrated in Fig. 4.4.

---

#### Algorithm 1: Sector Determination

---

##### Sector Determination ( $\phi_k^{ref}$ )

**Step I:** Ratio between  $\beta$  and  $\alpha$  components

of  $\mathbf{u}_{\alpha\beta}^{ref}$

$$\text{Ratio} = \frac{u_{\beta}^{ref}}{u_{\alpha}^{ref}}$$

**Step II:** Quadrant and Sector

Determination

**if**  $u_{\beta}^{ref} \geq 0$  **then**

**if**  $u_{\alpha}^{ref} \geq 0$  **then**

**if**  $\text{Ratio} \leq \sqrt{3}$  ( $=\tan(60^\circ)$ ) **then**

$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_2, \mathbf{u}_3\}$  ;

**else**

$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_3, \mathbf{u}_4\}$  ;

**end**

**else**

**if**  $\text{Ratio} \leq -\sqrt{3}$  **then**

$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_3, \mathbf{u}_4\}$  ;

**else**

$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_4, \mathbf{u}_5\}$  ;

**end**

**end**

**else**

**if**  $u_{\alpha}^{ref} \leq 0$  **then**

**if**  $\text{Ratio} \leq \sqrt{3}$  **then**

$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_5, \mathbf{u}_6\}$  ;

**else**

$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_6, \mathbf{u}_7\}$  ;

**end**

**else**

**if**  $\text{Ratio} \leq -\sqrt{3}$  **then**

$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_6, \mathbf{u}_7\}$  ;

**else**

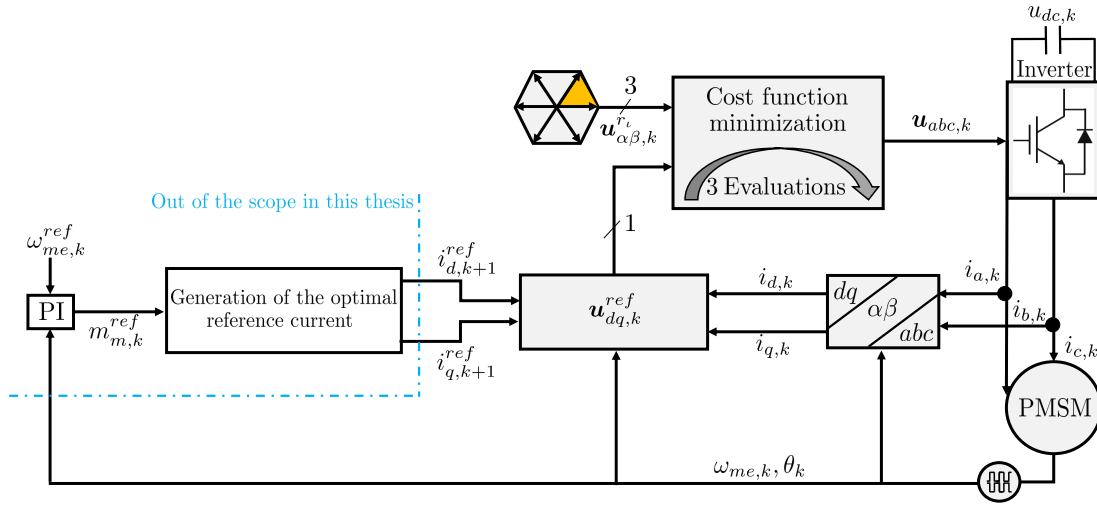
$\mathbf{u}_k \in \{\mathbf{u}_{1/8}, \mathbf{u}_7, \mathbf{u}_2\}$  ;

**end**

**end**

**end**

---

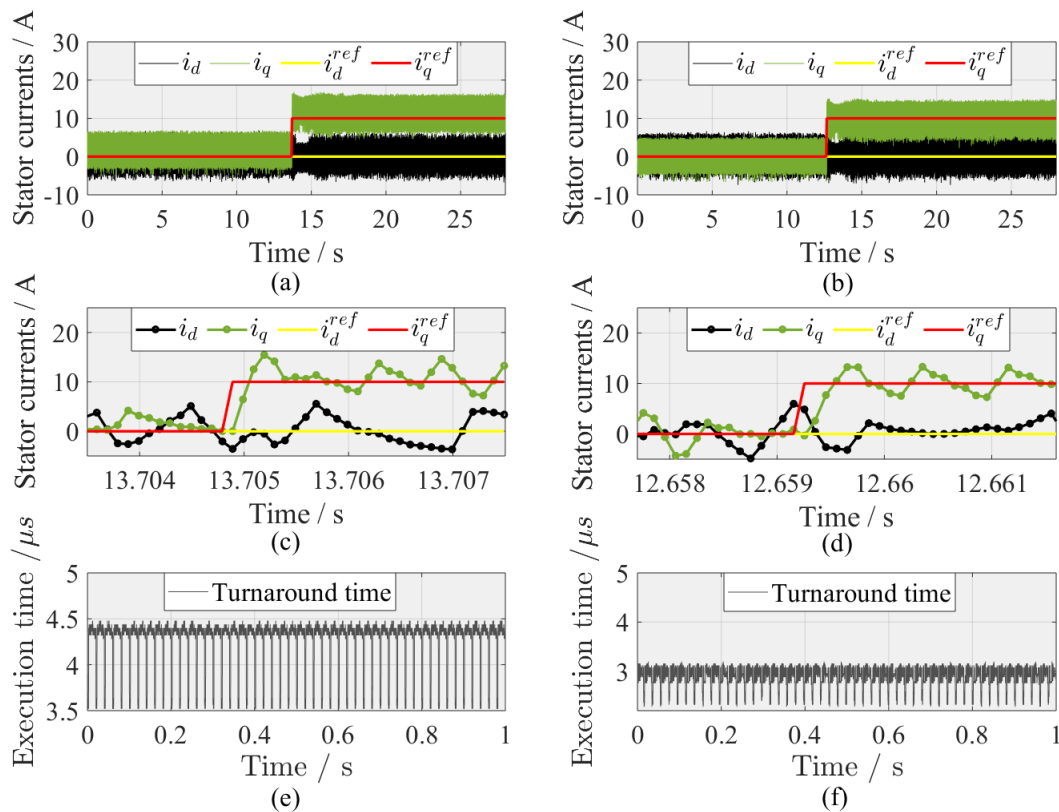


**Figure 4.4:** The Proposed computationally-efficient FCS-MPCC for synchronous machines.

#### 4.1.1.3 Experimental results and observations

In this section, the conventional as well as the proposed computationally-efficient one-step FCS-MPCC are experimentally validated on the SMPMSM shown in Fig. B.2 in Appendix B. Both methods are tested under same conditions and test scenario: the shaft speed was regulated via the load machine at  $90 \text{ rad s}^{-1}$ , and a step of the reference torque-producing current  $i_q^{ref}$  from 0 A to 10 A is made while  $i_d^{ref}$  is kept at 0 A. The steady-state reference and measured currents are shown for the traditional FCS-MPCC scheme in Fig. 4.5(a), and the same is shown for the proposed computationally-efficient FCS-MPCC scheme in Fig. 4.5(b). The transient instants of these two measurements are zoomed-in in Fig. 4.5(c) and Fig. 4.5(d) for the traditional and the computationally-efficient schemes, respectively. The control schemes were executed at same sampling frequency of 10 kHz, and the execution times of the control functions of both schemes are recorded on the dSPACE DS1007 platform in Fig. 4.5(e) and Fig. 4.5(f), respectively. From these results, the following is observed:

- The results show the general characteristics of FCS-MPC controllers of providing fast dynamical response and high current ripples, albeit with switching considerably less than indirect control schemes that use a modulator once sampled at the same sampling frequency. The high current ripples make direct control schemes not suitable for low-inductance machines and applications with strict THD limitations, unless it is sampled at very high sampling frequencies.
- The proposed computationally-efficient scheme provides similar performance with execution time of  $3.1 \mu\text{s}$  instead of  $4.5 \mu\text{s}$  for the traditional scheme. Even though this 31.1% reduction of the execution time might not be solving an issue while having a sampling period of  $100 \mu\text{s}$ , it can significantly push the maximum possible sampling frequency especially when the FCS-MPC schemes are deployed on FPGAs with sampling times of the sub-ten microseconds range. Sampling at such high frequencies will contribute in solving



**Figure 4.5:** Experimental results of the conventional and computationally-efficient FCS-MPCC schemes applied to control the SMPMSM in Appendix B.2.

the high-ripple issue with direct MPC methods [25, 137]. Moreover, the needed execution time of the proposed method is not dependent on the number of VVs of the used inverter as the case of the traditional FCS-MPC method, which makes the reduction of the execution time by the proposed method with respect to the traditional one even more significant for multi-level inverters.

- The proposed method is not supposed to outperform the traditional one in terms of performance such as providing better THD or less ripples when both methods incorporate the same model. To compensate for modelling mismatches and to provide offset-free tracking performance, an integral term can be added to the reference VV before the evaluations of the modified cost function in the proposed method as in [138], as an alternative of having a disturbance observer with the traditional scheme.

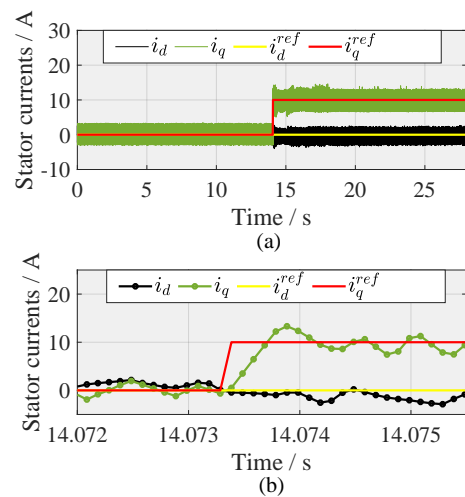
#### 4.1.1.4 Multiple-vector one-step FCS-MPC

Driven by the intention to reduce the high steady-state ripples that yield from applying only one VV at each sampling interval (i.e. the lack of a modulator) without increasing the sampling frequency, multiple-vector FCS-MPC methods have emerged under different names in the literature, such as: two- or three-vectors FCS-MPC, MPC based on virtual VVs, or modulated FCS-MPC [139–143]. The two-vectors FCS-MPC is typically employed by adding a zero VV within the duty cycle to damp the current gradients (i.e. to reduce the ripples), or generally by combining any two vectors. To reduce the current ripples of the computationally-efficient FCS-MPCC proposed in the previous subsection, the length of the active VVs is reduced by 50% by injecting a null VV at the first and last 25% of the sampling period, after adjusting the algorithm accordingly. As can be seen in Fig. 4.6(a), the current ripple is reduced by around 34%, while keeping the fast transient performance as shown in the zoom-in in Fig. 4.6(b). However, in multiple-vector FCS-MPC methods, two critical aspects should be noted:

- *Switching effort*: by applying more than one VV, will the switching effort be less than using a modulator? if not, then the motivation to use such a method instead of the well-known modulation schemes [144] with an indirect MPC (either DB-MPC or CCS-MPC) is questionable, if any. For example, in the method that proposes the use of null vector with the optimal active VV symmetrically by applying the active vector at the middle of the duty cycle, the switching sequence in case of 110 as an optimal vector would be: 000  $\rightarrow$  110  $\rightarrow$  000. The switching effort to yield the resultant VV in this case is more than the one needed in SVM using asymmetric pulsation with one zero vector, same as in SVM using symmetric pulsation with either one of the zero vectors, and slightly less than in SVM using symmetric pulsation with using both zero vectors.
- *Optimality*: must be revised. Once a second VV is to be added to the first optimal VV sequentially, global optimality of the first VV is not maintained, as it will not be applied to the sampling period  $T_s$  that it has been chosen optimal for.

From this discussion, the following questions arise: if the objective is to do modulation, is it meaningful to start with the optimal discrete-time VV out of the possible vectors within the finite-set of the power converter? In other words, is FCS-MPC a good starting point if the semi-conducting devices of each phase are to be switched more than once within a sampling period? How is it compared to CCS-MPC with a dedicated modulation scheme in that regard?

To achieve the objective of reducing the high steady-state ripples of FCS-MPC schemes, sampling at ultra high frequencies or using VSP-MPC methods seem to be more conceptually promising. For these reasons, the multiple-vector FCS-MPC methods are excluded from further investigations within this dissertation.



**Figure 4.6:** Experimental results of the computationally-efficient FCS-MPCC scheme with applying null vector applied to the SMPMSM in Appendix B.2.



## 4.1.2 Long-horizon FCS-MPC

### 4.1.2.1 Benefits of long-horizon FCS-MPC and the computational barrier

Utilizing long prediction horizons while limiting the average switching frequency proved to be beneficial in comparison with the one-step FCS-MPC at a given average switching frequency  $f_{sw,avg}$  [129, 145]. The steady-state performance enhancement associated with long horizon FCS-MPC schemes typically comes in the form of reduced current THD at a given  $f_{sw,avg}$  [145], and that it enhances the closed-loop system stability [146]. These enhancements become excessively remarkable for drive systems that operate at low switching frequencies (i.e. in the sub kHz range) [145], and for high-order systems [129]. Examples to prove these benefits were reported in [49, 147] for variable-speed drives with a 3-levels NPC-VSI.

Unfortunately, the computational demand to solve the mixed-integer OCP of FCS-MPC schemes increases exponentially with the length of the prediction horizon, which hinders the real-time implementation of long-horizon FCS-MPC schemes with the exhaustive-search combinatorial optimization. The number of the needed computations within a sampling period to find the optimal solution of FCS-MPC schemes as a function of the length of the prediction horizon  $N$  is expressed as

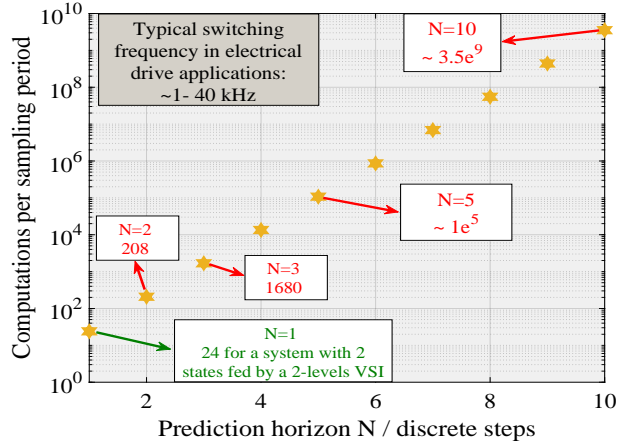
$$N_{CPSP} = n_x \times \underbrace{\sum_{i=1}^N N_{VV_s}^i}_{\text{prediction of states}} + \underbrace{N_{VV_s}^N}_{\text{cost function}}, \quad (4.17)$$

where  $N_{CPSP}$  is the number of computations per each sampling period,  $n_x$  is the number of states, and  $N_{VV_s}$  is the number of all possible VVs of the used converter to feed the plant.

To visualize how the needed computations exponentially increase with the prediction horizon, an example for a system with two states fed by the conventional 2-levels VSI is depicted in Fig. 4.7 with the needed computations for prediction horizons from 1 to 10 steps. Strategies to achieve long-prediction horizons with real-time implementation for power electronic systems, mainly based on branch and bound methods, were proposed in the literature with a detailed review in [48].

It has to be noted that the switching effort must be penalized in the

cost function, otherwise, the one-step and the long-horizon FCS-MPC schemes will provide the exact same solution [129]. To visualize the benefit of long-horizon FCS-MPC over the one-step, a simulation study for the current control task of an IPMSM with the parameters shown in Table. 4.1 is carried out. Three controllers with prediction horizons of 1, 5, 8 steps were tuned

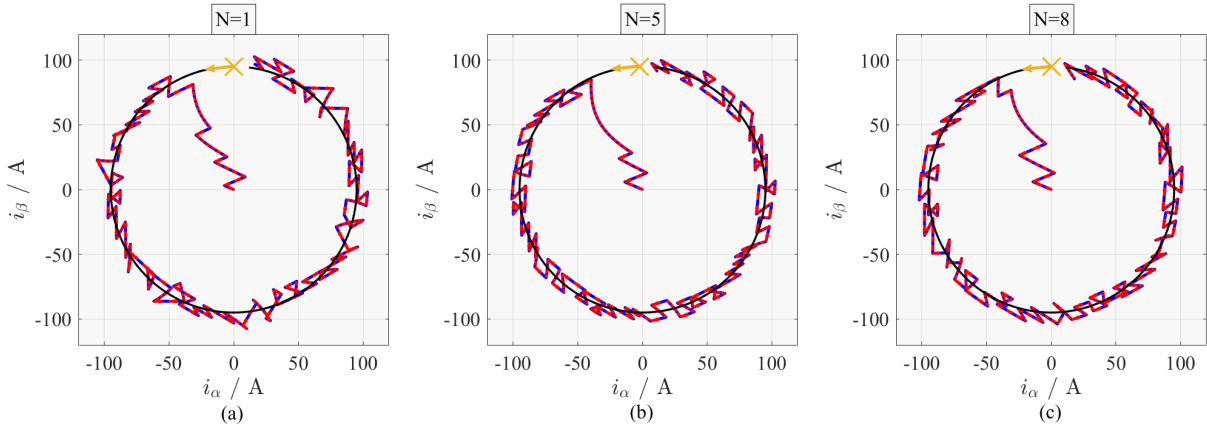


**Figure 4.7:** Required computations per each sampling period as a function of the prediction horizon  $N$  for FCS-MPC schemes for a system with two states.

to yield almost the same average switching frequency while being tested under same initial conditions of  $i_d = 0$  A and  $i_q = 0$  A, to track a constant reference current of  $i_d^{ref} = 0$  A and  $i_q^{ref} = 95$  A at a constant motor speed of 2000 rpm. For a better visualization, the first cycle of the three controllers in the stationary  $\alpha - \beta$  plane is depicted in Fig. 4.8. Visually, it can already be observed that the controllers with longer prediction horizons of 5 and 8 steps yield a more-in-harmony current than the one-step controller that seems to surge, in contrary to the controllers with longer prediction horizons that seem to plan the switching beforehand.

**Table 4.1:** Parameters of the IPMSM used for the simulation study in this section.

Name	Nomenclature	Value
dc-link voltage	$u_{dc}$	200 V
Maximum current	$I_{max}$	150 A
Stator resistance	$R_s$	12 m $\Omega$
$d - axis$ inductance	$L_d$	180 $\mu$ H
$q - axis$ inductance	$L_q$	400 $\mu$ H
Sampling frequency	$f_s$	50 kHz
Flux linkage	$\psi_p$	50 mWb



**Figure 4.8:** Steady-state comparison for one cycle of three FCS-MPC controllers with different prediction horizons of: (a)  $N = 1$ , (b)  $N = 5$ , (c)  $N = 8$ .

Translating this into an indicating evaluation metric, the reduction of the current THD with respect to the one obtained with the one-step controller is taken. This reduction can be measured via the relative current THD, which is defined as [129]

$$I_{THD,rel} = \left| \frac{I_{THD,long-horizon} - I_{THD,one-step}}{I_{THD,one-step}} \right|. \quad (4.18)$$

The current THD value for each controller in the  $\alpha - \beta$  frame is recorded in Table. 4.2, with the switching frequency of each phase as well as the average switching frequency recorded over 10 steady-state cycles. From these results it can be observed that at the same average switching frequency  $f_{sw,avg}$ , the controllers which utilized longer prediction horizons have achieved relative current THD improvement in the range from 10.6 % to 15.1 %. Despite that the overall

**Table 4.2:** Resultant THD and  $f_{sw,avg}$  of FCS-MPC controllers with different prediction horizons.

Prediction horizon	THD of $i_\alpha$	THD of $i_\beta$	$f_{sw,a}$	$f_{sw,b}$	$f_{sw,c}$	$f_{sw,avg}$
N= 1	7.08 %	7.19 %	3800 Hz	2950 Hz	3916 Hz	3555 Hz
N= 5	6.01 %	6.41 %	3500 Hz	3650 Hz	3500 Hz	3533 Hz
N= 8	6.33 %	6.11 %	3500 Hz	3575 Hz	3516 Hz	3530 Hz

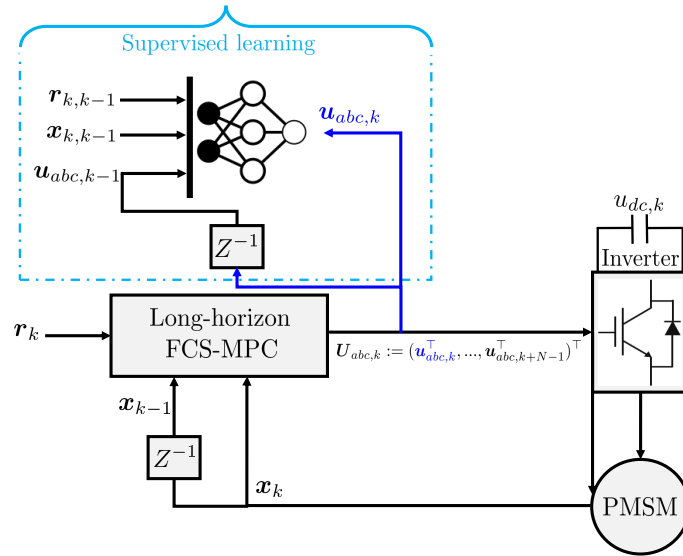
current THD content of the  $N = 8$  controller is not better than the one with the  $N = 5$  controller (i.e. better in the  $\beta$  direction but worse in the  $\alpha$  direction), its switching effort is better distributed over the three legs of the inverter, which is advantageous from derating perspective.

#### 4.1.2.2 *Computationally-implementable approximated long-horizon FCS-MPC based on NNs*

**Motivation and proposal** To benefit from the computationally-exhausting but beneficial optimal MPC control laws in real-time applications, explicit model predictive control (EMPC) is typically used. In EMPC, the control law is computed offline by solving a multiparametric programming problem for all system states using powerful computational resources without limiting sampling time constraints, then the optimal control policy is saved into a LUT that takes the current system states as its argument to provide the optimal control input [148]. In the electrical drives and power electronics community, this has been proposed with real-time implementation for a PWM inverter with an LCL filter in [149], and for the current control of an induction machine (IM) in [150]. However, this class of predictive controllers is associated with having a high memory footprint because the number of polyhedral regions is exponentially proportional to the number of constraints as well as the length of the prediction horizon [151]. An emerging approach to approximate MPC control laws to facilitate their real-time implementation is done by utilizing the ability of NNs as general function approximators, and hence, to mimic optimal control policies [152, 153].

In this subsection, the mixed-integer OCP stated in Eq. 4.4, and associated with the FCS-MPCC for synchronous machines is solved offline for a prediction horizon of  $N = 5$  for a range of operating points of interest, and then the first solution of the obtained optimal sequence is learned via a FNN with the hope to mimic the long-horizon controller with its beneficial performance in a computationally-implementable manner for electrical drive applications. An illustration of the training process of a NN controller that is aimed to mimic the long-horizon FCS-MPC performance is depicted in Fig. 4.9.

The VVs are classified as either voltage-increasing or voltage-decreasing for each phase [35]. From this perspective, the direct control problem can be seen as a multiclass classification problem with a finite-set of VVs representing the different classes. Based on the initial conditions, and whether the states need to have a positive or a negative slope to better follow their references at each discrete time step, a reduced finite-set of candidates can be applied as can be clearly seen from the possible phase voltages in the  $abc$  frame in Table 2.1. The proposal in this section is based on solving the long-horizon FCS-MPC problem for a proper excitation that



**Figure 4.9:** Training a NN to mimic the  $N = 5$  FCS-MPC controller.

covers the operating range of interest *offline* in order to generate a training data set for a FNN, that is aimed to approximate this control policy in a computationally-implementable manner, and that runs online as a controller in closed-loop. The chosen architecture for the case mentioned here consists of an input layer, one hidden layer, and an output layer. As the proposed NN controller is meant to be a direct controller, the number of neurons in the output layer is chosen to be equivalent to the number of VVs of the power converter, and hence, 8 neurons are used. A suitable choice for the activation function of the output layer is the softmax activation function, which squashes the outputs of the NN to probabilistic values that sum up to one. The neuron in the output layer that has the highest probability takes the binary value of 1, and every other neuron gets 0. By this, the one-hot neuron represents the suggested VV to be applied to the load according to the definition in Table 4.3.

$s_{abc}$	Class	One-hot encoding
000	1	10000000
100	2	01000000
110	3	00100000
010	4	00010000
011	5	00001000
001	6	00000100
101	7	00000010
111	8	00000001

**Table 4.3:** Classification of the discrete VVs of a 2-levels inverter with one-hot encoding.

The softmax activation is defined as

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{l=1}^n e^{z_l}}, \quad (4.19)$$

where  $\mathbf{z} = (z_1, \dots, z_n)^\top$ , and  $n$  representing the number of inputs to each neuron of the output layer. Therefore, the output of the FNN becomes an absolute probability distribution.

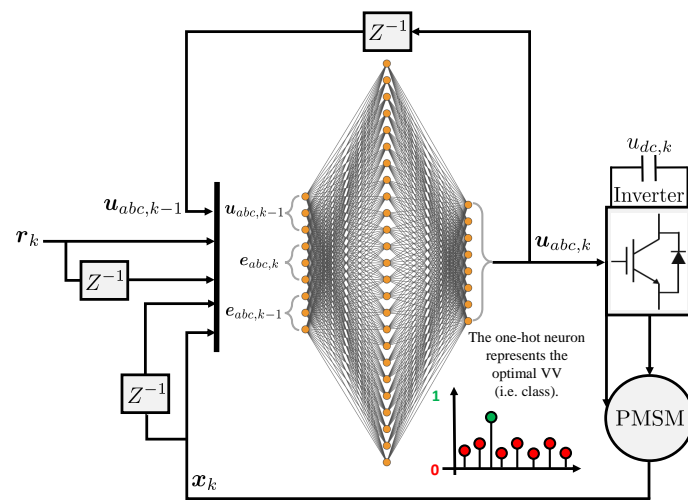
Regarding the input layer, it consists of 9 neurons, 3 for the previous switching state/control input  $\mathbf{u}_{abc,k-1}$ , and the other six neurons are for the error between the measured currents and reference currents at the current and the previous time instants  $k$  and  $k - 1$  in the  $abc$  frame. The error vector at time instant  $k$  is defined as

$$\mathbf{e}_{abc,k} = \begin{pmatrix} i_{a,k} - i_{a,k}^{ref} \\ i_{b,k} - i_{b,k}^{ref} \\ i_{c,k} - i_{c,k}^{ref} \end{pmatrix}, \quad (4.20)$$

and likewise for  $k - 1$ . For the hidden layer, 25 neurons have yielded excellent classification accuracy, and the further increase of the number of neurons in the hidden layer or even having more than one hidden layer did not noticeably yield a better accuracy. The used activation function here is the sigmoid activation function, where

$$y_i = \frac{1}{1 + e^{-a_i}}. \quad (4.21)$$

Once the training process is completed, the weights and biases corresponding to the chosen architecture are optimized, and the network is validated, it can then be deployed on embedded platforms such as dSPACE MicroAutobox as a current controller of the PMSM in closed-loop, with the aim that it captures the performance benefits of long-horizon FCS-MPC. The proposed control scheme based on a FNN that learns the  $N = 5$  FCS-MPC is depicted in Fig. 4.10.



**Figure 4.10:** The proposed long-horizon FCS-MPC control scheme based on NNs.

In order to generate the training data set of the NN, numerical simulations with a rich excitation reference current of the IPMSM with the parameters shown in Table. 4.1 were used, while the currents of the machine were controlled by a long-horizon FCS-MPC controller with  $N = 5$  steps. For the training task of the NN model, the NNs toolbox from MATLAB is used. Generating the training data set as well as training the NN were both done on a supercomputer at IAV GmbH. The simulations included different shaft speeds within the range of interest that the obtained network is to be tested within.

Concerning the constraints fulfilment, input constraints are naturally fulfilled as the output of the NN is one out of the possible VVs of the converter as the case with direct control schemes. However, for the current constraints, if the suggested VV by the NN controller would result in exceeding the maximum allowed current once substituted in the current prediction model, then it gets excluded. The rest of the VVs are evaluated in a descending probability order until one does not violate the current constraints is found and applied.

**Evaluation metrics:** The following metrics are used to evaluate the proposed controller:

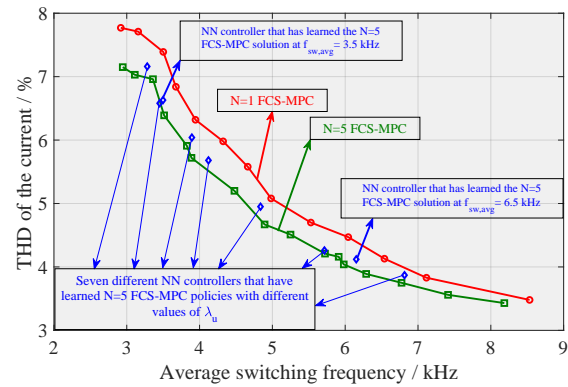
- *Confusion matrix:* is a typical measure of the classification accuracy of a supervised ML algorithm. Each class occupies a box in each axis of the matrix, and the intersection of each class with itself in the other axis gives the correct prediction/classification of that class (i.e. the green diagonal in Fig. 4.11), where the intersection of each class horizontally shows the miss-classification with the other classes. The confusion matrix of an example NN that has learned the solution of long-horizon FCS-MPC with  $N = 5$  at  $f_{sw,avg} = 3.5$  kHz is shown in Fig. 4.11, which yields that the NN has a prediction accuracy of 86.9%. It also shows that, the NN has almost never confused an active VV that has a magnitude phase voltage of  $\frac{2}{3}u_{dc}$  with any class that has an opposite phase polarity on that phase. Taking class number 2 as an example, which has the switching state 100 and has a voltage magnitude on phase  $a$  of  $u_a = \frac{2}{3}u_{dc}$ , it has a zero confusion with all of the classes/VVs which will produce a negative voltage on that phase, which are classes 4, 5, and 6 (referring to switching states of 010, 011, and 001, respectively). This generalizes to all of the other active VVs. Moreover, the two zero vectors 000 and 111 are also almost never confused with each other as can be seen in the top right and bottom left red boxes in the confusion matrix. This helps for a better regularization and mimicking of the average switching frequency of the control policy to be learned.
- *THD Vs.  $f_{sw,avg}$ :* is a typical performance measure in the field of electrical drives. It has been reported in the literature that utilizing long prediction horizon with FCS-MPC schemes yield significantly better regularization of the switching patterns, and hence, yields a better THD content of the currents at a given input energy/switching effort, especially for high-order power electronic systems and for systems that operate at low fundamental frequency [129, 145]. For the case considered in this subsection of the current control task of the IPMSM with the parameters summarized in Table. 4.1, the THD Vs.  $f_{sw,avg}$  curve is plotted in Fig. 4.12 at the rated torque and  $\omega_{me} = 2000$  rpm with a fixed sampling frequency of 50 kHz for three controllers: FCS-MPC with  $N = 1$  (plotted in red), FCS-MPC with  $N = 5$  (plotted in green), and the NN controllers which have learned

the  $N = 5$  long-horizon control policy (plotted in blue). All of these controllers have been tested on the same scenario, and the average switching frequency  $f_{sw,avg}$  is adjusted by the gradual increment of the penalty on the control effort in the FCS-MPC controllers with  $N = 1$  and  $N = 5$ , as well as for the  $N = 5$  controller that was used to generate the training data set for the NN controllers. It can be observed from Fig. 4.12 that the NN controllers have produced a similar THD content and they also have inherently learned the  $f_{sw,avg}$  of the long-horizon FCS-MPC that was used to generate their training data set.

