Technische Universität München
TUM School of Engineering and Design

Model Predictive Control of Small Mechatronic Drive Systems with PM Synchronous Machines

Sebastian Wendel

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

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Sebastian Steinhorst
Prüfende der Dissertation:	
	1. Prof. Dr.-Ing. Dr. h.c. Ralph Kennel
	2. Prof. Dr.-Ing. Armin Dietz

Die Dissertation wurde am 04.10.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 02.11.2022 angenommen.

Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für leistungselektronische Systeme der Technischen Hochschule Nürnberg. Mein Dank gilt Herrn Prof. Dr.–Ing. Armin Dietz für die Anregung zu dieser Arbeit, die Unterstützung sowie die Freiheiten, die ich während der Bearbeitung genießen durfte. Sein Fordern und Fördern sowie seine positive Einstellung gegenüber Mensch und Wissenschaft hat mich stets inspiriert und angetrieben. Herrn Prof. Dr.–Ing. Dr. h.c. Ralph Kennel möchte ich für die wissenschaftliche Begleitung des Themas, den konstruktiven Gesprächen und die Ermöglichung der kooperativen Promotion danken. Seine Leidenschaft für die prädiktive Regelung hat mich stets motiviert. Herrn Prof. Dr.–Eng. Petros Karamanakos möchte ich besonders herzlich danken für die intensive Zusammenarbeit, die tiefgehenden Diskussionen und zahlreichen Hinweise. Die vermittelte Bedeutung von mathematisch und sprachlich präzise geformten Aussagen hat mich stark geprägt und in hohem Maße bereichert.

Meinen Kollegen, insbesondere Philipp Löhdefink, Michael Hoerner, Tobias Schindler, Michael Gerstner, Eyke Liegmann, Richard Steckel, Sebastian Hörlin, Lukas Rabenstein, Martin Regnet, Andreas Geiger, Felix Rojas und Mattia Rossi, möchte ich für die gemeinsame Zeit—beruflich und privat—sowie den zahlreichen Diskussionen, oftmals bis spät in die Nacht, herzlich danken. Ebenso möchte ich mich bei meinen Studenten, vor allem bei Philipp Gebhardt, Barnabas Haucke-Korber, Lukas Foss, Thilo Wendt, Annette Mai und Martin Brinkhaus für ihr überdurchschnittliches Engagement bedanken.
Dem Bayerischen Staatsministerium für Wirtschaft und Medien, Energie und Technologie danke ich für die finanzielle Unterstützung im Rahmen des Förderprogramms „Elektronische Systeme“ (Förderkennzeichen ESB048/004) und „EnCN 2.0“. Den beteiligten Industriepartnern, u.a., der Siemens AG, der Kübrich Ingenieurgesellschaft mbH & Co. KG und der Afag GmbH danke ich für die konstruktive Zusammenarbeit.

Mein letzter und besonderer Dank geht an meine Freunde, Eltern und natürlich an meine liebe Frau Selina sowie meiner Tochter Kira. Meine Frau war durch ihre motivierenden Worte, Fürsorge, Geduld und Liebe immer ein notwendiger und unersetzlicher Ausgleich während meiner Tätigkeit.

Büchold, November 2022
Sebastian Wendel

Kurzfassung

Es wird eine modellprädiktive Regelung für den optimierten Betrieb von kleinen mechatronischen Antriebssystemen mit magnetisch nichtlinearen permanentmagneterregten Synchronmaschinen diskutiert. Die in dieser Arbeit behandelten Optimierungen zielen darauf ab, die maximale Dynamik auszunutzen und gleichzeitig eine optimale Stromqualität mit minimalen Schaltvorgängen im Wechselrichter zu erreichen, um die Gesamtverluste zu reduzieren.
Darüber hinaus wird eine verbesserte Bedienbarkeit hinsichtlich der Reglereinstellungen über den gesamten Betriebsbereich angestrebt. Dies ist z.B. für elektrische Antriebe in Werkzeugmaschinen von Vorteil, bei denen je nach Arbeitspunkt eine hohe Dynamik sowie eine geringe Drehmomentwelligkeit wichtig sind.
Um dieses Ziel zu erreichen, steht der Nachweis der Anwendbarkeit und des Nutzens der modellprädiktiven Regelung für elektrische Kleinantriebe im Vordergrund. Dazu wird eine Lösung vorgestellt, die es erlaubt, die modellprädiktive Regelung trotz erhöhten Rechenaufwands auch für Antriebe mit kleinen elektrischen Zeitkonstanten zu nutzen. Diese Arbeit leistet einen Beitrag zum Forschungsgebiet der modellprädiktiven Regelung.

Abstract

Model predictive control for optimized operation of small mechatronic drive systems with magnetically nonlinear permanent magnet synchronous machines is discussed. The optimizations addressed in this thesis aim to exploit the maximum dynamics while at the same time achieving optimum current quality with minimum switching operations in the inverter to reduce overall losses. Moreover, improved usability with regard to the controller settings is intended to be achieved over the entire operating range. This is beneficial, e.g. for electric drives in machine tools, in which, depending on the operating point, high dynamics and a low torque ripple are important.
To achieve this goal, the main focus is to prove the applicability and benefits of model predictive control for small electrical drives. To this end, a solution is presented that allows model predictive control to be solved even for drives with small electrical time constants, despite increased computational effort. This thesis contributes to the research field of model predictive control.

Enthusiasm releases the drive to carry you over obstacles and adds significance to all you do.
Norman V. Peale

Contents

1 Introduction

Small electrical drives—also called fractional horse power drives—are motors with a rated power of less than 746 W. A market study conducted by Grand View Research estimates global sales of electric motors in 2020 at USD 143 billion and predicts a compound annual growth rate from 2021 to 2028 of $6.4\,\%$ [GVR21]. Here, the small drives segment dominates the electric motors market, taking the largest revenue share of $87\,\%$ in 2020 [GVR21]. Consequently, their importance in industrial automation and the automotive sector is increasing steadily. A constantly growing part of fractional horse power drives are permanent magnet synchronous machines (PMSMs), which are mostly operated by block commutation. However, more and more low-cost microcontrollers are appearing in combination with a field-oriented control [Bla72], which enables a reduced torque ripple and an increased efficiency due to the field orientation. The high market share of these small drives and the steadily increasing demands from industry and the automotive sector will require not only cost reduction but also significantly more improvements in energy efficiency and performance optimization in the future. Thus, for an increasing number of applications, the state of the art control methods are not sufficient anymore.

1.1 Research Issue

In recent decades, several control methods have been proposed for controlling converters and drives. The most commonly used ones are shown in Fig. 1.1 [CKK$^+$08, Lin10, RC12, KPR$^+$15, LKF$^+$16, KLGK20]. Some of these methods, such as nonlinear hysteresis control, are very well established and simple, while newer control methods, which allow an improved system behavior, are generally more complex or require much more processing power from the control platform. Field-oriented control (FOC) [Has69, Leo96, Bla72] and direct torque control (DTC) [TN86] are particularly common and are used in a wide range of drive applications [GRS$^+$14]. Here, a distinction is made between direct and indirect control methods [Hol94].

One common method of direct control is DTC—as a type of hysteresis control—that already takes into account the nonlinear nature of the power inverter and thus directly manipulates the switching states of the power semiconductors. The origin of this control scheme is in analog electronics and thus requires a very high sampling frequency when implemented in a digital

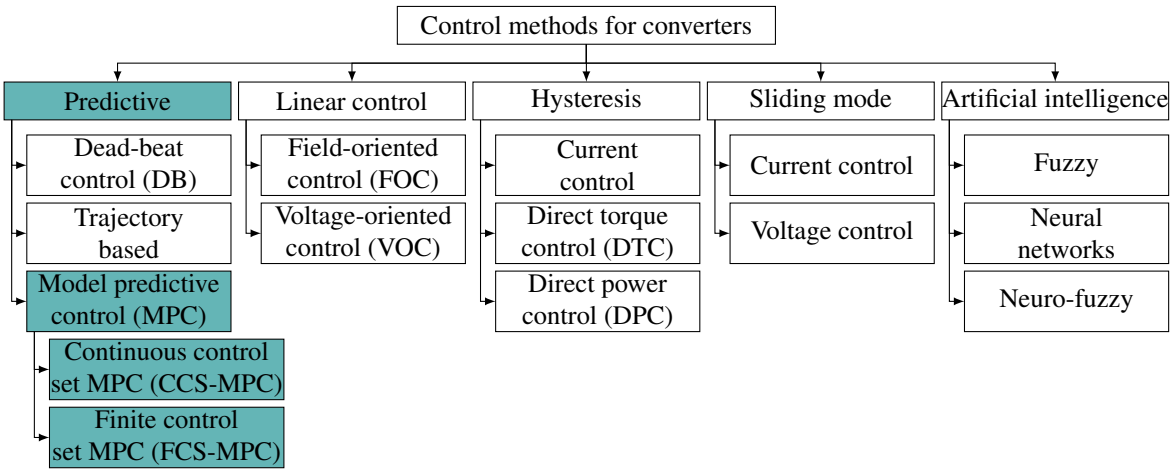


Figure 1.1: The most common types of control schemes for power converters and drives.[1]

platform. One common indirect control method, utilizing a modulator stage for generating the inverter switching signals, is FOC, which uses simple linear proportional–integral (PI) controllers. FOC operates on the principle of single-input single-output (SISO) [Sch15, p. 134], which requires decoupling the system states of the PMSM to obtain a stable control loop. Moreover, nonlinearities, once they occur, can be difficult to account for and can adversely affect the control performance, especially in transients [Leu14, p. 1].
Consequently, alternating between FOC and DTC can be advantageous, since DTC is particularly effective and dynamic in field weakening operation, while FOC with modulator is useful in steady-state operation, e.g., due to lower harmonics [Sch15, p. 844]. However, power converter systems are subject to several system constraints and technical requirements, such as maximum current, maximum switching frequency and total harmonic distortion (THD), which cannot be directly incorporated into a linear controller design.
To sum up, classical control theory has been constantly adapted to meet increasing requirements, although new approaches may be better suited to modern, digitally controlled converters.

With the development of more powerful microprocessors, new control schemes have been proposed. Among the most important are fuzzy logic control, neural networks, sliding mode control and predictive control. Of these new methods, predictive control seems to be a very interesting alternative for controlling power converters and drives. The predictive family comprises very different approaches. The idea behind all predictive controllers is that they first of all use a model of the controlled system to predict the future behavior of the controlled states, followed by optimization criteria to select suitable control signals. When it comes to designing a controller that enables the maximum possible dynamic range, predictive controllers seem to be one

[1] Specifically for MPC, a distinction between CCS- and FCS-MPC is made in this work, although a different and much more fine-granular distinction can be found, e.g., in [KLGK20].

of the most promising options [Sch15, p. 422]. One of the best known predictive controllers is dead-beat control [CKK+08]. Here, a model of the controlled system is used to calculate the control signal, i.e., the voltage, that brings the control error to zero in a single discrete sampling step using a modulator stage. A different and very powerful predictive control strategy that is now being applied in power electronics is model predictive control (MPC), which is the subject of this thesis. The functional principle of MPC is not completely new; the structure is based on optimal state control, i.e., linear quadratic regulator (LQR), where in the field of power electronic systems aspects of hysteresis control are incorporated [DP04, Wan09, Lin10, Leu14]. MPC operates as a multiple-input multiple-output (MIMO) controller, and its model-based character means that it can consider complex correlations between measured and control signals. Moreover, the predictive character of MPC enables a new control signal to be planned in advance for each control interval by optimizing a cost function online. This is in contrast to the reacting character of FOC and DTC.

With MPC, a distinction can be made between continuous control set-MPC (CCS-MPC) and finite control set-MPC (FCS-MPC).[1] The former optimizes continuous and the latter discrete control signals. In particular FCS-MPC, also known as direct model predictive control (DMPC), has gained a large momentum, where the aim is to find the constrained optimal discrete voltage space vector (SV) that minimizes a pre-defined cost function. This enables the control objectives, such as output reference tracking, to be met, while the direct control nature of FCS-MPC enables very fast transient responses to be achieved. Moreover, when long-horizon FCS-MPC is considered, two features are prominent. First, an improved steady-state system performance can be attained, as indicated by the reduced THD in the variables of concern for a given average switching frequency f_{sw}. Second, the stability of the system can be improved. In addition, MPC can increase usability by automatically adjusting the controller by virtue of its model-based character, eliminating the need for adjustments based on operating points.
Although research tends to concentrate on FCS-MPC for low voltage drives, in industry it is usually applied to medium voltage drives [Gey17] due to the slow electrical time constants that facilitate real-time MPC processing. Thus, it can be concluded that, although FCS-MPC is very promising, the implementation of the algorithm for drives with small electrical time constants is mostly inhibited by the limited computational power [Gey17, p. 19].

The optimizations addressed in this thesis aim at utilizing the maximum drive dynamics while at the same time achieving optimum current quality with minimum switching operations in the inverter. The latter tends to result in low losses. Moreover, improved usability with regard to the controller settings is intended to be achieved over the entire operating range. This is beneficial, e.g., for electrical drives in machine tools, where, depending on the operating point,

fast dynamics as well as low torque ripple are important. The problem considered in this thesis is therefore defined as follows. Determination of a control method for small electrical drives in highly dynamic and high-precision industrial applications with

- the most dynamic current, torque and speed control taking into account a predefined voltage, current, torque and/or speed limit,
- the lowest possible overshoots in transients,
- the lowest possible current THD[2], i.e., current ripple, at the lowest possible switching frequency, which tends towards a high energy efficiency,
- a reduction of the mechanical natural oscillations.

Defining the problem leads to the question of which control approach is capable of achieving all these goals, while also considering clarity and easy adaptation to new system topologies. Assuming that MPC is a possible solution to this question, but has not been applicable so far due to the high computational effort that needs to be solved in short computation intervals, the research question of the dissertation is defined as:

Is MPC useful for small electrical drives, and if so, when and how?

1.2 Contributions

To answer the research question considering the previously mentioned arguments, the main contributions of this thesis are as follows:

1. Improvement of the dynamic behavior of small electric drives by using FCS-MPC, where system constraints for, e.g., current and speed, are directly considered during optimization. The latter protects the system and avoids subsequent limits that would lead to a suboptimal control signal.
2. Introduction of heterogeneous calculation platforms, i.e., system-on-a-chip field-programmable gate arrays (SoC FPGAs), that enable FCS-MPC with a long horizon even for electrical drives with small time constants by being able to perform prediction steps with control frequencies of up to several hundred kHz. The combination of processor and FPGA with shared resources in a single chip enables the multidimensional execution of the algorithms, i.e., parallel prediction and serial solution of the optimization problem.
3. Online adaptation and tracking of system parameters to enable an accurate and robust prediction over the entire operating range, even for highly utilized synchronous machines.

[2] „While low harmonic current distortion ensures low torque ripple, achieving low torque ripple does not necessarily entail low current distortion.“ [Hol16]

4. Insertion of a variable switching point, such that FCS-MPC switches within the control interval, thus reducing the current ripple without increasing the switching frequency and losing the dynamic behavior of direct MPC.
5. Enabling the use of FCS-MPC even for highly utilized synchronous machines by preventing prediction errors due to a nonlinear magnetic circuit, i.e., saturation and cross-coupling effects. Since inductances can change significantly over the operating range—one half is not uncommon—these effects are important for appropriate control performance, even though a linear magnetic circuit is usually used to model FCS-MPC. For this purpose, a prediction based on flux linkage is introduced, which results in significantly improved control behavior, such as in transients and steady state.
6. Increasing steady-state accuracy by introducing a repetitive flux linkage correction term that compensates for parameter and model inaccuracies with respect to the rotor position.
7. Combining CCS- and FCS-MPC in a cascaded structure to enable the prediction and optimization of the mechanical and electrical system behavior without any loss in granularity. The aim of this is to additionally damp mechanical natural oscillations in the drive train.

1.3 Restrictions and Specifications

With respect to the research question, the following restrictions and specifications apply to the proposed case study:

1. A PMSM is typically chosen in applications that require electrical machines with highly dynamic behavior and high power density. The study considers up to five different PMSM types—with varying degrees of saturation, i.e., magnetic nonlinearity, and reluctance—from up to five different production batches, in a power class below a nominal power of $200\,\mathrm{W}$, see Table A.3 of Appendix A.3.1. Asynchronous machines are not considered.
2. Only three-phase systems are considered. All coils and thus inductances are symmetrical.
3. The skin and proximity effect are neglected.
4. A non-sinusoidal field distribution in the air gap field, e.g., due to an asymmetric air gap, winding harmonics, slotting effects or rotor eccentricity, is not modeled, but nevertheless compensated for by the controller, see Chapter 8.
5. Iron losses of the fundamental frequency are due to the machine design and cannot be mitigated by the control or modulation approach. However, the switching frequency and resulting harmonics cause current harmonics that can lead to significant inverter-based iron losses, i.e., losses in the stator and rotor. Hence, their reduction is a potential criterion

in the cost function. Nevertheless, as will be shown in Chapter 4, this effect is quite low for the small drives under consideration. Consequently, iron losses are neither modeled nor addressed in the optimization, but are still compensated for in the prediction stage, see Chapter 8.

6. Although a fixed switching frequency of the semiconductors is required in some applications, e.g., grid applications, it is not necessary for motor drives.
7. A rotor position sensor is used. Therefore, no encoderless method is considered.

1.4 Outline

Taking into account the above restrictions, the contributions to answer the research question are organized as follows. These include the author's own publications and supervised theses.

Chapter 2 describes the electrical and mechanical controlled system. It contains an overview and classification of MPC methods in comparison to the state of the art, i.e., FOC, DTC and LQR, in terms of their suitability for small electrical drives.
To ensure real-time capability, Chapter 3 proposes a powerful real-time computing system and computationally efficient execution of MPC by way of heterogeneous implementation. This chapter is mainly based on [WDK17b, WDK17a, WGL+19]. Furthermore, the importance of measurement accuracy is discussed on the basis of [Bri17].
Chapter 4 describes the determination and tracking of system parameters such as motor parameters, to significantly improve the control performance of MPC. This is based primarily on [HWD17, Hoe17, Hoe18, Geb18, Geb19, Geb20, Huf21].
Chapter 5 describes how the computational effort can be reduced and Chapter 6 describes the introduction of a variable switching point to improve the switching granularity in DMPC. Chapter 7 further develops the variable switching approach to allow the approach to be extended to PMSM with a nonlinear magnetic circuit. These chapters are based on [WKDK19, WKDK20, WKG+21].
Even if an exact analytical description of the controlled system—see Chapter 4—significantly increases the control quality of MPC, there will always be possible small deviations or nonlinearities that are either not describable at all or only with great effort. Thus, Chapter 8 introduces a flux linkage correction, the aim of which is to enable steady-state accuracy to be achieved over the entire operating rang.
Chapter 9 separates the control problem depending on the electrical and mechanical time constants, thus enabling a predictive calculation with an acceptable granularity for the entire mechatronic drive system. This chapter is based on [WHKDK18, WHKDK19, HK20, WHKD20].

2 State of the Art and Performance Classification

The main focus in this work is to prove the applicability and advantages of MPC for small mechatronic drive systems, distinguishing between FCS-MPC and CCS-MPC. For this purpose, this chapter briefly explains the controlled system under consideration, i.e., the two-level voltage source inverter, the permanent magnet synchronous machine and the mechanical drive train. Afterwards, an overview of the common control and actuation methods for small electrical drives is given. Although there are a variety of different control strategies, some of which are shown in Fig. 1.1, only the most common ones are used for comparison. Since a mechatronic drive system is considered, FOC, classical DTC and the algorithm of interest, i.e., MPC, are explained for the electrical controlled system according to the state of the art. For the mechanical controlled system, MPC competes with a PI controller and LQR[1]. Finally, the mentioned control methods are compared in simulation and experiment using the specified controlled system as an example. Thus, the potential for improvement and the open questions are clearly highlighted in this chapter to illustrate the motivation and goals of the work.

2.1 Control of Electrical Drives

The basic structure of the controller and the controlled system is shown in Fig. 2.1, where a power electronic system, i.e., inverter, with three-phase PMSM is directly assumed.
The system output, i.e., the controlled variable $y(k)$, is controlled using the reference signal $y^*(k)$ and calculating a control signal $u(k)$. The disturbance is labeled by $z(k)$. The control deviation $\xi(k)$ may already be part of the controller, depending on the selected control method. Moreover, some important preliminary considerations are necessary for the realization of the controllers. First, the space vector notation is briefly explained. Second, some indirect control methods, e.g., FOC, require a modulator, which is described in the following. Third, control schemes are typically implemented on a control hardware such as a microcontroller, which requires a discretization of the controlled system and is therefore discussed in the following. In addition, the most important definitions are briefly explained for clarification:

[1] The LQR controller is based on [Lun17, p. 304].

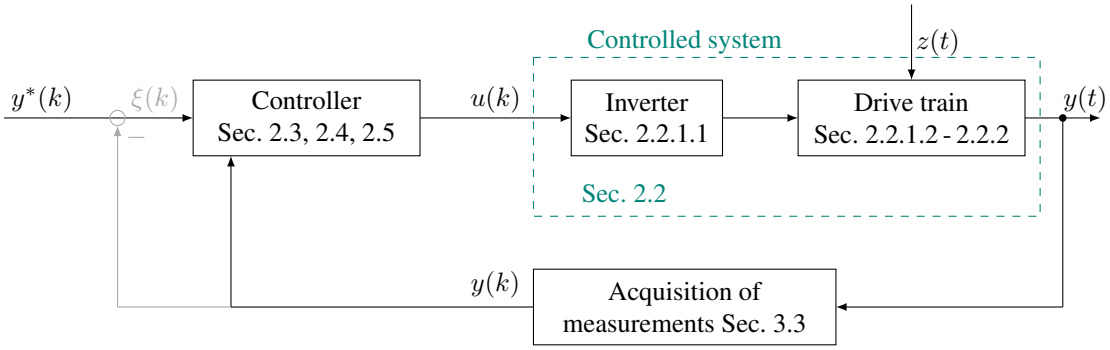


Figure 2.1: General scheme of the discrete control structure for a drive train with inverter.

1. Control frequency f_{c}: is the frequency for the cyclic calculations of the controller. More concretely, a distinction is made hereinafter between f_{cf} for the control of electrical variables and f_{cc} for the control of mechanical variables.
2. Sampling frequency f_{a}: is the frequency for the cyclic acquisition of new measurements, see Section 3.3.
3. Switching frequency f_{sw}: is the average switching frequency which is defined by one ON and one OFF transition per phase leg, see Section 2.2.1.1. Thus, the ON and OFF transitions constitute one switching pulse.

2.1.1 Space Vector Representation

The mathematical treatment of the states in a stationary three-(multi-)axis stator system are considerably simplified if the quantities are transformed into a two-axis orthogonal system (stationary or rotating). The word system refers to coordinate system, which is also called reference system. In doing so, a generally valid three-phase system of balanced and sinusoidal quantities can be described by a space vector $\overrightarrow{\zeta}$ [KR59, p. 295]. The space vector representation can be done by complex numbers when the spatial as well as the temporal assignment of the electromagnetic quantities are taken into account [Sch13, p. 301 ff.]. Separating the time dependency in the space vector, the phasor describes just the complex time independent variable $\underline{\zeta}$, where $\overrightarrow{\zeta} = \underline{\zeta} e^{j\omega t}$ [VPdD16, p. 109]. Thus, a stationary or rotating phasor has a real and an imaginary component [VPdD16, p. 35]. However, in this work the representation of the transformed states does not use the complex description, since it is only valid if temporal and spatial quantities are sinusoidal and geometrically related. As explained in [Lan14, p. 12], for highly dynamic non-sinusoidal operation of the controlled system, e.g., during transients when using an inverter, a distinction between temporal and spatial relations is recommended. [Gey17, p. 42] also recommends for the same reason an orthogonal system with lumped parameters instead of the

(complex-valued) space vector representation. Therefore, the author uses the word space vector for the real-valued vectorial notation, i.e., $\boldsymbol{\zeta} = [\Re\{\overrightarrow{\zeta}\}\ \Im\{\overrightarrow{\zeta}\}]^T$, in an orthogonal system.

For both representation possibilities it is assumed that the sum of the signals of the individual phases is zero (e.g., for the phase currents or phase voltages), i.e., in a three-phase system all phases are shifted against each other by exactly $\frac{2\pi}{3}$ rad and have identical amplitudes. The previous assumption is valid since only the fundamental wave of the system is considered, which leads to a fundamental wave model in the time domain.

The applied orthogonal theory allows using the Clarke ($\alpha\beta$) transformation [VPdD16, p. 98]. Here, the three-phase electromagnetic quantities are converted into an equivalent orthogonal system with axes α and β, where the α-axis is usually aligned with the axis of the first phase, see Fig. 2.2(a). The resulting space vector $\boldsymbol{\zeta}_{\alpha\beta}$ rotates with the same fundamental frequency as the phases in the stator coordinate system. Using the amplitude invariant Clarke transformation, i.e., $a = \frac{2}{3}$, the matrices

$$\boldsymbol{K}_{\alpha\beta0} = a \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad \boldsymbol{K}^{+}_{\alpha\beta0} = \frac{1}{a} \begin{bmatrix} \frac{2}{3} & 0 & \frac{2}{3} \\ -\frac{1}{3} & \frac{1}{\sqrt{3}} & \frac{2}{3} \\ -\frac{1}{3} & -\frac{1}{\sqrt{3}} & \frac{2}{3} \end{bmatrix} \tag{2.1}$$

are used to transform any three-phase state $\boldsymbol{\zeta}_{abc} = [\zeta_a\ \zeta_b\ \zeta_c]^T$ into the space vector plane with $\boldsymbol{\zeta}_{\alpha\beta0} = [\zeta_\alpha\ \zeta_\beta\ \zeta_0]^T$ through the operation

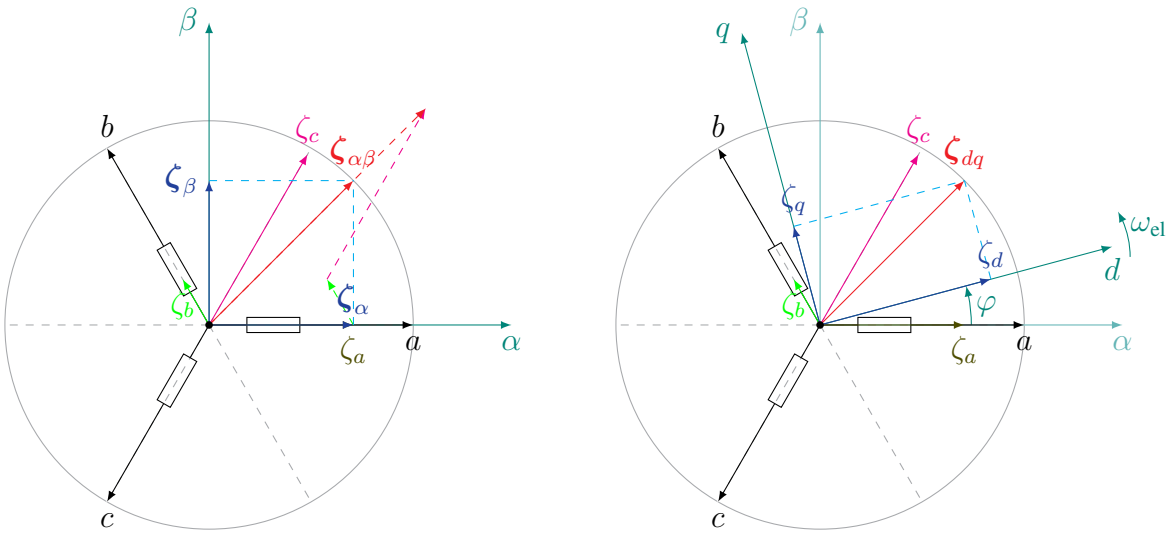
$$\boldsymbol{\zeta}_{\alpha\beta0} = \boldsymbol{K}_{\alpha\beta0}\boldsymbol{\zeta}_{abc} \tag{2.2}$$

and vice versa with the pseudo inverse of matrix $\boldsymbol{K}_{\alpha\beta0}$, i.e.,

$$\boldsymbol{\zeta}_{abc} = \boldsymbol{K}^{+}_{\alpha\beta0}\boldsymbol{\zeta}_{\alpha\beta}\,. \tag{2.3}$$

Even if, for example, triple-order harmonics in delta-connected PMSMs generate a circulating current i_0 in the windings—which causes ohmic losses—this current is not accessible from the terminal connection and therefore cannot be influenced by the controller [Min13, p. 42]. Consequently, the additional component ζ_0 is only necessary if the system is unbalanced and the neutral point of star-connected windings is not floating. However, since the neutral point is floating for the case considered in this thesis,

$$0 = \zeta_a + \zeta_b + \zeta_c \tag{2.4}$$



(a) Superposition of the three phases to a two-axis right-angled representation. Both are in the stator coordinate system.

(b) Superposition of the three phases to a two-axis right-angled representation in a rotating coordinate system.

Figure 2.2: Transformation of phase-related quantities into a rotating space vector using the amplitude invariant representation by the example of a rotor angle of $\varphi = \frac{\pi}{12}$ rad.

holds. Thus, the last row of $\boldsymbol{K}_{\alpha\beta0}$ and column of $\boldsymbol{K}^{+}_{\alpha\beta0}$ can be neglected and a two-dimensional vector $\boldsymbol{\zeta}_{\alpha\beta} = [\zeta_\alpha\ \zeta_\beta]^T$ results. Consequently, if symmetry is assumed, the measurement information of two phases would actually be sufficient.
It has to be noted that any value can be chosen for the factor a in the Clarke transformation, i.e., (2.2) and (2.3). Typically, a distinction is made between an amplitude invariant transformation ($a = \frac{2}{3}$) and a power invariant transformation ($a = \sqrt{\frac{2}{3}}$) [Sch15, p. 1554]. Although some authors chose the latter due to symmetry effects [Lan14, p. 12 ff.], in the following the amplitude invariant transformation is chosen, because, first, the focus is on current rather than power control and, second, for simplicity, i.e., parameter scaling is avoided.

Using the Park transformation, i.e., a complex rotation operator, the vectorial notation $\boldsymbol{\zeta}_{\alpha\beta}$ can be represented in a rotating rather than a stationary coordinate system. Here, the space vector remains stationary while the coordinate system rotates with the rotor flux. For synchronous machines, the rotor flux vector is synchronized with the rotor, so the coordinate system can simply rotate with the rotor, i.e., at the electrical angular speed ω_{el}, keeping the two resulting d- and q-axis orthogonal to each other. In steady state this results in constant quantities for the space vector $\boldsymbol{\zeta}_{dq}$ and the respective components, see Fig. 2.2(b). The positive d-axis always points towards the north pole of one pole pair on the rotor, i.e., towards the amplitude of the rotor flux of this pole, and is hence aligned with the rotor flux vector. For more information,

see, e.g., [Sch13, p. 301 ff.], [VPdD16, p. 83 ff.]. With the knowledge of the rotor flux angle φ and $a = \frac{2}{3}$, the matrices

$$\boldsymbol{K}(\varphi)=\frac{2}{3}\begin{bmatrix} \cos(\varphi) & \cos\left(\varphi-\frac{2\pi}{3}\right) & \cos\left(\varphi+\frac{2\pi}{3}\right) \\ -\sin(\varphi) & -\sin\left(\varphi-\frac{2\pi}{3}\right) & -\sin\left(\varphi+\frac{2\pi}{3}\right) \end{bmatrix}, \; \boldsymbol{K}^{+}(\varphi)=\begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \cos\left(\varphi-\frac{2\pi}{3}\right) & -\sin\left(\varphi-\frac{2\pi}{3}\right) \\ \cos\left(\varphi+\frac{2\pi}{3}\right) & -\sin\left(\varphi+\frac{2\pi}{3}\right) \end{bmatrix} \tag{2.5}$$

are used to transform any three-phase variable $\boldsymbol{\zeta}_{abc} = [\zeta_a \; \zeta_b \; \zeta_c]^T$ into a two-dimensional reference plane with the vector $\boldsymbol{\zeta}_{dq} = [\zeta_d \; \zeta_q]^T$ through the operation

$$\boldsymbol{\zeta}_{dq} = \boldsymbol{K}(\varphi)\boldsymbol{\zeta}_{abc} \tag{2.6}$$

and vice versa by using the pseudo inverse $\boldsymbol{K}^{+}(\varphi)$

$$\boldsymbol{\zeta}_{abc} = \boldsymbol{K}^{+}(\varphi)\boldsymbol{\zeta}_{dq}\,. \tag{2.7}$$

2.1.2 Modulation Methods

Indirect control methods, e.g., FOC, do not control the inverter directly, but instead control and output continuous control signals. In doing so, several modulation techniques are able to convert the continuous control signals into discrete switching signals. A good overview of modulation techniques is given, e.g., in [HKL98, LKF+16]. Basically, a distinction is made between carrier-based pulse width modulation (CB-PWM)—see the holistic overview in [HL03]—and space vector modulation (SVM). A comparison is given, e.g., in [HL03, p. 265 ff.]. The two main differences are that, first, the latter has gained popularity thanks to digital processors that allow duty cycles from vector control to be applied directly in vector coordinates [Hol94, HKL98]. Second, SVM splits the inverter zero states equally and thus can utilize the dc-link more than CB-PWM without additional common mode voltage injection.

More concretely, continuous modulation methods such as sinusoidal PWM (SPWM)—which is CB-PWM without any modifications—, space-vector PWM (SVPWM), i.e., King's method—in common parlance simply named SVM—and third harmonic injection PWM (THIPWM) [HKL98] can be distiguished, which are mainly used at low modulation indices. Furthermore, discontinuous modulation methods (DPWM)—which can be applied with SVM and CB-PWM—such as DPWM1, DPWM2, DPWM3, DPWMMIN and DPWMMAX [HKL98] can be used at high modulation indices to increase the modulation linearity range and to reduce switching losses. To improve flexibility, the pulse pattern can be changed depending on the operating point

by varying between the methods mentioned. Even though the modulator is a very powerful tool to apply the continuous control signals, it is not straightforward to define a suitable criterion based on which the modulation method is chosen or switched, although a hysteresis band can be implemented to avoid toggling [HKL98].
A further distinction can be made between synchronous and asynchronous PWM, whereby the former can be particularly advantageous for low pulse numbers [Hol92, Hol16], i.e., low f_{sw}/f_1 ratio, where f_1 is the fundamental frequency. For synchronous PWM, many methods exist for optimizing the switching angles, such as optimized pulse patterns (OPP), selective harmonic elimination (SHE) or selective harmonic mitigation (SHM). However, due to the high number of pulses required in the case of small electric drives, asynchronous—as state of the art—is chosen.
Finally, for digital controllers, a general distinction can be made between symmetrical and asymmetrical regularly sampled modulation [HL03]. Although the latter may be better for low pulse numbers [HL03, p. 241], the former is typically chosen for small drives, i.e., for high pulse numbers, and is hence used hereinafter.[2]

Space vector modulation (SVM) As aforementioned, SVM allows the modulation of the switching signals of an inverter—see Section 2.2.1.1—directly based on a continuous control signal, i.e., a reference voltage space vector $\boldsymbol{v}_{dq}^*$. In doing so, the exemplarily shown $\boldsymbol{v}_{dq}^*$ in Fig. 2.3(b) is devised as summation of existing discrete voltage space vectors (SVs) (see Table 2.1), which can be physically set by the inverter. Consequently, the four SVs that form a sector—wherein $\boldsymbol{v}_{dq}^*$ lies—in the VSI voltage hexagon of Fig. 2.3(b) are required.
As a result, the two active voltage space vectors (ASVs), i.e., the left $\boldsymbol{v}_{\mathrm{L}}$ and the right $\boldsymbol{v}_{\mathrm{R}}$, as well as the so-called zero voltage space vectors (ZSVs) must be determined. Using the specific example of setting $\boldsymbol{v}_{dq}^*$ in Fig. 2.3(b), $\boldsymbol{v}_{\mathrm{L}} = \boldsymbol{v}_2$, $\boldsymbol{v}_{\mathrm{R}} = \boldsymbol{v}_1$ and $\boldsymbol{v}_0$ as well as $\boldsymbol{v}_7$ correspond to the zero space vectors. To reproduce the vector $\boldsymbol{v}_{dq}^*$ with the discrete space vectors by vector addition, it is necessary to calculate the required switching times based on [Sch15, p. 667 ff.]

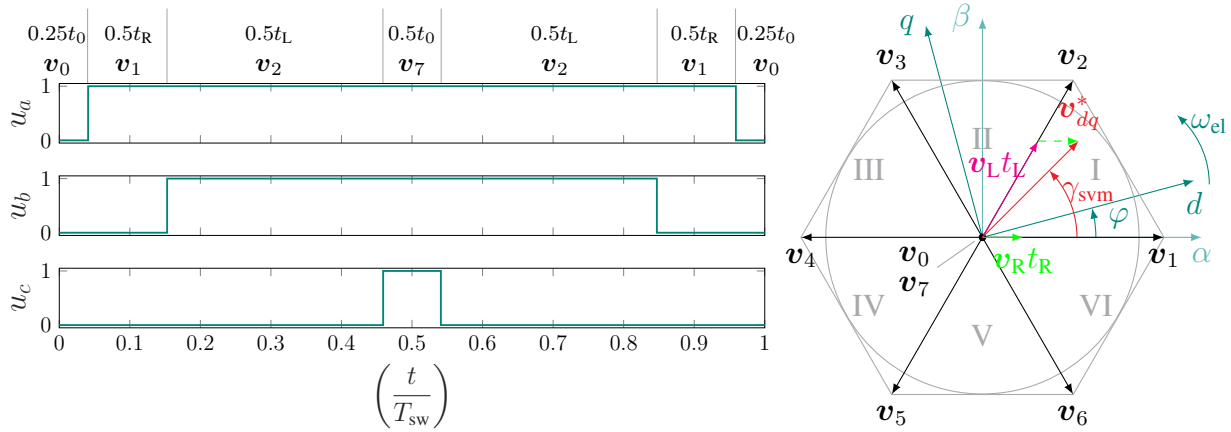
$$t_{\mathrm{R}} = \sqrt{3} T_{\mathrm{sw}} \frac{|\boldsymbol{v}_{dq}^*|}{V_{\mathrm{dc}}} \sin\left(\frac{\pi}{3} - \gamma_{\mathrm{svm}}\right) \tag{2.8a}$$

$$t_{\mathrm{L}} = \sqrt{3} T_{\mathrm{sw}} \frac{|\boldsymbol{v}_{dq}^*|}{V_{\mathrm{dc}}} \sin\left(\gamma_{\mathrm{svm}}\right) \tag{2.8b}$$

$$t_0 = T_{\mathrm{sw}} - t_{\mathrm{R}} - t_{\mathrm{L}}\,, \tag{2.8c}$$

where T_{sw} is the modulation interval and γ_{svm} is the angle between $\boldsymbol{v}_{dq}^*$ and $\boldsymbol{v}_{\mathrm{R}}$ of the corresponding sector. However, the implementation of trigonometric functions on an FPGA is

[2] It should be mentioned that the sampling frequency f_{a} is equal to f_{sw} if no oversampling is used and the control frequency f_{c} is equal to f_{sw} in the considered case of symmetrical regular sampling.



(a) Exemplarily illustrated switching times for sector I.

(b) Voltage space vector summation.

Figure 2.3: Schematic example of the space vector modulation method used.

usually very resource and computationally intensive, see Chapter 3. Therefore, the adaptions shown in [QD15, p. 23 ff.] are used to implement a computationally more favorable method of the described SVM, which avoids additional trigonometric functions, see Appendix A.2.1.

There are several variants to split and merge the determined SVs, i.e., $\boldsymbol{v}_\mathrm{R}$, $\boldsymbol{v}_\mathrm{L}$, $\boldsymbol{v}_0$, $\boldsymbol{v}_7$, in order to generate $\boldsymbol{v}^*_{dq}$. However, the most common SVM variant is the symmetrical arrangement of the pulse pattern, since the current harmonic distortion, i.e., I_THD, is particularly low and the fundamental can be sampled, allowing the use of PI controllers. More concretely, within the modulation interval T_sw one ZSV is applied at the beginning and end, the other ZSV is applied in the middle, and the two ASVs are applied to the left and right of the middle, see Fig. 2.3(a). This implies, that the calculated switching times, i.e., t_R, t_L, t_0, are symmetrically split. For the sequence of switching operations in odd sectors, i.e., I, III, V, t_R is the first and t_L the second applied ASV. This is exemplified in Fig. 2.3(a). For the even sectors, i.e., II, IV, VI, it is the other way round, i.e., t_L first and t_R second. This arrangement means that only one half-bridge at a time performs a switching action between the individual space vectors, thus keeping the number of switching transitions as low as possible.

For reasons of benchmarking, FOC with SVM, i.e., SVPWM, is used for all further considerations. However, due to the aforementioned challenges when changing between various modulation schemes or pulse patterns over the operating range, as well as the potential increase in dynamic response by eliminating the modulator stage, this thesis investigates the feasibility of applying DMPC to small electrical drives. The latter is achieved by a more direct approach between control target optimization and modification of inverter switches. Consequently, the proposed DMPC and the compared DTC do not use a modulation method, i.e., no modulator is present.

2.1.3 Discretization

In general, the analytical model equations are initially formulated in the continuous-time domain. However, these models need to be implemented in the control structures of digitally operating real-time computing platforms once they are used in the real-world for their experimental applications. Therefore, a discretization is necessary, e.g., according to Euler, Tustin or exact discretization. More specifically, the forward Euler method—as the most simple approximation method—and the exact discretization are used in the following.
Considering the general state space description, the exact discretization is specified by [MD17, p. 46]

$$\boldsymbol{A} = e^{\boldsymbol{A}_\mathrm{c} T_\mathrm{c}} = \sum_{l=0}^{\infty} \frac{\left(\boldsymbol{A}_\mathrm{c} T_\mathrm{c}\right)^l}{l!} = \boldsymbol{I} + \boldsymbol{A}_\mathrm{c} T_\mathrm{c} + \frac{\left(\boldsymbol{A}_\mathrm{c} T_\mathrm{c}\right)^2}{2} + \frac{\left(\boldsymbol{A}_\mathrm{c} T_\mathrm{c}\right)^3}{6} + \ldots \tag{2.9a}$$

$$\begin{aligned}\boldsymbol{B} &= \int_0^{T_\mathrm{c}} e^{\boldsymbol{A}_\mathrm{c}\tau} \boldsymbol{B}_\mathrm{c} \mathrm{d}\tau = \int_0^{T_\mathrm{c}} \sum_{l=0}^{\infty} \frac{\left(\boldsymbol{A}_\mathrm{c}\tau\right)^l}{l!} \boldsymbol{B}_\mathrm{c} \mathrm{d}\tau = \left| \sum_{l=0}^{\infty} \frac{\boldsymbol{A}_\mathrm{c}^l \tau^{l+1}}{l!\,(l+1)} \boldsymbol{B}_\mathrm{c} \right|_{\tau=0}^{\tau=T_\mathrm{c}} \\ &= \sum_{l=1}^{\infty} \frac{\boldsymbol{A}_\mathrm{c}^{l-1} T_\mathrm{c}^l}{l!} \boldsymbol{B}_\mathrm{c} = T_\mathrm{c} \boldsymbol{B}_\mathrm{c} + \frac{\boldsymbol{A}_\mathrm{c} T_\mathrm{c}^2}{2} \boldsymbol{B}_\mathrm{c} + \frac{\boldsymbol{A}_\mathrm{c}^2 T_\mathrm{c}^3}{6} \boldsymbol{B}_\mathrm{c} + \ldots \end{aligned} \tag{2.9b}$$

where the procedure can be aborted after the l-th element. T_c is the discrete control interval.
As shown, e.g., in [Gey17, p. 13], [Gey17, p. 155], for short sampling intervals as well as prediction horizons, forward Euler is sufficiently accurate. Forward Euler is described by

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} \approx \frac{\boldsymbol{x}(k+1) - \boldsymbol{x}(k)}{T_\mathrm{c}} \quad \Rightarrow s \approx \frac{z-1}{T_\mathrm{c}} \tag{2.10}$$

where the right-hand side shows merely for illustration purposes forward Euler in the s-plane using z-transform. This leads to the discretized system equations

$$\boldsymbol{A} = \boldsymbol{I} + \boldsymbol{A}_\mathrm{c} T_\mathrm{c} \tag{2.11a}$$

$$\boldsymbol{B} = \boldsymbol{B}_\mathrm{c} T_\mathrm{c}\,. \tag{2.11b}$$

For both discretization methods, $\boldsymbol{C} = \boldsymbol{C}_\mathrm{c}$ applies to the output matrix.

2.2 Controlled System

This section describes the controlled system. The restrictions in Section 1.3 have to be considered. Fig. 2.4 illustrates the controlled system under consideration, where the mechanical rotor position φ_m is measured and used to compute the mechanical angular speed ω_m. The latter relates to the electrical angular speed with $\omega_\mathrm{el} = p\omega_\mathrm{m}$, where p is the number of pole pairs.

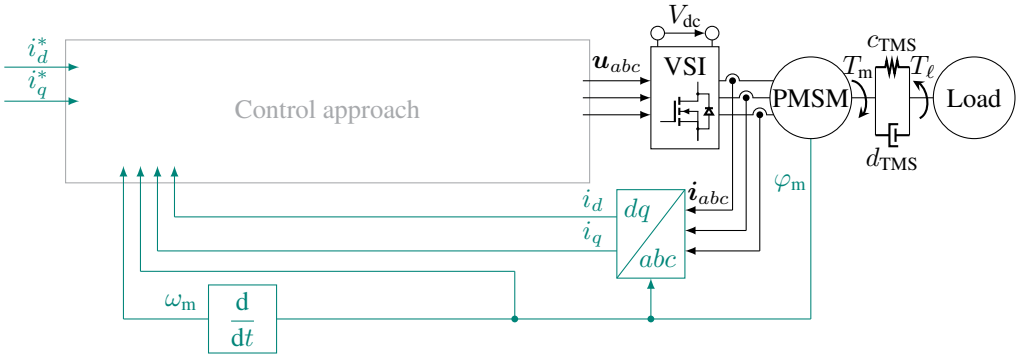


Figure 2.4: Current reference tracking for a two-level VSI with a PMSM.

2.2.1 Electrical Controlled System

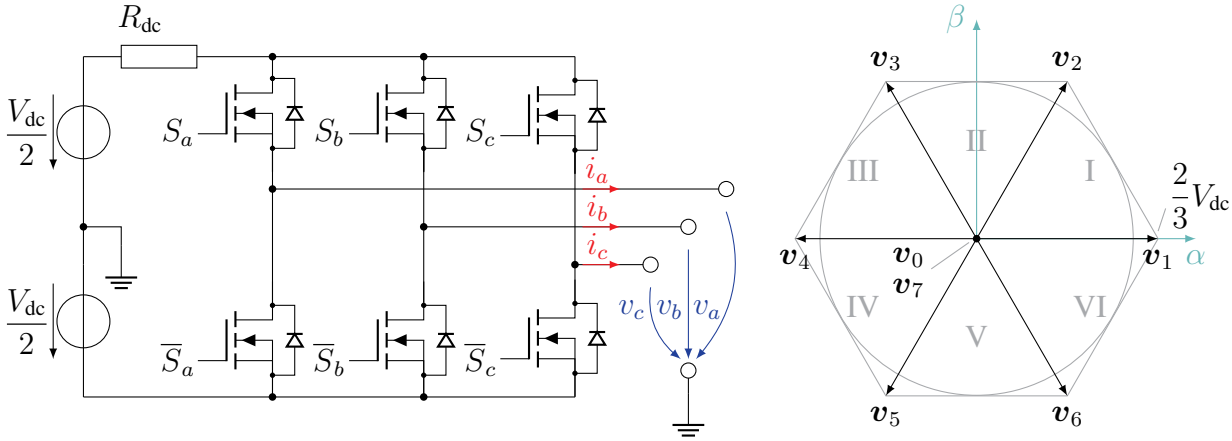
First of all, the considered inverter topology is presented. In the following, the nonlinear electrical machine, i.e., PMSM, is first derived in a relatively general way. Subsequently, a simplified model description is deduced, which forms the basis for state of the art MPC models.

2.2.1.1 Inverter

The drive system comprises a two-level voltage source inverter (VSI) with three phase legs that receives the control signals—optimized by a control algorithm on a calculation platform—and applies a corresponding voltage, see Fig. 2.5(a). Indirect control approaches use a modulator, while direct approaches directly manipulate the switches of the VSI. Given that the single-phase switch position assumes values $u_h \in \{-1, 1\}$, with $h \in \{a, b, c\}$, the output of the inverter in the dq-plane is

$$\boldsymbol{v}_{dq,\mathrm{inv}} = \frac{V_\mathrm{dc}}{2}\boldsymbol{K}(\varphi)\boldsymbol{u}_{abc} \tag{2.12}$$

where $\boldsymbol{u}_{abc} \in \{-1, 1\}^3$ is the three-phase, i.e., $n_h = 3$, switch position and V_dc the dc-link voltage. u_h results the position of the respective gate signals $S_h \in \{0, 1\}$ and the antivalent $\overline{S}_h \in \{0, 1\}$, where $u_h = 1$ gives $S_h = 1$, $\overline{S}_h = 0$ and $u_h = -1$ gives $S_h = 0$, $\overline{S}_h = 1$, see Fig. 2.5(a).



(a) Three phase two-level voltage source inverter.

(b) Voltage space vectors.

Figure 2.5: Structure of the three phase two-level voltage source inverter including the inverter switching states.

Table 2.1: Equivalence between voltage SVs and switch positions.

$\boldsymbol{v}_0 \equiv [-1\,-1\,-1]^T$, $\boldsymbol{v}_1 \equiv [\ 1\ -1\ -1]^T$, $\boldsymbol{v}_2 \equiv [\ 1\ \ 1\ -1]^T$, $\boldsymbol{v}_3 \equiv [-1\ \ 1\ -1]^T$
$\boldsymbol{v}_4 \equiv [-1\ \ 1\ \ 1]^T$, $\boldsymbol{v}_5 \equiv [-1\ -1\ \ 1]^T$, $\boldsymbol{v}_6 \equiv [\ 1\ -1\ \ 1]^T$, $\boldsymbol{v}_7 \equiv [\ 1\ \ 1\ \ 1]^T$

Note that the $2^3 = 8$ possible switch positions result in seven unique voltage space vectors (SVs), see Table 2.1 and Fig. 2.5(b) [HL03, p. 31]. Specifically, two of the switch positions, i.e., $\boldsymbol{u}_{abc} = [1\ 1\ 1]^T$ and $\boldsymbol{u}_{abc} = [-1\ -1\ -1]^T$, short circuit the load, and thus they correspond to two SVs, named zero SVs, that are redundant. The remaining combinations lead to unique SVs called active SVs. In the special case that both switches of the half-bridges are open, i.e., $S_h = 0$, $\overline{S}_h = 0$, this is described by $\boldsymbol{u}_{abc} = [Z\ Z\ Z]^T$, which is required, e.g., during parameter identification, see Section 4.1. As shown in Fig. 2.4, in a first approximation the VSI output voltage (2.12) is equal to the stator voltage $\boldsymbol{v}_{dq} = [v_d\ v_q]^T$, i.e., $\boldsymbol{v}_{dq,\text{inv}} \approx \boldsymbol{v}_{dq}$, where the contact resistance and the on-resistance of the semiconductor, i.e., $R_{\text{DS(on)}}$, are not considered separately.

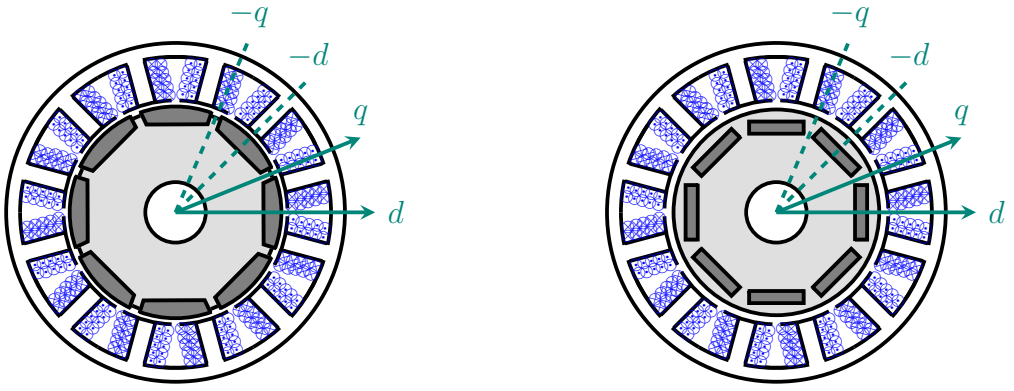
For the following considerations a comparison between fixed and variable switching frequency is necessary, which is why the average switching frequency is defined as [KG20]

$$f_{\text{sw}} = \lim_{M\to\infty} \frac{1}{m c_k M T_{\text{c}}} \sum_{\ell=0}^{M-1} ||\Delta \boldsymbol{u}_{abc}(\ell)||_1 \tag{2.13}$$

where $\Delta \boldsymbol{u}_{abc}(\ell) = \boldsymbol{u}_{abc}(\ell) - \boldsymbol{u}_{abc}(\ell - 1)$. In addition, M is the number of samples taken into account for the calculation, m is the number of the power semiconductor switches of the power converter of interest, i.e., $m = 6$ in the proposed case, and c_k is a converter topology dependent correction factor, i.e., $c_k = 2$ for a two-level VSI.

2.2.1.2 Nonlinear Model of the Permanent Magnet Synchronous Machine

The sinusoidal currents of the three stator windings generate a rotating magnetic field which couples the magnetic field of the rotor and leads to a mutual rotation. In contrast to asynchronous machines, the stator currents, the rotor shaft and the rotor flux linkage rotate synchronously in steady state. For the PMSM, a general distinction can be made between four different rotor types. The first two are shown in Fig. 2.6 with (a) surface mounted magnets (SPMSM), where the SM-PMSM has the magnets directly on the surface and the SI-PMSM the magnets embedded in the surface, and (b) interior, i.e., buried magnets [WB05, Sah08] (IPMSM), where the VI-PMSM has V-shaped buried magnets and the RI-PMSM has radial buried magnets [FHH10]. These typically use neodymium iron boron (NdFeB) magnets due to the high power density. The other two types are the permanent magnet assisted synchronous reluctance machine (PMASynRM), which uses ferrites instead of NdFeB due to costs, and the synchronous reluctance (SynRM), also called reluctance synchronous machine (RSM), which avoids magnets completely.
The main difference from the control perspective is the magnetic saliency—resulting in magnetic anisotropy—and thus the effect for producing torque. On the one side, the SPMSM produces the torque only if the mutual inductance changes as a function of the position, which results in a mutual (alignment) torque. On the other side, the SynRM produces the torque only if the self inductance changes as a function of the position, which results in a reluctance torque. The respective non-torque-forming effect is considered as a parasitic effect. This is similar to the cogging torque. The IPMSM and PMASynRM generate the torque using both principles, but with different intensity [Han12]. For small drives, however, the SynRM and PMASynRM have so far been rather uncommon and are thus not considered further. Most widely used is the SPMSM, where disadvantages like loosening of bonds for pasted magnets—which can block the rotor—are under control. Nevertheless, the IPMSM is becoming increasingly popular due to the lower costs, e.g., in manufacturing, since with buried magnets the automatic placement of



(a) SPMSM with four pole pairs.

(b) IPMSM with four pole pairs.

Figure 2.6: Comparison of different magnet arrangements in the rotor for the PMSM.

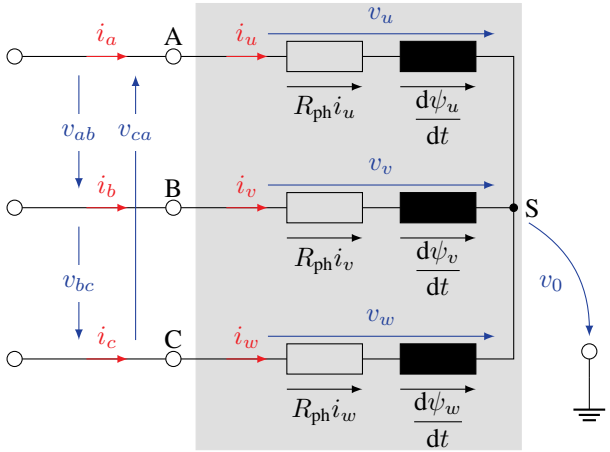


Figure 2.7: Symmetrical three-phase system with ohmic-inductive load in star (Y) connection. The star point voltage is assumed to be $v_0 \approx 0$.

the magnets in the rotor is much easier [WB05]. Further information on PMSM characteristics can be found, e.g., in [Sch15, p. 1069 ff.]. In general, a three-phase PMSM can be described in a simplified way as three star-connected stator windings with a resistance R_{ph} and the flux linkage ψ, see Fig. 2.7.[3] The Y-connection of the motor windings results in

$$i_a = i_u\,, \quad i_b = i_v\,, \quad i_c = i_w \tag{2.14}$$

for the phase currents and

$$v_a = v_u\,, \quad v_b = v_v\,, \quad v_c = v_w \tag{2.15}$$

for the phase voltages as well as

$$v_{ab} = v_a - v_b\,, \quad v_{bc} = v_b - v_c\,, \quad v_{ca} = v_c - v_a \tag{2.16}$$

for the terminal voltages. As illustrated in Fig. 2.7, if Ohm's law, Maxwell's law of induction and Kirchhoff's mesh rule are applied to the three motor phases, the phase voltages can be described by

$$\boldsymbol{v}_{abc}(t) = R_{\text{ph}} \boldsymbol{i}_{abc}(t) + \frac{\mathrm{d}\boldsymbol{\psi}_{abc}(t)}{\mathrm{d}t}\,, \tag{2.17}$$

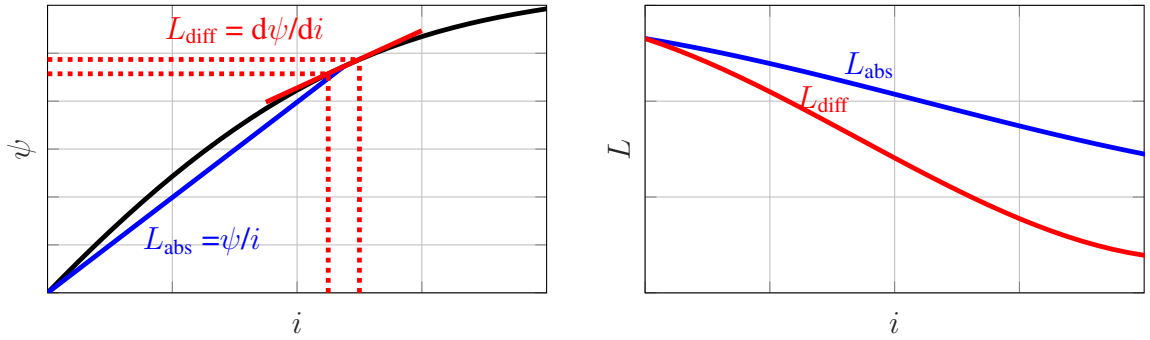
where $\boldsymbol{v}_{abc} = [v_a\ v_b\ v_c]^T$, $\boldsymbol{i}_{abc} = [i_a\ i_b\ i_c]^T$ and $\boldsymbol{\psi}_{abc} = [\psi_a\ \psi_b\ \psi_c]^T$. Respecting the rotor angle and by applying the coordinate transformation from Section 2.1.1, the general valid fundamental frequency model of the PMSM can be described in the dq-coordinate system by

$$\boldsymbol{v}_{dq}(t) = R_{\text{ph}} \boldsymbol{i}_{dq}(t) + \frac{\mathrm{d}\boldsymbol{\psi}_{dq}(t)}{\mathrm{d}t} + \omega_{\text{el}}(t) \boldsymbol{P} \boldsymbol{\psi}_{dq}(t) \tag{2.18}$$

[3] In this thesis, only star-connected windings are considered. Even if it is possible that the windings are connected in delta, this is not considered here, since so-called circular currents are generated.

where $\boldsymbol{v}_{dq} = [v_d\ v_q]^T$ and $\boldsymbol{i}_{dq} = [i_d\ i_q]^T$ are the stator voltage and current, respectively, $\boldsymbol{\psi}_{dq} = [\psi_d\ \psi_q]^T$ is the flux linkage. The flux linkage in the d-axis, in contrast to the q-axis, can be further separated by $\psi_d(t) = \psi_d^{\circledast}(t) + \psi_{\mathrm{pm}}(t)$, where ψ_{pm} is the permanent magnet flux linkage. Note that the last term of (2.18) comes by the chain rule in the derivation due to the space vector representation in the rotating dq-system [Sch13, p. 367], with $\boldsymbol{P} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Moreover, the generally valid PMSM model is dependent on numerous nonlinearities, which must be taken into account depending on their characteristics.

Saturation The magnetic saturation of the ferromagnetic rotor and stator steel sheets has a high influence on the operating behavior of the machine. The saturation behavior of ferromagnetic materials can be described by the model presentation of elementary magnets within a magnetic domain, also called Weiss domain [Ber98]. If an external magnetic field caused by the stator currents penetrates the material, the elementary magnets align along the external field and amplify it. Although the increase with current is initially linear, as more Weiss domains are aligned, the material becomes more saturated and the increase becomes more nonlinear, see Fig. 2.8(a). Within the electrical machine, the nonlinear saturation behavior of all coupled phases is superimposed and can thus be described in both stator and rotor fixed coordinate system. In order to consider the saturation in the introduced voltage equation (2.18), the flux linkage must be taken into account as a function of the prevailing d- and q-currents. Considering initially only self-saturation, the flux linkage in the d- and q-axis increases nonlinearly through the respective current in the d- and q-axis.



(a) Nonlinear current-flux linkage characteristic.

(b) Relationship between absolute and differential inductance.

Figure 2.8: Schematic representation of the nonlinear behavior of the magnetic circuit due to saturation.

Cross-coupling Despite self-saturation, it is usually assumed that the currents in the d- and q-axis each independently generate the d-and q-flux linkage components since the currents and voltages in the d- and q-axis are orthogonal to each other. However, in reality this is incorrect, since the magnetic circuits of the d- and q-axis cross and indirectly influence each other within

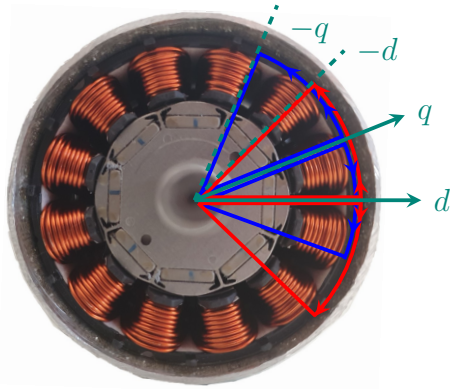
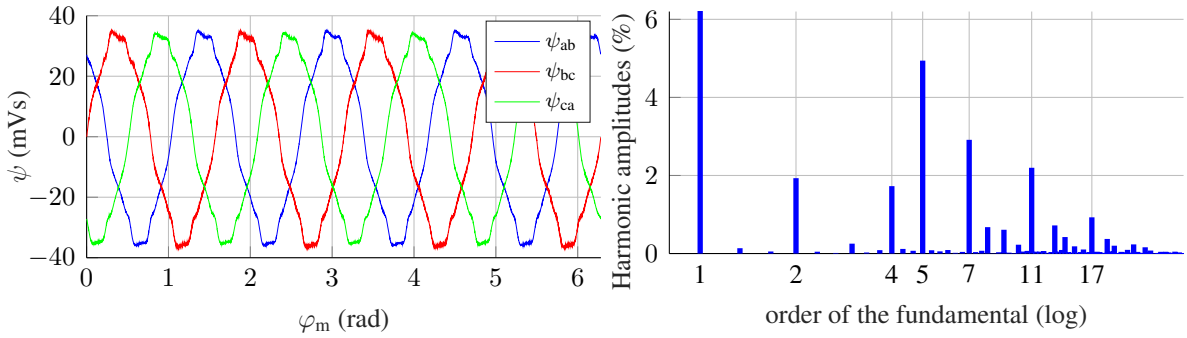


Figure 2.9: Cross-coupling due to shared flux linkage paths of d- (red) and q-axis (blue) using an IPMSM, i.e., motor M3 in Table A.3 of Appendix A.3.1, as an example.

the machine, see, e.g., [LL00, vvD$^+$03, MTSL08]. For illustration, Fig. 2.9 shows that they share the same path in the stator yoke. Depending on the machine segment, the magnetic circuits run in the same or opposite directions and thus reinforce or weaken each other. Hence, due to the nonlinear magnetic behavior of the stator iron, i.e., saturation, the effects of magnetization and demagnetization do not cancel each other out. Depending on the machine design, the influence of the cross-coupling can therefore vary considerably and is usually not negligible. In order to reflect also the cross-coupling of the d- and q-flux linkage in the machine model, a two-dimensional dependence on both current components, i.e., i_d and i_q, must be considered for ψ_d and ψ_q. The determination of the flux linkage—see Section 4.3—is an essential part of this work for reasons explained in Chapter 7.