- *Computational demand:* as its reduction while maintaining the performance enhancement of long-horizon FCS-MPC is considered as the sole motivation of this proposal. For comparison, hardware-in-the-loop (HIL) tests were performed. One of the NNs with the architecture of 9–25–8 neurons is deployed on the dSPACE MicroAutoBox II 1513/1514 platform and its needed execution time was recorded, once with exact computation of the sigmoid function and once with a LUT approximation of it as in [154]. Moreover, the needed execution time for the FCS-MPC schemes with prediction horizons of  $N = 1$  up to  $N = 5$  with exhaustive search are recorded and summarized in Table. 4.4. The NN needs only 15.84  $\mu$ s with the exact sigmoid activation function or 14.96  $\mu$ s with the LUT approximation, which makes it implementable in real-time within the sampling frequencies typically used within the field of power electronic systems and electrical drives, in contrary to the  $N > 2$  FCS-MPC schemes with exhaustive search.

1	523982 26.2%	13352 0.7%	5217 0.3%	15923 0.8%	5643 0.3%	14455 0.7%	6813 0.3%	18 0.0%	89.5% 10.5%
2	13648 0.7%	116458 5.8%	3090 0.2%	109 0.0%	0 0.0%	15 0.2%	3532 0.2%	3465 0.2%	83.0% 17.0%
3	3353 0.2%	3952 0.2%	116415 5.8%	3066 0.2%	119 0.0%	0 0.0%	13 0.0%	12660 0.6%	83.4% 16.6%
4	12598 0.6%	21 0.0%	3372 0.2%	113073 5.7%	3407 0.2%	125 0.0%	0 0.0%	2893 0.1%	83.5% 16.5%
5	3460 0.2%	0 0.0%	13 0.0%	3264 0.2%	115582 5.8%	3418 0.2%	96 0.0%	12330 0.6%	83.7% 16.3%
6	13121 0.7%	84 0.0%	0 0.0%	18 0.0%	3394 0.2%	116158 5.8%	3361 0.2%	3394 0.2%	83.2% 16.8%
7	2964 0.1%	3340 0.2%	147 0.0%	0 0.0%	13 0.0%	3503 0.2%	114096 5.7%	12994 0.6%	83.2% 16.8%
8	659 0.0%	5387 0.3%	14109 0.7%	6913 0.3%	14239 0.7%	4725 0.2%	15388 0.8%	523039 26.2%	89.5% 10.5%
	91.3% 8.7%	81.7% 18.3%	81.8% 18.2%	79.4% 20.6%	81.2% 18.8%	81.6% 18.4%	79.6% 20.4%	91.6% 8.4%	86.9% 13.1%
	1	2	3	4	5	6	7	8	
	Target VV								

**Figure 4.11:** The confusion matrix of the proposed NN with the architecture of 9–25–8 neurons that aims to mimic the long-horizon FCS-MPC with  $N = 5$  at  $f_{sw,avg} = 3.5$  kHz.



**Figure 4.12:** The THD Vs.  $f_{sw,avg}$  curve of  $N = 1$  and  $N = 5$  FCS-MPC schemes, and NN controllers that have learned the long-horizon FCS-MPC with  $N = 5$  at different  $f_{sw,avg}$ .

**Table 4.4:** Comparison of the computational demand of FCS-MPC controllers with prediction horizons from  $N = 1$  to  $N = 5$  steps with the proposed NN control scheme with an architecture of  $9 - 25 - 8$  using a sigmoid activation function on the hidden layer and a softmax activation function on the output layer, tested on dSpace MicroAutoBox II 1513/1514 platform.

Controller	FCS-MPC					NN schemes	
	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	sigmoid	sigmoid LUT
Execution time	5.04 $\mu$ s	13.16 $\mu$ s	74.76 $\mu$ s	1.33 ms	11.826 ms	15.84 $\mu$ s	14.96 $\mu$ s

**Simulation results:** In the following, the proposed NN controller is simulatively compared with the long-horizon FCS-MPC with  $N = 5$  controller that was used to generate its training data set. The steady-state performance at the rated torque of  $70 \text{ N} \cdot \text{m}$  and shaft speed of  $\omega = 2000 \text{ rpm}$  is shown in Fig. 4.13(a), where the FCS-MPC with  $N = 5$  controller was used to control the currents of the machine until  $t=0.1 \text{ s}$ , and then the NN controller took over. The electromagnetic torque driving the electrical machine is defined as

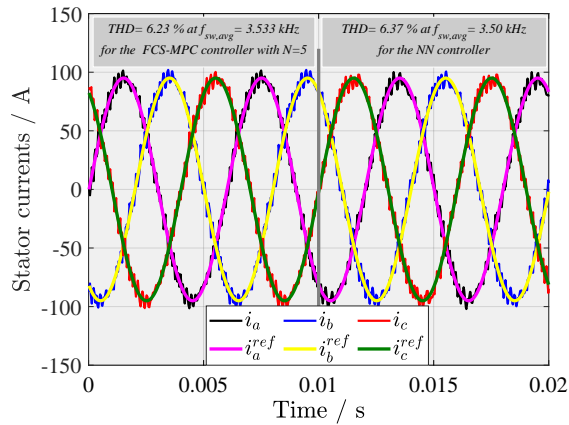
$$T_e = \frac{3}{2} n_p (\psi_p i_q - (L_q - L_d) i_d i_q). \quad (4.22)$$

It can be observed that the NN controller has mimicked almost the same average switching frequency and has produced very close THD value for the stator current of the long-horizon FCS-MPC controller that was used to generate its training data set. The dynamic performance of the obtained NN controller is tested for a step-up change from zero to full load and step-down again to zero at a constant shaft speed regulated via a load machine, and the results are depicted in Fig. 4.13(b) for both controllers. This test shows that the NN controller has also learned and mimicked a very similar dynamic performance to the long-horizon FCS-MPC. Moreover, different current steps were made to test both controllers on generating different reference torques and the results are shown in Fig. 4.13(c) for the NN controller, and in Fig. 4.13(d) for the FCS-MPC controller with  $N = 5$ . These results validate the NN controller in a wide range of operation. In order to test the current constraints fulfilment capability of the proposed NN controller, a step-up of the reference current amplitude from  $70 \text{ A}$  to  $100 \text{ A}$  was made, where  $I_{max}$  was set to  $95 \text{ A}$  and the result is shown Fig. 4.13(e), where it is shown that the current constraints are fulfilled.

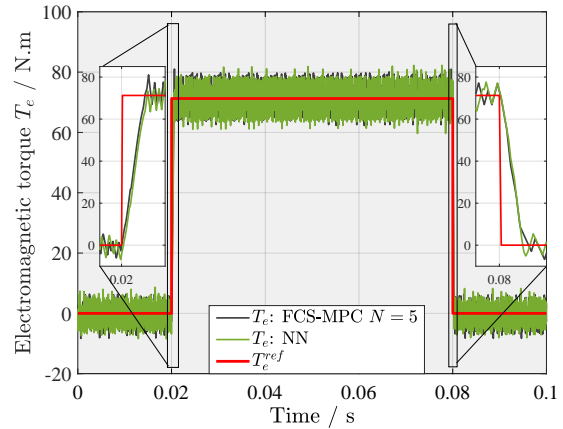
**Extensions and future work:** Following the publication of this proposal in [47], NNs started to get the attention for similar objectives in power electronic and electrical drive applications: in [155], the use of NNs to mimic the long-horizon MPC policies is extended to long-horizon CCS-MPC with experimental verification. In [156], an extension to mimic the direct MPC solution for multilevel converters was presented and validated in a HIL environment.

It will be of interest to extend this proposal not to only mimic long-horizon direct MPC problems but to even mimic optimal pulse pattern (OPP)s. The lack of the ability to tune the average switching frequency during operation is a limiting factor of this approach, and methods to overcome this limitation are of interest for future research.

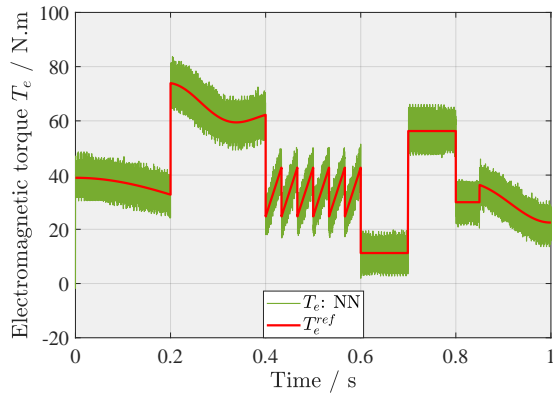




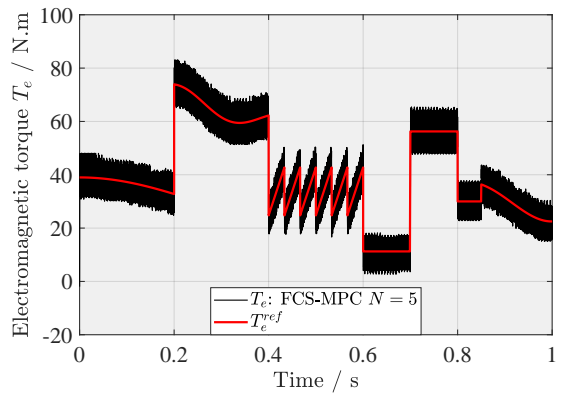
(a) Steady-state currents comparison, where the NN starts to control the currents at 0.1 s.



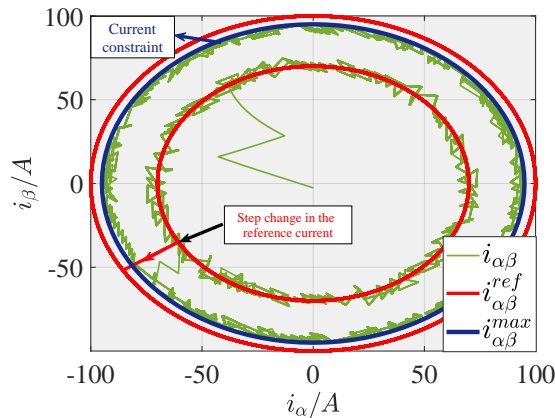
(b) Step-change comparison between zero and rated torque of both controllers.



(c) Resultant  $T_e$  from the NN controller under different given reference torques. .



(d) Resultant  $T_e$  from the FCS-MPC controller with  $N = 5$  for the same test case.



(e) Test scenario to prove the current constraints fulfilment capability of the NN controller: the  $i_q^{ref}$  is stepped up from 70 A to 100 A while  $i_d^{ref}$  is kept at 0 A, and the maximum allowed current  $I_{max}$  is set to 95 A.

**Figure 4.13:** Simulative results comparing the steady-state and the dynamic performance of the proposed NN current controller with the long-horizon FCS-MPC controller with  $N = 5$  that was used to generate its training data set, including steady-state operation, torque step change, torque dynamics, and a test of the current constraints fulfilment.

## 4.2 Indirect MPC

In this section, indirect MPC methods are extensively investigated for the current control task of synchronous machines, as an example. Firstly, the inherent similarities and differences, advantages and disadvantages, as well as costs and benefits of using indirect MPC methods over direct MPC are presented. Following that, a thorough examination of CCS-MPC is conducted. In a first step, the real-time implementation of a CCS-MPC scheme based on a simple LTI model is presented, while observing the suboptimal closed-loop control performance caused by the modelling mismatch. Following that, it is proposed to use the  $\delta u$  formulation in order to obtain an offset-free tracking performance with the linear model. Thereafter, the nonlinear current-dependent physics-based model is used within a constrained nonlinear first-principles continuous control set model predictive control (NL-FP-CCS-MPC) scheme. Finally, and for the sake of completeness and comparison, the analytical variant of indirect MPC using the deadbeat principle is presented with both the linear and nonlinear models. These results serve as base for the comparison with the proposal of using the nonlinear data-driven LSTM neural model to design a real-time data-driven CCS-MPC as presented in the next chapter. In Appendix C, the state-of-the-art PI-FOC is tested on similar test scenarios for benchmarking.

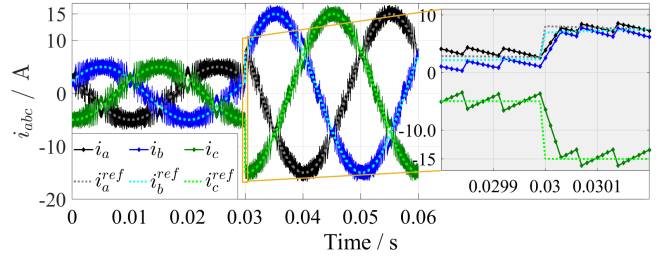
### 4.2.1 Indirect MPC: conceptual differences to direct MPC

Indirect MPC is based on using a modulation scheme to realize a continuous-time optimal VV as an average within the sampling period by the interplay of the switches of the inverter. The optimal VV is typically computed either numerically by minimizing a cost function over a pre-defined prediction horizon in the CCS-MPC variant, or analytically by utilizing a deadbeat function over the upcoming sampling period in DB-MPC. Specifically in the case of CCS-MPC, the key characteristics of predictive control are kept, such as the incorporation of nonlinear models, the fast dynamic response in transients, the states and inputs constraints fulfilment, and above all, the presence of the cost function that facilitates multi-objective optimization. This is achieved while obtaining a favorable steady-state performance. The inherent conceptual differences between direct (FCS) and indirect (CCS) MPC schemes can be summarized in the following:

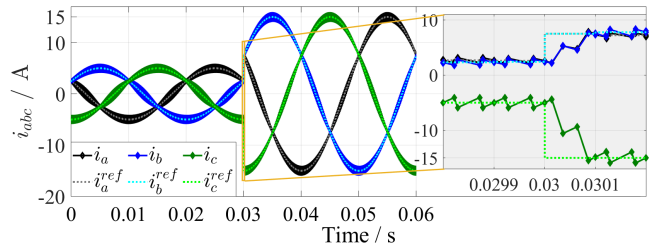
- 1) The computational complexity scales linearly with the prediction horizon for CCS-MPC, where it scales exponentially for FCS-MPC.
- 2) When the sampling frequency of both schemes is kept the same, CCS-MPC yields significantly favorable steady-state performance in terms of less ripples on the tracked states in comparison with FCS-MPC, albeit that this comes at the expense of having more switching losses due to the use of a modulator [157].
- 3) Even though that when the system is reachable in transients, indirect MPC schemes can reach the new reference state at the end of only one sampling step  $T_s$ , direct MPC schemes can provide faster dynamic response, however, when sampled at significantly higher sampling frequencies. This is due to the fact that the nature of the direct MPC problem as a mixed-integer problem which is typically solved by enumeration approaches makes it possible to achieve

extremely fast sampling frequencies for short prediction horizons in contrary to CCS-MPC, especially when using parallel computations capabilities by using FPGAs. For a comprehensive understanding of this fact, the resolution of current measurements need to be high enough to detect the complete state's evolution within one sampling period corresponding to each switching transition. This can be easily done in simulation, however, in practice, the current is mostly measured only once at the middle of the sampling period, as the case in all of the experimental results throughout this dissertation.

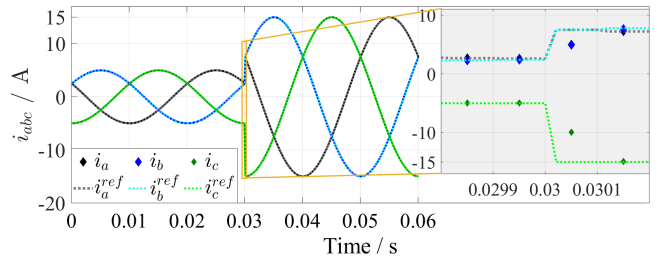
In order to shed light on the second and third points, a CCS-MPC and an FCS-MPC current controllers are simulated both with a prediction horizon of  $N = 1$  and with no penalty on the switching effort for the current tracking of an IPMSM. The FCS-MPC controller is sampled with  $f_s = 100$  kHz, where the CCS-MPC controller is sampled at  $f_s = 10$  kHz. The results are shown in Fig. 4.14(a) for the FCS-MPC controller where the current is measured once at the beginning of each sampling period, in Fig. 4.14(b) for the CCS-MPC controller with eight current measurements within each sampling period at each switching state transition for the used SVM scheme, and in Fig. 4.14(c) for the CCS-MPC with only one current measurement at the middle of each sampling interval, which is the most practical choice. Here it can be observed that in spite of being sampled at ten times higher frequency, the FCS-MPC scheme has yielded significantly higher current ripples. Regarding the dynamical response, it can be seen that the new reference current could be reached within only one sampling interval  $T_s = 100$   $\mu$ s using the CCS-MPC scheme, whereas it has been reached in only 40  $\mu$ s using the FCS-MPC scheme, despite in four sampling periods.



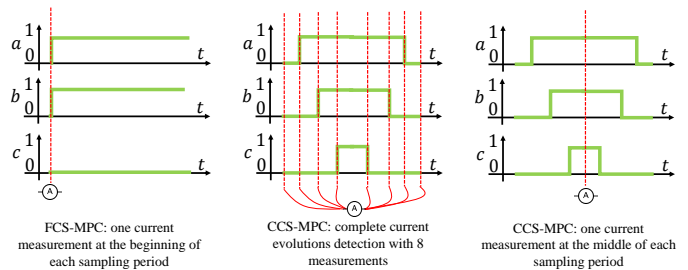
(a) FCS-MPC with  $f_s = 100$  kHz, and the current is measured once at the beginning of each sampling interval.



(b) CCS-MPC with  $f_s = 10$  kHz, and the current is measured 8 times at each transition within a PWM cycle within each sampling period.



(c) CCS-MPC with  $f_s = 10$  kHz, and the current is measured only once at the middle of each sampling period.



(d) A sketch to illustrate where the current is being measured within the sampling period and the symmetrical SVM.

**Figure 4.14:** An example to illustrate the conceptual difference in transients and steady-state ripple between FCS-MPC and CCS-MPC applied for the current control of a synchronous machine [57].

Having the fastest possible response in transients with respect to the available energy and physical limits of the system under control simultaneously while having an optimal steady-state performance in terms of minimal ripples and THD content is an ultimate objective for the classical control of electrical drives. Methods along these two lines were proposed in the literature, such as model predictive pulse pattern control (MP<sup>3</sup>C) [15, 158], which is characterized by having fast dynamic response as in deadbeat controllers while offering an optimal steady-state performance that yields from pre-computed OPPs [159]. However, the fact that the pulse patterns are pre-computed offline leads to a loss of accountability regarding unseen disturbances, faults, and model mismatches. Alternatively, the VSP-MPC [127, 160, 161] goes into the same objective by applying the optimal discrete VV at a variable switching point in time, and not at equidistant intervals as the case conventionally in direct MPC. By doing this, the steady-state ripples are reduced while maintaining the fast transient response of direct control methods. Nevertheless, despite that the ripples are reduced, they are still not minimal, and hence, the THD content in steady-state can be still theoretically further improved.

To this end, indirect MPC (CCS-MPC in particular) is an attractive advanced control strategy that comprises several advantages, and will be further investigated in the following sections. The investigation in the following will focus on the effect of the presence of an accurate mathematical prediction model on the overall closed-loop control performance, the problem formulation as well as the real-time implementation by an efficient numerical solver.

### 4.2.2 First-principles CCS-MPC

In the following, the general problem formulation of the CCS-MPC scheme while incorporating the first-principles model of the PMSM based on physical laws for the base example within this dissertation of the current control is presented. Next, the proposed numerical solver based on a slack formulation of the primal-dual IP method, which is used and deployed on the dSpace platform for the experimental validation of the linear and nonlinear CCS-MPC schemes within this dissertation is illustrated in details. Afterwards, experimental validation of the following CCS-MPC schemes is presented, which include:

- linear first-principles continuous control set model predictive control (L-FP-CCS-MPC) (i.e. incorporating an LTI model of the machine) applied to a 0.5 kW IPMSM, where the control performance degradation caused by the model mismatch is pointed out.
- Offset-free linear CCS-MPC by the proposal of using the  $\delta u$  formulation, applied to the same IPMSM.
- Linear CCS-MPC applied to a 14.5 kW SMPMSM.
- Nonlinear CCS-MPC that incorporates the current-dependent parameters of the IPMSM, with pointing out the closed-loop performance gains of using the nonlinear model.

The scenarios of the experiments are chosen with the intention to show the effects of modelling mismatches on the control performance under various points of transient as well as steady-state operation.

#### 4.2.2.1 Problem formulation and model-based controller architecture

As the name suggests, the optimal control input in CCS-MPC is not limited anymore to the finite-set of the inverter discrete VVs, but can lie anywhere within the hexagonal voltage constraints in the  $\alpha - \beta$  and  $d - q$  planes. In this work, the control input constraints are approximated by the inner circle inscribed within the hexagon, such that the following holds at each discrete-time instant  $k$ :  $\sqrt{u_{d,k}^2 + u_{q,k}^2} \leq \frac{u_{dc,k}}{\sqrt{3}}$ , where  $\frac{u_{dc,k}}{\sqrt{3}}$  represents the radius of the circle inscribed in the hexagonal constraints. It worth to be mentioned that in case the hexagonal constraints are considered (i.e. box constraints in the  $abc$  reference frame), a higher utilization of the dc-link voltage is achieved, however, at the expense of distorting the phase currents while operating in the over-modulation region. The optimization problem of the CCS-MPC for the current tracking task of synchronous machines is stated as

$$\min_U \quad J(\mathbf{U}) \quad (4.23a)$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{D}_k \mathbf{v}_k, \quad (4.23b)$$

$$\mathbf{y}_k = \mathbf{C} \mathbf{x}_k, \quad (4.23c)$$

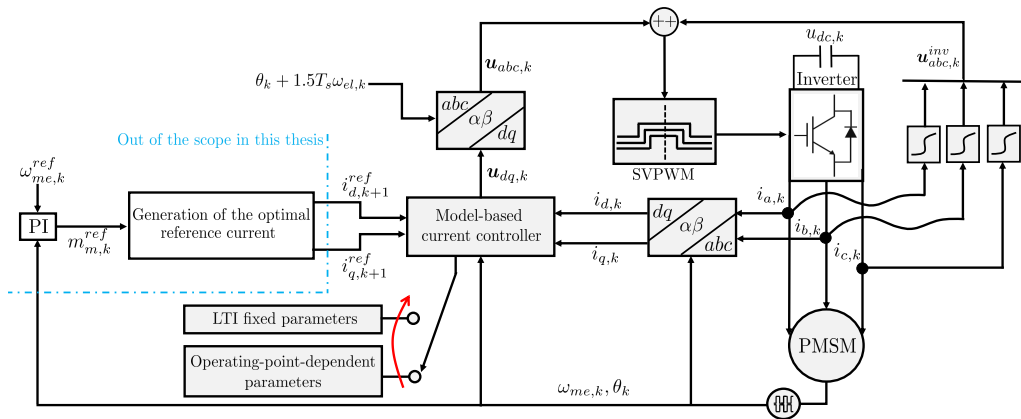
$$\mathbf{x}_{min} \leq \mathbf{x}_{k+i} \leq \mathbf{x}_{max}, \forall i \in \{1, \dots, N\}, \quad (4.23d)$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{k+i} \leq \mathbf{u}_{max}, \forall i \in \{0, \dots, N-1\}, \quad (4.23e)$$

$$\mathbf{A}_k = \begin{bmatrix} 1 - \frac{T_s R_s}{L_{d,k}^{(i_d, i_q)}} & \frac{L_{q,k}^{(i_d, i_q)}}{L_{d,k}^{(i_d, i_q)}} T_s \omega_{el,k} \\ -\frac{L_{d,k}^{(i_d, i_q)}}{L_{q,k}^{(i_d, i_q)}} T_s \omega_{el,k} & 1 - \frac{T_s R_s}{L_{q,k}^{(i_d, i_q)}} \end{bmatrix}, \mathbf{B}_k = \begin{bmatrix} \frac{T_s}{L_{d,k}^{(i_d, i_q)}} & 0 \\ 0 & \frac{T_s}{L_{q,k}^{(i_d, i_q)}} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{D}_k = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{T_s \psi_p^{(i_q)}}{L_{q,k}^{(i_d, i_q)}} \end{bmatrix}, \mathbf{x}_k = \begin{pmatrix} i_{d,k} \\ i_{q,k} \end{pmatrix}, \mathbf{u}_k = \begin{pmatrix} u_{d,k} \\ u_{q,k} \end{pmatrix}, \mathbf{v}_k = \begin{pmatrix} 0 \\ \omega_{el,k} \end{pmatrix}, \quad (4.24)$$

and  $J$  is defined in Eq. 1.4.(a). The model-based predictive current controller of PMSMs which can be used for numerical or analytical, linear or nonlinear indirect MPC is depicted in Fig. 4.15.



**Figure 4.15:** Indirect model-based predictive current controller for synchronous machines.

#### 4.2.2.2 Numerical solver

Finding an approximate solution of the OCP in Eq. 4.23 in real-time is computationally challenging with respect to the available sampling times typically used in electrical drive applications. For this reason, the real-time implementations of CCS-MPC schemes were limited and the FCS-MPC schemes have dominated the vast majority of the MPC publications within the power electronics and electrical drives community. Nonlinear constrained CCS-MPC controller using the Lagrangian method together with a real-time gradient method using the framework *grampc* was applied on an IPMSM drive with a sampling time of  $500\ \mu\text{s}$  in [54], and on an induction machine in [162]. Linear constrained CCS-MPC based on an active-set solver was proposed in [81, 82] with a sampling time of  $300\ \mu\text{s}$ . Recently, the framework *acado* was used for a real-time implementation of a nonlinear CCS-MPC for the current control of synchronous reluctance machines with sampling time of  $250\ \mu\text{s}$  [163].

In this dissertation, a numerical solver based on a slack formulation of the primal-dual IP method [164] is used to simplify and find an approximate solution of the stated optimization problem in Eq. 4.23 in real-time for a prediction horizon of up to  $N = 2$  within  $100\ \mu\text{s}$  and  $125\ \mu\text{s}$  sampling times for the two synchronous machines used in the experimental validation. In the following, the solver is presented in details, and in a general form to make it applicable to other electrical drive applications. The symbols  $\mathbf{O}$  and  $\mathbf{I}$  represent the zero and identity matrices, with their dimensions explicitly defined. For an efficient implementation, the barrier parameter  $\tau$  is fixed as in [165]. Furthermore, the maximum number of line search evaluations as well as the number of the solver iterations are limited. These values are tuned until the expected performance in real-time is achieved within the available sampling time. The solver starts iteratively at the beginning of each discrete sampling period  $T_s$  with an initial guess of the decision variables vector  $\boldsymbol{\eta}_k$  (i.e. primal variables vector) to repeatedly build up the Karush–Kuhn–Tucker (KKT) system of equations with  $\boldsymbol{\mu}_k$  and  $\boldsymbol{v}_k$  as the vectors of the dual variables, and  $\boldsymbol{s}_k$  is the vector of the slack variables. Once the KKT system of equations is solved, a Newton direction is decided and a line-search is performed to assure a reasonable reduction in each iteration. The scaling factor  $\epsilon$  is reduced within the line-search loop until the inequality constraints elements of the dual variables or the slack variables are positive. Once  $\epsilon$  is determined and fixed, a scaled Newton update is done, and the updated decision variables vector  $\boldsymbol{\eta}_k$  is computed and returned as an output of the algorithm.

The algorithm is generically presented in Algorithm 2 in order to make it applicable for any other optimization problems for other power electronic or electrical drive systems, where  $n_{dv}$ ,  $n_{inq}$  and  $n_{eq}$  represent the number of the decision variables, the number of the inequality and equality constraints, respectively. The Lagrangian is denoted as  $\mathcal{L}$ , and is defined as

$$\mathcal{L} = J(\boldsymbol{\eta}) + \boldsymbol{v}^\top \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{\mu}^\top \boldsymbol{e}(\boldsymbol{\eta}), \quad (4.25)$$

where its Hessian is denoted with  $\boldsymbol{H}$ . The equality constraints are denoted by  $\boldsymbol{g}$ , where the inequality constraints are denoted by  $\boldsymbol{e}$ . The  $\circ$  operation denotes element-wise multiplication. For future work, a further speedup of the execution time of the algorithm can be made by solving the KKT system in a more computationally-efficient manner based on the approach and results reported in [166].

**Algorithm 2:** Primal-dual IP solver**Input:**  $\eta_k$ Initialize  $\tau$ ,  $\boldsymbol{\mu}_k = \mathbf{I}(n_{inq}, 1)$ ,  $\boldsymbol{s}_k = \mathbf{I}(n_{inq}, 1)$ ,  $\boldsymbol{v}_k = \mathbf{I}(n_{eq}, 1)$ **for**  $i = 1$  to *MaxIter* **do**> Evaluate  $\mathbf{H}$ ,  $\boldsymbol{g}$ ,  $\boldsymbol{e}$ ,  $\nabla \boldsymbol{g}$ ,  $\nabla \boldsymbol{e}$ ,  $\nabla J$ , and  $\boldsymbol{r}_T$ 

&gt; Build the KKT system of equations

$$\begin{bmatrix} \mathbf{H}(\boldsymbol{\eta}_k, \boldsymbol{\mu}_k, \boldsymbol{v}_k) & \nabla \boldsymbol{g}(\boldsymbol{\eta}_k) & \nabla \boldsymbol{e}(\boldsymbol{\eta}_k) & \mathbf{O}(n_{dv}, n_{inq}) \\ \nabla \boldsymbol{g}(\boldsymbol{\eta}_k)^\top & \mathbf{O}(n_{eq}, n_{eq}) & \mathbf{O}(n_{eq}, n_{inq}) & \mathbf{O}(n_{eq}, n_{inq}) \\ \nabla \boldsymbol{e}(\boldsymbol{\eta}_k)^\top & \mathbf{O}(n_{inq}, n_{eq}) & \mathbf{O}(n_{inq}, n_{inq}) & \mathbf{I}(n_{inq}, n_{inq}) \\ \mathbf{O}(n_{inq}, n_{dv}) & \mathbf{O}(n_{inq}, n_{eq}) & \text{diag}(\boldsymbol{s}) & \text{diag}(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\eta} \\ \Delta \boldsymbol{v} \\ \Delta \boldsymbol{\mu} \\ \Delta \boldsymbol{s} \end{bmatrix} = -\boldsymbol{r}_T,$$

with

$$\boldsymbol{r}_T = \begin{bmatrix} \nabla J(\boldsymbol{\eta}_k) + \nabla \boldsymbol{g}(\boldsymbol{\eta}_k) \boldsymbol{v}_k + \nabla \boldsymbol{e}(\boldsymbol{\eta}_k) \boldsymbol{\mu}_k \\ \boldsymbol{g}(\boldsymbol{\eta}_k) \\ \boldsymbol{e}(\boldsymbol{\eta}_k) + \boldsymbol{s}_k \\ \boldsymbol{\mu}_k \circ \boldsymbol{s}_k - \tau \end{bmatrix}$$

&gt; Solve the KKT system of equations, and obtain a Newton direction

$$[\Delta \boldsymbol{\eta} \ \Delta \boldsymbol{v} \ \Delta \boldsymbol{\mu} \ \Delta \boldsymbol{s}]^\top.$$

> Compute the scaling factor  $\epsilon \in ]0, 1]$  by line search.> Initialize  $\epsilon = 1$ , and its reduction scaler  $0 < \kappa_{ls} < 1$ **for**  $j = 1$  to *MaxIterLineSearch* **do**

&gt; Compute an iterative step:

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_k + \epsilon \Delta \boldsymbol{\mu},$$

$$\boldsymbol{s}_t = \boldsymbol{s}_k + \epsilon \Delta \boldsymbol{s};$$

Check **if**  $\boldsymbol{\mu}_t > 0$  and  $\boldsymbol{s}_t > 0$  **then**| break; (fix  $\epsilon$ )**end**> Decrease  $\epsilon$ :

$$\epsilon = \epsilon \kappa_{ls}$$

**end**

&gt; Compute a Newton step:

$$\boldsymbol{\eta}_k = \boldsymbol{\eta}_k + \epsilon \Delta \boldsymbol{\eta},$$

$$\boldsymbol{v}_k = \boldsymbol{v}_k + \epsilon \Delta \boldsymbol{v},$$

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_k + \epsilon \Delta \boldsymbol{\mu},$$

$$\boldsymbol{s}_k = \boldsymbol{s}_k + \epsilon \Delta \boldsymbol{s};$$

**if**  $\|\boldsymbol{r}_T\|_2 \leq \text{threshold}$  **then**

| break; (an approximate solution is found)

**end****end****Output:**  $\eta_k$

### 4.2.2.3 L-FP-CCS-MPC for an IPMSM

In this subsection, the CCS-MPC scheme is implemented for the current control of the IPMSM shown in Fig. B.1 in Appendix B based on the control scheme shown in Fig. 4.15, using an LTI model with fixed parameters as written in the motor data-sheet, and with a prediction horizon of  $N = 1$  step. In Fig. 4.16(a),  $i_q^{ref}$  is kept constant at 5 A where step changes of  $i_d^{ref}$  from  $-10$  A to 0 A were made at a shaft mechanical speed of  $\omega_{me} = 500$  rpm regulated via the load machine. In Fig. 4.16(c),  $i_d^{ref}$  is kept constant at 0 A, where step changes of  $i_q^{ref}$  between 15 A and 5 A were applied at the same speed. In both of these tests shown in Fig. 4.16(a) and Fig. 4.16(c), the voltage drop on the inverter was not compensated for. The same tests were repeated while including the feedforward inverter nonlinearity compensation and the results are shown in Fig. 4.16(b) and Fig. 4.16(d), respectively. Furthermore, and to show the results on different speeds, the reference currents of the IPMSM under control were kept constant at  $i_d^{ref} = -2$  A and  $i_q^{ref} = 10$  A, where the shaft speed is stepped up and down between 500 rpm and 1500 rpm, and the results are depicted in Fig. 4.16(e)-(f). From these results, the effects of modelling mismatch between the mathematical model incorporated within the MPC and the real plant can be observed. Particularly for the current control task, these effects can be in the form of one or more of the following effects:

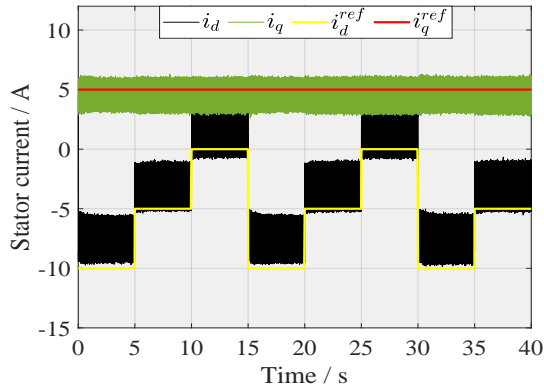
- Higher ripples
- Slower dynamics
- Over/under shooting
- Variant steady-state deviations (offsets)
- In severe cases, it leads to instability

### 4.2.2.4 L-FP-CCS-MPC for a SMPMSM

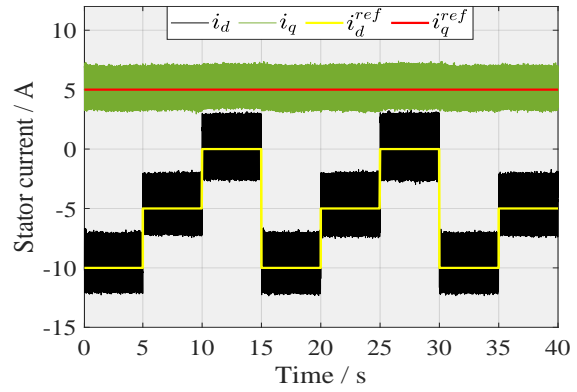
For further validation, the control scheme is implemented on the 14.5 kW SMPMSM shown in Fig. B.2 in Appendix B, incorporating fixed-parameters LTI model, and a prediction horizon of  $N = 2$  steps is used. The execution time  $T_{ex}$  is measured and depicted in Fig. 4.17. The controller is tested with step changes of the torque-producing current  $i_q^{ref}$  from 12 A to 24 A at  $i_d^{ref} = 0$  A and  $\omega_{me} = 120$  rad s<sup>-1</sup>. Firstly, the currents are tracked via the linear CCS-MPC scheme solely, and secondly, while using a discrete-time integral action as an additive control input to compensate for the modelling mismatch. The results are depicted in Fig. 4.18. Slight persistent offsets can be observed in Fig. 4.18(a), where they are mitigated once an additive integrating control input is added to the MPC control input as can be seen in Fig. 4.18(b). The further results are meant to show the dynamic performance as well as the steady-state offset of the linear CCS-MPC controller with and without the additive discrete-time integral action while fulfilling the input constraints. For this purpose, the voltage amplitude is limited to 200 V while the torque-producing current  $i_q^{ref}$  is stepped-up from 0 A to 15 A at  $\omega_{me} = 120$  rad s<sup>-1</sup>. In Fig. 4.18(c), the results are shown for the CCS-MPC scheme without an error integrating functionality with the voltage amplitude and its constraint depicted in Fig. 4.18(d). The same test was repeated while including the discrete-time integral action, and the results are shown in Fig. 4.18(e) and Fig. 4.18(f), respectively. The expense of using a simple fixed-parameters LTI model appears again as can be seen from the overshoot in the  $q$  - axis current tracking as well as the slight offset in the  $d$  - axis current as shown in Fig. 4.18(c). Once adding a



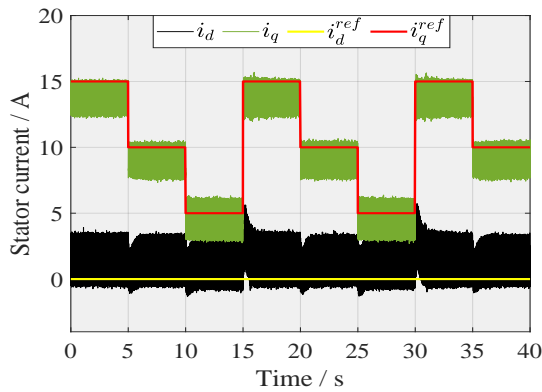
discrete-time integral action beside the MPC control input, an offset-free tracking is obtained as seen in Fig. 4.18(e), however, on the expense of having more fluctuations, and the overshoot in the  $q$  -  $axis$  current persist to occur.



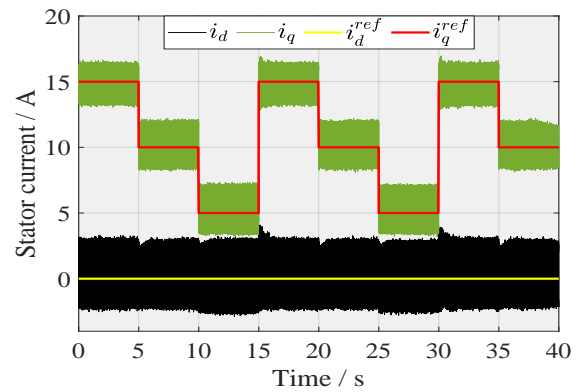
(a) Step changes of  $i_d^{ref}$  at constant  $i_q^{ref}$  and  $\omega_{me}$  of 500 rpm - without inverter nonlinearity compensation.



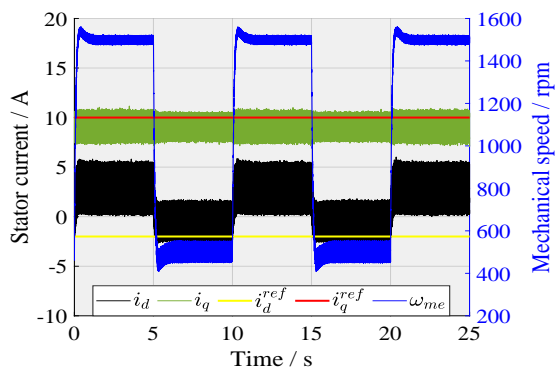
(b) Step changes of  $i_d^{ref}$  at constant  $i_q^{ref}$  and  $\omega_{me}$  of 500 rpm - with inverter nonlinearity compensation.



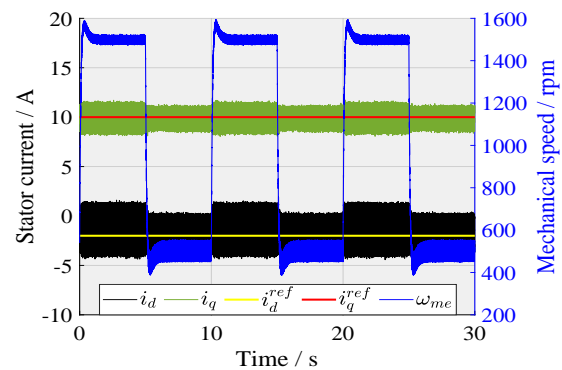
(c) Step changes of  $i_q^{ref}$  at constant  $i_d^{ref}$  and  $\omega_{me}$  of 500 rpm - without inverter nonlinearity compensation.



(d) Step changes of  $i_q^{ref}$  at constant  $i_d^{ref}$  and  $\omega_{me}$  of 500 rpm - with inverter nonlinearity compensation.

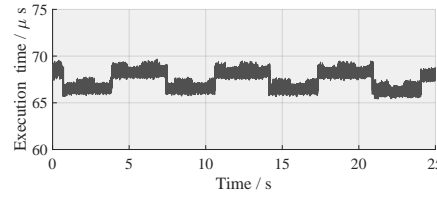


(e) Speed variation test - without inverter nonlinearity compensation.

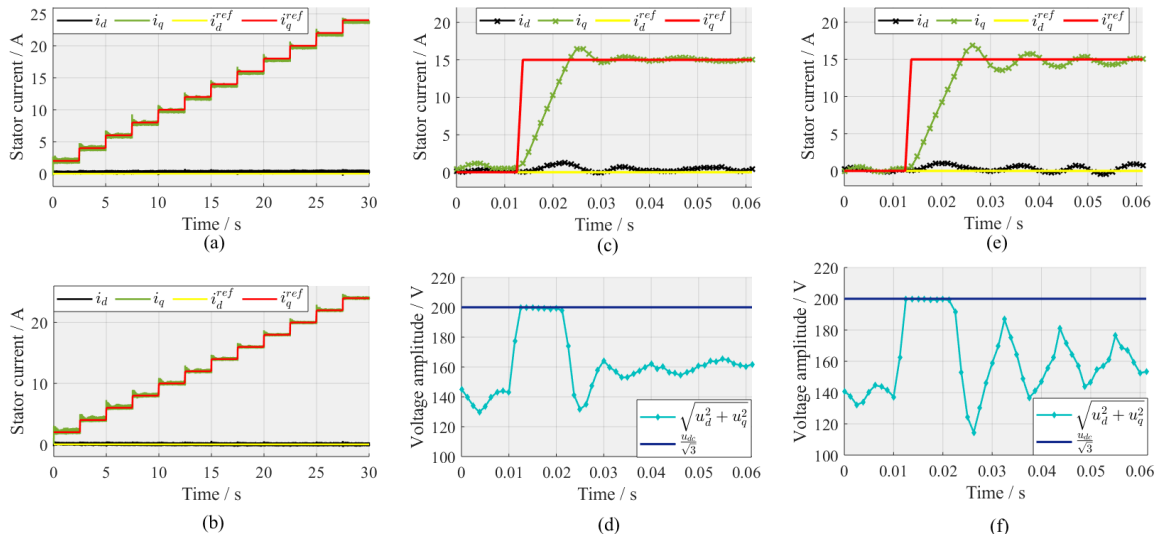


(f) Speed variation test - with inverter nonlinearity compensation.

**Figure 4.16:** Step current changes and speed variation test using CCS-MPC with an LTI model, with and without inverter nonlinearity compensation.



**Figure 4.17:** The measured execution time  $T_{ex}$  of the CCS-MPC scheme with  $N = 2$  measured on the dSPACE DS1007 platform of the SMPMSM test bench.



**Figure 4.18:** Experimental validation of the CCS-MPC scheme applied on the SMPMSM with an LTI first-principles model: (a) tracking results without any integral action functionality, (b) tracking results with an additive discrete-time integral action term, (c) transient of the torque-producing current  $i_q^{ref}$  from 0 A to 15 A at  $\omega_{me} = 120 \text{ rad s}^{-1}$  without integral action with its corresponding voltage constraint fulfilment shown in (d), and in (e) and (f) the same step change was made with the additive discrete-time integral action.

#### 4.2.2.5 Mitigation of modelling mismatch via integral action and parameters estimation: a review

In the literature, different methods have been proposed to enhance the robustness of model-based control schemes, and to mitigate the expense of using simple to obtain, and easy to evaluate LTI models for the control task of electrical machines. In this section, a review of the most used model mismatch compensation techniques with MPC in the field of power electronics and the control of electrical machines is made. These methods can be categorized in the following:

- *Discrete-time integrator:* in which an additive control input  $\mathbf{u}_k^{int}$  is added to the MPC control in order to obtain an offset mitigation performance. The integral control input is computed from a feedback loop with the measured outputs and their set points, such as

$$\mathbf{u}_k^{int} = \sum_{i=0}^k \mathcal{I}(\mathbf{y}_i - \mathbf{r}_i), \quad (4.26)$$

where  $\mathcal{I} = \text{diag}(\mathcal{I}_d \ \mathcal{I}_q)$  is the integral gains diagonal matrix, and for the current control problem of PMSMs  $\mathbf{y} = (i_d \ i_q)^\top$  and  $\mathbf{r} = (i_d^{ref} \ i_q^{ref})^\top$ . Within indirect MPC, this additive term is added to the control inputs yielded from the MPC controller, either by minimizing the corresponding cost functional in CCS-MPC or analytically by using a deadbeat function in DB-MPC [56]. The VV that is applied to the plant in these cases is then the summation of these two vectors, such as

$$\mathbf{u}_k^{i-ccsmc} = \mathbf{u}_k^{ccsmc} + \mathbf{u}_k^{int}, \quad (4.27a)$$

$$\mathbf{u}_k^{i-dbmpc} = \mathbf{u}_k^{dbmpc} + \mathbf{u}_k^{int}, \quad (4.27b)$$

for discrete-time integral CCS-MPC and for discrete-time integral DB-MPC, respectively. The input constraints are maintained in this method by the projection method after adding the integral term. An extension of this method to FCS-MPC was proposed in [138], where after computing the continuous-time VV that includes both the MPC term and the integral term, the optimal VV is transferred into the  $\alpha - \beta$  plane, and the three adjacent discrete inverter VVs are evaluated in a cost function by an enumeration approach. This method and its application with CCS-MPC, DB-MPC, and FCS-MPC is depicted in Fig. 4.19(a).

- *$\delta\mathbf{u}$  formulation:* which is based on using the input increments  $\delta\mathbf{u}$  as decision variables for the optimization problem instead of the absolute control inputs  $\mathbf{u}$ . The actual inputs to the plant are then obtained by integrating the input increments as in Eq. 4.33. This method is in favour to be used for offset compensation in practice in problems where the number of the measured states does not exceed the number of the manipulated control inputs in the vector  $\mathbf{u}$  [167], which is the case for the current control task of PMSM drives. Accordingly, the states and inputs constraints need to be reformulated in terms of their increments. An experimental validation of this method with CCS-MPC was presented in [56]. An incremental model combined with an inductance observer was proposed in [168] using FCS-MPC scheme, and in [169] using a DB-MPC scheme. An illustration of this method is shown in Fig. 4.19(b).
- *State-space model augmentation:* with a disturbance or integral state  $\mathbf{d}$ . The model augmentation can be in the form of discrete-time integrators [170], or the most common choice of a fictitious integrating disturbance model [167, 171–173]. The disturbance can then be estimated with a Kalman filter or a Luenberger observer. This approach has the advantage in the direct constraints fulfilment within the optimizer in contrary to the additive control input in the first approach, where the projection method is needed. Such an inclusion of the accumulative error in the cost function with the one-step FCS-MPC was proposed in [174, 175]. A model augmentation with a disturbance model with a constrained nonlinear CCS-MPC was presented in [163], and with an unconstrained CCS-MPC in [176]. A recent review of disturbance observers for MPC in electrical drive applications is made in [177]. This approach is illustrated in Fig. 4.19(c).
- *Persistence step disturbance compensation:* is the conventional method for MPC schemes to deal with incorrect steady-state gains yielded from the difference between the real model and the one used in the MPC formulation from the feedback loop. It is based on

shifting the reference trajectory by a slight modification of the cost function, such as

$$\mathcal{J}(\mathbf{U}) = \sum_{i=0}^{N-1} \left\| \mathbf{y}_{k+1+i|k} - \underbrace{(\mathbf{e}_{k|k} + \mathbf{r}_{k|k})}_{\mathbf{r}_{k|k}^*} \right\|_Q^2 + \|\Delta \mathbf{u}_{k+i|k}\|_R^2, \quad (4.28)$$

where  $e_{k|k}$  represents the discrepancy between the latest predicted values for the current time step at the previous step  $\tilde{\mathbf{y}}_{k|k-1}$  and the most recent measured output  $\mathbf{y}_{k|k}$ , such as

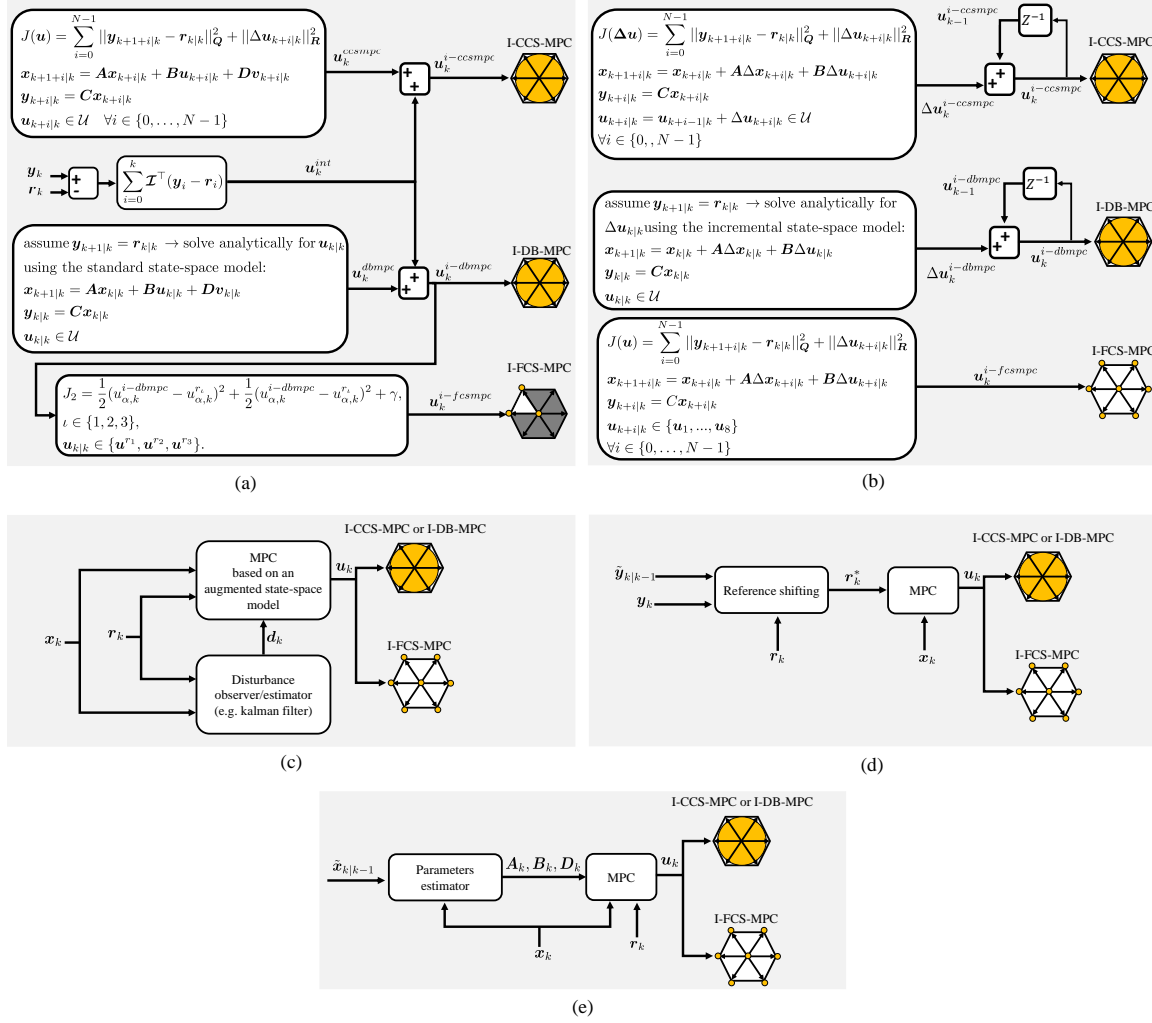
$$\mathbf{e}_{k|k} = \underbrace{\tilde{\mathbf{y}}_{k|k-1}}_{\text{predicted at } k-1} - \underbrace{\mathbf{y}_{k|k}}_{\text{measured at } k}. \quad (4.29)$$

By subtracting this discrepancy from the reference trajectory, an offset-free performance can be obtained for asymptotically stable systems [178]. This approach was applied with a CCS-MPC controller in [81]. This method is depicted in Fig. 4.19(d).

- *Online parameters estimation:* is a common approach to enhance the robustness of model-based controllers by estimating the parameters of the fixed-structure model used in the controller online. For this purpose, different estimation techniques were proposed, such as recursive least squares (RLS), extended and unscented Kalman filters (EKF) and (UKF), respectively, and moving horizon estimation (MHE), to mention a few. A detailed overview of these parameters estimation approaches for PMSMs can be found in [179–181], and the references therein. A general schematic of the parameter estimation approach in conjunction with MPC is shown in Fig. 4.19(e). The speed of convergence as well as the stability of such approaches are sensitive to tuning parameters such as weights and initial values, and thus, they may not yield an optimal performance especially in transients of high dynamic drives. Moreover, it has to be noted that at steady-state operation, the observability matrix of the IPMSM system is ranked deficient when considering any combination of three or all of the four parameters ( $L_d$ ,  $L_q$ ,  $R_s$ , and  $\psi_p$ ) as system states, and thus, these can not be simultaneously estimated. Only one or a combination of two (with some restrictions, for more details the reader is referred to [180]) can theoretically be estimated at the same time. Besides the well-known optimal parameters and state estimation methods, the author has proposed in [46] a novel online model-based inductance estimation approach in which the measured and known states and inputs are used to find an estimate of the parameters (i.e. in that case the inductances  $L_d$  and  $L_q$ ) by minimizing the error between the current measurement and its predicted value at the previous time step consecutively, such that the following quadratic optimization problem

$$\min_{L_d, L_q} \left\| \mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \right\|_2^2 \quad (4.30)$$

is solved for the inductances  $L_d$  and  $L_q$ . The estimated values yielded from solving this optimization problem are the ones which minimize the error between the measured current and its expression in terms of one previous current measurement and the other model parameters, therefore, this approach is highly dependent on the accuracy of the measurements and how accurate the other model parameters are with respect to their true values. The estimated parameters can then be fed back to the MPC algorithm and used for states prediction. This approach can be adjusted to any other combination of two out of the four electrical parameters of the machine model.



**Figure 4.19:** An overview about the different model mismatch compensation approaches typically used within linear MPC frameworks in electrical drives, and their integration with CCS-MPC, DB-MPC, and FCS-MPC: (a) additive integrating control input, (b)  $\delta \mathbf{u}$  formulation, (c) state-space model augmentation, (d) persistent step disturbance compensation, and (e) online parameters estimation.

#### 4.2.2.6 Offset-free linear CCS-MPC

To improve the robustness of the linear CCS-MPC scheme against modelling mismatches and uncertainty, an incremental formulation of the OCP (also referred to as  $\delta \mathbf{u}$  formulation) is used in order to embed error integrating functionality that ensures an offset-free current tracking performance. Revising the discrete-time state-space LTI model (i.e. constant inductances and flux linkage of the permanent magnet) that represents the current dynamics of the machine, and considering that  $\omega_{el}$  to be constant between two consecutive time instants, such as  $\omega_{el,k} \approx \omega_{el,k-1}$  would yield that  $\mathbf{v}_k \approx \mathbf{v}_{k-1}$ , and hence, the incremental states vector is stated as:

$$\Delta \mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k = \mathbf{A}\Delta \mathbf{x}_k + \mathbf{B}\Delta \mathbf{u}_k, \quad (4.31a)$$

$$\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}. \quad (4.31b)$$

To this end, the states prediction model can be stated in the following incremental form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{A}\Delta\mathbf{x}_k + \mathbf{B}\Delta\mathbf{u}_k, \quad (4.32)$$

where this form is devoid from the non-zero constant disturbance  $\mathbf{v}_k$  that appears in the typical state-space representation presented in Eq. 4.23b. Taking the control input increments  $\Delta\mathbf{u}_k$  as the decision variables of the OCP, the control input being applied to the plant is expressed as

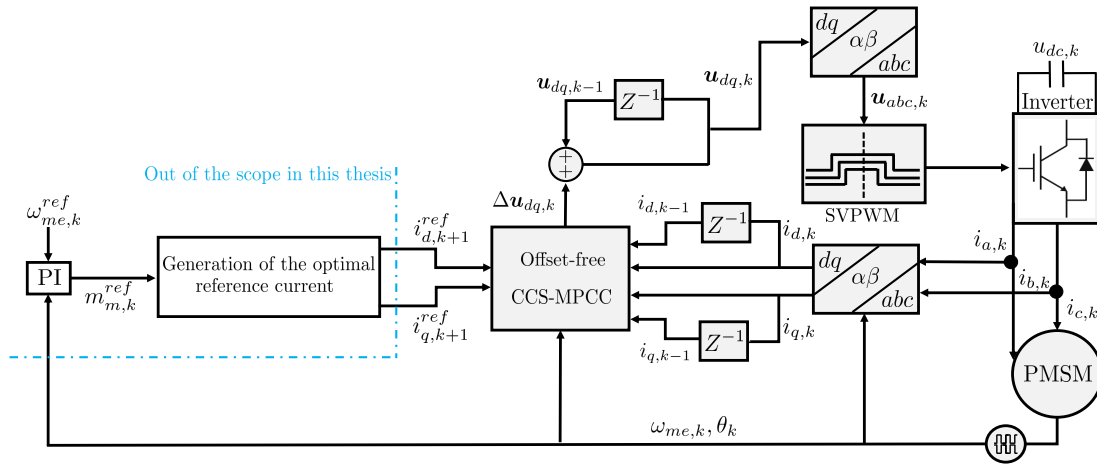
$$\mathbf{u}_k = \Delta\mathbf{u}_k + \mathbf{u}_{k-1}, \quad (4.33)$$

which is the addition of the computed optimal control input increment to the previous optimal control input. In the considered case in the base example, the outputs and states vectors are identical, and hence, a state observer is not needed, and the states can be predicted using the current and previously measured states as well as the previous known/measured control inputs and the model parameters. The use of the incremental state-space model to formulate the OCP, and by considering the input increments as the decision variables benefit the closed-loop control performance by an embedded error integration functionality that guarantees a zero steady-state current tracking error [182]. The optimization problem is the same as stated in Eq. 4.23, with the exception that the time-varying model in Eq. 4.23b is replaced with the incremental LTI states prediction model in Eq. 4.32, and the used input constraints in Eq. 4.23e are to be reformulated in terms of input increments

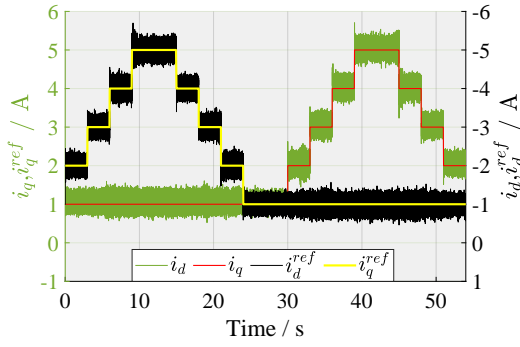
$$\Delta\mathbf{u}_{min} \leq \Delta\mathbf{u}_{k+i} \leq \Delta\mathbf{u}_{max}, \forall i \in \{0, \dots, N-1\}, \quad (4.34)$$

and is solved in real-time for a prediction horizon of  $N = 2$ . The architecture of the proposed offset-free linear CCS-MPC scheme for the current control of PMSMs is depicted in Fig. 4.20.

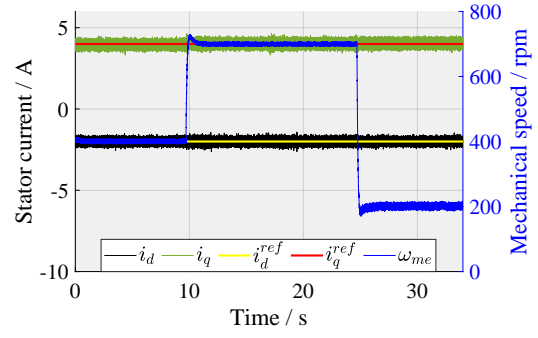
For validation purposes, the controller is tested with fixed parameters with mismatches in the inductances, resistance, and the flux linkage. Moreover, neither a compensation for the inverter nonlinearity nor the angle delay are considered. Different step changes of both axes currents were applied at a rotor mechanical speed of  $\omega_{me} = 500$  rpm and the results are shown in Fig. 4.21. It is clear that this formulation guaranteed an offset-free performance. Furthermore,



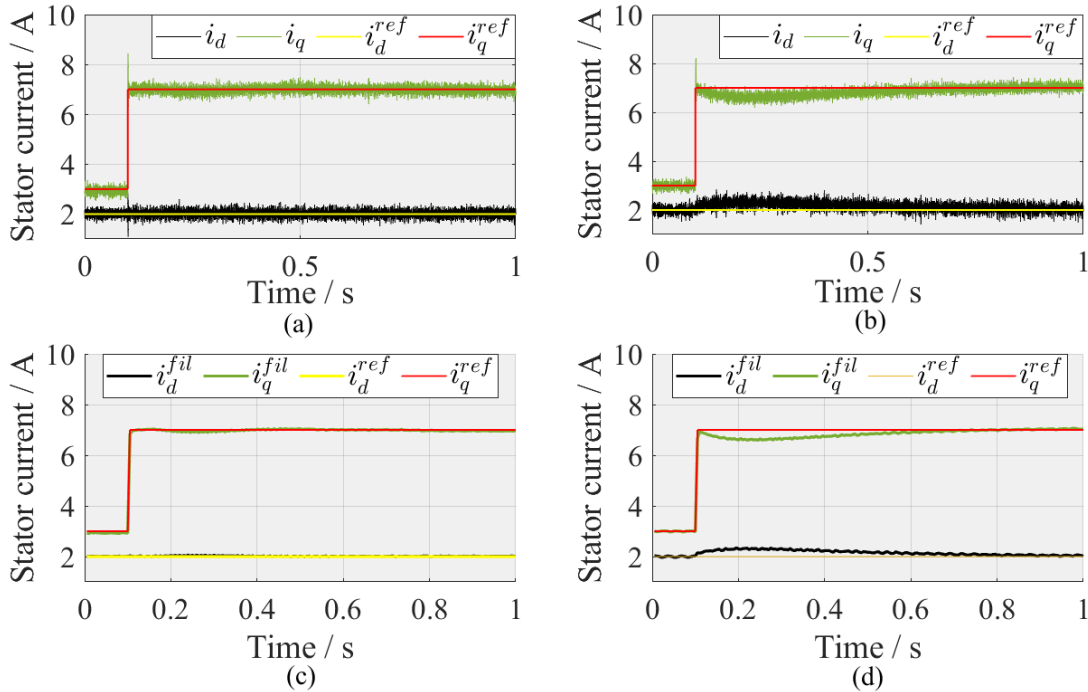
**Figure 4.20:** The proposed offset-free CCS-MPC for current control of synchronous machines.