Spatial harmonics The standard modeling approaches assume sinusoidal flux linkage in the air gap. However, as shown in Fig. 2.10, the experimentally measured behavior of the flux linkage, considered over one mechanical turn of the rotor angle φ_m, is mostly not sinusoidal. As can be seen, the 5^th, 7^th and 11^th spatial harmonics are usually clearly visible. Since the PMSM can also have a fractional slot winding, e.g., motor M4, the 2^nd and 4^th spatial harmonics are also present. Consequently, the flux linkage has a high proportion of harmonics in addition to the sinusoidal fundamental. As a consequence, this may result in current harmonics. Thus, apart from possible mechanical rotor eccentricities, it can be assumed that the electromagnetic conditions of the machine varies with the electrical rotor angle φ, as, e.g., also shown in [SA09, Min13].

The origin of these harmonics can have several reasons based on the machine topology. First, the permanent magnets can generate a non-sinusoidal magnetic field caused by their position on the rotor surface or within the rotor steel sheets. Second, the magnetic field in the air gap will be distorted by the transition between stator slots and stator teeth. Third, the winding concept has a significant influence as explained in more detail, e.g., in [Bin17, p. 61 ff.]. If these rotor angle



(a) Flux linkage for the three phases.

(b) Spatial harmonics averaged over all three phases.

Figure 2.10: Identified flux linkage over one turn of the mechanical rotor angle using an IPMSM with fractional slot non-overlapping concentrated windings, i.e., motor M4 in Table A.3 of Appendix A.3.1, as an example (experimental).

dependent flux linkage deviations are also taken into account for the machine model, the flux linkage characteristics must be extended by one more dimension and the multi-dimensional chain rule must be applied to the time derivative of the flux linkage. Consequently, the flux linkage has a three-dimensional nonlinear dependency on i_d, i_q and φ.

Temperature influence A further factor influencing the machine parameters and thus the operating behavior of the machine is the temperature. This affects both the phase resistance R_{ph} and the permanent magnet flux linkage ψ_{pm}. In general, the stator currents flowing through the (copper) windings—which are ohmic resistances—generate current heat losses, i.e., copper losses. As electrical energy is converted into thermal energy, the copper windings heat up, which changes the resistance due to the specific resistance of e.g. copper. Depending on the losses and the cooling concept, this results in different heating and cooling periods, which cause a change in the phase resistance during operation.

The same applies to ψ_{pm}, where it does not increase but decreases at higher temperatures. Here the temperature fluctuation results from the heating of the magnets during operation. This is caused by loss mechanisms, such as eddy currents within the magnets [GWB+11]. Since ψ_{pm} is part of the d-axis flux linkage, ψ_d is also affected by the temperature dependency. Fig. 2.11 exemplarily shows the experimentally determined dependence of ψ_{pm} on the temperature. For this purpose, the calculated inverter quantities, as described later in Section 4.1.2, were used. It was ensured that the motor was at thermal steady state for all measurements, i.e., the winding temperature ϑ was equal to the magnet temperature ϑ_{m}. NdFeB magnets typically lose $10\,\%$ flux per $100\,^{\circ}\mathrm{C}$ temperature increase [Spe14]. The trend line through the measuring points shows in good approximation this typical decrease, i.e., $7.3\,\%$ per $100\,^{\circ}\mathrm{C}$ in case of motor M1. Consequently, ψ_{pm} varies during operation depending on the operating point and the associated temperature. The influence of temperature on the control behavior is shown in Section 2.6.3.

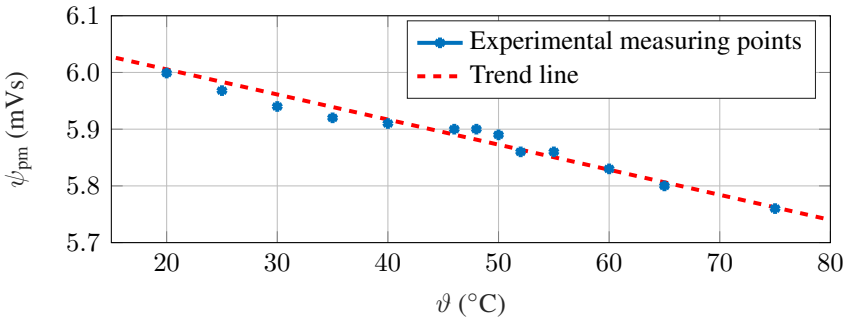


Figure 2.11: Influence of temperature on the flux linkage through the permanent magnets using the example of motor M1, see Table A.3 of Appendix A.3.1 (experimental).

Iron losses In the previous model descriptions only electrical losses due to the ohmic phase resistance are assumed. In a real machine, however, further losses occur. A part of the power loss is always caused by losses in the magnetic circuit of the machine [Jor24, Ber98, JHH+09]. These include eddy current losses, which occur due to induced currents within the rotor and stator lamination. In addition, magnetic hysteresis losses occur in the lamination packs, which are due to the constant remagnetization during commutation. In most cases, a separation of these loss mechanisms is unnecessary or not useful from the control perspective. Instead, they are summarized under the term iron losses P_{Fe}. As shown in Fig. 2.12, this term can be represented in a stationary equivalent circuit diagram by a resistance R_{Fe} connected in parallel. The latter can be different in d- and q-axis and depends mainly on i_d, i_q and the electrical frequency due to the mentioned loss mechanisms. As can be seen, the iron losses reduce the flux-forming currents $i_{d,\mathrm{m}}$ and $i_{q,\mathrm{m}}$, but still contribute to the losses over the resistance, i.e., copper losses. Using R_{Fe} and after several rearrangements explained e.g. in [Kel12, p. 71 ff.], the voltage equation (2.18) can be extended in steady state by an error voltage term $\Delta\boldsymbol{v}_{dq,\mathrm{Fe}} = [\Delta v_{d,\mathrm{Fe}}\ \Delta v_{q,\mathrm{Fe}}]^T$, leading to

$$\boldsymbol{v}_{dq}(t) = R_{\mathrm{ph}}\boldsymbol{i}_{dq}(t) + \omega_{\mathrm{el}}(t)\boldsymbol{P}\boldsymbol{\psi}_{dq}(t) + \Delta\boldsymbol{v}_{dq,\mathrm{Fe}}(t)\,. \tag{2.19}$$

Thus, the iron losses as a function of the steady-state quantities i_d, i_q and ω_{el} can be represented by four-dimensional maps of $\Delta v_{d,\mathrm{Fe}}$ and $\Delta v_{q,\mathrm{Fe}}$, where each map value is valid only for a particular steady-state operating point. For a detailed analytical description the interested reader is

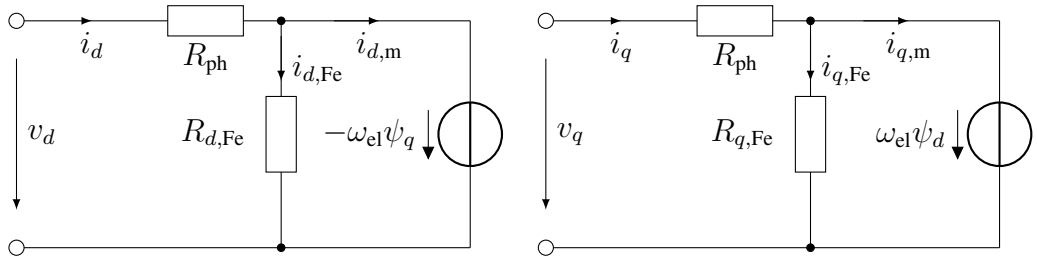


Figure 2.12: Stationary rotor fixed equivalent circuit diagram with iron loss resistance.

referred to [Kel12, p. 63 ff.], [RDD14]. From the control perspective, iron losses are important to consider since from the voltage that is applied, see (2.19), a part, i.e., $\Delta\boldsymbol{v}_{dq,\mathrm{Fe}}$, cannot be used. This means that these error voltages do not contribute to the flux linkage and should thus be known to the controller. Nevertheless, due to the aforementioned numerous dependencies, the invalidity of R_{Fe} in transients and the marginal influence on small PMSMs—demonstrated in Section 4.4.2—iron losses are not modeled in the following. Instead, they are taken into account by a more generic compensation term in the controller, see Chapter 8.

Model description Considering the mentioned influences, i.e., saturation, cross-coupling, rotor position and temperature dependency (i.e., the iron losses are not modeled), the voltage equation of the nonlinear model (2.18) can be described in more detail—particularly in the case of highly utilized synchronous machines—by

$$\boldsymbol{v}_{dq}(t) = R_{\mathrm{ph}}(\vartheta)\boldsymbol{i}_{dq}(t) + \frac{\mathrm{d}\boldsymbol{\psi}_{dq}(i_d, i_q, \varphi, \vartheta_{\mathrm{m}})}{\mathrm{d}t} + \omega_{\mathrm{el}}(t)\boldsymbol{P}\boldsymbol{\psi}_{dq}(i_d, i_q, \varphi, \vartheta_{\mathrm{m}})\,, \tag{2.20}$$

where $\psi_d(i_d, i_q, \varphi, \vartheta_{\mathrm{m}}) = \psi_d^{\circledast}(i_d, i_q, \varphi) + \psi_{\mathrm{pm}}(\varphi, \vartheta_{\mathrm{m}})$ and $\psi_q(i_d, i_q, \varphi)$.[4] Note that the respective flux linkages on the d- and q-axis depend on both currents and the rotor position φ, while ψ_{pm} varies with φ and the magnet temperature ϑ_{m}. The resistance varies with the temperature ϑ. More specifically, the flux linkage changes, i.e., the partial derivative of the flux linkage according to the mentioned nonlinearities, are given by [BB98, vvD+03, Ric16]

$$\frac{\mathrm{d}\psi_d(t)}{\mathrm{d}t} = \underbrace{\frac{\partial\psi_d^{\circledast}(i_d, i_q, \varphi)}{\partial i_d}}_{L_{dd}}\frac{\mathrm{d}i_d(t)}{\mathrm{d}t} + \underbrace{\frac{\partial\psi_d^{\circledast}(i_d, i_q, \varphi)}{\partial i_q}}_{L_{dq}}\frac{\mathrm{d}i_q(t)}{\mathrm{d}t} + \underbrace{\frac{\partial\psi_d(i_d, i_q, \varphi)}{\partial\varphi}}_{\Lambda_d\approx 0}\underbrace{\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t}}_{\omega_{\mathrm{el}}(t)} \tag{2.21a}$$

$$\frac{\mathrm{d}\psi_q(t)}{\mathrm{d}t} = \underbrace{\frac{\partial\psi_q(i_d, i_q, \varphi)}{\partial i_q}}_{L_{qq}}\frac{\mathrm{d}i_q(t)}{\mathrm{d}t} + \underbrace{\frac{\partial\psi_q(i_d, i_q, \varphi)}{\partial i_d}}_{L_{qd}}\frac{\mathrm{d}i_d(t)}{\mathrm{d}t} + \underbrace{\frac{\partial\psi_q(i_d, i_q, \varphi)}{\partial\varphi}}_{\Lambda_q\approx 0}\underbrace{\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t}}_{\omega_{\mathrm{el}}(t)}\,, \tag{2.21b}$$

where the derivative by temperature only affects ψ_{pm}, which happens much slower and is thus not represented. Here, the differential angular dependencies Λ_d and Λ_q can describe the change in the flux linkage components during the rotation of the rotor, under otherwise constant conditions. However, the influence of the rotor position can most often be neglected within a small tolerance band [GT18], i.e., $\Lambda_d \approx 0$, $\Lambda_q \approx 0$. As explained in Section 4.3.4 and Chapter 7, identifying rotor position dependencies is quite challenging for real-time applications, so they are better not to be modeled and instead accounted for by alternative possibilities, see, e.g., Chap-

[4] The described nonlinear model is realized in MATLAB/Simulink with the help of Simscape for the controlled system. Note that in Simscape the flux linkage is realized by LUTs, which use *linear* extrapolation and *smooth* interpolation. The latter is a *modified Akima* algorithm, which extends the *Akima* method by computing the mixed derivatives at the grid points using a weighted average of the finite differences, as described in [Aki74].

ter 8. Considering the remaining modeled nonlinearities in (2.21), the differential inductances describe the saturation and cross-coupling effects, which are becoming increasingly pronounced in highly utilized PMSMs. They correspond to the slope of the flux linkage characteristic for a specific combination of i_d and i_q and are thus represented by the tangent, see Fig. 2.8. L_{dd} and L_{qq} are the direct and the quadrature axis self-inductances defined by

$$L_{dd} = \frac{\partial \psi_d(i_d, i_q)}{\partial i_d} \qquad \text{and} \qquad L_{qq} = \frac{\partial \psi_q(i_q, i_d)}{\partial i_q} \,. \tag{2.22}$$

L_{dq} and L_{qd} are the direct- and the quadrature-axis cross-coupling (mutual) inductances defined by

$$L_{dq} = \frac{\partial \psi_d(i_d, i_q)}{\partial i_q} \qquad \text{and} \qquad L_{qd} = \frac{\partial \psi_q(i_q, i_d)}{\partial i_d} \,. \tag{2.23}$$

Since L_{dd} corresponds to the slope of the d-flux linkage in the direction of the d-axis and L_{qq} corresponds to the slope of the q-flux linkage in the direction of the q-axis, a decrease of L_{dd} and L_{qq} for higher currents indicates a self-saturation. Similarly, L_{dq} and L_{qd} represent the slopes of the d-flux linkage in the direction of the q-axis and the q-flux linkage in the direction of the d-axis, respectively. Their decrease at higher currents is an indicator for an existing cross-saturation. In some operating points the respective cross-coupling inductance can even reach $20\,\% - 40\,\%$ of the self-inductance [vvD+03],[Ric16, p. 50]. Using (2.22) and (2.23), the voltage equation (2.20) can be described by

$$\boldsymbol{v}_{dq}(t) = R_{\text{ph}}(\vartheta)\boldsymbol{i}_{dq}(t) + \boldsymbol{L}_{\text{Ddq}}\frac{\mathrm{d}\boldsymbol{i}_{dq}(t)}{\mathrm{d}t} + \omega_{\text{el}}(t)\boldsymbol{\Lambda}_{dq} + \omega_{\text{el}}(t)\boldsymbol{P}\boldsymbol{\psi}_{dq}(t) \tag{2.24}$$

where $\boldsymbol{L}_{\text{D}dq} = \begin{bmatrix} L_{dd} & L_{dq} \\ L_{qd} & L_{qq} \end{bmatrix}$ and $\boldsymbol{\Lambda}_{dq} = [\Lambda_d \; \Lambda_q]^T$ represents the differential angular dependency of the flux linkage.[5] As a consequence of the law of energy conversion, inductance matrices are symmetrical (i.e., $L_{dq} = L_{qd}$) [BB98] and positive semi-definite (e.g $\boldsymbol{x}^T\boldsymbol{L}_{\text{D}dq}\boldsymbol{x} \geq 0, \forall \boldsymbol{x} \in \mathbb{R}^2$) [Lan14, p. 22]. If saturation, cross-coupling and spatial harmonics are taken into account,

$$T_{\text{el}}(t) = \frac{3}{2}p\big((\psi_d(t) + \Lambda_q(t))i_q(t) - (\psi_q(t) - \Lambda_d(t))i_d(t)\big) \tag{2.25}$$

describes the electromagnetic torque calculation.[5] As mentioned before, spatial harmonics are usually neglected, e.g., due to their low impact or due to lack of processing power to consider them in real time. Consequently, the torque can be described sufficiently accurately by

$$T_{\text{el}}(t) \approx \frac{3}{2}p\big(\psi_d(t)i_q(t) - \psi_q(t)i_d(t)\big) \,. \tag{2.26}$$

[5] Although modeling of the rotor position dependence is omitted, it is included here for completeness.

2.2.1.3 Linear Model of the Permanent Magnet Synchronous Machine

The description of the PMSM can be simplified considerably if the magnetic circuit is additionally linearized. Thus, saturation and cross-coupling effects are neglected and the absolute inductances are introduced as a constant ratio between flux linkage and current in the respective axis, i.e., $\psi_d \approx \psi_{\text{pm}} + L_d i_d$ and $\psi_q \approx L_q i_q$. As shown in Fig. 2.8(a), the absolute inductances, i.e., L_d, L_q, each describe the gradient of an original straight line to any point of the flux linkage characteristic. While L_d and L_q can theoretically represent self-saturation, a constant value over the entire operating range—$L_d = L_q$ for SPMSM and $L_d \neq L_q$ for IPMSM—is usually assumed in the linear magnetic model. The salience is also taken into account as a constant factor, which is always aligned with the rotor position even under load. Although the comparison of absolute and differential inductances (see Fig. 2.8(b)) clearly shows that the latter describe the magnetic behavior more accurately [vvD+03], [Kel12, p. 21], [Min13, p. 35], absolute inductances are used in the design of most control methods. With these linearizations and the transformations of Section 2.1.1, (2.17) can be described in the rotating rotor coordinate system by

$$\boldsymbol{v}_{dq}(t) = R_{\text{ph}}\boldsymbol{i}_{dq}(t) + \boldsymbol{L}_{dq}\frac{\mathrm{d}\boldsymbol{i}_{dq}(t)}{\mathrm{d}t} + \omega_{\text{el}}(t)\left(\boldsymbol{P}\boldsymbol{L}_{dq}\boldsymbol{i}_{dq}(t) + \begin{bmatrix} 0 \\ \psi_{\text{pm}} \end{bmatrix}\right) \tag{2.27}$$

where $\boldsymbol{L}_{dq} = \operatorname{diag}(L_d, L_q)$. Consequently, (2.26) can be further simplified to

$$T_{\text{el}}(t) = \frac{3}{2}p\big(\psi_{\text{pm}} i_q(t) + (L_d - L_q)\, i_d(t) i_q(t)\big)\,. \tag{2.28}$$

For representing the controlled system, i.e., (2.27) and (2.28), as a linear time-invariant (LTI) system in the state space representation, an operating point linearization is necessary, e.g.,

$$\underbrace{\begin{bmatrix} \frac{\mathrm{d}i_d(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}i_q(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}\omega_{\text{m}}(t)}{\mathrm{d}t} \end{bmatrix}}_{\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t}} = \underbrace{\begin{bmatrix} -\frac{R_{\text{ph}}}{L_d} & \frac{L_q}{L_d}\omega_{\text{el},0} & 0 \\ -\frac{L_d}{L_q}\omega_{\text{el},0} & -\frac{R_{\text{ph}}}{L_q} & -\frac{\psi_{\text{pm}}}{L_q} \\ 0 & \frac{3p(\psi_{\text{pm}}+(L_d-L_q)i_{d,0})}{2J} & -\frac{d_{\text{fric}}}{J} \end{bmatrix}}_{\boldsymbol{A}_{\text{c}}} \underbrace{\begin{bmatrix} i_d(t) \\ i_q(t) \\ \omega_{\text{el}}(t) \end{bmatrix}}_{\boldsymbol{x}(t)}$$

$$+ \underbrace{\begin{bmatrix} \frac{1}{L_d} & 0 & 0 \\ 0 & \frac{1}{L_q} & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix}}_{\boldsymbol{B}_{\text{c}}} \underbrace{\begin{bmatrix} v_d(t) \\ v_q(t) \\ -T_{\text{fric}}(t) - T_\ell(t) \end{bmatrix}}_{\boldsymbol{u}(t)} \quad \text{and} \quad \underbrace{\begin{bmatrix} i_d(t) \\ i_q(t) \\ \omega_{\text{el}}(t) \end{bmatrix}}_{\boldsymbol{y}(t)} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{C}_{\text{c}}(t)} \boldsymbol{x}(t)\,, \tag{2.29}$$

where $\omega_{\text{el},0}$ and $i_{d,0}$ depend on the linearization, e.g., $\omega_{\text{el},0} = \omega_{\text{el,N}}$ and $i_{d,0} = 0\,\text{A}$. These system linearizations, e.g., the speed term, is necessary due to the requirement of some control approaches to have the controlled system as an LTI system[6]. However, in the following the word "linear" only refers to the magnetic circuit. T_ℓ is the load torque, T_{fric} the friction torque and J the moment of inertia. In general $\boldsymbol{u}(t)$ is the input vector with the control signals, $\boldsymbol{y}(t)$ the output vector, $\boldsymbol{A}_\text{c}$ the state matrix, $\boldsymbol{B}_\text{c}$ the input matrix and $\boldsymbol{C}_\text{c}$ the output matrix.
For estimations regarding the dynamics, the electrical time constant of the controlled system can be defined by $\tau_{\text{el}} \approx (L_d + L_q)/(2R_{\text{ph}}) \approx L_{\text{ph}}/R_{\text{ph}}$.

2.2.2 Mechanical Controlled System

Applying the balance of torques, as a starting point from the motor perspective, the equation of motion can be set up using Newton's basic equation of mechanics for rotational motion[7] by

$$\frac{J}{p}\frac{\mathrm{d}\omega_{\text{el}}}{\mathrm{d}t} = J\frac{\mathrm{d}\omega_{\text{m}}}{\mathrm{d}t} = T_{\text{el}} - T_{\text{fric}} - T_\ell\,. \tag{2.30}$$

2.2.2.1 Simplified Friction Model

There is a variety of friction models in the literature. A comprehensive overview is given, e.g., in [Rud12] and from a control perspective in [Hac12, p. 17 ff.]. To identify parameters of an easily computable model for MPC, this work focuses on three compatible basic friction models: the coulomb friction, the viscous friction and the breakaway torque model. The mentioned friction models are based on a simplified friction system as given in Fig. 2.13, containing the moment of inertia J, which is supported by k arbitrary friction contacts, that each generate

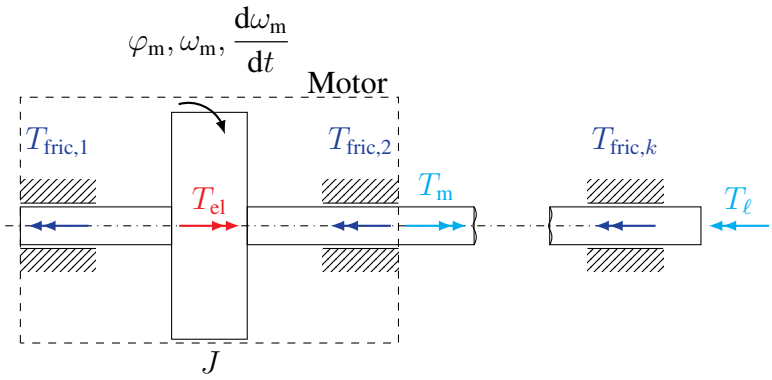


Figure 2.13: Simplified friction system.

[6]The theoretical nonlinearity due to the dq-transformation is not considered here.
[7]Cogging torque is not considered further.

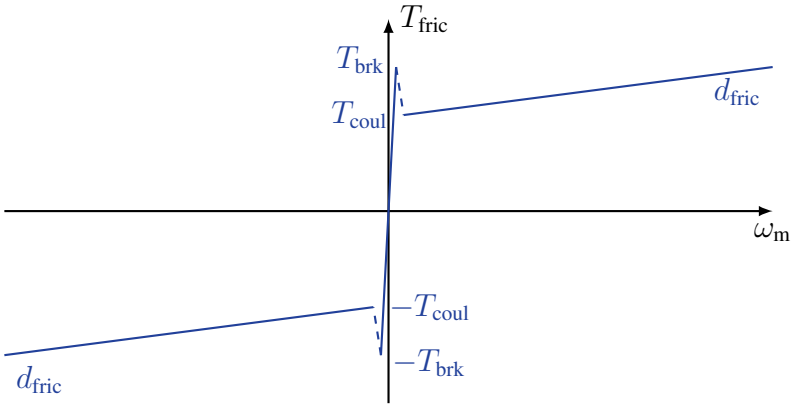


Figure 2.14: Simplified stick/slip friction model curve.

their own friction torque $T_{\text{fric},1}$, $T_{\text{fric},2}$,..., $T_{\text{fric},k}$. The single friction torques can be summed up to $T_{\text{fric}} = T_{\text{fric},1} + T_{\text{fric},2} + ... + T_{\text{fric},k}$. Contrary to these, the electromagnetic inner driving torque T_{el} is applied by exciting the motor windings. The breakaway torque T_{brk}, which characterizes the nonlinear start-up behavior, extends the coulomb (constant) T_{coul} and the viscous (speed dependent) model d_{fric} by considering two different friction coefficients for sticking and slipping. Hence, a simplified frictional torque model for MPC can be described in sufficient detail by

$$T_{\text{fric}} = \begin{cases} T_{\text{el}} & \text{if } (T_{\text{el}} \leq T_{\text{brk}}) \ \wedge \ (\omega_{\text{m}} \approx 0) \\ T_{\text{coul}} + \omega_{\text{m}} d_{\text{fric}} & \text{if } (T_{\text{el}} > T_{\text{coul}}) \ \wedge \ (\omega_{\text{m}} > 0) \end{cases} . \tag{2.31}$$

An illustration of the friction torque is given in Fig. 2.14—the rather nonlinear stribeck curve is not considered, i.e., not modeled, see dashed line—where the characteristic points T_{brk}, T_{coul} and the speed-dependent slope of the curve d_{fric} can be determined in steady state. In this case, the dynamic term $J\frac{\text{d}\omega_{\text{m}}}{\text{d}t}$ in (2.30) can be neglected and the motor torque can be considered equivalent to the friction torque. Moreover, for transients the mechanical time constant is of interest, which can be described by the coefficient of the derivation with the smallest order using (2.30) and (2.31), i.e.,

$$\tau_{\text{m}} = \frac{J}{d_{\text{fric}}} . \tag{2.32}$$

2.2.2.2 Drive Train Dynamics of a Two-Mass System

The elastically coupled drive train is usually described by a motor and load side, which are coupled by an elastic element, e.g., a transmission belt, a motor shaft or a gearbox. This is called two-mass system (TMS) or two-mass torsional oscillator, see Fig. 2.15. Although in some cases the drive train can be described as a multi-mass system, in the following the description

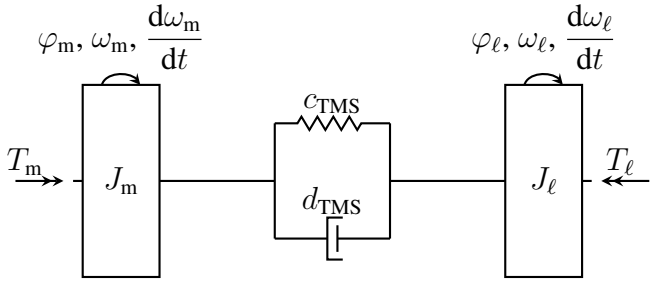


Figure 2.15: Schematic representation of a two-mass system.

as TMS is sufficient, as will be shown later. More detailed descriptions can be found, e.g., in [Vil07, Zou17],[DH16, p. 233 ff.]. The differential equation of motion for the motor side is given by

$$J_\mathrm{m}\frac{\mathrm{d}\omega_\mathrm{m}}{\mathrm{d}t} + d_\mathrm{TMS}(\omega_\mathrm{m} - \omega_\ell) + c_\mathrm{TMS}(\varphi_\mathrm{m} - \varphi_\ell) = T_\mathrm{m} \tag{2.33}$$

and for the load side by

$$J_\ell\frac{\mathrm{d}\omega_\ell}{\mathrm{d}t} + d_\mathrm{TMS}(\omega_\ell - \omega_\mathrm{m}) + c_\mathrm{TMS}(\varphi_\ell - \varphi_\mathrm{m}) = -T_\ell\,, \tag{2.34}$$

which can be determined by establishing the balance of torques on the motor and load side, respectively. The spring constant of the shaft, i.e., the torsional stiffness, c_TMS is calculated as the sum of the individual components involved. The viscous damping of the TMS can be described by using [DH16, p. 47]

$$d_\mathrm{TMS} = D\frac{2c_\mathrm{TMS}}{\underbrace{\sqrt{c_\mathrm{TMS}\frac{J_\ell + J_\mathrm{m}}{J_\ell J_\mathrm{m}}}}_{\omega_0}}\,, \tag{2.35}$$

where for steel shafts a damping ratio D of $0.02 - 0.08$ [DH16, p. 56] can be assumed, i.e., $D \approx 0.02$ in the following. Assuming that there is no additional torque acting on the load apart from the motor torque ($T_\ell \approx 0$), (2.33) and (2.34) can be transferred to the frequency domain. In doing so, the differential equations of motion are transferred by using the Laplace transformation, where the rotation angle φ_m and φ_ℓ are the integral of the speeds ω_m and ω_ℓ as well as the angular acceleration $\frac{\mathrm{d}\omega_\mathrm{m}}{\mathrm{d}t}$ and $\frac{\mathrm{d}\omega_\ell}{\mathrm{d}t}$ the time derivatives, i.e.,

$$J_\mathrm{m}\omega_\mathrm{m}(s)s + d_\mathrm{TMS}(\omega_\mathrm{m}(s) - \omega_\ell(s)) + \frac{c_\mathrm{TMS}}{s}(\omega_\mathrm{m}(s) - \omega_\ell(s)) = T_\mathrm{m}(s) \tag{2.36}$$

$$J_\ell\omega_\ell(s)s + d_\mathrm{TMS}(\omega_\ell(s) - \omega_\mathrm{m}(s)) + \frac{c_\mathrm{TMS}}{s}(\omega_\ell(s) - \omega_\mathrm{m}(s)) = 0\,. \tag{2.37}$$

If only the time-invariant solution for steady state is of interest, the complex Laplace operator s can be replaced by $j\omega$. The analytical transfer function $G_{\text{ana}}{=}\omega_{\text{m}}/T_{\text{m}}$ can be determined afterwards by eliminating ω_ℓ, which is based on inserting (2.37) into (2.36). Thus, G_{ana} can be described in dependence of the excitation frequency ω and the drive train parameters, i.e.,

$$G_{\text{ana}}(j\omega) = \frac{\omega_{\text{m}}(j\omega)}{T_{\text{m}}(j\omega)} = \frac{c_{\text{TMS}} + d_{\text{TMS}} j\omega + J_\ell (j\omega)^2}{c_{\text{TMS}} j\omega (J_{\text{m}} + J_\ell) + d_{\text{TMS}} (j\omega)^2 (J_{\text{m}} + J_\ell) + J_{\text{m}} J_\ell (j\omega)^3} \,. \tag{2.38}$$

If only small excitation frequencies are considered, the drive train can be assumed to be rigid, since during the limit value analysis, i.e., $\omega \to 0$, the transfer function approximates to the asymptote described by

$$\lim_{\omega \to 0} G_{\text{ana}}(j\omega) = \frac{1}{(J_{\text{m}} + J_\ell) j\omega} \,. \tag{2.39}$$

For the subsequent considerations, the damper and resonance frequency in the amplitude response must be distinguished. The damper frequency is the frequency at which the minimum change in the speed of the motor side can be observed despite excitation by an oscillating motor torque. The entire excitation energy thus goes into the load side. This frequency can be recognized as a local minimum in the amplitude response, since the amplitude ratio of $\omega_{\text{m}}/T_{\text{m}}$ becomes very small. The resonance frequency can be recognized as a local maximum in $|G_{\text{ana}}|$, i.e., the amplitude ratio of $\omega_{\text{m}}/T_{\text{m}}$ becomes very large. Thus, when excited by the motor torque at this frequency, the maximum speed change can be observed on the motor side (motor side resonance is equal to load side damper frequency). In the case of resonance excitation, the resulting speed amplitude can even become so large that the drive train can be damaged. Knowing this frequency and its dependence on the drive train parameters is thus of great importance.

As an illustration, Fig. 2.16 shows the transfer function G_{ana}—unchanged in black—of an analytically considered drive train. In order to get a better impression of the influence of the individual system parameters on the dynamics of a drive train, these are shown by way of example

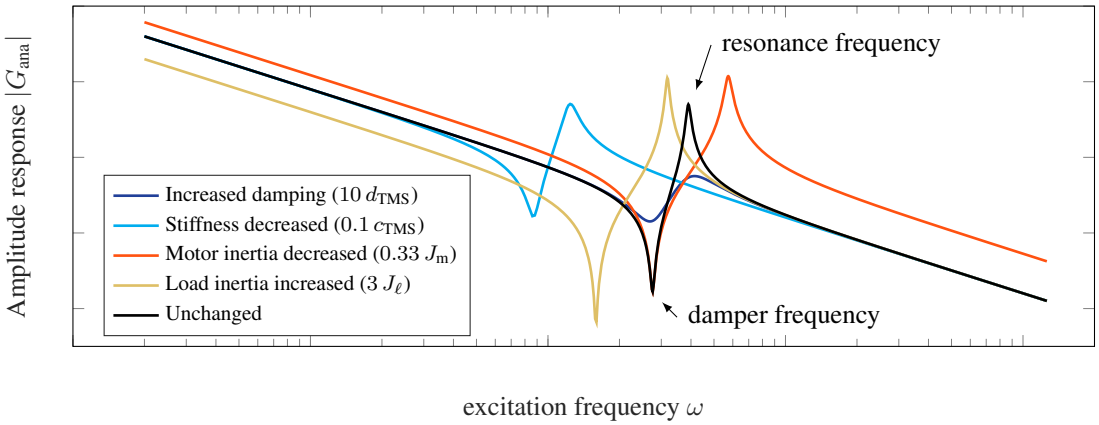


Figure 2.16: Schematic representation of parameter influence on G_{ana} with logarithmic scaling.

using several schematic amplitude responses of the transfer function with different parameter changes. If the load inertia changes, a shift in resonance and damper frequency can be seen (yellow). When the motor inertia changes, however, only the resonance frequency shifts (orange). If the stiffness changes, a shift of the two extrema can be detected (light blue). An increasing damping causes a reduction of the amplitude at resonance and damper frequency (dark blue).

2.3 Field-Oriented Control with PI Controller

Field-oriented control (FOC) is the most common control algorithm to control the previously explained controlled system. Consequently, it is used as a reference for comparison with the proposed MPC algorithms.

The principle of FOC aims at the separation of flux linkage and torque control in two separate SISO-systems. PI controllers are used to track the reference of each branch. Therefore, the transformation of Section 2.1.1 is applied to align the measured states to the rotor flux coordinate system. The basic principle can be described by Fig. 2.17. FOC is typically implemented as a cascaded control scheme. The inner loop implements the current and the outer loop the speed and flux linkage control. In doing so, the inner loop typically samples two to ten times faster than the outer loop. This comes from the fact, that the mechanical time constants are usually slower than the electrical time constants. Since common control hardware (see Chapter 3) works in the discrete-time domain, the current control can work quasi-continuous from the perspective of the speed controller. Clamping is used as anti-windup for all PI-based controllers. SVM, as described in Section 2.1.2, is used to apply the control signals.

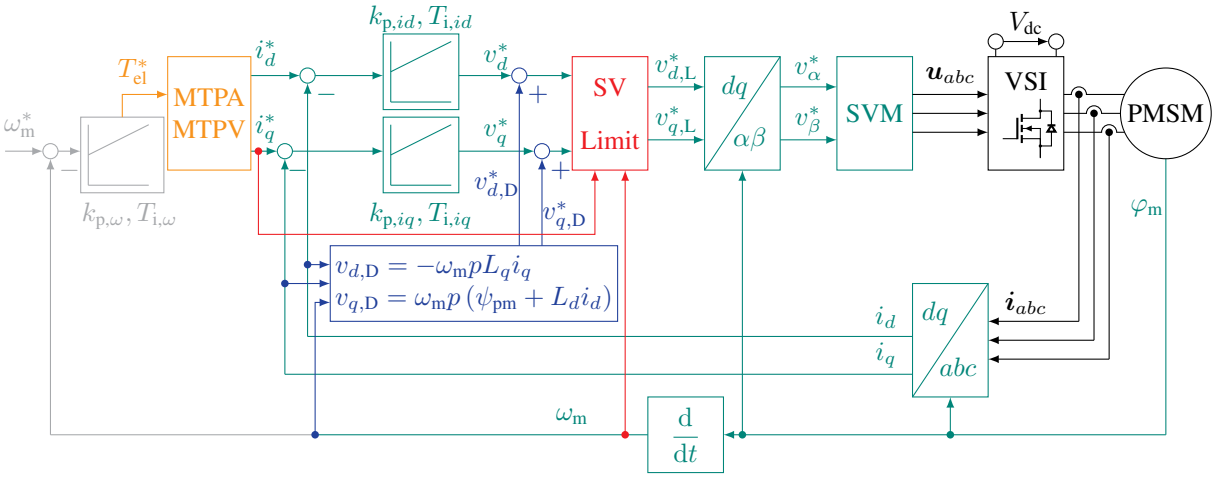


Figure 2.17: Block diagram of FOC with PI controllers and SVM. The current control, speed control, decoupling network [Sch15, p. 1118], space vector limitation [QD15, p. 183] and optional MTPA/MTPV [Sch15, p. 1095 ff.] are highlighted.

The essence of a proper FOC is a decoupling network that is as accurate as possible, with the PI controllers merely compensating for the remaining control deviation, see Fig. 2.17. The PI[8] gains of the current controller, i.e., proportional gains $k_{\mathrm{p},id}$, $k_{\mathrm{p},iq}$ and corresponding reset times of the integrator $T_{\mathrm{i},id}$, $T_{\mathrm{i},iq}$, are tuned according to the modulus optimum (pole-zero cancellation) criterion [Sch15, p. 50] [LW14, p. 515], where the controller can run up to the double of f_{sw}. The gains of the speed controller , i.e., proportional gain $k_{\mathrm{p},\omega}$ and corresponding reset time of the integrator $T_{\mathrm{i},\omega}$, use the symmetrical optimum [Sch15, p. 62] [LW14, p. 527]. These procedures represent adjustment rules for the continuous-time domain. However, since the smallest time constants in the respective controlled systems, i.e., τ_{el} for current and τ_{m} for velocity/position control, are usually at least a factor of 10 larger than the discrete control interval, a quasi-continuous system can be assumed and the factors mentioned above can be used.

FOC can be negatively affected mainly by two effects. First, by uncompensated delay times and second, by parameter changes as well as noisy measurements. Since parameter drifts and nonlinear behavior of the magnetic circuit are unavoidable for highly utilized PMSMs (see Section 2.2.1.2), specifically for one operating point calculated PI gains are valid only at this particular point. Consequently, PI gains and decoupling network parameters must be adjusted depending on the operating point to achieve acceptable performance over the entire operating range. Thus, a set of model parameters is required. Several researchers such as [Min13, Kel12, Gem15, Ric16] have successfully demonstrated how FOC can be adjusted to account for the aforementioned effects as accurately as possible, allowing for high dynamics with low overshoot during transient processes. Such an optimized FOC is compared in the following.

2.4 Direct Torque Control

Another well-known control strategy is direct torque control (DTC), which was invented by Takahashi/Noguchi [NT84, TN86] at the same time when Depenbrock invented a quite similar method [Dep84, Dep85]. As the name suggests, DTC is a direct control method, i.e., no modulator is required, instead the switching states of the power semiconductors are manipulated directly. The latter are determined by the comparison of the measured state to its reference, considering a given hysteresis band for the error. The hysteresis band and the nonlinearity of the system inherently lead to a variable switching frequency and thus generate a spread current spectrum. Although there are publications that introduce a modulator stage for DTC to achieve a fixed switching frequency, e.g., [MM98], this is not common and thus not considered here.

[8] A serial PI structure is used with $k_{\mathrm{p}}(1 + k_{\mathrm{i}}T_{\mathrm{c}}\frac{1}{z-1})$ in the discrete-time domain, where $k_{\mathrm{i}} = 1/T_{\mathrm{i}} = 1/\tau_{\mathrm{el}}$.

Using a given torque reference T_{el}^{*}, for a PMSM the flux linkage reference value can be calculated by using [Vas98, p. 230]

$$\left|\boldsymbol{\psi}_{dq}^{*}\right|\left(T_{\mathrm{el}}^{*}\right)=\sqrt{\psi_{\mathrm{pm}}^{2}+\left(\frac{2L_{q}}{3p\psi_{\mathrm{pm}}}T_{\mathrm{el}}^{*}(t)\right)^{2}}. \tag{2.40}$$

However, if a dominant reluctance effect is present, (2.40)—as described in [Vas98, p. 230]—must be redefined. The present flux linkage can be observed based on the linearized PMSM model, see (2.27), by using

$$\hat{\boldsymbol{\psi}}_{dq}=\begin{bmatrix}\psi_{\mathrm{pm}}\\0\end{bmatrix}+\boldsymbol{L}_{dq}\boldsymbol{i}_{dq} \tag{2.41}$$

and the present torque based on (2.26) by using

$$\hat{T}_{\mathrm{el}}=\frac{3}{2}p\left(\psi_{d}i_{q}-\psi_{q}i_{d}\right) \tag{2.42}$$

with the help of the measured currents, see Fig. 2.18. The disadvantage of the used current model compared to a voltage model—see, e.g., [Vas98, p. 188]—is the necessary angle φ for the dq-transformation. In addition, the inductances must be updated as soon as saturation arises and ψ_{pm} must be tracked as soon as a temperature change occurs. The advantage of the current model, however, is the significantly higher stability and the omission of a precise voltage measurement. The former can be justified by the fact that by using the current model, compared to the voltage model, no open integrator is required to estimate the flux linkage [Vas98, p. 226].[9]

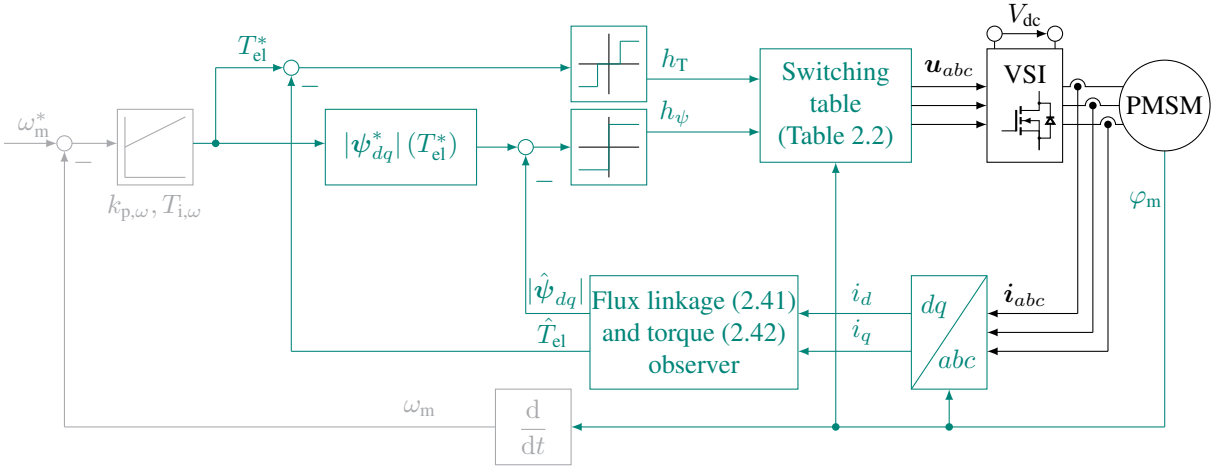


Figure 2.18: Block diagram of direct torque control.

[9] This open integrator can be stabilized by a first-order deceleration, but this leads to phase and amplitude errors when the drive is operated near the cut-off frequency of the deceleration [Mey10, p. 112]. Additionally, the voltage model uses R_{ph} to determine the flux linkage. Without a tracking of R_{ph} as a function of temperature, this leads to errors in the flux linkage estimation [Mey10, p. 111].

Table 2.2: Switching table for direct torque control (counterclockwise rotation of the flux is assumed) [Vas98, p. 515]. Note that the DTC sectors are rotated by $-\pi/6$ rad compared to the sectors in Fig. 2.5(b), see, e.g., [Vas98, p. 514].

$\mathrm{d}\psi$	$\mathrm{d}T_{\mathrm{el}}$	Sector I	Sector II	Sector III	Sector IV	Sector V	Sector VI
	1	$\boldsymbol{v}_2$	$\boldsymbol{v}_3$	$\boldsymbol{v}_4$	$\boldsymbol{v}_5$	$\boldsymbol{v}_6$	$\boldsymbol{v}_1$
1	0	$\boldsymbol{v}_7$	$\boldsymbol{v}_0$	$\boldsymbol{v}_7$	$\boldsymbol{v}_0$	$\boldsymbol{v}_7$	$\boldsymbol{v}_0$
	-1	$\boldsymbol{v}_6$	$\boldsymbol{v}_1$	$\boldsymbol{v}_2$	$\boldsymbol{v}_3$	$\boldsymbol{v}_4$	$\boldsymbol{v}_5$
	1	$\boldsymbol{v}_3$	$\boldsymbol{v}_4$	$\boldsymbol{v}_5$	$\boldsymbol{v}_6$	$\boldsymbol{v}_1$	$\boldsymbol{v}_2$
-1	0	$\boldsymbol{v}_0$	$\boldsymbol{v}_7$	$\boldsymbol{v}_0$	$\boldsymbol{v}_7$	$\boldsymbol{v}_0$	$\boldsymbol{v}_7$
	-1	$\boldsymbol{v}_5$	$\boldsymbol{v}_6$	$\boldsymbol{v}_1$	$\boldsymbol{v}_2$	$\boldsymbol{v}_3$	$\boldsymbol{v}_4$

Using the estimated system states, i.e., $\hat{T}_{\mathrm{el}}$ and $\hat{\boldsymbol{\psi}}_{dq}$, the control deviations can be calculated. The aim of DTC is to keep these deviations within adjustable hysteresis bands. Its common practice to use a three-point controller for the torque and a two-point controller for the flux linkage [Vas98, p. 225], [Gey17, p. 133]. In doing so, the torque control can use zero voltage SVs in the switching table. Consequently, the THD can be reduced for small hysteresis bands $|\Delta T_{\mathrm{el}}|$ without losing the dynamics in the control. The output of the torque controller, i.e., $\mathrm{d}T_{\mathrm{el}}$, is defined—counterclockwise rotation of the flux is assumed—by [Vas98, p. 515 ff.]

$$\mathrm{d}T_{\mathrm{el}} = \begin{cases} 1 & \text{if } \left|\hat{T}_{\mathrm{el}}\right| \leq |T_{\mathrm{el}}^*| - |\Delta T_{\mathrm{el}}| \\ -1 & \text{if } \left|\hat{T}_{\mathrm{el}}\right| \geq |T_{\mathrm{el}}^*| + |\Delta T_{\mathrm{el}}| \\ 0 & \text{else} \end{cases} . \tag{2.43}$$

The output of the flux linkage hysteresis controller is defined by [Vas98, p. 515]

$$\mathrm{d}\psi = \begin{cases} 1 & \text{if } \left|\hat{\boldsymbol{\psi}}_{dq}\right| \leq \left|\boldsymbol{\psi}_{dq}^*\right| - \left|\Delta\boldsymbol{\psi}_{dq}\right| \\ -1 & \text{if } \left|\hat{\boldsymbol{\psi}}_{dq}\right| \geq \left|\boldsymbol{\psi}_{dq}^*\right| + \left|\Delta\boldsymbol{\psi}_{dq}\right| \end{cases} . \tag{2.44}$$

ΔT_{el} and $\Delta\boldsymbol{\psi}_{dq}$ are used to define the hysteresis bands. The smaller the band, the lower the ripple of the flux linkage and torque, which also means that the switching frequency increases [Mey10, p. 119].

The optimal switching signal for the VSI is selected afterwards on the basis of the switching Table 2.2, whereby a voltage SV is applied over the entire control interval.

In the case that speed control is required, a PI-based controller is superimposed on DTC in this thesis, similar to FOC. So far, there is no known closed-loop control description as an LTI system when using DTC. For this reason, it is not possible to set the PI gains of the speed controller according to a symmetrical optimum or absolute optimum. Consequently, the Ziegler and Nichols method is applied. In doing so, the gain is increased until the stability limit is reached, where the parameters of the controller can be set using the table in [LW14, p. 494].

2.5 Model Predictive Control

Model predictive control is one of the most promising control algorithms for the increasing demands on performance and efficiency of power converters and drives, taking into account the real nature of these types of systems [Wan09, p. 43], [RC12, p. 3], [Gey17, p. 20]. Power converters and drives are nonlinear systems of a hybrid nature, including linear and nonlinear parts and a finite number of switching devices. The characteristics of the power converters and drives, as well as the characteristics of the control platforms used to form the control, converge in a natural way to the application of model predictive control, as summarized in Fig. 2.19.
The idea of MPC was born in the petrol-chemical industry 1979 by Cutler and Ramaker [CR79, Mor09]. Based on Dynamic Matrix Control and Generalized Predictive Control, MPC combines and extends the advantages of both methods. A distinction can be made between offline, i.e., explicit [TJB03, DP04, Leu14], MPC and online, i.e., implicit, MPC, whereby only online MPC is considered in the following. In the latter, the entire model is calculated in real time. Model predictive control, more specifically CCS-MPC and FCS-MPC, have become increasingly popular in recent years [CKK$^+$08, KPR$^+$15], with a good overview given in [CKK$^+$08, KLGK20]. Even if both approaches have their reason for being, especially FCS-MPC has gained a large momentum during the last years, as mentioned in Section 1.1. The direct optimization of the gate signals with the characteristic dead-beat behavior during transients, the direct inclusion of several hard and soft constraints as well as the optimization of the current performance, when using a long-horizon, are just some of the reasons for utilizing FCS-MPC. Only the finite number of possible switching states of the converter are considered during optimization, which respects the discrete nature of power converters and avoids the need for modulation. CCS-MPC, on the contrary, optimizes a continuous control signal, thus requiring a modulator (see Section 2.1.2) and hence separating model optimization and switching signal generation.

In general, both MPC schemes measure the actual states of the controlled system at a discrete time k, calculate an analytical model of the controlled system several steps into the future and, taking all steps into account, optimize the control signal with respect to a user-defined cost

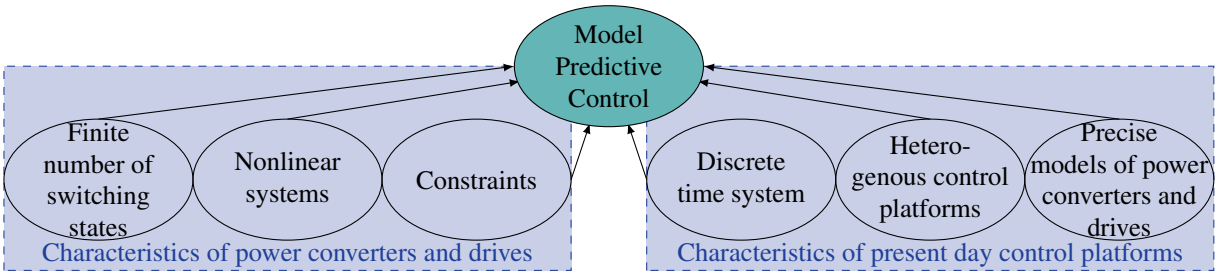


Figure 2.19: Characteristics of power converters and drives that make predictive control a natural solution [RC12, p. 14].

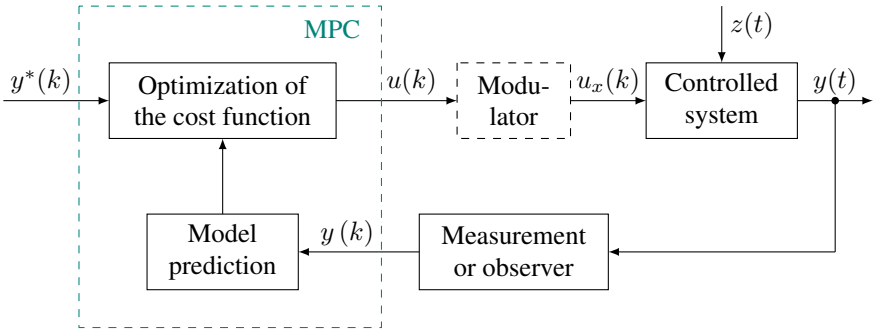


Figure 2.20: Block diagram of MPC.

function, i.e., optimization function, including different constraints. Consequently, in contrast to the above-mentioned FOC and DTC, MPC offers a MIMO optimization, i.e., can control several objectives at the same time, thus avoiding complex and probably suboptimal interactions between multiple SISO controllers. In doing so, apart from the discrete-time interval, the cost function[10] is the only tuning possibility of this control method.

The basic principle can be described by Fig. 2.20. The optional modulator is usually only available with CCS-MPC.[11] When talking about long horizon MPC, a prediction horizon $N_\mathrm{p} > 1$ is used. The flexibly adjustable discrete-time horizon can be visualized as demonstrated in Fig. 2.21. With a longer time horizon a stronger predictive character of the controller appears. In general, optimal controllers, e.g., MPC, with infinite horizon stabilize themselves, where a suitable Lyapunov function for establishing stability is the value function associated with the problem of optimal controllers with infinite horizon [MRRS00, p. 792]. Lyapunov is primarily used for MPC, since most other stability criteria apply only to linear control methods. In [AQ11, AQ13, AQ15] practical stability is proved with the Lyapunov theorem for linearized models, i.e., LTI, with integer inputs, i.e., for FCS-MPC, which is also valid for short horizons.

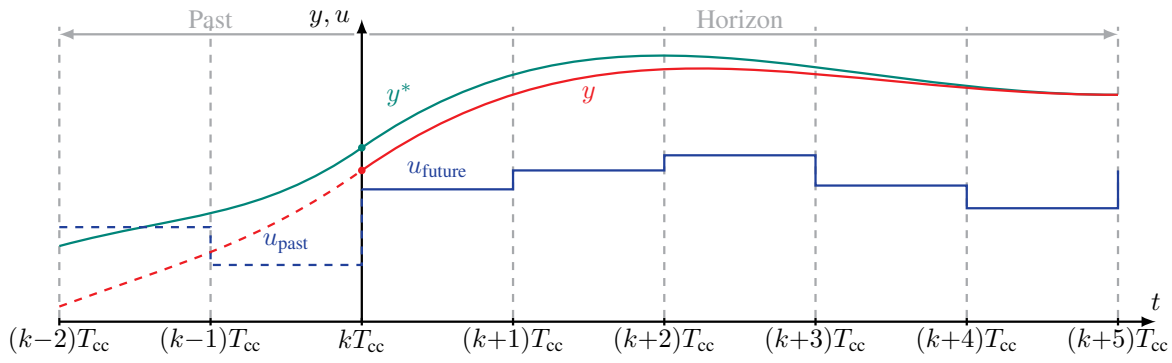


Figure 2.21: CCS-MPC with receding horizon policy for a five-step prediction horizon (N_p=5). u is the continuous control signal, y is the controlled signal and y^* is the reference.

[10] This includes the prediction horizon and the weighting factors in multi-criterion optimization.

[11] There are a few adaptations of FCS-MPC that also introduce a modulator. Some of them are mentioned in Chapter 6.

However, stability is inherently difficult to study and prove for FCS-MPC with nonlinear model, although a longer horizon tends to increase stability [BM15], [Gey17, p. 19]. Hereinafter, stability will not be discussed further; instead, reference is made to the overview in [KLGK20].
In addition, the most important definitions in the field of MPC are briefly summarized for a better understanding:

1. Moving horizon window: a time dependent window from an arbitrary discrete time step k to $k+N_\mathrm{p}$. The length of the window N_p—measured by the number of time steps—remains constant, but the window moves on by time.
2. Prediction horizon: corresponds to the length of the moving horizon window N_p and dictates how far the future states should be predicted.
3. Receding horizon policy: although the optimal trajectory of future control signals is calculated completely for the moving horizon window, only the control signal for the next discrete step is given out, while the remaining trajectory is neglected.
4. Control horizon: denoted by N_c and measured by the number of discrete steps, indicates how long in CCS-MPC the control signal should take before it reaches steady state.
5. Constraints: The controller can directly consider various constraints during optimization. Some of them are system-related, e.g., the maximum output voltage of the inverter, others are imposed for safety reasons, e.g., a maximum current to protect the inverter or the load. In general, any mathematically describable constraint can be considered, i.e., added in the cost function.
6. Long-horizon or multistep MPC: Solving the associated optimal control problem using a prediction horizon greater than one, i.e., $N_\mathrm{p} > 1$. The prediction for the delay time compensation—caused by the real-time system—is not included in the number of steps.

For comparison, it is useful to clearly state the main differences between CCS-MPC and FCS-MPC when considering the electrical controlled system (Section 2.2.1), see Table 2.3. The last three criteria are only applicable if an inverter is controlled by the control signal. As will be shown in Chapter 9, this is obviously needless as soon as CCS-MPC is used as a superimposed controller.

Table 2.3: Comparison of CCS-MPC and FCS-MPC.

	CCS-MPC	FCS-MPC
Problem	quadratic programming (QP)	(mixed) integer quadratic programming ((M)IQP)
Constraints	can be included	can be included
Modulator	requires a modulator	no modulator
Switching frequency	fixed	usually variable
Direct influence on the switching states	no, this is done by the modulator	yes

2.5.1 Continuous Control Set-MPC

CCS-MPC, as the name implies, optimizes a continuous control signal, i.e., a voltage in the case considered first (although a torque is optimized in the later stage of the thesis, see Chapter 9), taking into account the entire horizon. The basic principle is exemplified in Fig. 2.22. In the majority of cases the use of CCS-MPC is based on two arguments. First, due to the usually reduced computational load, even if operating point-dependent linearizations must most often be accepted for this. Second, a continuous control signal is optimized, which in case of an optimized voltage requires a modulator for switching signal generation, resulting in a naturally fixed switching frequency. In general, a distinction can be made between unconstrained and constrained solution. The need for constraints can be explained very simple by understanding that the controlled signal, e.g., applied voltage, is limited by the maximum available dc-link voltage. As descriptively shown in [Wan09, p. 43 ff.], instead of simply saturating the control signal after optimization, i.e., truncate the signal, the performance is dramatically increased by considering the boundary conditions already during optimization. Given this, first, Section 2.5.1.3 introduces the unconstrained solution using an analytical solution. Second, Section 2.5.1.4 shows the numerical calculation of the constrained solution based on an active-set method.

Linear or nonlinear CCS-MPC CCS-MPC depends strongly on whether a nonlinear controlled system is considered or whether a limitation of the solution space, i.e., a linear controlled system or an operating point linearization, is possible. In general, CCS-MPC approaches use a linearized state space description since the approach is based on an LTI system. Furthermore, almost all solvers are also based on an LTI system [KZPF15]. However, since there are also nonlinear approaches, a distinction can be made between linear and nonlinear CCS-MPC in the literature. CCS-MPC optimized for nonlinear control systems can be found, e.g., in [Rau03], where a CCS-MPC with online learning state space models is presented to consider isolated static nonlinearities, e.g., friction, during optimization. Alternatively, even if this is an explicit approach, it is also possible to switch online between different linear models depending on the

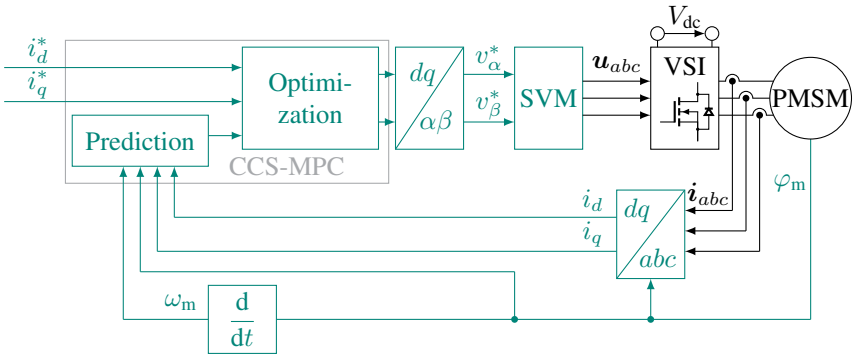


Figure 2.22: Block diagram of CCS-MPC.

operating point [PPS+16]. In [Sae15] discrete voltage test vectors are applied iteratively, i.e., by varying the angle and length, to approach a continuous position in the voltage hexagon and then output this position as a continuous voltage vector. Another method is presented in [WAB16], which uses particle swarm optimization where a first optimization routine scans fixed points in the hexagon and then searches with further subroutines in smaller areas close to the ideal solution to optimize a continuous control signal. However, it must be said that the latter procedures are usually quite complex and hardly intuitive. Although linear CCS-MPC is also used initially in this thesis, the idea of [Rau03] for nonlinear CCS-MPC is taken up in Section 9.2.3.

2.5.1.1 Quadratic Programming Formulation

With regard to CCS-MPC, the cost function—including the control targets—is defined as a quadratic optimization problem which is solved by using quadratic programming (QP). In general the quadratic problem is defined by

$$\begin{aligned} &\underset{\boldsymbol{\eta}\in\mathbb{R}^n}{\text{minimize}}\ J_\mathrm{c} = \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{H}\boldsymbol{\eta} + \boldsymbol{\eta}^T\boldsymbol{\Theta} \\ &\text{subject to}\ \ \boldsymbol{M}\boldsymbol{\eta} \leq \boldsymbol{\gamma} \end{aligned} \tag{2.45}$$

where $\boldsymbol{\eta}$ is a vector with n-elements in which the number of elements depends on the number of control signals to be optimized, i.e., decision variables. $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ is a symmetrical matrix including the control targets and the vector $\boldsymbol{\Theta} \in \mathbb{R}^n$ determines the initial state of the system. As mentioned before, constraints can be defined, where the matrix $\boldsymbol{M}$ defines the relation between the control signals $\boldsymbol{\eta}$ and the boundaries of the optimization problem $\boldsymbol{\gamma}$. $\boldsymbol{M} \in \mathbb{R}^{m\times n}$, where m is the number of constraints.

2.5.1.2 Linearization and Discretization of the Controlled System

The above-mentioned cost function requires an LTI system. Using e.g. the model of the controlled system in Section 2.2.1.3, it can be described—operating point linearized—as LTI system in the state-space representation as shown in (2.29).
To be able to use the LTI system in a discrete-time calculation platform, the state space description must be transferred by means of a Z-transform. For CCS-MPC—including its real-time implementation described in Chapter 3—this is done by means of exact discretization, using T_cc as sampling interval, see Section 2.1.3. Here, the procedure in (2.9) terminates after $l = 10$, although l is flexible and usually a termination after $l = 2$ up to $l = 3$ elements is already sufficiently accurate for the determination of the discrete-time matrices.