**Figure 4.21:** Different current tracking steps using the proposed offset-free linear CCS-MPC scheme.



**Figure 4.22:** Steady current tracking with load speed variation using the proposed offset-free linear CCS-MPC scheme.



**Figure 4.23:** Performance comparison under a step change of  $i_q^{ref}$  from 3 A to 7 A at a fixed  $i_d^{ref}$  current of 2 A and  $\omega_{me}$  was kept at 500 rpm: (a) and (c) belong to the proposed offset-free linear CCS-MPC, and the results in (b) and (d) belong to the classical PI-FOC for comparison.

the stator reference currents are kept at  $i_q^{ref} = 4$  A and  $i_d^{ref} = -2$  A, and the shaft mechanical speed is stepped up from 400 rpm to 700 rpm and then stepped down to 200 rpm as shown in Fig. 4.22, which manifest the robustness of the proposed scheme. Lastly,  $\omega_{me}$  was kept at 500 rpm and  $i_d^{ref} = 2$  A, and a step-up change of the torque-producing current  $i_q^{ref}$  from 3 A to 7 A is made. The results are shown in Fig. 4.23(a) for the proposed offset-free linear CCS-MPC scheme, and the currents are filtered by taking the mean value of each 50 measurements and is depicted in Fig. 4.23(c). The same test and measurements with using the classical PI-FOC are shown in Fig. 4.23(b) and (d) for comparison. It is clear that the offset-free MPC controller has a superior decoupling performance, however with a similar transient overshoot.

#### 4.2.2.7 NL-FP-CCS-MPC for an IPMSM

In the previous sections, it is shown that using simple LTI models for the design of MPC controllers yield a sub-optimal closed-loop control performance. Despite that an offset-free performance could be achieved by adding a discrete-time integrator or using the proposed  $\delta u$  formulation, improving the dynamical response and minimizing current ripples would require the use of a more accurate, sophisticated, and probably nonlinear models. To test and show the performance gains of using the current-dependent inductances in Fig. 2.12(c) and (d), as well as the current-dependent permanent magnet flux linkage marked in red in Fig. 2.12(a), they are incorporated in the NL-FP-CCS-MPC scheme. The constrained optimization problem is solved as before using the IP numerical solver presented in Algorithm. 2 for a horizon of  $N = 1$ , and the compensation for the inverter nonlinearity as well as the inevitable angle delay is considered.

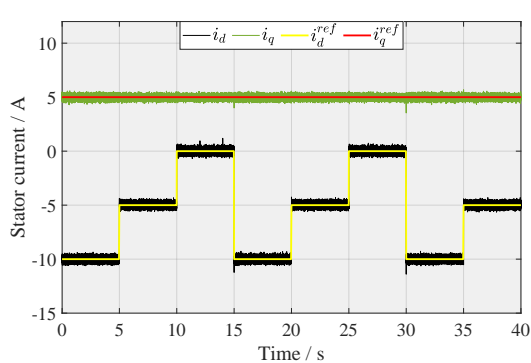
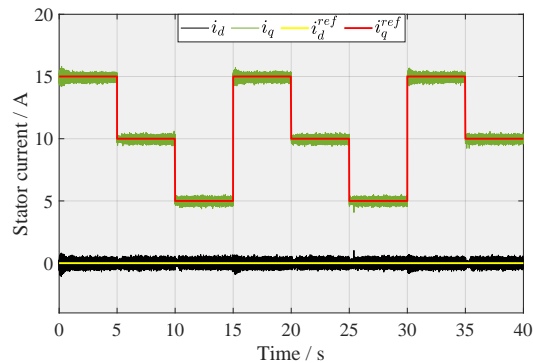
To test the steady-state current tracking performance,  $i_d^{ref}$  steps from  $-10$  A to  $0$  A at a constant  $i_q^{ref}$  of  $5$  A, and then  $i_q^{ref}$  steps from  $15$  A to  $5$  A at a constant  $i_d^{ref}$  of  $0$  A are done at three different speeds of  $250$  rpm,  $500$  rpm, and  $1000$  rpm in Fig. 4.24, Fig. 4.25, and Fig. 4.26, respectively. From these results, the benefit of using the current-dependent nonlinear model appears in terms of an offset-free tracking in both current axes without using any error integrating functionality, as well as in terms of minimal ripples on the currents in comparison with using an LTI model. However, at  $1000$  rpm (which is double the speed at which the model parameters were obtained at), a slight offset in  $i_d$  is observed as seen in Fig. 4.26(a). Therefore, and for a more accurate tracking in the whole speed range of the machine, it is recommended to obtain the parameter maps at different speeds and to interpolate in between. Moreover, having a simple steady-state integral action functionality in the model-based control design is a necessity to assure an offset-free performance over the whole speed range, even while using a nonlinear model.

An expected key performance gain of using the nonlinear model of the machine is to have the fastest possible dynamic performance of the current control with respect to the available control input (i.e. the voltage) without over/undershoots. To investigate this, a step change of the reference torque-producing current  $i_q^{ref}$  from  $1$  A to  $17$  A is made, and the tracking results are shown in Fig. 4.27(a) with the used voltage amplitude and its constraint for this test case depicted in Fig. 4.27(b). The same test is repeated at  $\omega_{me} = 1500$  rpm, and the corresponding results are shown in Fig. 4.27(c)-(d), respectively. Bearing in mind that the current measurements are being obtained at the *middle* of the sampling period as illustrated in Fig. 2.11, and while fulfilling the control input constraints, it is clear that the NL-FP-CCS-MPC scheme is characterized by having fast dynamic response with minimal over/undershoot while fulfilling the input constraints. For the  $d$  - axis current, a step of  $i_d^{ref}$  from  $0$  A to  $-12$  A is made at  $\omega_{me} = 500$  rpm and the results are shown in Fig. 4.27(e), and the same is repeated at  $1500$  rpm in Fig. 4.27(g), with their corresponding voltage amplitudes shown in Fig. 4.27(f) and Fig. 4.27(h), respectively. Again, the same conclusion regarding the fast dynamic performance are drawn.

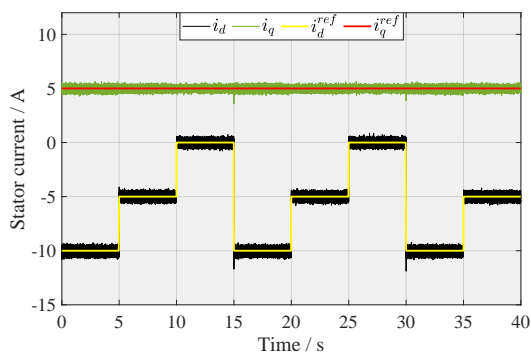
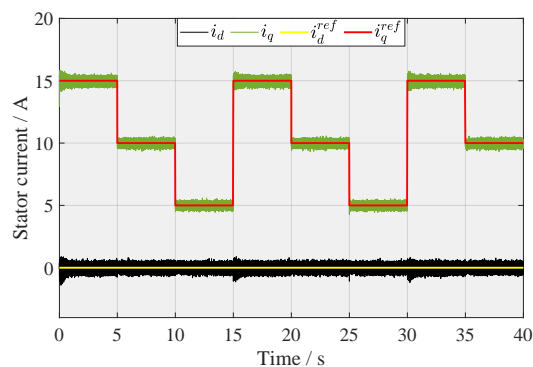
Showing the robustness of the steady current tracking at load speed variation, constant reference currents of  $i_q^{ref} = 10$  A and  $i_d^{ref} = -2$  A are given while the shaft speed is stepped up and down between  $500$  rpm and  $1500$  rpm, and the results are presented in Fig. 4.28. It is noticed that the current tracking is decoupled from the speed variation. The higher current ripple at  $1500$  rpm in comparison with the current ripple at  $500$  rpm is because that the model parameters were obtained at  $500$  rpm. For machines that operate at a way higher speed range, the



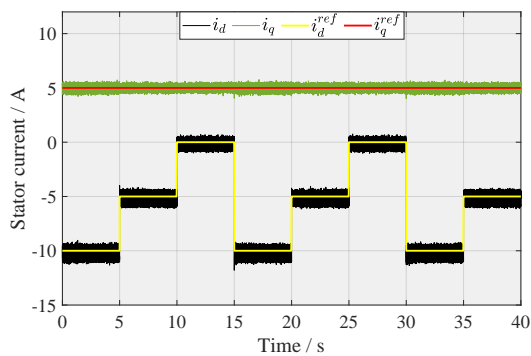
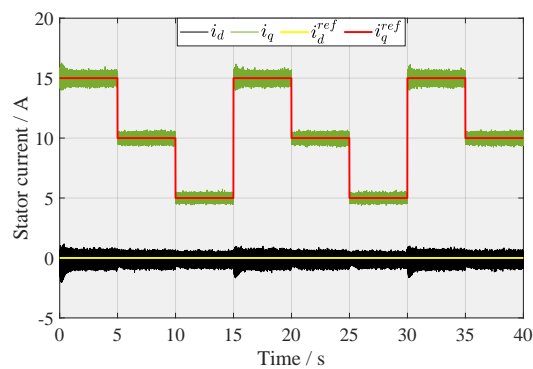
motor parameters maps have to be obtained at different speeds with the same procedure, and it is recommended to consider the operating-dependent parameters to be functions of the speed  $\omega_{me}$  beside the currents  $i_d, i_q$  in a 3D LUT.

(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.

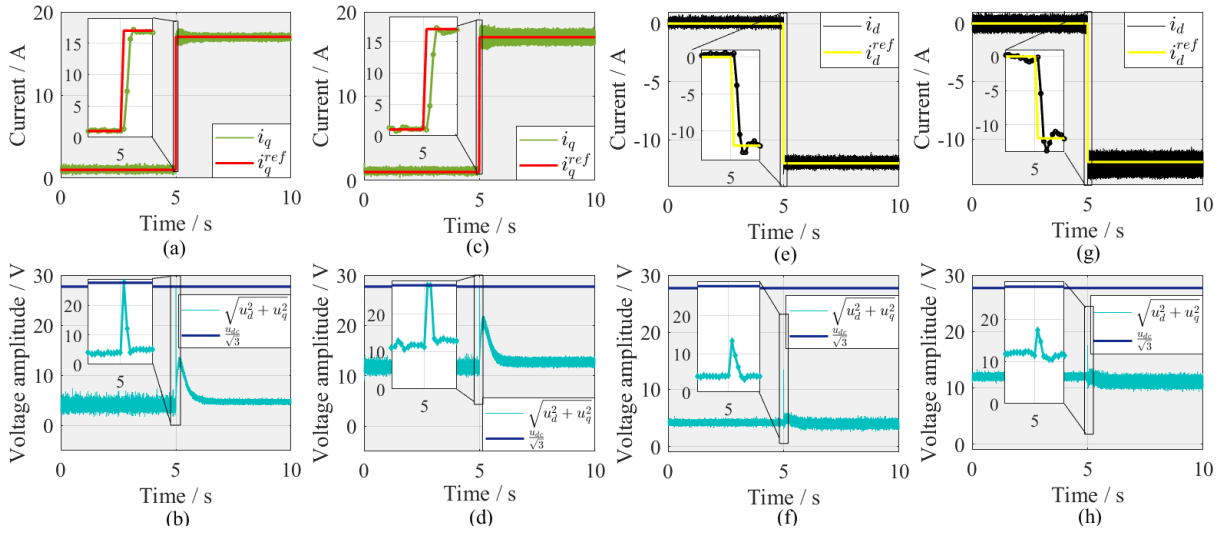
**Figure 4.24:** Experimental current tracking steady-state results of the NL-FP-CCS-MPC at 250 rpm.

(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.

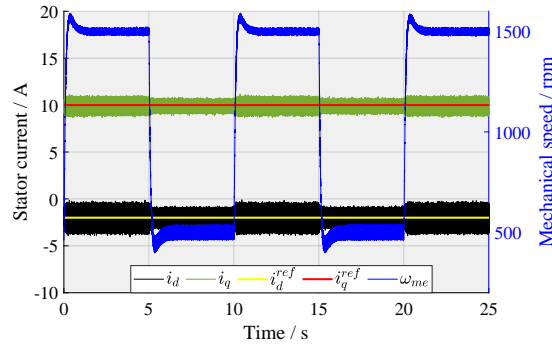
**Figure 4.25:** Experimental current tracking steady-state results of the NL-FP-CCS-MPC at 500 rpm.

(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.

**Figure 4.26:** Experimental current tracking steady-state results of the NL-FP-CCS-MPC at 1000 rpm.



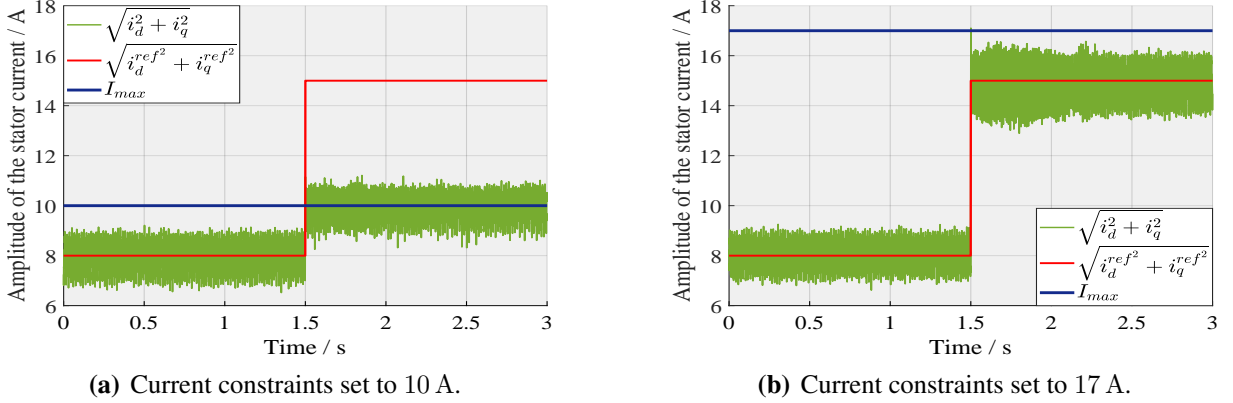
**Figure 4.27:** High dynamic test of the IPMSM using the NL-FP-CCS-MPC scheme.



**Figure 4.28:** Steady current tracking with load speed variation using the NL-FP-CCS-MPC scheme.

Finally, a test scenario is made to validate the current constraints fulfilment. Firstly, the current constraints fed to the numerical solver is set to  $I_{max} = \sqrt{i_d^2 + i_q^2} = 10$  A and the shaft speed is regulated via the load machine to  $\omega_{me} = 1000$  rpm. Initially, the reference current of the main machine is set to  $i_d^{ref} = 0$  A and  $i_q^{ref} = 0$  A. At time instant  $t = 1.5$  s,  $i_q^{ref}$  is stepped up to 15 A. As can be seen in the results depicted in Fig. 4.29(a), the stator current is successfully constrained to the mean value of 10 A. It is observed that the current constraints are held as a mean value, therefore, and for applications where the absolute current value must not be exceeded, it is recommended to subtract half of the expected ripple peak-to-peak value to the current constraint in order to end-up with the intended limit. The same test scenario is repeated with the current constraints being set back to their original value of 17 A, and the results are shown in Fig. 4.29(b). It is here clear that the step up in the stator current is normally achieved.

The fulfilment of the voltage and current constraints is in particular of interest for an optimal operation of the machine with respect to its physical limit in the different operating strategies, such as maximum torque per ampere (MTPA), maximum torque per volt (MTPV), maximum torque per flux (MTPF), and for the FW operation [130].



**Figure 4.29:** Current constraints fulfilment using the NL-FP-CCS-MPC.

### 4.2.3 First-principles DB-MPC

An analytical alternative to solving the OCP numerically in CCS-MPC is the DB-MPC. Revising the current prediction model in Eq. 2.15, and assuming that the ultimate control objective is to be met by the end of the upcoming sampling interval  $k + 1$ , the predicted current values will be equal to their references, such that

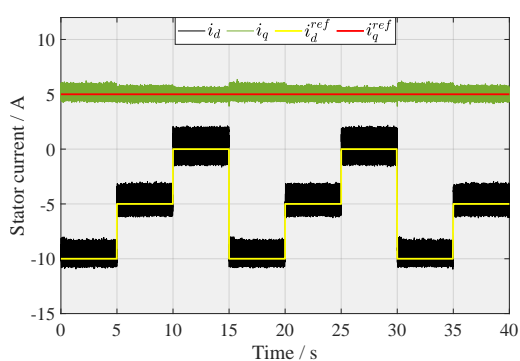
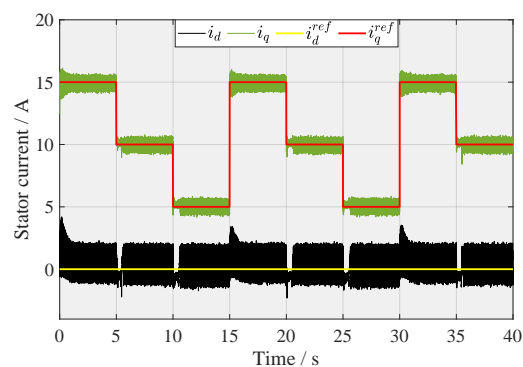
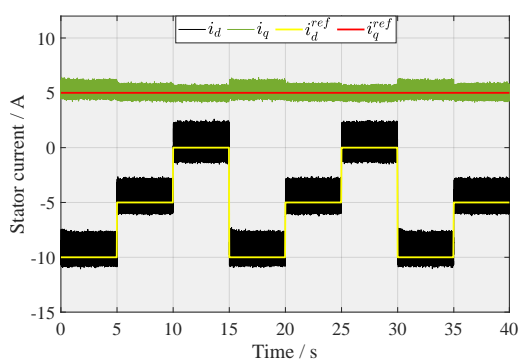
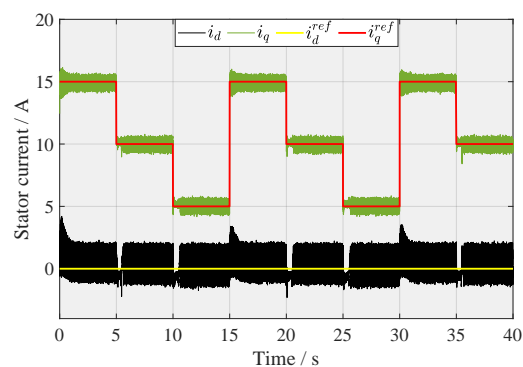
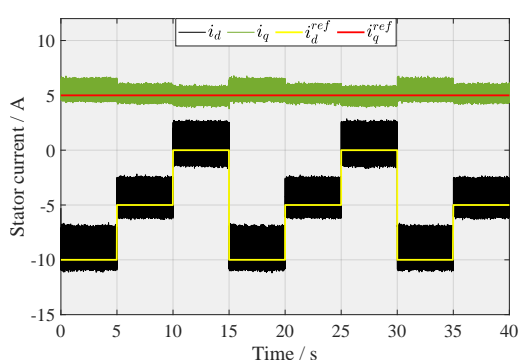
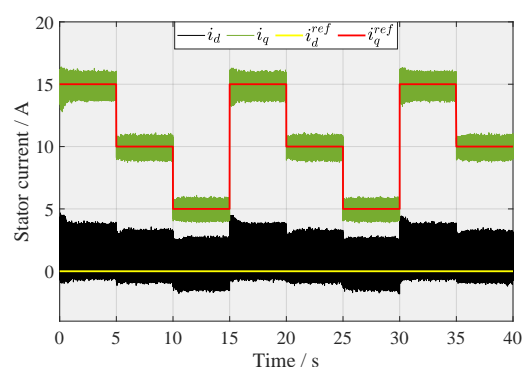
$$\begin{pmatrix} \hat{i}_{d,k+1} \\ \hat{i}_{q,k+1} \end{pmatrix} = \begin{pmatrix} i_{d,k+1}^{ref} \\ i_{q,k+1}^{ref} \end{pmatrix}, \quad (4.35)$$

hence, substituting  $\hat{i}_{dq,k+1}$  with  $i_{dq,k+1}^{ref}$  and solving Eq. 2.15 for  $\mathbf{u}_{dq,k}$  would yield a VV that minimizes the current tracking error. The controller architecture stays within the indirect model-based current control framework depicted in Fig. 4.15. On the one side, deadbeat predictive controllers are characterized by their very fast dynamic response, the relatively low steady-state ripples due to the use of a modulation scheme, and the very low computational demand. On the other side, the limitation to only one-step prediction, the absence of the cost function, and the sub-optimal constraints fulfilment by the projection method<sup>5</sup> are considerable drawbacks in comparison with the numerical indirect CCS-MPC. For validation, and also to show the effect of proper modelling in the closed-loop performance, the DB-MPC controller is tested for the steady-state current tracking test case once with using the LTI model, and again with using the current-dependent nonlinear model.

#### 4.2.3.1 Linear DB-MPC

In Fig. 4.30, Fig. 4.31, and Fig. 4.32, the experimental results of the DB-MPC using a simple LTI model at three different motor speeds of 250 rpm, 500 rpm, and 1000 rpm, respectively, are shown. As expected, and similar to the L-FP-CCS-MPC, the modelling mismatch effects of high current ripples and coupling, slower dynamics, and offsets did appear with different extents depending on the operating point of the machine.

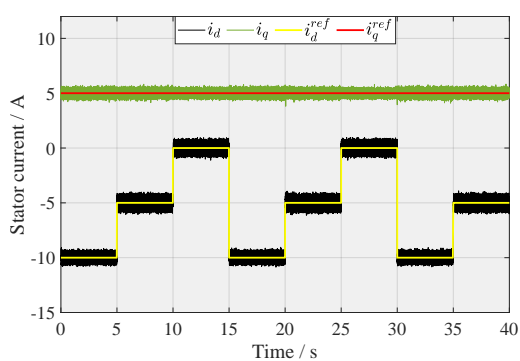
<sup>5</sup>This is important in particular for drives which operate at their maximum available input voltage (e.g. in the FW operation), where the normalized scaling of the deadbeat solution to the voltage constraints circle might not be optimal, as the  $d$  - axis direction needs to be prioritized.

(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure 4.30:** Experimental current tracking steady-state results of the linear DB-MPC at 250 rpm.(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure 4.31:** Experimental current tracking steady-state results of the linear DB-MPC at 500 rpm.(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure 4.32:** Experimental current tracking steady-state results of the linear DB-MPC at 1000 rpm.

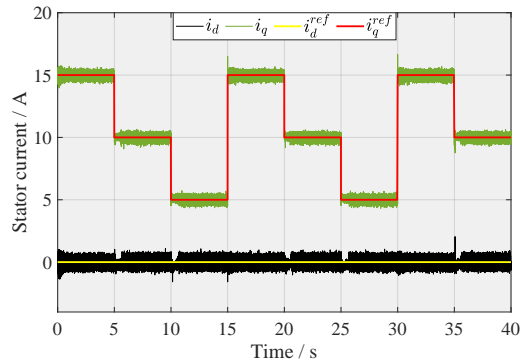
#### 4.2.3.2 Nonlinear DB-MPC

Here, the same test was repeated while using the current-dependent inductances and flux

linkage in the DB-MPC scheme, and the results are shown in Fig. 4.33, Fig. 4.34, and Fig. 4.35. The tracking is almost offset-free in a wide operating range with minimal ripples and coupling effects as the case with the previously presented NL-FP-CCS-MPC.

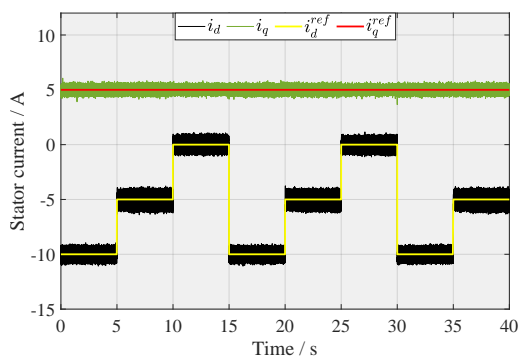


(a) Step changes of the  $d$  – axis current.

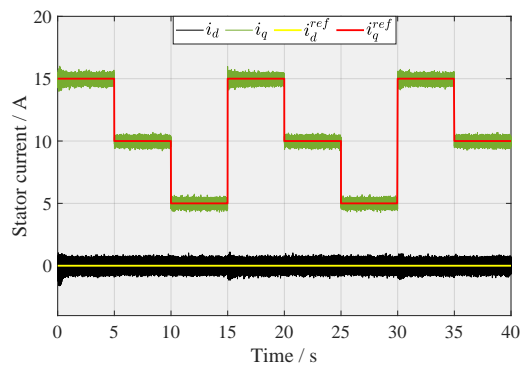


(b) Step changes of the  $q$  – axis current.

**Figure 4.33:** Experimental current tracking steady-state results of the nonlinear DB-MPC at 250 rpm.

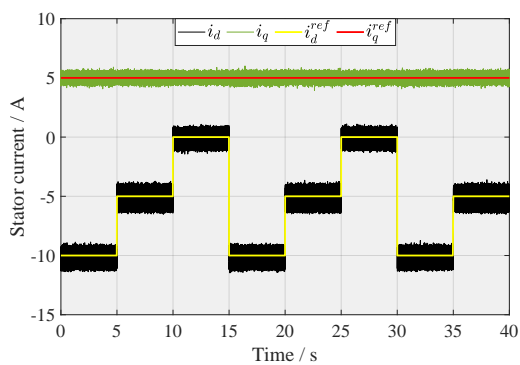


(a) Step changes of the  $d$  – axis current.

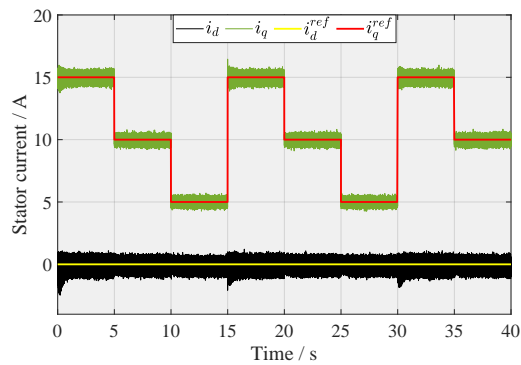


(b) Step changes of the  $q$  – axis current.

**Figure 4.34:** Experimental current tracking steady-state results of the nonlinear DB-MPC at 500 rpm.



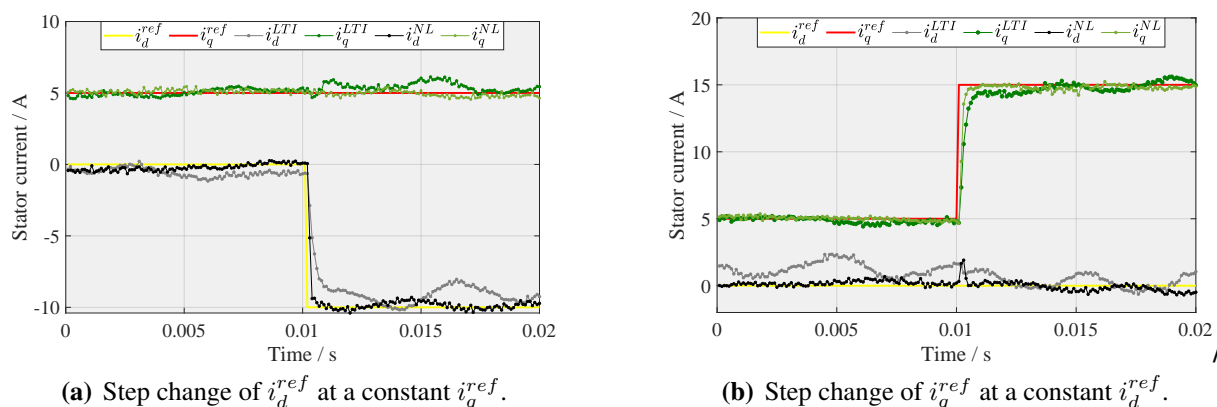
(a) Step changes of the  $d$  – axis current.



(b) Step changes of the  $q$  – axis current.

**Figure 4.35:** Experimental current tracking steady-state results of the nonlinear DB-MPC at 1000 rpm.

Finally, to show the difference between using the LTI model and the nonlinear model in the transient state of the DB-MPC scheme, a step-down of  $i_d^{ref}$  from 0 A to  $-10$  A is made at a constant  $i_q^{ref}$  of 5 A and  $\omega_{me} = 500$  rpm for the controllers with both models as shown in Fig. 4.36(a). Moreover, a step-up of  $i_q^{ref}$  from 5 A to 15 A at a constant  $i_d^{ref}$  of 0 A is made at the same speed, and the results are shown in Fig. 4.36(b). As expected, using the LTI model has yielded a significantly slower dynamics than the nonlinear model in both axes due to the modelling mismatch.



**Figure 4.36:** Experimental comparison in transient states while using the LTI and the nonlinear models within DB-MPC scheme.

---

## CHAPTER 5

---

### Learning-based model predictive control

---

*“Learn continually. There’s always “one more thing” to learn.”*

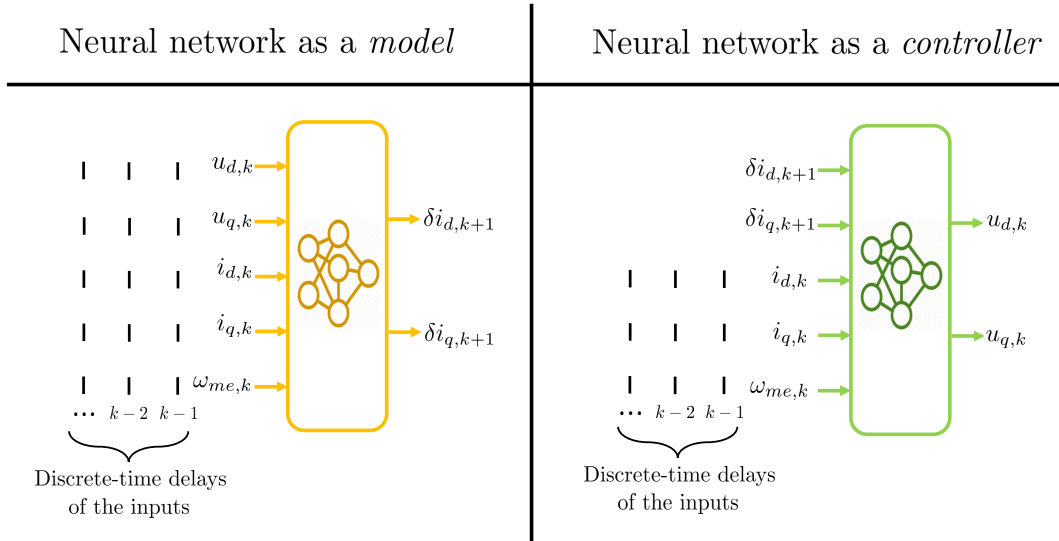
Steve Jobs

This chapter is intended to investigate how the data-driven models of the current dynamics of the PMSM can be used for the design of a closed-loop learning-based continuous control set model predictive control (LB-CCS-MPC) scheme for the current control task, with the focus on using the obtained data-driven LSTM model in Section 3.5 as an example, but in general other data-driven models can be used similarly. Moreover, it deals with the practical challenges of the real-time implementation of such an LB-CCS-MPC scheme within the submillisecond sampling times typically used in electrical drive applications.