2.5.1.3 Unconstrained Solution by using First Derivation

A solution space without side conditions, i.e, constraints, is considered first, see, e.g., [Wan09, p. 1-42], [WDK17c, WLH+18], which is used as starting point for the constrained solution in Section 2.5.1.4. Neglecting constraints, the solution can be easily determined analytically. More in detail, to solve the quadratic optimization problem, the first derivative of the unconstrained cost function $J_{\text{c,unc}}$ is formed and set to zero. Therefore, the optimization problem must be defined, for which the discrete-time matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are first extended by the respective control targets—added in additional rows—to achieve an integral action for CCS-MPC [Wan09, p. 23], resulting in an augmented state space model, i.e., $\boldsymbol{A}_\text{e}, \boldsymbol{B}_\text{e}, \boldsymbol{C}_\text{e}$, with

$$\overbrace{\begin{bmatrix}\Delta \boldsymbol{x}\left(k+1\right)\\ \boldsymbol{y}\left(k+1\right)\end{bmatrix}}^{\boldsymbol{x}_\text{e}(k+1)} = \overbrace{\begin{bmatrix}\boldsymbol{A} & \boldsymbol{0}_{n_x\times n_y}\\ \boldsymbol{C}\boldsymbol{A} & \boldsymbol{I}_{n_y}\end{bmatrix}}^{\boldsymbol{A}_\text{e}} \overbrace{\begin{bmatrix}\Delta \boldsymbol{x}\left(k\right)\\ \boldsymbol{y}\left(k\right)\end{bmatrix}}^{\boldsymbol{x}_\text{e}(k)} + \overbrace{\begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{C}\boldsymbol{B}\end{bmatrix}}^{\boldsymbol{B}_\text{e}} \Delta \boldsymbol{u}_\text{unc}\left(k\right) \tag{2.46a}$$

$$\boldsymbol{y}\left(k\right) = \underbrace{\begin{bmatrix}\boldsymbol{0}_{n_x\times n_y}^T & \boldsymbol{I}_{n_y}\end{bmatrix}}_{\boldsymbol{C}_\text{e}} \begin{bmatrix}\Delta \boldsymbol{x}\left(k\right)\\ \boldsymbol{y}\left(k\right)\end{bmatrix} \tag{2.46b}$$

where $\boldsymbol{I}_{n_y} \in \mathbb{N}^{n_y\times n_y}$ and $\boldsymbol{0}_{n_x\times n_y} \in \mathbb{N}^{n_x\times n_y}$ with n_x being the number of states, n_y being the number of states to be optimized (e.g., n_y=2 when i_d and i_q in (2.29) are optimized) and n_u being the number of control signals. Including these states, the augmented model can be used to optimize the output of the system, i.e., $\boldsymbol{y}$, based on the change of the states, i.e., $\Delta\boldsymbol{x}$, by varying the system input, i.e., control signal, $\Delta\boldsymbol{u}_\text{unc}$. $\Delta\boldsymbol{u}_\text{unc}$ is used instead of $\boldsymbol{u}_\text{unc}$, as the rate of change can emulate integrator behavior that MPC normally lacks due to its proportional nature.
Using the extended state space model, the system behavior is calculated for several steps into the future with [Wan09, p. 26]

$$\begin{aligned}
\boldsymbol{x}_\text{e}(k+1|k) &= \boldsymbol{A}_\text{e}\boldsymbol{x}_\text{e}(k) + \boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k)\\
&\vdots\\
\boldsymbol{x}_\text{e}(k+N_\text{p}|k) &= \boldsymbol{A}_\text{e}^{N_\text{p}}\boldsymbol{x}_\text{e}(k) + \boldsymbol{A}_\text{e}^{N_\text{p}-1}\boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k) + \boldsymbol{A}_\text{e}^{N_\text{p}-2}\boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k+1)\\
&\quad + \cdots + \boldsymbol{A}_\text{e}^{N_\text{p}-N_\text{c}}\boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k+N_\text{c}-1)
\end{aligned} \tag{2.47}$$

for the states and $\boldsymbol{y}(k) = \boldsymbol{C}_\text{e}\boldsymbol{x}_\text{e}(k)$, i.e.,

$$\begin{aligned}
\boldsymbol{y}(k+1|k) &= \boldsymbol{C}_\text{e}\boldsymbol{A}_\text{e}\boldsymbol{x}_\text{e}(k) + \boldsymbol{C}_\text{e}\boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k)\\
&\vdots\\
\boldsymbol{y}(k+N_\text{p}|k) &= \boldsymbol{C}_\text{e}\boldsymbol{A}_\text{e}^{N_\text{p}}\boldsymbol{x}_\text{e}(k) + \boldsymbol{C}_\text{e}\boldsymbol{A}_\text{e}^{N_\text{p}-1}\boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k) + \boldsymbol{C}_\text{e}\boldsymbol{A}_\text{e}^{N_\text{p}-2}\boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k+1)\\
&\quad + \cdots + \boldsymbol{C}_\text{e}\boldsymbol{A}_\text{e}^{N_\text{p}-N_\text{c}}\boldsymbol{B}_\text{e}\Delta\boldsymbol{u}_\text{unc}(k+N_\text{c}-1)
\end{aligned} \tag{2.48}$$

for the output signals based on the control horizon N_c within the prediction horizon N_p, similar to the functionality of an auto-regressive-moving average model (ARMA model) [SB17, p. 552 ff.]. Moreover, $N_\mathrm{c} \leq N_\mathrm{p}$ applies when selecting the control horizon. Currently there is no general rule for selecting the horizon or weighting factors [RKE+13]. Theoretically, however, it would make sense to cover the entire transient process, i.e., the time window within which an optimization should be performed, by choosing the prediction horizon.
The denoted future states $\boldsymbol{x}_\mathrm{e}(k+1|k)$, $\boldsymbol{x}_\mathrm{e}(k+2|k)$, $\cdots$, in which $\boldsymbol{x}_\mathrm{e}(k+2|k)$ is the predicted state variable at $k+2$ based on a current state variable $\boldsymbol{x}_\mathrm{e}(k)$, describe the predicted system behavior at a certain point in time and are used to predict the system output with

$$\boldsymbol{Y}(k) = \boldsymbol{\Gamma}_\mathrm{c}\boldsymbol{x}_\mathrm{e}(k) + \boldsymbol{\Upsilon}_\mathrm{c}\Delta\boldsymbol{U}_\mathrm{unc}(k)\,, \tag{2.49}$$

where $\boldsymbol{Y}(k)$ contains the predicted output signals, e.g., i_d, i_q, and $\Delta\boldsymbol{U}_\mathrm{unc}(k)$ contains the predicted change of the input signals, e.g., v_d, v_q. The predictive state matrices are described by

$$\boldsymbol{\Gamma}_\mathrm{c} = \begin{bmatrix} \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e} \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^2 \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^3 \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^4 \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^5 \\ \vdots \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^{N_\mathrm{p}} \end{bmatrix}, \quad \boldsymbol{\Upsilon}_\mathrm{c} = \begin{bmatrix} \boldsymbol{C}_\lambda\boldsymbol{B}_\mathrm{e} & \boldsymbol{0}_{n_y\times n_u} & \cdots & \boldsymbol{0}_{n_y\times n_u} \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}\boldsymbol{B}_\mathrm{e} & \boldsymbol{C}_\lambda\boldsymbol{B}_\mathrm{e} & & \vdots \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^2\boldsymbol{B}_\mathrm{e} & \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}\boldsymbol{B}_\mathrm{e} & & \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^3\boldsymbol{B}_\mathrm{e} & \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^2\boldsymbol{B}_\mathrm{e} & & \boldsymbol{C}_\lambda\boldsymbol{B}_\mathrm{e} \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^4\boldsymbol{B}_\mathrm{e} & \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^3\boldsymbol{B}_\mathrm{e} & & \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}\boldsymbol{B}_\mathrm{e} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^{N_\mathrm{p}-1}\boldsymbol{B}_\mathrm{e} & \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^{N_\mathrm{p}-2}\boldsymbol{B}_\mathrm{e} & \cdots & \boldsymbol{C}_\lambda\boldsymbol{A}_\mathrm{e}^{N_\mathrm{p}-N_\mathrm{c}}\boldsymbol{B}_\mathrm{e} \end{bmatrix}, \tag{2.50}$$

where $\boldsymbol{\Upsilon}_\mathrm{c}$ is a Toeplitz-matrix. For the weighting of the individual states in the optimization, the output matrix $\boldsymbol{C}_\mathrm{e}$ is occupied by the weighting matrix $\boldsymbol{C}_\lambda$. The latter allows a prioritization between the different optimization criteria or balancing when the criteria have very different magnitudes. As an illustration, $\boldsymbol{C}_\lambda$ for (2.29)—optimizing the two currents i_d and i_q—might look like

$$\boldsymbol{C}_\lambda = \begin{bmatrix} 0 & 0 & 0 & \lambda_{i_\mathrm{d}} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{i_\mathrm{q}} \end{bmatrix}. \tag{2.51}$$

Moreover, N_c selects the number of steps used to determine the future control trajectory, i.e.,

$$\Delta\boldsymbol{U}_\mathrm{unc}(k) = [\Delta\boldsymbol{u}_\mathrm{unc}^T(k)\ \Delta\boldsymbol{u}_\mathrm{unc}^T(k+1)\ \ldots\ \Delta\boldsymbol{u}_\mathrm{unc}^T(k+N_\mathrm{c}-1)]^T \in \mathbb{R}^{n_u N_\mathrm{c}\times 1}\,. \tag{2.52}$$

Here, based on $\boldsymbol{x}_\mathrm{e}(k)$, the future state variables, i.e.,

$$\boldsymbol{x}_\mathrm{e}^T(k+1|k),\ \boldsymbol{x}_\mathrm{e}^T(k+2|k),\ \ldots,\ \boldsymbol{x}_\mathrm{e}^T(k+N_\mathrm{p}|k)\,, \tag{2.53}$$

are predicted for the length of the prediction horizon N_p. N_p is also the length of the optimization window for the output signals, i.e.,

$$\boldsymbol{Y}(k) = [\boldsymbol{y}^T(k+1|k)\ \boldsymbol{y}^T(k+2|k)\ \ldots\ \boldsymbol{y}^T(k+N_\mathrm{p}|k)]^T \in \mathbb{R}^{n_y N_\mathrm{p} \times 1}\,. \tag{2.54}$$

The aforementioned steps, i.e., calculating the matrices $\boldsymbol{\Gamma}_\mathrm{c}$ and $\boldsymbol{\Upsilon}_\mathrm{c}$ based on $\boldsymbol{A}_\mathrm{c}$, $\boldsymbol{B}_\mathrm{c}$ and $\boldsymbol{C}_\mathrm{c}$, can be performed once offline or, as will be shown in Section 9.2.3, online during operation. By knowing a predicted trend of the model, the most suitable control signal trajectory $\Delta\boldsymbol{U}_\mathrm{unc}$ can be calculated in each control step by minimizing the quadratic cost function, which is described by

$$\begin{aligned} J_\mathrm{c,unc}(k) &= \underbrace{(\boldsymbol{Y}^*(k) - \boldsymbol{Y}(k))^T(\boldsymbol{Y}^*(k) - \boldsymbol{Y}(k))}_{\text{reference tracking}} + \underbrace{\boldsymbol{\Delta U}^T(k)\boldsymbol{R}_\mathrm{v}\Delta\boldsymbol{U}_\mathrm{unc}(k)}_{\text{change of control variable}} \\ &= ||\boldsymbol{Y}^*(k) \underbrace{-\boldsymbol{\Gamma}_\mathrm{c}\boldsymbol{x}_\mathrm{e}(k) - \boldsymbol{\Upsilon}_\mathrm{c}\Delta\boldsymbol{U}_\mathrm{unc}(k)}_{-\boldsymbol{Y}(k)}||_2^2 + \underbrace{||\Delta\boldsymbol{U}_\mathrm{unc}(k)||_{\boldsymbol{R}_\mathrm{v}}^2}_{\text{change of control variable}} \end{aligned} \tag{2.55}$$

where $\boldsymbol{Y}^*$ contains the output reference trajectory and $\boldsymbol{R}_\mathrm{v}$ represents the weighting factor for the change of the input signal. Consequently, the second term enables to scale $\Delta\boldsymbol{U}_\mathrm{unc}$, i.e., adjust the allowed change, by using the overall tuning factor r_v, where $\boldsymbol{R}_\mathrm{v} = r_\mathrm{v}\boldsymbol{I}_{\mathrm{N_c}\times\mathrm{N_c}}$ with $r_\mathrm{v} \in \mathbb{R}_0^-$. The larger r_v is, the smaller the changes in $\Delta\boldsymbol{U}_\mathrm{unc}$ are allowed to be and the slower a control error is corrected. Inserting (2.49) into (2.55) yields

$$\begin{aligned} J_\mathrm{c,unc}(k) &= (\boldsymbol{Y}^*(k) - \boldsymbol{\Gamma}_\mathrm{c}\boldsymbol{x}_\mathrm{e}(k))^T(\boldsymbol{Y}^*(k) - \boldsymbol{\Gamma}_\mathrm{c}\boldsymbol{x}_\mathrm{e}(k)) - 2\Delta\boldsymbol{U}_\mathrm{unc}^T(k)\boldsymbol{\Upsilon}_\mathrm{c}^T(\boldsymbol{Y}^*(k) - \boldsymbol{\Gamma}_\mathrm{c}\boldsymbol{x}_\mathrm{e}(k)) \\ &\quad + \Delta\boldsymbol{U}^T(k)(\boldsymbol{\Upsilon}_\mathrm{c}^T\boldsymbol{\Upsilon}_\mathrm{c} + \boldsymbol{R}_\mathrm{v})\Delta\boldsymbol{U}_\mathrm{unc}(k)\,. \end{aligned} \tag{2.56}$$

The minimum of a convex QP (when positive semidefinite), i.e., the minimum of $J_\mathrm{c,unc}(k)$, is reached when the derivation of the cost function by the change of the control signal $\Delta\boldsymbol{U}$, i.e.,

$$\frac{\partial J_\mathrm{c,unc}(k)}{\partial\Delta\boldsymbol{U}_\mathrm{unc}(k)} = -2\boldsymbol{\Upsilon}_\mathrm{c}^T(\boldsymbol{Y}^*(k) - \boldsymbol{\Gamma}_\mathrm{c}\boldsymbol{x}_\mathrm{e}(k)) + 2(\boldsymbol{\Upsilon}_\mathrm{c}^T\boldsymbol{\Upsilon}_\mathrm{c} + \boldsymbol{R}_\mathrm{v})\Delta\boldsymbol{U}_\mathrm{unc}(k)\,, \tag{2.57}$$

is equal to zero. For convex QP the local minimum is the global minimum. This leads to

$$\Delta\boldsymbol{U}_\mathrm{unc}(k) = (\boldsymbol{\Upsilon}_\mathrm{c}^T\boldsymbol{\Upsilon}_\mathrm{c} + \boldsymbol{R}_\mathrm{v})^{-1}\boldsymbol{\Upsilon}_\mathrm{c}^T(\boldsymbol{Y}^*(k) - \boldsymbol{\Gamma}_\mathrm{c}\boldsymbol{x}_\mathrm{e}(k))\,, \tag{2.58}$$

which is the unconstrained solution of the defined optimization problem. This procedure is called $\mathrm{CCS_{unc}}$ or $\mathrm{CCS_{unc}}$-MPC in the following. Due to the receding horizon policy, only the first element of the input variable $\Delta\boldsymbol{U}_\mathrm{unc}(k)$, i.e., $\Delta\boldsymbol{u}_\mathrm{unc}(k)$, is integrated in each control cycle

T_{cc} to determine the controller output, i.e., system input,

$$\boldsymbol{u}_{\text{unc}}(k) = \boldsymbol{u}(k-1) + \Delta\boldsymbol{u}_{\text{unc}}(k)\,. \tag{2.59}$$

2.5.1.4 Constrained Solution by using Hildreth's Programming

As aforementioned, constraints should not simply restrict the controlled signal after optimization, but rather be taken into account already during the optimization process. Based on the previous section, constraints for optimization are considered hereinafter. Consequently, the optimization of the QP problem requires a non-analytical solution, e.g., numerical, evolutionary. In particular for numerical optimization methods there are numerous possibilities, although mostly gradient, e.g., fast gradient method (FGM)—also called Nesterov method—[Nes83, RMM10, RJM09, PBD13]), interior-point, e.g., [RWR98] or *CVXGEN* [MWB11], alternating direction method of multipliers (ADMM), e.g., *OSQP* [SBGB20], and active-set methods can be found. The latter is especially well suited and popular for drive control problems, since for embedded real-time implementations at least a suboptimal solution can be found in the small and limited control interval [CBLB20]. Moreover, the number of optimization variables and constraints to be considered is rather low to medium, which also argues for active-set [MCK18].
There are numerous solvers based on active-set, e.g., *quadprog* [KJP11] handles a primal problem representation (but mostly not real-time capable), *ODYS* [CBLB20] handles a dual problem representation or *qpOASES* [FKP$^+$14, TGFB19] handles a primal and/or dual problem representation. Another active-set solver using a dual problem representation, shown e.g., in [Hil57], [Wan09, p. 63], [Lue69, p. 299], is based on *Hildreth's quadratic programming procedure*, and is used in the following. In general, dual active-set solvers are preferable to primary ones because they usually find the optimum with significantly fewer iterations [CBLB20]. A good overview of the methods is given, e.g., in [MCK18] and the solvers, e.g., in [KZPF15, Bem].

To determine the constrained solution of the QP problem, i.e., the decision variables $\Delta\boldsymbol{U}_{\text{con}}$, the constraints $\boldsymbol{\gamma}$ and the matrix $\boldsymbol{M}$, which interconnects $\Delta\boldsymbol{U}_{\text{con}}(k)$ to $\boldsymbol{\gamma}$, must be described, see (2.45). Here, two types of input and one type of output constraints are most often defined:

1. Restrict the change of the input signal, i.e., $\Delta u(k)$, to a range of $\Delta u_{\text{min}} \leq \Delta u(k) \leq \Delta u_{\text{max}}$ with the constraint

$$\boldsymbol{M} = \begin{bmatrix} -\boldsymbol{I}_{n_u N_{\text{c}}} \\ \boldsymbol{I}_{n_u N_{\text{c}}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\gamma} = \begin{bmatrix} -\boldsymbol{\Delta U}_{\text{min}} \\ \boldsymbol{\Delta U}_{\text{max}} \end{bmatrix}, \tag{2.60}$$

   where the identity matrix $\boldsymbol{I}_{n_u N_{\text{c}}} \in \mathbb{N}^{n_u N_{\text{c}} \times n_u N_{\text{c}}}$. $\boldsymbol{\Delta U}_{\text{min}}$ and $\boldsymbol{\Delta U}_{\text{max}} \in \mathbb{R}^{n_u N_{\text{c}} \times 1}$ contain the boundary Δu_{min} and Δu_{max}, respectively, for each decision variable.

2. The limitation of the input signal, i.e., $u(k)$, to a range of $u_{\min} \leq u(k) \leq u_{\max}$ is described by

$$\boldsymbol{M} = \begin{bmatrix} -\boldsymbol{M}_{\mathrm{I}} \\ \boldsymbol{M}_{\mathrm{I}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\gamma}(k) = \begin{bmatrix} -\boldsymbol{U}_{\min} + \mathbf{1}_{n_u N_{\mathrm{c}}} u(k-1) \\ \boldsymbol{U}_{\max} - \mathbf{1}_{n_u N_{\mathrm{c}}} u(k-1) \end{bmatrix}, \tag{2.61}$$

where $\boldsymbol{M}_{\mathrm{I}}$, with $N_{\mathrm{c}} \times N_{\mathrm{c}}$ elements, is a block lower triangular matrix filled with identity matrices whose size corresponds to the number of inputs n_u, i.e., $\boldsymbol{I}_{n_u} \in \mathbb{N}^{n_u \times n_u}$, and $\mathbf{1}_{n_u N_{\mathrm{c}}} \in \mathbb{R}^{n_u N_{\mathrm{c}} \times 1}$ is a vector filled with ones.

3. The restriction of the system output $y(k)$ with a constraint of $y_{\min} \leq y(k) \leq y_{\max}$ is described by[12]

$$\boldsymbol{M} = \begin{bmatrix} -\boldsymbol{\Upsilon}_{\mathrm{c}} \\ \boldsymbol{\Upsilon}_{\mathrm{c}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\gamma}(k) = \begin{bmatrix} -\boldsymbol{Y}_{\min} + \boldsymbol{\Gamma}_{\mathrm{c}} \boldsymbol{x}_{\mathrm{e}}(k) \\ \boldsymbol{Y}_{\max} - \boldsymbol{\Gamma}_{\mathrm{c}} \boldsymbol{x}_{\mathrm{e}}(k) \end{bmatrix}. \tag{2.62}$$

A combination of several of these constraints is achieved by stacking the rows of the matrices on the top of each other. For example, for $\Delta u(k)$ and $u(k)$ it looks like

$$\boldsymbol{M} = \begin{bmatrix} -\boldsymbol{I}_{n_u N_{\mathrm{c}}} \\ \boldsymbol{I}_{n_u N_{\mathrm{c}}} \\ -\boldsymbol{M}_{\mathrm{I}} \\ \boldsymbol{M}_{\mathrm{I}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\gamma}(k) = \begin{bmatrix} -\boldsymbol{\Delta U}_{\min} \\ \boldsymbol{\Delta U}_{\max} \\ -\boldsymbol{U}_{\min} + \mathbf{1}_{n_u N_{\mathrm{c}}} u(k-1) \\ \boldsymbol{U}_{\max} - \mathbf{1}_{n_u N_{\mathrm{c}}} u(k-1) \end{bmatrix}. \tag{2.63}$$

After defining the constraints, Lagrange multipliers (also called dual variables), i.e., ρ, are introduced for solving the equality constraints. An equality constraint means that Δu_{con} attempts to assume the value of the constraint γ through the optimization. Based on (2.45), the quadratic optimization problem is reformulated—adding the constraints as an additional term—to the Lagrange representation with

$$\begin{aligned} J_{\mathrm{c,con}}(k) = & \underbrace{\frac{1}{2} \Delta \boldsymbol{U}_{\mathrm{con}}^T(k) \boldsymbol{H}_{\mathrm{c}} \Delta \boldsymbol{U}_{\mathrm{con}}(k) + \Delta \boldsymbol{U}_{\mathrm{con}}^T(k) \boldsymbol{\Theta}_{\mathrm{c}}(k)}_{J_{\mathrm{c,unc}}(k)} \\ & + \boldsymbol{\rho}^T(k) \left(\boldsymbol{M} \Delta \boldsymbol{U}_{\mathrm{con}}(k) - \boldsymbol{\gamma}(k) \right), \end{aligned} \tag{2.64}$$

where the Hessian[13] matrix $\boldsymbol{H}_{\mathrm{c}} = \boldsymbol{\Upsilon}_{\mathrm{c}}^T \boldsymbol{\Upsilon}_{\mathrm{c}} + \boldsymbol{R}_{\mathrm{v}}$ —assumed to be positive definite—and $\boldsymbol{\Theta}_{\mathrm{c}}(k) = \boldsymbol{\Upsilon}_{\mathrm{c}}^T (\boldsymbol{Y}^*(k) - \boldsymbol{\Gamma}_{\mathrm{c}} \boldsymbol{x}_{\mathrm{e}}(k))$, both of which can be previously determined using (2.58).

[12] Usually, soft rather than hard constraints should be used for the output, since the latter can theoretically lead to feasibility problems, see, e.g., [RKCD20].

[13] Strictly speaking $2\boldsymbol{H}_{\mathrm{c}}$ is the Hessian matrix, see (2.57).

To solve the optimization problem, the partial derivative to $\Delta \boldsymbol{U}_{\text{con}}$ and $\boldsymbol{\rho}$ is formed and set to zero.

$$\frac{\partial J_{\text{c,con}}}{\partial \Delta \boldsymbol{U}_{\text{con}}} = \underbrace{\boldsymbol{H}_{\text{c}} \Delta \boldsymbol{U}_{\text{con}}(k) + \boldsymbol{\Theta}_{\text{c}}(k)}_{\frac{\partial J_{\text{c,unc}}}{\partial \Delta \boldsymbol{U}_{\text{unc}}}} + \boldsymbol{M}^T \boldsymbol{\rho}(k) = \boldsymbol{0} \tag{2.65a}$$

$$\frac{\partial J_{\text{c,con}}}{\partial \boldsymbol{\rho}} = \boldsymbol{M} \Delta \boldsymbol{U}_{\text{con}}(k) - \boldsymbol{\gamma}(k) = \boldsymbol{0} \tag{2.65b}$$

In doing so, the optimal $\Delta \boldsymbol{U}_{\text{con}}$ and $\boldsymbol{\rho}$ can be determined analytically for the constrained solution by using

$$\boldsymbol{\rho}_{\text{con}}(k) = -\left(\boldsymbol{M}_{\text{c}} \boldsymbol{H}_{\text{c}}^{-1} \boldsymbol{M}^T\right)^{-1} \left(\boldsymbol{\gamma}(k) + \boldsymbol{M} \underbrace{\boldsymbol{H}_{\text{c}}^{-1} \boldsymbol{\Theta}_{\text{c}}(k)}_{-\Delta \boldsymbol{U}_{\text{unc}}(k)}\right) \tag{2.66a}$$

$$\Delta \boldsymbol{U}_{\text{con}}(k) = \underbrace{-\boldsymbol{H}_{\text{c}}^{-1} \boldsymbol{\Theta}_{\text{c}}(k)}_{\Delta \boldsymbol{U}_{\text{unc}}(k)} - \boldsymbol{H}_{\text{c}}^{-1} \boldsymbol{M}^T \boldsymbol{\rho}(k)\,, \tag{2.66b}$$

where the first term represents the unconstrained solution. However, the described problem can only be solved if equality constraints can be imposed.
In the case of equality constraints, their number must be less than or equal to the number of decision variables ρ [Wan09, p. 57]. However, if the number of constraints is equal to the number of optimization variables—which is another name for decision variables—there is only one solution, the one that satisfies all constraints. Consequently, the original target of the cost function, i.e., reference tracking, is ignored. If the number of equality constraints is even greater than the number of optimization variables, the problem is infeasible. It becomes obvious that on the one hand the constraints often cannot be described as equality constraints, i.e., using „$=$“, and, on the other hand, the number of constraints can be identical or even greater than the number of optimization variables, see Section 9.2.2.
For this reason the restrictions are no longer described as equality but rather as inequality constraints, i.e., using „$\leq$“ and „$\geq$“. In the minimization with inequality constraints, the number of constraints could be larger than the number of decision variables [Wan09, p. 58].

In doing so, the Kuhn-Tucker conditions, also called Karush–Kuhn–Tucker (KKT) conditions, are introduced in the following. They define, which of the inequality constraints, i.e., $M_i \Delta u_{\text{con}} \leq \gamma_i$—which are summarized in several rows of $\boldsymbol{M}$ and $\boldsymbol{\gamma}$ where i is the number of rows—are active and which are inactive [Wan09, p. 58]. Active means that the inequality constraint is written as $M_i \Delta u_{\text{con}} = \gamma_i$ with a non-negative Lagrange multiplier, i.e., $\rho_i > 0$. Inactive means that the inequality constraint is written as $M_i \Delta u_{\text{con}} < \gamma_i$ with a zero Lagrange multiplier, i.e., $\rho_i = 0$. In doing so, based on $\boldsymbol{M} \Delta \boldsymbol{U}_{\text{con}} \leq \boldsymbol{\gamma}$, only the active constraints are summarized with $\boldsymbol{M}_{\text{act}} \Delta \boldsymbol{U}_{\text{con}} = \boldsymbol{\gamma}_{\text{act}}$.

An active-set method decides at each step of the optimization which of the constraints are active, i.e., must be fulfilled, and which are inactive, i.e., can be dropped. Consequently, at each step an equality constraint problem is solved [Wan09, p. 60]. This allows the use of the previously introduced Lagrange multipliers to solve the quadratic optimization problem. Using the active-set method, the optimization problem defined in (2.66) can be redefined by

$$\boldsymbol{\rho}_{\text{act}}(k) = -\left(\boldsymbol{M}_{\text{act}}\boldsymbol{H}_{\text{c}}^{-1}\boldsymbol{M}_{\text{act}}^{T}\right)^{-1}\left(\boldsymbol{\gamma}_{\text{act}}(k) + \boldsymbol{M}_{\text{act}}\underbrace{\boldsymbol{H}_{\text{c}}^{-1}\boldsymbol{\Theta}_{\text{c}}(k)}_{-\Delta\boldsymbol{U}_{\text{unc}}(k)}\right) \tag{2.67a}$$

$$\Delta\boldsymbol{U}_{\text{con}}(k) = \underbrace{-\boldsymbol{H}_{\text{c}}^{-1}\boldsymbol{\Theta}_{\text{c}}(k)}_{\Delta\boldsymbol{U}_{\text{unc}}(k)} - \boldsymbol{H}_{\text{c}}^{-1}\boldsymbol{M}_{\text{act}}^{T}\boldsymbol{\rho}_{\text{act}}(k)\,. \tag{2.67b}$$

However, since this is a primal representation, i.e., the active constraints (2.67a) need to be iteratively identified along with the optimal decision variables (2.67b), the occurrence of numerous constraints can lead to a huge computational burden. Therefore, real-time optimization can be quite time-consuming as an iterative solution process is required and inactive constraints can become active constraints during optimization.

To overcome this issue, a dual problem can first be solved using Lagrange multipliers to identify the inactive constraints with less computational effort. In doing so, assuming feasibility, i.e., there is an ρ_i such that $M_i\rho_i < \gamma_i$, the primal problem is described by

$$\underset{\rho\geq 0}{\text{maximize}} \quad \underset{\Delta\boldsymbol{U}_{\text{con}}}{\text{minimize}} \; J_{\text{c,con}} \;\; \text{(see (2.64))}\,, \tag{2.68}$$

where the minimization regarding $\Delta\boldsymbol{U}_{\text{con}}$, see (2.66b), is unconstrained. Inserting (2.66b) in (2.68), the dual problem can be rewritten by [Wan09, p. 62]

$$\underset{\rho\geq 0}{\text{minimize}}\ \left(\frac{1}{2}\boldsymbol{\rho}^{T}(k)\boldsymbol{E}\boldsymbol{\rho}(k) + \boldsymbol{\rho}^{T}(k)\boldsymbol{g}(k) + \frac{1}{2}\boldsymbol{\gamma}^{T}(k)\boldsymbol{H}_{\text{c}}^{-1}\boldsymbol{\gamma}(k)\right) \tag{2.69}$$

with

$$\boldsymbol{E} = \boldsymbol{M}\boldsymbol{H}_{\text{c}}^{-1}\boldsymbol{M}^{T} \tag{2.70a}$$

$$\boldsymbol{g}(k) = \boldsymbol{\gamma}(k) + \boldsymbol{M}\boldsymbol{H}_{\text{c}}^{-1}\boldsymbol{\Theta}_{\text{c}}(k)\,. \tag{2.70b}$$

Although it is still a QP problem on the dual variables, the constraints are much simplified ($\boldsymbol{\rho} \geq \boldsymbol{0}$) so a simple iterative procedure can be used to obtain the optimal solution of the Lagrange multipliers. Consequently, first—and quite fast—the active constraints $\boldsymbol{\rho}_{\text{act}}$ with $\boldsymbol{M}_{\text{act}}$ can be obtained by solving the dual problem (2.69)-(2.70b) to subsequently facilitate the solution of the primal problem.

Hildreth's quadratic programming procedure A simple method to solve the mentioned dual problem is Hildreth's quadratic programming algorithm [Hil57], [Lue69, p. 299 ff.], which is based on an element-by-element search, i.e., is an iterative solution. Thus, it does not require any matrix inversion. This has the important advantage that if the active constraints are linearly independent and their number is less than or equal to the number of decision variables, then the dual variables converge [Wan09, p. 63 ff.]. Moreover, the algorithm provides a near-optimal solution even in the presence of conflicting constraints [Wan09, p. 64] without being numerically unstable [Wan09, p. 82], which makes it advantageous for real-time applications and is the reason for the choice.

In doing so, the individual Lagrange multipliers are optimized element by element using[14]

$$\rho_i^{k+1} = \max(0, w_i^{k+1}) \quad \text{with} \tag{2.71}$$

$$w_i^{k+1} = -\frac{1}{e_{ii}} \left(g_i + \sum_{j=1}^{i-1} e_{ij} \rho_j^{k+1} + \sum_{j=i+1}^{m} e_{ij} \rho_j^k \right) , \tag{2.72}$$

where e_{ij} is the element i, j in the matrix $\boldsymbol{E}$ of (2.70a) and g_i the element i in the vector $\boldsymbol{g}$ of (2.70b) [Hil57]. Here, m equals the length of $\boldsymbol{\gamma}$.
If in the solution of the dual problem the Lagrange multipliers converge, i.e., $\boldsymbol{\rho}^{\circledast}$ results with either zero or positive values for the Lagrange multipliers, only the rows of $\boldsymbol{\rho}^{\circledast}$ with nonzero values can be chosen to determine $\boldsymbol{\rho}_{\text{act}}$ and $\boldsymbol{M}_{\text{act}}$. Consequently, the constrained solution of the optimization problem, i.e., the primal problem in (2.67b), can be computed directly and much more easily. Similar to (2.59), only the first element of the constraint decision variable $\Delta \boldsymbol{U}_{\text{con}}(k)$, i.e., $\Delta \boldsymbol{u}_{\text{con}}(k)$, is applied afterwards. This procedure is called CCS_{con} or CCS_{con}-MPC in the following.

Since it is an iterative algorithm, many active constraints may prevent the calculation from being completed in the required time, i.e., the hard real time would be violated. Consequently, a termination criterion is inserted, which terminates the optimization after 80 %[15] of the available processing time to ensure sufficient computing capacity for remaining tasks. This is especially important for the real-time implementation of the algorithm[16], see, e.g., Chapter 9.

[14] $\max$ returns for each i-th element the largest element taken from the first or second argument. Thus it results for each specific constraint zero if the constraint is inactive or a positive value if the constraint is active.

[15] The value was determined heuristically based on empirical results and depends on the workload of the processor.

[16] For manageable problems, as shown in Section 9.2, experimental evaluations have shown that the processing time was not violated at any time

2.5.2 Finite Control Set-MPC

FCS-MPC—also named DMPC—is a direct control strategy. The underlying optimization problem, which is most often formulated as a reference tracking problem, is solved by enumerating the finite number of possible switch positions of the power electronic converter [CKK+08], i.e., for a two-level VSI eight discrete voltage SVs, see Fig. 2.5. This implies that the converter switches are directly manipulated, thus the control and modulation problem are addressed in one computational stage [Gey17, p. 18]. When using long-horizon FCS-MPC, two features are prominent. First, an improved steady-state system performance can be achieved, as indicated by a reduced THD of the variables of concern for a given average switching frequency f_{sw} [GKK14, GQ15, LKG+17]. Second, the stability of the system can be improved [KLGK20]. A good overview of multistep FCS-MPC is given, e.g., in [ARB+19]. For a long horizon the calculations expand like a tree, see Fig. 2.23. Respecting the number of available switching states, see Table 2.1, an exponential increase with $\mathrm{SV}^{N_{\mathrm{p}}}$ results.

With conventional FCS-MPC, switching can only take place at discrete time steps, i.e., at the end of the control interval in which the optimization was performed, see Fig. 2.23. Here, the algorithm can change the switch positions, but does not have to. In doing so, direct control strategies such as FCS-MPC provide fast transient response, but also operate the converter with a time-varying, i.e., variable, switching frequency—resulting in an average f_{sw}—and, consequently, a non-discrete harmonic spectrum [Gey11]. This suggests that for a three-phase system significant harmonic components exist not only at non-triplen, odd integer multiples of the fundamental frequency, but also at even and inter-harmonics. For grid-side connected inverters, this makes conventional FCS-MPC unsuitable since the relevant grid codes—such as IEEE 519 [IEE14] and IEC 61000-2-4 standard—impose tight limits on these harmonics. Hence, grid-

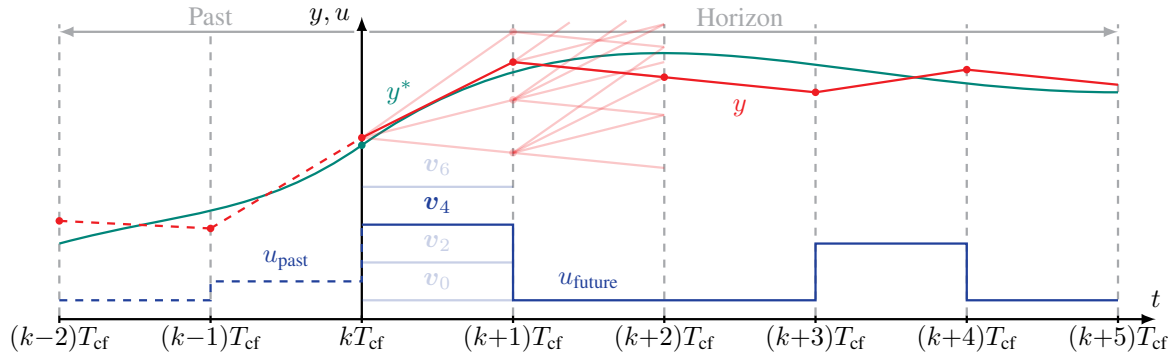


Figure 2.23: FCS-MPC with receding horizon policy for a five-step prediction horizon, i.e., $N_{\mathrm{p}} = 5$, exemplary shown at kT_{cf} for $\boldsymbol{v}_0$, $\boldsymbol{v}_2$, $\boldsymbol{v}_4$, $\boldsymbol{v}_6$ with not chosen options in light color. u are the discrete control signals, y are the resulting controlled signals and y^* is the reference signal. Even though the signals are mostly vectors, for simplicity only the trend of a scalar is shown.

side connected applications requiere modified DMPC approaches, as shown, e.g., in [NKG19]. However, for inverters connected on the motor side, a spread spectrum, resulting in reduced amplitudes in certain harmonics, is acceptable or may even be advantageous [Hol94].

In the field of DMPC—see also [KLGK20]—two basic procedures are distinguished:

1. Conventional FCS-MPC, which allows only one switching state to be used for the entire control interval. This basic principle, which is used in almost all DMPC publications, is presented in this chapter.
2. DMPC strategies with implicit modulator that introduce a variable switching point (VSP), i.e., a time (switching) instant, within the control interval at which a new switching state is applied to the converter as shown, e.g., in [KSK+14, SKT+13, TZW+15, ZXLZ14, WKDK19]. This variant is used and enhanced in Chapter 6.

Both are described by Fig. 2.24, where the latter may use an additional counter as shown by the dashed lines.
When DMPC for PMSM-based drives is employed to control the stator current, the common practice is to compute the evolution of the latter based on the (time-varying) absolute inductances L_d and L_q [WDK17b]. Specifically, by applying forward Euler discretization—see Section 2.1.3—to the rearranged (2.27), the predicted current $\boldsymbol{i}_{dq}(k+1)$ is given by

$$\boldsymbol{i}_{dq}(k+1) = T_{\mathrm{cf}}\boldsymbol{L}_{dq}^{-1}\left(\boldsymbol{v}_{dq}(k) - R_{\mathrm{ph}}\boldsymbol{i}_{dq}(k) - \omega_{\mathrm{el}}(k)\left(\boldsymbol{P}\boldsymbol{L}_{dq}\boldsymbol{i}_{dq}(k) + \begin{bmatrix} 0 \\ \psi_{\mathrm{pm}} \end{bmatrix}\right)\right) + \boldsymbol{i}_{dq}(k)\,, \quad (2.73)$$

where T_{cf} is the control interval. In the case of DMPC, its common practice to use forward Euler discretization [Gey17, p. 155], [KG20]. Thus for all DMPC considerations in this thesis forward Euler is used. It is also common to use constant absolute inductances for the prediction, although (2.73) may lead to less accurate predictions and, consequently, to potential performance deterioration when magnetic nonlinearities become dominant. Therefore, this issue is addressed in Chapter 7.

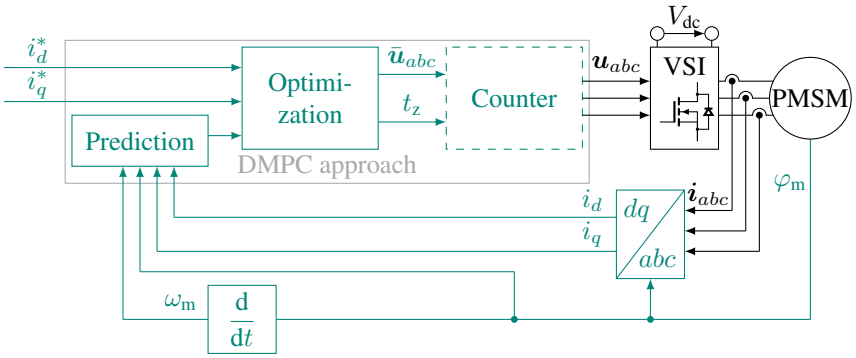


Figure 2.24: DMPC with current reference tracking for a two-level VSI with a PMSM.

2.5.2.1 Optimization Problem

Using the predicted current trends, a cost function can be defined with the reference signal(s) to obtain the most appropriate discrete switching signals for the subsequent control step. A common cost function can be described by [Gey17, p. 197]

$$\begin{aligned} J_{\mathrm{f}}(k) &= \sum_{\ell=k}^{k+N_{\mathrm{p}}-1} \underbrace{||\boldsymbol{y}^*(\ell+1)-\boldsymbol{y}(\ell+1)||^2_{\boldsymbol{Q}}}_{\text{reference tracking}} + \underbrace{\lambda_{\mathrm{u}}||\Delta\boldsymbol{u}(\ell)||^2_2}_{\text{switching penalization}} \\ &= \underbrace{\sum_{\ell=k}^{k+N_{\mathrm{p}}-1} ||\boldsymbol{y}^*(\ell+1)-\boldsymbol{y}(\ell+1)||^2_{\boldsymbol{Q}}}_{J_{\mathrm{f},1}(k)} + \underbrace{\sum_{\ell=k}^{k+N_{\mathrm{p}}-1} \lambda_{\mathrm{u}}||\Delta\boldsymbol{u}(\ell)||^2_2}_{J_{\mathrm{f},2}(k)} \end{aligned} \tag{2.74}$$

where, similar to (2.55), the first term is the reference tracking term and the second term is the penalty for changing the control variable, i.e., a switching transition. More precisely, the reference tracking term with the reference $\boldsymbol{y}^*$ and the optimized output $\boldsymbol{y}$ can be weighted by the matrix $\boldsymbol{Q} \in \mathbb{R}^{n_y \times n_y}$, where n_{y} is the number of output variables. This can also be described by the output tracking term $\boldsymbol{\xi}$ with $||\boldsymbol{y}^*(\ell+1)-\boldsymbol{y}(\ell+1)||^2_{\boldsymbol{Q}} = ||\boldsymbol{\xi}(\ell+1)||^2_{\boldsymbol{Q}} = (\boldsymbol{\xi}(\ell+1))^T\boldsymbol{Q}\boldsymbol{\xi}(\ell+1)$ and leads after summation to $J_{\mathrm{f},1} = [\boldsymbol{\xi}^T(k+1)...\boldsymbol{\xi}^T(k+N_{\mathrm{p}})]\tilde{\boldsymbol{Q}}[\boldsymbol{\xi}^T(k+1)...\boldsymbol{\xi}^T(k+N_{\mathrm{p}})]^T$ with the block diagonal matrix $\tilde{\boldsymbol{Q}} = \mathrm{diag}(\boldsymbol{Q}, \cdots, \boldsymbol{Q})$. The penalty of the switching is described by $\lambda_{\mathrm{u}}||\Delta\boldsymbol{u}(\ell)||^2_2 = \lambda_u(\boldsymbol{u}(\ell)-\boldsymbol{u}(\ell-1))^T(\boldsymbol{u}(\ell)-\boldsymbol{u}(\ell-1))$.

However, many different forms of the cost function can be found in the literature, see, e.g., [KLGK20]. Even if these forms of representation apply equally to CCS-MPC and DMPC, especially in the latter case many different variants can be found, mainly for computational reasons. The main differences are first the norm, second the presence of weighting factors and third the number of terms to be optimized and constraints to be respected. With regard to the former, a squared ℓ_2-norm is usually used for all terms of the cost function, although an ℓ_1-norm is also possible. As described in [KGK18], the ℓ_2-norm avoids any closed-loop stability issues and provides a performance improvement (at least for the reference tracking term), although this increases the computational effort. With regard to the second, by changing (online) the weighting factors, the targets to be optimized, i.e., $\boldsymbol{y}^*$, have a variable influence, i.e., more or less, on the resulting control signal $\Delta\boldsymbol{u}$. With regard to the latter, a penalization of the switching action is always recommended to avoid unnecessary current distortions [KG20] and to adjust f_{sw}. Moreover, current or voltage restrictions are quite often included as additional constraints.

Based on the above, the cost function used in this thesis is composed of a ℓ_2-norm for reference tracking—due to stability reasons—, a ℓ_2-norm for the current constraints—since the current vector length is calculated based on a quadratic form—and ℓ_1-norm for the switching

penalization—in order to reduce the computational burden. The proposed cost function is described by

$$J_{\mathrm{f}}(k) = \sum_{\ell=k}^{k+N_{\mathrm{p}}-1} \Big(||\boldsymbol{y}^*(\ell+1) - \boldsymbol{y}(\ell+1)||_2^2 + \hat{f}(\boldsymbol{i}_{dq}(\ell+1)) + \lambda_{\mathrm{u}} ||\boldsymbol{\Delta}[\boldsymbol{u}_{abc}^T(\ell-1)\ \boldsymbol{u}_{abc}^T(\ell)]^T||_1 \Big) \quad (2.75)$$

where

$$\hat{f}(\boldsymbol{i}_{dq}(\ell+1)) = \begin{cases} i_{\max} & \text{if } ||\boldsymbol{i}_{dq}(\ell+1)||_2 > i_{\max} \\ 0 & \text{if } ||\boldsymbol{i}_{dq}(\ell+1)||_2 \leq i_{\max} \end{cases} \quad (2.76)$$

with $\boldsymbol{y}^* = \boldsymbol{i}_{dq}^* \in \mathbb{R}^2$ being the reference vector and $\boldsymbol{y} = \boldsymbol{i}_{dq} \in \mathbb{R}^2$ the output vector. The sequence of manipulated switch positions over a finite horizon of $N_{\mathrm{p}} \in \mathbb{N}^+$ time steps is defined as

$$\boldsymbol{U}(k) = [\boldsymbol{u}_{abc}^T(k)\ \boldsymbol{u}_{abc}^T(k+1)\ \ldots\ \boldsymbol{u}_{abc}^T(k+N_{\mathrm{p}}-1)]^T \in \mathbb{U} \quad (2.77)$$

with $\boldsymbol{u}_{abc}$ and $\mathbb{U} = \{-1, 1\}^{n_h N_{\mathrm{p}}}$. Moreover, $\boldsymbol{\Delta}[\boldsymbol{u}_{abc}^T(\ell-1)\ \boldsymbol{u}_{abc}^T(\ell)]^T$ with $\boldsymbol{\Delta} = \begin{bmatrix} -\boldsymbol{I}_{n_h} & \boldsymbol{I}_{n_h} \end{bmatrix}$, where $\boldsymbol{I}_{n_h} \in \mathbb{N}^{n_h \times n_h}$ and n_h being the number of phase legs (here $n_h = 3$, see Section 2.2.1.1), denotes the penalization of the control action, and, consequently, of the switching frequency, which is weighted by $\lambda_{\mathrm{u}} > 0$. In this way, the average switching frequency can be reduced at the expense of a higher current ripple. In this work, $\lambda_{\mathrm{u}} > 0$ is adjusted for all conventional FCS-MPC approaches so that a comparison—in terms of current THD and ripple—between the methods at almost equal average f_{sw} is possible. In contrast to the general form, i.e., (2.74), the proposed form, i.e., (2.75), for simplicity's sake avoids an unequal weighting of the reference tracking terms, i.e., avoids $\boldsymbol{Q}$. This is meaningful since the d- and q-axis are equivalent, which also considerably facilitates the adjustment of the remaining weighting factor. Finally, (2.76) represents a hard constraint on the stator current, implemented as a protection mechanism, with $i_{\max}$ being the maximum current in per-unit (pu).[17]

With the cost function (2.75), the DMPC problem is stated as

$$\begin{aligned} &\underset{\boldsymbol{u}_{abc} \in \mathbb{U}}{\text{minimize}} \ \ J_{\mathrm{f}} \text{ (see (2.75))} \\ &\text{subject to } \ (2.73),\ (2.76)\,. \end{aligned} \quad (2.78)$$

Solving (2.78) yields the optimal switch position(s) $\boldsymbol{u}_{abc,\mathrm{con}}(k)$ which is (are) to be applied at the end of the control interval. Note that according to the receding horizon policy, only the first element of the switching sequence $\boldsymbol{U}_{\mathrm{con}}$ is applied to the inverter. In doing so, feedback is provided and a degree of robustness to system uncertainties is achieved [RM09].

[17] Theoretically, a soft constraint is to be preferred since it can avoid feasibility problems when solving (2.75). The per-unit system is used as explained in [Gey17, p. 31 ff.].

Algorithm 1 Inductance-based DMPC

function $\boldsymbol{u}_{\text{ABC,CON}} = L\text{-DMPC}(\boldsymbol{i}_{dq}^*, \boldsymbol{i}_{dq}, \varphi, N_\text{p}, \omega_\text{el}, V_\text{dc})$
  $\boldsymbol{i}_{dq}(k-1) \leftarrow$ read dq-transformation
  $\boldsymbol{i}_{dq}(k) \leftarrow$ predict using (2.73) ▷ delay time compensation
  for $\ell = 1, \ldots, N_\text{p}$ **do**
    for $j = 1, \ldots, 8$ **do**
      $\boldsymbol{i}_{dq,j}(k+\ell) \leftarrow$ predict using (2.73)
      $||\boldsymbol{i}_{dq,j}(k+\ell)|| \leftarrow \boldsymbol{i}_{dq,j}(k+\ell)$ using (2.76) ▷ current constraint
    end for
    $\boldsymbol{u}_{abc,\text{con}}(k) \leftarrow$ solve (2.78)
  end for
end function

This leads to Algorithm 1. Here, the delay time of the current is compensated for only once after sampling. However, for the rotor position, a compensation is required after sampling and each horizon step ℓ, i.e., $1, \ldots, N_\text{p}$. Thus, in each step $\boldsymbol{v}_{dq}$ in (2.73) is updated using (2.12) as the rotor shaft continues to rotate and thus has another position. Moreover, exhaustive search is used, i.e., all possible switching possibilities are considered to find the constrained solution. Thus, an exponential growth of the computational effort with the horizon, i.e., SV^{N_p}, occurs. To reduce this increase, e.g., the following two sections or alternatively Chapter 5 are required.

2.5.2.2 Integer Quadratic Programming Formulation

Contrary to CCS-MPC, see Section 2.5.1, FCS-MPC is a (mixed) integer problem. Therefore, similar to CCS-MPC, a quadratic optimization problem can be defined, but is solved by means of (mixed) integer quadratic programming ((M)IQP), since the solution space contains (mostly) integer values. In doing so, similar to (2.49), the predicted output is defined by

$$\boldsymbol{Y}(k) = \boldsymbol{\Gamma}_\text{f}\boldsymbol{x}(k) + \boldsymbol{\Upsilon}_\text{f}\boldsymbol{U}(k) \tag{2.79}$$

with

$$\boldsymbol{\Gamma}_\text{f} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{N_\text{p}} \end{bmatrix}, \quad \boldsymbol{\Upsilon}_\text{f} = \begin{bmatrix} \boldsymbol{CB} & \boldsymbol{0}_{n_y \times n_h} & \boldsymbol{0}_{n_y \times n_h} & \cdots & \boldsymbol{0}_{n_y \times n_h} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \boldsymbol{0}_{n_y \times n_h} & \cdots & \boldsymbol{0}_{n_y \times n_h} \\ \boldsymbol{CA}^2\boldsymbol{B} & \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & \boldsymbol{0}_{n_y \times n_h} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{CA}^{N_\text{p}-1}\boldsymbol{B} & \boldsymbol{CA}^{N_\text{p}-2}\boldsymbol{B} & \boldsymbol{CA}^{N_\text{p}-3}\boldsymbol{B} & \cdots & \boldsymbol{CB} \end{bmatrix} \tag{2.80}$$

using prediction horizon N_p. Here, the state space description in (2.29) is discretized by forward Euler—see Section 2.1.3—, where $\boldsymbol{B}_\text{c} = \frac{V_\text{dc}}{2} \begin{bmatrix} 1/L_d & 0 & 0 \\ 0 & 1/L_q & 0 \end{bmatrix}^T \boldsymbol{K}(\varphi)$ is redefined since in

$\boldsymbol{U}$ discrete control signals, i.e., the discrete switch positions, are the decision variables rather than the change of a continuous control signal as shown for CCS-MPC. Moreover, since the switch positions are the decision variables, $n_u = n_h$. In contrast to the proposed CCS-MPC approach, see Section 2.5.1, $\boldsymbol{\Upsilon}_\mathrm{f}$ provides only a prediction horizon, i.e., no control horizon N_c, and a square format, where n_y indicates the number of output variables in the vector $\boldsymbol{y}$.
Based on (2.74), using the output tracking error over the whole horizon, i.e., $\boldsymbol{\Xi}$, the first part of the cost function can be described by $J_{\mathrm{f},1} = [\boldsymbol{\xi}^T(k+1)...\boldsymbol{\xi}^T(k+N_p)]\tilde{\boldsymbol{Q}}[\boldsymbol{\xi}^T(k+1)...\boldsymbol{\xi}^T(k+N_p)]^T = \boldsymbol{\Xi}^T(k)\tilde{\boldsymbol{Q}}\boldsymbol{\Xi}(k) = ||\boldsymbol{\Xi}^T(k)||^2_{\tilde{\boldsymbol{Q}}}$. Consequently, with the output reference trajectory $\boldsymbol{Y}^*(k)$, $\boldsymbol{\Xi}(k) = \boldsymbol{Y}^*(k) - \boldsymbol{Y}(k)$ results and leads to

$$J_\mathrm{f}(k) = \underbrace{||\boldsymbol{Y}^*(k) \underbrace{-\boldsymbol{\Gamma}_\mathrm{f}\boldsymbol{x}(k) - \boldsymbol{\Upsilon}_\mathrm{f}\boldsymbol{U}(k)}_{-\boldsymbol{Y}(k)}||^2_{\tilde{\boldsymbol{Q}}}}_{J_{\mathrm{f},1}} + \underbrace{\lambda_u||\boldsymbol{S}\boldsymbol{U}(k) - \boldsymbol{E}\boldsymbol{u}(k-1)||^2_2}_{J_{\mathrm{f},2}} \tag{2.81}$$

where $J_{\mathrm{f},2} = \lambda_\mathrm{u}(\boldsymbol{S}\boldsymbol{U}(k) - \boldsymbol{E}\boldsymbol{u}(k-1))^T(\boldsymbol{S}\boldsymbol{U}(k) - \boldsymbol{E}\boldsymbol{u}(k-1)) = \lambda_\mathrm{u}||\boldsymbol{S}\boldsymbol{U}(k) - \boldsymbol{E}\boldsymbol{u}(k-1)||^2_2$ with

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{I}_{n_h} \\ \boldsymbol{0}_{n_h\times n_h} \\ \boldsymbol{0}_{n_h\times n_h} \\ \vdots \\ \boldsymbol{0}_{n_h\times n_h} \end{bmatrix} \quad \text{and} \quad \boldsymbol{S} = \begin{bmatrix} \boldsymbol{I}_{n_h} & \boldsymbol{0}_{n_h\times n_h} & \boldsymbol{0}_{n_h\times n_h} & \cdots & \boldsymbol{0}_{n_h\times n_h} \\ -\boldsymbol{I}_{n_h} & \boldsymbol{I}_{n_h} & \boldsymbol{0}_{n_h\times n_h} & \cdots & \boldsymbol{0}_{n_h\times n_h} \\ \boldsymbol{0}_{n_h\times n_h} & -\boldsymbol{I}_{n_h} & \boldsymbol{I}_{n_h} & \cdots & \boldsymbol{0}_{n_h\times n_h} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{0}_{n_h\times n_h} & \boldsymbol{0}_{n_h\times n_h} & \boldsymbol{0}_{n_h\times n_h} & \cdots & \boldsymbol{I}_{n_h} \end{bmatrix}. \tag{2.82}$$

Afterwards several rearrangements are required—see, e.g., [Gey17, p. 215]—to separate the time variant terms in (2.81). In doing so, the cost function can be simplified [Gey17, p. 201-202], i.e.,

$$J_\mathrm{f}(k) = (\boldsymbol{U}(k) + \underbrace{\boldsymbol{H}_\mathrm{f}^{-1}\boldsymbol{\Theta}_\mathrm{f}(k)}_{-\boldsymbol{U}_\mathrm{unc}(k)})^T\boldsymbol{H}_\mathrm{f}(\boldsymbol{U}(k) + \underbrace{\boldsymbol{H}_\mathrm{f}^{-1}\boldsymbol{\Theta}_\mathrm{f}(k)}_{-\boldsymbol{U}_\mathrm{unc}(k)}) \tag{2.83}$$

with

$$\boldsymbol{H}_\mathrm{f} = \boldsymbol{\Upsilon}_\mathrm{f}^T\tilde{\boldsymbol{Q}}\boldsymbol{\Upsilon}_\mathrm{f} + \lambda_u\boldsymbol{S}^T\boldsymbol{S} \tag{2.84a}$$

$$\boldsymbol{\Theta}_\mathrm{f}^T(k) = -(\boldsymbol{Y}^*(k) - \boldsymbol{\Gamma}_\mathrm{f}\boldsymbol{x}(k))^T\tilde{\boldsymbol{Q}}\boldsymbol{\Upsilon}_\mathrm{f} - \lambda_u(\boldsymbol{E}\boldsymbol{u}(k-1))^T\boldsymbol{S}\,, \tag{2.84b}$$

where $\boldsymbol{H}_\mathrm{f}$ is the Hessian[18] matrix as a function of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\tilde{\boldsymbol{Q}}$, λ_u and $\boldsymbol{S}$. Assuming that the system parameters are time-invariant, $\boldsymbol{H}_\mathrm{f}$ is also time-invariant[19]. Similar to Section 2.5.1.4, $\boldsymbol{H}_\mathrm{f}$ is symmetric and positive definite for $\lambda_u > 0$ [Gey17, p. 201]. $\boldsymbol{\Theta}_\mathrm{f}$ is a time-varying vector of the system state vector at time k, the output reference trajectory $\boldsymbol{Y}^*$ and the previously selected

[18] Strictly speaking $2\boldsymbol{H}_\mathrm{f}$ is the Hessian Matrix, see [Gey17, p. 201].
[19] As will be shown in the following and especially in Chapter 5, this can only be partially assumed for small electrical drives due to the small electrical and mechanical time constants.

switch position $\boldsymbol{u}(k-1)$. Afterwards, it is possible to calculate the solution of the optimization problem, i.e., the optimal discrete switching signals (integer solution) $\boldsymbol{U}_{\text{con}}(k)$, by using

$$\begin{aligned} &\underset{\boldsymbol{U}(k)\in\mathbb{U}}{\text{minimize}} \;\; J_{\text{f}} \text{ (see (2.83))} \\ &\text{subject to} \;\; (2.76), \forall\, \ell = k, \ldots, k+N_{\text{p}}-1\,. \end{aligned} \tag{2.85}$$

As shown in (2.83), the unique unconstrained solution $\boldsymbol{U}_{\text{unc}}$—since $\boldsymbol{H}_{\text{f}}$ is positive definite—can be calculated directly by neglecting the side conditions in (2.85), similar to the first term of (2.66b). Since this solution does not consider any constraints, it cannot be output directly, but can be used to simplify the constrained solution. Based on this, as explained in [Gey17, p. 202], the optimization problem can be redefined as integer quadratic program [GQ14], i.e.,

$$J_{\text{f}}(k) = (\boldsymbol{V}\boldsymbol{U}(k) - \bar{\boldsymbol{U}}_{\text{unc}}(k))^T(\boldsymbol{V}\boldsymbol{U}(k) - \bar{\boldsymbol{U}}_{\text{unc}}(k)) = ||\boldsymbol{V}\boldsymbol{U}(k) - \bar{\boldsymbol{U}}_{\text{unc}}(k)||_2^2\,, \tag{2.86}$$

with the lattice generator matrix $\boldsymbol{V}$ where

$$\boldsymbol{V}^{-1}\boldsymbol{V}^T = \boldsymbol{H}_{\text{f}}^{-1} \tag{2.87a}$$

$$\bar{\boldsymbol{U}}_{\text{unc}}(k) = \boldsymbol{V}\boldsymbol{U}_{\text{unc}}(k) = -\boldsymbol{V}\boldsymbol{H}_{\text{f}}^{-1}\boldsymbol{\Theta}_{\text{f}}(k)\,. \tag{2.87b}$$

Thus, a problem similar to (2.85) can be formulated and solved as described in the following.

2.5.2.3 Branch and Bound by using Sphere Decoding

The definition of the optimization problem as an integer quadratic program aims at the use of an IQP solver to find the constrained solution in a computationally efficient manner, i.e., to find it faster. One of the most promising IQP solver, using a branch and bound (BnB) method, is the sphere decoding algorithm (SDA) which narrows down the search space of FCS-MPC and has become very popular in the recent years, see, e.g., [GQ14, GQ15, KGMK16, DMKG17, LKG+17, ARB+19, BAA+20]. The idea is not to examine all discrete switch positions in $\boldsymbol{U}(k)$, but to search only within a radius $\rho_{\text{sph}}(k)$ (>0) centered at $\bar{\boldsymbol{U}}_{\text{unc}}(k)$. This leads to

$$\rho_{\text{sph}}(k) \geq ||\boldsymbol{V}\boldsymbol{U}(k)) - \bar{\boldsymbol{U}}_{\text{unc}}(k)||_2\,. \tag{2.88}$$

Since $\boldsymbol{V}$ is a triangular matrix, it allows the SDA to find the minimum sphere relatively easily. More precisely, $\boldsymbol{V}$ is a lower triangular matrix in our case, so (2.88) can be rewritten to

$$\rho_{\text{sph}}^2(k) \geq (v_{1,1}u_1(k) - \bar{u}_{\text{unc},1}(k))^2 + (v_{2,1}u_1(k) + v_{2,2}u_2(k) - \bar{u}_{\text{unc},2}(k))^2 + \cdots\,. \tag{2.89}$$

It holds, $\bar{u}_{\text{unc,i}}(k)$ is the i-th element of $\bar{\boldsymbol{U}}_{\text{unc}}(k)$, $u_i(k)$ is the i-th element of $\boldsymbol{U}(k)$ and $v_{i,j}(k)$ is the $(i,\,j)$-th element of $\boldsymbol{V}$. The solution is a sequential solution, similar to Gaussian elimination,

in the sense that each step is only a one-dimensional problem, see, e.g., [Gey17, p. 205].
To calculate $\boldsymbol{U}_{\mathrm{con}}$, an initial value of the radius ρ_{sph} is required for the first step. On the one hand, the radius should be as small as possible to minimize the calculation time. On the other hand, the radius should be as large as possible to find a solution at any time, i.e., at least one switch option should be within the radius. More details regarding the choice of the initial radius and the SDA in general can be found, e.g., in [KGK15], [Gey17, p. 204 ff.].

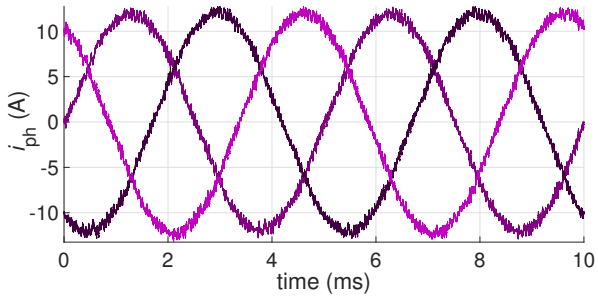
2.6 Summary and Performance Evaluation

A brief performance evaluation compares the control approaches considered with respect to the requirements defined in Section 1.1, especially for the electrical controlled system. For this purpose, the steady-state performance, i.e., a low THD and high steady-state accuracy, a highly dynamic response to transients, the computational effort and the robustness with respect to model, parameter and measurement inaccuracies are evaluated. For the steady-state operation (in this and the following chapters), two operating points are highlighted, one for nominal operation and one in the low partial load range. The used test bench is described in Chapter 3.
Here, some assumptions can already be made in advance. Comparing the two basic MPC concepts, i.e., CCS-MPC and FCS-MPC, the latter generally allows for higher dynamics, i.e., the best performance during transients when an inverter is present, due to its direct behavior. However, in steady state—at least in the partial load range of small electrical drives—, CCS-MPC (similar to FOC) enables lower current ripple, i.e., lower THD, compared to conventional FCS-MPC due to the modulator. Since from the modulator's point of view, both CCS-MPC and FOC use the same SVM to apply the controller output to the gate signals, the steady-state performance will be similar. Further comparisons of the two MPC approaches—using an explicit, i.e., pre-computed, formulation—are shown, e.g., in [PB13a], where both methods show similar performance, but also point out the lower computational effort when using CCS-MPC. However, as explained in Section 2.5.1, CCS-MPC requires an operating point linearization. With FCS-MPC, the latter is only required if a (M)IQP is used, see Section 2.5.2.2. Thus, in an implicit CCS-MPC implementation, i.e., optimization is performed at runtime, high-frequency online updating of the matrices used would be necessary to achieve acceptable control dynamics. Consequently, the computational effort of CCS-MPC increases compared to [PB13a].
Considering these arguments, only direct control approaches, i.e., DTC and FCS-MPC, compete with FOC as an exemplary procedure with modulator for the current control loop. Hence, LQR and CCS-MPC are only compared for the speed control loop in Chapter 9. The following evaluation serves to identify the challenges that prevent the requirements in Section 1.1 from being achieved and finally as motivation for the subsequent Chapters 3-9. In these, the disclosed challenges—see also Section 1.2—especially for DMPC, are successively addressed.

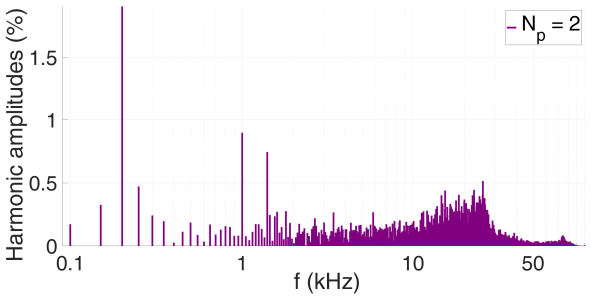
2.6.1 Steady-State Performance

To compare FCS-MPC and DTC with modulator-based methods, similar conditions have to be applied. This means that in steady state the same average switching frequency f_{sw} must be applied. Consequently, an increased control frequency is required for DTC and FCS-MPC compared to FOC with SVM. However, this can be quite challenging for small electrical drives. In the following use case, a control frequency of $f_{\mathrm{cf}} = 100\,\mathrm{kHz}$ is considered for the direct approaches where for FCS-MPC switching is penalized by using λ_{u} and for DTC the switching frequency is lowered by increasing $|\Delta T_{\mathrm{el}}|$. Nevertheless, the switching granularity for both direct approaches is $2T_{\mathrm{cf}} = 20\,\mu\mathrm{s}$, since in each control interval only one switching transition can appear, i.e., ON or OFF, see also Section 2.1. Thus, FCS-MPC and DTC have a maximum switching frequency of half the control frequency. Considering (2.13), it can be seen that this only happens if each half-bridge switches in each control interval. Since this is rather unrealistic, i.e., only one or two half-bridges switch per interval, f_{sw} is always lower.

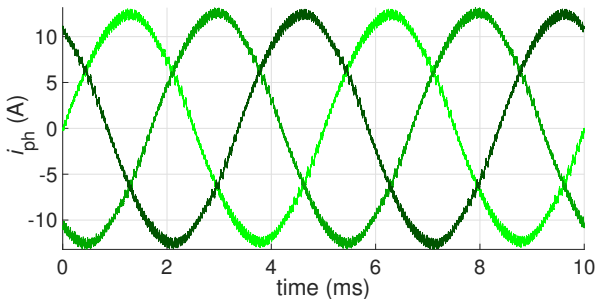
Fig. 2.25 shows the phase currents and spectrum at the nominal operating point for motor M1, see Table A.3 of Appendix A.3.1. Fig. 2.26 shows operation in the partial load range, i.e., at low modulation index. As can be seen, especially in the partial load range, the limited switching granularity becomes a substantial problem and thus disadvantage compared to modulator-based approaches. Phase currents and spectrum of DTC are shown in Fig. A.13 of Appendix A.5.