#### 5.1 LB-CCS-MPC using the LSTM model

##### 5.1.1 Controller design and implementation

The data-driven model based on the LSTM NN presented in Section 3.5 shows an excellent prediction accuracy in the testing data set, which motivates its use within a model-based control scheme. However, despite that model architecture is chosen as compact as possible to allow its real-time implementation, one evaluation of the model requires  $25\ \mu\text{s}$ . This hinders the use of this neural model in an iterative numerical optimization in the available  $100\ \mu\text{s}$  sampling time for the current control loop of the IPMSM. Moreover, it is not possible to invert the neural model in order to obtain an analytical solution for the optimal control input for a desired change of the states. The NN that acts as a *model* of the current dynamics of the synchronous machine receives the currents, voltages, and motor speed at the current and finite previous discrete time instants as inputs, and provides the change of the currents at the upcoming discrete time instant  $k + 1$  as an output. On the flip side, a NN that acts as a current *controller* has to provide the



**Figure 5.1:** Sketch illustrating the use of NNs either as a model of the current dynamics of synchronous machines (on the left), or as a current controller (on the right) with the corresponding inputs and outputs.

optimal voltages  $u_{d,k}$  and  $u_{q,k}$  as an output while receiving the desired change of the currents in the upcoming sampling interval  $\delta i_{d,k+1}$  and  $\delta i_{q,k+1}$  besides the currents and the electrical speed as inputs. An illustration of these two different uses is sketched in Fig. 5.1.

To this end, it is here proposed to solve the MPC problem *offline* in an explicit-like manner while incorporating the LSTM neural model to generate a data set that represents the mapping between the initial states and optimal control inputs for a desired evolution of the states. For this purpose, the excitation data set presented in Section 3.4 is used here again as different initial conditions for the constrained LB-CCS-MPC problem to be solved for limited randomly chosen desired  $\delta i_{dq,k+1}$  while using the LSTM neural model as a prediction model. For most of the low-order power electronics and electrical machines applications, a one-step prediction is sufficient. Therefore, the following OCP is formulated

$$\min_{u_{d,k}, u_{q,k}} J_3 = \frac{1}{2}(\delta i_{d,k+1} + (i_{d,k} - i_{d,k}^{ref}))^2 + \frac{1}{2}(\delta i_{q,k+1} + (i_{q,k} - i_{q,k}^{ref}))^2 \quad (5.1a)$$

$$\text{s.t.} \quad \delta \mathbf{i}_{dq,k+1} = \mathbf{f}(\mathbf{u}_{dq,k}, \mathbf{u}_{dq,k-1}, \mathbf{i}_{dq,k}, \mathbf{i}_{dq,k-1}, \omega_{me,k}, \omega_{me,k-1}), \quad (5.1b)$$

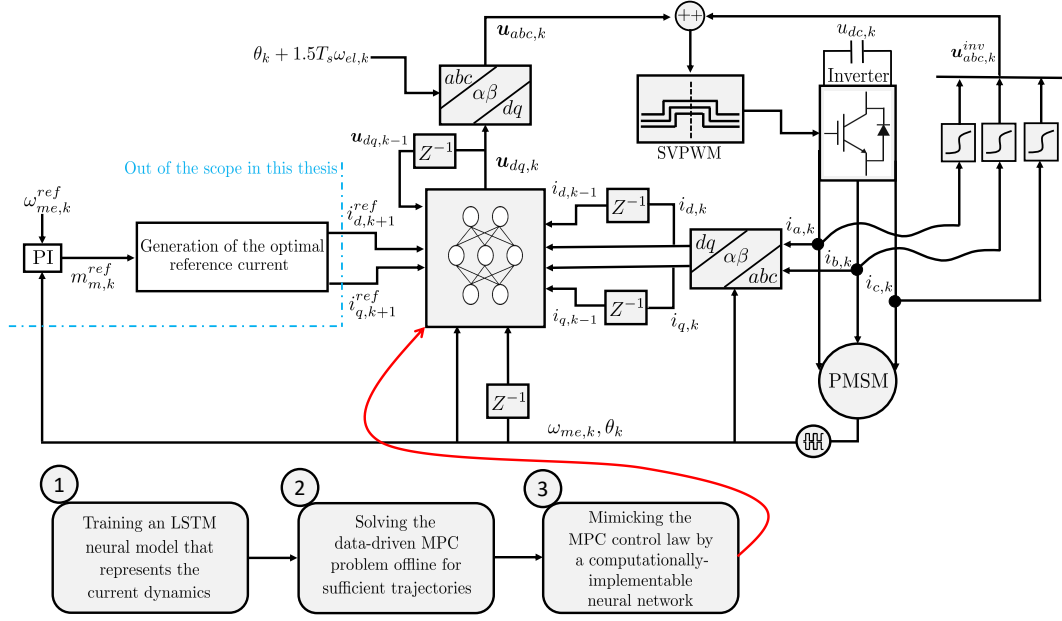
$$\mathbf{x}_{min} \leq \mathbf{x}_{k+1} \leq \mathbf{x}_{max}, \quad (5.1c)$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max}, \quad (5.1d)$$

where  $\delta i_{d,k+1}$  and  $\delta i_{q,k+1}$  are provided by the LSTM NN. The data set contains 12.6 million data points representing different initial conditions and different desired evolution. To make the implementation computationally efficient, the symbolic expression of the NN output is provided via a self-written  $C$  function implemented using CasADi [183]. The optimization problem is then solved in parallel on a CPU with 28 cores<sup>1</sup> using the open source software package IPOPT (Interior Point Optimizer) [184]. With this setup, solving the MPC with the LSTM neural model

<sup>1</sup>Provided by IAV GmbH





**Figure 5.2:** Indirect data-driven predictive current controller for synchronous machines.

for the 12.6 million different trajectories takes  $\approx 6$  hours to compute the optimal solutions. By this, a data set which represents the MPC control policy is generated. After that, this mapping is to be *learned* via a computationally implementable NN that approximate this control policy in a real-time implementable manner. For this purpose, an LSTM NN with three hidden layers is trained, with the following inputs and outputs:

Inputs:  $u_{d,k-1}, u_{q,k-1}, i_{d,k-1}, i_{q,k-1}, i_{d,k}, i_{q,k}, \omega_{me,k}, \omega_{me,k-1}, \delta i_{d,k+1}, \delta i_{q,k+1}$ .

Outputs:  $u_{d,k}, u_{q,k}$ .

Once the NN is trained, fine-tuned, and it exhibits excellent approximation accuracy, all the weights and biases are exported from python to MATLAB, and a MATLAB function that provides the NN output is written and deployed on the dSPACE platform as a current controller of the IPMSM. The desired changes of the currents at the upcoming sampling instant  $k + 1$  which are fed to the NN controller are defined in a deadbeat-like way, such as:

$$\delta i_{d,k+1} = i_{d,k}^{ref} - i_{d,k}, \quad (5.2a)$$

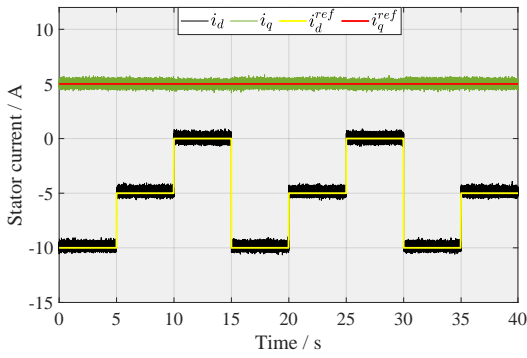
$$\delta i_{q,k+1} = i_{q,k}^{ref} - i_{q,k}. \quad (5.2b)$$

In this implementation, if the VV that the NN provides has an amplitude which lies outside of the input circular constraints, it is projected back to within the circular constraints using the projection method. The proposed LB-CCS-MPC scheme for synchronous machines with the three design steps is depicted in Fig. 5.2.

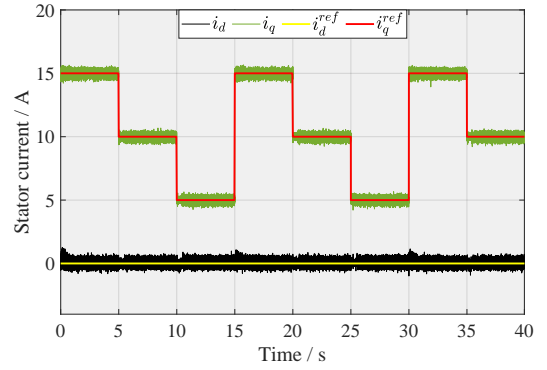
### 5.1.2 Experimental results

The proposed LB-CCS-MPC scheme is deployed on the dSpace MicroAutoBox II to be tested as a current controller of the IPMSM machine shown in Fig. B.1. In the following, the proposed method is to be experimentally validated, and its steady-state and dynamic performance as well as its robustness against speed variation are to be highlighted. The steady-state current tracking scenarios are tested at the same three different speeds of 250, 500, and 1000 rpm, and

the results are depicted in Fig. 5.3, Fig. 5.4, and Fig. 5.5, respectively. The proposed control scheme exhibits excellent steady-state performance in terms of an offset-free tracking without including an error-integrating functionality, minimal current ripples and a good decoupling.

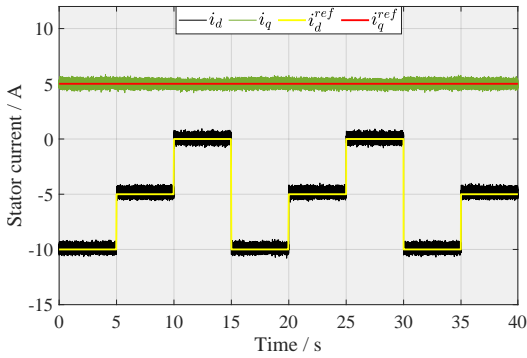


(a) Step changes of the  $d$  – axis current.

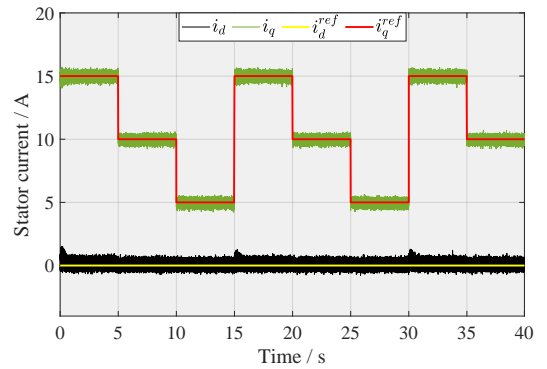


(b) Step changes of the  $q$  – axis current.

**Figure 5.3:** Experimental steady-state tracking results of the nonlinear LB-CCS-MPC at 250 rpm.

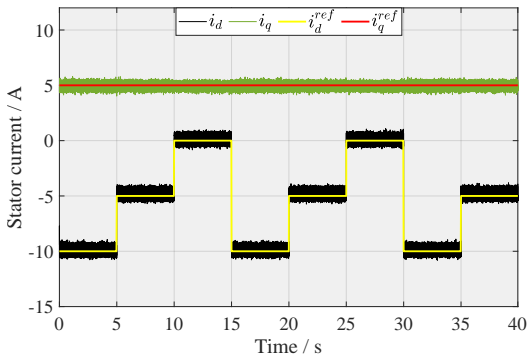


(a) Step changes of the  $d$  – axis current.

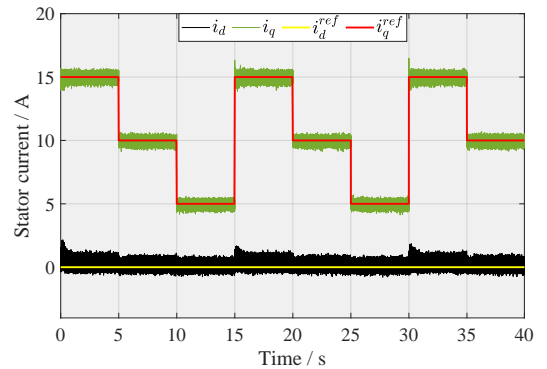


(b) Step changes of the  $q$  – axis current.

**Figure 5.4:** Experimental steady-state tracking results of the nonlinear LB-CCS-MPC at 500 rpm.



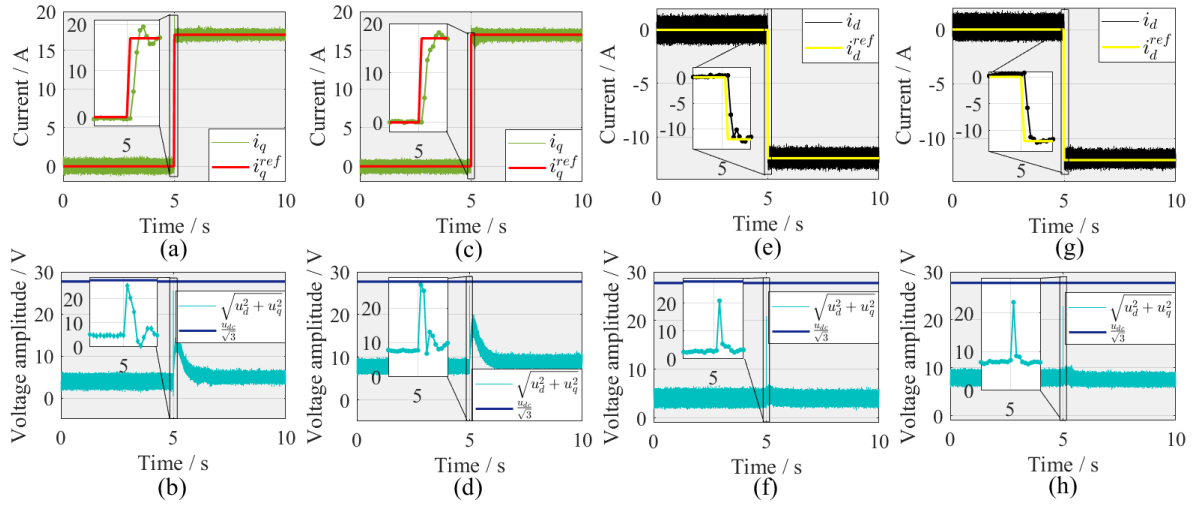
(a) Step changes of the  $d$  – axis current.



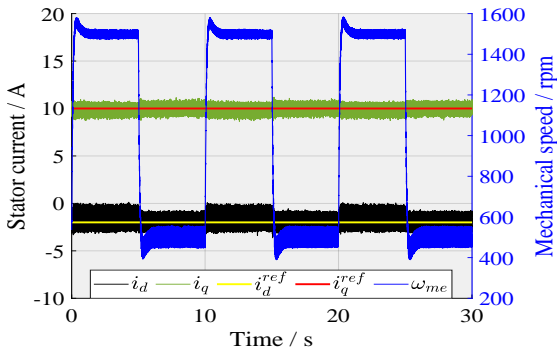
(b) Step changes of the  $q$  – axis current.

**Figure 5.5:** Experimental steady-state tracking results of the nonlinear LB-CCS-MPC at 1000 rpm.

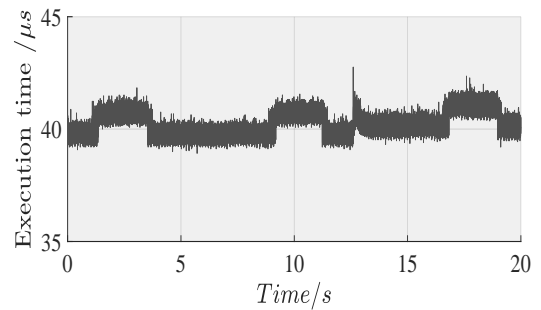
The transient performance is tested by making a step-up of  $i_q^{ref}$  from 0 A to 17 A at a constant  $i_d^{ref}$ , and a step-down of  $i_d^{ref}$  from 0 A to  $-12$  A at a constant  $i_q^{ref}$  at two different speeds of 500 rpm and 1000 rpm. The measured currents with their references as well as the commanded voltages from the proposed controller with their constraint are presented in Fig. 5.6. It is clear that the proposed LB-CCS-MPC provides a very fast dynamic performance with minimal overshooting and while fulfilling the input constraints. This is valid in transients in both current axes. In Fig. 5.7, the robustness of the controller against speed variation is proved while injecting constant currents. Finally, the execution time of the proposed control scheme is recorded on the dSpace MicroAutoBox II platform, and is depicted in Fig. 5.8. Taking only  $40 \mu\text{s}$  makes this approach by approximating beneficial MPC control policy via a NN appealing, and computationally-implementable for other power electronic systems.



**Figure 5.6:** Dynamic response using the proposed LB-CCS-MPC scheme for a step-up change of  $i_q^{ref}$  from 0 A to 17 A at 500 rpm is shown in (a) with the stator voltage amplitude and its constraint shown for that test in (b), the same test was repeated at 1000 rpm and the results are shown in (c) and (d), a step-down change of  $i_d^{ref}$  from 0 A to  $-12$  A at 500 rpm is shown in (e) with its stator voltage in (f), and in (g) and (h) the same results are shown for the same step change at 1000 rpm.



**Figure 5.7:** Steady current tracking with load speed variation using the nonlinear LB-CCS-MPC scheme.



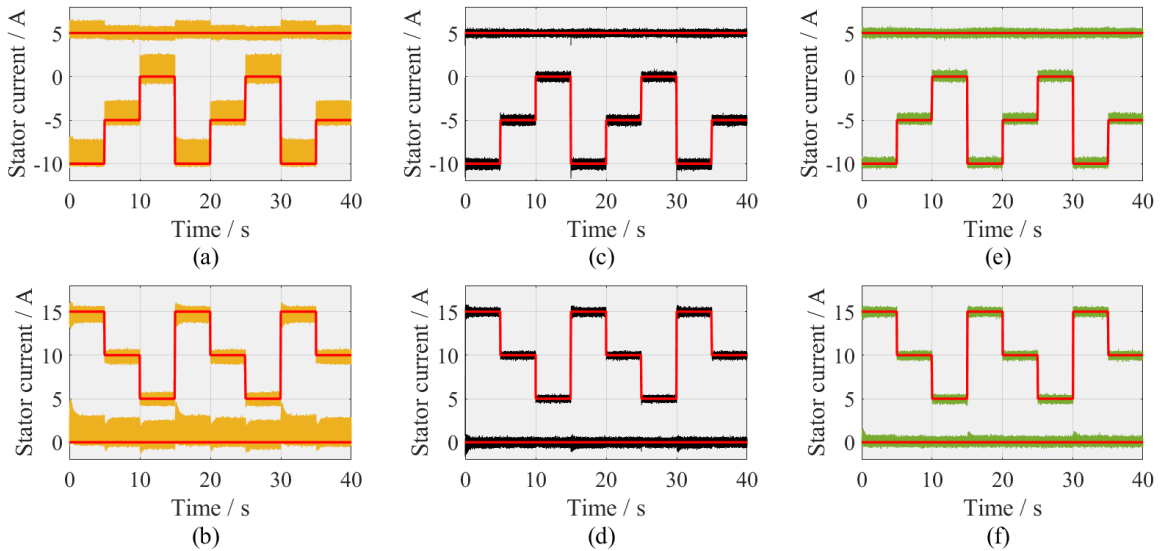
**Figure 5.8:** The measured execution time  $T_{ex}$  of the proposed LB-CCS-MPC scheme, measured on the dSPACE MicroAutoBox II platform of the IPMSM test bench.

### 5.1.3 Benchmarking against the first-principles MPC

In the following, the closed-loop performance of the proposed LB-CCS-MPC scheme based on the LSTM neural model is benchmarked against the linear and nonlinear first-principles CCS-MPC schemes, both in steady-state and in transients.

#### 5.1.3.1 Steady-state operation

The three controllers are tested under the same test scenario by controlling the shaft speed to 500 rpm via the load motor and injecting constant reference current value to one axis while stepping the other axis reference current each 5 s. The tests are carried out at the IPMSM test bench shown in Fig. B.1, and the experimental results are shown in Fig. 5.9 for the three control schemes. The upper sub-figures represent the test where  $i_q^{ref}$  is fixed at 5 A and  $i_d^{ref}$  is stepped repeatedly between  $-10$  A and 0 A, where the bottom sub-figures represent the test in which  $i_d^{ref}$  is fixed at 0 A and  $i_q^{ref}$  is stepped repeatedly between 15 A and 5 A. The measurements using the L-FP-CCS-MPC scheme are shown in orange, and using the NL-FP-CCS-MPC are shown in black, where the ones obtained using the proposed nonlinear LB-CCS-MPC are shown in green. These results show that the proposed novel LB-CCS-MPC exhibits a similar steady-state tracking performance to the NL-FP-CCS-MPC in terms of offset-free tracking and minimal current ripples, with a slightly superior decoupling performance. The two nonlinear schemes which incorporate nonlinear models of the machine whether by first-principles (NL-FP-CCS-MPC) or data-driven (LB-CCS-MPC) are clearly superior to the simple L-FP-CCS-MPC in the aspects of ripples amplitude, offsets, and decoupling.

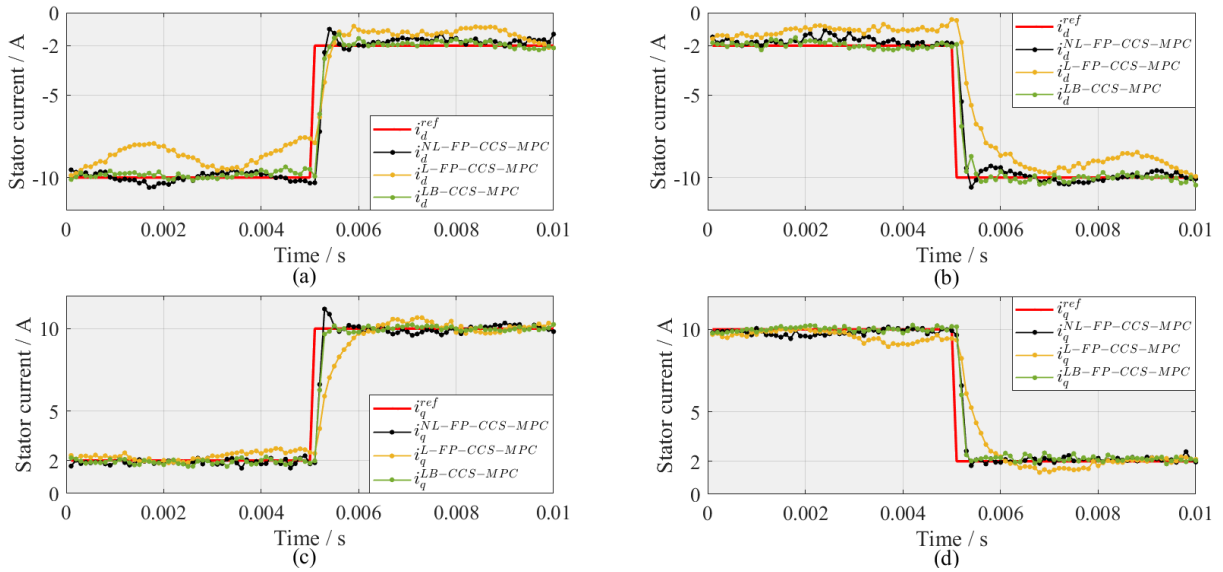


**Figure 5.9:** Benchmarking of the proposed nonlinear LB-CCS-MPC against the L-FP-CCS-MPC and the NL-FP-CCS-MPC schemes in steady-state current tracking operation at 500 rpm: (a)  $i_q^{ref}$  at 5 A and  $i_d^{ref}$  is stepped up and down between  $-10$  A to 0 A with 5 A increments using the L-FP-CCS-MPC scheme, (b)  $i_d^{ref}$  at 5 A and  $i_q^{ref}$  is stepped up and down between 15 A to 5 A with 5 A increments using the L-FP-CCS-MPC scheme, (c) and (d) are for the same test scenario using the NL-FP-CCS-MPC scheme, (e) and (f) are obtained using the proposed LB-CCS-MPC scheme with the LSTM neural model.

— Reference signals, — L-FP-CCS-MPC, — NL-FP-CCS-MPC, — LB-CCS-MPC.

### 5.1.3.2 Transient operation

The aim of this subsection is to highlight the performance enhancement in transient operation of using a nonlinear modelling approach (either data-driven or physics-based) against the by-far widely used LTI models for model-based control schemes for the current control loop in electric drive applications. For this purpose, a step up and a step down of  $i_q^{ref}$  between 2 A and 10 A and of  $i_d^{ref}$  between  $-2$  A and  $-10$  A were made at  $\omega_{me}$  of 750 rpm, and the results of the proposed LB-CCS-MPC as well as the NL-FP-CCS-MPC and the L-FP-CCS-MPC schemes are shown in Fig. 5.10 with a zoom-in at the transient instants. Recalling the fact that the currents are being measured at the middle of each sampling period, and considering that the current evolution will continue with the same slope in the second half of the sampling period as within the first half, it is shown that the proposed novel LB-CCS-MPC controller with the nonlinear LSTM data-driven model of the machine and the physics-based NL-FP-CCS-MPC scheme take the fastest possible way to the new references, unlike the slow L-FP-CCS-MPC that incorporates a simple LTI model of the machine with fixed parameters. Modelling discrepancy between real plants and the obtained models persist to occur to an extent, even when complex and nonlinear models are obtained. Temperature variation and components aging may also lead to a modelling mismatch. Therefore, it is a necessity for robust control to have an error integrating functionality that compensates for modelling mismatch. Recently, an offset-free formulation with an error integrating state for controllers based on NNs is proposed in [185], which is essential future research direction.



**Figure 5.10:** Comparison of the proposed LB-CCS-MPC scheme (in green) with the L-FP-CCS-MPC (in orange) and the NL-FP-CCS-MPC (in black) in transients at 750 rpm: (a) an ascending step in  $i_d^{ref}$ , (b) a descending step in  $i_d^{ref}$ , (c) an ascending step in  $i_q^{ref}$ , (d) a descending step in  $i_q^{ref}$ .



## CHAPTER 6

---

### Conclusion and outlook

---

#### 6.1 Conclusion

*“Life is the art of drawing sufficient conclusions from insufficient premises.”*

Samuel Butler

This dissertation has thoroughly dealt with MPC for electrical drive applications, taking the current control loop of PMSMs as a base example. The possible use of AI methods to alleviate some of MPC shortcomings is investigated. Two use cases are found to be potentially promising when the AI methods are used as an identification tool to obtain data-driven models from collected measurements without the necessity to bring foreknowledge about the plant to be controlled, or when used to approximate beneficial but computationally challenging control laws, whether the computational difficulty comes from the nature of the optimization problem or from the complexity of the incorporated model.

The foundation of the thesis is laid by defining the concept of MPC and thoroughly reviewing the literature regarding the use of MPC for power electronic and electrical drive applications in a chronological order, and highlighting the most recent active research areas in this domain of research. After that, the modelling part is preceded by calling attention to the importance of the pre-modelling foreknowledge in terms of inverter nonlinearity and angle delay compensation, as well as the needed synchronization between the applied voltages and measured currents. The importance of considering these effects for modelling and for model-based control is emphasized.

Following, the conventional and well-known nonlinear current dynamics model of the PMSM based on the physical laws that govern the functionality of the machine was derived, and its current-dependent parameters were obtained and smoothed via the splines interpolation method for efficient real-time implementation within model-based control schemes. The model was validated on real measurements from the test bench at different operating points, and it has shown excellent prediction accuracy, and hence, it can be used within different MPC schemes. Along

the modelling task, data-driven modelling is proposed as an appealing alternative, motivated by the hope for a less modelling effort without the necessity to bring detailed physical knowledge about the plant, along with better representation of the plant in terms of prediction accuracy. In particular, a simple FNN was able to learn the inverter nonlinearity from the measurements generated from one test that covers the  $\alpha - \beta$  plane, and to provide the needed feedforward compensation voltage as a function of the phase current with an accuracy close to 95% in comparison with the conventional LUT-based method which reacquires 3 tests and a slightly more effort to obtain the compensation curves. Moreover, an LSTM neural model was trained to capture the current dynamics of PMSMs purely from collected measurements using a simple PI controller to generate an identification data set without bringing any physical knowledge about the plant under control, and hence the obtained neural model is purely data-driven. It has shown an excellent prediction accuracy that motivates its suitability for the use within model-based predictive control. Remarks on a proper design of experiment to generate the identification data set from which a data-driven model of the machine under identification can be learned are given.

Among other data-driven modelling methods revisited from the literature, the chosen modelling approach via LSTM NNs features manageable computational and memory demand for the training process of the relatively large training data set in contrary to the GP regression modelling, as well as that it does not rely on bringing physical knowledge about the system to be identified in contrary to the KO-based identification in which bringing the known physical knowledge about the current dynamics proved to be advantageous in choosing the observable functions. In addition, the prediction accuracy of the LSTM NN in terms of maximum absolute prediction accuracy and RMSE value was the best among the other methods. For these reasons, the neural model is chosen for further incorporation within the proposed learning-based predictive control framework. By this, the conventional modelling approach based on the physical laws as well as the proposed data-driven model via an LSTM NN are derived, optimized, validated, and ready for the use to design MPC control schemes.

On the control side, the general formulation of MPC is stated, and it is classified according to whether a modulator is used or not into direct and indirect MPC. In the direct MPC, the general concept in its simplest form of current tracking is illustrated as an example showing its intuitiveness, simplicity, and suitability for electrical drive applications. Next, the conventional direct MPC scheme with a one-step prediction horizon is applied for current control of PMSMs. Owing the high current ripples due to applying only one VV at the sampling period, the necessity to shrink the computational demand of the algorithm arises in order to achieve high sampling frequencies, and consequently switching frequencies that would reduce the current ripples. Motivated by this, a computationally-efficient one-step direct MPC scheme for the current control is proposed, which reduces the computational demand by 31.1%, and consequently, allows for sampling at higher switching frequencies. After that, it is justified why multiple-vector direct MPC is excluded from further investigation within this dissertation, and that it is recommended to go for an indirect MPC alternative with an appropriate modulation scheme in case more than one VV are to be applied within a sampling period to minimize ripples. The impact of extending the prediction horizon is analyzed and it is found that for a synchronous machine drive, it yields an improved THD content at a given average switching frequency  $f_{sw,avg}$  as well as a better distribution of the switching events over the three phases at a given THD content. Due to the computational intractability of the underlying mixed-integer optimization problem in long-horizon direct MPC, which originates from the exponential increase of the needed computations



with the increase in the prediction horizon length, it is proposed to solve the optimization problem offline on a powerful computing platform for sufficient trajectories that cover the operation range of interest, and then to learn the optimal solution via a computationally-implementable NN that runs in real-time as a controller with the aim to mimic the beneficial performance characteristics of the long-horizon direct MPC that it has learned. This proposal is applied in simulation on an IPMSM drive system to mimic the direct MPC with prediction horizon of  $N = 5$  steps, and the proposed NN has achieved a prediction accuracy of 85 – 90% in comparison to the long-horizon direct MPC that it has learned, significantly outperformed the one-step direct MPC, and it is capable of running in real-time for power electronic and electrical drive systems as the HIL test has revealed that it requires only  $\sim 15 \mu\text{s}$ .