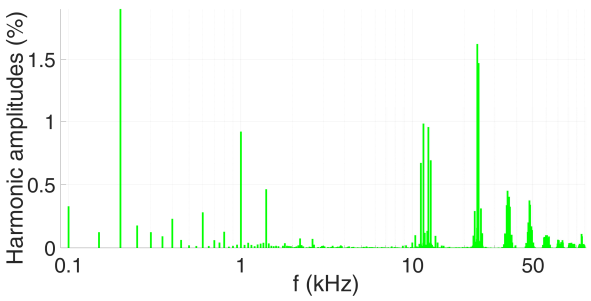
(a) Three-phase stator current for conventional FCS-MPC with $\lambda_{\mathrm{u}} = 0.45$, $f_{\mathrm{sw}} \approx 12.2\,\mathrm{kHz}$.



(b) Stator current spectrum for conventional FCS-MPC with $I_{\mathrm{THD}} = 4.41\,\%$.

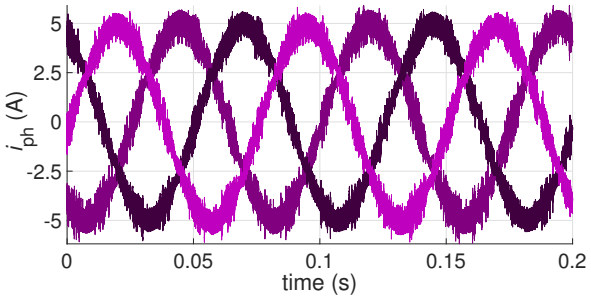


(c) Three-phase stator current for FOC and SVM with $f_{\mathrm{sw}} = 12.0\,\mathrm{kHz}$.

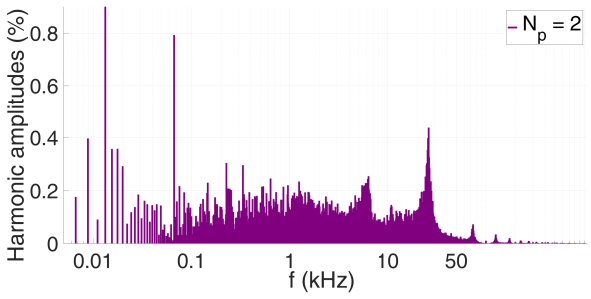


(d) Stator current spectrum for FOC and SVM with $I_{\mathrm{THD}} = 4.34\,\%$.

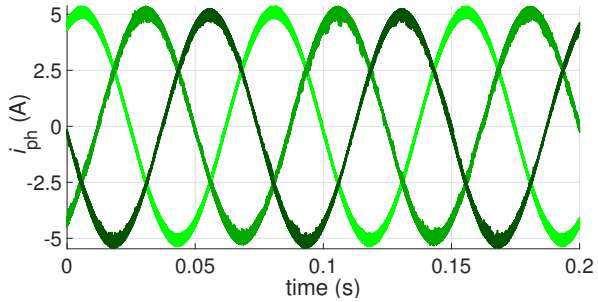
Figure 2.25: Motor M1: Three-phase stator current and spectrum for $i_d^* = 0\,\mathrm{A}$, $i_q^* = 12.16\,\mathrm{A}$, $n_{\mathrm{m}} = 3000\,\mathrm{rpm}$ with $f_{\mathrm{sw}} \approx 12.0\,\mathrm{kHz}$ (experimental).



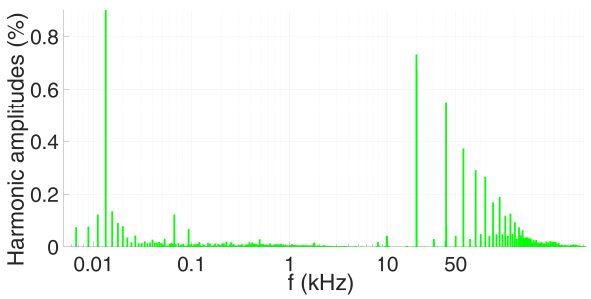
(a) Three-phase stator current for conventional FCS-MPC with $\lambda_u = 0.253$, $f_{sw} \approx 10.0\,\text{kHz}$.



(b) Stator current spectrum for conventional FCS-MPC with $I_{THD} = 8.77\,\%$.



(c) Three-phase stator current for FOC and SVM with $f_{sw} = 10.0\,\text{kHz}$.



(d) Stator current spectrum for FOC and SVM with $I_{THD} = 2.65\,\%$.

Figure 2.26: Motor M1: Three-phase stator current and spectrum for $i_d^* = 0\,\text{A}$, $i_q^* = 5.0\,\text{A}$, $n_m = 200\,\text{rpm}$ with $f_{sw} \approx 10.0\,\text{kHz}$ (experimental).

Fig. 2.27 illustrates the problem with the switching granularity, in a close-up view of the phase current. It becomes apparent that with both direct approaches the granularity is too low to achieve an acceptable current ripple, e.g., compared to FOC with SVM. Here, SVM has a switching granularity corresponding to the processor or FPGA clock. This is also claimed, for example, in [Hol16], where, despite high control frequencies, conventional FCS-MPC is thus recommended only for drives with low switching frequencies, e.g., less than 500 Hz.

In summary, when using conventional FCS-MPC, a very high ratio between control and (average) switching frequency must be maintained to obtain an acceptable current ripple. In [KG20],

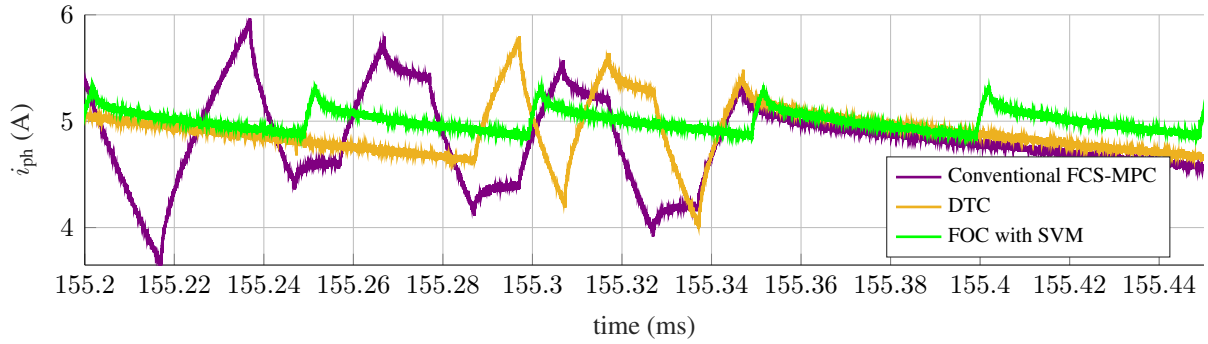


Figure 2.27: Motor M1: Single-phase stator current as close-up view from Fig. 2.26 and Fig. A.13 to show the switching resolution in comparison for operation using FOC with SVM, the conventional FCS-MPC and DTC in steady state for $i_d^* = 0\,\text{A}$, $i_q^* = 5.0\,\text{A}$, $n_m = 200\,\text{rpm}$ with $f_{sw} \approx 10\,\text{kHz}$ (experimental).

for example, the factor 100 is recommended. However, considering small electrical drives—with time constants of just a few ms or even µs—and state of the art calculation platforms, this ratio is quite challenging. Therefore, more advanced methods are required for FCS-MPC to increase the switching granularity, which can reduce the current ripple while maintaining the switching frequency. Hence, this will be addressed in the following of this work.
In case of DTC, a modulator stage is usually added for ripple reduction—see, e.g., [MM98]—but since it does not allow for additional optimization criteria such as current constraints, among other disadvantages compared to FCS-MPC, this will not be considered further.

2.6.2 Transient Performance

At a constant speed, a current reference step is applied to compare the transient performance of FOC, DTC and FCS-MPC. Fig. 2.28 clearly shows the advantageous dynamics of both direct control approaches, which avoid the operating point-dependent adjustment of the PI controller gains. The PI controllers were tuned according to the modulus optimum as described in Section 2.3, with the second FOC measurement showing a slightly increased k_p at the expense of minimal overshoot. In contrast, with FCS-MPC, assuming correct machine parameters, overshoots do not occur at any operating point, which is quite advantageous. In addition, FCS-MPC takes current constraints directly into account in the cost function, so that limit violations are avoided before they even occur. In contrast, controllers such as FOC require a control signal limitation in the form of clamping or back-calculation, which reacts as soon as the limitation has already been reached. In summary it can be concluded that when using FCS-MPC—similar to DTC—the maximum dynamic range can be achieved as claimed in Section 1.1.
In the following chapters, the ITAE criterion (integral of time multiplied by absolute value of error) [LW14, p. 461] is applied to the measured step responses to compare the dynamic control performance of the considered control methods. Here, the control error ξ is integrated over time,

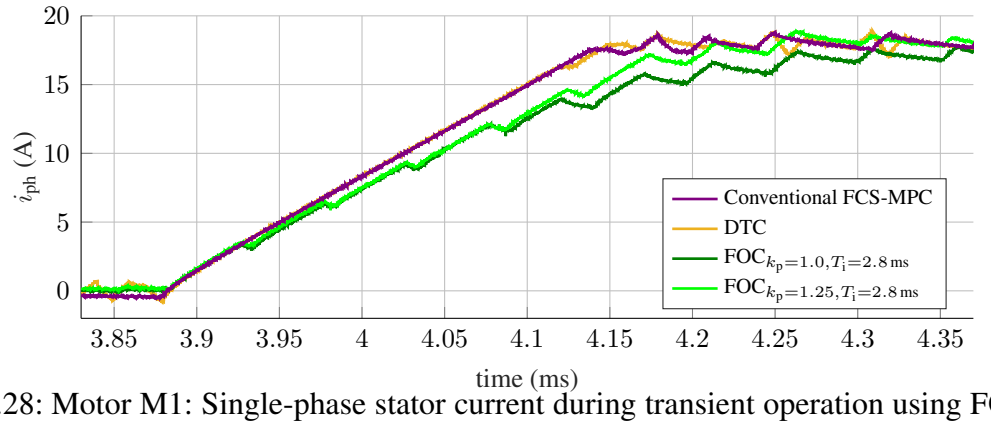


Figure 2.28: Motor M1: Single-phase stator current during transient operation using FOC with SVM, FCS-MPC with $N_\mathrm{p}=2$ and DTC for a reference step from $i_q^*=0\,\mathrm{A}$ to $i_q^*=18.24\,\mathrm{A}$ with $i_d^*=0\,\mathrm{A}$, $n_\mathrm{m}=200\,\mathrm{rpm}$, $f_\mathrm{sw}\approx 10\,\mathrm{kHz}$ (experimental).

with a control deviation being weighted more heavily as time progresses.

$$Q_{\text{ITAE}} = \int_0^\infty |\xi(t) - \xi(\infty)| t \mathrm{d}t \tag{2.90}$$

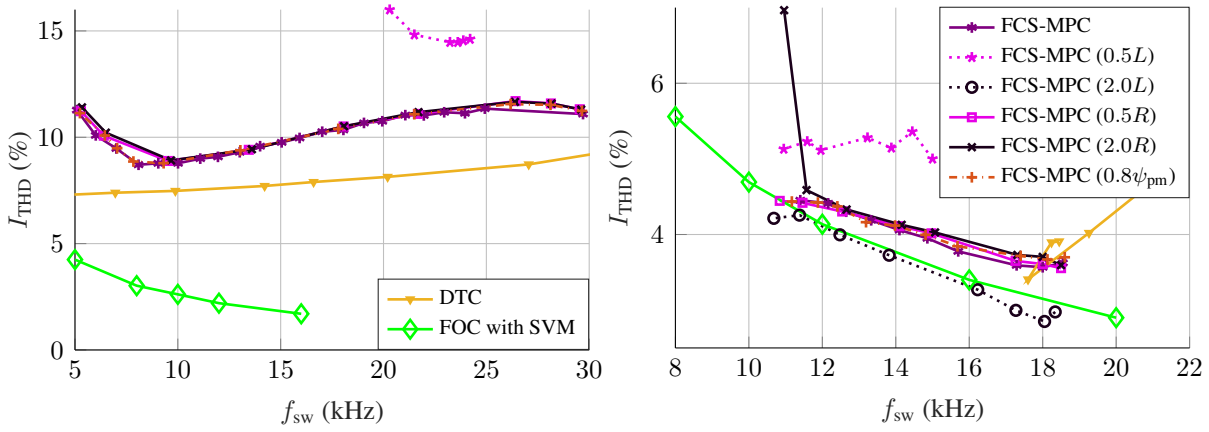
2.6.3 Influence of Model, Parameter and Measurement Inaccuracies

A model-based control algorithm is highly dependent on the accuracy of the model (see Section 2.2.1.2) and used parameters. The model requires a compromise between model accuracy, i.e., which effects of the controlled system are modeled and which are not, and computational load. Thus, as detailed as possible, but no more detailed than necessary. The parameter accuracy should generally be as accurate as possible. However, since not all parameters have the same influence on the quality of the control, a detailed evaluation is useful to determine which parameters need to be determined with particular accuracy. Focusing on FCS-MPC, Fig. 2.29 shows the stator current I_{THD} as a function of the average switching frequency f_{sw} for different parameter mismatches[20] at different operating points compared to the classical approaches. Once for a low modulation index, i.e., Fig. 2.29(a), and once for the nominal operating point, i.e., Fig. 2.29(b). Figs. 2.30-2.31 show the steady-state deviation for the two operating points respectively. Finally, Fig. 2.32 shows the transient behavior for a parameter mismatch.

Resistance In case of $0.5R$ the resistance in the machine has halved, e.g., due to a cold environment or insufficient determination. In case of $2.0R$ the resistance in the machine has doubled, e.g., due to the heating of the machine during operation, as explained in Section 2.2.1.2. The influence of the resistance on the quality of control is small, even if a noticeable steady-state deviation becomes visible as soon as a high current is present, see Fig. 2.31. In transients, however, no influence is recognizable, see Fig. 2.32(b).

Inductance In case of $0.5L$ the inductance in the machine has halved, e.g., due to saturation, as explained in Section 2.2.1.2. In the case of $2.0L$ the inductance in the machine is twice the value assumed for the prediction model. This can happen, e.g., due to an insufficient parameter assumption. In the first case, i.e., $0.5L$, the current ripple increases, as can be clearly seen in Figs. 2.30-2.32(a). Since the current increase per interval, i.e., the gradient, is steeper than expected by the model, switching appears too late. Consequently, the current ripple increases while the controller tries to counteract by additional switching operations. Due to the inaccurate prediction, however, without success. As a result, the THD increases for a defined switching frequency, see Fig. 2.29. In the second case, i.e., $2.0L$, the current ripple may even decrease for

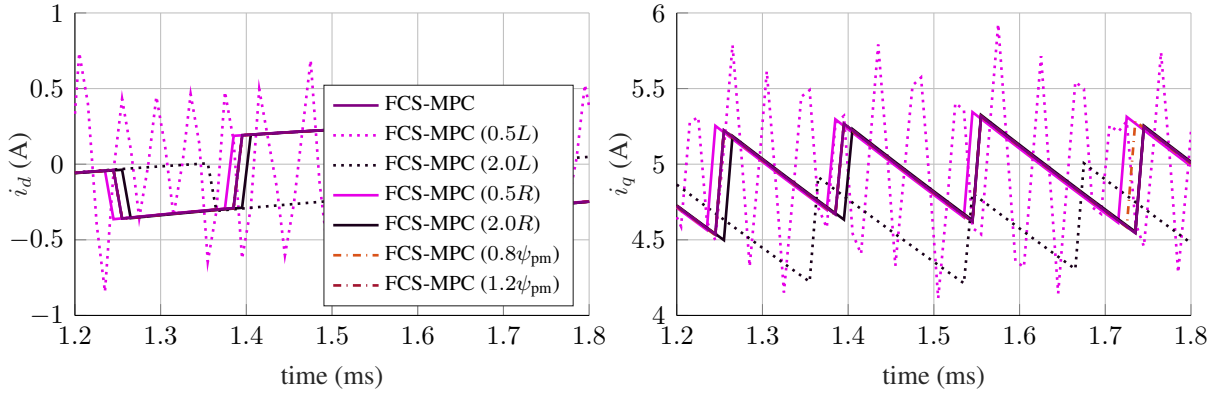
[20] For the study, the parameters in the model were changed while those of the machine remained the same. However, to be closer to the physical behavior under real conditions, the labeling in the following figures shows a change in the machine parameters.



(a) Trade-off for i_d^*=0 A, i_q^*=5.0 A, n_{m}=200 rpm.

(b) Trade-off for i_d^*=0 A, i_q^*=12.16 A, n_{m}=3000 rpm.

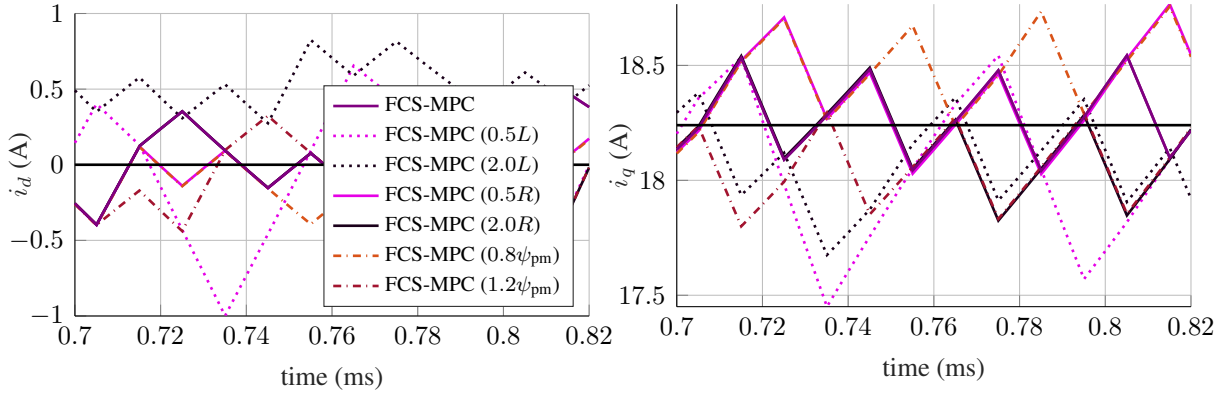
Figure 2.29: Motor M1: Trade-off between stator current I_{THD} and f_{sw} with different parameter mismatches for $N_{\text{p}} = 2$ (experimental).



(a) Steady-state deviation for the $d-$current.

(b) Steady-state deviation for the $q-$current.

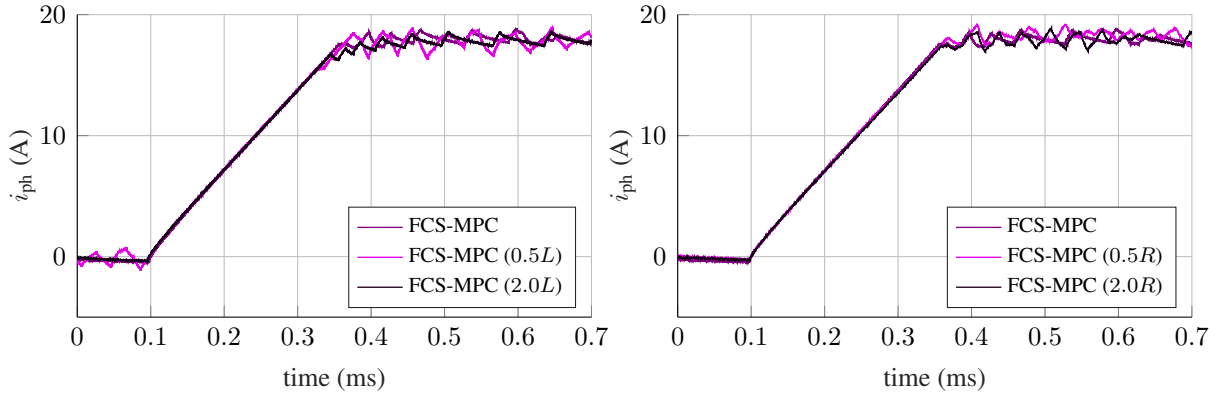
Figure 2.30: Motor M1: Steady-state deviation for $i_d^* = 0$ A, $i_q^* = 5.0$ A, $n_{\text{m}} = 200$ rpm with different parameter mismatches using $N_{\text{p}} = 2$ and a constant $\lambda_{\text{u}} = 0.045$ ($f_{\text{sw}} = 4.6$ kHz without parameter variations) (simulation).



(a) Steady-state deviation for the $d-$current.

(b) Steady-state deviation for the $q-$current.

Figure 2.31: Motor M1: Steady-state deviation for $i_d^* = 0$ A, $i_q^* = 12.16$ A, $n_{\text{m}} = 3000$ rpm with different parameter mismatches using $N_{\text{p}} = 2$ and a constant $\lambda_{\text{u}} = 0.112$ ($f_{\text{sw}} = 16.1$ kHz without parameter variations) (simulation).



(a) Parameter mismatch for the inductances.

(b) Parameter mismatch for the resistance.

Figure 2.32: Motor M1: Single-phase stator current during transient operation using the conventional FCS-MPC with $N_{\mathrm{p}} = 2$ and different parameter mismatches for a reference step from $i_q^* = 0$ A to $i_q^* = 18.24$ A with $i_d^* = 0$ A at $n_{\mathrm{m}} = 200$ rpm and $f_{\mathrm{sw}} \approx 10$ kHz (experimental).

certain switching frequencies, as shown in Fig. 2.29(b). This, however, happens at the expense of a steady-state deviation, as shown in Figs. 2.30-2.31. Moreover, in the case of $2.0L$ at low speeds it is not possible to switch at all at the desired frequencies and is therefore not shown in Fig. 2.29(a). The transient behavior is nevertheless relatively robust, even though Fig. 2.32 also shows the steady-state deviation for $2.0L$ and the increased ripple for $0.5L$.

Flux linkage constant In case of $0.8\psi_{\mathrm{pm}}$, the flux linkage caused by the magnets has reduced, e.g., due to the heating of the machine during operation, as explained in Section 2.2.1.2. In case of $1.2\psi_{\mathrm{pm}}$ the flux caused by the magnets has increased, e.g., due to a colder ambient temperature or insufficient parameter determination. Similar to the resistance, the influence on the control performance is small. However, at higher speeds, see Fig. 2.31, a steady-state deviation occurs. In steady state, at high speeds, it is more critical to have an inaccuracy in the flux linkage constant ψ_{pm} than in the resistance and the other way around at high currents and low speeds. No influence was observed for the transient behavior.

Moreover, the correctness of the measured quantities, i.e., $\boldsymbol{i}_{abc}$, φ_{m} and ω_{m}, is of great importance for the control performance[21]. Since all control methods considered are performed in the dq-system, rotor angle accuracy is equally important. However, current measurement has a bigger effect on the performance of MPC since it is the starting point for prediction. Both are addressed in Section 3.3, where the impact of an inaccurate alignment is illustrated in Section 4.4.2. The influence of an inaccurate speed is quite similar to the influence of an inaccurate ψ_{pm}, i.e., small and mainly affects the steady-state accuracy, see Figs. 2.29-2.31. For this reason, speed accuracy is of lower priority compared to the other two measured quantities.

[21] Although a voltage measurement can be used for optional parameter identification in Chapter 4, it is not necessary for the control and is thus not considered here. However, it is discussed in Section 3.3 and Section 4.4.2.

In summary It becomes obvious that especially the accuracy of the parameters has a decisive influence on the results of model-based control algorithms, i.e., DMPC, while it is expected that CCS-MPC behaves in the same way. Here, the inductance has the highest influence and is therefore the most crucial parameter, which can be explained by the fact that it defines the change of the current, i.e., the slope. Identical observations are shown, e.g., in [OKZ20] with simulation results. It is also shown there that longer prediction horizons tend to amplify the impact of inaccurate model parameters. Especially a mismatch for ψ_{pm} leads to a significantly increasing steady-state deviation for longer horizons. The summary of all results leads to the conclusion that an underestimated parameter is to be preferred, since the consequences are less critical, i.e., the THD is less influenced and mainly a steady-state deviation occurs. Unfortunately, both the saturation for the inductance and the temperature for ψ_{pm} have exactly the opposite tendency, i.e., they lead to overestimated parameters.
Moreover, experimental measurements have shown that motor parameters specified in the data sheet can deviate from the real values by up to $100\,\%$. This may be caused by inaccurate manufacturer's specifications, manufacturing inaccuracies or parameter changes during operation.
In case of conventional DMPC, there are several approaches that address the adverse impact of inaccurate parameters on the controller performance, see, e.g., [SKR17, LZW$^+$20, ZZZ19, CTBZ19, YZZ20]. For instance, a cost function is proposed in [LZW$^+$20] where an integral term of the current tracking error is added to equip the control scheme with an element of integrating nature. The associated control gains, however, are tuned heuristically, implying that optimal performance for the whole operating range is not ensured. [ZZZ19] superimposes an observer, but due to the convergence of the algorithm—which requires several fundamental periods—it is not suitable for dynamic processes. In [CTBZ19] and [YZZ20], model-free—also called non-parametric—MPC is used to tackle potential model mismatches. In principle, these model-free approaches are based on the difference between the last measured and previously predicted current. Moreover, they have to be averaged over several samples to avoid a negative influence of measurement outliers. Thus, parameters that are initially determined inaccurately or are slowly varying, e.g., due to temperature drifts, can be compensated for very well. However, in case of, e.g., an increasing current reference, a significant degree of magnetic saturation can be reached within a few control steps. If this event is to be covered, i.e., predicted over the horizon, the knowledge of previous control steps does not provide any valuable information. Hence, it can be claimed that a different approach needs to be devised that will enable accurate predictions both during transients and in steady state, shortly after a transient.
Furthermore, using a non-parametric approach, it is not possible to distinguish between different parameter influences, i.e., it cannot be decided whether it is an influence of, e.g., the resistance, the inductance or a rotor position-dependent and thus cyclic effect. As can be imagined, for

optimal predictive control over the entire operating range, it is necessary to separate the crucial parameter changes. To address that, this thesis aims to increase modeling and parameter accuracy rather than correcting simple models by adding averaged model deviations from the past. Consequently, it seems reasonable to automatically determine the parameters of each drive under real conditions offline—and some of them even online—to provide an ideal basis for MPC. This is addressed in Chapter 4. Subsequently, the increased prediction accuracy can be used to achieve improved control performance, as will be demonstrated, e.g., in Chapter 7.

2.6.4 Resource and Timing Evaluation

One of the main obstacles to the use of MPC is the implementation, i.e., real-time calculation. Especially since, as shown in the previous sections for small drives, the average switching frequency should be at least twice to ten times as high as $1/\tau_{\mathrm{el}}$ to avoid high current ripple.
Although not used for the measurements shown previously, a digital signal processor (DSP) is typically used to execute the control algorithms. Once the control-independent functions, e.g., reading measurements and dq-transformation, have been subtracted from the total processing time (see dashed lines), Fig. 2.33 shows the required calculation time for the considered control methods. In case of MPC, the calculation effort is shown as a function of the horizon. For CCS-MPC, Fig. 2.33 shows an explicit approach for the calculation of the unconstrained solution. Here, the computational effort increases linearly with the horizon, while for a constrained solution it would increase significantly depending on the operating point, i.e., the active constraints. For FCS-MPC, the calculation effort increases exponentially over the horizon with $\mathrm{SV}^{N_\mathrm{p}}$ when using exhaustive search. The exemplarily calculated FCS-MPC with $N_\mathrm{p} = 1$ allows a maximum cyclic interrupt of 14 kHz. Even though the computational effort depends on the solver used, it can be stated that the computational effort of MPC exceeds classical FOC by far. The question thus arises as to how such promising, but also computationally intensive algorithms, can be computed more efficiently. For this reason, the following Chapter 3, as the first contribution of this thesis, presents some conceptual considerations for an improved implementation.

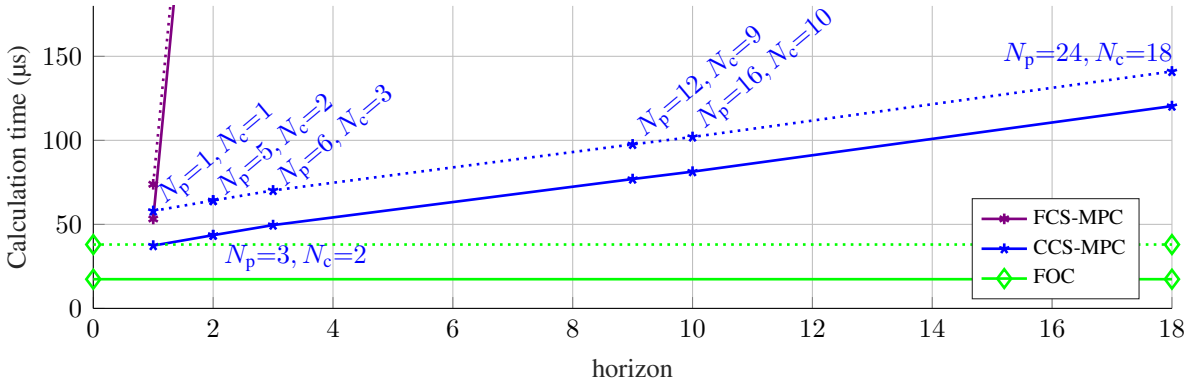


Figure 2.33: Calculation effort for implicit FCS-MPC, explicit CCS-MPC and FOC using a DSP (TI F28M36 @ 150 MHz). The dashed lines show the total computational load and the solid lines those of the control method used (experimental).

3 Heterogeneous Real-Time Computing Platform for Control Algorithms

Nowadays, almost all control algorithms in the field of power electronic systems are implemented on digital platforms that run in discrete time steps. In the field of power electronics, two main challenges can be highlighted. First, the development of new topologies, e.g., multilevel and multi-phase converters, has increased rapidly in recent years. Their ever-increasing demand for switching signals and the need to measure more states make it difficult to simultaneously handle this large amount of I/Os as well as the resulting computational complexity in a single processing system [MUC+19]. Second, new and more complex control strategies, such as DMPC, are used for common topologies, allowing the consideration of new objectives that have been considered secondary in the past. Depending on the control objective, this can lead to an increase in efficiency, a more dynamic step response or the control of additional criteria with the same time constant without cascading control loops [LKG+17]. This is usually accompanied by a higher computational effort, since complex calculations have to be performed in time intervals of a few microseconds, see Fig. 2.33. Therefore, in research and development, both academic and industrial, a powerful and flexible control platform is essential.

For this purpose, several commercial and self-made solutions have been developed in the past. A detailed evaluation of existing commercial and academic control platforms is provided in [WGL+19]. In general, a distinction can be made between software-based controllers running on a processor, e.g., DSP or ARM, and hardware-based controllers implemented on an FPGA. A processor is usually cheaper to purchase, enables efficient floating-point calculations, is easy to program with short compilation times, but is also usually relatively limited in computing power for hard real-time control due to the serial execution of calculations. Thus, in software-based implementation, even quite simple control algorithms, such as FOC, are limited to control frequencies of tens of kHz, e.g., $30\,\mathrm{kHz}$ in Fig. 2.33 or $40\,\mathrm{kHz}$ with a DSP running at $200\,\mathrm{MHz}$ in [Sto14, p. 42]. Higher frequencies, e.g., $100\,\mathrm{kHz}$, require code optimization or even more powerful processors. An FPGA enables enormous computing operations by simultaneous serial as well as parallel execution, but in the past it was usually also more expensive and more complex to program and debug. Despite more complex control algorithms, i.e., DMPC, FPGAs allow an execution frequency of up to several hundred kHz [WDK17b], [Sto14, p. 27].

However, all previous solutions focus either on an implementation, i.e., closing the MPC control loop, in the processor [Sto14, KJP11] or on the FPGA [LB13, LB14, SGG17, DMGK20]. In case of connected systems, i.e., processor and FPGA are used simultaneously, latency, jitter and complexity are usually a major problem when interacting [WGL+19].

For overcoming these issues, thanks to Moore's Law, new and powerful calculation platforms such as system-on-a-chip FPGAs (SoC FPGAs) can be used, which have recently become available on the market and are becoming increasingly affordable as well as popular. The combination of processor and FPGA with shared resources in one chip enables heterogeneous computing. This facilitates the multidimensional execution of the algorithms, e.g., parallel prediction and serial solution of the optimization problem. Therefore, within the scope of this thesis, a real-time control platform based on an SoC FPGA with high performance capabilities was build from scratch. More concretely, two platforms were developed. The first system, based on a Zynq-7000 (XA7Z020), was first published in [WDK17b, WDK17a] and is briefly explained in the following. A second system, based on a Zynq Ultra-Scale+ MPSoC (XCZU9EG), is a significant advancement and was first published in [WGL+19]. The latter is shown in Fig. A.2 of Appendix A.3, while further information can be found in [WGL+19]. Both systems are used in the following. Within the context of the research work, however, it was a requirement to find an affordable, industry-oriented solution, so that, unless explicitly stated otherwise, all experimental measurements are performed on the first system. In the case that the performance was not sufficient, i.e., for longer horizons, system two is used, which serves as a feasibility study and is clearly marked in the respective measurement, see, e.g., Fig. 7.7(d) and Fig. 7.11(d).

Dedicated heterogeneous algorithms are developed hereafter to obtain a resource efficient calculation of the control problems, which are implemented using rapid control prototyping (RCP), see Section 3.2. The test bench and the used sensors are explained in Section 3.3.

3.1 Heterogeneous Computing on SoC FPGAs

The main advantage of heterogeneous computing in the context of control of power electronic systems is the implementation of the time-uncritical—but yet complex—parts of an algorithm on a processor, whereas the time-critical parts that can be parallelized are implemented on an FPGA, see, e.g., [WDK17b, DMGK20], by using high clock frequencies with low jitter and low latency. Splitting an algorithm into routines and procedures based on their best execution is therefore recommended and also implemented in this work, see Fig. 3.1. More precisely, besides other tasks, the parameter identification shown in the following Chapter 4 as well as the speed

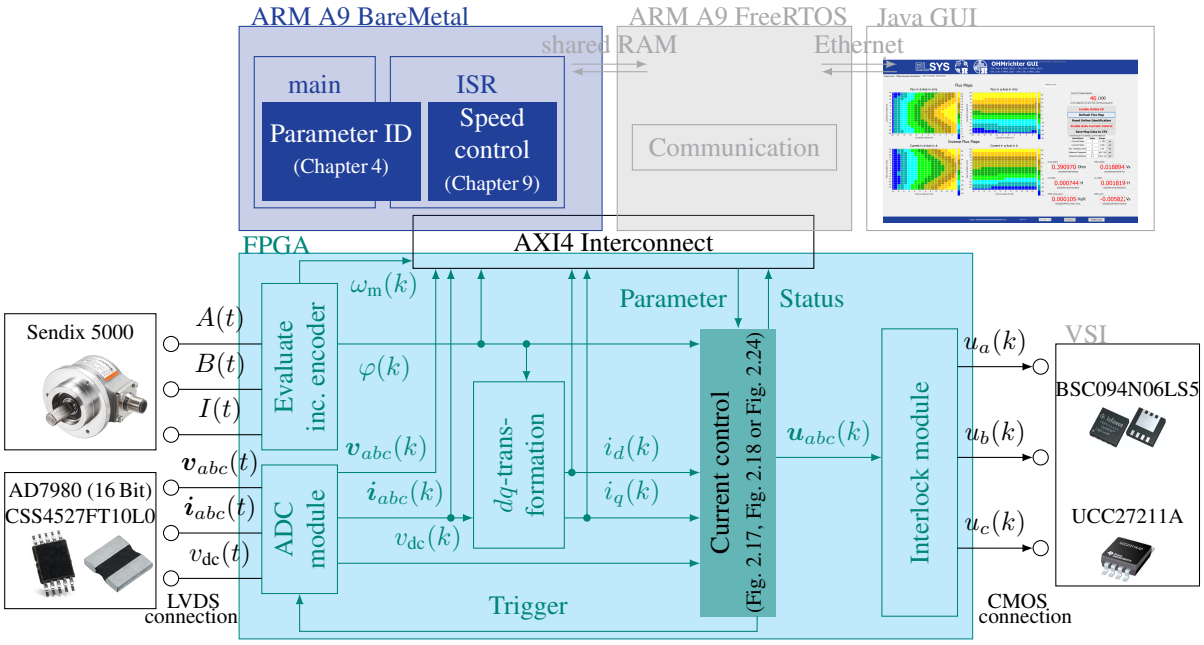


Figure 3.1: Schematic representation of the heterogeneous calculation platform with interaction between BareMetal, FPGA and further hardware when using a Zynq-7000.

control in Chapter 9 are performed on a processor. In contrast, the current control loop discussed in Section 2.6 and Chapters 6-8, an analog-to-digital converter (ADC) oversampling, the setting of interlock times and other minor tasks are implemented on the FPGA, as explained in the following. However, the current controller discussed in Chapter 7 is heterogeneous, i.e., the processor and FPGA share the computational load. The corresponding timing diagram is shown in Fig. 3.2, with the individual computational tasks discussed in more detail in the following chapters.

The main changemaker when using the proposed SoC FPGA is the AXI4 interconnect [Inc], which enables the interaction of the different computational units and thus a deterministic control with low latency and low jitter. Here, the proposed platform allows multiple channels for data exchange. The challenge is to decide which channel is best for the particular purpose. Consequently, it is decided to use the resource efficient AXI4-Lite for non-deterministic data exchange, e.g., reference values, the AXI4-Full interface for deterministic and fast (low latency) data exchange, e.g., the speed, and the AXI4-Stream for fast data copies to the memory. Although the respective transmission times depend on the amount of overall data traffic and the specific measurement case, a rough estimate can be made. On average, an AXI4-Lite read operation, i.e., the processor reads from an FPGA register, requires about 220 ns and for a write operation about 250 ns, both for 32 bits. An estimate for AXI4-Full yields 150 ns for the read and 170 ns for the write operation. For AXI4-Stream it is 50 ns for writing to the memory, i.e.,

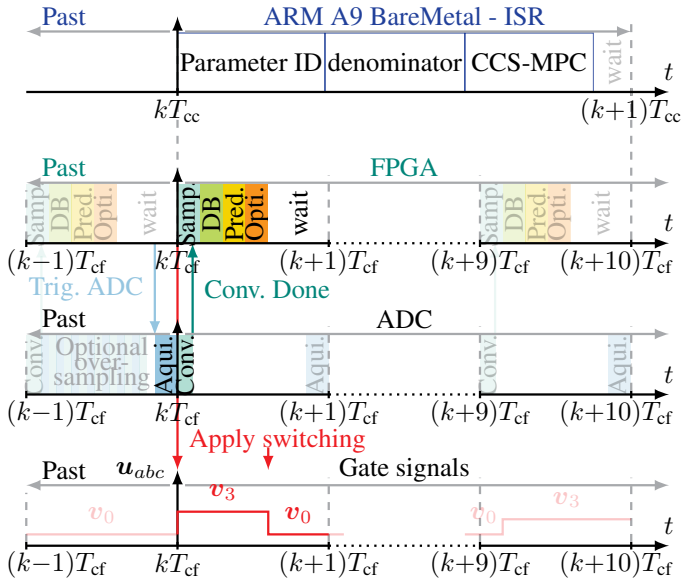


Figure 3.2: Timing sequence of the proposed heterogeneous calculation, using the example of the MPC algorithm in Chapter 9, where a factor of 10 is chosen between f_{cc} and f_{cf}.

on-chip memory (OCM), where the write transaction has 40 ns overhead and requires 10 ns for each 32 bit value. 67 ns are measured for the processor to read 32 bits from the OCM. Hence, AXI4-Stream becomes beneficial once large amounts of data are to be written to or read from a contiguous memory area. A more detailed AXI4 timing evaluation is proposed for the second control platform in [WGL+19]. Further information about AXI4 can be found, e.g., in [Inc].

3.1.1 Software-based Controller

As shown in Fig. 3.1, when using a Zynq-7000, there are two ARM A9 processors with different tasks. One of the processors runs a FreeRTOS operating system for communication tasks, including receiving reference values from the user and transmitting actual values to a remote GUI for visualization. The other core runs BareMetal with a main routine interrupted by a cyclic interrupt service routine (ISR), where a frequency of $f_{cc} = 10\,\text{kHz}$ is chosen for the considered drive systems. The BareMetal main routine e.g. parameterizes the MPC algorithm implemented on the FPGA by writing the references and parameters, i.e., λ_u, R_{ph}, $\boldsymbol{L}_{dq}$ or flux linkage maps. The BareMetal ISR reads all measurements from the FPGA and mainly performs the mentioned parameter identification and speed control. Both require a certain amount of processing capacity, but are not that time-critical so that operation at several kHz is absolutely sufficient. Consequently, rather complex out-of-the-box optimizers can be implemented quite easily manually or by using any RCP tool for C code generation, see Section 3.2.

Explained in more detail using the example of parameter identification, an online adaptation of the MPC parameters runs in parallel to the actual control task. Part of this is the adaptation of

flux linkage maps. Thus, the ISR detects steady-state operation of the drive and calculates the flux linkage changes as described in Section 4.3.4.1. In addition, flux linkage maps are subsequently adjusted in the main routine in a time-uncritical manner, where more sophisticated but also more accurate interpolation and extrapolation methods can be used, see Section 4.3.4.2.

3.1.2 Hardware-based Controller

The current control algorithm, e.g., MPC, is implemented primarily on the FPGA with a sampling and current control frequency of $f_{\mathrm{cf}} = 100\,\mathrm{kHz}$. This as well as even higher sampling and control frequencies, which are possible even with long prediction horizons, are realized by high clock frequencies in the FPGA, e.g., 100 MHz, and the possibility to parallelize the algorithm while some parts are still executed serially. Thus, thanks to the use of an FPGA as hardware accelerator, the computational burden of FCS-MPC discussed in Section 2.6.4 is enormously reduced, leading to the computation times shown in Section 9.4.2. To fully utilize the resources of the control platform as well as to implement the controller in a computationally efficient manner, the following principles are applied.

First, a fixed-point arithmetic is chosen for the FPGA. While this requires careful selection of value ranges, it has reduced the resource utilization by up to a factor of three—[RSF+22] shows factor two—compared to a floating-point implementation. However, the fixed-point value range is not a problem since it is chosen based on a normalized, i.e., pu, scaling.
Second, slow time-varying, i.e., time-uncritical, but yet complex calculations are outsourced to the ISR—which runs on the processor—to save resources. More specifically, divisions are not really efficient in real time and are a computationally intensive task, especially for the FPGA. For example, the denominator in (7.4) of Chapter 7 indicates the need to perform a division. However, since the denominator depends only on the speed, it can be assumed to be constant within several control intervals T_{cf}. Hence, to avoid the division, the reciprocal value of the denominator is calculated on the processor—see Fig. 3.2—and fed into the FPGA at the beginning of the prediction and optimization procedure.
Third, for divisions that cannot be outsourced to the processor and therefore must be performed on the FPGA, a carefully chosen bit size and fixed-point value range is used to keep the required processing resources modest. This applies, e.g., to the division in (6.4) of Chapter 6.
Fourth, efficient resource streaming and sharing is implemented on the FPGA. Therefore, subsequent prediction steps use the resources of the previous steps. This is illustrated in Fig. 3.3, which shows all core components of the current control loop distributed between the processor and the FPGA. The different control approaches are color-coded, where specific functionalities encapsulated by the use of soft intellectual property (IP) cores.

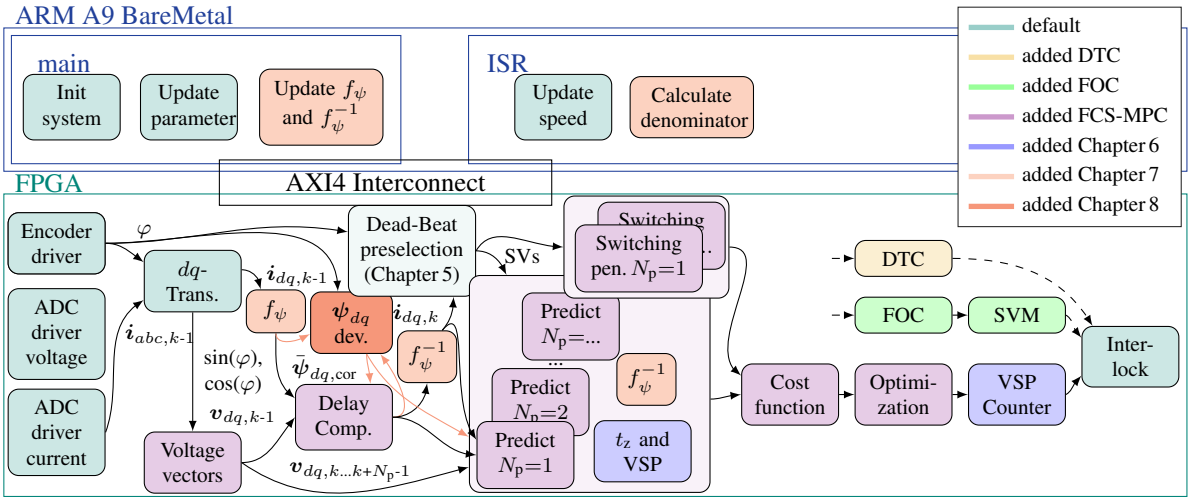


Figure 3.3: Schematic representation of the hardware description in the FPGA at IP Core level including the interaction with the processor for the current control loop. Additional tasks, e.g., speed control and parameter ID, can be used for all control methods considered but are not shown for reasons of clarity.

Some of these IP cores can be reused due to the natural calculation process. More precisely, within the first prediction step all discrete switching possibilities are calculated in parallel, see, e.g., line five of Algorithm 1. Subsequently, the used multipliers and adders, i.e., DSP slices of the FPGA, from the first prediction step can be reused in the second and subsequent predictions. A similar procedure is illustrated in Fig. 3.3 for the calculation of the switching penalization, i.e., last term of (2.75), which is calculated parallel to the prediction.

Another example for resource streaming—and heterogeneous computing—is the use of flux linkage maps. For reasons explained in more detail in Section 7.2, flux linkage maps for prediction are entirely implemented on the FPGA, while they are updated by the processor at arbitrary update rates using the previously identified flux linkage maps stored in the RAM, see Section 4.3.4. Thus, each map exists in two copies, while the one in the RAM has a more precise granularity. Since the use of such maps in the FPGA is more resource intensive compared to a simple multiplication that an inductance-based prediction would require, resource streaming is used. Thus both mapping tasks, i.e., f_ψ and f_ψ^{-1}, are implemented only once, but can be reused in each prediction step, see Fig. 3.3.

In addition, due to the possibility of bit shift operations, it is recommended to use a granularity with the power of two for the map grid points of the x- and y-axis, e.g., 0.5, 1, 2. As will be shown in Section 7.3, this enables extremely resource-efficient use of interpolation in the FPGA. However, even though linear interpolation is used for the maps in the FPGA, only $40\,\mathrm{ns}$—using a $25\,\mathrm{MHz}$ clock—are required for each conversion. Therefore, the additional clock cycles, i.e., time, are manageable despite streaming.

3.2 Generation and Optimization of C and HDL Code

While the basic software framework for the processors and the FPGA is written manually, the control algorithms are automatically generated as C or HDL code from the MATLAB/Simulink simulation environment using RCP, i.e., the MathWorks Embedded Coder and HDL Coder[1]. For the latter, only VHDL[2] generation is considered, as described in [WDK17b, WKDK19].
While the Embedded Coder is quite straightforward, the HDL Coder offers a number of optimization options for a resource and timing efficient FPGA implementation. Some of these options have proven to be very useful. If RCP is not selected, the same optimizations or even more can be chosen directly in Vivado, the development environment for the Xilinx SoC FPGAs. Here, the terminology resources comprises the basic hardware components of an FPGA, i.e., flip-flops (FFs), look-up tables (LUTs), digital signal processing (DSP) slices and block RAMs (BRAMs). The available and finally utilized resources are discussed in Section 9.4.2.

Regarding optimization, first, pipelining allows to split long calculation paths. In order to maximize the computing capacity of the FPGA, it is attempted to operate nearly all IP cores and AXI4 interfaces with the maximum clock. In the context of this work, the clock is limited to $100\,\mathrm{MHz}$, although theoretically up to $250\,\mathrm{MHz}$ (first system) or $1600\,\mathrm{MHz}$ (second system) would be possible. Assuming a calculation path with multiple additions and multiplications, this is not feasible in $10\,\mathrm{ns}$ since the signal and logic propagation delay for multiple operations add up. Thus, it is useful to insert a register that splits the calculation path into several parts, each running at the maximum clock, i.e., $100\,\mathrm{MHz}$. It is therefore recommended to set at least one input or output pipeline in Simulink for each IP core prior to HDL generation, although additional pipeline stages can be set within the IP core as desired. However, each stage implies an additional clock cycle.
Second, sharing and streaming DSP slices is recommended. While sharing reuses DSP slices regardless of functionality, streaming is mainly useful for uniform and repetitive structures such as for-loops. The latter is useful for MPC, as described in Section 3.1.2, since for each prediction step the entire implemented model equation can be reused, i.e., streamed. This saves valuable DSP slices.
Third, some multiplications with constant values, e.g., 2π, can be implemented in a more resource-saving way in the FPGA by means of bit shift operations. In the terminology of the HDL Coder this is called *Constant multiplier optimization* [Mat]. This saves valuable DSP slices.
Fourth, larger data structures, e.g., the flux linkage maps, can be stored to BRAMs, which saves FFs and LUTs.

[1] Although Xilinx's HLS is another promising HDL RCP tool, it will not be considered more closely.
[2] FPGAs are mostly described by using VHDL (very high speed integrated circuit hardware description language) or Verilog. VHDL is more widely used in Europe and is therefore applied in the following.

3.3 Test Bench

The corresponding experimental setup for Section 2.2, i.e., the Zynq SoC FPGA in combination with the two-level VSI and the PMSM with TMS, is illustrated in Fig. 3.4. This setup is used for experimental evaluation, where the first system, i.e., the Zynq-7000, is demonstrated. The alternatively used second system is shown in Appendix A.3. The evaluated PMSMs are presented in Appendix A.3.1 and the TMS in Appendix A.3.2. Since small electrical drives are discussed in this work, nominal voltages of $\leq 60\,\mathrm{V}$ are present. Therefore, metal-oxide-semiconductor field-effect transistors (MOSFETs) are used for the VSI.
In addition, the thesis addresses—in the context of MPC—how to achieve the most accurate and dynamic data acquisition and emphasizes the importance of correctness.

Current and voltage measurement The current measurement is performed in the path of the current-carrying phase as shown in Figs. 2.4-2.5. For the selection of the measurement a detailed evaluation of current sensors and ADCs in terms of accuracy, bandwidth and signal-to-noise ratio was carried out in advance, see [Bri17]. As a result, a shunt measurement with low-noise current sense amplifier is used.[3] Even if there is no galvanic isolation, a shunt current measurement is typical for small drives for reasons of space and costs. The current measurement uses a first-order low-pass filter with a cut-off frequency of $f_{\mathrm{g,i}} = 400\,\mathrm{kHz}$.
Due to the lack of galvanic isolation, a voltage measurement can be realized at low cost. The voltage measurement uses a first-order low-pass filter with a cut-off frequency of $f_{\mathrm{g,v}} = 211\,\mathrm{Hz}$. The voltage filter, i.e., phase shift and damping, is compensated for in the processor.

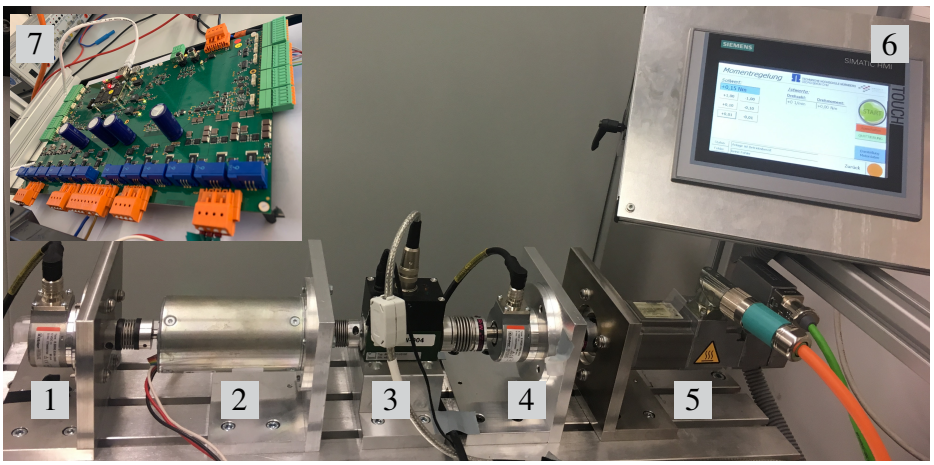


Figure 3.4: Test bench - (1) motor speed encoder, (2) test motor (PMSM), (3) torque measuring shaft, (4) load speed encoder, (5) load machine, (6) HMI for the load machine, (7) SoC FPGA platform, combined with the VSI.

[3] The fluxgate current sensors also shown in Fig. 3.4 are not used since they exhibit inherent oscillations at high sampling frequencies.

All phase currents and voltages as well as the dc-link voltage are sampled in parallel, i.e., no ADC multiplexer is used, which allows quite high sampling rates. These are triggered directly by the FPGA and read out with low latency and low jitter, while the circuit design also achieves very high signal integrity. For both platforms, 16-bit ADCs are used, each operating according to the successive approximation register (SAR) principle. The used components are listed in Table A.2 of Appendix A.3.
The experimental determined repeat accuracy of the overall current measurement is approximately $\pm 0.04\,\%$, i.e., $\pm 3\sigma = \pm\ 11\,\text{mA}$ for a current measurement range of $[-28, 28]\,\text{A}$. An external sensor was also used to confirm the accuracy, i.e., proximity to the true value.
Moreover, a linear approximation of the over-sampled current, as described in [Lan14, p. 55 ff.], is implemented, allowing an oversampling by a factor of eight at $f_{\text{cf}} = 100\,\text{kHz}$. In doing so, first, the gradient of the current can be determined [Lan14, p. 73], which could be used for a comparison with the predicted gradient, see Chapter 6. Second, a more accurate current value for the center and the end of the sampling interval can be achieved based on all sampled measurements. This can provide more robust behavior against signal noise, see [Lan14, p. 74 ff.]. However, the experimental comparison showed no noticeable improvement for the shunt measurement and inverter under consideration by using oversampling that would justify the additional resources in the FPGA required for the implementation. This comes from the already exceedingly good current measurement quality and the low dc-link voltage. For other applications, e.g., medium-voltage drives in combination with fast-switching semiconductors based on silicon carbide (SiC) or gallium nitride (GaN), there may be an advantage as higher dc-link voltages with higher $\mathrm{d}v/\mathrm{d}t$ may negatively affect the current signal. Such a benefit by using oversampling is demonstrated, e.g., in [DLS+18].

Position and speed measurement Since a rotating coordinate system was chosen for the control, the measurement of the rotor position is very demanding and crucial. In particular, the position, current and voltage values must be synchronous, i.e., their respective delay times must be compensated for so that they subsequently have the same time stamp. The latter is mandatory to work in the dq-system. An incremental quadrature encoder is used, which is read via an RS422 interface and evaluated by a corresponding IP core directly in the FPGA. Here the time between two lines is measured (beneficial at low speed) as well as the number of lines within a certain time window is counted (beneficial at high peed) with a smoothed transition between both methods. This yields ω_{m}, φ_{m} and φ. In Chapter 9 two incremental encoders are evaluated synchronously and with an identical IP core in parallel, thus achieving an equal time stamp.
For the correct adjustment of the rotor position sensor, i.e., the alignment of the sensor to the α-axis of the motor, a simple energization of the a-axis is not sufficient, since for the considered setup an angular inaccuracy of $\Delta\varphi_\Delta \leq \pm\,\pi/90\,\text{rad}$ can arise on average due to cogging

torque and friction. To achieve the smallest possible angular error, the procedure described in [RDD14], [Ric16, p. 45] is applied. Here, the test machine is accelerated to different speeds by the load machine and the voltages v_d and v_q are measured in steady state when the test machine has nearly zero current, i.e., $i_d^* \approx i_q^* \approx 0$. The same amount must result for the voltage components in both directions of rotation, while the sign also being identical in the d-axis, i.e., $v_q(\mathrm{n_m}) \approx \text{-}v_q(\text{-}\mathrm{n_m})$ and $v_d(\mathrm{n_m}) \approx v_d(\text{-}\mathrm{n_m})$. As can be seen in Fig. 2.12, v_d can also be nonzero due to the parallel iron loss resistance. The amount of v_q increases with speed. The offset of the electrical angle, i.e., $\Delta\varphi$, is varied until the above conditions are satisfied. This results in an experimentally determined precision and accuracy of $\Delta\varphi_\Delta \leq \pm\,\pi/3200\,\mathrm{rad}$ for the components considered on the test bench shown.

Respecting interlock times With respect to the VSI, it should be noted that in order to switch between the upper and lower semiconductor of a phase leg, a certain interlock time is typically provided to avoid a short circuit of the dc-link. An interlock time of $200\,\mathrm{ns}$ is chosen for the VSI used. This interlock time must be considered since it has a significant influence—at least for high calculation frequencies—on the accuracy of the predictions. The proposed FCS-MPC directly accounts for this due to its holistic approach by using a modified $\boldsymbol{v}_{dq}$—analog to [ITF+12]—in the prediction equation, see, e.g., (2.73), as soon as a switch position other than the last applied one is considered. For example, assuming that the control frequency is $f_{\mathrm{cf}} = 100\,\mathrm{kHz}$, the last applied switch position can remain for $200\,\mathrm{ns}$ (i.e., the time that corresponds to the interlock time), in the subsequent control interval, while the new discrete voltage space vector is applied for the remaining time, i.e., $9.8\,\mu\mathrm{s}$. However, due to the body diode of the MOSFET (or freewheeling diode in case of an IGBT), the kind of modification depends on the current direction, see [ITF+12]. Thus, there is also the case that a new discrete voltage space vector is applied for the entire control interval, i.e., $T_{\mathrm{cf}} = 10\,\mu\mathrm{s}$, despite the switch position change. Such a compensation method is quite straightforward in the case of FPGA-based FCS-MPC due to the cyclic $\boldsymbol{v}_{dq}$ update in the prediction equation. In this way, a possibly reduced voltage time area can be recognized during the optimization and thus a possible change of switching states, e.g., during transients, can be avoided.

This is in contrast to indirect control methods, i.e., when using a modulator. Here, the missing voltage time area must be compensated for by a displacement of the switching times subsequent to the calculation of the control signals. Consequently, the control stage and compensation are separated from each other without knowledge, which means that valuable voltage time area cannot be optimally utilized in transients.

Other nonlinearities such as diode effects are not considered, since no influence—within the scope of measurement accuracy—on the control and identification behavior was observed for the inverter under consideration.

4 Offline and Online Parameter Identification

This chapter describes both a procedure and the identification methods necessary for this purpose, which allow a reasonable determination of the parameters of the controlled system considered in this thesis. As demonstrated in Section 2.6.3, parameter accuracy has a significant influence on the control behavior of MPC. Therefore, it is necessary to design appropriate and robust identification methods.[1] The entire identification (ID) procedure is implemented on an ARM processor, which is part of the experimental hardware described in Chapter 3.

Overview of the identification methods In the field of parameter and system identification, a fundamental distinction is made between offline and online ID. Offline ID is mostly used for automated controller commissioning [Ric16, Vil07, Zou17, Bei00, Sch02, Gem15]. Here, an identification sequence is applied before operation, which identifies the parameters required for control. Online ID is a method for improving the control behavior during operation. For this purpose, previously made parameter identifications or data sheet parameters are optimized iteratively to determine operating point-dependent parameters or their changes during operation. Parameter changes can occur due to short-term effects such as temperature drifts or saturation as well as long-term effects such as wear and tear.

Furthermore, a distinction is made between excitation-free ID methods and those methods that rely on system excitation. The latter are particularly suitable for offline ID, since the controller does not have to follow any reference value during the offline process and thus almost any operating point can be approached. Nevertheless, the use of additional excitation signals to support online ID is also widespread. Here, as described for example in [Kel12], the additionally injected signals are used for excitation by manipulating the reference values. However, particular care must be taken to keep the influence of the applied injection signals as low as possible so that the original function of the system is not unnecessarily impaired. Therefore, in most cases the aim is to use the *natural* exitation during operation. This can be done, e.g., as described in [DSL$^+$19], by using the inverter switching or, as described in [BIFfMS14], by detecting parameter changes in transients when the system is sufficiently excited.

[1] The developed identification process was first simulated in MATLAB/Simulink and second used for RCP in order to generate C code. The evaluation in Simulink is supported by the Simscape and the Stateflow toolbox.

Depending on the evaluation of the measured data, a distinction is also made between steady-state and dynamic parameter identification methods [Bei00, p. 18]. A steady-state parameter identification assumes that during the estimation process, i.e., during a short period of time, the parameter to be identified, other parameters that may influence the estimation, and the system states, i.e., $x(t)$, do not change or change only insignificantly. For example, assuming that the thermal time constant is orders of magnitude larger than the duration of the measurement, temperature-dependent parameters such as resistance, permanent magnet flux linkage or friction can be identified at steady state, see, e.g., [BÏ4, HWD17]. More complex identification methods that measure a waveform or system response over a longer period of time during operation are also part of steady-state methods. An example of the latter is the signal injection method used in Section 4.3.1 similar to that of [Kel12, Hoe18]. In this method, a square wave signal is modulated onto the d-axis current and the change in voltage required for this is used to infer the prevailing machine parameters. Similarly, the identification procedures from [Vil07, Zou17, Bei00, Sch02], which rely on spectral analysis, are among the steady-state methods. They are the basis for the offline ID procedures described in Section 4.2.2. The advantage of steady-state methods is that they are generally easier to implement than dynamic methods since they are based on simplified system equations.
Moreover, they can be separated into a data acquisition and data processing part. This facilitates the implementation and processing of the algorithms in real time. The data acquisition, which is generally not computationally intensive, may run in real time on even less powerful control platforms and the computationally intensive data processing can be performed externally or in a time-uncritical task. Thus, for the latter, i.e., time-uncritical part, computationally intensive least square procedures and recursive least square procedures over large amounts of data [Vil07, Zou17, Bei00, Sch02] can be used for parameter ID or parameter correction [BAC+19].
A disadvantage of the steady-state parameter estimation can be the limited validity of the mostly simplified models. Thus, influences that have not been taken into account, e.g., iron losses, or changes in the boundary conditions can lead to measurement inaccuracies and thus parameter drifts. Hence, such uncertain operating ranges must be avoided.
A dynamic parameter identification takes into account and even requires a parameter or state change during the ID process, where several system states at different points in time are linked by an iterative procedure. As a result, dynamic methods can track parameter changes during runtime and immediately readjust the parameters in the event of transients or disturbances.
There are a variety of methods for dynamic parameter ID especially by using observer structures [Bei00, BIFfMS14, BLNMMT11, KBY18, Spe14]. The observer most commonly reported in literature is the Kalman filter, which is suitable in its general form for linear models and in its extended form for nonlinear models, both of which are used in time-domain prob-

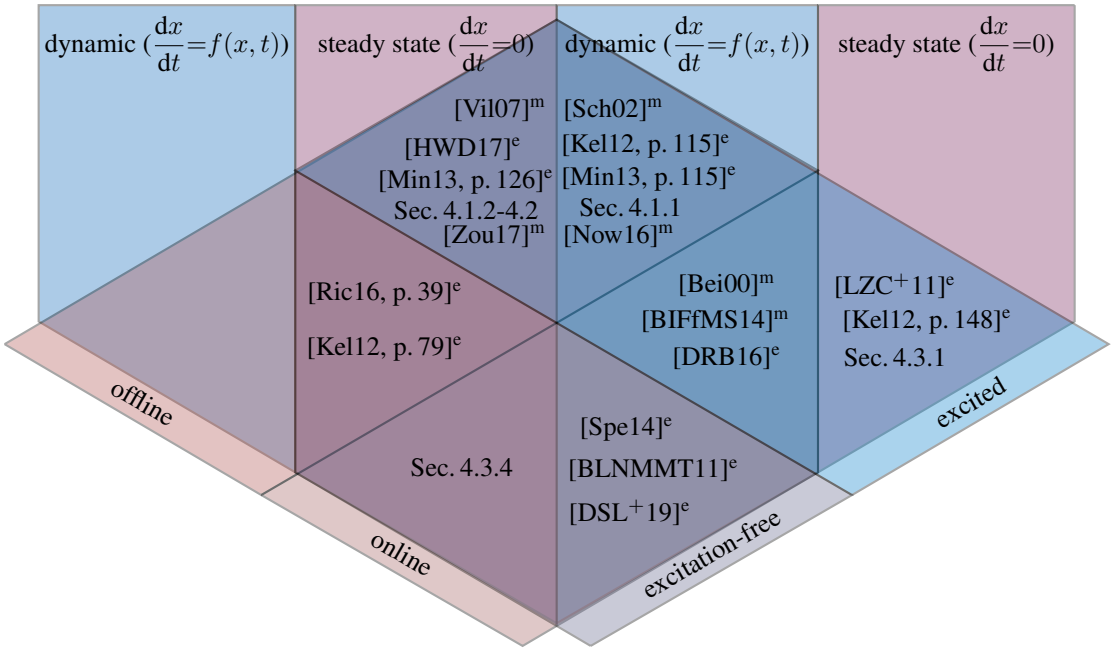


Figure 4.1: Overview and classification of commonly used methods for parameter identification.

lems. Since transfer functions are frequently used in control engineering, there are similar approaches in the frequency domain where a Luenberger observer is used, see, e.g., [Bei00, p. 31]. Besides these established procedures, further recursive identification procedures, such as recursive least squares (RLS) with *forgetting factor*, extensions of Kalman and Lueneberger observer by means of basic function networks or recursive neural networks, are described, e.g., in [Bei00, BNMMT08, Now16]. The disadvantage of these methods is the high computational effort, which, depending on the parameter under investigation, must also be performed in real time due to the iterative structure of the algorithms. Here, the accuracy of the discretization method also affects the result. In addition, inaccuracies of the model and noise parameters of the processed measurement signals must be estimated, which is physically possible only with high effort [Bei00]. Moreover, the dependence between several dynamically estimated parameters can cause parameter drifts, which, as described in [LZC+11], are due to the ranking problem of an under-determined system of equations. Another problem in dynamic parameter estimation, discussed, e.g., in [BIFfMS14] and [Bei00, p. 21], is ensuring sufficient system excitation so that a reliable parameter value can be estimated. Consequently, parameter changes by the observer can only be allowed when sufficient excitation is provided.
Finally, a distinction can be made between linear and nonlinear methods, see, e.g., [Bei00, p. 19], although this distinction is not done here.

For a better overview, Fig. 4.1 summarizes the different possibilities depending on the discussed criteria. The superscript „e“ stands for electrical and „m“ for mechanical parameter ID.

Selection of the used identification methods In order to select the identification methods to be used, an overview of the parameters of interest and the factors that affect them is required. Table 4.1 summarizes these parameters. In this context, it must be analyzed whether a parameter change is to be expected within the application dealt with in this thesis and thus an online identification is required or whether an offline identification is sufficient.

Table 4.1: Assessment of parameter relevance for the identification process.

Parameter	Influencing factors that cause a change during operation	Change in application probable?	Offline ID chosen?	Online ID chosen?
Linear machine model				
R_{ph}	temperature, contacting	yes	yes	yes
L_d, L_q	saturation (leaving the validity range of the linear machine model)	motor type specific	yes	no
ψ_{pm}	temperature	yes	yes	yes
Nonlinear machine model				
ψ_d	stator current, temperature	yes	no	yes
ψ_q	stator current	yes	no	yes
L_{dd}, L_{qq}	saturation	motor type specific	no	yes
L_{dq}, L_{qd}	cross-coupling	motor type specific	no	yes
Λ_d, Λ_q	rotor position	yes	no	no
R_{Fe}	stator current, speed	yes	no	no
Mechanical system parameters				
J_{m}	none	no	yes	no
J_ℓ	gear ratio, load	application specific	yes	no
c_{TMS}	loaded component length, aging	application specific	yes	no
d_{TMS}	temperature, aging	application specific	yes	no
T_{coul}, d_{fric}	load condition, speed, temperature, attrition	yes	yes	no
φ_Δ	none	no	yes	no

The parameters R_{ph} and ψ_{pm} must be identified online in any case, since they change with temperature and have a noticeable influence on the control performance, i.e., steady-state accuracy, of MPC, see Section 2.6.3. The absolute inductances L_d and L_q do not change their value over time as long as the validity range of the linear machine model is not left. Therefore, an offline ID is sufficient.

However, as explained in Section 2.2.1.2, the linear model of the magnetic circuit is not sufficient for most operating points. As shown in Section 2.6.3, the inductances have the most significant influence on the control performance, i.e., both in transients but especially in steady state. Hence, the flux linkage ψ_d and ψ_q or alternatively the differential inductances L_{dd}, L_{qq}, L_{dq} and L_{qd} are necessary. As the motor would have to be operated in the entire operating range to determine these parameters offline, this would be very time-consuming and often also impracticable. Thus, for reasons of usability, it is decided to perform the identification (adap-

tion) of the nonlinear magnetic behavior—based on the previously offline determined absolute inductances—online during operation as soon as an operating point is naturally reached. From the perspective of the model, this can be seen as a learning process. Since both nonlinear representations require current-dependent maps to be stored, it was decided to use the two ψ_d and ψ_q maps instead of the four differential inductance maps, as explained in more detail in Chapter 7. Moreover, rotor position dependencies are neglected for reasons explained in Section 4.3.4. Thus, an identification method can be chosen that averages the measured values over several rotor positions and rotor revolutions, thereby making the identified parameters more robust.

The iron loss effects are not modeled, as mentioned in Section 2.2.1.2. Two main reasons for not modeling the iron loss are given there. First, if R_{Fe} is used, as shown in the equivalent circuit diagram in Fig. 2.12, only steady-state operating points can be described mathematically, making R_{Fe} invalid for dynamic situations (and dynamic control approaches). In addition, since R_{Fe} depends on the d- and q-axis currents and the motor speed, a four-dimensional map would be necessary to adequately describe the effect. Consequently, this can be quite computationally intensive for real-time control. Second, the influence of iron losses on the investigated small PMSMs is relatively low, see Section 4.4.2. Therefore, an alternative approach to account for iron losses in the control loop is presented in Chapter 8.

Since a significant change of the mechanical parameters to be considered, i.e., J_{m}, J_{ℓ}, c_{TMS}, d_{TMS} and φ_{Δ}, is unlikely during operation in the context of the present application, the mentioned parameters are determined only offline. Merely for the friction, an online ID would be theoretically useful, since it changes with temperature, see Section 2.2.2.1. To determine whether this effect has a significant influence for small electric drives, it was determined using the example of motor M5, see Table A.3 of Appendix A.3. For this purpose, the idle friction characteristic was measured at ambient and at operating temperature using the offline ID described in Section 4.2.1. The temperature was measured at the bearing housing with thermocouples. As shown in Fig. 4.2, the temperature influence on friction is very small for the considered machines, i.e.,

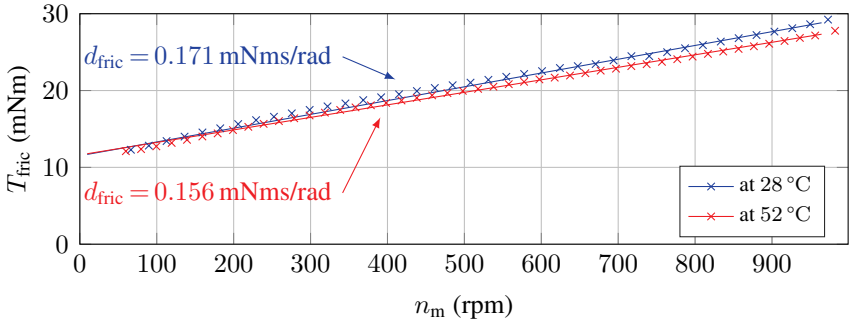


Figure 4.2: Temperature influence on the frictional torque, illustrated using the example of motor M5. Here, 8.7 % difference in the slope leads to 5.0 mNm deviation at n_{N}.

1.0 % of T_N for the worst operating point of motor M5. This confirms the assumption that the temperature influence on the friction can be neglected for the present application and thus the frictional torque only has to be determined offline.[2] Nevertheless, the offline ID can be executed repeatedly, i.e., whenever a temperature change is likely after a longer period of time. In addition, the speed controller, as described in Chapter 9, can compensate for possible friction deviations, since these occur quite slowly, both of which have been verified experimentally.

In summary, Fig. 4.3 shows the chosen parameter ID concept and the dependencies. Four possibilities must be distinguished. First (red), adopts a rigid system without elasticity. Second (dark blue), adopts an elastic, i.e., non-rigid, two-mass system. Third (bright blue), the electrical parameters are known, only the mechanical parameters of the drive train are identified. Fourth, the parameters are manually specified, i.e., are known.

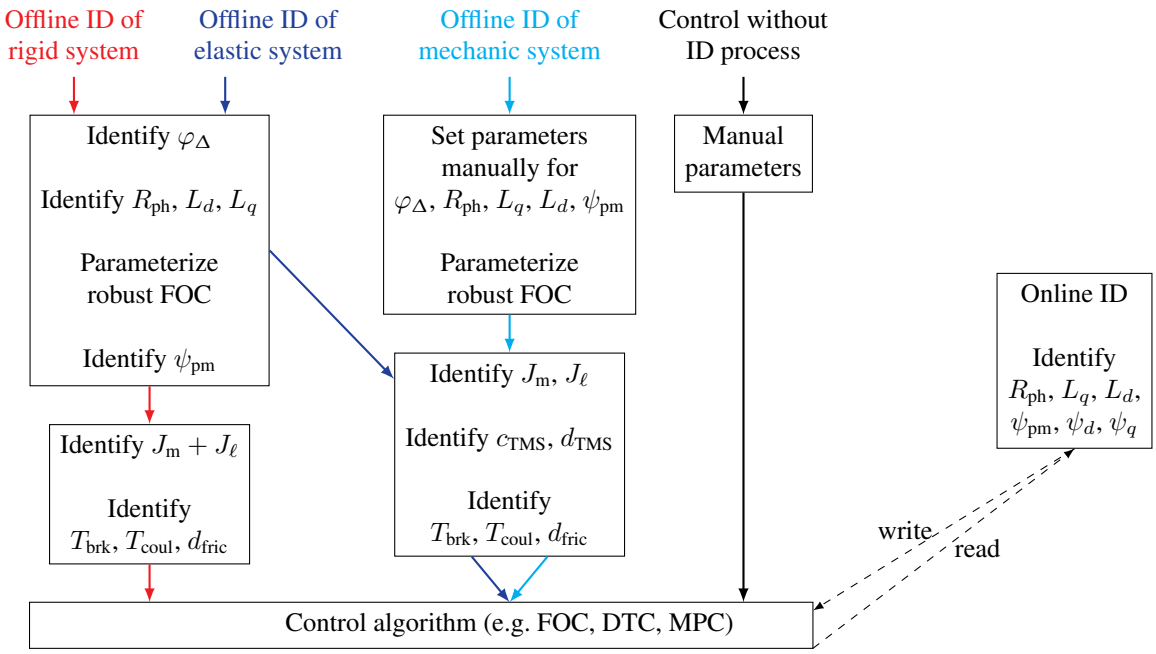


Figure 4.3: Structure of the parameter identification.

The identification procedures of the parameters to be determined are described in the following. The only requirement is that the number of pole pairs of the motor under consideration must be known at the beginning. The procedures were experimentally verified on up to five different types of PMSM—with different degrees of saturation, i.e., magnetic nonlinearity, and reluctance—from up to five different production batches in a power class below a nominal power of 200 W, see Table A.3 of Appendix A.3.

[2] It should be noted, however, that this assumption only applies to the frictional torque caused by the motor itself. For controlled systems with more complex load systems, e.g., with gearboxes, or extreme temperatures, e.g., −40 °C, it must be reconsidered [Rud12].

4.1 Electrical Offline Parameter Identification

A general overview of the electrical offline procedure is given in Fig. 4.4. The flow chart of the identification procedure shows the relevant identification steps for each parameter. The encoder and machine axis are aligned as described in Section 3.3. The experimental setup in Section 3.3 is assumed. The concept was first published in [HWD17, Hoe17, Hoe18]. The methodology for identifying the individual parameters is presented in detail in the following.

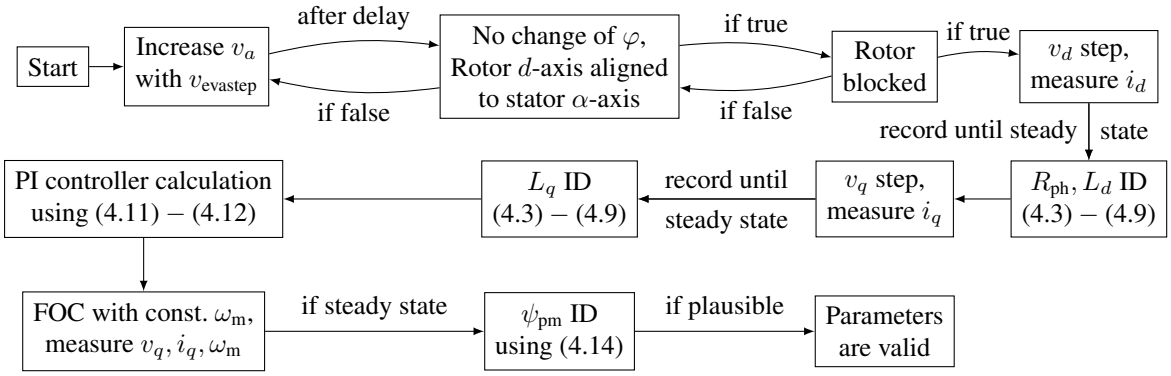


Figure 4.4: Flow chart for the electrical offline parameter identification.

4.1.1 Stator Resistance and Absolute Inductances

The phase resistance R_{ph} and the d-axis inductance L_d are identified while the rotor is at standstill. For this purpose, the rotor must first be aligned so that the d-axis lies on the α-axis by actively energizing the α-axis using the inverter, i.e., $\boldsymbol{u}_{abc} = [1 \ -1 \ -1]^T$. Assuming an initially unblocked rotor shaft, v_{a} is iteratively increased in each control cycle with a heuristically chosen step size, i.e., $v_{\text{evastep}} = 0.0002 V_{\text{dc}}$, up to I_{N} is reached to overcome the cogging torque. After alignment, (2.27) can be simplified at standstill to

$$v_d(t) = R_{\text{ph}} i_d(t) + L_d \frac{\mathrm{d}i_d(t)}{\mathrm{d}t} \,, \tag{4.1}$$

which describes a linear first-order system often used as the basis for estimating the parameters with RLS algorithms. However, RLS would require the numerical derivation of the measured current i_d, whereby the measurement noise has a negative influence on the parameters to be estimated. Therefore, a numerical minimization problem that avoids the derivation of measured signals is preferred instead. Assuming the initial condition to be zero, solving (4.1) leads to

$$\hat{i}_d(t) = \frac{v_d}{\hat{R}_{\text{ph}}} \left(1 - e^{-\frac{\hat{R}_{\text{ph}} t}{\hat{L}_d}} \right) \,, \tag{4.2}$$

where the calculated current $\hat{i}_d$ depends nonlinearly on the parameters to be estimated, i.e., determined.[3] Consequently, the error between the measured and calculated current, i.e.,

$$\xi(t) = i_d(t) - \hat{i}_d(t)\,, \tag{4.3}$$

must simply be minimized numerically to determine the parameters, i.e., $\hat{R}_{\text{ph}}$ and $\hat{L}_d$. Using the hardware in Chapter 3, after applying a voltage step in the d-axis, i.e., $\boldsymbol{u}_{abc} = [1\ -1\ -1]^T$, the step response of the measured current $i_d(k)$ is recorded and compared to the step response of the calculated current $\hat{i}_d(k)$ in each discrete time step, i.e., $\xi(k) = i_d(k) - \hat{i}_d(k)$. Thus a vector $\boldsymbol{\xi}(k)$ with a length of κ discrete steps—which covers the entire time range of interest, i.e., at least the duration of the step response—using an interval of T_{cc}, results. Doing so, a cost function with the sum of the squared errors over the entire recorded length is described by

$$J_{\text{LM}}(k) = ||\boldsymbol{\xi}(k)||_2^2 = \sum_{\ell=k}^{k+\kappa-1} \Big(i_d(\ell) - \hat{i}_d(\ell)\Big)^2 = \sum_{\ell=k}^{k+\kappa-1} \left(i_d(\ell) - \frac{v_d}{\hat{R}_{\text{ph}}}\left(1 - e^{-\frac{\hat{R}_{\text{ph}}\ell T_{\text{cc}}}{\hat{L}_d}}\right)\right)^2 . \tag{4.4}$$

Through the numerical minimization of (4.4) it is possible to obtain the parameters $\hat{R}_{\text{ph}}$ and $\hat{L}_d$. It is important to note that if the motor has a noticeable saliency, the motor shaft must be blocked to ensure both alignment in d-axis and standstill. Since it is often not known whether the windings are delta or star connected, the motor is always considered to be star connected.[4]

Levenberg-Marquardt algorithm For the minimization of (4.4), the Levenberg-Marquardt algorithm (LM) is used. In general the LM is a Newton method, which combines the Gauss-Newton method with the gradient descent method via a weighting factor and is described in detail, e.g., in [IM11, p. 524] or [SB17, p. 368]. Furthermore, in [Vil07, p. 52] an illustrative flow chart of the LM is shown. The LM is given as a particularly suitable solution method for nonlinear minimization problems due to its iterative adjustment of the weighting factor λ_{LM}. Here, the Jacobian matrix $\boldsymbol{J}_{\text{J}} \in \mathbb{R}^{\kappa\times 2}$ including all discrete time steps needs to be determined. For the sake of simplicity the Jacobian matrix in the continuous-time domain is specified by

$$\boldsymbol{J}_{\text{J,c}} = \frac{\partial \xi}{\partial \hat{\boldsymbol{\Theta}}_{\text{LM}}} = \left[\frac{-v_d(e^{\frac{-\hat{R}_{\text{ph}}t}{\hat{L}_d}} - 1)}{\hat{R}_{\text{ph}}^2} - \frac{v_d t e^{\frac{-\hat{R}_{\text{ph}}t}{\hat{L}_d}}}{\hat{L}_d \hat{R}_{\text{ph}}} \quad \frac{v_d t e^{\frac{-\hat{R}_{\text{ph}}t}{\hat{L}_d}}}{\hat{L}_d^2}\right], \tag{4.5}$$

[3] The measured or the inverter effect compensated applied voltage—for both the arithmetic mean over the entire measurement interval—is used. Due to the existing voltage measurement, see Section 3.3, both can also be performed and the difference formed. This difference can be used to approximately identify $R_{\text{DS(on)}}$ and R_{sh} since the voltage measurement point is after the shunt.

[4] Although circulating currents may occur in a delta connection of the windings and cause additional losses and saturation of the iron, these are not considered in the following.

which is the partial derivative of (4.3) with respect to the parameters $\hat{R}_{\text{ph}}$ and $\hat{L}_d$, where

$$\hat{\boldsymbol{\Theta}}_{\text{LM}} = \begin{bmatrix} \hat{R}_{\text{ph}} \\ \hat{L}_d \end{bmatrix} . \tag{4.6}$$

Afterwards, the LM iterates over several cycles, considering all discrete samples κ in each iteration. Here, in each LM iteration a new parameter approximation $\hat{\boldsymbol{\Theta}}_{\text{LM}}(k+1)$ based on the previously estimated parameters $\hat{\boldsymbol{\Theta}}_{\text{LM}}(k)$ and the correction vector is calculated, using[5]

$$\hat{\boldsymbol{\Theta}}_{\text{LM}}(k+1) = \hat{\boldsymbol{\Theta}}_{\text{LM}}(k) - \underbrace{(\boldsymbol{J}_{\text{J}}^T \boldsymbol{J}_{\text{J}} + \lambda_{\text{LM}}(k)\boldsymbol{I})^{-1} \boldsymbol{J}_{\text{J}}^T \boldsymbol{\xi}(k)}_{\text{correction vector } \Delta\hat{\boldsymbol{\Theta}}_{\text{LM}}} \tag{4.7}$$

in order to minimize the costs J_{LM}. $\boldsymbol{I} \in \mathbb{N}^{2\times 2}$ is the identity matrix. To avoid the complex calculation of the second derivative in the Hessian matrix $\boldsymbol{H}_{\text{LM}}$, it is approximated according to [SB17, p. 368] using the Jacobian matrix, i.e., $\boldsymbol{H}_{\text{LM}} \approx \boldsymbol{J}_{\text{J}}^T \boldsymbol{J}_{\text{J}}$.
For approximation, the weighting factor λ_{LM} is adjusted iteratively. By adding $\lambda_{\text{LM}}\boldsymbol{I}$ to the Hessian matrix, λ_{LM} adjusts how strongly the gradient or the curvature should be taken into account. For large values of λ_{LM}, the search direction is based more on the gradient. For small values of λ_{LM}, however, the curvature is increasingly taken into account. After each iteration step of (4.7), the cost function (4.4) is evaluated. If J_{LM} becomes smaller, i.e., the optimization step was a success, λ_{LM} is adjusted by

$$\lambda_{\text{LM}}(k+1) = \frac{\lambda_{\text{LM}}(k)}{\theta} \tag{4.8}$$

to a smaller value, where θ is the scaling factor. If J_{LM} becomes larger, i.e., the optimization step has not been successful, λ_{LM} is set to a larger value with

$$\lambda_{\text{LM}}(k+1) = \lambda_{\text{LM}}(k)\theta\,. \tag{4.9}$$

The iteration process is repeated for 100 iterations, which are shown to be sufficient for all considered motors, since the parameters are usually converged after at most 11 iterations.
For all considerations, $\theta = 10$ and $\lambda_{\text{LM},0} = 0.01$ is set initially, as recommended, e.g., in [SB17, p. 369]. Due to the necessary matrix inversion, the method is only suitable for simple system models or, as in the present case, for time-uncritical optimizations. The identification process is exemplified for motor M1, see Table A.3 of Appendix A.4.1.

The q-axis inductance L_q is identified in a similar way as L_d. In the standstill case, (2.27) can be rewritten to

$$v_q(t) = R_{\text{ph}} i_q(t) + L_q \frac{\mathrm{d}i_q(t)}{\mathrm{d}t}\,, \tag{4.10}$$

[5] For the considered drives $\kappa = 512$ (at $T_{\text{cc}} = 100\,\mu\text{s}$) is chosen for the length of $\boldsymbol{\xi}$ and $\boldsymbol{J}_{\text{J}}$.

where identical to (4.1), a first order system results. The identification of $\hat{L}_q$ is performed by evaluating a step response in the q-axis, i.e., $\boldsymbol{u}_{abc} = [Z \quad 1 \quad -1]^T$, by using the proposed LM algorithm. The latter identifies the stator resistance a second time. Although this is not necessary, it is still available and evaluated for statistical reasons in Section 4.4.1. Regardless of saliency, the rotor must be aligned and locked in α-axis position to prevent the rotor from rotating during the q-axis voltage step.

4.1.2 Permanent Magnet Flux Linkage

Robust FOC for identification Since at this stage of the identification process not all parameters are yet available for the correct adjustment of FOC (see Section 2.3), robust controller settings are required to identify ψ_{pm}. No dynamic requirements are necessary, but the settings should work in steady state for every motor. Thus, the desired bandwidth of the current controller, i.e., BW_{i}, is chosen to $270\,\text{Hz}$ ($\approx 1700\,\text{rad/s}$). Consequently, simple but also robust PI controllers for d- and q-axis current and speed control are calculated according to [Ins19] by using

$$k_{\text{p},id} = \hat{L}_d BW_{\text{i}}\,, \;\; T_{\text{i},id} = \frac{\hat{L}_d}{\hat{R}_{\text{ph}}}\,, \quad k_{\text{p},iq} = \hat{L}_q BW_{\text{i}}\,, \;\; T_{\text{i},iq} = \frac{\hat{L}_q}{\hat{R}_{\text{ph}}} \tag{4.11}$$

and

$$k_{\text{p},\omega} = \frac{k_{\text{p},id}}{\hat{L}_d 14 \frac{3p}{2} 300}\,, \;\; T_{\text{i},\omega} = \frac{14^2 \hat{L}_d}{k_{\text{p},id}}\,. \tag{4.12}$$

Afterwards the permanent magnet flux linkage, ψ_{pm}, is identified while the motor is controlled to a certain speed by using the robust FOC settings. The number of pole pairs p must be known. Assuming steady state, i.e., $\frac{\text{d}i_q}{\text{d}t} \approx 0$, and that the current in the d-axis is controlled to zero, i.e., $i_d \approx 0\,\text{A}$, (2.27) can be simplified to

$$v_q = R_{\text{ph}} i_q + \psi_{\text{pm}} \omega_{\text{el}}\,. \tag{4.13}$$

Assuming low speed[6], thus neglecting iron-loss effects, see Section 2.2.1.2, only the ohmic losses and the induced voltage remains. Using the previously identified stator resistance $\hat{R}_{\text{ph}}$, an approximation for the flux constant $\hat{\psi}_{\text{pm}}$ can be calculated by

$$\hat{\psi}_{\text{pm}} = \frac{v_q - \hat{R}_{\text{ph}} i_q}{\omega_{\text{el}}}\,. \tag{4.14}$$

[6]For the considered small drives a speed range of 200 - 1200 rpm, i.e., $n_{\text{m}} \in [5 - 30]\,\%$ of n_{N}, is assumed.

4.2 Mechanical Offline Parameter Identification

In addition to the electrical parameters, the mechanical parameters of the controlled system must be identified if they are required for control, see, e.g., Chapter 9. The mechanical ID process uses FOC, see Section 2.3, which is parameterized as described in the previous Section 4.1.2.

4.2.1 Frictional Behavior

The identification of the frictional behavior is based on the simplified friction model shown in Section 2.2.2.1. Particular care is required when estimating the validity of the model given in (2.31).[7] The simplified description requires three parameters, i.e., the breakaway torque, the coulomb and the viscous friction. However, since for the control approach presented in Chapter 9 only the latter two are of interest, the identification of the breakaway torque can be found in Appendix A.4.2.1. The results are based on [Geb18].

To identify the coulomb and viscous friction, the torque is recorded at no-load as a function of the speed. Since (2.31) assumes that the viscous friction d_{fric} is directly proportional to the velocity, it can be described as a linear regression line through the measured values. Here a positive offset in the torque axis can be considered as the coulomb friction T_{coul}.
Therefore, a heuristically defined initial speed $\omega_{\text{m,max}}$ is necessary, which should be in a speed range where iron losses do not dominate, e.g., $10\,\%$ of the nominal motor speed. After the speed controller reaches this speed reference ω_{m}^{*}, the latter is reduced with a step size of $\omega_{\text{m,max}}/N_{\text{visco}}$, where N_{visco} is the number of speed steps. The reference speed is kept constant at each step for a certain time interval, which should be greater than $3\tau_{\text{m}}$ to guarantee steady state for current and speed. After a steady-state operating point is detected, the current and speed are sampled over $\kappa = 512$ samples[8], averaged, and stored in a measurement array. The reduction of the speed reference stops as soon as the smallest speed value $\omega_{\text{m,min}}$ that can be detected by the encoder is reached. Although the recorded measurements are averaged over 512 samples, N_{visco} should be chosen large enough to ensure a high number of operating points to smooth out outliers. As shown with the experimental results in Appendix A.4.2.3, records with 200 data values, i.e., $N_{\text{visco}} = 200$, are well suited. A flowchart illustrating the ID process is shown in Fig. 4.5.

[7] The unconsidered iron loss resistance of the first harmonic adulterates the measured torque-generating current measurements. Due to the nonlinear increase of the iron losses with the frequency, as well as nonlinear effects of changing viscosity in the lubrication gap of the friction contact, higher speeds lead to a nonlinearity in the measured friction torque-speed curve, see Fig. 2.14. Thus, after every ID procedure, the recorded friction curve should be checked to see if the linear approximation in the recorded velocity range meets the requirements.

[8] The number of samples is chosen heuristically and depends on the available memory of the control platform. Above $\kappa = 512$ samples, no improvement was noticeable.

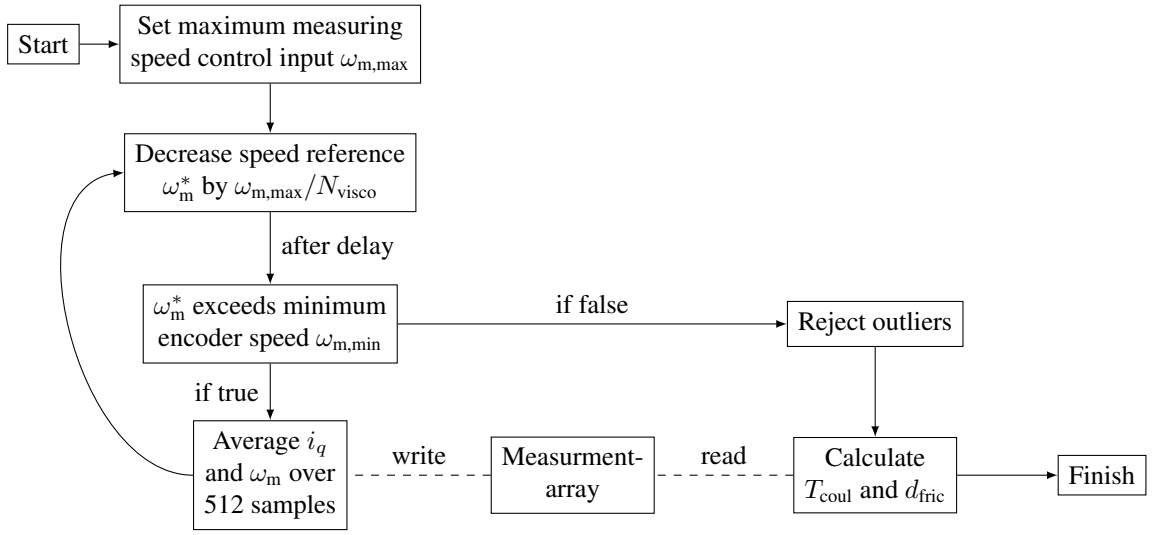


Figure 4.5: Coulomb and viscous friction torque ID flow chart.

By using the recorded measurement data, d_{fric} and T_{coul} can be calculated by solving a least square minimization problem, i.e.,

$$\underset{d_{\mathrm{fric}},T_{\mathrm{coul}}\in\mathbb{R}}{\text{minimize}} \sum_{\ell=k}^{k+N_{\mathrm{visco}}-1} \left(T_{\mathrm{m}}(k) - \left(d_{\mathrm{fric}}\omega_{\mathrm{m}}(k) + T_{\mathrm{coul}}\right)\right)^2 . \tag{4.15}$$

To prevent the error influence of outliers, measured values exceeding a heuristically chosen threshold of more than $30\,\%$ of their direct neighbors are excluded from the following regression process. The solution of the problem starts by calculating the mean value over all measurements of both states, i.e., speed and torque[9], with

$$\bar{T}_{\mathrm{m}} = \frac{1}{N_{\mathrm{visco}}} \sum_{\ell=k}^{k+N_{\mathrm{visco}}-1} T_{\mathrm{m}}(\ell) \tag{4.16}$$

$$\bar{\omega}_{\mathrm{m}} = \frac{1}{N_{\mathrm{visco}}} \sum_{\ell=k}^{k+N_{\mathrm{visco}}-1} \omega_{\mathrm{m}}(\ell)\,, \tag{4.17}$$

resulting in the two center coordinates $\bar{T}_{\mathrm{m}}$ and $\bar{\omega}_{\mathrm{m}}$ respectively. Afterwards the average value of the sum of torque and speed residuals product is calculated by

$$S_{\omega\mathrm{T}} = \frac{1}{N_{\mathrm{visco}}} \sum_{\ell=k}^{k+N_{\mathrm{visco}}-1} (\omega_{\mathrm{m}}(\ell) - \bar{\omega}_{\mathrm{m}})(T_{\mathrm{m}}(\ell) - \bar{T}_{\mathrm{m}})\,, \tag{4.18}$$

[9]The torque is calculated from the measured i_q, see (2.28), where i_d was controlled to zero.

where the variance over all speed values, i.e., averaged squared residuals, is calculated by

$$S_\omega^2 = \frac{1}{N_{\text{visco}}} \sum_{\ell=k}^{k+N_{\text{visco}}-1} (\omega_{\text{m}}(\ell) - \bar{\omega}_{\text{m}})^2 \,. \tag{4.19}$$

The minimization problem (4.15) can be solved afterwards, where d_{fric} is calculated by using

$$d_{\text{fric}} = S_{\omega\text{T}}/S_\omega^2 \tag{4.20}$$

and T_{coul} is calculated by using

$$T_{\text{coul}} = \bar{T}_{\text{m}} - d_{\text{fric}}\bar{\omega}_{\text{m}} \tag{4.21}$$

based on the intersection of d_{fric} with the y-axis with the help of $\bar{\omega}_{\text{m}}$ and $\bar{T}_{\text{m}}$.
Experimental measurements and a detailed evaluation of the friction identification procedure can be found for two different PMSMs in Appendix A.4.2.3. Based on this evaluation, the standard deviation of the ID procedure can be assessed with smaller than $4\,\%$ for the breakaway torque (see Appendix A.4.2.2), with smaller than $3\,\%$ for coulomb friction and with smaller than $5\,\%$ for the viscous friction. The simplification of a linear friction curve thus fits very well in the investigated speed range to the accuracy required for control purposes.

4.2.2 Two-Mass System

For the identification of the parameters J_{m}, J_ℓ, c_{TMS} and d_{TMS}, the analytical transfer function G_{ana} from the motor torque T_{m} to the motor speed ω_{m} of the elastically coupled drive train, as described in Section 2.2.2.2, is used. The aim of the identification process is to compare the analytical frequency response of the TMS, i.e., G_{ana}, with a measured frequency response of the real drive train, i.e., G_{exp}, over the widest possible range of excitation frequencies.
By means of a numerical optimization, i.e., using the LM algorithm, the requested parameters in the analytical frequency response G_{ana} are iteratively adapted until the two frequency responses (the amplitude responses), i.e., G_{ana} and G_{exp}, match. The proposed method is similar to [Vil07, Zou17], but has a different solution procedure, i.e., using an undamped analytical solution for fast convergence, adapted for small drives and was first published in [Geb19].

The designed procedure takes place in several sequential steps and can be summarized as follows.[10] First, the drive train is excited in a frequency range that must be sufficiently high to

[10] An alternative, computationally more efficient identification possibility for J, applicable only for one-mass systems and not pursued herein, is presented in [HWD17]. Here, only a single frequency is excited, which allows to calculate the Fourier coefficients only at this particular excitation frequency and not for the whole spectrum. A computational efficient way to calculate the parameters of interest is the Goertzel algorithm.

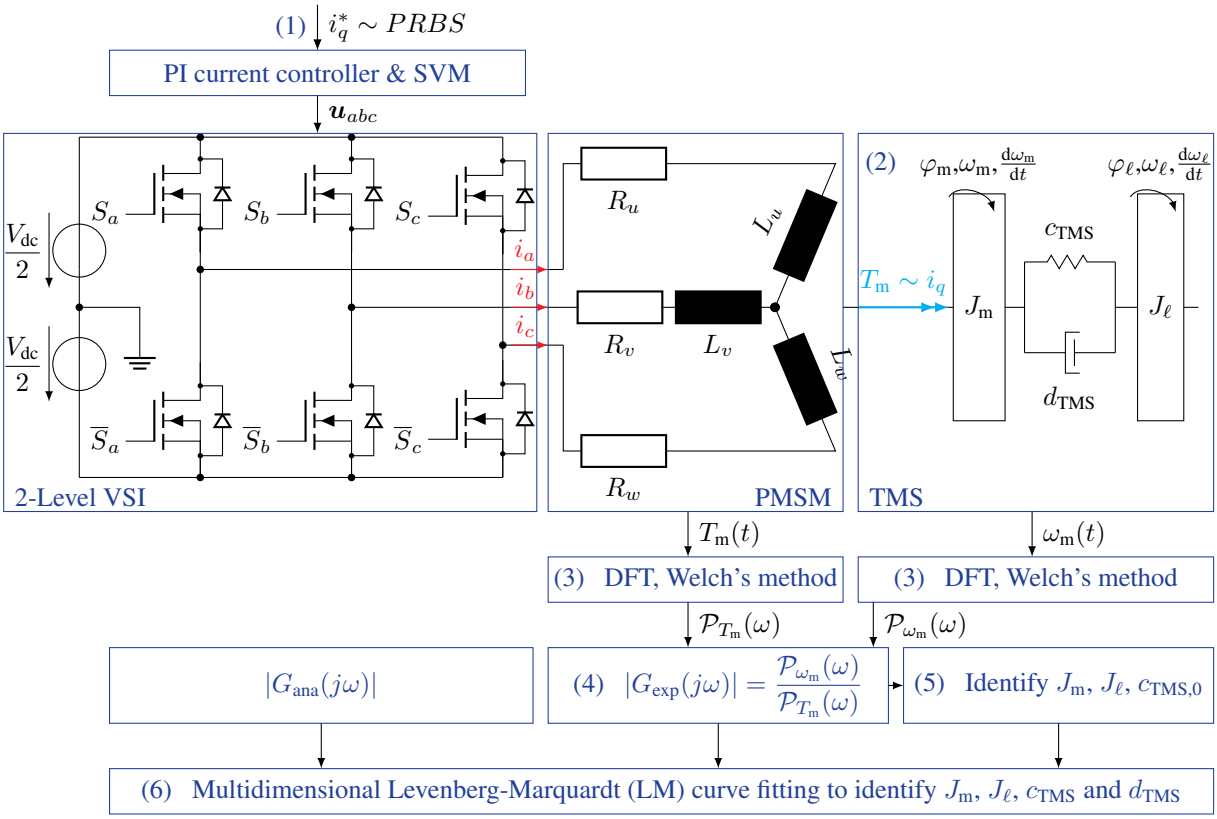


Figure 4.6: Visualization of the signal flow.

excite the resonance frequency of the drive train by applying a noise signal to the reference signal of the motor torque. Second, the phase currents and speed are measured, whereby the currents are used to calculate the torque, based on the linear model (2.28). Third, the spectral power densities of both signals, i.e., ω_m and T_m, are determined. Fourth, the quotient of the spectral power densities is used to generate the amplitude response of the transfer function. Fifth, based on the assumption of an undamped system, the motor and load side inertias are determined analytically. Finally, the stiffness and damping of the damped system can be determined numerically by using the LM. The exact procedure is described in the following and can be visualized using Fig. 4.6.

4.2.2.1 System Excitation

In order to be able to reconstruct the transfer function from the measured signals in the later steps, the drive train must be excited in the entire frequency range of interest, i.e., at least up to the resonance frequency. For this purpose, [Zou17, Vil07, Bei00] use noise signals, which can be generated with the help of a pseudorandom binary sequence (PRBS)[11]. These have an

[11] In the following the MATLAB *PNSequence* function is used in simulation and for code generation.

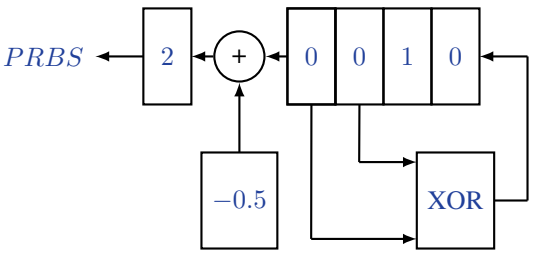
almost uniform spectral power density within a parameterizable frequency range and thus ensure broadband excitation, where especially the latter is important for the proposed strategy. The drive train can be excited in the open loop of the speed controller by feeding the PRBS into the reference signal of the torque-generating current component, see Fig. 4.6.

The excitation signal is generated on the basis of the last bit of a shift register with n_{PRBS} stages. As shown in Fig. 4.7(a), using the example of a four-stage register, each stage has a bit that is randomly assigned to the register during initialization. After a defined time interval, i.e., the hold time $T_{\text{S, PRBS}}$, each bit is shifted one stage to the left. The resulting free stage, at the right end of the register, is filled by an exclusive OR (XOR). The choice of which bits are fed back to the gate is of crucial importance for the period of the PRBS. Only if the whole period time $T_{\text{P,PRBS}}$ is reached, the broadband excitation of all frequencies is guaranteed. Since the shift register outputs a binary sequence, -0.5 must be added to the output and the modified sequence should be multiplied by 2 so that $PRBS \in \{-1, 1\}$. The resulting output signal of the shift register is shown schematically over time in Fig. 4.7(b). Moreover, a sufficient frequency range must be ensured during excitation. To do this, the hold time $T_{\text{S, PRBS}}$ between the switching operations of the shift register must be selected depending on the largest frequency of interest $f_{\text{ID, max}}$, i.e.,

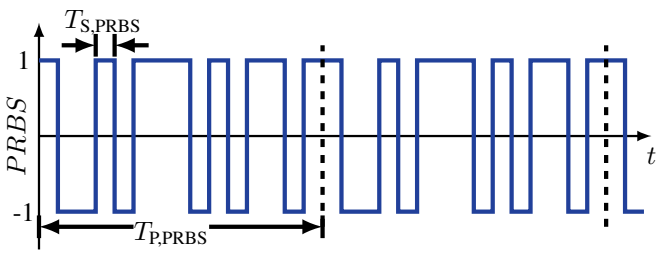
$$T_{\text{S, PRBS}} <= \frac{1}{2 f_{\text{ID, max}}} \,. \tag{4.22}$$

Otherwise, the spectral power density of the torque excitation can become so small—at a multiple of the PRBS clock frequency $f_{\text{S, PRBS}} = 1/T_{\text{S, PRBS}}$—that no sufficient excitation at this frequency takes place. The factor two in (4.22) results from the fact that $T_{\text{S, PRBS}}$ represents only a half-period of the highest frequency oscillation.

For the subsequent identification steps, it is necessary to be able to correctly transform the time signals into the frequency range. In order to record all excited vibrations with their full-period, the maximum period of the excitation sequence must be determined. To do this, the maximum number of switching operations of the register N_{PRBS} must be determined. This depends on the



(a) Schematic PRBS generator with shift register.



(b) Time response of the schematic PRBS.

Figure 4.7: PRBS generation based on a shift register exemplified for n_{PRBS}=4 steps.

number of shift register levels n_{PRBS}, i.e.,

$$N_{\text{PRBS}} = 2^{n_{\text{PRBS}}} - 1\,. \tag{4.23}$$

A detailed parameter study on the choice of n_{PRBS} and $T_{\text{S, PRBS}}$ is shown, e.g., in [Vil07, p. 117-119]. Note that the shift register bits should be increased iteratively from small values, because the higher their number, the longer the excitation time and the finer the frequency response. It is particularly important to note that the resolution of the frequency response may reflect the overshoot at the extremum of the damper and resonant frequency. Since the frequency response has a high gradient before and after these extrema, a high level of granularity is necessary to map the correct function value at these frequencies. A resolution that is too low could otherwise lead to a damping value that may be identified as being too big, because the true extreme values lie between the measured values. However, since every increase in the number of shift register stages doubles the recording time and thus enormously increases the computing effort for the spectral analysis, n_{PRBS} should only be chosen as big as necessary. Finally,

$$T_{\text{P,PRBS}} = T_{\text{S,PRBS}} N_{\text{PRBS}} \tag{4.24}$$

shows that the previously determined hold time $T_{\text{S, PRBS}}$ can be used to determine the maximum period of the excitation sequence, i.e., $T_{\text{P,PRBS}}$. In the following, e.g., $n_{\text{PRBS}} = 11$ and $f_{\text{ID, max}} = 500\,\text{Hz}$ are chosen, which results $N_{\text{PRBS}} = 2047$ and $T_{\text{P,PRBS}} = 2.047\,\text{s}$.
To ensure sufficient excitation, the amplitude of the excitation signal must be scaled depending on the drive, i.e., by using the nominal torque or nominal current. This is exemplified in Fig. 4.6 with i_q^*, where $i_q^* = \lambda_{\text{PRBS}} I_{\text{N}} PRBS$. Here, the scaling factor λ_{PRBS} is increased iteratively—starting from zero—until the damper and resonance frequency can be identified sufficiently well, see Section 4.2.2.3.

4.2.2.2 Preparation of Measurements

In order to be able to reconstruct the amplitude response from the recorded measurements of speed and torque, the spectral power density of both signals must be estimated. As pointed out in [Vil07, p. 33], the spectral analysis of the results of the stochastic excitation signal can be affected by the *leakage effect*, which can lead to deviations in the spectral power densities. This disruptive effect occurs when the recording duration does not correspond to an integer multiple of the period of the harmonic vibrations to be analyzed. Villwock therefore suggests to determine the spectral power densities with *Welch's method* [Vil07, p. 37 ff.], which reduces the influence of the *leakage effect* by dividing the time signals into individual segments, so-called

windows.[12] Afterwards, the measured values of each signal window are compared and evaluated using a window function. In [Vil07, p. 36 ff.], three different window functions based on cosine functions are examined. Even though all results are quite similar, the *Hanning* window function is named as the preferred window function and is hence used in the following. The signal sections evaluated by multiplication with the window function are converted into the frequency range by Fourier transformation. Afterwards, the modified periodograms of the individual signal sections, resulting from the Fourier transformation, are added up and divided by the number of signal sections in order to obtain the spectral power densities from the mean values. For generating the measured amplitude response, the ratio of the magnitude of the spectral power densities of motor speed and motor torque is finally formed.

4.2.2.3 Estimation of Parameters

A comparison of the amplitude response determined from the measurements with the amount of the analytically determined transfer function from Section 2.2.2, allows the determination of the mechanical parameters of interest. First,

$$(J_\mathrm{m} + J_\ell) = \frac{1}{\lim\limits_{\omega\to 0}(\omega|G_\mathrm{exp}(j\omega)|)} \tag{4.25}$$

shows how the total inertia of the drive train can be determined based on the asymptote described in (2.39) with measurements at low excitation frequencies.[13] Furthermore, by comparing analytical and measured extreme amplitude response values, the resonance and the damper frequency of the drive train (DT) can be determined, see Fig. 4.8. The measured signals of the transfer function, i.e., speed and torque are shown in Fig. A.12 of Appendix A.4.3. In the measured amplitude response, the resonance frequency ω_res can be recognized as a local maximum and the damper frequency ω_dam as a local minimum. If, first of all, an undamped system, i.e., $d_\mathrm{TMS} = 0$, with the stiffness $c_\mathrm{TMS,0}$ is assumed, the extremes mentioned can be regarded as zero points of the frequency response numerator and denominator. They can, as given by

$$c_\mathrm{TMS,0} - J_\ell\omega_\mathrm{dam}^2 = 0 \tag{4.26}$$

$$c_\mathrm{TMS,0}\omega_\mathrm{res}\left(J_\mathrm{m} + J_\ell\right) - J_\mathrm{m}J_\ell\omega_\mathrm{res}^3 = 0\,, \tag{4.27}$$

[12] The spectral analysis with *Welch's* method is offered within MATLAB/Simulink as part of the *Spectrum Estimator*. For the example with $f_\mathrm{ID,\,max} = 500\,\mathrm{Hz}$, $f_\mathrm{S,\,PRBS} = 1\mathrm{kHz}$ is chosen as sample rate in the estimator.

[13] This is also the simplest solution for J, i.e., assuming a one-mass system. This is the case when the DT cannot be identified as a TMS since the DT is too stiff and thus the resonance frequency cannot be excited during the ID.

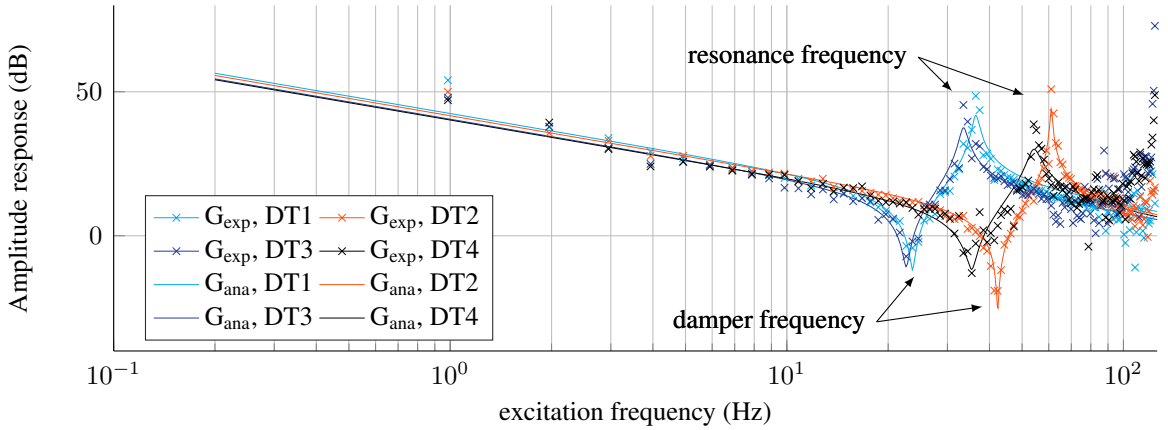


Figure 4.8: Experimentally identified and analytically adapted transfer function.

be determined, where the denominator of (2.38) and the numerator of (2.38) are equated with the value zero. Overall, a system based on the three equations (4.25) - (4.27), including three unknowns J_m, J_ℓ and $c_{\mathrm{TMS},0}$, is given. Hence, a unique analytical solution for the undamped system, i.e., the two moments of inertia and the stiffness of the undamped system, can be determined. The latter can be used as an initial value to determine the stiffness and damping of the damped system.

The parameters of the damped system require a numerical optimization that minimizes the sum of the squared errors between the amount of the analytically determined transfer function (2.38) and the measured transfer function. In doing so, the parameters of interest, i.e., $\hat{c}_\mathrm{TMS}$ and $\hat{d}_\mathrm{TMS}$, are iteratively adjusted until the two amplitude responses match while considering several termination criteria. For this purpose, a cost function including the discrete samples κ, i.e.,

$$J_\mathrm{LM}(k) = ||\boldsymbol{\xi}(k)||_2^2 = \sum_{\ell=k}^{k+\kappa-1} \left(|G_\mathrm{exp}(j\omega_\ell)| - |G_\mathrm{ana}(j\omega_\ell, \hat{\boldsymbol{\Theta}}_\mathrm{LM})|\right)^2, \tag{4.28}$$

where

$$\hat{\boldsymbol{\Theta}}_\mathrm{LM}(k) = \begin{bmatrix} \hat{c}_\mathrm{TMS} \\ \hat{d}_\mathrm{TMS} \end{bmatrix} \tag{4.29}$$

is minimized across all examined discrete excitation frequencies.[14] Considering the dependency of the sum of squared errors on the relative deviations of the estimated parameters, i.e., $\Delta c_{\mathrm{TMS,rel}} = \dfrac{\hat{c}_\mathrm{TMS} - c_{\mathrm{TMS,ideal}}}{c_{\mathrm{TMS,ideal}}}$ and $\Delta d_{\mathrm{TMS,rel}} = \dfrac{\hat{d}_\mathrm{TMS} - d_{\mathrm{TMS,ideal}}}{d_{\mathrm{TMS,ideal}}}$, where $c_{\mathrm{TMS,ideal}}$ and $d_{\mathrm{TMS,ideal}}$ are the respective ideal parameters, it is noticeable that their course show a strong nonlinearity, as shown in Fig. 4.9. Near the solution parameters, the sum of squared errors has a high gradient. However, this gradient becomes very small with high estimation errors and can even

[14] $\omega_{\mathrm{ID,\,min}} = 2\pi f_{\mathrm{ID,\,min}}$ and $\omega_{\mathrm{ID,\,max}} = 2\pi f_{\mathrm{ID,\,max}}$ are the limits of the analyzed frequency range. Since f_dam and f_res were typically in the range of $20{-}500\,\mathrm{Hz}$ for the studied DTs, and to avoid outliers at the frequency range boarders, the analyzed frequencies can be limited for, e.g., $0{-}512\,\mathrm{Hz}$ to $f_{\mathrm{ID,\,min}} = 5\,\mathrm{Hz}$ and $f_{\mathrm{ID,\,max}} = 500\,\mathrm{Hz}$, where the discrete step size is $1\,\mathrm{Hz}$, i.e., $\kappa = 495$.

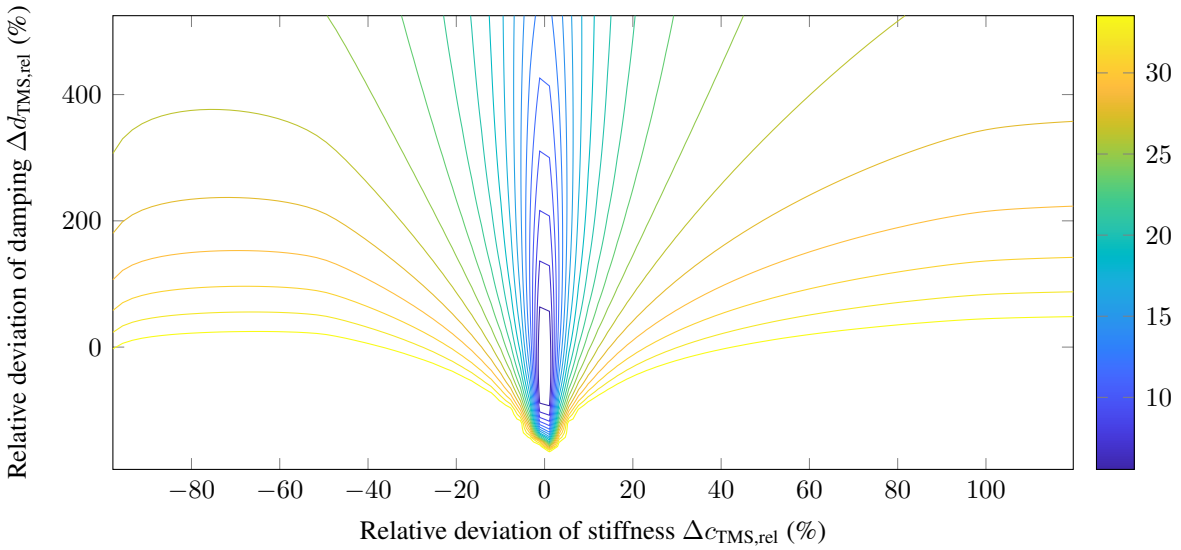


Figure 4.9: Dependency of the sum of squared errors, i.e., J_{LM}, on the relative parameter deviations $\Delta c_{\text{TMS,rel}}$ and $\Delta d_{\text{TMS,rel}}$.

point in the opposite direction of the solution parameters, see, e.g., left-hand side in Fig. 4.9, where $\hat{c}_{\text{TMS}}$ is initially too small. This nonlinearity can be circumvented, at least to some extent, if the stiffness from the undamped system, i.e., $c_{\text{TMS,0}}$, is used as starting value for $\hat{c}_{\text{TMS}}$ and the minimization thus begins near the solution parameters. In doing so, J_{LM} is minimized using the LM algorithm, similar as described in Section 4.1.1. Here, the Jacobian matrix is defined by

$$\boldsymbol{J}_{\text{J,c}} = \left[\frac{\partial \xi}{\partial \hat{c}_{\text{TMS}}} \qquad \frac{\partial \xi}{\partial \hat{d}_{\text{TMS}}} \right] . \tag{4.30}$$

Based on an empirical evaluation according to [Vil07, p. 50], the scaling factor is initialized with $\lambda_{\text{LM}} = 0.001$. Consequently, the new parameter estimate is given by (4.7), see Section 4.1.1. In contrast to Section 4.1.1, with respect to [Vil07, p. 52], λ_{LM} in (4.8) is decreased by using $\theta = 2$ and λ_{LM} in (4.9) is increased by using $\theta = 10$. Furthermore, in the former case $\hat{\boldsymbol{\Theta}}_{\text{LM}}(k+1)$ and in the latter case $\hat{\boldsymbol{\Theta}}_{\text{LM}}(k)$ is used for the subsequent iteration step of the LM algorithm.
Note that in each iteration step three termination criteria are checked, which can terminate the minimization algorithm (4.28). First, the minimization is stopped when a limit for the sum of squared errors is reached, i.e., 500. Second, it is terminated when reaching a maximum number of iterations, i.e., 50. Third, it is stopped when falling below a minimum in the correction vector $\Delta\hat{\boldsymbol{\Theta}}_{\text{LM}}$, i.e., a correction step size of $\Delta\hat{c}_{\text{TMS}} = 10^{-4}$ Nm/rad for the stiffness and $\Delta\hat{d}_{\text{TMS}} = 10^{-7}$ Nms/rad for the damping. The latter termination criterion depends on the size of the used data type and is intended to prevent a variable overflow. All three values of the termination criteria are parameterized with iteratively determined empirical values.
The method has been experimentally verified for several drive trains, see Fig. 4.8, where the identified parameters for the test bench in Fig. 3.4 are given in Table A.4 of Appendix A.3.

4.3 Online Parameter Tracking

As described in Section 2.2.1.2, the linear machine model is only applicable in a certain operating range and is not valid for a nonlinear magnetic circuit. Furthermore, parameter drifts can occur, e.g., due to temperature changes. Table 4.1 summarizes the significantly affected parameters. Therefore, an online ID is applied that adapts the relevant parameters to provide robust MPC control, see Fig. 4.10. The chosen online ID methods belong to the family of steady-state methods, see Fig. 4.1. Thus, the parameters can only be updated in steady state. If steady-state operating points cannot be guaranteed, other methods must be used, see Fig. 4.1. The following results include outcomes from [Geb20, Huf21].

The online ID is an independent state machine and is divided into two basic parts. First, a low-computation part, but time-critical and hence requires hard real-time execution. Thus, it is executed in an ISR as described in Section 3.1. Second, a computationally intensive part, but which can be computed in a standard routine that is not time-critical, i.e., the main. The online ID runs in parallel, i.e., in the background, next to the core control task. Thus, it can be combined with any control algorithm, e.g., MPC or FOC, as long as sufficient steady-state operating points can be guaranteed at certain time intervals (otherwise the parameters are not updated). From the control algorithm point of view, the online ID can be seen as an optional extension to optimize the parameters used during operation.[15] As shown in Fig. 4.3, the initial

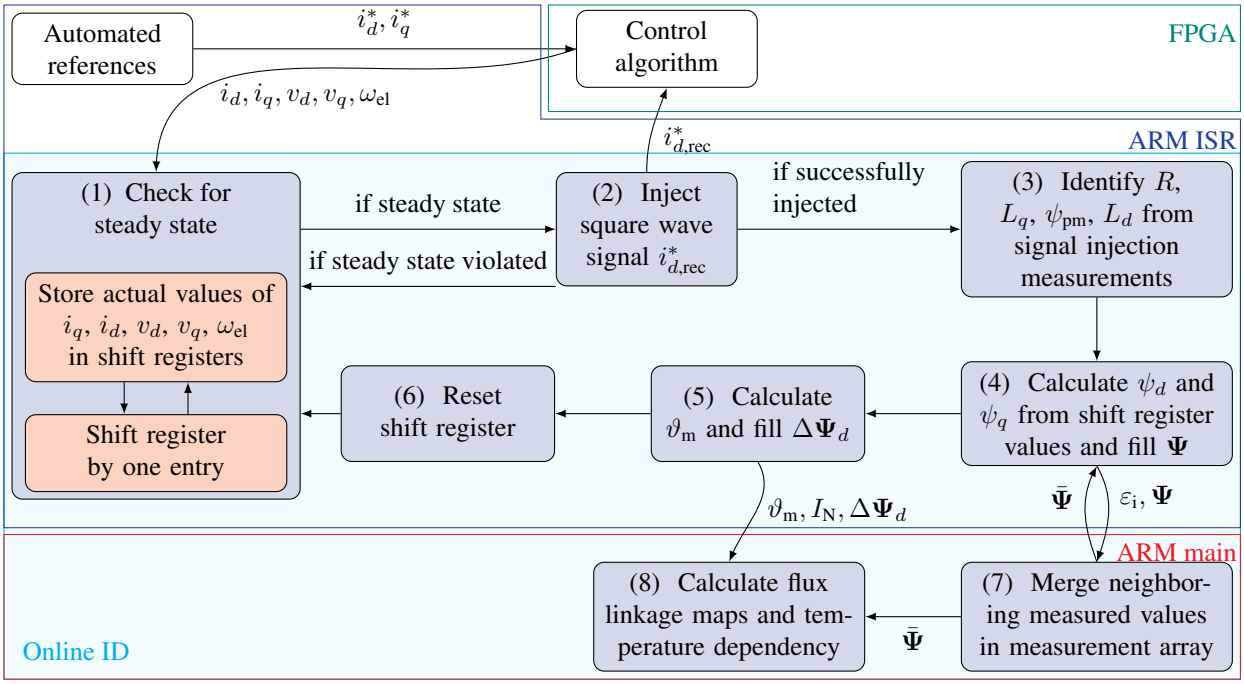


Figure 4.10: Schematic representation of the state machine of the online identification process.

[15] Alternatively, by activating an optional automatic reference value generator, the control approach can be forced to reach any operating point under constant measuring conditions by using reference sequences.

parameters of the online ID can be provided manually or determined from the previous offline ID. The online ID consists of eight functions that interact depending on the state, see Fig. 4.10. On the one hand, there are six time-critical functions that must be executed in hard real time and—depending on the state—require $14 - 68\,\mu\text{s}$ to be executed.

1. First, all necessary measured values are stored in shift registers.
2. If steady state is detected, a square wave signal $i^*_{d,\text{rec}}$ with a specified period duration and amplitude, see Section 4.3.1, is provided to the control algorithm. The controller then decides whether the signal should be injected at the current operating point or not. For a constant torque, a corresponding compensation is performed in the q-axis. During signal injection, the shift register for steady-state detection remains in its initial position. Thus, the recorded system states are stored in a separate buffer, i.e., RAM area, during the positive and negative quadratic half-period. If the steady-state condition is violated by external transients during injection, the measurements are discarded.
3. Using the buffer, R_ph, L_d, L_q and ψ_pm are identified, as explained in Section 4.3.1. The determined parameters are filtered by a moving averaging with past values, e.g., in the following three, so that outliers and erroneous measurements do not have much influence.
4. Using R_ph, the flux linkage components ψ_d and ψ_q are calculated and stored in an array, i.e., $\mathbf{\Psi}$, coupled with the prevailing currents and temperature, see Section 4.3.4.1.
5. The winding temperature ϑ is calculated using the last resistance value identified online and a reference resistance, see Section 4.3.2. Assuming that ϑ_m is close to ϑ, a temperature dependency of the flux linkage, i.e., α_ψ is determined by means of a data field for the flux linkage difference over temperature, i.e., $\Delta\mathbf{\Psi}_d$, see Section 4.3.3.
6. Before the state machine returns to its initial state, the current shift register is cleared so that the re-evaluation of the steady state is based exclusively on new measured values.

On the other hand, the time-uncritical part of the online ID includes two functions.

7. A function merges neighboring measured values of the measurement array $\mathbf{\Psi}$, since it is filled irregularly by the time-critical part. Thereby, all measured values which do not maintain a specified minimum distance ε for the d, q-current combinations are combined to avoid unnecessary use of RAM. This results in $\bar{\mathbf{\Psi}}$, see Section 4.3.4.1.
8. Finally, the nominal current I_N, the current winding temperature ϑ—which is assumed to be close to the magnet temperature ϑ_m—with the resulting flux linkage difference $\Delta\mathbf{\Psi}_d$ and the scattered measurements in $\bar{\mathbf{\Psi}}$ are used to calculate the currently valid—in case of ψ_d temperature compensated—flux linkage, see Section 4.3.4.2. The resulting characterization of ψ_d and ψ_q is required for the proposed flux linkage-based MPC approach in Chapter 7.

4.3.1 Linear Model Parameters

The online identification of the linear model parameters is done by injecting a square wave test signal $i^*_{d,\mathrm{rec}}$ into the d-axis, resulting in a current i_d, as shown in Fig. 4.11 similar to [Kel12, p. 153].

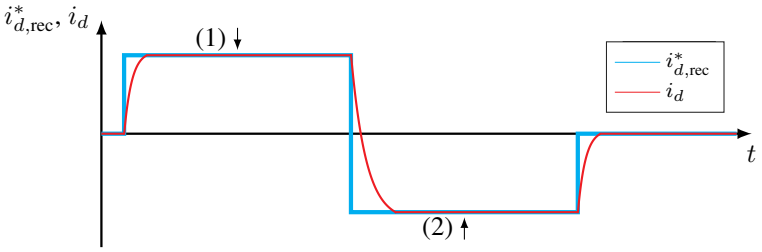


Figure 4.11: Schematic representation of the injected test signal.

The amplitude of the square wave signal is chosen to be small (e.g., 10 %) compared to the nominal current in order to generate only a small reluctance torque in PMSMs with pronounced magnetic saliency. Even if the magnetic saliency is small, the torque is compensated for by a corresponding q-axis current component. The frequency of the square wave signal, i.e., f_{rec}, is selected by

$$4T_{\mathrm{el}} < \frac{1}{f_{\mathrm{rec}}} << \tau_\vartheta \,, \tag{4.31}$$

where τ_ϑ is the thermal time constant of the motor using a duty cycle of $50\,\%$.[16] The frequency is a compromise between reaching steady state after the current change and avoiding a significant temperature change of the machine during a measurement interval. By adding the rectangular amplitude to the reference value of the d-axis current in the first state (1) and then subtracting it in state (2), the voltage change of the d-axis voltage required for the current change can be used to determine the phase resistance. This is based on a simplification of the d-axis voltage component in (2.27), where in steady state $\frac{\mathrm{d}i_d}{\mathrm{d}t} \approx 0$ holds, thus

$$v_d^{(1)} = R_{\mathrm{ph}} i_d^{(1)} - \omega_{\mathrm{el}}^{(1)} i_q^{(1)} L_q \quad \text{and} \quad v_d^{(2)} = R_{\mathrm{ph}} i_d^{(2)} - \omega_{\mathrm{el}}^{(2)} i_q^{(2)} L_q \,. \tag{4.32}$$

By measuring φ, ω_{el} and the phase values $\boldsymbol{i}_{abc}$, $\boldsymbol{v}_{abc}$, all quantities except R_{ph} and L_q can be determined by means of the transformations described in Section 2.1.1. To infer the resistance, (4.32) is solved for each of the two states (1) and (2) according to the inductance of the q-axis, i.e.,

$$L_q = \frac{R_{\mathrm{ph}} i_d^{(1)} - v_d^{(1)}}{\omega_{\mathrm{el}}^{(1)} i_q^{(1)}} \quad \text{and} \quad L_q = \frac{R_{\mathrm{ph}} i_d^{(2)} - v_d^{(2)}}{\omega_{\mathrm{el}}^{(2)} i_q^{(2)}} \,. \tag{4.33}$$

[16] For the considered drives $f_{\mathrm{rec}} = 1.0\,\mathrm{Hz}$ is chosen.