The indirect variant of MPC took the center of attention of this dissertation with the objectives of evaluating the performance gain of using the derived and obtained nonlinear models, whether by physical laws or data-driven, in comparison with the by-far most used LTI model of the machine, as well as the real-time implementation of the indirect MPC schemes in the sub-millisecond range, and thus, tackling the main argument against the numerical indirect MPC for electrical drive applications. To facilitate the implementation of the numerical indirect MPC (i.e. CCS-MPC), a numerical solver based on a slack formulation of the primal-dual IP method is proposed and used to find an approximate solution of the associated constrained optimization problem while incorporating the linear or nonlinear first-principles models of the machine, and fulfilling the states and inputs constraints. The proposed solver was deployed on two dSPACE platforms and experimentally tested for the current control loop of a SMPMSM and an IPMSM with sampling frequencies of 8 kHz and 10 kHz, respectively, and it can be directly modified to be used for other electrical drive systems. To evaluate the modelling effect on the closed-loop performance, the simple LTI model of the IPMSM is firstly incorporated in a CCS-MPC scheme with and without compensating the inverter nonlinearity. The experimental results have suffered from remarkable tracking offsets and higher ripples on the current with poor dynamic performance in transients. Still, these effects persists to occur with having the inverter nonlinearity compensated while using the LTI model of the machine. To mitigate the offsets, it is proposed to embed an error integrating functionality within the MPC formulation by using the input increments  $\delta \mathbf{u}$  as decision variables instead of the absolute inputs  $\mathbf{u}$ . This has been implemented and tested experimentally, and has proved to guarantee an offset-free performance, however, as expected it did not improve the performance in transients and overshooting has occurred. After that, the nonlinear current-dependent model is incorporated, and it has shown an offset-free tracking with minimal current ripples, and it has provided the fastest possible transient performance with respect to the system constraints. However, having an error-integration functionality such as the proposed  $\delta \mathbf{u}$  formulation is necessary even while using the nonlinear model in order to enhance the robustness of model-based control against slight modelling mismatches and disturbances. Moreover, the proposed CCS-MPC scheme fulfills the states constraints which is beneficial to protect the machine and the inverter, and the input constraints which makes the machine operates at its physical limit. This makes the proposed CCS-MPC in particular an interesting approach especially in FW operation. From the computational perspective, the proposed solver has tackled the underlying optimization problem of the CCS-MPC with either the linear or the nonlinear first principles models and with prediction horizon up to 2 steps within  $70 \mu\text{s}$ . Along the lines of indirect MPC, its analytical variant (i.e. DB-MPC) that is characterized by its significantly low computational demand and model-dependent transient and

steady-state performance is investigated and tested. Same conclusions regarding the modelling effects on the control performance were drawn. However, the way of tackling the constraints by the projection method is sub-optimal, and the lack of having a cost function and consequently multi-objective optimization are drawbacks which make the CCS-MPC the favorable indirect variant from a design and performance perspectives, especially when multi control objectives are simultaneously required.

Closing the loop, the possibility to incorporate the proposed data-driven model based on the LSTM NN within a CCS-MPC scheme was investigated. One evaluation of the model takes  $\sim 25 \mu\text{s}$ , thus, using this model in an iterative manner to solve the CCS-MPC problem is unmanageable within the used sampling frequency of 10 kHz. Therefore, an explicit-like MPC approach is proposed. The proposed approach is based on solving the problem offline while covering the span of operating points of interest to generate the mapping between initial conditions and desired evolution of the states to the optimal control input computed by solving the CCS-MPC problem while using the LSTM neural model to make the current predictions. Once a data set that represents this mapping is generated, a NN is trained to learn this mapping/control policy in a computationally-implementable way to be deployed as a controller in real-time. This proposal was implemented and the yielded LB-CCS-MPC was experimentally tested and benchmarked against the CCS-MPC schemes with the linear and nonlinear physics-based models. The excellent closed-loop performance it has shown by significantly outperforming the classical PI-FOC and the L-FP-CCS-MPC, and exhibiting similar performance in comparison with the NL-FP-CCS-MPC, have cemented the benefit of the two use cases of AI methods with MPC in the area of power electronic and electrical drive systems, these are: *modelling* purely from the measurements without the necessity for prior physical knowledge, and approximating beneficial but computationally-complex *control* laws in a computationally-implementable manner.

## 6.2 Potential future research directions

The following research directions are from the author's point of view attractive, and needed for further advancement of MPC in the domain of power electronic and electrical drive applications:

- Extending the application of data-driven modelling further than the current dynamics to address specific KPIs. One possibility in this direction is to benefit from data-driven modelling to enhance the acoustic modelling of the machine to improve the noise, vibration, and harshness (NVH) behaviour of the drive.
- Developing dedicated efficient numerical solvers for the nonlinear CCS-MPC to facilitate the use of longer prediction horizons and complex nonlinear models. This step is crucial in motivating and widening its applicability and benefit especially for high-order systems.
- Applying the proposed data-driven methods for other power electronic and electrical drive applications where nonlinearities are harder to model via first-principles and physical laws. For instance, modelling the current dynamics of synchronous reluctance machines

as well as the nonlinearity of multilevel inverters/ inverters based on wide-bandgap semiconductors are foreseen potential examples, where data-driven modelling might outperform first-principles modelling.

- For the proposed NN-based controllers, the constraints fulfilment via the projection method is considered sub-optimal, and is a direction for further investigation. Moreover, the lack of possibility to adjust the priorities of different control objectives (e.g. to adjust the average sampling frequency in case of NN-based direct MPC) is yet a substantial limitation.
- Elegant error integrating functionalities with the data-driven MPC would enhance its robustness against modelling mismatch due to aging, temperature variation and disturbances.
- Extending the use of NNs not only to mimic MPC policies based on complex nonlinear models or long prediction horizons, but to mimic OPPs.



# APPENDIX A

---

## List of publications

---

### A.1 Journal papers

1. **I. Hammoud**, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, “Learning-Based Model Predictive Current Control For Synchronous Machines: an LSTM Approach”, *European Journal of Control*, pp. 100663, June. 2022, doi: <https://doi.org/10.1016/j.ejcon.2022.100663>.
2. **I. Hammoud**, S. Hentzelt, K. Xu, T. Oehlschlaegel, M. Abdelrahem, C. Hackl, and R. Kennel, “On Continuous Set Model Predictive Control of Permanent Magnet Synchronous Machines”, *IEEE Transactions on Power Electronics*, vol. 37, no. 9, pp. 10360-10371, Sept. 2022, doi: 10.1109/TPEL.2022.3164968.
3. **I. Hammoud**, K. Morsy, M. Abdelrahem, and R. Kennel, “Efficient model predictive power control with online inductance estimation for photovoltaic inverters”, *Electrical Engineering*, vol. 102, no. 2, pp. 549-562, 2020.

### A.2 Conference papers

1. **I. Hammoud**, K. Morsy, M. Abdelrahem, and R. Kennel, “Computationally Efficient Model Predictive Direct Power Control with Online Finite Set Model Inductance Estimation Technique for Grid-Connected Photovoltaic Inverters”, in *IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, Quanzhou, China, 2019, pp. 1-6.
2. **I. Hammoud**, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, “Computationally efficient finite-set model predictive current control of interior permanent magnet synchronous motors with model-based online inductance estimation”, in *IEEE Conference on Power Electronics and Renewable Energy (CPERE)*, Aswan, Egypt, 2019, pp. 290-295.

3. **I. Hammoud**, S. Hentzelt, K. Xu, T. Oehlschlaegel, and R. Kennel, “On Offset-free Continuous Model Predictive Current Control of Permanent Magnet Synchronous Motors”, in *21<sup>st</sup> IFAC World Congress*, Berlin, Germany, 2020, vol. 53, no. 2, pp. 6662-6669, 2020.
4. **I. Hammoud**, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, “Long-Horizon Direct Model Predictive Control Based on Neural Networks for Electrical Drives”, in *The 46<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Singapore, Singapore, 2020, pp. 3057-3064
5. **I. Hammoud**, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, “Modelling for Nonlinear Predictive Control of Synchronous Machines: First Principles Vs. Data-Driven Approaches”, in *IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, Jinan, China, 2021, pp. 1-6.
6. H. M. Calderon, **I. Hammoud**, T. Oehlschlaegel, H. Werner, and R. Kennel, “Data-Driven Model Predictive Current Control for Synchronous Machines : a Koopman Operator Approach”, in *the IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, Sorrento , Italy, 2022, pp. 942-947

### A.3 Awards

1. **Best student paper award**, received from the 5<sup>th</sup> IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE 2019), Quanzhou, China, for the paper entitled “Computationally Efficient Model Predictive Direct Power Control with Online Finite Set Model Inductance Estimation Technique for Grid-Connected Photovoltaic Inverters”.
2. **Best paper award**, received from the 6<sup>th</sup> IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE 2021), Jinan, China, for the paper entitled “Modelling for Nonlinear Predictive Control of Synchronous Machines: First Principles Vs. Data-Driven Approaches”.

### A.4 Supervised and co-supervised theses

1. Hang Yu, Master Thesis: Data-Driven Modelling of Permanent Magnet Synchronous Motors via Gaussian Process Regression, Technical University of Munich.
2. Muhammad Shaur Khan, Master Thesis: Koopman Operator-Based Predictive Control of a Permanent Magnet Synchronous Machine, Hamburg University of Technology.
3. Christian Linseis, Master Thesis: Conventional and model predictive control for induction machines: A Simulative Comparison, Technical University of Munich.

## A.5 List of symbols

$\mathbb{R}$	Real numbers
$\frac{d}{dt}$	Time derivative
$\frac{\partial f}{\partial x}$	Partial derivative of $f$ with respect to $x$
$abc$	Referring to quantities represented in the three-phase system
$\alpha\beta$	Referring to quantities represented in the stationary two-phase frame
$dq$	Referring to quantities represented in the rotary two-phase frame
$\delta, \Delta$	Delta operator
$\nabla$	Gradient operator
$\circ$	Element-wise multiplication
$\sigma$	Arbitrary activation function
$\sigma_g$	Sigmoid activation function
$\sigma_h$	Tangent hyperbolic activation function
$\sigma_s^2$	Signal variance
$\sigma_n^2$	Measurement noise variance
$\ell_1, \ell_2$	The absolute value and the Euclidean norms
$\ell_l, \ell_u$	Lower and upper scaling bounds used in normalization
$\ell_D$	Length scale
$\delta_{p,q}$	The Kronecker delta
$\top$	Transpose of a vector
$i, j, \iota$	Symbols used as counters
$E$	Expectation
$\text{Var}$	Variance
$\dagger$	Moore-Penrose pseudo-inverse
$\ \cdot\ _F^2$	The Frobenius norm
$\ \cdot\ _p^p$	Norm
$ \cdot $	Absolute value of a scalar
$a$	Weighted and biased sum of a neuron to be activated
$\mathbf{A}_c$	Continuous LTI state-space system matrix
$\mathbf{A}$	Discrete time-invariant state-space system matrix
$\mathbf{A}_k$	Discrete time-varying state-space system matrix, which contains parameters as functions of the states
$b$	Bias
$\mathbf{b}$	Bias vector
$\mathbf{B}_c$	Continuous LTI state-space system matrix
$\mathbf{B}$	Discrete time-invariant state-space system matrix
$\mathbf{B}_k$	Discrete time-varying state-space system matrix, which contains parameters as functions of the states
$\mathbf{c}$	LSTM cell state vector

$\tilde{\mathbf{c}}$	LSTM cell input activation vector
$\mathbf{C}_c$	Continuous LTI state-space system matrix
$\mathbf{C}$	Discrete time-invariant state-space system matrix
$D$	Inputs dimension of the training data set
$d_a^{Missing}$	Faulty duty cycle in phase $a$ due to the VSI nonlinear effect
$d_b^{Missing}$	Faulty duty cycle in phase $b$ due to the VSI nonlinear effect
$d_c^{Missing}$	Faulty duty cycle in phase $c$ due to the VSI nonlinear effect
$\mathbf{d}$	Disturbance or integral state vector
$\mathcal{D}$	Training data set
$\mathbf{D}_c$	Continuous LTI state-space system matrix
$\mathbf{D}$	Discrete time-invariant state-space system matrix
$err_{max}$	Maximum absolute error in the current prediction
$e / \mathbf{e}$	Error signal or error vector as explicitly defined in the context
$\mathbf{e}$	Inequality constraints vector
$f_{sw,avg}$	Average switching frequency
$f_{sw,a}, f_{sw,b}, f_{sw,c}$	Switching frequency of phases $a$ , $b$ , and $c$ , respectively
$f_s$	Sampling frequency
$\mathbf{f}$	LSTM forget gate activation vector
$\mathbf{f}_c$	Continuous-time general nonlinear system dynamics function
$\mathbf{f}$	Discrete-time general nonlinear system dynamics function
$\mathbf{g}$	Equality constraints vector
$\mathbf{h}$	LSTM hidden state/LSTM cell output vector
$\mathbf{h}_c$	Continuous-time general nonlinear output function
$\mathbf{h}$	Discrete-time general nonlinear output function
$\mathbf{H}$	The Heessian of the Lagrangian
$I$	Amplitude of the stator current
$I_{max}$	Maximum allowed current amplitude
$I_{THD}$	Total harmonic distortion of the stator current
$I_{THD,rel}$	Relative current THD
$i_d$	$d - axis$ stator current
$i_q$	$q - axis$ stator current
$i_d^{ref}$	$d - axis$ reference stator current
$i_q^{ref}$	$q - axis$ reference stator current
$i_d^{Measured}$	Measured $d - axis$ current
$i_q^{Measured}$	Measured $q - axis$ current
$i_d^{Predicted-FP}$	Predicted $d - axis$ current using the physics-based first-principles model
$i_q^{Predicted-FP}$	Predicted $q - axis$ current using the physics-based first-principles model



$\hat{i}_d^{Predicted-NN}$	Predicted $d - axis$ current using the LSTM neural model
$\hat{i}_q^{Predicted-NN}$	Predicted $q - axis$ current using the LSTM neural model
$\hat{i}_d^{Predicted-GP}$	Predicted $d - axis$ current using the GP model
$\hat{i}_q^{Predicted-GP}$	Predicted $q - axis$ current using the GP model
$i_d^{fil}$	Filtered $d - axis$ current
$i_q^{fil}$	Filtered $q - axis$ current
$\mathbf{i}, \mathbf{i}_{abc}, \mathbf{i}_{\alpha\beta}, \mathbf{i}_{dq}$	Stator current
$\mathbf{i}^{rated}$	Rated stator current
$\mathbf{i}$	LSTM input gate activation vector
$\mathbf{I}$	Identity matrix
$\mathcal{I}$	Integral gains diagonal matrix
$J$	The general cost function definition
$\mathcal{J}$	The general cost function formulation with persistence step disturbance compensation
$J_1$	The cost function used for FCS-MPC
$J_2$	The cost function used for the proposed computationally-efficient FCS-MPCC scheme
$J_3$	The cost function used for the LB-CCS-MPC
$\mathbf{j}$	Vector containing manipulation terms used in the current prediction equation of IPMSMs.
$k$	Discrete time index
$\mathbf{k}(\mathbf{x}_p, \mathbf{x}_q)$	Covariance (also called kernel) function evaluated at $\mathbf{x}_p$ and $\mathbf{x}_q$
$\mathcal{K}$	The Koopman operator
$\mathcal{K}$	Matrix representing a finite-dimensional approximation of the KO
$L_d, L_q$	Absolute inductances
$L_{dd}, L_{qq}$	Self differential inductances
$L_{dq}, L_{qd}$	Mutual differential inductances
$L$	Length of the data set (number of sampling instants)
$\mathcal{L}$	The Lagrangian
$m_m^{ref}$	Reference mechanical torque
$m_m$	Mechanical torque
$N$	Length of the prediction horizon
$N_u, N_x, N_y$	Number of inputs, states, and outputs, respectively
$n_p$	Number of pole pairs
$n_{u,max}, n_{x,max}$	Maximum number of discrete-time delays on the inputs and states
$n$	Maximum number of discrete-time delays of the inputs of a neural network/ a symbol denoting the last entry of a vector
$n$	Number of neurons in the previous layer of a neural network
$N_{CPSP}$	Number of computations per sampling period
$N_{VVs}$	Number of output voltage vectors of a power converter

$n_{eq}$	Number of the equality constraints for the IP solver
$n_{inq}$	Number of the inequality constraints for the IP solver
$n_{dv}$	Number of the decision variables for the IP solver
$n_{\psi}$	Number of observable functions
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$	Gaussian (normal) distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{K}$
$\mathbf{o}$	LSTM output gate activation vector
$\mathcal{O}(\cdot)$	Notation the computational complexity of an algorithm
$\mathbf{O}$	Zero matrix
$p$	Variable to define which norm is being used
$P$	Power
$\mathbf{P}_k$	Transformation matrix to facilitate the direct use of the switching states as control inputs in the state-space representation for direct MPC design
$\mathbf{Q}$	Weighting matrix
$R_s$	Stator resistance
$\mathbf{r}$	Output's reference vector
$\mathbf{r}^*$	Shifted output's reference vector by the most recent prediction error
$\mathbf{r}_T$	The residuals used within the IP numerical solver
$\mathbf{R}$	Weighting matrix
$\mathbf{s}_{abc}$	Switching states vector
$\mathbf{s}_k$	Slack variables vector for the IP numerical solver
$t$	Time
$T_s$	Sampling time/ period
$T_{step}$	Time duration of each segment in the excitation data set
$T_{ex}$	Execution time
$T_e$	Electromagnetic torque
$T_e^{ref}$	Reference electromagnetic torque
$\mathbf{T}_{ptp}, \mathbf{T}_{abc}$	Coefficient matrices
$u_{dc}$	$dc$ -link voltage
$u_{max}$	Voltage constraint
$u_{mag}$	Magnitude of the reference stator voltage vector
$u_d$	$d$ – axis stator voltage
$u_q$	$q$ – axis stator voltage
$u_d^{ref}$	$d$ – axis reference stator voltage
$u_q^{ref}$	$q$ – axis reference stator voltage
$u_a^{Missing}$	Voltage drop in phase $a$ due to the VSI nonlinear effect
$u_b^{Missing}$	Voltage drop in phase $b$ due to the VSI nonlinear effect

$u_c^{Missing}$	Voltage drop in phase $c$ due to the VSI nonlinear effect
$u_\alpha^{Missing}$	$\alpha - axis$ component of the voltage drop due to the VSI nonlinear effect
$u_\beta^{Missing}$	$\beta - axis$ component of the voltage drop due to the VSI nonlinear effect
$u_\alpha^{Error}$	$\alpha - axis$ component of the voltage error between the NN output and the real inverter nonlinearity function
$u_\beta^{Error}$	$\beta - axis$ component of the voltage error between the NN output and the real inverter nonlinearity function
$\mathbf{u}_{ptp}^f$	Phase-to-phase voltage of the inverter
$\mathbf{u}_{abc}^f$	Inverter output voltage vector
$\mathbf{u}$	Inputs vector
$\tilde{\mathbf{u}}$	Inputs vector at the beginning of a sampling period on a digital platform
$\mathbf{u}_{abc}, \mathbf{u}_{\alpha\beta}, \mathbf{u}_{dq}$	Stator voltage
$\mathbf{u}_{min}, \mathbf{u}_{max}$	Lower and upper bounds of the inputs constraints
$\mathbf{u}_{abc}$	Inputs vector defined by the switching state of the power converter
$\mathbf{u}_{\alpha\beta}^{ref}$	Reference stator voltage vector in the $\alpha - \beta$ frame
$\mathbf{u}_{dq}^{ref}$	Reference stator voltage vector in the $d - q$ frame
$\mathbf{u}_1, \dots, \mathbf{u}_8$	The discrete VVs of a 2-levels VSI
$\mathbf{u}^{r1}, \mathbf{u}^{r2}, \mathbf{u}^{r3}$	Dynamically defined 3 discrete VVs that define the sector in which the reference VV lies.
$\mathbf{u}_{abc}^{inv}$	Three-phase feedforward voltage compensation of the inverter nonlinearity
$\Delta \mathbf{u}_k$	Vector of control inputs increments
$\mathbf{u}_k^{i-ccsmc}$	Summation of the integral and the CCS-MPC control inputs
$\mathbf{u}_k^{i-dbmc}$	Summation of the integral and the DB-MPC control inputs
$\mathbf{u}_k^{int}$	Additive integral control input
$\mathbf{U}$	Control input sequence
$\mathbf{v}$	Measured disturbance vector
$\mathcal{W}$	Weight of a neuron
$\mathbf{W}$	Weighting matrix for the LSTM NN
$\chi$	Input feature of a neuron
$\mathbf{x}$	States vector
$\dot{\mathbf{x}}$	System dynamics vector
$\tilde{\mathbf{x}}_0$	Initial state/ the states measurement/observations at the beginning of the prediction period
$\tilde{\mathbf{x}}$	States vector measured at the middle of a sampling period on a digital platform
$\mathbf{x}_{min}, \mathbf{x}_{max}$	Lower and upper bounds of the states constraints
$\boldsymbol{\chi}$	Input vector to the neural network

$\Delta \mathbf{x}_k$	Vector of states increments
$\mathbf{X}$	Input of the training data set for the Gaussian process
$y$	Output of a neuron
$\mathbf{y}$	Outputs vector
$\tilde{\mathbf{y}}_{k k-1}$	Most recent output predictions at time instant $k$ made at $k - 1$
$\mathbf{y}$	Output vector of the neural network
$\mathbf{y}$	Output of the training data set for the Gaussian process
$\mathbf{z}$	Vector that contains the summed weighted inputs of the output layer of a neural network
$\mathbf{Z}$	Weighting matrix for the LSTM NN
$\gamma$	Term to represent soft states constraints
$\epsilon$	Scaling parameter in the IP numerical solver
$\zeta$	Arbitrary vector used to define vector transformation
$\eta$	Decision variables vector for the IP numerical solver
$\theta$	Electrical angle of rotation
$\kappa_{ls}$	Reduction parameter in the IP numerical solver
$\lambda_u$	Scalar weighting term
$\mu_k,$	Dual variable vector for the IP numerical solver
$\mu(\mathbf{x})$	Mean function of a Gaussian process
$\nu$	Arbitrary vector used to define norms
$\xi$	Arbitrary vector used to define normalization
$\mathbf{\Pi}, \tilde{\mathbf{\Pi}}$	Two snapshot matrices representing the observables vector computed at consecutively equidistant discrete-time sampling with one sampling time step difference between $\mathbf{\Pi}$ and $\tilde{\mathbf{\Pi}}$
$\rho$	Vector of residuals/ approximation errors
$\tau$	The barrier parameter in the IP numerical solver
$\mathbf{v}_k$	Dual variable vector for the IP numerical solver
$\phi_k^{ref}$	Angle of the reference stator voltage vector computed by the deadbeat principle

---

$\psi_d, \psi_q$	Flux components
$\psi_p$	Permanent magnet flux linkage
$\boldsymbol{\psi}$	Vector containing the user-defined scalar observable functions
$\omega_{el}$	Electrical angular speed
$\omega_{el}^{rated}$	Rated electrical speed
$\omega_{me}$	Mechanical angular speed
$\omega_{me}^{ref}$	Reference mechanical angular speed

## A.6 List of abbreviations

<i>ac</i>	Alternating current
ADC	Angle delay compensation
AI	Artificial intelligence
CCS-MPC	Continuous control set model predictive control
CPSP	Computations per sampling period
DB-MPC	Deadbeat model predictive control
<i>dc</i>	Direct current
DFIM	Doubly fed induction machine
DMC	Dynamic matrix control
DMD	Dynamic mode decomposition
DSC	Direct self control
DTC	Direct torque control
EDMD	Extended dynamic mode decomposition
EKF	Extended Kalman filter
EMPC	Explicit model predictive control
FCS-MPC	Finite control set model predictive control
FCS-MPCC	Finite control set model predictive current control
FNN	Feedforward neural network
FOC	Field oriented control
FPGAs	Field programmable gate arrays
FW	Field-weakening
GP	Gaussian process
GPC	Generalized predictive control
HIL	Hardware-in-the-loop
I-CCS-MPC	Integral continuous control set model predictive control
I-DB-MPC	Integral deadbeat model predictive control
I-FCS-MPC	Integral finite control set model predictive control
IM	Induction machine
IP	Interior-point
IPMSM	Interior permanent magnet synchronous motor
KKT	Karush–Kuhn–Tucker
KO	Koopman operator
KPI	Key performance index
LB-CCS-MPC	Learning-based continuous control set model predictive control
LCL	Inductive-capacitive-inductive
L-FP-CCS-MPC	Linear first-principles continuous control set model predictive control
LSTM	Long-short term memory
LTI	Linear time-invariant
LUT	Lookup table
MBPC	Model-based predictive control
MCUs	Microcontrollers

---

MHE	Moving horizon estimation
ML	Machine learning
MLP	Multilayer perceptron
MPC	Model predictive control
MP <sup>3</sup> C	Model predictive pulse pattern control
MSE	Mean-squared error
MTPA	Maximum torque per ampere
MTPF	Maximum torque per flux
MTPV	Maximum torque per volt
MV-MPC	Multiple vector model predictive control
NL-FP-CCS-MPC	Nonlinear first-principles continuous control set model predictive control
NN	Neural network
NPC	Neutral point clamped
NVH	Noise, vibration, and harshness
OCP	Optimal control problem
OPP	Optimal pulse pattern
PI	Proportional-integral
PI-FOC	Field oriented control based on proportional-integral controllers
PMSM	Permanent magnet synchronous motor
PWM	Pulse width modulation
ReLU	Rectified linear unit
RLS	Recursive least squares
RMSE	Root-mean-squared error
SISO	Single-input single-output
SMPMSM	Surface-mounted permanent magnet synchronous motor
SQP	Sequential quadratic programming
SVM	Space vector modulation
THD	Total harmonic distortion
UKF	Unscented Kalman filter
VSI	Voltage source inverter
VSP-MPC	Variable switching point model predictive control
VV	Voltage vector





# APPENDIX B

---

## Test benches

---

In this appendix, the test benches that were used to experimentally validate the proposed modelling and control methods throughout this dissertation are introduced, with the corresponding real-time platforms, power converters, and measurement systems. The first test bench was built at IAV GmbH in Gifhorn, Germany, and it has been used as the main test bench where most of the methods have been validated. The second test bench is at the Laboratory for Mechatronic and Renewable Energy Systems (LMRES) in Munich University of Applied Sciences, Munich, Germany, and it has been used minorly to validate the *direct* control techniques<sup>1</sup>, and to validate the CCS-MPC algorithm on the other kind of synchronous machines (i.e. SMPMSM).

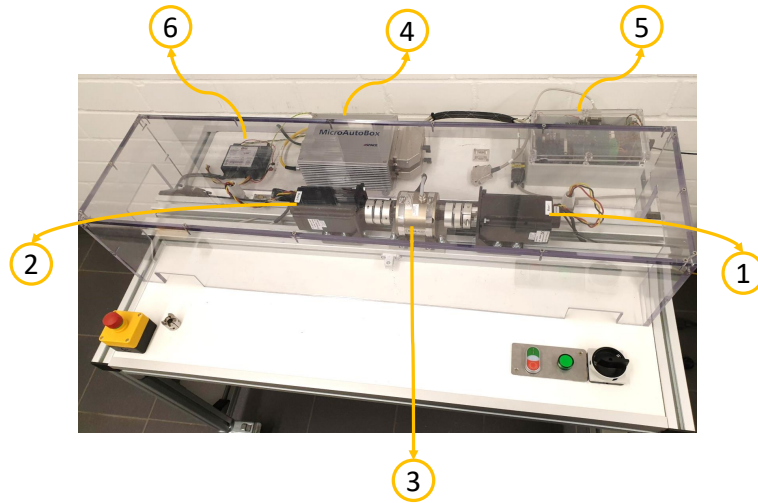
### B.1 Test bench of the IPMSM

This test bench consists of two IPMSMs equipped with incremental encoder (WEDL/WEDS5541(1000CPR)) manufactured by Nanotec Electronic GmbH & Co KG in Germany, and coupled via a torque sensor. The main motor used for the experimental part to validate the modelling and control contributions of this dissertation is the one on the right, and it is fed by a 2-levels VSI within a DRV830x Kit from Texas Instruments. The three-phase currents are measured using three LEM current sensors, and the dc-link voltage is measured using a LEM voltage sensor as well. The current and voltage measurements are processed using a separate measurements board designed internally at IAV GmbH. The load machine is speed-regulated. The speed control of the load machine is done via a commercial drive controller (Ott Box AMI1018-01) from Ott GmbH & Co. KG in Germany. The control algorithms of the main motor are developed on MATLAB/Simulink, and the models are deployed to a dSPACE MicroAutoBox II 1513/1514. The motors are supplied by a dc power supply that can provide up to 30 A. The test bench is depicted in Fig. B.1, and the parameters of both motors are summarized

---

<sup>1</sup>As the motors in the first test bench have very low inductances, and hence, direct control methods are not suitable for them at normal sampling frequencies.

in Table B.1.



**Figure B.1:** IPMSM test bench: (1) main motor, (2) load machine, (3) torque sensor, (4) dSPACE MicroAutoBox II 1513/1514, (5) 2-levels VSI with the phase currents and dc-link voltage sensors and their corresponding measurements board, (6) Speed regulator for the load machine from Ott GmbH.

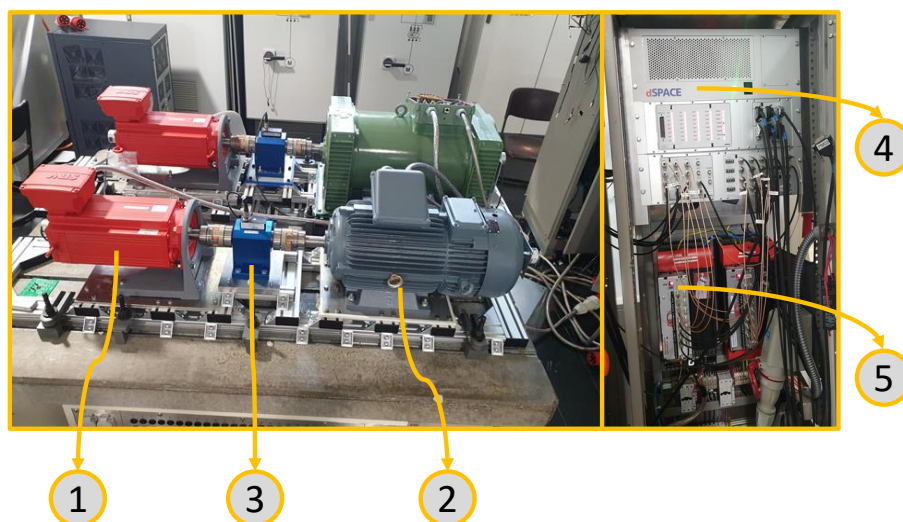
**Table B.1:** Parameters of the IPMSMs shown in Fig. B.1.