It is assumed that the q-axis inductance remains approximately constant due to the low rectangular amplitude and due to only slightly pronounced cross-coupling effects.[17] Thus, by equating the two q-axis inductances in (4.33) and solving for R_{ph}, the estimated phase resistance $\hat{R}_{\text{ph}}$ can be determined by[18]

$$\hat{R}_{\text{ph}} = \frac{v_d^{(2)}\omega_{\text{el}}^{(1)}i_q^{(1)} - v_d^{(1)}\omega_{\text{el}}^{(2)}i_q^{(2)}}{i_d^{(2)}\omega_{\text{el}}^{(1)}i_q^{(1)} - i_d^{(1)}\omega_{\text{el}}^{(2)}i_q^{(2)}} \,. \tag{4.34}$$

The measurement states recorded during square wave signal injection are also used to identify ψ_{pm} and the inductances in d- and q-axis. By substituting the previously determined resistance $\hat{R}_{\text{ph}}$ into (4.32), the q-axis inductance is first determined. To ensure low error influence, the two resulting $\hat{L}_q$ values of the two half square wave periods, i.e., (4.33), are averaged, which results in

$$\hat{L}_q = \frac{1}{2}\left(\frac{\hat{R}_{\text{ph}}i_d^{(1)} - v_d^{(1)}}{\omega_{\text{el}}^{(1)}i_q^{(1)}} + \frac{\hat{R}_{\text{ph}}i_d^{(2)} - v_d^{(2)}}{\omega_{\text{el}}^{(2)}i_q^{(2)}}\right) . \tag{4.35}$$

If not only the d-axis voltage component but also the q-axis voltage component is recorded during the injection of the d-axis rectangular current, ψ_{pm} and L_d can be determined. Thus, using (2.27) where in steady state $\frac{\mathrm{d}i_q}{\mathrm{d}t} \approx 0$ holds, the q-axis voltage of both half-periods can be described by

$$v_q^{(1)} = \hat{R}_{\text{ph}}i_q^{(1)} + \omega_{\text{el}}^{(1)}i_d^{(1)}L_d + \omega_{\text{el}}^{(1)}\psi_{\text{pm}} \quad \text{and} \quad v_q^{(2)} = \hat{R}_{\text{ph}}i_q^{(2)} + \omega_{\text{el}}^{(2)}i_d^{(2)}L_d + \omega_{\text{el}}^{(2)}\psi_{\text{pm}} \,. \tag{4.36}$$

Due to the mathematical dependencies, ψ_{pm} is identified first. Using the already identified $\hat{R}_{\text{ph}}$ and (4.36), the latter can be solved for ψ_{pm}. For this purpose, the same L_d is assumed in both half-periods, so that (4.36) can be simply rearranged by equating the L_d of both terms, i.e.,

$$\hat{\psi}_{\text{pm}} = \frac{(v_q^{(1)} - \hat{R}_{\text{ph}}i_q^{(1)})i_d^{(2)}}{(i_d^{(2)} - i_d^{(1)})\omega_{\text{el}}^{(1)}} - \frac{(v_q^{(2)} - \hat{R}_{\text{ph}}i_q^{(2)})i_d^{(1)}}{(i_d^{(2)} - i_d^{(1)})\omega_{\text{el}}^{(2)}} \,. \tag{4.37}$$

Finally, from (4.36), state (1) is subtracted from state (2) and solved for L_d using $\hat{\psi}_{\text{pm}}$, i.e.,

$$\hat{L}_d = \frac{(v_q^{(2)} - v_q^{(1)}) - \hat{R}_{\text{ph}}(i_q^{(2)} - i_q^{(1)}) - \hat{\psi}_{\text{pm}}(\omega_{\text{el}}^{(2)} - \omega_{\text{el}}^{(1)})}{\omega_{\text{el}}^{(2)}i_d^{(2)} - \omega_{\text{el}}^{(1)}i_d^{(1)}} \,. \tag{4.38}$$

[17] To ensure that the online parameter tracking only takes place if a linear magnetic circuit can be assumed and thus the absolute inductances are valid, a threshold value below half the nominal current is set. Moreover, a second limit is implemented that allows the online resistance tracking after a specified time (e.g., three minutes), even if the linear validity range of the model does not hold. In this case, a small error in resistance identification is accepted, while L_d, L_q and ψ_{pm} are not tracked. Both limits, i.e., current and time, can be adjusted.

[18] By equating, possible sensor or measurement offsets cancel each other out.

4.3.2 Winding Temperature

The actual winding temperature is estimated based on the online identified $\hat{R}_{\text{ph}}$, a reference value $R_{\text{ph,ref}}$ and an offline identified relative temperature dependency coefficient α_{R},[19] by using

$$\hat{\vartheta}(k) = \frac{1}{\alpha_{\text{R}}} \frac{\hat{R}_{\text{ph}}(k) - R_{\text{ph,ref}}}{R_{\text{ph,ref}}} + \vartheta_{\text{ref}} = \Delta\vartheta(k) + \vartheta_{\text{ref}} \,. \tag{4.39}$$

The reference value $R_{\text{ph,ref}}$ can be determined using the offline ID in Section 4.1.1, i.e., $R_{\text{ph,ref}} = \hat{R}_{\text{ph,ref}}$. Where the latter must be measured at a known reference temperature ϑ_{ref} for the motor winding, i.e., at ambient temperature before operation or after sufficient rest periods.

4.3.3 Temperature Dependency of the Flux Linkage due to the Permanent Magnets

To properly account for the effect of temperature on the flux linkage described in Section 2.2.1.2, i.e., a temperature drift $\Delta\vartheta_{\text{m}}$ on ψ_{pm}, two steps are necessary.

First, similar to the resistance, a temperature-dependent coefficient for the flux linkage, i.e., α_ψ, is necessary. α_ψ is determined by comparing the values of $\hat{\psi}_{\text{pm}}$ over several measurements at different temperatures using the online ID—see Section 4.3.1—with an offline identified reference flux linkage $\hat{\psi}_{\text{pm,ref}}$, see Section 4.1.2.
More precisely, comparable to R_{ph}, the offline identification of ψ_{pm} also allows the prevailing magnet temperature $\vartheta_{\text{m,ref}}$ to be approximated by the ambient temperature, whereby a sufficient rest period is assumed. If the offline identified $\hat{\psi}_{\text{pm,ref}}$ is obtained at the same temperature at which $\hat{R}_{\text{ph,ref}}$ is determined, it can be assumed that $\vartheta_{\text{m,ref}} = \vartheta_{\text{ref}}$. Furthermore, due to the low mass and compact design of small electric drives, a good thermal conductivity is adopted due to the low thermal resistance, which—in thermal steady state—allows to use the online tracked winding temperature for the permanent magnets, i.e., $\vartheta_{\text{m}} \approx \vartheta$.[20] As soon as a temperature change is detected (see Section 4.3.2), a $\hat{\psi}_{\text{pm}}(k)$ measurement with the corresponding $\hat{\vartheta}(k)$ is used for calculating the present deviation with $\Delta\psi_{\text{pm,temp}}(k) = \hat{\psi}_{\text{pm}}(k) - \hat{\psi}_{\text{pm,ref}}$ and $\Delta\vartheta(k) = \hat{\vartheta}(k) - \vartheta_{\text{ref}}$. Then the differences are iteratively stored in an array $\Delta\boldsymbol{\Psi}_d$ (see Fig. 4.10) of the RAM. Once a sufficient number of data pairs, i.e., m, is stored in $\Delta\boldsymbol{\Psi}_d$, thus averaged over

[19] Either the temperature dependency coefficient of the winding material, e.g., copper ($\alpha_{\text{R}} = 0.00393\,1/\text{K}$) can be used, or, as it is done in this work, α_{R} is determined offline by measuring the resistance at two defined temperatures.

[20] The assumption applies to the considered small drives due to their low thermal resistance and was experimentally verified with acceptable accuracy in terms of measurements. If this is not valid for a drive to be used, an independent estimation of the rotor temperature is necessary, see, e.g., [WHPB14, RFT+16, KWB21].

several measurements where outliers are rejected, α_ψ can be determined by the slope of a linear regression line, using

$$\alpha_\psi = \frac{\sum_{\ell=k}^{k+m-1}(\Delta\vartheta(\ell) - \Delta\bar{\vartheta})(\Delta\psi_{\text{pm,temp}}(\ell) - \Delta\bar{\psi}_{\text{pm,temp}})}{\sum_{\ell=k}^{k+m-1}(\Delta\vartheta(\ell) - \Delta\bar{\vartheta})^2} \tag{4.40}$$

with $\Delta\bar{\psi}_{\text{pm}}$ and $\Delta\bar{\vartheta}$ being the arithmetic mean. At the same time, the deviation between the offline $\hat{\psi}_{\text{pm,ref}}$ and the online identified $\hat{\psi}_{\text{pm}}$ can be determined by the y-axis intercept of the regression line, since the regression line represents a half-line starting from the origin in case of an ideal match. With acceptable accuracy in terms of measurements, a linear slope fits the nonlinear temperature increase sufficiently well, see, e.g., Fig. 2.11 or [Spe14, p. 19].
Second, based on α_ψ and $\hat{\vartheta}$ (see Section 4.3.2), the offset $\Delta\psi_{\text{pm}}$ is calculated during operation using

$$\Delta\psi_{\text{pm}} = \alpha_\psi \Delta\vartheta\,. \tag{4.41}$$

Since the slope α_ψ is used to determine $\Delta\psi_{\text{pm}}$ instead of using a single $\Delta\psi_{\text{pm,temp}}$ measurement, the method is more robust to outliers and measurement inaccuracies as long as in (4.37) almost the same L_d can be assumed in both half-periods. This adaptation of the permanent magnet flux linkage increases the modeling accuracy and is required for Section 4.3.4.1.

4.3.4 Online Identification of Flux Linkage Maps

The online determination of the absolute inductances—see Section 4.3.1—is only useful if the machine under investigation can be represented by the linear machine model. For nonlinear machines, instead of L_d and L_q, the flux linkage or differential inductances must be determined. This determination is based on the voltage (2.20), as described in Section 2.2.1.2. For reasons which are explained in Chapter 7, the flux linkage is preferred to differential inductances.

The flux linkage can be described in two ways. First, analytically, by identifying a (high-order) polynomial function. With such an approach self-saturation can be easily described, while bivariate polynomials can be used for the modeling of cross-saturation effects [KLK12]. However, it is not trivial which polynomial order sufficiently describes the nonlinearity; a heuristic choice is to use a 3^{rd} to 5^{th}-order polynomial [KLK12], which may be computationally intractable in real time. Because of this as well as for implementation reasons (see Section 3), a second method is preferred which uses flux linkage maps, also named flux linkage characteristic diagrams. The identified three-dimensional flux linkage maps represent the value of the flux linkage components ψ_d and ψ_q as a function of the motor current components i_d and i_q. In this way the maps provide information about linearity deviations including self- and cross-saturation.

There are a variety of approaches for online identification of the flux linkage maps, e.g., [KK13], [UH09]. However, the method proposed in this thesis uses an online flux linkage map ID, which enables the adaptation of the maps in a time-uncritical task during operation. The advantage of this method is its ease of implementation; it allows a simple and fast start-up by first using the linear parameters, i.e., the absolute inductances, while nonlinear effects are taken into account later in the process. Here, temperature effects are also taken into account, as explained in Sections 4.3.1 and 4.3.3 for the resistance and the magnets, respectively, although the temperature caused only a maximum flux change of about $7.3\,\%$ at the maximum operating point for the magnets of the examined PMSMs, see, e.g., Fig. 2.11.[21]
However, the presented online ID of the flux linkage maps has the following restrictions:

- The approach only works in steady state. Therefore, if no steady state is achieved occasionally, no online adjustment of flux linkage maps can be performed.
- The rotor position dependencies of the flux linkage explained in Section 2.2.1.2, i.e., higher-order harmonics, are averaged and thus neglected, see the last term in (2.21). This is done to avoid a fourth dimension in the generated maps and thus to keep the memory and computational load low. This is almost always acceptable since manufacturing imperfections which can lead, e.g., to an unequal air gap—and thus an unequal air-gap field—are negligible within a small tolerance band [GT18].
- Speed ranges that are too small are not applicable, since the necessary voltage measurement would take place in a measuring range that is too small, i.e., too insufficient.[22]
- Iron loss effects, which can cause fault voltages and thus affect the flux linkage maps, are neglected here.[23] This is valid for the examined small PMSMs within the limits of measurement accuracy, since the influence at the maximum operating point causes only a maximum flux linkage change by a small percentage, see Table 4.4.[24]
- Since iron losses are not considered in the following, it is also not recommended to perform the identification at too high speeds, even if the impact is small.
- For all flux linkage and inductance maps a strict rule is important. The values must be monotonically nondecreasing. This is also required for the vectors in x- and y-axis, i.e., i_d and i_q. However, this is always a given for electric drives.

In summary, a speed range of 200 - $1200\,$rpm, i.e., $n_\mathrm{m} \in [5 - 30]\,\%$ of n_N, is recommended for the online ID in case of the examined motors in order to keep the accuracy of flux linkage maps as high as possible. A similar value of $1000\,$rpm is recommended, e.g., in [Ric16, p. 93].

[21] Similar, although lower, influences were found, e.g., with $6\,\%$ in [Ric16, p. 102].

[22] Monitoring is integrated to ensure that the process is aborted if the speed drops below a user defined minimum.

[23] In case no online ID is required and both generator and motor operation are possible, the flux linkage maps can be identified in both operating modes. Since the iron losses have opposite signs in the two operating modes, they can thus be mathematically truncated out (subtracted), see, e.g., [Kel12, RDD14, SP14, Ric16].

[24] Similar influences were found, e.g., with $4.2\,\%$ in [Ric16, p. 103].

4.3.4.1 Data Acquisition

The identification algorithm is initially based on the linear machine parameters R_{ph}, L_d, L_q and ψ_{pm}, which describe the machine behavior sufficiently well for operation at low currents. With these parameters—which can be predetermined either by using the offline ID process (see Section 4.1) or merely the data sheet values—the initial flux linkage $\boldsymbol{\psi}_{dq,\text{init}}$ is found by

$$\boldsymbol{\psi}_{dq,\text{init}} = \boldsymbol{L}_{dq}\boldsymbol{i}_{dq} + \begin{bmatrix} \psi_{\text{pm}} \\ 0 \end{bmatrix} . \tag{4.42}$$

Subsequently, during operation, the flux linkage maps are adapted step by step to cover the nonlinear behavior of the magnetic circuit at higher currents. Taking into account the restrictions mentioned above, the algorithm is able to calculate the present flux linkage after rearrangement of (2.20), i.e.,

$$\boldsymbol{\psi}_{dq} = \boldsymbol{P}^{-1}\frac{\boldsymbol{v}_{dq} - R_{\text{ph}}\boldsymbol{i}_{dq} - \dfrac{\mathrm{d}\boldsymbol{\psi}_{dq}}{\mathrm{d}t}}{\omega_{\text{el}}} , \tag{4.43}$$

by utilizing the voltage, current, and rotor position measurements and/or estimates. Neglecting the rotor angular dependency of the flux linkage by averaging the measurements over several complete rotor revolutions [Ric16, p. 25] and by assuming steady-state operation[25], $\frac{\mathrm{d}\boldsymbol{\psi}_{dq}}{\mathrm{d}t} \approx \boldsymbol{0}$ results. On this basis, (4.43) can be simplified to

$$\boldsymbol{\psi}_{dq} = \boldsymbol{P}^{-1}\frac{\boldsymbol{v}_{dq} - R_{\text{ph}}\boldsymbol{i}_{dq}}{\omega_{\text{el}}} . \tag{4.44}$$

Using (4.44), the flux linkage combinations can be calculated and are subsequently stored in the array $\boldsymbol{\Psi}$, i.e., the RAM (see Chapter 3), together with the actual current $\boldsymbol{i}_{dq}$, the winding temperature ϑ and speed ω_{el}.

Data processing Once $\boldsymbol{\Psi}$ has sufficient data points, the flux linkage maps can be reasonably adjusted. The number of these points should be large enough to sufficiently describe the aforementioned relationship, but also relatively small to avoid increased memory requirements. Hence, due to RAM limitations the identification algorithm must avoid storing congruent or similar measurements. Since the online ID can only use the randomly prevailing conditions during operation to gradually fill the maps, identical measurement points are obviously often measured several times. Therefore, the information of data pairs whose projected distance

[25] Detection of steady-state operation in real time is performed by specifying an acceptable degree of deviation (e.g., $\pm 0.4\%$) of the variables of concern, i.e., ω_{el}, $\boldsymbol{i}_{dq}$, from their nominal value.

ε does not exceed a minimum distance threshold ε_{min}[26], are merged by using averaging, i.e., $\bar{\boldsymbol{\Psi}}(x,y,z) = 0.9\bar{\boldsymbol{\Psi}}(x,y,z) + 0.1\boldsymbol{\Psi}(x,y,z)$, and are stored in $\bar{\boldsymbol{\Psi}}$. The projected distance ε (Euclidean distance) between two arbitrary points in an arbitrary three-dimensional system, with the arbitrary axes x, y and z, can be described by

$$\varepsilon_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}\,, \tag{4.45}$$

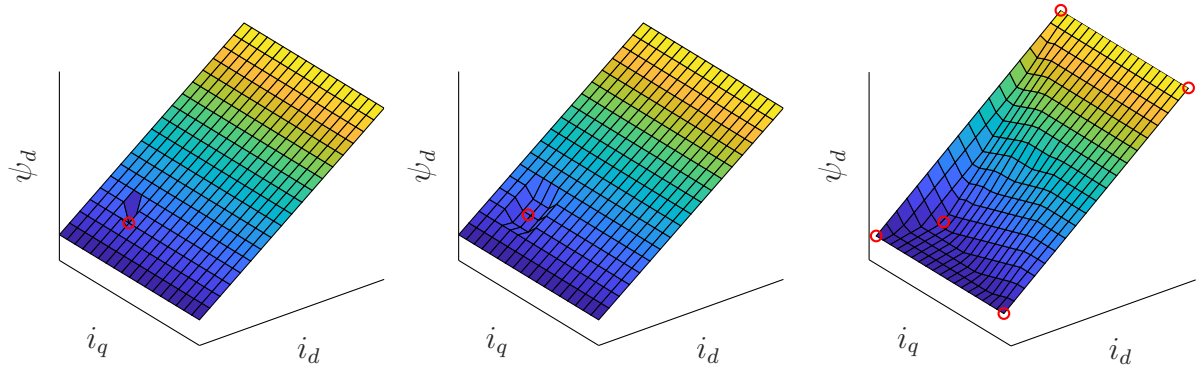
where x and y can represent i_d and i_q, respectively, while the z-axis can represent any of the flux linkage components (ψ_d or ψ_q). This ensures that, due to the limited RAM, enough independent measurement pairs are stored with a sufficient distance, i.e., ε_{i}, to each other to allow a reasonable adaptation of the maps. Furthermore, the combination of several measurements increases the significance of the map grid points, since outliers are averaged.

Data modification For a high identification accuracy, the time-varying and influencing parameters, i.e., R_{ph} and ψ_{pm}, are tracked online in regular intervals, see Sections 4.3.1 and 4.3.3 respectively. The influences here are mainly due to temperature changes. Since ψ_d depends mainly on ψ_{pm}, the flux linkage map on the d-axis is adapted for temperature drifts by using (4.41). Consequently, the ψ_d flux linkage map stored in the RAM is corrected according to the offset $\Delta\psi_{\text{pm}}$, i.e., whenever the temperature is updated, the ψ_d map is shifted along the z-axis. Finally, it should be mentioned that the offset is independent of the current combinations. This ensures that the currently valid flux linkage map is always up-to-date and related to the current winding temperature, assuming $\vartheta_{\text{m}} \approx \vartheta$.

4.3.4.2 Interpolation and Extrapolation

For the generation of the flux linkage maps, data points are required that describe the (nonlinear) relationship between the stator current and flux linkage over the whole operating range. To obtain fully adapted flux linkage maps, a $\boldsymbol{\psi}_{dq}$ value would have to be stored for each conceivable motor current combination of $\boldsymbol{i}_{dq}$. Thus, it would have to be assumed that each current combination of $\boldsymbol{i}_{dq}$ is reached at least once during operation and that consequently each initial value $\boldsymbol{\psi}_{dq,\text{init}}$ is replaced by an identified value $\boldsymbol{\psi}_{dq}$. However, the online ID can only use the randomly prevailing conditions during operation to gradually fill the maps. Since the PMSM is supposed to follow the reference values specified by the user, some operating points may never be reached. Therefore, a procedure has to be developed to adapt parts of the map as soon as a few measured values, e.g., four, are available.

[26] The distance depends on the array size $\bar{\boldsymbol{\Psi}}$ and thus the available RAM. Using the example of a 20x20 array, $\varepsilon_{\text{min}} = 0.125 I_{\text{N}}$ seems adequate to cover a measuring range of $2.5 I_{\text{N}}$.



(a) Local deformation at the updated measured value.

(b) Smoothing around the measured value.

(c) Interpolation in the range of the determined linearity deviation.

Figure 4.12: Schematic representation of the local effect of a detected linearity deviation during iterative adaptation of the flux linkage maps. The red circles are considered known measurement points. For simplicity, only the flux linkage of the d-axis is shown.

Furthermore, if a single pair of data were used to fit the flux linkage map, then once there is a significant linearity deviation, the map would only be deformed locally at this particular point, see, e.g., Fig. 4.12(a). The disadvantage of this procedure is obvious, because the information about a possible linearity deviation is only used exactly at this point. However, it can be expected with high probability that a linearity deviation of the flux linkage is also present for surrounding current combinations, since the course of the stator inductance and thus also the flux linkage change must be monotonic. Consequently, averaging with the surrounding grid points seems reasonable to smooth the map, as shown, e.g., in Fig. 4.12(b).

However, it is still not possible with these methods to adapt the complete flux linkage map up to the map borders based on a patchy measurement map $\bar{\boldsymbol{\Psi}}$. Moreover, as explained later in Section 7.3, the control algorithm of interest requires reliable flux linkage maps that cover the entire possible operating range with parameter combinations at specified, regular intervals. Therefore, the recorded data pairs in $\bar{\boldsymbol{\Psi}}$ must be modified.

In the simplest case, this can be done by using linear interpolation. For example, it can be done as shown in [Kel12, p.83], by setting up a simple plane equation with three measurement (grid) points, see, e.g., Fig. 4.12(c). Following, based on the initial flux linkage calculated with (4.42), the flux linkage maps are adapted iteratively using the stored data sets. If a minimum number of measured values is stored in the RAM, the identification algorithm starts interpolating between the individual measurements.

However, in order to include also the outer area towards the map border, an algorithm must be developed that not only interpolates between known measured values, but also extrapolates beyond them. The left-hand side of Fig. 4.13 visualizes the problem with the four given points A to D. The points A, B and C can be regarded as initial values, which originate from the linear machine model. Point D represents a measured value affected by a linearity deviation.

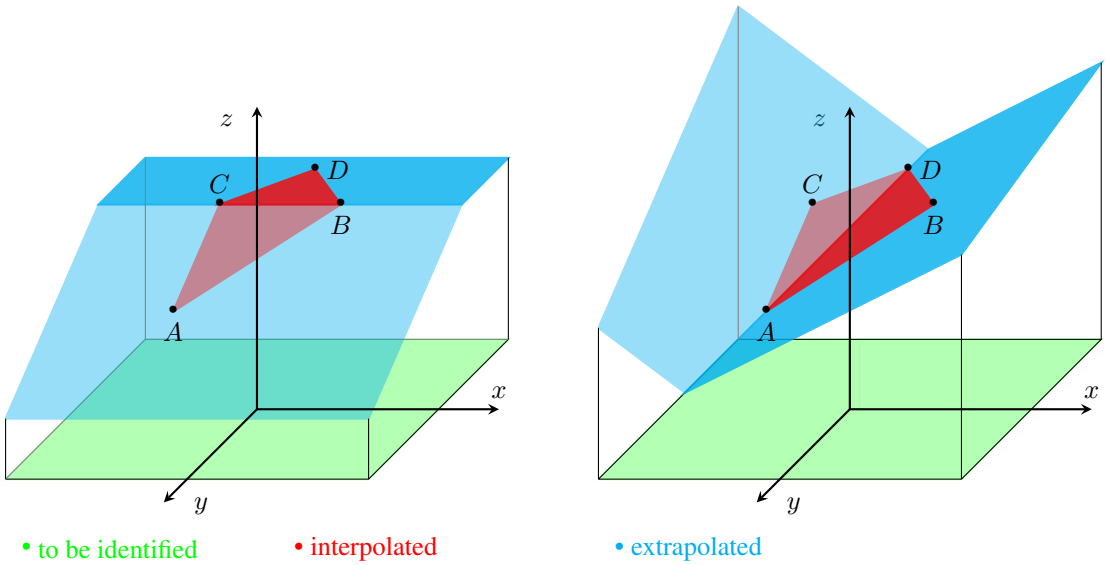


Figure 4.13: Schematic interpolation and extrapolation ranges for identification of arbitrary maps based on scattered data sets, including the triangle mesh (shown as red area) on which the gradient calculation is based. Points A, B, C, and D are considered known.

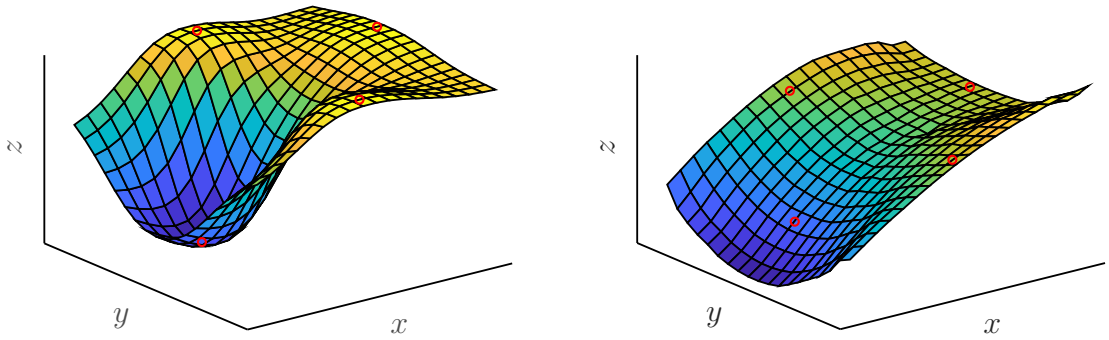
As can be seen from the figure, all characteristic values between point D and the start values are affected by the linearity deviation. In doing so, the flux linkage maps cover a wide range of current values, rendering the method useful over a broad span of operating points.

There is a variety of interpolation and extrapolation methods, e.g., spline [GT18], bicubic spline [Kel12], Kriging [Eme05, KHK11], radial basis function [WSJ+13] and several triangle-based methods [WP84]. However, most of them are quite complex and resource-intensive for real-time implementations. For this purpose, the proposed identification approach uses an inverse distance weighting (IDW) algorithm suitable for interpolation and extrapolation. A detailed description can be found, e.g., in [Ach11, Rie]. Using the conventional IDW, the to-be-found z-values are calculated at the pairs of the searched (x, y)-positions with

$$z(x,y) = \frac{\sum \left(\frac{1}{\varepsilon_i}\right)^r z(x_i, y_i)}{\sum \left(\frac{1}{\varepsilon_i}\right)^r} \tag{4.46}$$

using again the euclidean distance in (4.45). According to (4.46), the influence of the measured values in the vicinity of the to-be-found points is inversely proportional to their distance, which is weighted by the factor r.[27] By applying conventional IDW, the solution to the described problem of an unknown surface with four initial scattered values results in a map as shown in Fig. 4.14(a). Although Fig. 4.14(a) has a continuously differentiable course, the calculated

[27] Commonly, the distance between the known and to-be-found points is quadratically penalized (i.e., $r = 2$) to reduce the influence of points located further away.



(a) Conventional IDW approach.

(b) Adapted IDW approach.

Figure 4.14: Schematic solution of the map identification problem using an IDW approach. The red circles are measured points on which the IDW is based.

surface exhibits strong curvatures and is not monotonically nondecreasing. This behavior is implausible within the scope of flux linkage maps. For this reason, the IDW method is adapted by exploiting the knowledge of the flux linkage behavior. This is done by weighting and averaging the gradient of the surface instead of the z-values themselves. This is reasonable from a physical point of view, since the gradient of the flux linkage maps corresponds to the partial inductances. The latter change only comparatively slightly and in a monotonic manner, even if the electrical machine is characterized by strong saturation and cross-coupling effects.

Given the above, two steps are required to determine the local surface equations between the scattered measurements. First, a local gradient is calculated at the position of each sample. This procedure corresponds to the formation of a triangular plane defined by a sample and its neighboring points. Considering the possibility of triangular meshing several triangular points may result for a given number of samples. For example, as shown with the red area in Fig. 4.13, for four sampled points, i.e., A, B, C, and D, four triangles (see $\overset{\triangle}{ABC}$ and $\overset{\triangle}{BCD}$ on the left-hand side as well as $\overset{\triangle}{ACD}$ and $\overset{\triangle}{ABD}$ on the right-hand side of Fig. 4.13) and thus four local normal vectors result. Note that the latter indicate the surface gradient. As a result, a normal vector $\boldsymbol{n}$ and a known supporting point suffice to establish a plane equation. To this aim, the normal vectors are determined based on the known points of each triangle. For example, the normal vector of the triangle $\overset{\triangle}{ABC}$ at point A, is given by

$$\boldsymbol{n} = \boldsymbol{b} \times \boldsymbol{c} \tag{4.47}$$

based on the two vectors $\boldsymbol{b} = \vec{AB}$ and $\boldsymbol{c} = \vec{AC}$. To ensure that the normal vector describes the slope of the flux linkage map in the vicinity of the searched measurement point (i.e., A), the points used (i.e., B and C) are selected according to the following criteria. For the first vector the closest neighbor is selected and for the calculation of the second vector, according to [Kle05, p.236], a point is selected which, according to the *Delaunay Triangulation*, avoids

small angles in the spanned triangle. Second, a plausibility check is performed. Due to the fact that the differential inductance cannot be negative, negative components of the gradient are prevented along the main axis (i_d-axis for ψ_d-map, and i_q-axis for ψ_q-map). In other words, if the d-axis current increases, ψ_d must also increase. The same applies for i_q and ψ_q. In doing so, map errors resulting from corrupted measurements are avoided. Finally, after determining the normal vectors for each grid point, the z values are calculated for all grid coordinates with

$$\boldsymbol{n}(x,y) = \frac{\sum \left(\frac{1}{\varepsilon_i}\right)^r \boldsymbol{n}(x_i,y_i)}{\sum \left(\frac{1}{\varepsilon_i}\right)^r} \,. \tag{4.48}$$

As can be seen in the above expression, all normal vectors are taken into account according to their distance, i.e., inversely proportional to the distance from the searched grid point. Furthermore, strong curvatures are avoided by averaging the normal vectors. Hence, thanks to the proposed modifications, the adapted IDW approach results in monotonically nondecreasing z values and a monotonic flux linkage map, as depicted in Fig. 4.14(b).

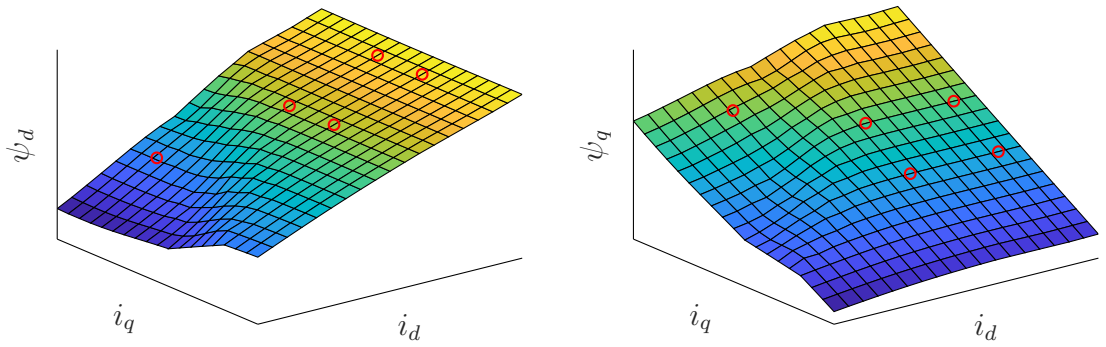
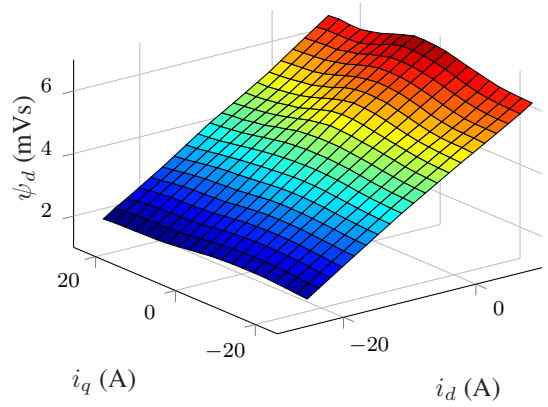


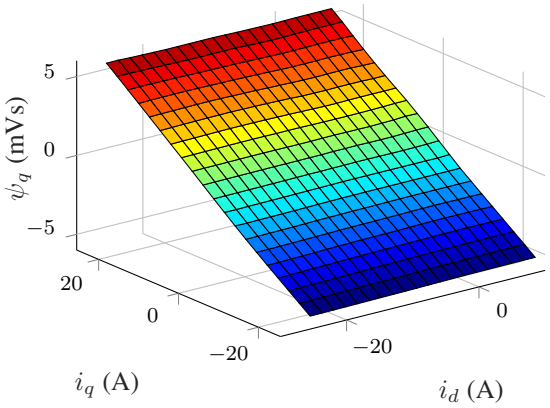
Figure 4.15: Schematic representation of an identified linearity deviation in the extrapolated flux linkage map.

A final evaluation of the extrapolation is shown in Fig. 4.15 using the example of the d- and q-axis flux linkage maps, where a measured point affected by a linearity deviation in combination with four initial values on the basis of the linear machine model—in a rectangle around the zero point—is represented. The source code of the ID algorithm is discussed in detail in [Geb20].[28] Finally, Fig. 4.16 shows the identified flux linkage maps for the test motors M3 and M4 (see Table A.3). As can be seen, with the proposed method, the main nonlinearities in the flux linkage can be relatively easily identified online during operation and used to optimize the model accuracy, i.e, the initially assumed linear machine model. This has a positive impact on the prediction accuracy, as will be shown in Chapter 7.

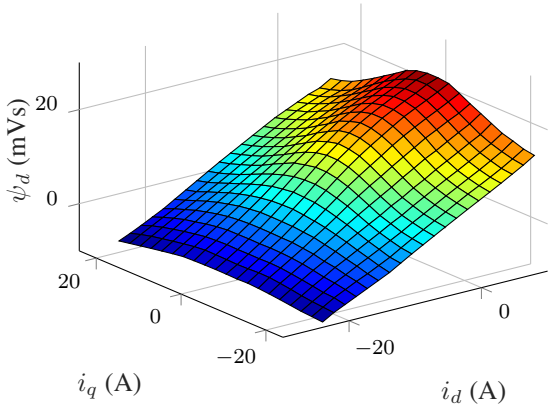
[28] The algorithm is identical for the d- and q-axis flux linkage maps except for the plausibility check. Here only the axis component of the normal vector differs, i.e., ψ_d must increase with i_d and ψ_q with i_q.



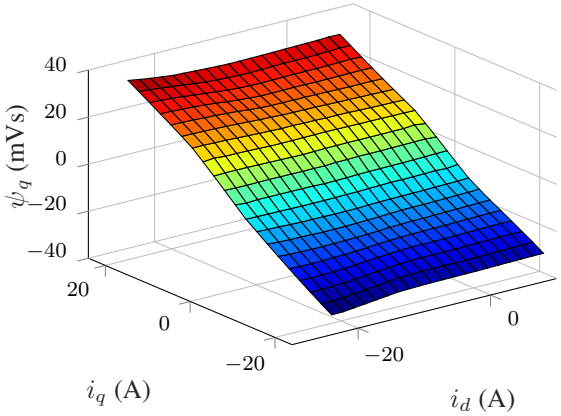
(a) $\psi_d(i_d, i_q)$ for motor M3.



(b) $\psi_q(i_d, i_q)$ for motor M3.



(c) $\psi_d(i_d, i_q)$ for motor M4.



(d) $\psi_q(i_d, i_q)$ for motor M4.

Figure 4.16: Identified flux linkage maps with 20x20 grid points for motor M3 and M4.

4.4 Evaluation of Identification Accuracy

In the following a brief error calculation and analysis for the parameter ID are presented. The validity of the methods was assessed by repeating them several times with five different motors.

4.4.1 Statistical Evaluation of Linear Parameters

An important requirement for an ID method is that the results are reproducible and do not scatter, i.e., a low standard derivation. If the results are also close to the real parameters, they have a high accuracy. Exemplary for the offline ID in Section 4.1 using motor M1—see the identified and data sheet values in Table A.3 of Appendix A.3.1—this is shown for ten iterations in Table 4.2, including the unused $R_{\mathrm{ph},q}$ resulting from the ID in q-axis. The measurement accuracy determined in Section 3.3 while using the applied voltage is the basis for the evaluation.
When analyzing the results, it is notable that the stator resistance identified in the d- and q-axis, differs by a mean of $2.23\,\mathrm{m\Omega}$. The difference in the wiring of the d- and q-axis identification

Table 4.2: Motor M1: Results of ten consecutive offline identifications.

Iteration	$R_{\text{ph},d}$ (mΩ)	L_d (mH)	$R_{\text{ph},q}$ (mΩ)	L_q (mH)	ψ_{pm} (mVs)
1	107.458	0.2565	105.334	0.2573	6.042
2	107.483	0.2576	105.337	0.2567	5.837
3	107.619	0.2582	105.151	0.2556	5.827
4	107.585	0.2612	105.164	0.2579	5.867
5	107.309	0.2578	105.107	0.2546	5.846
6	107.428	0.2609	105.158	0.2555	5.833
7	107.390	0.2576	105.169	0.2551	5.860
8	107.387	0.2569	105.290	0.2618	5.822
9	107.430	0.2617	105.112	0.2545	5.897
10	107.251	0.2570	105.190	0.2567	5.847
Mean	107.434	0.2585	105.201	0.2566	5.868
MAX	107.619	0.2617	105.337	0.2618	6.042
MIN	107.251	0.2565	105.107	0.2545	5.822
Range	0.368	0.0052	0.229	0.0073	0.219
Standard deviation σ	0.106	0.0019	0.082	0.0021	0.0649

procedure is the most likely reason for this. However, this small deviation is acceptable. Assuming a normal distribution of the measurements, the standard deviation σ and multiples of it are a measure of how the measurements scatter around their mean value. Considering a range of $\pm 3\sigma$, $99.7\,\%$ of the measurements are within this range. Thus, Table 4.3 shows the scattering of the measurements for $99.7\,\%$ of the measurements. In contrast to the other electrical parameters, the resistance is strongly influenced by the test setup. More precisely, the $R_{\text{DS(on)}}$ of the used MOSFET (see Section 2.2.1.1), the shunt current measurement, the wiring and the terminal resistance influence the total resistance. For example, motor M1 shows in the data sheet—see Table A.3—a resistance of $R_{\text{ph}} = 85\,\text{m}\Omega$, where the MOSFET used indicates a resistance of $R_{\text{DS(on)}} = 7.7\,\text{m}\Omega$ and the shunt of $R_{\text{sh}} = 10\,\text{m}\Omega$. As can be seen, the $R_{\text{DS(on)}}$ is almost $9\,\%$ and R_{sh} is approx. $11\,\%$ of the stator resistance. Considering a MOSFET, a shunt and the wiring, the actually identified resistance in Table 4.2 shows $125\,\%$ of the data sheet value of the motor. Hence, it seems reasonable to identify the resistance with the help of the inverter in the end-use setup by means of an integrated ID algorithm instead of using data sheet values.

The same applies to the other parameters. For them, however, a higher scatter is observed and thus lower precision is achieved. Nevertheless, a relatively good identification of the linear parameters in the respective setup can be achieved. From online ID, see Section 4.3.1, only R_{ph} and ψ_{pm} are used due to the limited validity of the linear parameters. Consequently, for MPC, the exemplary parameter mismatches in Section 2.6.3 can be avoided to some extent, i.e., as long as nonlinearities such as saturation are not present.

Table 4.3: Motor M1: Achievable precision of identified parameters during ID.

	$R_{\text{ph},d}$	L_d	$R_{\text{ph},q}$	L_q	ψ_{pm}
Offline - absolute $\pm 3\sigma$	$\pm 0.319\,\text{m}\Omega$	$\pm 0.0052\,\text{mH}$	$\pm 0.247\,\text{m}\Omega$	$\pm 0.0061\,\text{mH}$	$\pm 0.195\,\text{mVs}$
Offline - relative $\pm 3\sigma$	$0.59\,\%$	$4.30\,\%$	$0.47\,\%$	$4.80\,\%$	$6.64\,\%$
Online - relative $\pm 3\sigma$	$0.3 - 1.2\,\%$	$0.3 - 4.5\,\%$	—	$0.6 - 9.1\,\%$	$0.3 - 2.6\,\%$

4.4.2 Evaluation of Flux Linkage

To determine the precision of the flux linkage, i.e., ψ_d and ψ_q, and the influence of the required parameters on their identification, the tolerance of the results is first determined by estimating the error propagation. For this purpose, the maximum possible error $\Delta\zeta$ given by

$$\Delta\zeta = \sum_{l=1}^{\kappa} \left| \frac{\partial \zeta}{\partial x_l} \Delta x_l \right| \tag{4.49}$$

can be calculated based on the partial derivatives of the variable of concern, i.e., ζ, with respect to x for the number of parameters κ in the system.[29] More concretely, for calculating the maximum measurement error, i.e., $\Delta\psi_{d,\zeta}$ and $\Delta\psi_{q,\zeta}$, (4.44) is partially differentiated with respect to each of the varying parameters, i.e., R_{ph}, $\boldsymbol{i}_{dq}$, $\boldsymbol{v}_{dq}$, and ω_{el}, i.e.,

$$\frac{\partial \psi_d}{\partial R_{\text{ph}}} = -\frac{i_q}{\omega_{\text{el}}}, \quad \frac{\partial \psi_q}{\partial R_{\text{ph}}} = \frac{i_d}{\omega_{\text{el}}} \tag{4.50a}$$

$$\frac{\partial \psi_d}{\partial i_q} = -\frac{R_{\text{ph}}}{\omega_{\text{el}}}, \quad \frac{\partial \psi_q}{\partial i_d} = \frac{R_{\text{ph}}}{\omega_{\text{el}}} \tag{4.50b}$$

$$\frac{\partial \psi_d}{\partial v_q} = \frac{1}{\omega_{\text{el}}}, \quad \frac{\partial \psi_q}{\partial v_d} = -\frac{1}{\omega_{\text{el}}} \tag{4.50c}$$

$$\frac{\partial \psi_d}{\partial \omega_{\text{el}}} = -\frac{v_q - R_{\text{ph}} i_q}{\omega_{\text{el}}^2}, \quad \frac{\partial \psi_q}{\partial \omega_{\text{el}}} = \frac{v_d - R_{\text{ph}} i_d}{\omega_{\text{el}}^2}. \tag{4.50d}$$

Thus, using (4.50), the maximum possible error for each flux linkage component can be calculated with

$$\Delta\boldsymbol{\psi}_{dq,\zeta} = \left| \boldsymbol{P} \frac{\boldsymbol{i}_{dq}}{\omega_{\text{el}}} \Delta R_{\text{ph}} \right| + \left| \frac{R_{\text{ph}}}{\omega_{\text{el}}} \boldsymbol{P} \Delta \boldsymbol{i}_{dq} \right| + \left| -\frac{1}{\omega_{\text{el}}} \boldsymbol{P} \Delta \boldsymbol{v}_{dq} \right| + \left| -\frac{\boldsymbol{\psi}_{dq}}{\omega_{\text{el}}} \Delta\omega_{\text{el}} \right|. \tag{4.51}$$

The course of the maximum measurement error determined from error propagation is shown in Fig. 4.17 for motor M4, the parameters of which are provided in Table A.3 of Appendix A.3.1. Analyzing (4.51) and by visual inspection of Fig. 4.17, the following assessment can be made for the measurement uncertainty of the flux linkage identification:

1. It gets bigger with increasing current due to the uncertainty in the estimated resistance.
2. It increases with decreasing speed due to the decreasing induced voltage, which is consequently measured in a very poor measuring range.
3. It is adversely affected by unmodeled effects, i.e., higher-order harmonics and iron losses.

[29] Note that (4.49) represents the worst-case scenario in which the individual measurement deviations Δx do not compensate for each other.

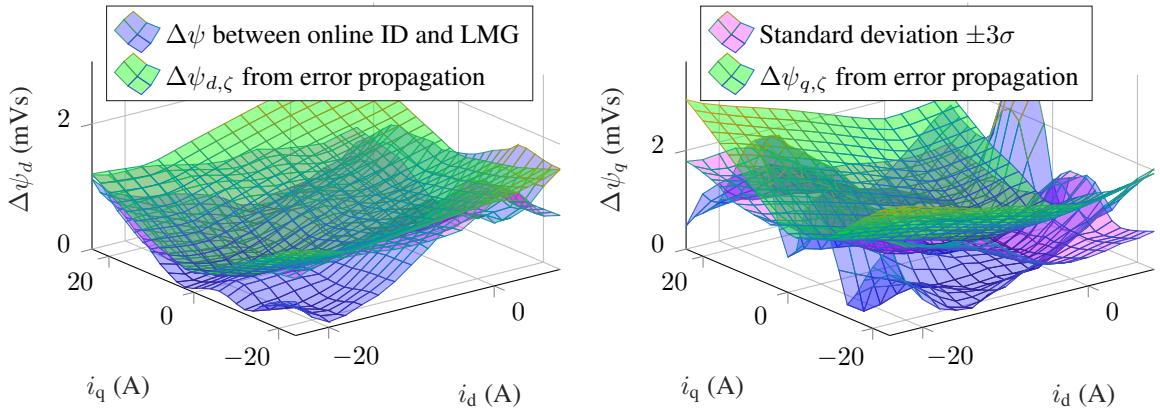


Figure 4.17: Motor M4: Maximum measurement error exemplified for the most nonlinear motor M4 at 1000 rpm. Measurements of a power analyzer (LMG) are also compared. For a better clarification, Fig. 4.16 shows the identified flux linkage.

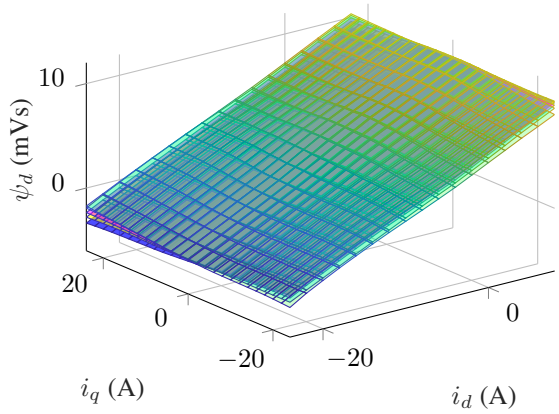
4. It increases as the measurement/estimate of the rotor angle becomes less accurate. An insufficient phase shift correction—which is necessary for the compensation of the voltage-measurement filter—as well as an increasing misalignment due to frictional effects introduce inaccuracies in the transformation from the three-phase to the dq-plane, see (2.6).

Even if the rotor position dependent flux linkage deviations are averaged over several mechanical rotor revolutions, the other effects can have a decisive impact.

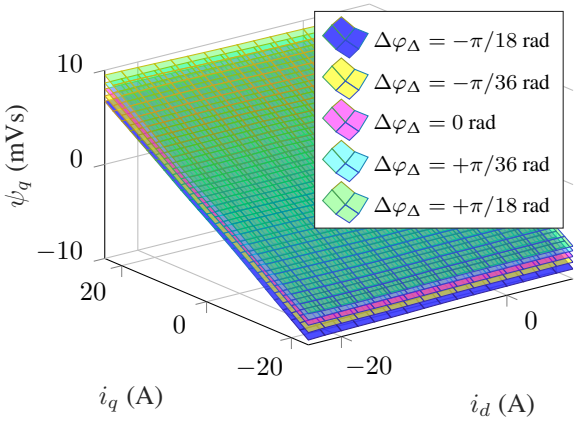
Influence of the rotor angle alignment As mentioned, the rotor angle has a large influence on an accurate voltage measurement and thus on a correct flux linkage ID. For illustration, Fig. 4.18 shows how the flux linkage maps shift/rotate when the rotor position angle is incorrect (by $\Delta\varphi_\Delta$), e.g., due to an unconsidered encoder offset $\Delta\varphi$. It is noteworthy that there is a rotational movement on the d-axis, while an offset shift appears on the q-axis.

Influence of the filter compensation Since the measured voltage[30] has harmonics—due to the switching nature of the inverter—a low-pass filter is necessary. The influence of this filter must be compensated in phase and amplitude. Fig. 4.19(a) exemplifies the effects of incorrect filter compensation at low speed, i.e., at 500 rpm. In the case study shown, three different values are used for the real low-pass filter capacitors, namely 98 nF, 100 nF, and 47 nF. While in the case of 100 nF and 47 nF the correct values are used for the filter compensation, a significant error occurs as soon as, e.g., a real low-pass filter capacitor of 98 nF is compensated with 100 nF due to unconsidered component tolerances. As exemplified by the orange line in Fig. 4.19(a), the incorrect compensation of the filter introduces an offset into the identified flux linkage. On the other hand, a correct capacitor value, e.g., 100 nF or 47 nF, will successfully compensate for the low-pass filter influence, and enable correct flux linkage identification (light and dark blue are nearly congruent).

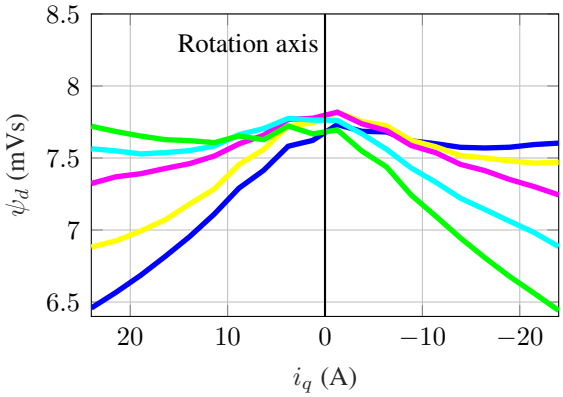
[30]The applied voltage is not used, but rather the actual measured voltage, which is common for small drives even without galvanic isolation.



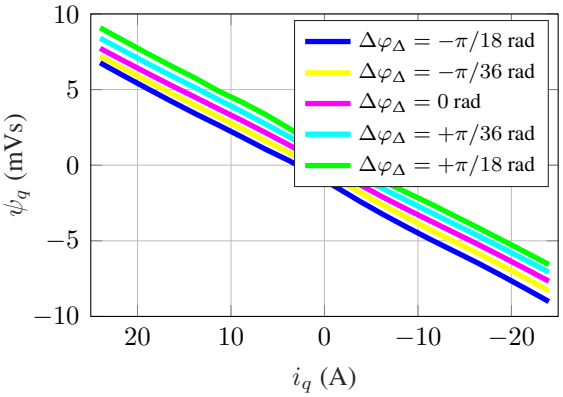
(a) $\psi_d(i_d, i_q)$ for motor M5.



(b) $\psi_q(i_d, i_q)$ for motor M5.

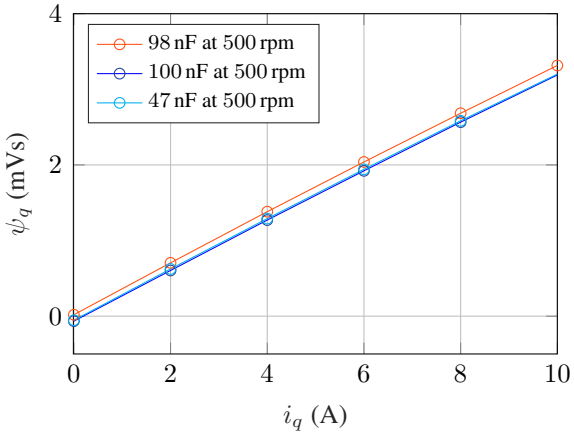


(c) $\psi_d(i_d, i_q)$ side view for $i_d = 1.3$ A.

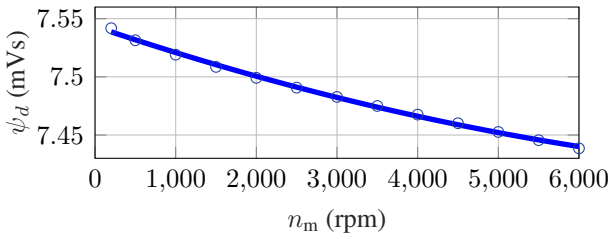


(d) $\psi_q(i_d, i_q)$ side view for $i_d = 1.3$ A.

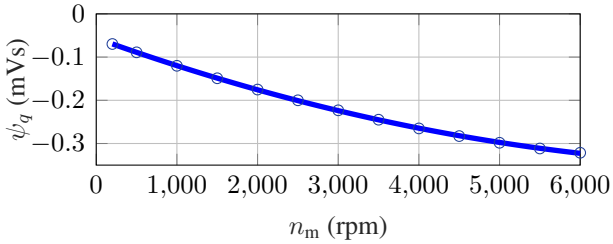
Figure 4.18: Motor M5: Identified flux linkage maps with different rotor angle alignment errors $\Delta\varphi_\Delta$ (experimental).



(a) Identification of ψ_q with different capacitors in the low-pass filter. In case of 98 nF the effect of the low-pass filter is compensated incorrectly, while for the other two capacitors it is done correctly.

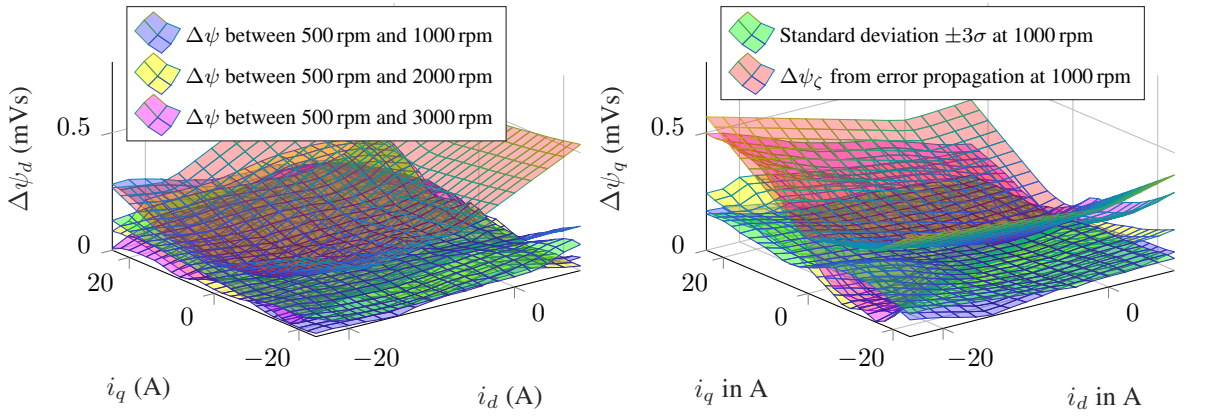


(b) Influence of iron losses on ψ_d.



(c) Influence of iron losses on ψ_q.

Figure 4.19: Motor M5: Identification of flux linkage using the online ID (left) and with open motor windings using the LMG670 (right) when ramped up by the load machine.



(a) Influence on ψ_d.

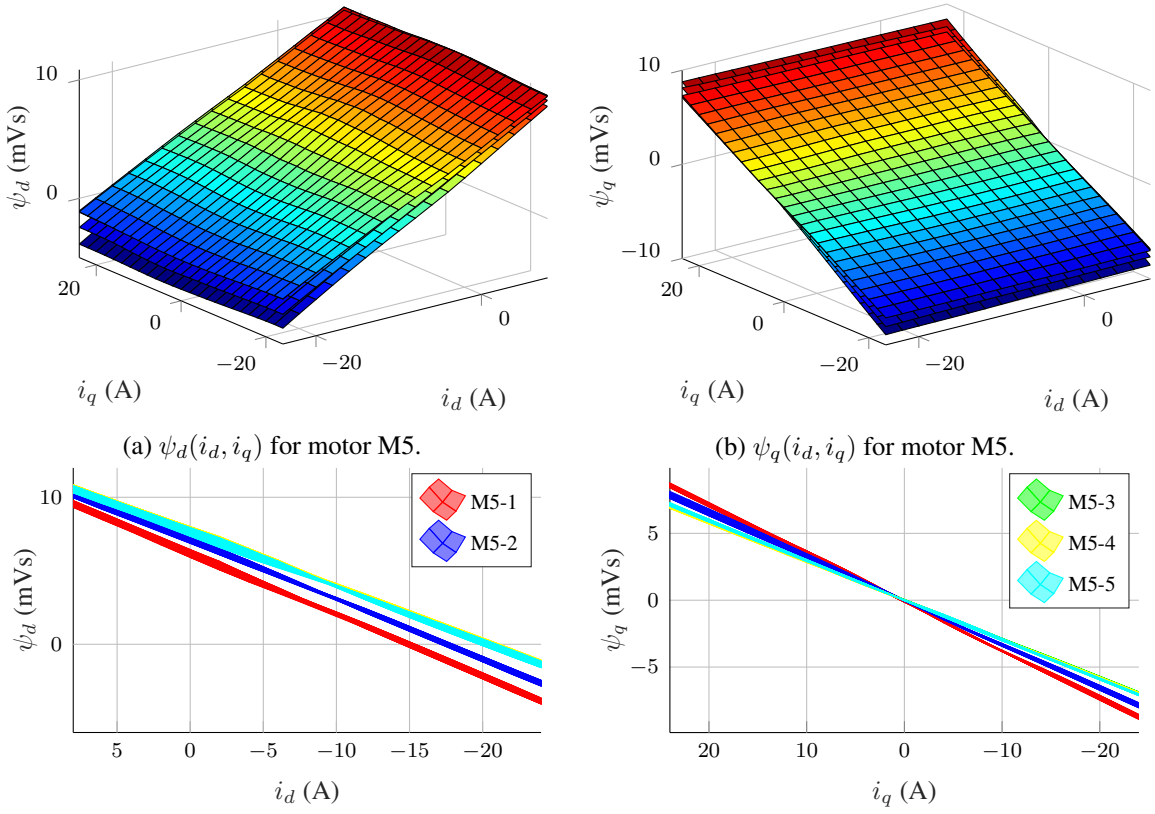
(b) Influence on ψ_q.

Figure 4.20: Motor M5: Comparison of standard deviation, error propagation and influence of speed, i.e., iron losses, on the flux linkage maps (experimental).

Influence of the iron losses Figs. 4.19(b)-4.19(c) show the influence of iron losses exemplary using motor M5 for the case where all other effects can be neglected, i.e., with open motor windings. Using the online ID, Fig. 4.20 shows the flux linkage change for different speeds respectively compared to $500\,\text{rpm}$, i.e., low speed where the influence of iron losses is small. Evidently, an increasing offset occurs with increasing speed due to the iron losses.
Considering Fig. 2.12, the effect is obvious. Since the induced voltages are proportional to the speed, so are the iron loss currents $\boldsymbol{i}_{dq,\text{Fe}}$. Since $\boldsymbol{i}_{dq}$ is composed of $\boldsymbol{i}_{dq,\text{Fe}}$ and the magnetizing current $\boldsymbol{i}_{dq,\text{m}}$, this will result in an increasing $\boldsymbol{i}_{dq,\text{Fe}}$ and decreasing $\boldsymbol{i}_{dq,\text{m}}$ for a constant $\boldsymbol{i}_{dq}$ at increasing speed. Due to the decreasing $\boldsymbol{i}_{dq,\text{m}}$, the induced voltage (or flux linkage) in the respective other axis also decreases. Since due to ψ_{pm} the induced voltage in the q-axis is bigger than in the d-axis, $i_{q,\text{Fe}}$ is also bigger than $i_{d,\text{Fe}}$. This proportionally reduces $i_{q,\text{m}}$ and thus ψ_q more than it does for ψ_d. Table 4.4 summarizes the worst case iron loss effects identified using the proposed online ID as well as an external measurement device, i.e., LMG670, as reference. Similar results were obtained e.g. in [Ric16, p. 103]. The results show that although the influence of iron losses is present and measurable, it is quite small even at rated speed. In addition, the influence must be taken with caution when comparing the standard deviation and the previously calculated error propagation. Even if the influence is small, a limited speed range, i.e., $n_{\text{m}} \in [5 - 30]\,\%$ of n_{N}, is recommended for the identification of the considered test machines. In order to be able to use the proposed online ID at higher speeds, an additional term should be included in (4.44) to account for the iron losses—if pronounced—during steady state, e.g., by using a voltage error term [Ric16, p. 25], or a parallel resistance [RDD14, KSP11, BAC+19].

Table 4.4: Comparison of the standard deviation and the influence of the iron losses over speed.

	$\Delta\psi_d$			$\Delta\psi_q$		
	Motor M1	Motor M4	Motor M5	Motor M1	Motor M4	Motor M5
Standard deviation $\pm 3\sigma$	2.5 %	7.0 %	2.6 %	4.0 %	4.9 %	2.5 %
Iron loss influence online ID	2.8 %	3.1 %	2.8 %	5.4 %	2.0 %	5.1 %
Iron loss influence LMG670 ID	2.6 %	3.3 %	2.3 %	4.6 %	2.4 %	2.7 %



(a) $\psi_d(i_d, i_q)$ for motor M5.

(b) $\psi_q(i_d, i_q)$ for motor M5.

(c) $\psi_d(i_d, i_q)$ side view for $i_q = 1.3$ A.

(d) $\psi_q(i_d, i_q)$ side view for $i_d = 1.3$ A.

Figure 4.21: Motor M5: Identified flux linkage maps with 25x17 grid points for five motors with the same part number but from different production batches (experimental).

Influence of manufacturing and aging Fig. 4.21 shows the proposed flux linkage online ID for motor M5, repeated and compared for several motors of the same type, i.e., similar part number, but different production dates. The associated production date and the linear parameters are summarized in Table 4.5. As can be seen, even if a similar data sheet is provided, the motors differ a lot, at least for different batches. Hence, data sheet values or even finite element method (FEM) calculations are too general and do not help much in practice to recognize manufacturing influences or effects occurring through aging [HG15]. The latter or other magnets are e.g. a possible explanation for the reduced ψ_{pm} in case of the older motors in Table 4.5.
Therefore, parameter identification as an integrated part of the control algorithm, as presented in this chapter, is beneficial. In this way, the parameters can be adjusted as accurately as possible in the model while still achieving reasonable usability.

Table 4.5: Comparison of motors with the same part number but different production batches.

Motor	Production date	R_{ph} (mΩ)	L_d (mH)	L_q (mH)	ψ_{pm} (mVs)	J (gcm^2)
M5-1	06.11.2015	88.5	0.31	0.35	6.59	511.6
M5-2	07.03.2017	86.5	0.29	0.32	7.51	516.9
M5-3	10.04.2018	86.2	0.25	0.26	8.12	494.3
M5-4	10.04.2018	89.4	0.26	0.27	8.13	495.1
M5-5	10.04.2018	86.6	0.26	0.26	8.07	475.0

5 Reduction of the Search Space for FCS-MPC

This chapter describes how to reduce the computational burden of FCS-MPC. In this context a distinction must be made between „achieving a long horizon“ and „reducing the search space“. The former is discussed, e.g., in [KGN+14], where a review of strategies for long prediction intervals is shown, which points out the move blocking strategy, the extrapolation strategy and the event-based horizon. These methods allow a long horizon, but not by limiting the search space, i.e., discrete switching possibilities.
The latter, which is discussed in this chapter, deals with the reduction of the discrete number of switching states within the search space with respect to a specific prediction interval.

5.1 Sphere Decoding Algorithm

As briefly described in Section 2.5.2.3, the sphere decoding algorithm is the most widely used solver for IQPs. Consequently, using SDA, with the same computational effort a much longer horizon can be calculated. However, two main disadvantages are present. First, the SDA requires an LTI model, see Section 2.5.2.2. Second, the computational feasibility can be exceeded in the case of transients, see, e.g., [ARB+19].

As mentioned in Section 2.5.1.2, for an LTI system e.g. the velocity dependence must be linearized and constant inductances as well as resistance must be used, i.e., it must be assumed that they have a certain fixed value. This linearization may be acceptable for drives with slow mechanical time constants, which almost always operate in the same operating range and have a linear magnetic circuit. However, avoiding an update of the state matrices and assuming a time-invariant Hessian matrix $\boldsymbol{H}_\mathrm{f}$—see Section 2.5.2.2—means that the same linearized matrices are used over the whole operating range which can lead to suboptimality, e.g., 2.1 % in [LKG+17]. However, for highly utilized PMSMs treated in this work—see Section 2.2.1.2—the parameters of the electrical model vary significantly during operation, e.g., due to saturation or temperature. This leads to additional suboptimality in the solution of the SDA if the matrices, e.g., the Hessian matrix $\boldsymbol{H}_\mathrm{f}$, are not updated, i.e., tracked, over the operating range. Consequently, an update of the state matrices and recalculation of the Hessian matrix are theoretically necessary as soon as the operating point changes [GQ15].

As shown in Section 2.5.2.2, this requires matrix inversions. Considering small electrical drives and the respective small mechanical and electrical time constants, speed, temperature and nonlinear magnetic effects vary even faster, i.e., the matrices have to be recalculated more often. This drastically increases the computational load and limits the advantages of SDA at the aforementioned high control frequencies, e.g., $f_{\mathrm{cf}} = 100\,\mathrm{kHz}$. Depending on the update rate, this may imply that real-time calculation of the problem is not feasible. As a side note, the need for parameter tracking, see Chapter 4, becomes apparent when updating the state matrices and is even emphasized hereafter in Chapter 7.

Another issue when using the SDA is the increase of the computational load and thus the possible infeasibility of the solution in the presence of transients [ARB+19]. Even though, as shown, e.g., in [KGA18, ARB+19, BAA+20], more and more solutions are emerging to address this drawback. Nevertheless, the computational effort for a horizon $N_{\mathrm{p}} = 4$ is, e.g., $59.43\,\mu\mathrm{s}$ [ARB+19] or about $40\,\mu\mathrm{s}$ [BAA+20], which exceeds the required control interval of $T_{\mathrm{cf}} = 10\,\mu\mathrm{s}$.

5.2 Preselection Based on Dead-Beat Control Action

To keep the computational complexity of the proposed direct MPC method modest so as to render its real-time implementation with the desired (high) control frequencies possible, a preselection method—introduced as "heuristic preselection" in [SKT+13]—is preferred and employed. This method utilizes the dead-beat control action so as to reduce the search space.
As a result, instead of evaluating eight candidate switch positions (i.e., seven SVs) at each prediction step, only four candidate switch positions (i.e., three unique SVs) are taken into consideration. The dead-beat solution is given by

$$\boldsymbol{v}_{dq,\mathrm{db}}(k) = \boldsymbol{L}_{dq}\frac{\boldsymbol{i}_{dq}^{*}(k) - \boldsymbol{i}_{dq}(k)}{T_{\mathrm{cf}}} + R_{\mathrm{ph}}\boldsymbol{i}_{dq}(k) + \omega_{\mathrm{el}}(k)\left(\boldsymbol{P}\boldsymbol{L}_{dq}\boldsymbol{i}_{dq}(k) + \begin{bmatrix} 0 \\ \psi_{\mathrm{pm}} \end{bmatrix}\right). \tag{5.1}$$

With (5.1), based on the dead-beat control action, the angle of the desired voltage vector $\boldsymbol{v}_{dq,\mathrm{db}}(k)$ that drives the current to its reference within one control interval T_{cf} can be found with

$$\gamma_{\mathrm{db}}(k) = \arctan2(\boldsymbol{v}_{dq,\mathrm{db}}(k)) + \varphi,\ \{\gamma_{\mathrm{db}} \in \mathbb{R} \mid 0 \leq \gamma_{\mathrm{db}} < 2\pi\}\,. \tag{5.2}$$

Using the angle γ_{db} of the dead-beat solution (5.2), the triangular sector (one out of the six, see Fig. 2.5) in which $\boldsymbol{v}_{dq,\mathrm{db}}(k)$ lies can be determined, for details see Table 5.1. Thus, only two active and two zero SVs that form the sector are preselected, as illustrated in Table 5.1.

Table 5.1: Candidate Voltage SVs Depending on $\gamma_{\text{db}}(k) = \angle \boldsymbol{v}_{dq,\text{db}}(k)$.

γ_{db}	Sector	SVs
$\left[0, \frac{\pi}{3}\right)$	I	$\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_{0/7}$
$\left[\frac{\pi}{3}, \frac{2\pi}{3}\right)$	II	$\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_{0/7}$
$\left[\frac{2\pi}{3}, \pi\right)$	III	$\boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_{0/7}$
$\left[\pi, \frac{4\pi}{3}\right)$	IV	$\boldsymbol{v}_4, \boldsymbol{v}_5, \boldsymbol{v}_{0/7}$
$\left[\frac{4\pi}{3}, \frac{5\pi}{3}\right)$	V	$\boldsymbol{v}_5, \boldsymbol{v}_6, \boldsymbol{v}_{0/7}$
$\left[\frac{5\pi}{3}, 2\pi\right)$	VI	$\boldsymbol{v}_6, \boldsymbol{v}_1, \boldsymbol{v}_{0/7}$
Equivalence between voltage SVs and switch positions		
$\boldsymbol{v}_0 \equiv [-1\ -1\ -1]^T, \boldsymbol{v}_1 \equiv [\ 1\ -1\ -1]^T, \boldsymbol{v}_2 \equiv [\ 1\ \ 1\ -1]^T, \boldsymbol{v}_3 \equiv [-1\ \ 1\ -1]^T$		
$\boldsymbol{v}_4 \equiv [-1\ \ 1\ \ 1]^T, \boldsymbol{v}_5 \equiv [-1\ -1\ \ 1]^T, \boldsymbol{v}_6 \equiv [\ 1\ -1\ \ 1]^T, \boldsymbol{v}_7 \equiv [\ 1\ \ 1\ \ 1]^T$		

Consequently, only these four SVs are candidate solutions for the subsequent MPC problem. In order to reduce the calculation effort for the reference tracking term in the cost function, only one zero SV is considered since the impact of both zero SVs on the stator phase current is identical. Moreover, the switching penalization term in the cost function respects both zero SVs to reduce the switching frequency (and thus switching losses), i.e., the zero SV ($\boldsymbol{v}_0$ or $\boldsymbol{v}_7$) that results in less switching effort—with respect to the previously applied SV—will be chosen by the cost function, see, e.g., (2.75).

It has to be mentioned that although there is a possibility that the reduction of the search space based on the location of the dead-beat control action on the plane can lead to suboptimal results[1] [KG20]—especially during transients and at the sector border—the simulation and experimental results based on the chosen case study, i.e., two-level VSI with three phase legs for PMSM where $N_{\text{p}} \leq 5$, do not show any suboptimal performance.

Finally, it is worth mentioning that the proposed method still provides an exponential increase of the computational effort over the horizon. However, for the considered application, i.e., a two-level VSI, only $3^{N_{\text{p}}}$ instead of $8^{N_{\text{p}}}$ possibilities arise. Compared to the SDA, which offers a linear increase in the computational load over the horizon, this is still disadvantageous. However, no operating point-dependent linearizations are necessary for the dead-beat preselection. Consequently, for small drives, as shown in the following chapters, the dead-beat preselection is preferable.

[1]The main reason for a possible suboptimality is that the dead-beat solution is theoretically not identical to the solution of the cost function used for the DMPC problem. This is due to the fact that the mathematical formulation of the dead-beat problem only considers reference tracking, which is fundamentally different from the mathematical formulation of the DMPC cost function, which also includes possible further criteria and constraints. For this reason, dead-beat preselection may not be called an unconstrained solution.

6 Increasing the Switching Granularity of FCS-MPC

One of the main issues when using FCS-MPC for small electrical drives—with time constants of just a few ms or even µs—is the insufficient switching granularity, as it was demonstrated in Section 2.6.1. This is especially critical in the partial load range, i.e., at low modulation index, where this leads to unnecessarily high current ripple and thus torque ripple. Although such a low granularity might be acceptable during transient operation, since a highly dynamic behavior is of greater importance, for steady-state operation it has an adverse effect. Specifically, since one switching state is applied to the converter for the whole control interval T_{cf}, the theoretical maximum f_{sw} is limited to half of f_{cf}. Therefore, the minimization of the torque ripple at steady-state becomes more challenging. In particular, as reported in [KG20, Section V], the control frequency should be about two orders of magnitude higher than the switching frequency for a favorable performance. Even though, as shown in Chapter 3, new and powerful calculation platforms are available that allow control frequencies f_{cf} up to several hundred kHz [WDK17b], this is still insufficient, as demonstrated in Section 2.6.1.
Consequently, a modulator, such as CB-PWM or SVM, seems to be advantageous compared to FCS-MPC, since the state of the converter switch positions can change at any time instant within T_{cf}. Owing to the higher granularity of modulator-based schemes, lower torque ripple—especially at low modulation index—can be achieved for the same f_{cf} compared to conventional FCS-MPC. Therefore, introducing the concept of variable switching points (VSPs), i.e., time instants within the control interval T_{cf} at which the converter switches change state (also referred to as switching instants), to the FCS-MPC problem is meaningful. In doing so, FCS-MPC can apply more than one switch position to the converter within one T_{cf}. Thus, higher switching granularity can be achieved, which, in turn, allows for a reduction of the current and torque distortions. As can be understood, a combination of high switching granularity and the fast dynamic response of FCS-MPC seems beneficial.

Motivated by this, several works have adopted the notion of VSP and combined it with FCS-MPC, see, e.g., [LK12, KSK+14, SKT+13, TZW+15, ZXLZ14, ZPY16, ZXL+14, TFG+17, ZBY18, KMG18, WKDK19, ZDMB20, MCDA+19, KNG20]. Owing to the introduced VSPs, these approaches can implement up to four different switch positions in one control interval T_{cf}. A good overview is given in [KLGK20]. FCS-MPC with two different switch positions during one T_{cf} was introduced, e.g., in [Kar13, KSK+14, SKT+13, TZW+15, ZXLZ14], whereas

four different switch positions (akin to SVM) are implemented in [TFG+17, ZBY18, KMG18, KNG20]. In [KSK+14] a so-called variable switching point predictive torque control (VSP^{2}TC) and in [SKT+13] a variable switching point predictive current control (VSP^{2}CC) are presented. However, the approaches in [Kar13, KSK+14, SKT+13, SKK+15, KAK18] for VSP^{2}TC and VSP^{2}CC use only one-step prediction horizon, i.e., $N_\mathrm{p} = 1$. Furthermore, in [SKT+13] only one transition per T_cf is considered, since a combination of the previous and one new SV is applied. On the contrary, long-horizon VSP^{2}CC ($N_\mathrm{p} = 5$) is evaluated in [AKMK15], showing improved performance of the drive system, but only in simulation. In [TZW+15] a so-called modulated MPC (M^{2}PC) is introduced. Methods such as [TZW+15, ZXLZ14] solve the optimization problem in two sequential steps, first the optimal SVs are chosen and second their application time is computed. The sequential structure, however, can lead to suboptimal results. Furthermore, strategies such as the one introduced in [KMG18], although achieving fixed f_sw and deterministic harmonic spectra—making them suitable for grid-tied converters—do not exhibit the fast response of FCS-MPC during transients and the resulting three VSPs enforce switching—and switching losses—even if it is unnecessary, e.g., during transients.

6.1 Proposed VSP^{2}CC Concept and Algorithm

All previous strategies usually apply two (or more) different switch positions within each control interval. In addition, the dynamics are often reduced by multistage control principles. The optimization problem proposed in this chapter, first published in [WKDK19], however, allows the controller to decide in real time, based on the cost function, whether to apply one or two new switch positions within one T_cf. In the former case, similar to the classic FCS-MPC, a new switch position can be applied or the previous one can be kept. In the latter case, a variable switching point, i.e., a time instant within the control interval at which the inverter switches change state, is computed. With this degree of freedom, during transients, the maximum available voltage can be applied to the inverter, either by implementing one active or a combination of two active SVs. At steady-state operation, on the one hand, one active SV can be changed into a zero one (or vice versa) at the computed VSP, thus reducing the current and torque ripples. On the other hand, a zero SV can be applied for the full T_cf. Thus, the proposed controller aims to reduce the current ripples while achieving excellent (the maximum) dynamic behavior.

For the realization of the proposed algorithm (see Fig. 6.1), a preselection is first done to minimize the computational effort. By doing so, the search space is reduced from eight candidate SVs to three, see Section 5.2. Afterwards, the algorithm evaluates the stator currents for different combinations of the three candidate SVs by calculating (2.73) for a horizon of N_p steps.

As a compromise between possibilities and computational effort, a distinction is made:

1. For the first prediction step, the three candidate SVs are evaluated by taking into account two possibilities: either one SV is applied to the inverter for the whole T_{cf} (i.e., akin to the conventional FCS-MPC), or two SVs are executed within one T_{cf} according to the principles of VSP^{2}CC. See the blue calculations on the right-hand side of Fig. 6.1.

2. For the prediction steps further in the horizon, i.e., $N_{\mathrm{p}} > 1$, the algorithm theoretically has the ability to evaluate all the possible SV combinations by employing the VSP^{2}CC concept (nine per step). This implies that two SVs can be applied in each prediction step. By doing so, nonetheless, the possible solutions to be enumerated are $3^{2N_{\mathrm{p}}}$, i.e., the optimization problem can become computationally intractable for $N_{\mathrm{p}} > 1$. To overcome this issue, the concept of conventional FCS-MPC is employed for steps $N_{\mathrm{p}} > 1$, meaning that only three candidate solutions are considered from the second step of the horizon onwards. See the central (purple) steps and the left-hand branch in Fig. 6.1.

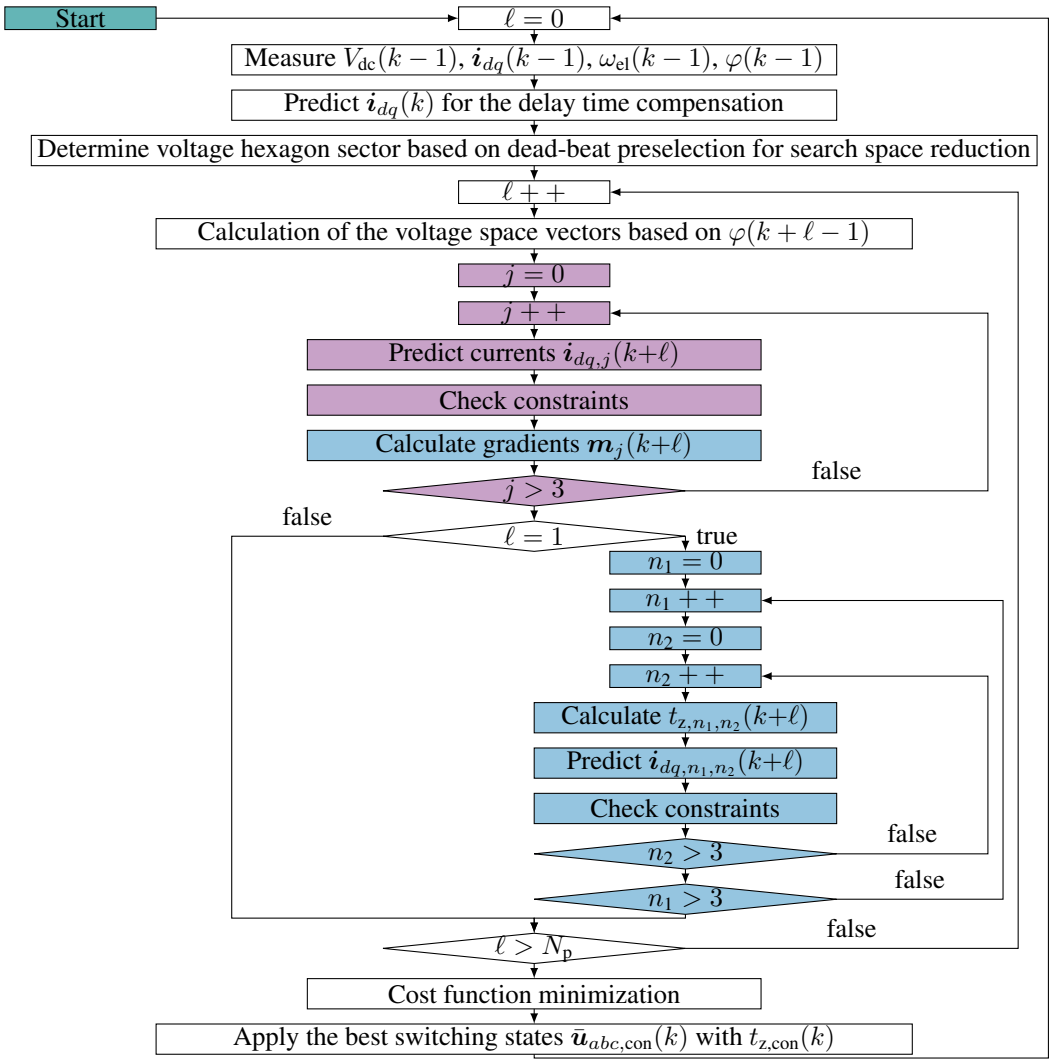


Figure 6.1: Block diagram of the proposed VSP^{2}CC concept.

As a result, the MPC algorithm has to evaluate a total of only $3^{N_\mathrm{p}+1}$ possible solutions. As can be understood, the aforementioned simplification greatly reduces the computational complexity, thus facilitating its real-time implementation. Finally, it should be stressed that, owing to the receding horizon, only the first element (i.e., control action) of the solution is implemented. From a performance point of view, this means that the first step of the horizon is the one with the greatest significance. Hence, adopting the conventional FCS-MPC for $N_\mathrm{p} > 1$ does not adversely affect the system performance.

To decide which SVs meet the control objectives, as mentioned in the beginning of this section, the slopes of i_d and i_q are calculated. Before doing so, the following assumptions are made:

- Due to the very small T_cf (i.e., 10 µs), the saturation does not affect the slopes during one control interval T_cf. Therefore, the slopes are assumed to be piecewise linear.[1]
- The rotor angle φ is kept constant during one interval T_cf.[2] Note that the possible values of the stator voltage (2.12) are updated at each prediction step.
- Nonlinear time-varying parameters such as R_ph, ψ_pm and ω_m do not affect the current slopes due to the significantly larger associated time constants as compared to T_cf.

By assuming constant current slopes for the entire interval T_cf, similar to [KSK+14, KAK18, AKMK15], the slopes of the resulting current trajectory for each of the three SVs can be calculated by using

$$\boldsymbol{m}_{dq}(k) = \frac{\boldsymbol{i}_{dq}(k) - \boldsymbol{i}_{dq}(k-1)}{T_\mathrm{cf}} = \frac{\Delta \boldsymbol{i}_{dq}(k)}{T_\mathrm{cf}}\,, \tag{6.1}$$

where $\boldsymbol{m}_{dq} = [m_d\ m_q]^T \in \mathbb{R}^2$ and $\boldsymbol{i}_{dq}(k)$ is calculated based on (2.73).[3] Since absolute inductances are used in this calculation, the VSP^{2}CC approach will be renamed to L-VSP^{2}CC in the following for reasons explained later in the thesis.

Given the constant slopes over one T_cf, the calculated $\boldsymbol{m}_{dq}$ is identical in the time steps k and $k+1$. By denoting these slopes with the subscripts n_1 and n_2, respectively, the rms current error on the d- and q-axis can be calculated over the whole interval T_cf by using [SKT+13, SKK+15]

$$e_{\mathrm{rms}^2,n_1,n_2}(t_\mathrm{z}) = \frac{1}{T_\mathrm{cf}}\left(\int_0^{t_{\mathrm{z},n_1,n_2}} ||\boldsymbol{i}_{dq,0} + \boldsymbol{m}_{dq,n_1} t - \boldsymbol{i}^*_{dq}||_2^2 \mathrm{d}t + \int_{t_{\mathrm{z},n_1,n_2}}^{T_\mathrm{cf}} ||\boldsymbol{i}_{dq,t_\mathrm{z}} + \boldsymbol{m}_{dq,n_2} t - \boldsymbol{i}^*_{dq}||_2^2 \mathrm{d}t \right), \tag{6.2}$$

where $t_z \in [0, T_\mathrm{cf}]$. An illustrative example of the resulting error area (in red) is shown in Fig. 6.2 with a combination of $\boldsymbol{v}_3$ and $\boldsymbol{v}_1$ for i_q. Expression (6.2) is calculated for each SV

[1] In the case where the PMSM used exhibits a significant saturation effect during one T_cf, this assumption does not apply and the approach must be refined.

[2] For machines with many pole pairs, the variation of the rotor position within one T_cf may affect the VSP calculation at high speeds. For example, when $n_\mathrm{m} = 4000$ rpm, $p = 4$ and $T_\mathrm{cf} = 10$ µs, the rotor position change is $\varphi \approx 0.017$ rad in one T_cf, leading to a possible voltage variation up to $1.8\,\%$.

[3] The interlock time (here 200 ns) is compensated for according to [ITF+12], see also Section 3.3.

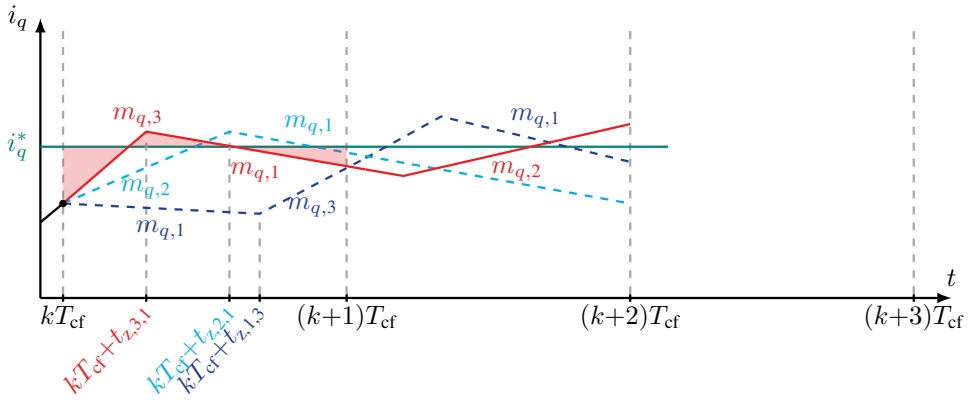


Figure 6.2: Variable switching points by the intersection of two i_q current trajectories for $N_\mathrm{p}=2$.

combination including the possibility that one SV is applied for the whole interval T_cf, as shown in Fig. 6.1. The variable switching point t_{z,n_1,n_2}, which minimizes the current ripple for each SV combination, can be obtained by setting the derivative of (6.2) to zero, i.e., $\mathrm{d}e_{\mathrm{rms}^2}/\mathrm{d}t_\mathrm{z}=0$. This yields

$$t_{\mathrm{z},n_1,n_2}=\frac{a_{n_1,n_2}+b_{n_1,n_2}}{c_{n_1,n_2}+d_{n_1,n_2}} \quad \text{where} \tag{6.3}$$

$$a_{n_1,n_2}=(m_{d,n_2}-m_{d,n_1})(2i_d-2i_d^*+T_\mathrm{cf}m_{d,n_2}),\quad b_{n_1,n_2}=(m_{q,n_2}-m_{q,n_1})(2i_q-2i_q^*+T_\mathrm{cf}m_{q,n_2}),$$

$$c_{n_1,n_2}=(m_{d,n_1}-m_{d,n_2})(2m_{d,n_1}-m_{d,n_2}),\quad d_{n_1,n_2}=(m_{q,n_1}-m_{q,n_2})(2m_{q,n_1}-m_{q,n_2}).$$

To further simplify the calculation of t_{z,n_1,n_2}, so as to alleviate the associated computational effort, (6.1) is utilized. In doing so, T_cf is set as a common factor, whereat the range of the values t_{z,n_1,n_2} is reduced considerably, resulting in

$$t_{\mathrm{z},n_1,n_2}=T_\mathrm{cf}\frac{a_{n_1,n_2}+b_{n_1,n_2}}{c_{n_1,n_2}+d_{n_1,n_2}} \quad \text{where} \tag{6.4}$$

$$a_{n_1,n_2}=(\Delta i_{d,n_2}-\Delta i_{d,n_1})(2i_d-2i_d^*+\Delta i_{d,n_2}),\quad b_{n_1,n_2}=(\Delta i_{q,n_2}-\Delta i_{q,n_1})(2i_q-2i_q^*+\Delta i_{q,n_2}),$$

$$c_{n_1,n_2}=(\Delta i_{d,n_1}-\Delta i_{d,n_2})(2\Delta i_{d,n_1}-\Delta i_{d,n_2}),\quad d_{n_1,n_2}=(\Delta i_{q,n_1}-\Delta i_{q,n_2})(2\Delta i_{q,n_1}-\Delta i_{q,n_2}).$$

This is advantageous for the subsequent implementation (especially when using fixed-point) on an FPGA, as described in Section 6.3, since the value range is greatly reduced and the calculation of the fraction is therefore independent of the respective calculation interval T_cf. Observing (6.4), it is evident that when the same voltage SV is evaluated for both time steps k and $k+1$ the current slope remains constant over the whole T_cf. This implies that the associated switching time is $t_{\mathrm{z},n_1,n_2}=0$, i.e., only one switch position $\boldsymbol{u}_{abc}$ is applied within one T_cf. As a result, either one switch transition occurs within T_cf, or switching is avoided altogether in case the previously applied switch position is the same.

Furthermore, it should be mentioned that voltage SV combinations that lead to current trajectories that do not intersect at all within T_{cf}, or lead to $t_{\mathrm{z},n_1,n_2} > T_{\mathrm{cf}}$ are excluded as infeasible. Afterwards, by substituting T_{cf} with t_{z,n_1,n_2} in (2.73), all currents $\boldsymbol{i}_{dq,n_1,n_2}$ are calculated.

6.2 Optimization Problem

The cost function is designed to minimize the current ripple while at the same time controlling the switching frequency. Moreover, it should take into account the possibility of applying one or two voltage SVs (i.e., switch positions) in the first prediction horizon step. Depending on the previous switch positions, no switching action, one or two switching transitions are performed. On this basis, the formulated cost function to be minimized in real time is[4]

$$J_{\mathrm{f}}(k) = \sum_{\ell=k}^{k+N_{\mathrm{p}}-1} \Big(||\boldsymbol{y}^*(\ell+1) - \boldsymbol{y}(\ell+1)||_2^2 + \hat{f}(\boldsymbol{i}_{dq}(\ell+1)) + \lambda_{\mathrm{u}} ||\boldsymbol{\Delta}[\bar{\boldsymbol{u}}_{abc}^T(\ell-1)\ \bar{\boldsymbol{u}}_{abc}^T(\ell)]^T||_1 \Big) \tag{6.5}$$

where

$$\hat{f}(\boldsymbol{i}_{dq}(\ell+1)) = \begin{cases} i_{\max} & \text{if } ||\boldsymbol{i}_{dq}(\ell+1)||_2 > i_{\max} \\ 0 & \text{if } ||\boldsymbol{i}_{dq}(\ell+1)||_2 \leq i_{\max} \end{cases} \tag{6.6}$$

with $\boldsymbol{y}^* = [\boldsymbol{i}_{dq}^{*T}\ \boldsymbol{i}_{dq}^{*T}]^T \in \mathbb{R}^4$ being the reference vector and $\boldsymbol{y} = [\boldsymbol{i}_{dq,\mathrm{t_z}}^T\ \boldsymbol{i}_{dq,\mathrm{T_{cf}}}^T]^T \in \mathbb{R}^4$ the output vector. The sequence of manipulated switch positions over a finite horizon of $N_{\mathrm{p}} \in \mathbb{N}^+$ time steps is defined as

$$\boldsymbol{U}(k) = [\bar{\boldsymbol{u}}_{abc}^T(k)\ \bar{\boldsymbol{u}}_{abc}^T(k+1)\ \ldots\ \bar{\boldsymbol{u}}_{abc}^T(k+N_{\mathrm{p}}-1)]^T \in \mathbb{U} \tag{6.7}$$

where $\bar{\boldsymbol{u}}_{abc} = [\boldsymbol{u}_{abc,0}^T\ \boldsymbol{u}_{abc,\mathrm{t_z}}^T]^T$ and $\mathbb{U} = \{-1, 1\}^{2n_h N_{\mathrm{p}}}$. Moreover, $\boldsymbol{\Delta}[\bar{\boldsymbol{u}}_{abc}^T(\ell-1)\ \bar{\boldsymbol{u}}_{abc}^T(\ell)]^T$ with $\boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{0}_{n_h \times n_h} & -\boldsymbol{I}_{n_h} & \boldsymbol{I}_{n_h} & \boldsymbol{0}_{n_h \times n_h} \\ \boldsymbol{0}_{n_h \times n_h} & \boldsymbol{0}_{n_h \times n_h} & -\boldsymbol{I}_{n_h} & \boldsymbol{I}_{n_h} \end{bmatrix}$, where $\boldsymbol{0}_{n_h \times n_h} \in \mathbb{N}^{n_h \times n_h}$ and $\boldsymbol{I}_{n_h} \in \mathbb{N}^{n_h \times n_h}$ are the zero and identity matrices of appropriate dimensions (here $n_h = 3$, see Section 2.2.1.1), denotes the penalization of the control action, and, consequently, of the switching frequency, which is weighted by $\lambda_{\mathrm{u}} > 0$ similar to (2.75). In this way, the average switching frequency can be reduced at the expense of higher current ripple. Finally, (6.6) represents a hard constraint on the stator current implemented as a protection mechanism with $i_{\max}$ being the maximum current in

[4]Note that, in order to reduce the switching frequency (and thus the switching losses), while keeping the computational load modest, the reference tracking error term is calculated only for one zero voltage SV (i.e., $\boldsymbol{v}_0$), whereas the switching error term considers both $\boldsymbol{v}_0$ and $\boldsymbol{v}_7$. Following, the zero voltage SV that results in less switching effort—with respect to the previously applied voltage SV—is chosen.

per-unit, similar to (2.76).[5]

Inspecting (6.5), it can be seen that the cost function is designed to account for both possibilities when solving the L-VSP^{2}CC problem, namely that either two or one switch positions can be implemented within one control interval T_{cf}. In the former case, the algorithm uses the pair of switch positions (i.e., voltage SVs) and the corresponding current errors of i_q and i_d which are calculated at the time instants t_{z} and T_{cf}, respectively. In the latter case, due to $t_{\mathrm{z}} = 0$, the first current term of (6.5) for $\boldsymbol{i}_{dq}$ uses the same current error as the second term at T_{cf} in order to enable a comparable cost, i.e., $\boldsymbol{i}_{dq,\mathrm{t_z}} = \boldsymbol{i}_{dq,\mathrm{T_{cf}}}$. The cost function (6.5) uses the squared ℓ_2-norm for error tracking, both to avoid any closed-loop stability issues and to enable performance

Algorithm 2 Inductance-based VSP^{2}CC

function $\bar{\boldsymbol{u}}_{\mathrm{ABC,CON}}, t_{\mathrm{Z,CON}} = L$-VSP^{2}CC$(\boldsymbol{i}^*_{dq}, \boldsymbol{i}_{dq}, \varphi, N_{\mathrm{p}}, \omega_{\mathrm{el}}, V_{\mathrm{dc}})$
  $\boldsymbol{i}_{dq}(k-1) \leftarrow$ read dq-transformation
  $\boldsymbol{i}_{dq}(k) \leftarrow$ predict using (2.73) with t_{z} and T_{cf} ▷ delay time compensation
  $\boldsymbol{v}_{dq,\mathrm{db}}(k) \leftarrow$ dead-beat solution using (5.1) ▷ search space reduction
  $\gamma_{\mathrm{db}}(k) \leftarrow \angle \boldsymbol{v}_{dq,\mathrm{db}}(k)$ using (5.2) ▷ sector of candidate $\boldsymbol{v}_{dq}(k)$
  for $\ell = 1, \ldots, N_{\mathrm{p}}$ **do**
    $\boldsymbol{v}_{dq}(k+\ell-1) \leftarrow$ based on $\varphi(k+\ell-1)$, $V_{\mathrm{dc}}(k)$ ▷ three candidate SVs based on $\gamma_{\mathrm{db}}(k)$
    for $j = 1, \ldots, 3$ **do** ▷ one SV within T_{cf}
      $\boldsymbol{i}_{dq,j}(k+\ell) \leftarrow$ predict using (2.73)
      $||\boldsymbol{i}_{dq,j}(k+\ell)|| \leftarrow \boldsymbol{i}_{dq,j}(k+\ell)$ using (6.6) ▷ current constraint
      if $\ell = 1$ **then**
        $\boldsymbol{m}_j(k+\ell) \leftarrow$ using (6.1) ▷ $\boldsymbol{i}_{dq}$ gradients
      end if
    end for
    if $\ell = 1$ **then** ▷ VSP for $N_{\mathrm{p}} = 1$
      for $n_1 = 1, \ldots, 3$ **do**
        for $n_2 = 1, \ldots, 3$ **do**
          if $n_1 = n_2$ **then** ▷ one SV within T_{cf}
            $\boldsymbol{i}_{dq,n_1,n_2}(k+\ell) \leftarrow \boldsymbol{i}_{dq,j}(k+\ell)$
          else ▷ VSP for two SVs
            $t_{\mathrm{z},n_1,n_2}(k+\ell) \leftarrow$ (6.4) ▷ VSP
            $\boldsymbol{i}_{dq,n_1,n_2}(k+\ell) \leftarrow$ predict with (2.73)
            $||\boldsymbol{i}_{dq,n_1,n_2}(k+\ell)|| \leftarrow \boldsymbol{i}_{dq,n_1,n_2}(k+\ell)$ using (6.6) ▷ current constraint
          end if
        end for
      end for
    end if
    $\bar{\boldsymbol{u}}_{abc,\mathrm{con}}(k), t_{\mathrm{z,con}}(k) \leftarrow$ solve (6.8)
  end for
end function

[5] Theoretically, a soft constraint is to be preferred since it can avoid feasibility problems when solving (6.5).

improvements [KGK18], even if this increases the computational load. Finally, with the cost function (6.5) the L-VSP^{2}CC problem is stated as

$$\begin{aligned} &\underset{\bar{\boldsymbol{u}}_{abc} \in \mathbb{U}}{\text{minimize}} \;\; J \text{ (see (6.5))} \\ &\text{subject to} \;\; (2.73),\; (6.6)\,. \end{aligned} \tag{6.8}$$

Solving (6.8) yields the optimal switch position(s) $\bar{\boldsymbol{u}}_{abc,\text{con}}$ which is (are) to be applied at the corresponding optimal switching time instant(s), i.e., $t = 0$ and/or $t_{\text{z,con}}$. Note that according to the receding horizon policy, the elements of the switching sequence $\boldsymbol{U}_{\text{con}}$ that correspond to the predictions steps $N_\text{p} \geq 2$ are discarded. In doing so, feedback is provided and a degree of robustness to system uncertainties is achieved [RM09]. The proposed method is summarized in the pseudo Algorithm 2.

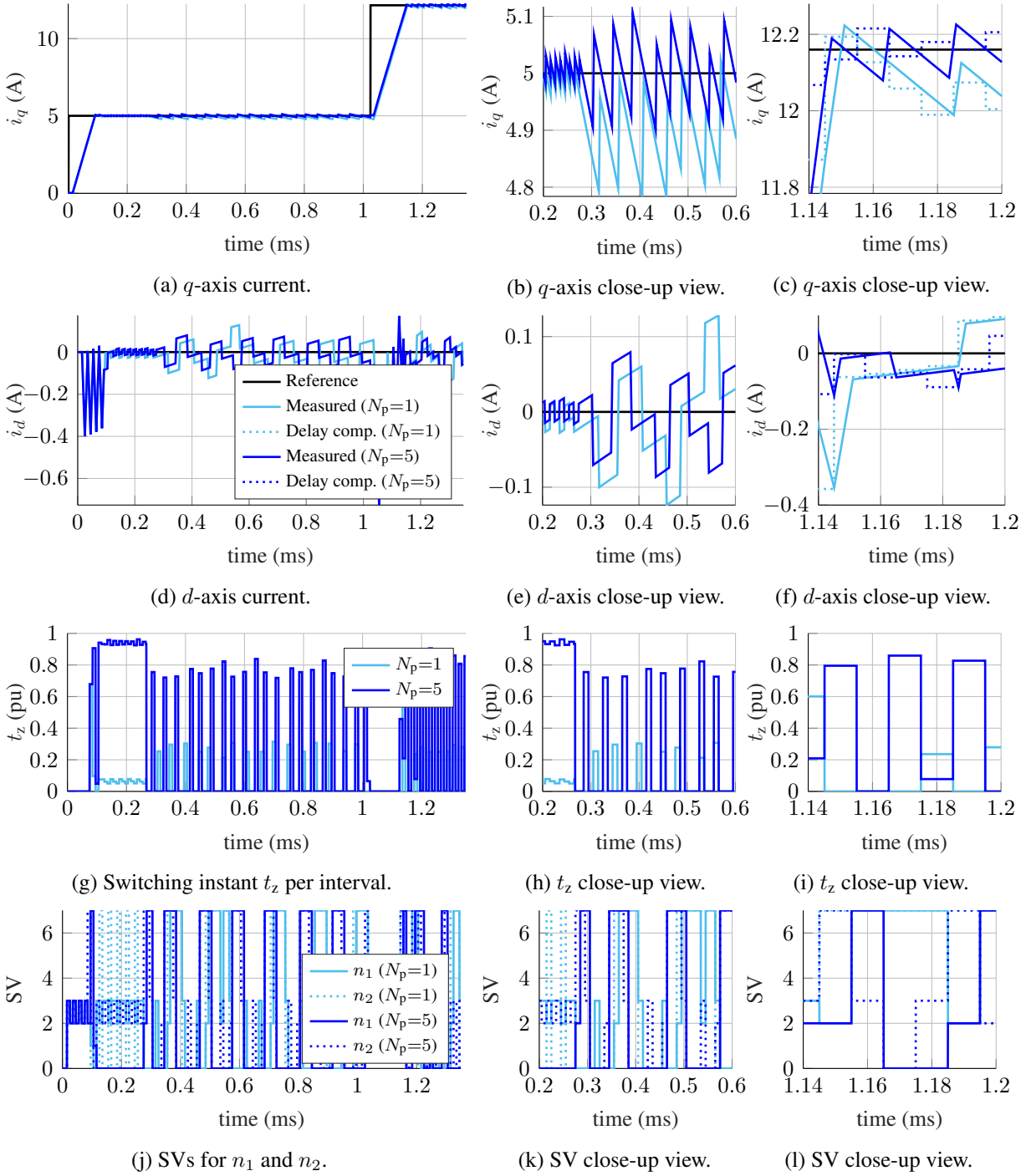
6.3 Performance Evaluation

In the following, the proposed method is evaluated by using simulation and experimental results. Here, λ_u is adjusted for the conventional FCS-MPC and VSP^{2}CC approaches to allow a comparison—in terms of current THD and ripple—at almost equal average f_sw.

The first evaluation in Fig. 6.3 shows simulation results for the currents i_d and i_q—both for a horizon of $N_\text{p} = 1$ and $N_\text{p} = 5$—during start-up for motor M1, see Table A.3. In addition, it shows t_z normalized with respect to the interval T_cf, which describes the application time of SV_n1 in each discrete time step. Finally, it also shows the corresponding selected SVs, i.e., switch positions. At the beginning—$t = 0{-}0.27\,\text{ms}$—$\lambda_\text{u} = 0.001$ is applied, thus switching is only minimally penalized. Afterwards $\lambda_\text{u} = 0.018$ for $N_\text{p} = 1$ and $\lambda_\text{u} = 0.045$ for $N_\text{p} = 5$ is applied, which leads to $f_\text{sw} \approx 20\,\text{kHz}$. As can be seen, for both reference current steps, the algorithm decides to use only one or two active SV(s) to achieve the fastest possible response.

Moreover, in steady-state operation, the ripple is minimized by using two SVs (one active and one zero) or by applying just a single zero SV. This case occurs, e.g., for $t = 1.14{-}1.2\,\text{ms}$, where a zero SV is applied after a VSP was chosen. More precisely, an active SV is applied up to $t = 1.15\,\text{ms}$, where a VSP first retains the active vector, followed by a zero vector for the remaining interval, which is also kept for the following control interval. As a result, the number of switching transitions is minimized while keeping the current ripple acceptable. With a further increasing λ_u the advantage becomes even more apparent. However, it can be noticed, that for $N_\text{p} = 1$ a steady-state deviation exists. As experiments have shown, this effect occurs mainly at a low modulation index. This is due to the short horizon, where only the error area—see Fig. 6.2—for the corresponding control interval is minimized. Thus, it is not taken into

account that—if $\lambda_u > 0$—zero SVs are present in the following intervals, which would cause a steady-state deviation if averaged over several intervals. For $N_p = 5$ this deviation does not occur. In any case, already with $N_p > 1$, e.g., $N_p = 2$, this deviation is no longer observed for the considered small drives.



(a) q-axis current. (b) q-axis close-up view. (c) q-axis close-up view.

(d) d-axis current. (e) d-axis close-up view. (f) d-axis close-up view.

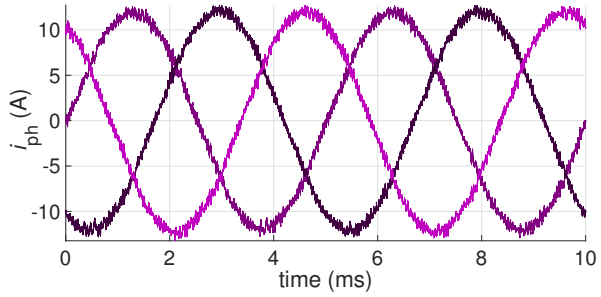
(g) Switching instant t_z per interval. (h) t_z close-up view. (i) t_z close-up view.

(j) SVs for n_1 and n_2. (k) SV close-up view. (l) SV close-up view.

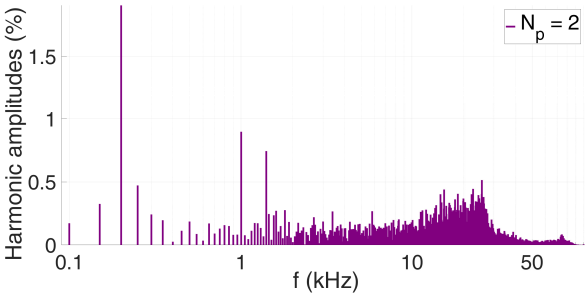
Figure 6.3: Motor M1: Reference current step for i_q^* showing the resulting currents i_d, i_q, switching time t_z and SVs by using L-VSP^{2}CC with $N_p = 1$ and $N_p = 5$ for different λ_u with $n_m = 200$ rpm (simulation).

6.3.1 Steady-State Performance

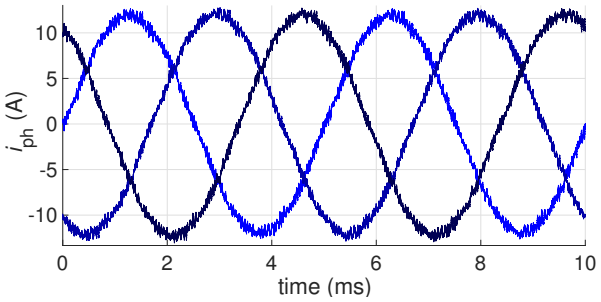
So far, all previous VSP^{2}C(T)C methods, e.g., [Kar13, KSK$^+$14, SKT$^+$13, SKK$^+$15, KAK18, AKMK15], were only on a simulation level or implemented on the processor, thus struggling with the computational load and therefore show only short prediction horizons and/or small sampling frequencies. In contrast, the proposed algorithm is experimentally validated by fully implementing the control loop in an FPGA using the platform described in Chapter 3. This allows for longer prediction horizons and a sampling frequency of $f_{\mathrm{cf}} = 100\,\mathrm{kHz}$. The test bench is shown in Fig. 3.4. At the example of a two-step horizon ($N_{\mathrm{p}} = 2$) the drive performance is evaluated in the following, similar to Section 2.6. Fig. 6.4 shows the phase currents and phase



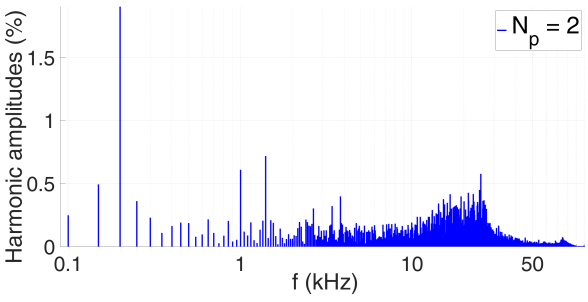
(a) Three-phase stator current for conventional FCS-MPC with $N_{\mathrm{p}} = 2$, $\lambda_{\mathrm{u}} = 0.45$, $f_{\mathrm{sw}} \approx 12.2\,\mathrm{kHz}$.



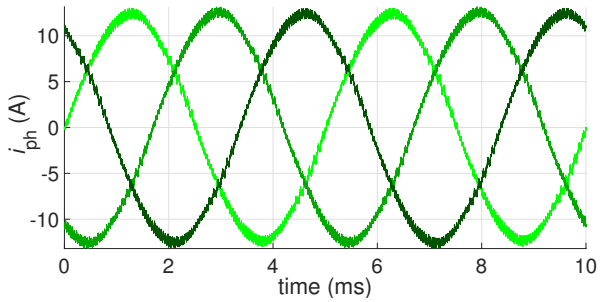
(b) Stator current spectrum for conventional FCS-MPC with $I_{\mathrm{THD}} = 4.41\,\%$.



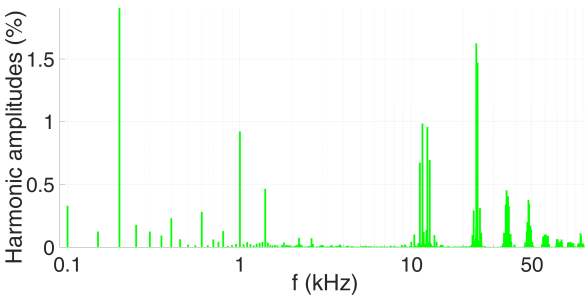
(c) Three-phase stator current for L-VSP^{2}CC with $N_{\mathrm{p}} = 2$, $\lambda_{\mathrm{u}} = 0.44$, $f_{\mathrm{sw}} \approx 12.0\,\mathrm{kHz}$.



(d) Stator current spectrum for L-VSP^{2}CC with $I_{\mathrm{THD}} = 4.39\,\%$.

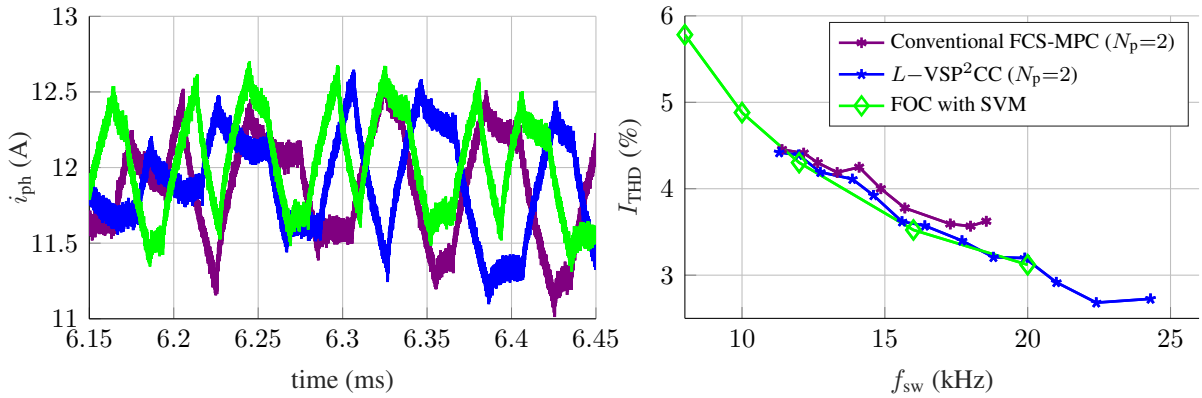


(e) Three-phase stator current for FOC and SVM with $f_{\mathrm{sw}} = 12.0\,\mathrm{kHz}$.



(f) Stator current spectrum for FOC and SVM with $I_{\mathrm{THD}} = 4.34\,\%$.

Figure 6.4: Motor M1: Three-phase stator current and spectrum for $i_d^* = 0\,\mathrm{A}$, $i_q^* = 12.16\,\mathrm{A}$, $n_{\mathrm{m}} = 3000\,\mathrm{rpm}$ with $f_{\mathrm{sw}} \approx 12.0\,\mathrm{kHz}$ (experimental).



(a) Stator current ripple comparison for $f_{sw} \approx 12.0$ kHz.

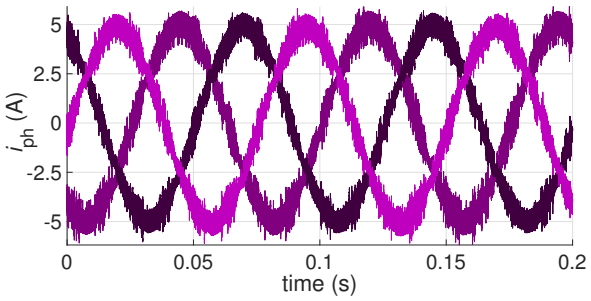
(b) Trade-off between I_{THD} and f_{sw}.

Figure 6.5: Motor M1: Stator current ripple and I_{THD} with $i_d^* = 0$ A, $i_q^* = 12.16$ A, $n_m = 3000$ rpm and $N_p = 2$ (experimental).

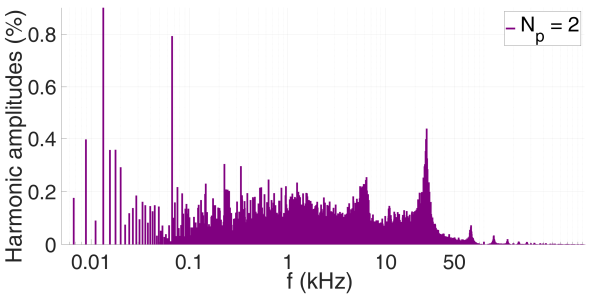
current spectrum for the proposed $L-$VSP^{2}CC with dead-beat preselection in comparison to the conventional FCS-MPC and FOC using SVM at the rated operating point. More precisely, for $n_m = 3000$ rpm, $i_q^* = 12.16$ A and $i_d^* = 0$ A, where $I_N^* = 8.6$ A is the nominal current for motor M1, see Table A.3. As expected, FCS-MPC and L-VSP^{2}CC have slightly different average f_{sw} and thereby produce different current ripples depending on the operating point. Hence, λ_u is adjusted for both MPC approaches to achieve a comparable average f_{sw} and thus a fair comparison of all three methods of interest. The chosen average switching frequency is $f_{sw} \approx 12$ kHz. The spectrum is calculated over 20 fundamental periods. From the comparison at high speed and current, i.e., high modulation index, it can be concluded that the conventional FCS-MPC and L-VSP^{2}CC achieve a comparable current ripple and THD, see also Fig. 6.5.

However, the main advantage of the proposed method becomes apparent as soon as a low modulation index is required. In the partial load range, FCS-MPC uses active SVs for the entire control interval, which results in an unnecessarily high current ripple due to the limited granularity for the switching instant. Fig. 6.6 shows the phase currents and phase current spectrum for $n_m = 200$ rpm, $i_q^* = 5.0$ A and $i_d^* = 0$ A with an average switching frequency of $f_{sw} \approx 10$ kHz. From the comparison at low speed and current, i.e., low modulation index, it can be concluded that reduced current ripple and lower current THD can be achieved by using L-VSP^{2}CC instead of the conventional FCS-MPC at almost the same average f_{sw}, see also Fig. 6.7.

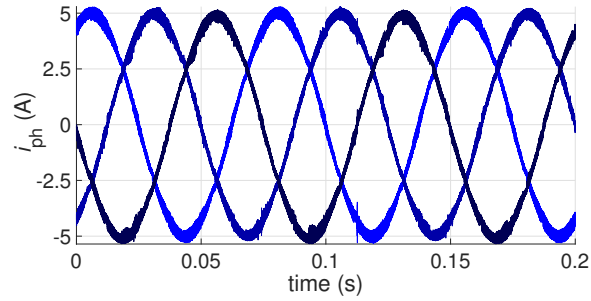
In general, at high speed or for high reference currents (in contrast to low speed and/or for small reference currents) a higher voltage margin is required. This implies that with direct MPC (FCS-MPC and VSP^{2}CC), more active SVs are used, whereas zero SVs are less frequently utilized. The other way around, for low speeds and/or small reference currents, where zero vectors are mostly used, the advantage of VSP^{2}CC is obvious, since a VSP, i.e., a time instant within the control interval at which the inverter switches change, is chosen.



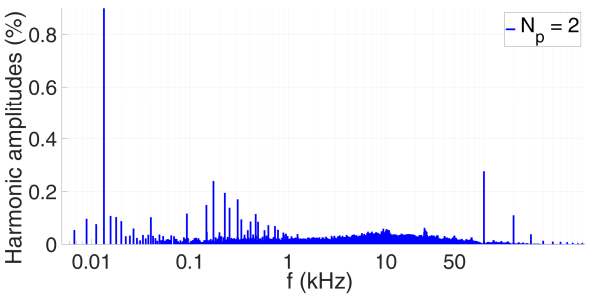
(a) Three-phase stator current for conventional FCS-MPC with $N_\mathrm{p} = 2$, $\lambda_\mathrm{u} = 0.253$, $f_\mathrm{sw} \approx 10.0\,\mathrm{kHz}$.



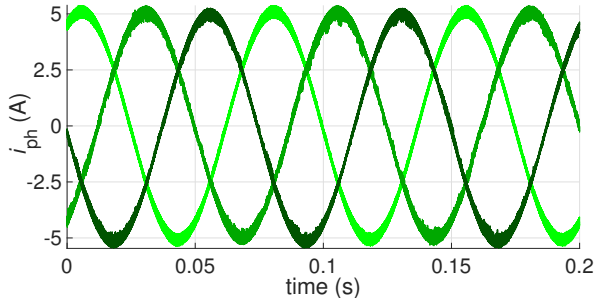
(b) Stator current spectrum for conventional FCS-MPC with $I_\mathrm{THD} = 8.77\,\%$.



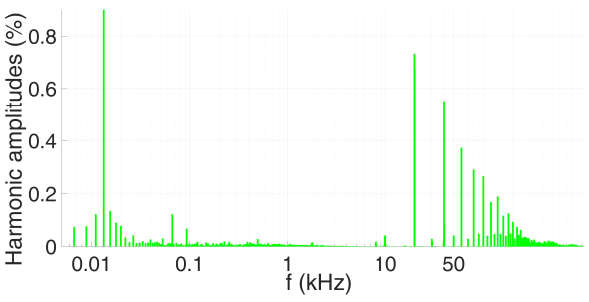
(c) Three-phase stator current for L-VSP^{2}CC with $N_\mathrm{p} = 2$, $\lambda_\mathrm{u} = 0.07$, $f_\mathrm{sw} \approx 10.1\,\mathrm{kHz}$.



(d) Stator current spectrum for L-VSP^{2}CC with $I_\mathrm{THD} = 2.67\,\%$.



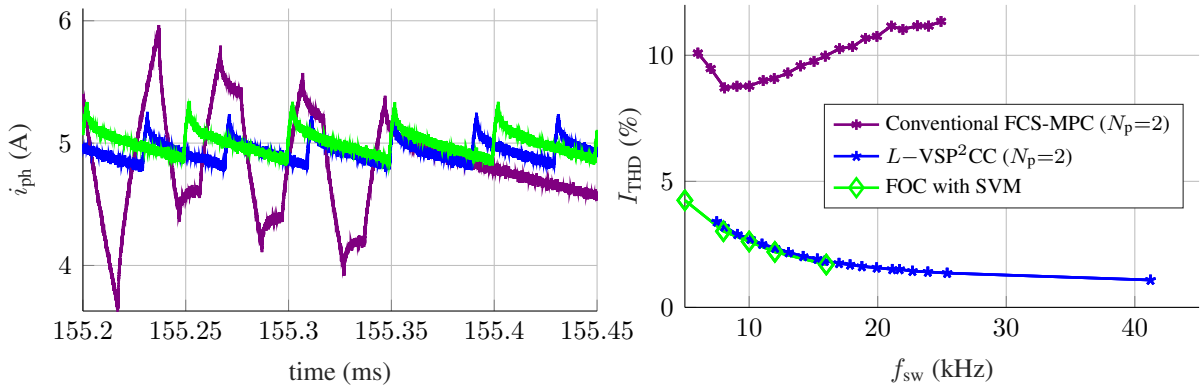
(e) Three-phase stator current for FOC and SVM with $f_\mathrm{sw} = 10.0\,\mathrm{kHz}$.



(f) Stator current spectrum for FOC and SVM with $I_\mathrm{THD} = 2.65\,\%$.

Figure 6.6: Motor M1: Three-phase stator current and spectrum for $i_d^* = 0\,\mathrm{A}$, $i_q^* = 5.0\,\mathrm{A}$, $n_\mathrm{m} = 200\,\mathrm{rpm}$ with $f_\mathrm{sw} \approx 10.0\,\mathrm{kHz}$ (experimental).

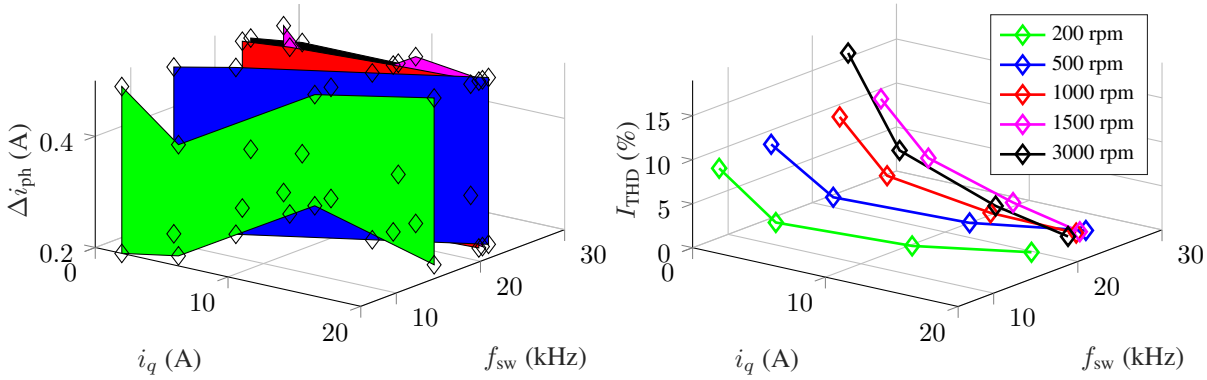
For the proposed cost function, it can be further assumed that when a constant λ_u is specified, an approximately constant current ripple is achieved independent of the speed and reference current, see Fig. 6.8. Here, exemplarily shown for $\lambda_\mathrm{u} = 0.1$, the ripple keeps between $0.2 - 0.5\,\mathrm{A}$. Thus, the algorithm decides to switch less at lower speeds. At higher speeds, where it is also necessary, there are more switching operations on average. Finally, on the voltage limit—for a high modulation index—the controller switches the inverter less frequently to fully utilize the dc-link voltage (up to six-step operation). Even though a fixed switching frequency is required for grid applications due to the filter specification, a switching frequency dependent on the operating point (mainly on the speed)—thus varying—seems to be naturally correct for electrical drives. With respect to VSP^{2}CC, where a similar THD is obtained compared to FOC, this can



(a) Stator current ripple comparison for $f_{sw} \approx 10.0$ kHz.

(b) Trade-off between I_{THD} and f_{sw}.

Figure 6.7: Motor M1: Stator current ripple and I_{THD} for $i_d^* = 0$ A, $i_q^* = 5.0$ A, $n_m = 200$ rpm and $N_p = 2$ (experimental).



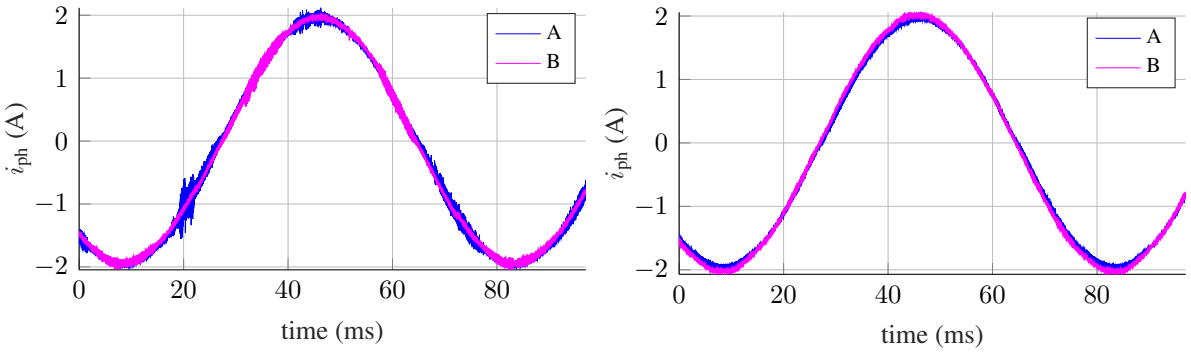
(a) Resulting range of stator current ripple and f_{sw}.

(b) Resulting I_{THD} and f_{sw}.

Figure 6.8: Motor M1: Stator current ripple and I_{THD} using L-VSP^{2}CC with a constant λ_u, here $\lambda_u = 0.1$, while varying i_q^* and n_m where $i_d^* = 0$ A and $N_p = 2$ (experimental).

be considered as an advantage. It is also conceivable that a characteristic curve or map can be stored for λ_u to adapt the permissible current ripple (or f_{sw}) as a function of the operating point.

With regard to the weighting factor λ_u, as discussed in [KG20], its value can significantly affect the system performance. For VSP^{2}CC, if $\lambda_u = 0$ an unnecessary high current ripple results, see Fig. 6.9(a). As can be observed, an increased current ripple occurs in every second sector. The reason for this relates to the implementation of the MPC algorithm, and specifically the evaluation order of the voltage SVs. If switching transitions are not penalized the zero voltage SV evaluated first (i.e., $\boldsymbol{v}_0$) is the one to be always chosen. Since both zero voltage SVs yield the same predicted current gradient, the associated cost of the current error in (6.5) is the same. Hence, if a zero voltage SV yields the minimum cost among all SVs, then the one evaluated first is the solution to (6.8), and thus the corresponding position is the one applied to the inverter. This—depending on the sector—may lead to unnecessary switching, and thus deadtime for the interlock, eventually causing an increased current ripple.



(a) Increased current ripple. "A": Sectors II, IV, VI, when $\boldsymbol{v}_0$ is evaluated first; "B": Sectors I, III, V, when $\boldsymbol{v}_7$ is evaluated first, $\lambda_{\mathrm{u}} = 0$, $t_d = 500\,\mathrm{ns}$, $f_{\mathrm{sw}} \approx 46.7\,\mathrm{kHz}$.

(b) Reduced current ripple. "A": $\lambda_{\mathrm{u}} = 0.001$, $t_d = 500\,\mathrm{ns}$, $f_{\mathrm{sw}} \approx 42.5\,\mathrm{kHz}$; "B": $\lambda_{\mathrm{u}} = 0$, $t_d = 200\,\mathrm{ns}$, $f_{\mathrm{sw}} \approx 49.5\,\mathrm{kHz}$.

Figure 6.9: Motor M3: Single-phase stator current with $i_d^* = 0\,\mathrm{A}$, $i_q^* = 2\,\mathrm{A}$ at $n_{\mathrm{m}} = 200\,\mathrm{rpm}$ using L-VSP^{2}CC with $N_{\mathrm{p}} = 2$ (experimental).