Name	Nomenclature	Value
dc-link voltage	$u_{dc}$	48 V
Maximum current	$i^{rated}$	17 A
Stator resistance	$R_s$	38.5 m $\Omega$
$d - axis$ inductance	$L_d$	50 $\mu$ H
$q - axis$ inductance	$L_q$	65 $\mu$ H
Sampling frequency	$f_s$	10 kHz
Flux linkage	$\psi_p$	0.02 Wb

## B.2 Test bench of the SMPMSM

This test bench consists of a SMPMSM as a main machine where the proposed control algorithms are to be tested and deployed, and a doubly fed induction machine (DFIM) as a load machine. A ROD 486 incremental encoder from DR. JOHANNES HEIDENHAIN GmbH with 2048 pulses per revolution is equipped to the SMPMSM and is connected to a DS3002 board. The currents and dc-link voltage measurements are processed by the controller via DS2004 analog to digital converter board. The inverter gate signals are being sent via a DS5104 PWM

board. The control algorithms are deployed on a dSPACE DS1007 platform, and the motor is fed by a conventional 2-levels VSI from SEW-Eurodrive GmbH & Co KG. The test bench is depicted in Fig. B.2, and the parameters of the SMPMSM are summarized in Table B.2, where the parameters of the load machine are summarized in Table B.3.



**Figure B.2:** SMPMSM test bench: (1) main motor (SMPMSM), (2) load machine (DFIM), (3) torque sensor, (4) dSPACE DS1007 platform, (5) 2-levels VSI from SEW-Eurodrive GmbH & Co KG.

**Table B.2:** Parameters of the SMPMSM shown in Fig. B.2.

Name	Nomenclature	Value
dc-link voltage	$u_{dc}$	560 V
Power	$P$	14.5 kW
Rated speed	$\omega_{el}^{rated}$	209 rad s <sup>-1</sup>
Stator resistance	$R_s$	0.15 $\Omega$
$d$ – axis inductance	$L_d$	3.4 mH
$q$ – axis inductance	$L_q$	3.4 mH
Sampling frequency	$f_s$	8 kHz
Flux linkage	$\psi_p$	0.375 Wb

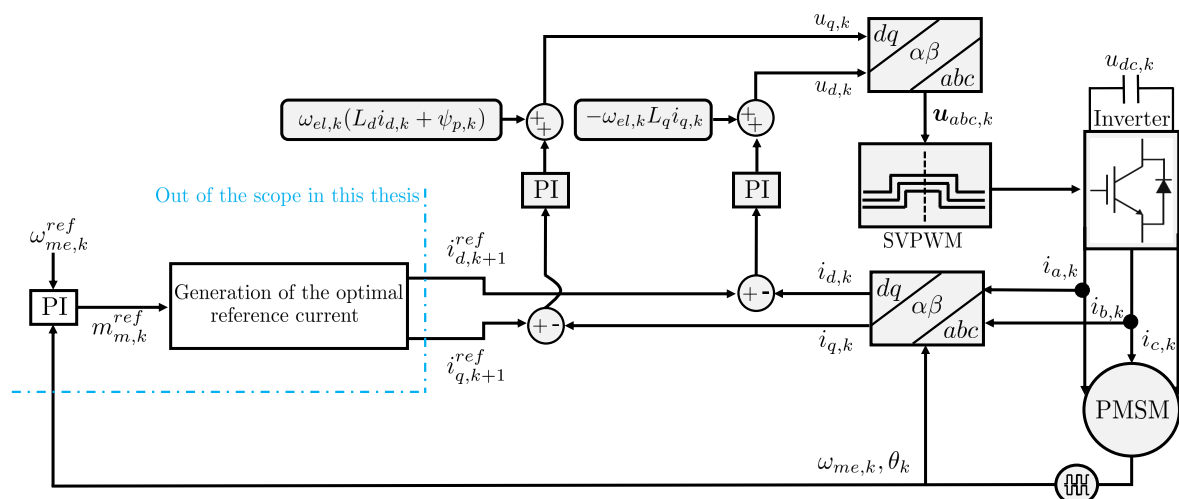
**Table B.3:** Parameters of the DFIM shown in Fig. B.2.

Name	Value
dc-link voltage	560 V
Stator line-line voltage	400 V
Power	10 kW
Rated speed	157 rad s <sup>-1</sup>
Stator resistance	0.72 $\Omega$
Rotor resistance	0.55 $\Omega$
Stator inductance	73.5 mH
Rotor inductance	86 mH
Mutual inductance	60 mH
Sampling frequency	8 kHz

## APPENDIX C

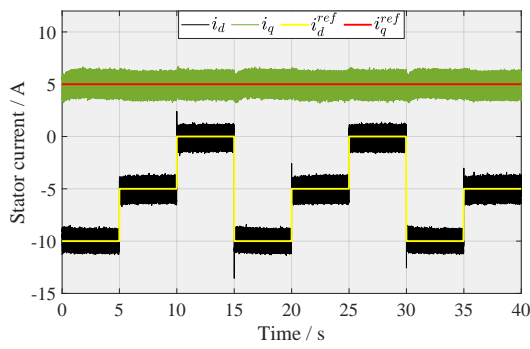
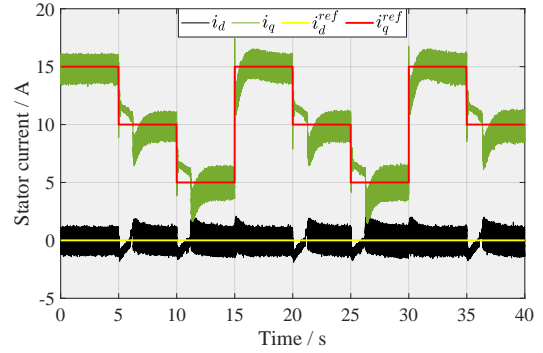
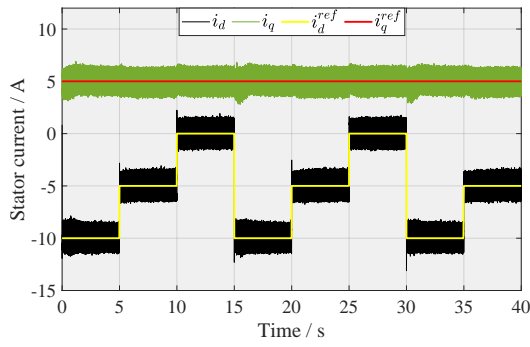
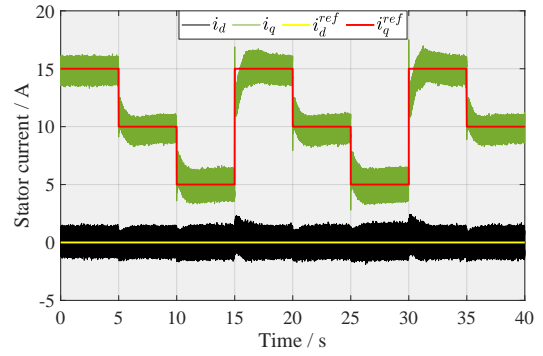
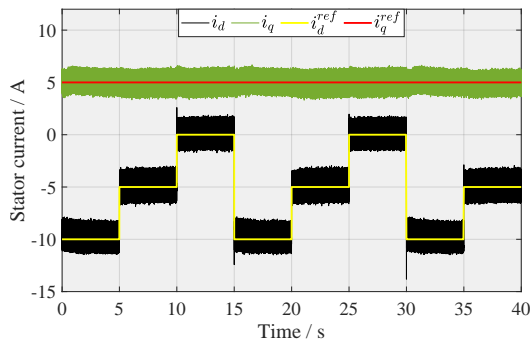
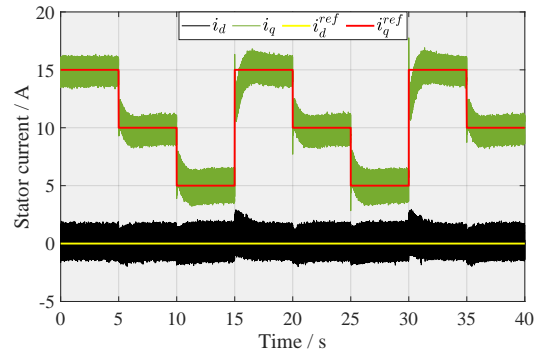
### Classical PI-FOC

In this appendix, the state-of-the-art classical field oriented control (FOC) with proportional-integral controllers is briefly presented, and results for the test scenarios used to validate the different predictive controllers throughout the dissertation are obtained here with the PI-FOC scheme for benchmarking. This control is based on the instantaneous decoupling of the direct and quadrature components of the electrical machine's current in order to permit the separate control of the torque and flux of the machine. Often, it is combined with a feedforward decoupling network to allow the independent current control of each axis in the  $d-q$  reference frame [186]. The schematic diagram of the classical PI-FOC for the current control of synchronous machines is depicted in Fig. C.1.

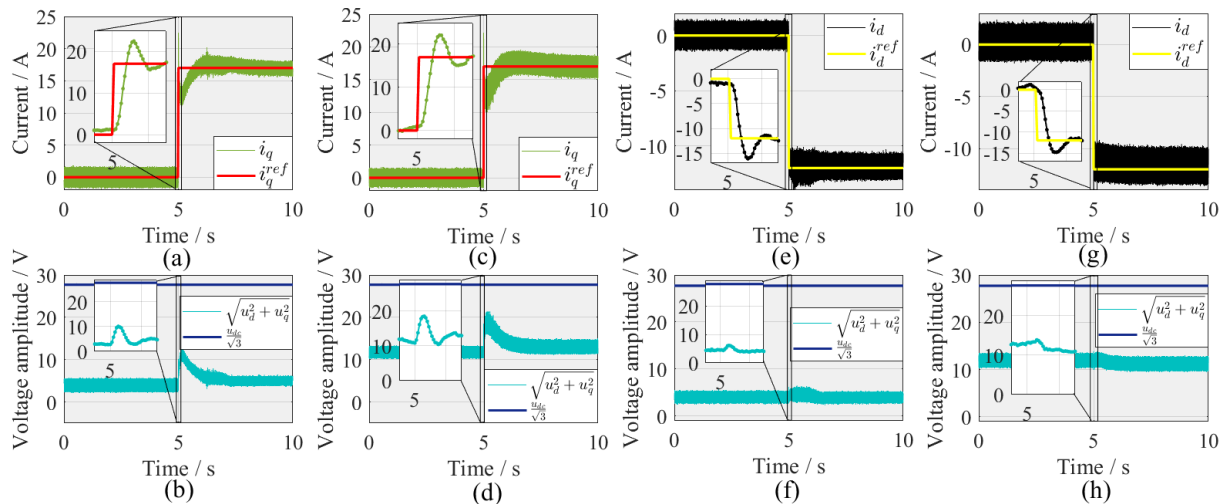


**Figure C.1:** Indirect proportional-integral field-oriented current controller for synchronous machines.

The steady-state current tracking results of both axes using the classical PI-FOC scheme are shown for the three mechanical speeds of 250 rpm, 500 rpm, and 1000 rpm in Fig. C.2, Fig. C.3, and Fig. C.4, respectively, as used as a steady-state test scenario with the predictive controllers. Obviously, the currents do suffer from coupling and overshoots.

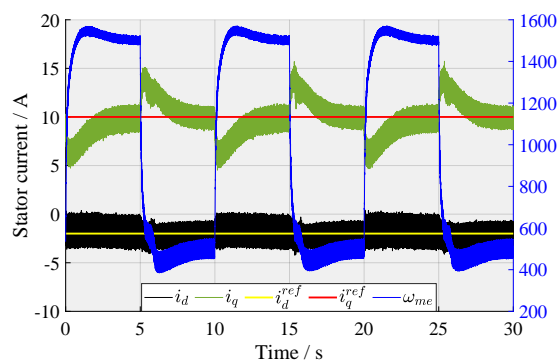
(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure C.2:** Experimental steady-state results of the PI-FOC at 250 rpm.(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure C.3:** Experimental steady-state results of the PI-FOC at 500 rpm.(a) Step changes of the  $d$  – axis current.(b) Step changes of the  $q$  – axis current.**Figure C.4:** Experimental steady-state results of the PI-FOC at 1000 rpm.

For benchmarking, the high-dynamic test of stepping  $i_q^{ref}$  from 1 A to 17 A and  $i_d^{ref}$  from 0 A to  $-12$  A at two different speeds of 500 rpm and 1500 rpm is repeated while the machine is controlled via the classical PI-FOC scheme. The tracking results with the corresponding voltages are shown in Fig. C.5.



**Figure C.5:** Dynamic response using the classical PI-FOC scheme for a step-up change of  $i_q^{ref}$  from 1 A to 17 A at 500 rpm is shown in (a) with the stator voltage amplitude and its constraint shown for that test in (b), the same test was repeated at 1500 rpm and the results are shown in (c) and (d), a step-down change of  $i_d^{ref}$  from 0 A to  $-12$  A at 500 rpm is shown in (e) with its stator voltage in (f), and in (g) and (h) the same results are shown for the same step change at 1500 rpm.

Finally, the classical PI-FOC was tested to maintain constant reference currents while changing the speed via the load motor, and the results are shown in Fig. C.6. The decoupling can be better tuned by using the identified inductances instead of the ones mentioned in the motor data-sheet.



**Figure C.6:** Reaction of the classical PI-FOC for constant current control against speed variation.





---

## List of Figures


---

1.1	The general concept of discrete-time model predictive control with a prediction horizon length $N$ . . . . .	2
1.2	(a) Typical cascaded control structure of an electrical machine (b) cascade-free alternative using model-based predictive control. . . . .	5
2.1	Coordinate systems: (a) the $abc$ reference frame, (b) the stationary $\alpha - \beta$ frame, (c) the rotating $d - q$ frame. . . . .	15
2.2	An example of a current vector in a typical steady-state operation: (a) in the $abc$ frame, (b) the transformed current in the stationary $\alpha - \beta$ frame, (c) the equivalent current in the rotating $d - q$ frame, (d) one cycle of the rotating vector in the $\alpha - \beta$ plane. . . . .	15
2.3	2-levels VSI and its output VVs in the stationary $\alpha - \beta$ plane with an illustration of the symmetrical SVM. . . . .	17
2.4	Direct and indirect control structure for electrical drives: (a) direct control, (b) indirect control, with $r$ as the reference vector. . . . .	18
2.5	Example illustrating the difference between measuring the current only once at the middle of each symmetrical pulsation SVM period and at each switching transition within the sampling period assuming an ideal inverter. . . . .	18
2.6	Standstill test results to investigate the VSI nonlinearity effect: (a)-(c) ascending reference profile to one phase, descending reference profile to the other while keeping the third phase at 0 V, (d) fine grid of current vectors in the complete $\alpha - \beta$ plane. . . . .	20
2.7	VSI missing voltage curves as a function of the phase currents. . . . .	20
2.8	Validation of the feedforward compensation of the VSI non-ideal behaviour at standstill: (a)-(c) two-phase configuration test results, (d) fine grid test results in the $\alpha - \beta$ plane. . . . .	21
2.9	Spiral and circular validation tests of the feedforward state-of-the-art compensation method of a 2-levels VSI: (a)-(c) spiral reference trajectory, (d) circular reference current trajectories with different amplitudes. . . . .	22

2.10	Injecting zero current while increasing the shaft speed via the load machine: (a) stator voltages and their references and the measured speed without ADC, (b) stator voltages and their references and the measured speed with ADC, (c) and (d) contains the stator current without and with ADC, respectively. . . . .	23
2.11	Voltage and current measurement synchronization: (a) correction of the initial states for control, (b) correction of the control inputs, which is useful in online parameters estimation or while learning the system dynamics. . . . .	24
2.12	The identified and smoothed nonlinear model parameters. . . . .	27
2.13	Experimental validation results of the nonlinear physics-based model at 250 rpm. . . . .	29
2.14	Experimental validation results of the nonlinear physics-based model at 500 rpm. . . . .	29
2.15	Experimental validation results of the nonlinear physics-based model at 1000 rpm. . . . .	29
3.1	Correction phase voltages per the individual phase current. . . . .	33
3.2	Correction duty cycles per the individual phase current. . . . .	33
3.3	Correction phase voltages as functions of the three-phase currents. . . . .	34
3.4	Correction phase duty cycles as functions of the three-phase currents. . . . .	34
3.5	Correction phase duty cycles as functions of the $\alpha - \beta$ currents. . . . .	34
3.6	Uni-output NN architecture to fit the function of the inverter nonlinearity compensation. . . . .	35
3.7	Bi-output NN architecture to fit the function of the inverter nonlinearity compensation. . . . .	35
3.8	Missing $\alpha - axis$ voltage as a function of $i_{\alpha\beta}$ . . . . .	36
3.9	Missing $\beta - axis$ voltage as a function of $i_{\alpha\beta}$ . . . . .	36
3.10	The setup of generating the excitation data set for the model identification task. . . . .	38
3.11	Sketch to depict the reference excitation signals of the motor to be identified. . . . .	39
3.12	Sketch to illustrate the resulting processed data set in an input-output form, to be used for the data-driven modelling. . . . .	39
3.13	The architecture of the LSTM neural model. . . . .	41
3.14	The execution time of the LSTM NN on the dSPACE MicroAutobox II platform. . . . .	41
3.15	Experimental validation results of the LSTM model at 250 rpm. . . . .	42
3.16	Experimental validation results of the LSTM model at 500 rpm. . . . .	42
3.17	Experimental validation results of the LSTM model at 1000 rpm. . . . .	42
3.18	Validation results of the approximate GP model at 250 rpm. . . . .	45
3.19	Validation results of the approximate GP model at 500 rpm. . . . .	45
3.20	Validation results of the approximate GP model at 1000 rpm. . . . .	45
4.1	Visualization of the operation concept of the one-step FCS-MPC for an example of current tracking of synchronous machines. . . . .	54
4.2	Conventional FCS-MPCC scheme for synchronous machines. . . . .	55
4.3	Example of locating the reference VV in the $\alpha - \beta$ frame. . . . .	57
4.4	The Proposed computationally-efficient FCS-MPCC for synchronous machines. . . . .	58
4.5	Experimental results of the conventional and computationally-efficient FCS-MPCC schemes applied to control the SMPMSM in Appendix B.2. . . . .	59
4.6	Experimental results of the computationally-efficient FCS-MPCC scheme with applying null vector applied to the SMPMSM in Appendix B.2. . . . .	60

4.7	Required computations per each sampling period as a function of the prediction horizon $N$ for FCS-MPC schemes for a system with two states. . . . .	61
4.8	Steady-state comparison for one cycle of three FCS-MPC controllers with different prediction horizons of: (a) $N = 1$ , (b) $N = 5$ , (c) $N = 8$ . . . . .	62
4.9	Training a NN to mimic the $N = 5$ FCS-MPC controller. . . . .	64
4.10	The proposed long-horizon FCS-MPC control scheme based on NNs. . . . .	65
4.11	The confusion matrix of the proposed NN with the architecture of $9 - 25 - 8$ neurons that aims to mimic the long-horizon FCS-MPC with $N = 5$ at $f_{sw,avg} = 3.5$ kHz. . . . .	67
4.12	The THD Vs. $f_{sw,avg}$ curve of $N = 1$ and $N = 5$ FCS-MPC schemes, and NN controllers that have learned the long-horizon FCS-MPC with $N = 5$ at different $f_{sw,avg}$ . . . . .	67
4.13	Simulative results comparing the steady-state and the dynamic performance of the proposed NN current controller with the long-horizon FCS-MPC controller with $N = 5$ that was used to generate its training data set, including steady-state operation, torque step change, torque dynamics, and a test of the current constraints fulfilment. . . . .	69
4.14	An example to illustrate the conceptual difference in transients and steady-state ripple between FCS-MPC and CCS-MPC applied for the current control of a synchronous machine [57]. . . . .	71
4.15	Indirect model-based predictive current controller for synchronous machines. . . . .	73
4.16	Step current changes and speed variation test using CCS-MPC with an LTI model, with and without inverter nonlinearity compensation. . . . .	77
4.17	The measured execution time $T_{ex}$ of the CCS-MPC scheme with $N = 2$ measured on the dSPACE DS1007 platform of the SMPMSM test bench. . . . .	78
4.18	Experimental validation of the CCS-MPC scheme applied on the SMPMSM with an LTI first-principles model: (a) tracking results without any integral action functionality, (b) tracking results with an additive discrete-time integral action term, (c) transient of the torque-producing current $i_q^{ref}$ from 0 A to 15 A at $\omega_{me} = 120$ rad s <sup>-1</sup> without integral action with its corresponding voltage constraint fulfilment shown in (d), and in (e) and (f) the same step change was made with the additive discrete-time integral action. . . . .	78
4.19	An overview about the different model mismatch compensation approaches typically used within linear MPC frameworks in electrical drives, and their integration with CCS-MPC, DB-MPC, and FCS-MPC: (a) additive integrating control input, (b) $\delta u$ formulation, (c) state-space model augmentation, (d) persistent step disturbance compensation, and (e) online parameters estimation. . . . .	81
4.20	The proposed offset-free CCS-MPC for current control of synchronous machines. . . . .	82
4.21	Different current tracking steps using the proposed offset-free linear CCS-MPC scheme. . . . .	83
4.22	Steady current tracking with load speed variation using the proposed offset-free linear CCS-MPC scheme. . . . .	83

4.23	Performance comparison under a step change of $i_q^{ref}$ from 3 A to 7 A at a fixed $i_d^{ref}$ current of 2 A and $\omega_{me}$ was kept at 500 rpm: (a) and (c) belong to the proposed offset-free linear CCS-MPC, and the results in (b) and (d) belong to the classical PI-FOC for comparison. . . . .	83
4.24	Experimental current tracking steady-state results of the NL-FP-CCS-MPC at 250 rpm. . . . .	85
4.25	Experimental current tracking steady-state results of the NL-FP-CCS-MPC at 500 rpm. . . . .	85
4.26	Experimental current tracking steady-state results of the NL-FP-CCS-MPC at 1000 rpm. . . . .	85
4.27	High dynamic test of the IPMSM using the NL-FP-CCS-MPC scheme. . . . .	86
4.28	Steady current tracking with load speed variation using the NL-FP-CCS-MPC scheme. . . . .	86
4.29	Current constraints fulfilment using the NL-FP-CCS-MPC. . . . .	87
4.30	Experimental current tracking steady-state results of the linear DB-MPC at 250 rpm. . . . .	88
4.31	Experimental current tracking steady-state results of the linear DB-MPC at 500 rpm. . . . .	88
4.32	Experimental current tracking steady-state results of the linear DB-MPC at 1000 rpm. . . . .	88
4.33	Experimental current tracking steady-state results of the nonlinear DB-MPC at 250 rpm. . . . .	89
4.34	Experimental current tracking steady-state results of the nonlinear DB-MPC at 500 rpm. . . . .	89
4.35	Experimental current tracking steady-state results of the nonlinear DB-MPC at 1000 rpm. . . . .	89
4.36	Experimental comparison in transient states while using the LTI and the nonlinear models within DB-MPC scheme. . . . .	90
5.1	Sketch illustrating the use of NNs either as a model of the current dynamics of synchronous machines (on the left), or as a current controller (on the right) with the corresponding inputs and outputs. . . . .	92
5.2	Indirect data-driven predictive current controller for synchronous machines. . .	93
5.3	Experimental steady-state tracking results of the nonlinear LB-CCS-MPC at 250 rpm. . . . .	94
5.4	Experimental steady-state tracking results of the nonlinear LB-CCS-MPC at 500 rpm. . . . .	94
5.5	Experimental steady-state tracking results of the nonlinear LB-CCS-MPC at 1000 rpm. . . . .	94

5.6	Dynamic response using the proposed LB-CCS-MPC scheme for a step-up change of $i_q^{ref}$ from 0 A to 17 A at 500 rpm is shown in (a) with the stator voltage amplitude and its constraint shown for that test in (b), the same test was repeated at 1000 rpm and the results are shown in (c) and (d), a step-down change of $i_d^{ref}$ from 0 A to $-12$ A at 500 rpm is shown in (e) with its stator voltage in (f), and in (g) and (h) the same results are shown for the same step change at 1000 rpm. . . . .	95
5.7	Steady current tracking with load speed variation using the nonlinear LB-CCS-MPC scheme. . . . .	95
5.8	The measured execution time $T_{ex}$ of the proposed LB-CCS-MPC scheme, measured on the dSPACE MicroAutoBox II platform of the IPMSM test bench. . . . .	95
5.9	Benchmarking of the proposed nonlinear LB-CCS-MPC against the L-FP-CCS-MPC and the NL-FP-CCS-MPC schemes in steady-state current tracking operation at 500 rpm: (a) $i_q^{ref}$ at 5 A and $i_d^{ref}$ is stepped up and down between $-10$ A to 0 A with 5 A increments using the L-FP-CCS-MPC scheme, (b) $i_d^{ref}$ at 5 A and $i_q^{ref}$ is stepped up and down between 15 A to 5 A with 5 A increments using the L-FP-CCS-MPC scheme, (c) and (d) are for the same test scenario using the NL-FP-CCS-MPC scheme, (e) and (f) are obtained using the proposed LB-CCS-MPC scheme with the LSTM neural model. 	96
5.10	Comparison of the proposed LB-CCS-MPC scheme (in green) with the L-FP-CCS-MPC (in orange) and the NL-FP-CCS-MPC (in black) in transients at 750 rpm: (a) an ascending step in $i_d^{ref}$ , (b) a descending step in $i_d^{ref}$ , (c) an ascending step in $i_q^{ref}$ , (d) a descending step in $i_q^{ref}$ . . . . .	97
B.1	IPMSM test bench: (1) main motor, (2) load machine, (3) torque sensor, (4) dSPACE MicroAutoBox II 1513/1514, (5) 2-levels VSI with the phase currents and dc-link voltage sensors and their corresponding measurements board, (6) Speed regulator for the load machine from Ott GmbH. . . . .	118
B.2	SMPMSM test bench: (1) main motor (SMPMSM), (2) load machine (DFIM), (3) torque sensor, (4) dSPACE DS1007 platform, (5) 2-levels VSI from SEW-Eurodrive GmbH & Co KG. . . . .	119
C.1	Indirect proportional-integral field-oriented current controller for synchronous machines. . . . .	121
C.2	Experimental steady-state results of the PI-FOC at 250 rpm. . . . .	122
C.3	Experimental steady-state results of the PI-FOC at 500 rpm. . . . .	122
C.4	Experimental steady-state results of the PI-FOC at 1000 rpm. . . . .	122
C.5	Dynamic response using the classical PI-FOC scheme for a step-up change of $i_q^{ref}$ from 1 A to 17 A at 500 rpm is shown in (a) with the stator voltage amplitude and its constraint shown for that test in (b), the same test was repeated at 1500 rpm and the results are shown in (c) and (d), a step-down change of $i_d^{ref}$ from 0 A to $-12$ A at 500 rpm is shown in (e) with its stator voltage in (f), and in (g) and (h) the same results are shown for the same step change at 1500 rpm. . . . .	123

C.6 Reaction of the classical PI-FOC for constant current control against speed variation. . . . .	123
--	-----

---

## List of Tables

---

2.1	Output VVs of a 2-levels VSI in the three-phase $abc$ reference frame. . . . .	17
2.2	Output VVs of a 2-levels inverter in the stationary $\alpha - \beta$ reference frame. . . . .	17
4.1	Parameters of the IPMSM used for the simulation study in this section. . . . .	62
4.2	Resultant THD and $f_{sw,avg}$ of FCS-MPC controllers with different prediction horizons. . . . .	63
4.3	Classification of the discrete VVs of a 2-levels inverter with one-hot encoding. . . . .	64
4.4	Comparison of the computational demand of FCS-MPC controllers with prediction horizons from $N = 1$ to $N = 5$ steps with the proposed NN control scheme with an architecture of $9 - 25 - 8$ using a sigmoid activation function on the hidden layer and a softmax activation function on the output layer, tested on dSpace MicroAutoBox II 1513/1514 platform. . . . .	68
B.1	Parameters of the IPMSMs shown in Fig. B.1. . . . .	118
B.2	Parameters of the SMPMSM shown in Fig. B.2. . . . .	119
B.3	Parameters of the DFIM shown in Fig. B.2. . . . .	120





---

## Bibliography

---

- [1] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109899002149>
- [2] E. Camacho and C. Bordons, *Model Predictive Control*, 01 2004, vol. 13.
- [3] J. Li, X.-y. Huang, F. Niu, C. You, L. Wu, and Y. Fang, "Prediction error analysis of finite-control-set model predictive current control for ipmsms," *Energies*, vol. 11, p. 2051, 08 2018.
- [4] J. Ramirez-Hernandez, O. U. Juarez-Sandoval, L. Hernandez-Gonzalez, D. Cortes, J. C. Sanchez-Garcia, and P. Guevara-Lopez, "Performance comparison of numerical methods in a predictive controller for an ac-dc power converter," *Mathematics*, vol. 10, no. 15, 2022. [Online]. Available: <https://www.mdpi.com/2227-7390/10/15/2818>
- [5] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model algorithmic control of industrial processes," *IFAC Proceedings Volumes*, vol. 10, no. 16, pp. 103–120, 1977, preprints of the 5th IFAC/IFIP International Conference on Digital Computer Applications to Process Control, The Hague, The Netherlands, 14-17 June, 1977. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017695132>
- [6] J. Richalet, A. Rault, J.L. Testud and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109878900018>
- [7] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control - a computer control algorithm," *IEEE Transactions on Automatic Control*, vol. 17, p. 72, 1979.
- [8] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control - a computer control algorithm," *Joint Automatic Control Conference*, vol. 17, p. 72, 1980.
- [9] C. E. Garcia and M. Morari, "Internal model control. a unifying review and some new results," *Industrial & Engineering Chemistry Process Design and Development*, vol. 21, no. 2, pp. 308–323, 1982. [Online]. Available: <https://doi.org/10.1021/i200017a016>

- [10] D. Clarke, C. Mohtadi, and P. Tuffs, "Generalized predictive control—part i. the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0005109887900872>
- [11] D. Clarke, "Application of generalized predictive control to industrial processes," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 49–55, 1988.
- [12] J. Ortega and E. Camacho, "Mobile robot navigation in a partially structured static environment, using neural predictive control," *Control Engineering Practice*, vol. 4, no. 12, pp. 1669–1679, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0967066196001840>
- [13] S. Yelneedi, S. Lakshminarayanan, and G. Rangaiah, *Automatic regulation of anesthesia by simultaneous administration of two anesthetic drugs using model predictive control*, 01 2007, vol. 14, pp. 82–86.
- [14] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0967066102001867>
- [15] T. Geyer, *Model Predictive Control of High Power Converters and Industrial Drives*, 09 2016.
- [16] A. Linder, R. Kanchan, P. Stolze, and R. Kennel, *Model-Based Predictive Control of Electric Drives*, 07 2010.
- [17] R. Kennel and A. Linder, "Predictive control of inverter supplied electrical drives," in *2000 IEEE 31st Annual Power Electronics Specialists Conference. Conference Proceedings (Cat. No.00CH37018)*, vol. 2, 2000, pp. 761–766 vol.2.
- [18] R. Kennel, A. Linder, and M. Linke, "Generalized predictive control (gpc)-ready for use in drive applications?" in *2001 IEEE 32nd Annual Power Electronics Specialists Conference (IEEE Cat. No.01CH37230)*, vol. 4, 2001, pp. 1839–1844 vol. 4.
- [19] A. Linder and R. Kennel, "Model predictive control for electrical drives," in *2005 IEEE 36th Power Electronics Specialists Conference*, 2005, pp. 1793–1799.
- [20] D. Quevedo, R. Aguilera, and T. Geyer, *Predictive Control in Power Electronics and Drives: Basic Concepts, Theory, and Methods*, 01 2014, vol. 531, pp. 181–226.
- [21] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, "Model predictive control—a simple and powerful method to control power converters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1826–1838, 2009.
- [22] I. Takahashi and T. Noguchi, "A new quick-response and high-efficiency control strategy of an induction motor," *IEEE Transactions on Industry Applications*, vol. IA-22, no. 5, pp. 820–827, 1986.
- [23] I. Takahashi and Y. Ohmori, "High-performance direct torque control of an induction motor," *IEEE Transactions on Industry Applications*, vol. 25, no. 2, pp. 257–264, 1989.

- [24] M. Kroneisl, V. Šmídl, Z. Peroutka, and M. Janda, “Predictive control of im drive acoustic noise,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5666–5676, 2020.
- [25] A. Andersson and T. Thiringer, “Assessment of an improved finite control set model predictive current controller for automotive propulsion applications,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 91–100, 2020.
- [26] X. Wu, Z. Q. Zhu, and N. M. A. Freire, “High frequency signal injection sensorless control of finite-control-set model predictive control with deadbeat solution,” in *2021 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2021, pp. 4698–4703.
- [27] 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)*, 2017, pp. 89–94.
- [28] C. Hackl, F. Larcher, A. Dötlinger, and R. Kennel, “Is multiple-objective model-predictive control “optimal”?” in *2013 IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE)*, 2013, pp. 1–8.
- [29] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
- [30] J. Holtz and S. Stadtfeld, “A predictive controller for the stator current vector of ac machines fed from a switched voltage source,” in *IEEE International Power Electronics Conference*, 1983, pp. 1665–1675.
- [31] J. Holtz and S. Stadtfeld, “Field-oriented control by forced motor currents in a voltage fed inverter drive,” *IFAC Proceedings Volumes*, vol. 16, no. 16, pp. 103–110, 1983, 3rd IFAC Symposium on Control in Power Electronics and Electrical Drives, Lausanne, Switzerland, 12-14 September, 1983. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S147466701761855X>
- [32] R. Kennel, “Prädiktives führungsverfahren für stromrichter: Herleitung und realisierung an mikrocomputergeführten netzgeführten stromrichtern,” Ph.D. dissertation, Technical University of Kaiserslautern, 1984.
- [33] R. Kennel and D. Schröder, “Predictive control strategy for converters,” in *Control in Power Electronics and Electrical Drives 1983*, ser. IFAC Symposia Series, R. ZWICKY, Ed. Oxford: Pergamon, 1984, pp. 415–422. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780080305363500611>
- [34] D. Schröder and R. Kennel, “Model-control promc—a new control strategy with microcomputer for drive applications,” *IEEE Transactions on Industry Applications*, vol. IA-21, no. 5, pp. 1162–1167, 1985.
- [35] M. Depenbrock, “Direct self-control (dsc) of inverter-fed induction machine,” *IEEE Transactions on Power Electronics*, vol. 3, no. 4, pp. 420–429, 1988.

- [36] T. Geyer, G. Papafotiou, and M. Morari, "On the optimal control of switch-mode dc-dc converters," in *Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 342–356.
- [37] J. Rodriguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortes, and U. Ammann, "Predictive current control of a voltage source inverter," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 495–503, 2007.
- [38] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive control in power electronics and drives," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4312–4324, 2008.
- [39] J. Rodriguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young, and C. A. Rojas, "State of the art of finite control set model predictive control in power electronics," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 1003–1016, 2013.
- [40] P. Cortes, J. Rodriguez, P. Antoniewicz, and M. Kazmierkowski, "Direct power control of an afe using predictive control," *IEEE Transactions on Power Electronics*, vol. 23, no. 5, pp. 2516–2523, 2008.
- [41] J. Rodriguez and P. Cortes, *Predictive control of power converters and electrical drives*. John Wiley & Sons, 2012, vol. 40.
- [42] L. Wang, S. Chai, D. Yoo, L. Gan, and K. Ng, *PID and predictive control of electrical drives and power converters using MATLAB/Simulink*. John Wiley & Sons, 2015.
- [43] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model predictive control of power electronic systems: Methods, results, and challenges," *IEEE Open Journal of Industry Applications*, vol. 1, pp. 95–114, 2020.
- [44] K. A. Islam, M. Abdelrahem, and R. Kennel, "Efficient finite control set-model predictive control for grid-connected photovoltaic inverters," in *2016 International Symposium on Industrial Electronics (INDEL)*, 2016, pp. 1–6.
- [45] I. Hammoud, K. Morsy, M. Abdelrahem, and R. Kennel, "Efficient model predictive power control with online inductance estimation for photovoltaic inverters," *Electrical Engineering*, vol. 102, no. 2, pp. 549–562, 2020.
- [46] I. Hammoud, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, "Computationally efficient finite-set model predictive current control of interior permanent magnet synchronous motors with model-based online inductance estimation," in *2019 IEEE Conference on Power Electronics and Renewable Energy (CPERE)*, 2019, pp. 290–295.
- [47] I. Hammoud, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, "Long-horizon direct model predictive control based on neural networks for electrical drives," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 3057–3064.

- [48] P. Karamanakos, T. Geyer, N. Oikonomou, F. D. Kieferndorf, and S. Manias, "Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 32–43, 2014.
- [49] T. Geyer and D. E. Quevedo, "Multistep finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6836–6846, 2014.
- [50] M. Dorfling, H. Mouton, P. Karamanakos, and T. Geyer, "Experimental evaluation of sphere decoding for long-horizon direct model predictive control," in *2017 19th European Conference on Power Electronics and Applications (EPE'17 ECCE Europe)*, 2017, pp. P.1–P.10.
- [51] T. Dorfling, H. du Toit Mouton, T. Geyer, and P. Karamanakos, "Long-horizon finite-control-set model predictive control with nonrecursive sphere decoding on an fpga," *IEEE Transactions on Power Electronics*, vol. 35, no. 7, pp. 7520–7531, 2020.
- [52] E. Liegmann, P. Karamanakos, and R. Kennel, "Real-time implementation of long-horizon direct model predictive control on an embedded system," *IEEE Open Journal of Industry Applications*, vol. 3, pp. 1–12, 2022.
- [53] S. Richter, S. Mariéthoz, and M. Morari, "High-speed online mpc based on a fast gradient method applied to power converter control," in *Proceedings of the 2010 American Control Conference*, 2010, pp. 4737–4743.
- [54] T. Englert and K. Graichen, "Nonlinear model predictive torque control of pmsms for high performance applications," *Control Engineering Practice*, vol. 81, pp. 43–54, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0967066118304854>
- [55] A. Zanelli, J. Kullick, H. M. Eldeeb, G. Frison, C. M. Hackl, and M. Diehl, "Continuous control set nonlinear model predictive control of reluctance synchronous machines," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 130–141, 2022.
- [56] I. Hammoud, K. Xu, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, "On offset-free continuous model predictive current control of permanent magnet synchronous motors," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6662–6669, 2020, 21st IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S240589632030344X>
- [57] I. Hammoud, S. Hentzelt, K. Xu, T. Oehlschlaegel, M. Abdelrahem, C. M. Hackl, and R. Kennel, "On continuous set model predictive control of permanent magnet synchronous machines," *IEEE Transactions on Power Electronics*, pp. 1–1, 2022.
- [58] P. Cortes, S. Kouro, B. La Rocca, R. Vargas, J. Rodriguez, J. I. Leon, S. Vazquez, and L. G. Franquelo, "Guidelines for weighting factors design in model predictive control of power converters and drives," in *2009 IEEE International Conference on Industrial Technology*, 2009, pp. 1–7.

- [59] C. A. Rojas, J. Rodriguez, F. Villarroel, J. R. Espinoza, C. A. Silva, and M. Trincado, "Predictive torque and flux control without weighting factors," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 681–690, 2013.
- [60] T. Geyer, "Algebraic weighting factor selection for predictive torque and flux control," in *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2017, pp. 357–364.
- [61] M. Norambuena, J. Rodriguez, Z. Zhang, F. Wang, C. Garcia, and R. Kennel, "A very simple strategy for high-quality performance of ac machines using model predictive control," *IEEE Transactions on Power Electronics*, vol. 34, no. 1, pp. 794–800, 2019.
- [62] S. A. Davari, M. Norambuena, V. Nekoukar, C. Garcia, and J. Rodriguez, "Even-handed sequential predictive torque and flux control," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 7334–7342, 2020.
- [63] T. Dragičević and M. Novak, "Weighting factor design in model predictive control of power electronic converters: An artificial neural network approach," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8870–8880, 2019.
- [64] M. Novak, H. Xie, T. Dragicevic, F. Wang, J. Rodriguez, and F. Blaabjerg, "Optimal cost function parameter design in predictive torque control (ptc) using artificial neural networks (ann)," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7309–7319, 2021.
- [65] O. Babayomi, Y. Zhang, yu li, Y. Wang, Z. Li, and Z. Zhang, "A Comparative Study on Weighting Factor Design Techniques for the Model Predictive Control of Power Electronics and Energy Conversion Systems," 7 2021. [Online]. Available: [https://www.techrxiv.org/articles/preprint/A\\_Comparative\\_Study\\_on\\_Weighting\\_Factor\\_Design\\_Techniques\\_for\\_the\\_Model\\_Predictive\\_Control\\_of\\_Power\\_Electronics\\_and\\_Energy\\_Conversion\\_Systems/15022236](https://www.techrxiv.org/articles/preprint/A_Comparative_Study_on_Weighting_Factor_Design_Techniques_for_the_Model_Predictive_Control_of_Power_Electronics_and_Energy_Conversion_Systems/15022236)
- [66] R. P. Aguilera and D. E. Quevedo, "On stability and performance of finite control set mpc for power converters," in *2011 Workshop on Predictive Control of Electrical Drives and Power Electronics*, 2011, pp. 55–62.
- [67] R. P. Aguilera, and D. E. Quevedo, "Predictive control of power converters: Designs with guaranteed performance," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 1, pp. 53–63, 2015.
- [68] I. Hammoud, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, "Modelling for nonlinear predictive control of synchronous machines: First principles vs. data-driven approaches," in *2021 IEEE International Conference on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 2021, pp. 715–724.
- [69] I. Hammoud, S. Hentzelt, T. Oehlschlaegel, and R. Kennel, "Learning-based model predictive current control for synchronous machines: An lstm approach," *European Journal of Control*, vol. 68, p. 100663, 2022, 2022 European Control Conference Special Issue. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0947358022000553>

- [70] C. J. O'Rourke, M. M. Qasim, M. R. Overlin, and J. L. Kirtley, "A geometric interpretation of reference frames and transformations: dq0, clarke, and park," *IEEE Transactions on Energy Conversion*, vol. 34, no. 4, pp. 2070–2083, 2019.
- [71] C. Hackl, *Non-identifier Based Adaptive Control in Mechatronics*, 05 2017, vol. 466.
- [72] G. Shen, W. Yao, B. Chen, K. Wang, K. Lee, and Z. Lu, "Automeasurement of the inverter output voltage delay curve to compensate for inverter nonlinearity in sensorless motor drives," *IEEE Transactions on Power Electronics*, vol. 29, no. 10, pp. 5542–5553, 2014.
- [73] B.-H. Bae and S.-K. Sul, "A compensation method for time delay of full-digital synchronous frame current regulator of pwm ac drives," *IEEE Transactions on Industry Applications*, vol. 39, no. 3, pp. 802–810, 2003.
- [74] M. Diehl, H. J. Ferreanu, and N. Haverbeke, "Efficient numerical methods for nonlinear mpc and moving horizon estimation," *Lecture Notes in Control and Information Sciences*, pp. 391–417, 2009.
- [75] H. M. Calderón, I. Hammoud, T. Oehlschlägel, H. Werner, and R. Kennel, "Data-driven model predictive current control for synchronous machines: a koopman operator approach," in *2022 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, 2022, pp. 942–947.
- [76] S. L. Kellner, "Parameteridentifikation bei permanenterregten synchronmaschinen," Ph.D. dissertation, Friedrich–Alexander University Erlangen–Nürnberg, Germany, 2012.
- [77] L. L. Schumaker, *Spline functions: Computational methods*. Society for Industrial and Applied Mathematics, 2015.
- [78] S. L. Kellner and B. Piepenbreier, "General pmsm d,q-model using optimized interpolated absolute and differential inductance surfaces," in *2011 IEEE International Electric Machines Drives Conference (IEMDC)*, 2011, pp. 212–217.
- [79] S. Ebersberger and B. Piepenbreier, "Identification of differential inductances of permanent magnet synchronous machines using test current signal injection," in *International Symposium on Power Electronics Power Electronics, Electrical Drives, Automation and Motion*, 2012, pp. 1342–1347.
- [80] D. Geweth, A. Zanelli, G. Frison, U. Vollmer, and M. Diehl, "Field oriented economic model predictive control for permanent magnet synchronous motors\*\*this research was supported by the german federal ministry for economic affairs and energy (bmwi) via eco4wind (0324125b) and dyconpv (0324166b), and by dfg via research unit for 2401." *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9093–9099, 2020, 21st IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896320327956>

- [81] G. Cimini, D. Bernardini, S. Levijoki, and A. Bemporad, "Embedded model predictive control with certified real-time optimization for synchronous motors," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 893–900, 2021.
- [82] G. Cimini, D. Bernardini, A. Bemporad, and S. Levijoki, "Online model predictive torque control for permanent magnet synchronous motors," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, 2015, pp. 2308–2313.
- [83] L. Verkroost, J. Druant, H. Vansompeel, F. De Belie, and P. Sergeant, "Predictive current control vs. pi control for surface mounted permanent magnet machines," in *2018 XIII International Conference on Electrical Machines (ICEM)*, 2018, pp. 1663–1669.
- [84] P. Kakosimos and H. Abu-Rub, "Predictive speed control with short prediction horizon for permanent magnet synchronous motor drives," *IEEE Transactions on Power Electronics*, vol. 33, no. 3, pp. 2740–2750, 2018.
- [85] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: an engineering perspective," *The International Journal of Advanced Manufacturing Technology*, vol. 117, pp. 1–23, 11 2021.
- [86] U. Rosolia, X. Zhang, and F. Borrelli, "Data-driven predictive control for autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 259–286, 2018. [Online]. Available: <https://doi.org/10.1146/annurev-control-060117-105215>
- [87] Y. Zhang, Z. Wang, H. Wang, and F. Blaabjerg, "Artificial intelligence-aided thermal model considering cross-coupling effects," *IEEE Transactions on Power Electronics*, vol. 35, no. 10, pp. 9998–10 002, 2020.
- [88] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 4653–4660.
- [89] N. Spielberg, M. Brown, N. Kapania, J. Kegelman, and J. Gerdes, "Neural network vehicle models for high-performance automated driving," *Science Robotics*, vol. 4, p. eaaw1975, 03 2019.
- [90] L. C. Sousa and H. V. H. Ayala, "Nonlinear tire model approximation using machine learning for efficient model predictive control," *IEEE Access*, vol. 10, pp. 107 549–107 562, 2022.
- [91] J.-S. Wang and Y.-P. Chen, "A fully automated recurrent neural network for unknown dynamic system identification and control," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 6, pp. 1363–1372, 2006.
- [92] H. Dinh, S. Bhasin, and W. E. Dixon, "Dynamic neural network-based robust identification and control of a class of nonlinear systems," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5536–5541.



- [93] J. Gonzalez and W. Yu, “Non-linear system modeling using lstm neural networks,” *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 485–489, 2018, 2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318310814>
- [94] Y. Wang, “A new concept using lstm neural networks for dynamic system identification,” in *2017 American Control Conference (ACC)*, 2017, pp. 5324–5329.
- [95] O. P. Ogunmolu, X. Gu, S. B. Jiang, and N. R. Gans, “Nonlinear systems identification using deep dynamic neural networks,” *ArXiv*, vol. abs/1610.01439, 2016.
- [96] L. Hewing, J. Kabzan, and M. N. Zeilinger, “Cautious model predictive control using gaussian process regression,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2020.
- [97] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, “Learning-based model predictive control for autonomous racing,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.
- [98] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, “Data-driven mpc for quadrotors,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.
- [99] D. Bergmann, R. Geiselhart, and K. Graichen, “Modelling and control of a heavy-duty diesel engine gas path with gaussian process regression,” in *2019 18th European Control Conference (ECC)*, 2019, pp. 1207–1213.
- [100] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S000510981830133X>
- [101] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, “Data-driven control of soft robots using koopman operator theory,” *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 948–961, 2021.
- [102] M. Netto, “Robust identification, estimation, and control of electric power systems using the koopman operator-theoretic framework,” 2019.
- [103] H. Arbabi, M. Korda, and I. Mezić, “A data-driven koopman model predictive control framework for nonlinear flows,” *arXiv: Fluid Dynamics*, 2018.
- [104] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review*, vol. 65 6, pp. 386–408, 1958.
- [105] C. Wang and Z. Luo, “A review of the optimal design of neural networks based on fpga,” *Applied Sciences*, vol. 12, no. 21, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/21/10771>
- [106] F. Chollet *et al.*, “Keras,” <https://github.com/fchollet/keras>, 2015.

- [107] M. Abadi, *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [108] S. Lucia and B. Karg, “A deep learning-based approach to robust nonlinear model predictive control,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 511–516, 2018, 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318326958>
- [109] S. W. Chen, K. Saulnier, N. A. Atanasov, D. D. Lee, V. R. Kumar, G. J. Pappas, and M. Morari, “Approximating explicit model predictive control using constrained neural networks,” *2018 Annual American Control Conference (ACC)*, pp. 1520–1527, 2018.
- [110] R. Gençay and T. Liu, “Nonlinear modelling and prediction with feedforward and recurrent networks,” *Physica D: Nonlinear Phenomena*, vol. 108, no. 1, pp. 119–134, 1997. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016727899782009X>
- [111] S. Han, J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang, H. Yang, and W. J. Dally, “Ese: Efficient speech recognition engine with sparse lstm on fpga,” *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017.
- [112] V. Carbune, P. Gonnet, T. Deselaers, H. A. Rowley, A. N. Daryin, M. C. Lafarga, L.-L. Wang, D. Keysers, S. Feuz, and P. Gervais, “Fast multi-language lstm-based online handwriting recognition,” *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 23, pp. 89–102, 2020.
- [113] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.
- [114] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2005. [Online]. Available: <https://books.google.de/books?id=Tr34DwAAQBAJ>
- [115] D. Bergmann, K. Harder, J. Niemeyer, and K. Graichen, “Nonlinear mpc of a heavy-duty diesel engine with learning gaussian process regression,” *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 113–129, 2022.
- [116] D. J. C. Mackay, “Gaussian processes - a replacement for supervised neural networks?” 1997.
- [117] M. Deisenroth and J. W. Ng, “Distributed gaussian processes,” in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1481–1490. [Online]. Available: <https://proceedings.mlr.press/v37/deisenroth15.html>

- [118] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, “When gaussian process meets big data: A review of scalable gps,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4405–4423, 2020.
- [119] J. Hensman, N. Fusi, and N. Lawrence, “Gaussian processes for big data,” *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*, 09 2013.
- [120] J. Gardner, G. Pleiss, D. Bindel, K. Weinberger, and A. Wilson, “Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration,” 09 2018.
- [121] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” *Automatica*, vol. 93, pp. 149–160, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S000510981830133X>
- [122] C. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. Henningson, “Spectral analysis of nonlinear flows,” *Journal of Fluid Mechanics*, vol. 641, pp. 115 – 127, 12 2009.
- [123] I. Abraham, G. Torre, and T. Murphey, “Model-based control using koopman operators,” 09 2017.
- [124] P. J. SCHMID, “Dynamic mode decomposition of numerical and experimental data,” *Journal of Fluid Mechanics*, vol. 656, p. 5–28, 2010.
- [125] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data-driven approximation of the koopman operator: Extending dynamic mode decomposition,” *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2014.
- [126] S. L. Brunton, “Notes on koopman operator theory,” 2019.
- [127] P. Stolze, “Advanced finite-set model predictive control for power electronics and electrical drives,” Ph.D. dissertation, 01 2014.
- [128] P. Karamanakos, T. Geyer, and R. Kennel, “On the choice of norm in finite control set model predictive control,” *IEEE Transactions on Power Electronics*, vol. 33, no. 8, pp. 7105–7117, 2018.
- [129] P. Karamanakos and T. Geyer, “Guidelines for the design of finite control set model predictive controllers,” *IEEE Transactions on Power Electronics*, vol. 35, no. 7, pp. 7434–7450, 2020.
- [130] H. Eldeeb, C. Hackl, L. Horlbeck, and J. Kullick, “A unified theory for optimal feed-forward torque control of anisotropic synchronous machines,” *International Journal of Control*, vol. 91, 06 2017.
- [131] C. Hackl, J. Kullick, and N. Monzen, “Generic loss minimization for nonlinear synchronous machines by analytical computation of optimal reference currents considering copper and iron losses,” in *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, vol. 1, 2021, pp. 1348–1355.

- [132] A. Glac, V. Šmídl, and Z. Peroutka, "Optimal feedforward torque control of synchronous machines with time-varying parameters," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 613–618.
- [133] P. Cortes, J. Rodriguez, C. Silva, and A. Flores, "Delay compensation in model predictive current control of a three-phase inverter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1323–1325, 2012.
- [134] S.-K. Kim, H. S. Park, J. H. Han, and Y. I. Lee, "Stabilizing model predictive control for torque control of permanent magnet synchronous motor," in *Proceedings of the 33rd Chinese Control Conference*, 2014, pp. 7772–7777.
- [135] Z. Zhang, C. M. Hackl, and R. Kennel, "Computationally efficient dmpc for three-level npc back-to-back converters in wind turbine systems with pmsg," *IEEE Transactions on Power Electronics*, vol. 32, no. 10, pp. 8018–8034, 2017.
- [136] I. Hammoud, K. Morsy, M. Abdelrahem, and R. Kennel, "Computationally efficient model predictive direct power control with online finite set model inductance estimation technique for grid-connected photovoltaic inverters," in *2019 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 2019, pp. 1–6.
- [137] S. Wendel, A. Dietz, and R. Kennel, "Fpga based finite-set model predictive current control for small pmsm drives with efficient resource streaming," in *2017 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 2017, pp. 66–71.
- [138] M. Abdelrahem, C. M. Hackl, R. Kennel, and J. Rodríguez, "Efficient direct-model predictive control with discrete-time integral action for pmsgs," *IEEE Transactions on Energy Conversion*, vol. 34, no. 2, pp. 1063–1072, 2019.
- [139] Y. Zhang, D. Xu, J. Liu, S. Gao, and W. Xu, "Performance improvement of model-predictive current control of permanent magnet synchronous motor drives," *IEEE Transactions on Industry Applications*, vol. 53, no. 4, pp. 3683–3695, 2017.
- [140] Y. Zhang and H. Yang, "Model predictive torque control of induction motor drives with optimal duty cycle control," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6593–6603, 2014.
- [141] Y. Zhang, Y. Bai, and H. Yang, "A universal multiple-vector-based model predictive control of induction motor drives," *IEEE Transactions on Power Electronics*, vol. 33, no. 8, pp. 6957–6969, 2018.
- [142] Y. Zhang and H. Yang, "Two-vector-based model predictive torque control without weighting factors for induction motor drives," *IEEE Transactions on Power Electronics*, vol. 31, no. 2, pp. 1381–1390, 2016.

- [143] Y. Zhang and H. Yang, "Generalized two-vector-based model-predictive torque control of induction motor drives," *IEEE Transactions on Power Electronics*, vol. 30, no. 7, pp. 3818–3829, 2015.
- [144] G. Holmes, T. Lipo, and T. Lipo, *Pulse Width Modulation for Power Converters: Principles and Practice*, 01 2003.
- [145] 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)*, 2014, pp. 3520–3527.
- [146] C. Dirscherl, C. M. Hackl, and K. Schechner, "Explicit model predictive control with disturbance observer for grid-connected voltage source power converters," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, 2015, pp. 999–1006.
- [147] 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)*, 2017, pp. 89–94.
- [148] A. Grancharova and T. Johansen, *Explicit Nonlinear Model Predictive Control: Theory and Applications*, 03 2012, vol. 429.
- [149] S. Mariethoz and M. Morari, "Explicit model-predictive control of a pwm inverter with an lcl filter," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 389–399, 2009.
- [150] N. A. Ameen, B. S. Galal, R. M. Kennel, and R. S. Kanchan, "The polynomial approximation of the explicit solution of model-based predictive controller for drive applications," in *2011 Workshop on Predictive Control of Electrical Drives and Power Electronics*, 2011, pp. 76–81.
- [151] S. Lucia, D. Navarro, B. Karg, H. Sarnago, and O. Lucia, "Deep learning-based model predictive control for resonant power converters," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 409–420, 2021.
- [152] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 1520–1527.
- [153] M. Hertneck, J. Kohler, S. Trimpe, and F. Allgower, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, jul 2018. [Online]. Available: <https://doi.org/10.1109%2F1csys.2018.2843682>
- [154] A. H. Namin, K. Leboeuf, R. Muscedere, H. Wu, and M. Ahmadi, "Efficient hardware implementation of the hyperbolic tangent sigmoid function," in *2009 IEEE International Symposium on Circuits and Systems*, 2009, pp. 2117–2120.

- [155] M. Abu-Ali, F. Berkel, M. Manderla, S. Reimann, R. Kennel, and M. Abdelrahem, "Deep learning-based long-horizon mpc: Robust, high performing, and computationally efficient control for pmsm drives," *IEEE Transactions on Power Electronics*, vol. 37, no. 10, pp. 12 486–12 501, 2022.
- [156] M. Novak and F. Blaabjerg, "Supervised imitation learning of fs-mpc algorithm for multilevel converters," in *2021 23rd European Conference on Power Electronics and Applications (EPE'21 ECCE Europe)*, 2021, pp. P.1–P.10.
- [157] A. A. Ahmed, B. K. Koh, and Y. I. Lee, "Continuous control set-model predictive control for torque control of induction motors in a wide speed range," *Electric Power Components and Systems*, vol. 46, no. 19-20, pp. 2142–2158, 2018. [Online]. Available: <https://doi.org/10.1080/15325008.2018.1533602>
- [158] A. K. Rathore, J. Holtz, and T. Boller, "Synchronous optimal pulsewidth modulation for low-switching-frequency control of medium-voltage multilevel inverters," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2374–2381, 2010.
- [159] T. Geyer, N. Oikonomou, G. Papafotiou, and F. D. Kieferndorf, "Model predictive pulse pattern control," *IEEE Transactions on Industry Applications*, vol. 48, no. 2, pp. 663–676, 2012.
- [160] P. Karamanakos, P. Stolze, R. M. Kennel, S. Manias, and H. du Toit Mouton, "Variable switching point predictive torque control of induction machines," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 2, pp. 285–295, 2014.
- [161] L. Wang, T. Zhao, and J. He, "Investigation of variable switching frequency in finite control set model predictive control on grid-connected inverters," *IEEE Open Journal of Industry Applications*, vol. PP, pp. 1–1, 06 2021.
- [162] T. Englert, and K. Graichen, "Nonlinear model predictive torque control and setpoint computation of induction machines for high performance applications," *Control Engineering Practice*, vol. 99, p. 104415, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S096706612030071X>
- [163] A. Zanelli, J. Kullick, H. M. Eldeeb, G. Frison, C. M. Hackl, and M. Diehl, "Continuous control set nonlinear model predictive control of reluctance synchronous machines," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2021.
- [164] M. Schmidt, "An interior-point method for nonlinear optimization problems with locatable and separable nonsmoothness," *EURO Journal on Computational Optimization*, vol. 3, no. 4, pp. 309–348, Nov. 2015. [Online]. Available: <http://link.springer.com/10.1007/s13675-015-0039-6>
- [165] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

- [166] A. Malyshev, R. Quirynen, and A. Knyazev, "Preconditioned krylov iterations and condensing in interior point mpc method," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 388 – 393, 2018, 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896318327228>
- [167] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109809002969>
- [168] X. Zhang, L. Zhang, and Y. Zhang, "Model predictive current control for pmsm drives with parameter robustness improvement," *IEEE Transactions on Power Electronics*, vol. 34, no. 2, pp. 1645–1657, 2019.
- [169] H. Xu, X. Gui, T. Luan, X. Lang, and D. Xu, "A robust predictive current controller with incremental model and inductance observer for pmsm drive," in *2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, 2017, pp. 1–6.
- [170] J. Oravec, M. Bakošová, L. Hanulová, and M. Horváthová, "Design of robust mpc with integral action for a laboratory continuous stirred-tank reactor," in *2017 21st International Conference on Process Control (PC)*, 2017, pp. 459–464.
- [171] T. Badgwell and K. Muske, "Disturbance model design for linear model predictive control," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 2, 2002, pp. 1621–1626 vol.2.
- [172] K. R. Muske and T. A. Badgwell, "Disturbance modeling for offset-free linear model predictive control," *Journal of Process Control*, vol. 12, no. 5, pp. 617–632, 2002. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959152401000518>
- [173] G. Pannocchia and E. C. Kerrigan, "Offset-free receding horizon control of constrained linear systems," *Aiche Journal*, vol. 51, pp. 3134–3146, 2005.
- [174] M. Norambuena, P. Lezana, and J. Rodriguez, "A method to eliminate steady-state error of model predictive control in power electronics," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 7, no. 4, pp. 2525–2530, 2019.
- [175] X. Liu, L. Zhou, J. Wang, X. Gao, Z. Li, and Z. Zhang, "Robust predictive current control of permanent-magnet synchronous motors with newly designed cost function," *IEEE Transactions on Power Electronics*, vol. 35, no. 10, pp. 10 778–10 788, 2020.
- [176] X. Li, W. Tian, X. Gao, Q. Yang, and R. Kennel, "A generalized observer-based robust predictive current control strategy for pmsm drive system," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1322–1332, 2022.

- [177] O. Wallscheid and E. F. B. Ngoumtsa, “Investigation of disturbance observers for model predictive current control in electric drives,” *IEEE Transactions on Power Electronics*, vol. 35, no. 12, pp. 13 563–13 572, 2020.
- [178] J. Maciejowski, *Predictive Control with Constraints*. England.: Prentice Hall, 2002.
- [179] Z. Q. Zhu, D. Liang, and K. Liu, “Online parameter estimation for permanent magnet synchronous machines: An overview,” *IEEE Access*, vol. 9, pp. 59 059–59 084, 2021.
- [180] X. Li and R. Kennel, “Comparison of state-of-the-art estimators for electrical parameter identification of pmsm,” in *2019 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 2019, pp. 1–6.
- [181] A. Glac, V. Šmídl, Z. Peroutka, and C. M. Hackl, “Comparison of ipmsm parameter estimation methods for motor efficiency,” in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 895–900.
- [182] R. Sharma, *Lecture notes for the course IIA 4117: Model Predictive Control*, 2019. [Online]. Available: [https://web01.usn.no/~roshans/mpc/downloads/lecture\\_notes\\_MPC.pdf](https://web01.usn.no/~roshans/mpc/downloads/lecture_notes_MPC.pdf)
- [183] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [184] A. Wächter, “Short Tutorial: Getting Started With Ipopt in 90 Minutes,” in *Combinatorial Scientific Computing*, ser. Dagstuhl Seminar Proceedings (DagSemProc), U. Naumann, O. Schenk, H. D. Simon, and S. Toledo, Eds., vol. 9061. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009, pp. 1–17. [Online]. Available: <https://drops.dagstuhl.de/opus/volltexte/2009/2089>
- [185] P. Pauli, J. Köhler, J. Berberich, A. Koch, and F. Allgöwer, “Offset-free setpoint tracking using neural network controllers,” in *LADC*, 2021.
- [186] D. Schröder, *Elektrische Antriebe - Regelung von Antriebssystemen*, 01 2009.