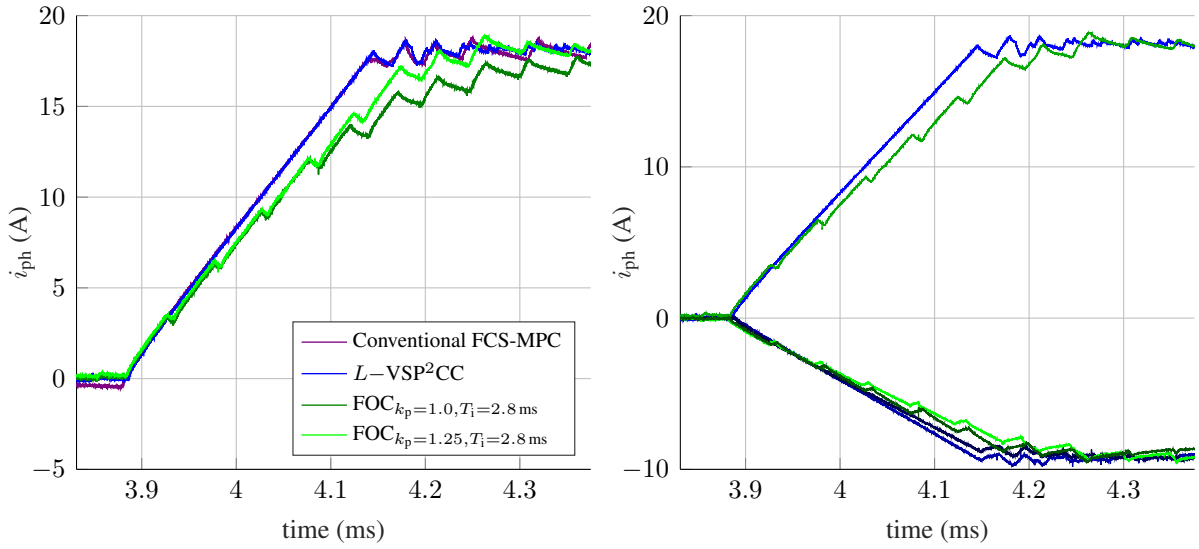
In contrast to this, SVM addresses this issue by appropriately changing the order with which the zero and active voltage SVs are applied. For example, if the reference voltage space vector $\boldsymbol{v}_{dq}^*$ is in the first sector, the sequence of SVs applied within one modulation period is $\boldsymbol{v}_0$–$\boldsymbol{v}_1$–$\boldsymbol{v}_2$–$\boldsymbol{v}_7$–$\boldsymbol{v}_2$–$\boldsymbol{v}_1$–$\boldsymbol{v}_0$. If it is in the second sector, the sequence becomes $\boldsymbol{v}_0$–$\boldsymbol{v}_3$–$\boldsymbol{v}_2$–$\boldsymbol{v}_7$–$\boldsymbol{v}_2$–$\boldsymbol{v}_3$–$\boldsymbol{v}_0$, and so on. Thus, SVM has an inherent mechanism for reducing the current harmonics by changing only one switch position when a new voltage SV is applied. Such a behavior, however, can be achieved with direct MPC provided that $\lambda_{\mathrm{u}} > 0$. To confirm the above analysis the following evaluation is made:

- If $\boldsymbol{v}_0$ is evaluated first, the current ripple is increased in sectors II, IV, and VI, as shown with "A" in Fig. 6.9(a).
- If $\boldsymbol{v}_7$ is evaluated first, the current ripple is increased in sectors I, III, and V, as shown with "B" in Fig. 6.9(a).
- If the control effort in (6.5) is at least minimally penalized, e.g., by setting $\lambda_{\mathrm{u}} = 0.0001$, the current ripple is approximately constant in all sectors, regardless of the evaluation order of the voltage SVs, as shown with "A" in Fig. 6.9(b).
- If the interlock time t_d is decreased, e.g., from $500\,\mathrm{ns}$ to $200\,\mathrm{ns}$, the current ripple reduces even if the control effort is not penalized, as shown with "B" in Fig. 6.9(b).

Based on the above observations, it can be concluded that when $\lambda_{\mathrm{u}} = 0$ the current ripple is higher even if the interlock time is reduced. On the other hand, $\lambda_{\mathrm{u}} > 0$ not only allows one to decrease the switching frequency—and, in effect, the switching power losses—but also it improves the current quality, which can be translated into lower current THD, and thus lower thermal losses. Hence, penalizing the control effort is the preferred choice, see also [KG20, Section IV].

6.3.2 Transient Performance

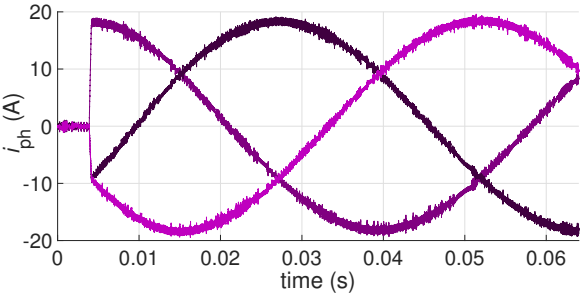
Since a torque measuring shaft is usually not able to accurately describe the transient behavior of the torque, i.e., the update rate of the torque signal of the measuring shaft is only 1 ms [Pra], no torque measuring values are shown. Instead, Fig. 6.11 shows the phase currents for the proposed L-VSP^{2}CC in comparison to the conventional FCS-MPC and FOC during transients. Fig. 6.11 (left side) shows the current during a reference current step from $i_q^* = 0$ A to $i_q^* = 18.24$ A (which is $1.5I_\mathrm{N}$) for a constant speed and Fig. 6.11 (right side) the phase currents for a speed transient generated by the load machine. Fig. 6.10 shows a close-up view of Fig. 6.11 (left side) for the single and three phase currents. Here, both DMPC approaches show the maximum achievable dynamics by using only active SVs. FOC, adjusted according to the modulus optimum, requires a trade-off between fast settling-time and low overshoots. However, as can be seen, for a certain time—even if it is minimal—a zero SV is used, since the switching sequence is fixed for SVM, see Section 2.1.2. To improve the transient performance, i.e., to reduce (avoid) the turn-on time of the zero SVs and thus to exploit the available voltage margin more effectively, it is common practice to change the pulse pattern depending on the operating point, e.g., alternating between continuous SVM at low speed (i.e., low modulation index up to $2/\sqrt{3}$) and discontinuous SVM or switching with the fundamental frequency at high speed (i.e., high modulation index up to $4/\pi$). The latter would be able to use the same dc-link voltage, i.e., $\frac{2}{3}V_\mathrm{dc}$, similar to the two DMPC approaches. However, it is not straightforward to define a suitable criterion on the basis of which the pulse patterns are changed. For this reason, and also because it is rarely used for small drives, only continuous SVM is compared.



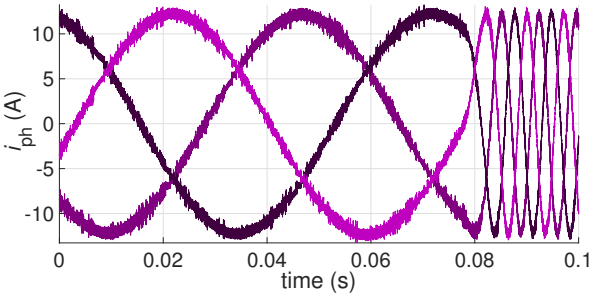
(a) Single-phase stator current for comparison of switching operations.

(b) Three-phase stator current for L-VSP^{2}CC and FOC with $k_\mathrm{p} = 1.25$, $T_\mathrm{i} = 3.6$ ms.

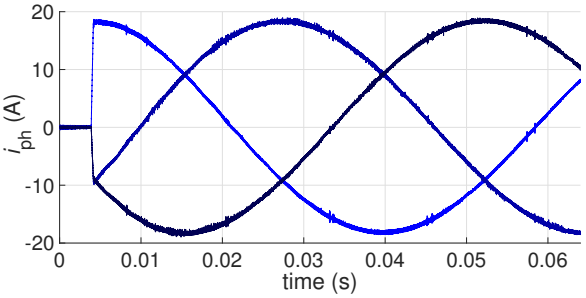
Figure 6.10: Motor M1: Stator current during transient operation for a reference step from $i_q^* = 0$ A to $i_q^* = 18.24$ A with $i_d^* = 0$ A at $n_\mathrm{m} = 200$ rpm and $f_\mathrm{sw} \approx 10$ kHz with $N_\mathrm{p} = 2$ as close-up view from Fig. 6.11 (left side) (experimental).



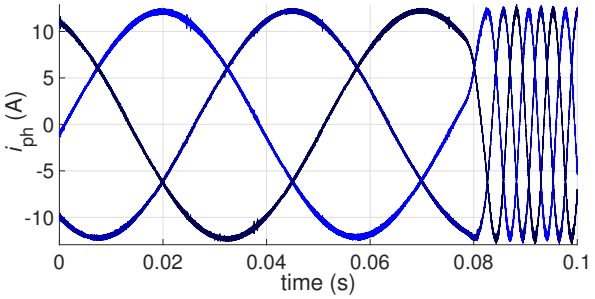
(a) Current step for FCS-MPC with N_p=2, λ_u=0.253.



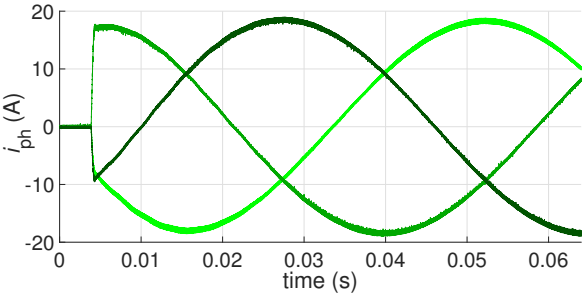
(b) Speed change for FCS-MPC with N_p=2, λ_u=0.253.



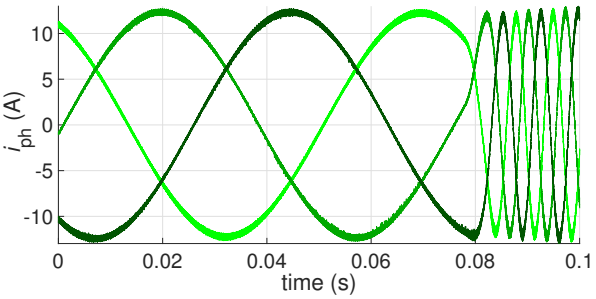
(c) Current step for L-VSP^{2}CC with N_p=2, λ_u=0.07.



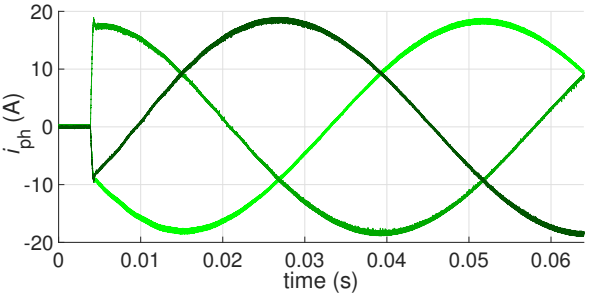
(d) Speed change for L-VSP^{2}CC with N_p=2, λ_u=0.07.



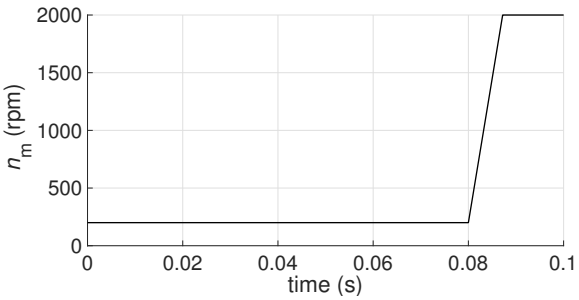
(e) Current step for FOC with k_p=1.0, T_i=2.8 ms.



(f) Speed change for FOC with k_p=1.0, T_i=2.8 ms.

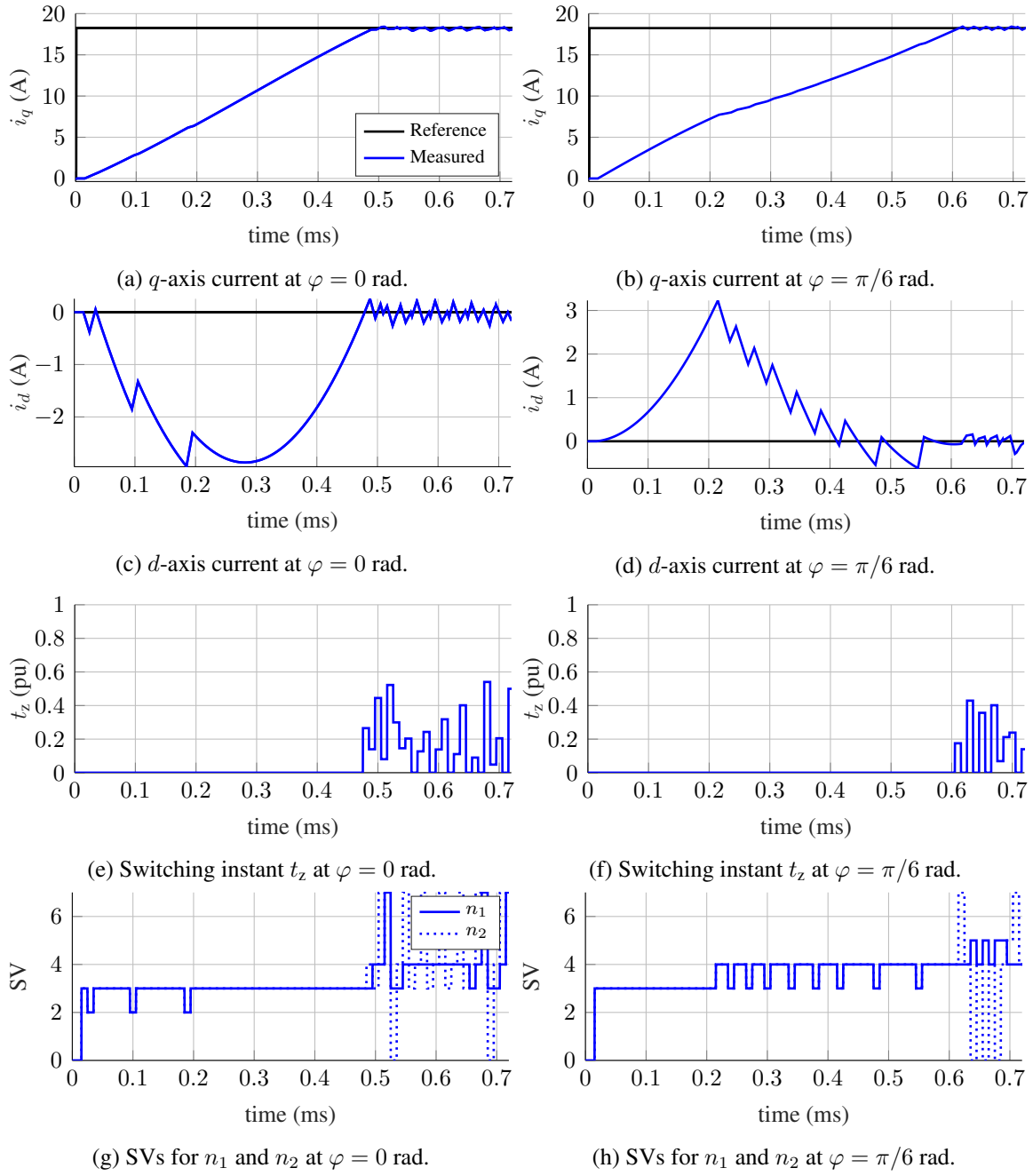


(g) Current step for FOC with k_p=1.25, T_i=2.8 ms.



(h) Reference speed change n_m=200 to 2000 rpm.

Figure 6.11: Motor M1: Three-phase stator current showing a current reference step (left side) from $i_q^* = 0.0$ A to $i_q^* = 18.24$ A with $i_d^* = 0$ A, $n_m = 200$ rpm and a load speed change (right side) from $n_m = 200$ to 2000 rpm with $i_q^* = 12.16$ A, $i_d^* = 0$ A. $f_{sw} \approx 10.0$ kHz before the transient occurs and $N_p = 2$ (experimental).



(a) q-axis current at $\varphi = 0$ rad.

(b) q-axis current at $\varphi = \pi/6$ rad.

(c) d-axis current at $\varphi = 0$ rad.

(d) d-axis current at $\varphi = \pi/6$ rad.

(e) Switching instant t_z at $\varphi = 0$ rad.

(f) Switching instant t_z at $\varphi = \pi/6$ rad.

(g) SVs for n_1 and n_2 at $\varphi = 0$ rad.

(h) SVs for n_1 and n_2 at $\varphi = \pi/6$ rad.

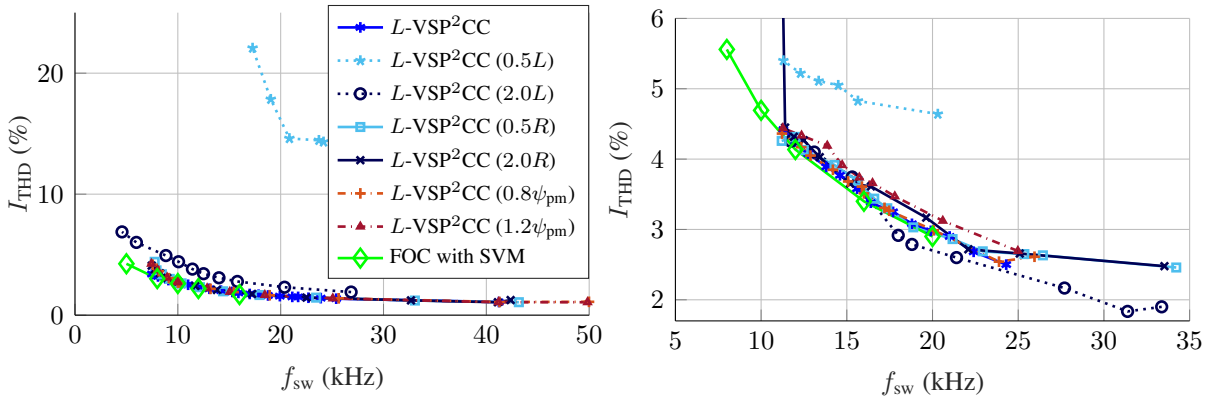
Figure 6.12: Motor M1: Reference current step with $i_q^* = 18.24$ A showing the resulting currents i_d, i_q, switching time t_z and SVs by using L-VSP^{2}CC with $N_p = 2$, $\lambda_u = 0.043$, $n_m = 3000$ rpm when starting at different electrical rotor angles φ (simulation).

Another reason for the high dynamics with VSP^{2}CC—which can also be observed with FCS-MPC—is the MIMO optimization, see Fig. 6.12. More specifically, if, e.g., the reference current in the $q-$axis is increased, the error in this axis dominates the cost function. Hence, only discrete voltage SVs, i.e., active SVs, are selected first, which minimize the $q-$axis current error as much as possible. This results in an increased current error in the $d-$axis, i.e., this current de-

viation is accepted. Depending on the position in the hexagon—see Fig. 2.3(b)—the deviation is even advantageous in transients. In Fig. 6.12 the explained effect is demonstrated for $\varphi = 0$ rad, i.e., when the q-axis is between two discrete SVs, and for $\varphi = \frac{\pi}{6}$ rad, i.e., when the q-axis is aligned with $\boldsymbol{v}_3$. All other sectors behave similar. Consequently, two extreme conditions must be distinguished. First, the d-axis or, second, the q-axis is aligned with a discrete SV.
For the first case, e.g., at $\varphi = 0$ rad, the cost function almost always chooses $\boldsymbol{v}_3$, resulting in a negative i_d that leads to field weakening, i.e., it reduces the induced voltage $\omega_{\text{el}}(L_d i_d + \psi_{\text{pm}})$. This allows more voltage, i.e., v_q, to reach i_q^* faster.
In the second case, e.g., at $\varphi = \frac{\pi}{6}$ rad, a positive i_d results. However, since the q-axis is aligned with a discrete SV, i.e., $\boldsymbol{v}_3$ contributes only to v_q, it is still the best choice to minimize the cost function and reach i_q^*. However, this effect is mainly visible at higher speeds, i.e., at higher induced voltage, and is not as pronounced at low speeds as shown for $\varphi = 0$ rad in Fig. 6.3.
In summary, using DMPC, both currents reach their respective references, i.e., steady state, much faster compared to FOC with decoupling network. The latter uses independent SISO PI-based controllers, whereas the d-axis controller does not and cannot know—possible superimposed strategies and feedback paths not included—that if it would allow a temporary error for i_d, this would help the q-axis current, whereby collectively seen the steady state would be reached faster.

6.3.3 Influence of Model and Parameter Inaccuracies

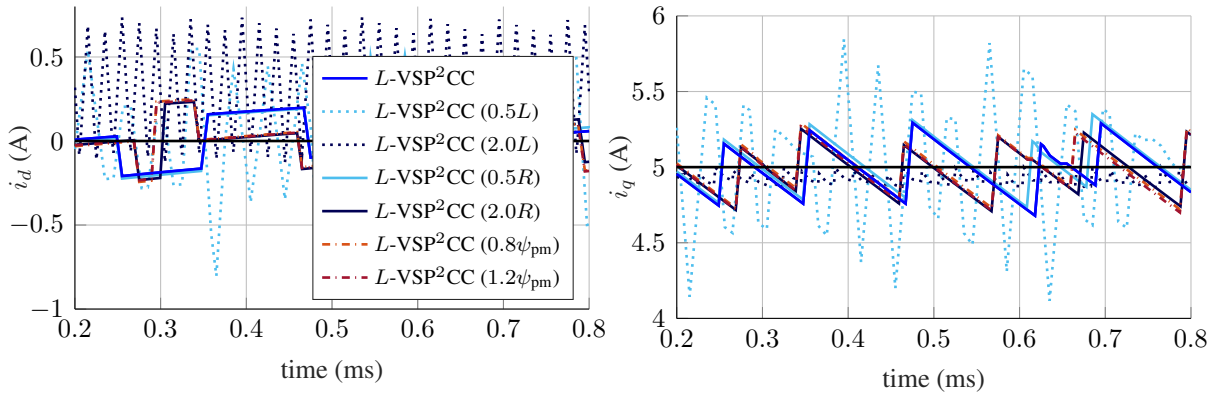
Akin to Section 2.6.3, the parameter sensitivity of the algorithm is evaluated by changing the parameters in the model while those of the machine remain the same. Fig. 6.13—similar to Fig. 2.29—shows the influence of the proposed method regarding to parameter mismatches.
Similar to conventional FCS-MPC, the inductance is the most important parameter as it determines the gradient for the predicted current change. If, e.g., the inductance in the machine is only half of the inductance assumed in the prediction, i.e., $0.5L$ in the figures, an unnecessarily high current ripple will occur. In contrast, an inductance assumed to be too small for the prediction ($2.0L$) reduces the THD, which, however, also leads to a steady-state deviation, since the slope of the predicted current is steeper than in reality.
These effects are also shown for the simulated dq-currents in Figs. 6.14-6.15. Similar to FCS-MPC, at higher speeds, a mismatch in ψ_{pm} results in a steady-state drift. The resistance has an identical behavior on L-VSP^{2}CC as on FCS-MPC, whose mismatch has a marginal effect on the control performance even at rated current. Even though a small steady-state error is also apparent here, see Fig. 6.15.
Not shown but nevertheless with high influence is V_{dc}, which is an integral part—based on



(a) Trade-off for i_d^*=0 A, i_q^*=5.0 A, n_m=200 rpm.

(b) Trade-off for i_d^*=0 A, i_q^*=12.16 A, n_m=3000 rpm.

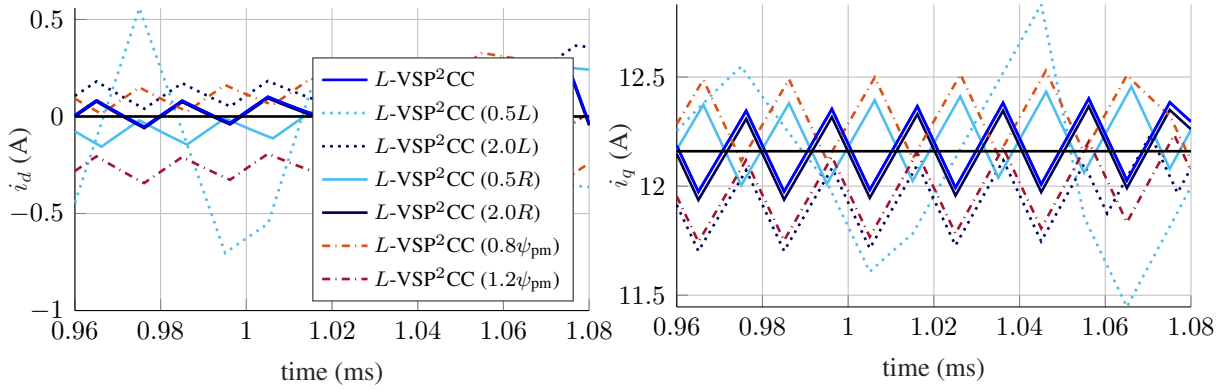
Figure 6.13: Motor M1: Trade-off between stator current I_{THD} and f_{sw} with different parameter mismatches for $N_p = 2$ (experimental).



(a) Steady-state deviation for the $d-$current.

(b) Steady-state deviation for the $q-$current.

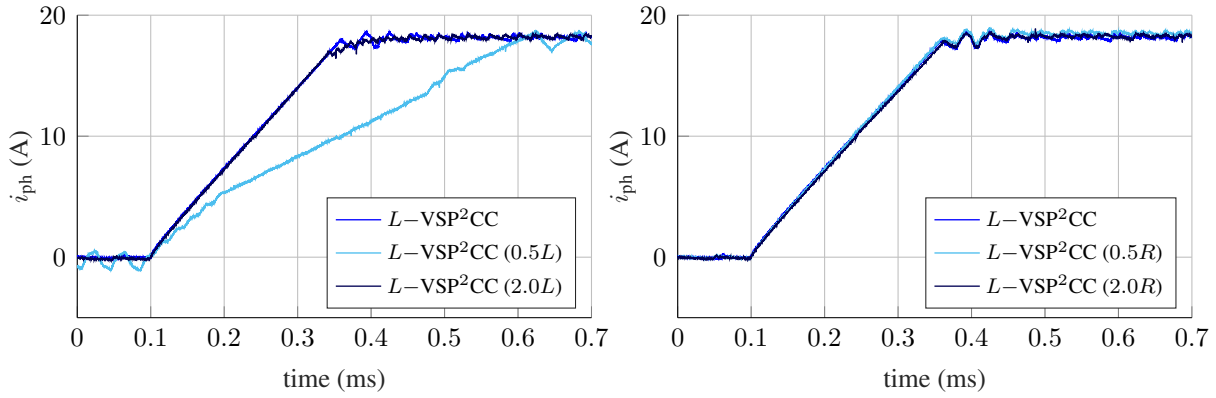
Figure 6.14: Motor M1: Steady-state deviation for $i_d^* = 0$ A, $i_q^* = 5.0$ A, $n_m = 200$ rpm with different parameter mismatches using $N_p = 2$ and a constant $\lambda_u = 0.045$ ($f_{sw} \approx 4.6$ kHz without parameter variations) (simulation).



(a) Steady-state deviation for the $d-$current.

(b) Steady-state deviation for the $q-$current.

Figure 6.15: Motor M1: Steady-state deviation for $i_d^* = 0$ A, $i_q^* = 12.16$ A, $n_m = 3000$ rpm with different parameter mismatches using $N_p = 2$ and a constant $\lambda_u = 0.112$ ($f_{sw} \approx 18.6$ kHz without parameter variations) (simulation).



(a) Parameter mismatch for the inductances.

(b) Parameter mismatch for the resistance.

Figure 6.16: Motor M1: Single-phase stator current during transient operation using L-VSP^{2}CC and different parameter mismatches for a reference step from i_q^*=0 A to i_q^*=18.24 A with i_d^*=0 A, n_{m}=200 rpm, $f_{\mathrm{sw}} \approx 10$ kHz and N_{p}=2 (experimental).

(2.12)—of the different SVs, which in turn are necessary for the current prediction (2.73). An incorrect V_{dc} during prediction can cause a similar behavior as an incorrect inductance. However, since the voltage is measured and can therefore be assumed to be accurate, this will not be discussed further here.

Finally, Fig. 6.16 shows the influence of parameter mismatches on the transient performance of VSP^{2}CC. In contrast to FCS-MPC, the inductance may also have an influence on the transient behavior. In case of $2.0L$ the dynamic range is ideal at first. However, close before reaching the reference, i.e., as soon as a VSP is used, a too flat predicted current gradient leads to a too short application of an active SV, thus voltage-time area is missing. As a result the current in the last section, i.e., $t = 0.35 - 0.4$ ms, of Fig. 6.16(a) approaches the reference only slowly. In case of $0.5L$, suboptimal VSPs create unnecessary switching and thus reduce the dynamic in transients. Although the currents of the other two phases are not plotted in Fig. 6.16(a), they show a current ripple that explains the loss of dynamics. However, this may not be the case if VSP^{2}CC applies one SV for the entire control interval, i.e., if the use of a VSP is not selected during the transient. The latter is shown for comparison, e.g., in Fig. 6.12.
In case of a mismatch for the resistance, i.e., Fig. 6.16(b), the transient performance is not affected, similar to the conventional FCS-MPC in Fig. 2.32(b).

6.4 Final Assessment

The combination of the two performance criteria for PMSMs, i.e., low THD and fast transient response, shows the great advantage of the presented L-VSP^{2}CC approach.

L-VSP^{2}CC enables a high granularity for the switching instants. As a result, the drive performance, as quantified by the current distortions, i.e., current THD, is significantly improved. Compared to FCS-MPC, the current ripple, and thus indirectly the torque ripple, can be significantly reduced depending on the operating point (assuming a comparable average f_{sw}). Compared to FOC, similar performance is achieved. However, regarding noise, vibration, and harshness (NVH) and the possible excitation of resonances, the spread spectrum of L-VSP^{2}CC can even be seen as an advantage over FOC with SVM, see also [Hol94]. Even though the fixed switching frequency of SVM is a prerequisite for grid applications due to the required filter design, this is not mandatory for motor applications. Consequently, due to the spread spectrum—comparable to white noise—a uniform noise of low amplitude is generated. This avoids the fixed high-frequency noise with comparatively higher amplitudes—in the audible range up to $20\,\mathrm{kHz}$—resulting from the modulator, see, e.g., Fig. 6.4.

Moreover, VSP^{2}CC comes with all FCS-MPC advantages, such as direct consideration of constraints, or the possibility to apply only active SVs during transients. The latter allows for the fastest possible dynamic operation of the drive, limited only by the available dc-link voltage margin. As a result, very fast transients are achieved with very short settling times. More specifically, VSP^{2}CC can be executed at a frequency of, e.g., $100\,\mathrm{kHz}$ and thus reacts in transients quite fast, even though it operates the drive system with a much lower switching frequency depending on the operating point and the selected λ_{u}. Such an ability to adapt the switching frequency depending on the operating point (e.g., at different motor speeds) is particularly advantageous for electrical drives, as switching only takes place when required. As a result, unnecessary switching power losses can be avoided over the whole operating range.
FOC, in contrast, makes decisions at a much lower frequency rate, which is at most twice the (constant) switching frequency for asymmetrical sampling or the same for symmetrical sampling, regardless of the operating point. Thus, in transients FOC reacts much slower compared to VSP^{2}CC (assuming a similar f_{sw}), although for FOC the delay times can also be compensated for predictively, see, e.g., [Ric16]. Finally, when using FOC, the PI gains must also be adjusted depending on the operating point.

The above imply that the proposed VSP^{2}CC shifts the control effort from the design to the computational stage. The latter, nevertheless, is simplified thanks to the control platform used, see Chapter 3. Although the industry-oriented first system was chosen for the measurements shown here, the second system—see Fig. A.2 of Appendix A.3—can be used to increase the performance even further, i.e., increase the control frequency—thus the granularity of switching [KG20]—or the prediction horizon.

7 Prediction of the Flux Linkage Behavior

For a highly dynamic control, and especially for MPC, the nonlinear effects of the controlled system, i.e., of the PMSM and inverter, must be taken into account to achieve the best possible performance. Since in Section 6.3.3 the inductance, which can be rather nonlinear due to saturation, was identified as the most critical parameter with the greatest impact on the performance of VSP^{2}CC, a high accuracy for the inductance over the entire operating range should be ensured. As shown in Section 2.6.3, the same necessity applies to conventional FCS-MPC. Thus, it can be generally concluded that the inductance has the greatest influence on the performance of direct MPC. Consequently, this chapter presents an extension of the VSP^{2}CC algorithm explained in Chapter 6, which uses the flux linkage instead of the machine inductances for the stator current prediction. The results were first published in [WKDK20, WKG$^+$21].

All previous DMPC and expecially VSP^{2}CC approaches, when designed for electrical drives, see, e.g., [LK12, KSK$^+$14, SKT$^+$13, ZXL$^+$14, ZBY18, KMG18, WKDK19], use prediction models that rely on knowledge of the (absolute) motor inductances for computing the optimal switch positions and the corresponding VSPs. This might be accurate enough for drives based on induction motors, SPMSMs or IPMSMs operating in the linear region of the magnetic circuit. Since most of the small electrical drives do not leave this linear region, and therefore the nonlinearities are negligible, a prediction can be made with sufficient accuracy.
However, when machines with a strong nonlinear magnetic circuit are of concern, such as most IPMSMs or SynRMs, the prediction model accuracy degrades. Especially for highly utilized IPMSMs, which are becoming increasingly popular due to the associated lower manufacturing costs and weight, it becomes evident that an accurate prediction model is challenging since saturation effects already occur in the nominal range [RGD14]. Such a need becomes even more prominent given that cross-coupling effects are almost always present and cannot be neglected for most IPMSM designs [vvD$^+$03, MTSL08].

Predictive approaches using nonlinear models still hardly exist [Gem15, p. 8], but can be promising for MPC, as pointed out, e.g., in [Pre13, p. 123]. As Schroeder has already recognized, „the effort of implementing a current control in the rotating coordinate system lies in reducing the effects of the coupling dynamics on the current control dynamics to a minimum ac-

ceptable level“ [Sch15, p. 738].[1] For indirect current control methods, i.e., when using a modulator, first predictive approaches for saturated cross-coupled PMSMs are shown, e.g., in [RGD14], [DRB16], [Mey10, p. 71 ff.], [Min13, p. 162 ff.], [Gem15, p. 107 ff.] or [Ric16, p. 61 ff.], which were mainly used to compensate for the delay time—introduced by sampling and calculation times—and for decoupling the PI-based current controllers. A detailed description of these predictive controllers—which were used in combination with PI-based controllers—and the respective nonlinear models are described, e.g., in [Ric16, p. 62], [RGD14] or [Gem15, p. 108]. The latter two focus on accounting for the nonlinearities of the machine in the control algorithm to optimize the control behavior in dynamic situations. Consequently, since the trend goes more and more in the direction of highly utilized synchronous machines—also for small drives—direct current control methods, e.g., the proposed DMPC approach, should be extended in order to include saturation and cross-coupling effects as accurate as reasonable.

In an attempt to improve the modeling accuracy of PMSMs, differential inductances L_{dd}, L_{dq}, L_{qq} and L_{qd} can be used, see Section 2.2.1.2. When using differential inductances for the prediction model of DMPC, the nonlinear voltage equation (2.24) can be used. However, predicting the change in current for each discrete space vector, similar to (2.73) with the absolute inductances, has some drawbacks when using differential inductances. Although these inductances can be used to separate the physical processes, a closed-loop control would require accurate knowledge of all the individual partial derivatives of the flux linkage. Due to their dependency on the changes in current, they are susceptible to noise-contaminated current measurements when identified by the inverter, as explained in Section 4.3.4. Moreover, introducing four (three if symmetry is utilized) three-dimensional maps—as a function of the two current components in d- and q-axis—implies a higher memory requirement, which also implies an increased computational effort for traversing and using them. The latter aspect becomes even more important when considering the real-time implementation of the algorithm, see Section 7.6. In summary, differential inductances are not used to predict the change in current due to the following two disadvantages:

1. In contrast to differential inductances, the flux linkage term directly describes the real machine behavior and is therefore concise, i.e., fewer calculations are required to predict the machine equations, i.e., less complexity.
2. To determine differential inductances, the flux linkage must be derived according to the current, which can be affected by noise-contaminated current measurements. Thus avoiding the derivation is more robust.

[1] Original text: „Daher liegt der Aufwand der Implementierung einer Stromregelung im rotierenden Koordinatensystem darin, die Auswirkungen der Verkopplungsdynamiken auf die Stromregelungsdynamiken auf ein minimal akzeptierbares Maß zu reduzieren.“

A much more resource efficient, intuitive, and mathematically equivalent approach would be to use (2.20) to predict the change in flux linkage. This requires a transfer function from current to the flux linkage and vice versa, i.e., a polynomial function, neural network or map.
Motivated by the argumentation in Section 4.3.4, this chapter proposes the utilization of flux linkage maps—instead of polynomial functions—in the prediction process. In this way, a relatively computationally efficient MPC algorithm can be developed since only two maps are required, while being more intuitive since the flux linkage directly describes the behavior of the machine. Moreover, the proposed MPC algorithm is more robust to system parameter variations because it automatically identifies or adapts the flux linkage in real time and therefore does not require detailed prior knowledge of the machine inductances, see Section 4.3.4.
As a result, the predictions of the future behavior of the PMSM are more accurate, enabling the calculation of more effective VSPs and thus improving the system performance, which is quantified by the THD of the stator current. This is highlighted by the experimental results presented, especially for drive systems with significant nonlinearities.

7.1 Derivation of the Change in Flux Linkage

The change of the flux linkage is described by (2.21). Even if saturation and cross-coupling effects increase through the use of highly utilized synchronous machines (see Section 2.2.1.2), the mentioned equations consider these effects. However, negative manufacturing influences that result in an unequal air gap and thus an unequal air-gap field are difficult to account for, due to the following two reasons:

1. Identifying a flux linkage map, which respects not only the influence of i_d and i_q, but also the electrical rotor angle, implies one current combination at each discrete rotor position. On top of this, an adequate discrete step size for the rotor position granularity is necessary. As can be expected, the measurement of such an exact rotor position during the control process seems to be very complex and probably also very inaccurate.
2. Working in a rotating dq-frame, each harmonic, or at least the dominant ones, must be transformed with the corresponding harmonic frequency. The number of pronounced harmonic frequencies, i.e., the harmonics to be respected, can vary for several PMSMs (see, e.g., motor M4), so a generic approach seems difficult.

Since the identification of the rotor position-dependent influence would be quite challenging for real-time applications and the processing in the control algorithm would be quite computationally intensive—since a four-dimensional map would result for the flux linkage maps—a

small prediction error is accepted. Even if it were possible to identify the position effect in real time, it is questionable whether the accuracy would be good enough to generate additional value for MPC performance. Consequently, the position-dependent asymmetry in the air gap is not modeled, i.e., the last term in (2.21) is neglected similar to [Ric16, p. 25], and therefore not considered in the MPC algorithm. In addition, negative manufacturing influences that lead to an unequal air gap and thus an unequal air-gap field are negligible within a small tolerance band [GT18]. However, as shown in Chapter 8, a generic correction term can deal with the compensation of harmonics that are present due to the design of the machine in order to achieve the smoothest possible current (or torque).

Neglecting the rotor dependency for the flux linkage maps, the voltage equation (2.20) can be rearranged regarding the flux linkage change as

$$\frac{\mathrm{d}\boldsymbol{\psi}_{dq}(t)}{\mathrm{d}t} = \boldsymbol{v}_{dq}(t) - R_{\mathrm{ph}}(t)\boldsymbol{i}_{dq}(t) - \omega_{\mathrm{el}}(t)\boldsymbol{P}\boldsymbol{\psi}_{dq}(t)\,. \tag{7.1}$$

If (7.1) would be discretized with the help of forward Euler discretization—see Section 2.1.3—a prediction of the flux linkage could be done in the same way as in the previous chapters for the current. Thereby it would be assumed that the speed and current at the time step k can be used to predict the slope up to $k+1$, see also [Gey17, p. 155]. In other words, forward Euler is sufficiently accurate as long as the product of speed and the change of flux linkage during one prediction interval is negligible, i.e., $\omega_{\mathrm{el}}(\boldsymbol{\psi}_{dq}(k+1) - \boldsymbol{\psi}_{dq}(k)) \approx \mathbf{0}$. However, this cannot be guaranteed for small electrical drives—which often have a high number of pole pairs and thus high electrical angular speed—in combination with longer calculation intervals T_{cf}. An illustrative explanation of these effect can be found, e.g., in [Ric16, p. 23].

In an attempt to improve the prediction accuracy—in addition to the use of the flux linkage—the prediction equation is adapted. Therefore, an additional modification is added to the forward Euler discretization in order to make the approach more robust at high speeds. Consequently, the voltage equation (2.20) gives [RGD14]

$$\boldsymbol{v}_{dq}(k) = R_{\mathrm{ph}}\boldsymbol{i}_{dq}(k) + \frac{\Delta\boldsymbol{\psi}_{dq}(k+1)}{T_{\mathrm{cf}}} + \frac{1}{2}\omega_{\mathrm{el}}(k)\boldsymbol{P}\Sigma\boldsymbol{\psi}_{dq}(k+1) \tag{7.2}$$

where $\Delta\boldsymbol{\psi}_{dq}(k+1) = \boldsymbol{\psi}_{dq}(k+1) - \boldsymbol{\psi}_{dq}(k)$ and $\Sigma\boldsymbol{\psi}_{dq}(k+1) = \boldsymbol{\psi}_{dq}(k) + \boldsymbol{\psi}_{dq}(k+1)$. The voltage equation can be rearranged in order to get the flux linkage at the end of the discrete interval $(k+1)$ by using

$$\boldsymbol{\psi}_{dq}(k+1) = \boldsymbol{\psi}_{dq}(k) + T_{\mathrm{cf}}\Big(\boldsymbol{v}_{dq}(k) - R_{\mathrm{ph}}\boldsymbol{i}_{dq}(k) - \frac{1}{2}\omega_{\mathrm{el}}(k)\boldsymbol{P}(\boldsymbol{\psi}_{dq}(k) + \boldsymbol{\psi}_{dq}(k+1))\Big). \tag{7.3}$$

Assuming a constant electrical speed ω_{el} as well as temperature for the permanent magnets and the stator resistance R_{ph} during one control interval T_{cf}[2], after some algebraic manipulations shown in Appendix A.2.2, the flux linkage at time step $k+1$ is given by[3]

$$\boldsymbol{\psi}_{dq}(k{+}1) \approx \boldsymbol{\psi}_{dq}(k) + T_{\text{cf}} \frac{\boldsymbol{v}_{dq}(k) - R_{\text{ph}} \boldsymbol{i}_{dq}(k) - \omega_{\text{el}}(k) \boldsymbol{P} \boldsymbol{\psi}_{dq}(k)}{1 + \frac{1}{4} T_{\text{cf}}^2 \omega_{\text{el}}^2(k)} \,. \tag{7.4}$$

As can be seen, the above expression shows that the future behavior of the drive has been decoupled from the machine inductances, adding a high degree of robustness to variations in them. Finally, it should be mentioned that (7.4) is also used for the compensation of the delay time, i.e., the time interval between the measurements occur and the execution of the control action, caused by the real-time system.

7.2 VSP^{2}CC with Flux Linkage-Based Prediction

In contrast to Chapter 6, and actually all other FCS-MPC as well as VSP^{2}CC approaches, the flux linkage is used for the current prediction in the presented method. Consequently, since the stator current is both measured and optimized, a function to map the current into the flux linkage and vice versa is required. To this end, the following expressions are utilized

$$f_{\psi} : \mathbb{R}^2 \to \mathbb{R}^2, (i_d, i_q) \to (\psi_d, \psi_q) \tag{7.5a}$$

$$f_{\psi}^{-1} : \mathbb{R}^2 \to \mathbb{R}^2, (\psi_d, \psi_q) \to (i_d, i_q) \,. \tag{7.5b}$$

Here, f_{ψ} is identified as described in Section 4.3.4. Since f_{ψ} is uniquely invertible for all PMSMs according to Maxwell's 2nd law, f_{ψ}^{-1} can be calculated numerically relatively easily. In doing so, (7.5a) is used to map the measured current $\boldsymbol{i}_{dq}(k-1)$ into the corresponding flux linkage. This procedure is required only once per sampling interval, i.e., after the current measurement. Next, the predicted flux linkage is calculated using (7.4). Subsequently, (7.5b) is used to get $\boldsymbol{i}_{dq}(k+\ell)$, with $\ell = 1, \ldots, N_p$. Thus, the procedure of the reverse mapping is performed after each individual prediction step, since the optimization problem underlying VSP^{2}CC (see Section 7.4) optimizes the current behavior.

[2]The temperature for the permanent magnets and the stator resistance R_{ph} is updated before each prediction process, see Section 4.3, but is kept constant during the prediction.

[3]It has to be mentioned that the adaption in (7.2) is only required at high speeds, as exemplified in Appendix A.2.2. When such extreme conditions are avoided, forward Euler is sufficient and $1 + \frac{1}{4} T_{\text{cf}}^2 \omega_{\text{el}}^2(k) \approx 1$ can be assumed for the denominator.

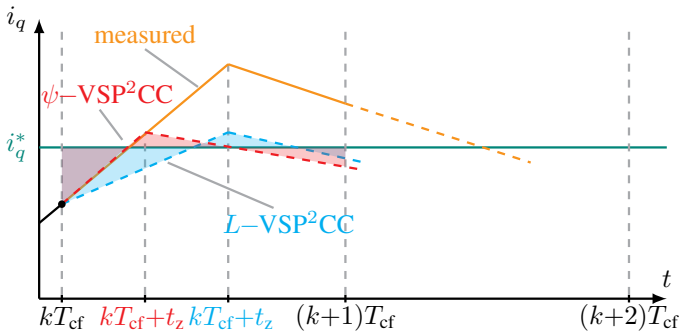


Figure 7.1: Flux linkage- and inductance-based VSPs t_{z} resulting by the intersection of two possible trajectories of the q-component of the stator current $\boldsymbol{i}_{dq}$. The measured current is obtained when L-VSP^{2}CC is used while saturation occurs. For simplicity only the q-component of the current is shown.

Thanks to the utilization of the flux linkage maps and the decoupling of the prediction process from the machine inductances, an improved gradient calculation can be acknowledged, especially when considering nonlinear PMSMs. This is exemplified in Fig. 7.1, where the current trajectory computed with the inductance-based prediction model (2.73) leads to a suboptimal t_{z}, causing an unnecessarily high current ripple compared to the resulting ripple from the proposed method.

More specifically, the example in Fig. 7.1 shows the case where the inductance-based prediction model assumes a constant absolute inductance, but in the real system (i.e., the machine) the inductance is halved due to saturation. This also halves the electrical time constant τ_{el}, resulting in an actual current that is twice as high as the predicted value. The proposed method, however, avoids such pitfalls, enabling a more accurate calculation of the current trajectories, and, consequently, of VSPs thanks to which the current ripple can be significantly decreased.

Regarding the flux linkage, the nonlinear functions f_ψ and f_ψ^{-1} are stored in flux linkage maps and identified online during operation, as explained in Section 4.3.4. At this point, an important distinction must be made when using the online parameter identification method, as each map exists in two copies, i.e., in the RAM of the processor and the FPGA. Regarding the former, the flux linkage maps in the RAM are accessed by the processor and are modified in real time with the identification procedure described in Section 4.3.4. However, since the prediction procedure and the subsequent optimization process are fully implemented on the FPGA, the maps are also stored there so that can be directly used without introducing any communication delays. The procedure is illustrated in Fig. 3.3. This, however, may tax the computational and memory resources required for the proposed algorithm. To address this, methods that alleviate the computational load are implemented, as explained in the sequel.

7.3 Interpolation and Extrapolation

The online identification of the flux linkage maps is performed on a processor in a time-uncritical manner by—theoretically—using any kind of interpolation and extrapolation methods, e.g., IDW or cubic spline, see Section 4.3.4. For the FPGA implementation, however, interpolation and extrapolation are relatively complex.

Interpolation An option for avoiding interpolation on the FPGA is to use flux linkage maps with a very big number of sampling points. In this way, (valuable) DSP slices required for interpolation on the FPGA remain available for other functionalities, see Table 9.3 in Section 9.4.2. There are two ways to do this. One option is to store them in the RAM, where they are accessed during each control and prediction iteration. However, as mentioned before, storing these points in the RAM is not practical due to the very long latency associated with the memory access. If one adds up each memory access in each prediction step, the required prediction time would increase significantly. Alternatively, all sampling points could be stored in FPGA registers. However, this is also a poor choice from an implementation point of view since a significant amount of resources would be required for an acceptable degree of granularity. For example, considering the Zynq-7000 in Chapter 3, all available LUTs of the FPGA would be used for one 256×256 map. Maps with lower granularity, e.g., a 100×100 map, would result in a granularity of 600 mA for a current range of $[-30, 30]$ A, which is not accurate enough.
It is therefore not recommended to use the "nearest" choice, but rather interpolation. More specifically, the use of linear interpolation is recommended due to its resource efficiency—using FPGA registers—and dramatically reduced calculation time (and latency) inside the FPGA, as shown in Table 9.3 of Section 9.4.2. Moreover, as can be seen in Fig. 7.2, such an interpolation method is accurate enough despite its simplicity. When considering the test motor M3 (see Table A.3), a 16×16 map and a current range of $[-30, 30]$ A, the maximum deviation is 3 mA, which is negligible for the considered measurement range. Hence, no performance deterioration is observed comparing with the cubic spline method. Furthermore, as explained in Section 4.3.4

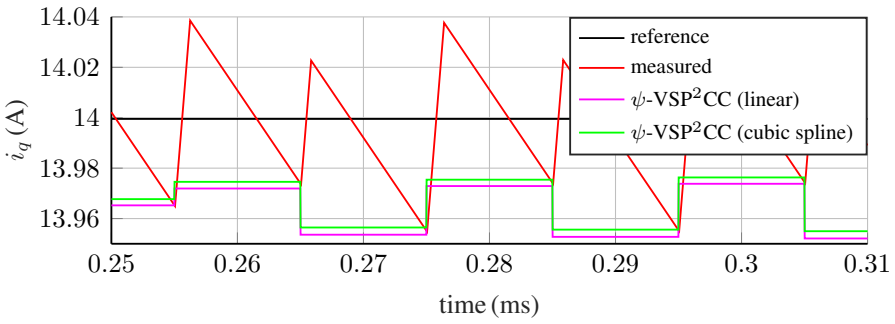


Figure 7.2: Motor M3: Comparison of interpolation methods for prediction using $i_d^* = -1$ A, $i_q^* = 14$ A at steady state with a speed of $n_\mathrm{m} = 100$ rpm (simulation).

flux linkage maps have always a continuous course and, if rotor position angle dependencies are neglected or averaged, they are also monotonically nondecreasing. Consequently, based on the proper identified flux linkage maps—interpolated by using a more accurate interpolation method, i.e., IDW, in a time-uncritical task—a linear interpolation for the control approach is sufficient and feasible in real time.

Extrapolation Considering resource efficiency and control frequencies of $f_{\mathrm{cf}} = 100\,\mathrm{kHz}$, extrapolation on the FPGA is complex and therefore better avoided. Thus, only interpolation is performed on the FPGA. Consequently, the resulting flux linkage maps must cover at least the entire current measurement range to allow prediction of current trajectories over the entire operating range. For this reason, the maps are generated in the processor using both interpolation and extrapolation so as to cover the whole measurement range of the current sensors as described in Section 4.3.4. This means that if, e.g., the measuring range of the current sensor is $[-30, 30]\,\mathrm{A}$, the flux linkage map created in the processor should cover at least this range, i.e., the processor uses IDW to calculate a flux linkage map that covers a range of $[-32, 32]\,\mathrm{A}$. Due to reasons explained in Section 3.1.2, parameter combinations at fixed, regular intervals (grid intervals), e.g., $2\,\mathrm{A}$, are chosen for the FPGA implementation. This results in 32×32 maps, which are then transferred from the processor to the FPGA during operation so that they are available for all steps of the prediction process. Hence, not performing extrapolation on the FPGA during the prediction has no adverse effects since the required information is readily available.

7.4 Optimization Problem

The optimization problem remains nearly the same compared to the one used for the inductance-based prediction in Section 6.2. The cost function (6.5), the current constraint definition (6.6) and the same switching states (6.7) are used. With cost function (6.5) and the additional mapping functions (7.5a) and (7.5b), the VSP^{2}CC problem is slightly adjusted and is stated as

$$\begin{aligned} &\underset{\bar{\boldsymbol{u}}_{abc} \in \mathbb{U}}{\text{minimize}} \;\; J_{\mathrm{f}} \text{ (see (6.5))} \\ &\text{subject to } \; (7.4),\; (7.5a),\; (7.5b),\; (6.6)\,. \end{aligned} \tag{7.6}$$

Solving (7.6) yields the optimal switch position(s) $\bar{\boldsymbol{u}}_{abc,\mathrm{con}}$ which is (are) to be applied at the corresponding optimal switching time instant(s), i.e., $t = 0$ and/or $t_{\mathrm{z,con}}$. Although the optimization remains the same, the prediction in this case is based on the flux linkage.

The entire optimization problem is shown in Algorithm 3, where the additional calculation steps compared to the inductance-based VSP^{2}CC variant (see Algorithm 2) are highlighted in gray. In addition, the modified—(2.73) is replaced by (7.4)—calculation steps are highlighted in red.

Algorithm 3 Flux linkage-based VSP^{2}CC

function $\bar{\boldsymbol{u}}_{\mathrm{ABC,CON}}, t_{\mathrm{z,CON}} = \psi$-VSP^{2}CC$(\boldsymbol{i}^*_{dq}, \boldsymbol{i}_{dq}, \varphi, N_\mathrm{p}, \omega_\mathrm{el}, V_\mathrm{dc})$
  $\boldsymbol{\psi}_{dq}(k-1) \leftarrow \boldsymbol{i}_{dq}(k-1)$ using (7.5a)
  $\boldsymbol{\psi}_{dq}(k) \leftarrow$ predict using (7.4) with t_z and T_cf ▷ delay time compensation
  $\boldsymbol{i}_{dq}(k) \leftarrow \boldsymbol{\psi}_{dq}(k)$ using (7.5b)
  $\boldsymbol{v}_{dq,\mathrm{db}}(k) \leftarrow$ dead-beat solution using (5.1) ▷ search space reduction
  $\gamma_\mathrm{db}(k) \leftarrow \angle \boldsymbol{v}_{dq,\mathrm{db}}(k)$ using (5.2) ▷ sector of candidate $\boldsymbol{v}_{dq}(k)$
  for $\ell = 1, \ldots, N_\mathrm{p}$ **do**
    $\boldsymbol{v}_{dq}(k+\ell-1) \leftarrow$ based on $\varphi(k+\ell-1), V_\mathrm{dc}(k)$ ▷ three candidate SVs based on $\gamma_\mathrm{db}(k)$
    for $j = 1, \ldots, 3$ **do** ▷ one SV within T_cf
      $\boldsymbol{\psi}_{dq,j}(k+\ell) \leftarrow$ predict using (7.4)
      $\boldsymbol{i}_{dq,j}(k+\ell) \leftarrow \boldsymbol{\psi}_{dq,j}(k+\ell)$ using (7.5b)
      $||\boldsymbol{i}_{dq,j}(k+\ell)|| \leftarrow \boldsymbol{i}_{dq,j}(k+\ell)$ using (6.6) ▷ current constraint
      if $\ell = 1$ **then**
        $\boldsymbol{m}_j(k+\ell) \leftarrow$ using (6.1) ▷ $\boldsymbol{i}_{dq}$ gradients
      end if
    end for
    if $\ell = 1$ **then** ▷ VSP for $N_\mathrm{p} = 1$
      for $n_1 = 1, \ldots, 3$ **do**
        for $n_2 = 1, \ldots, 3$ **do**
          if $n_1 = n_2$ **then** ▷ one SV within T_cf
            $\boldsymbol{i}_{dq,n_1,n_2}(k+\ell) \leftarrow \boldsymbol{i}_{dq,j}(k+\ell)$
          else ▷ VSP for two SVs
            $t_{\mathrm{z},n_1,n_2}(k+\ell) \leftarrow$ (6.4) ▷ VSP
            $\boldsymbol{\psi}_{dq,n_1,n_2}(k+\ell) \leftarrow$ predict with (7.4) [4]
            $\boldsymbol{i}_{dq,n_1,n_2}(k+\ell) \leftarrow \boldsymbol{\psi}_{dq,n_1,n_2}(k+\ell)$ using (7.5b)
            $||\boldsymbol{i}_{dq,n_1,n_2}(k+\ell)|| \leftarrow \boldsymbol{i}_{dq,n_1,n_2}(k+\ell)$ using (6.6) ▷ current constraint
          end if
        end for
      end for
    end if
    $\bar{\boldsymbol{u}}_{abc,\mathrm{con}}(k), t_{\mathrm{z,con}}(k) \leftarrow$ solve (7.6)
  end for
end function

7.5 Assessment of Influencing Factors

The utilization of the flux linkage maps in the prediction process, i.e., the ψ-based—in comparison to the L-based in Chapter 6—prediction, is particularly advantageous for (small) drives with saturation and cross-coupling effects for several reasons. More specifically, MPC with an L-based prediction model suffers from the following pitfalls:

- The bigger the phase resistance R_{ph}, as is the case, e.g., with small motors (ironless winding), the bigger the voltage drop across it. This, in turn, can lead to a significant prediction error. Since in case of the L-based prediction the inductance appears in the denominator of (2.73), a decrease in it results in an underestimated voltage drop across R_{ph}. A reduced inductance due to saturation, e.g., by half, erroneously leads to a predicted voltage drop across R_{ph}, which is half than it actually is.
- The larger the ratio between L_d and L_q, the more significant the prediction error when saturation happens. However, if the ratio remains constant over the entire operating range such effects may cancel each other in the term $\omega_{\mathrm{el}}\boldsymbol{L}_{dq}$ of (2.73).
- For inductances that are almost equal, i.e., $L_d \approx L_q$, the influence of ω_{el} on the prediction error is practically nonexistent in the d-axis and relatively small in the q-axis.
- Assuming a constant ϑ_{m}, the lower the ω_{el} is, the bigger the prediction error. This is due to a lower induced voltage at low speed which leads to a more significant voltage drop across the resistance and inductances. In other words, the increased current change causes an increase in the prediction error within one T_{cf}. This, however, as mentioned above, affects only the q-axis.
- The less ψ_d is affected by ψ_{pm}, i.e., more reluctance difference, the bigger the prediction error on the q-axis.
- For a given IPMSM and saturation effect, the prediction error increases with a higher dc-link voltage V_{dc} or longer control interval T_{cf}, i.e., when the ratio between f_{cf} and f_{sw} decreases.

[4]Theoretically, at time instant t_{z}, (7.5b) must be used to get $\boldsymbol{i}_{dq,n_1,n_2}$ so that the voltage drop over the resistance in the second part of the control interval is computed based on a more accurate current.

7.6 Performance Evaluation

The good steady-state and dynamic behavior of the L-VSP^{2}CC was shown in Section 6.3. Therein, low current distortions and fast transient responses for a PMSM-based drive system are demonstrated. As shown in this section, such features remain in place with the proposed VSP^{2}CC scheme that utilizes flux linkage maps (i.e., ψ-VSP^{2}CC). Moreover, such a favorable performance is even more pronounced with nonlinear machines. To show this, one PMSM with (mostly) linear behavior, i.e., motor M1, and two nonlinear machines (M3 and M4) have been validated as case studies. The parameters of the PMSMs under considerations are provided in Table A.3, while the flux linkage maps of motor M3 and M4 are shown in Figs. 4.16(a)-4.16(b), and Figs. 4.16(c)-4.16(d), respectively. Motor M1 and M3 are commercial ones, whereas motor M4 is a self-built prototype with fractional slot non-overlapping concentrated windings, where harmonics are more pronounced in the machine design. Given the rather linear nature of motor M1, L-VSP^{2}CC and ψ-VSP^{2}CC—under the assumption that the inductances were correctly identified—showed completely identical behavior, so the comparison was used for verification purposes only. Thus the reader is referred to the results in Section 6.3.
Similar to the previous chapter, FOC with SVM serves as benchmark for comparing the performance of the proposed ψ-VSP^{2}CC and L-VSP^{2}CC. Note that for a fair and meaningful comparison, the weighting factor λ_u in the MPC methods is adjusted such that (approximately) the same average switching frequency f_sw results for operation under the same conditions.

7.6.1 Steady-State Performance

Figs. 7.3 and 7.4 show the simulated steady-state behavior of the i_d and i_q currents for the nonlinear motor M3. For comparison purposes, the performance with the L-VSP^{2}CC scheme is also shown in the same figures. As can be seen, the latter control approach leads to an unnecessarily high current ripple (see the measured current shown with the blue line) and—consequently—high THD due to the inaccurate calculation of t_z (see the predicted current shown with the light blue line). On the other hand, the proposed ψ-VSP^{2}CC, which uses the flux linkage maps—see Figs. 4.16(a) and 4.16(b)—in the prediction process, computes the VSP instant t_z such that the current ripple (red line) is significantly lower. Consequently, the current THD also reduces.
Such an improved steady-state performance is even more pronounced with the prototype motor M4 due to its stronger nonlinear behavior.
The steady-state performance comparison—in terms of current THD (I_THD)—for the different drive systems and control approaches in question is summarized in Figs. 7.5-7.11.

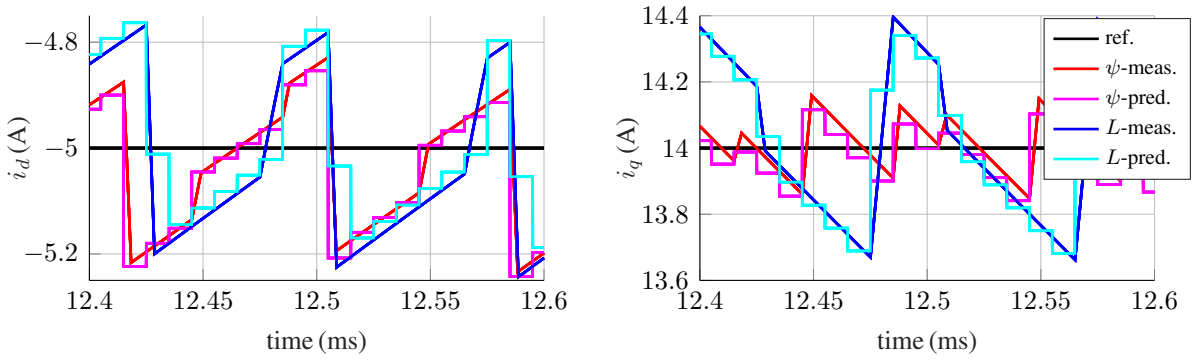


Figure 7.3: Motor M3: VSP^{2}CC at $\angle\boldsymbol{\psi}_{dq} = \frac{\pi}{6}$ rad with $N_\mathrm{p} = 2$, $f_\mathrm{sw} \approx 11.5\,\mathrm{kHz}$ and $n_\mathrm{m} = 100\,\mathrm{rpm}$ (simulation).

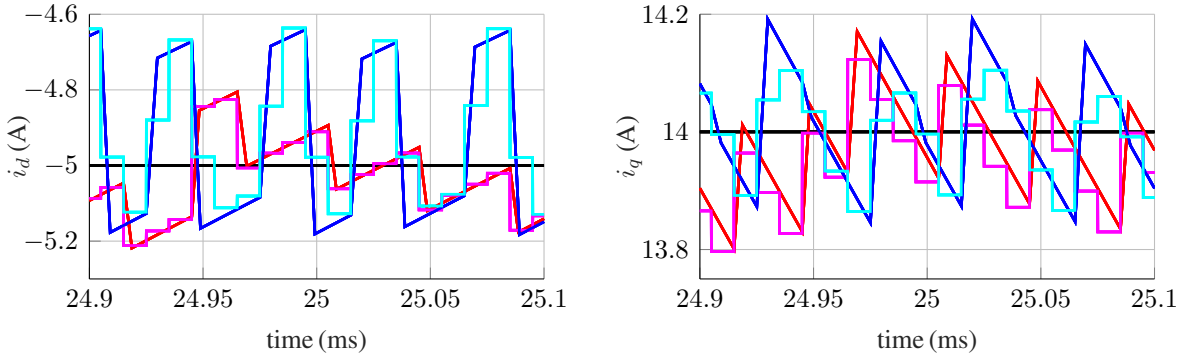
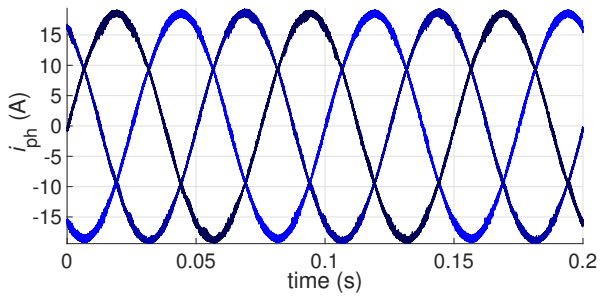


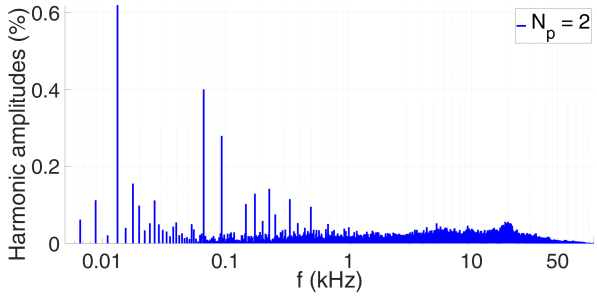
Figure 7.4: Motor M3: VSP^{2}CC at $\angle\boldsymbol{\psi}_{dq} = \frac{\pi}{3}$ rad with $N_\mathrm{p} = 2$, $f_\mathrm{sw} \approx 11.5\,\mathrm{kHz}$ at $n_\mathrm{m} = 100\,\mathrm{rpm}$ (simulation).

First, Fig. 7.5 shows the steady-state three-phase current and corresponding spectra, respectively, with both VSP^{2}CC approaches and FOC for current reference values in the range of current saturation for motor M3. Low speed, i.e., a low modulation index, is chosen since parameter inaccuracies have a higher impact in that case, see Sections 2.6.3 and 6.3.3. In addition, Fig. A.14 of Appendix A.5 shows operation at the nominal speed, i.e., 3000 rpm. As can be seen, ψ-VSP^{2}CC outperforms L-VSP^{2}CC, while the produced current THD with the proposed method is similar to that of the conventional control solution, i.e., FOC with SVM. For all spectra, the typically 5^{th} and 7^{th} harmonic are visible. In addition, in case of SVM, the amplitudes of the harmonic group of the 2^{nd} multiple of f_sw are pronounced and also higher than the amplitudes of the harmonics around f_sw due to the low modulation index. In contrast, with VSP^{2}CC the typically spread spectrum is visible, whereby the amplitudes of the L-VSP^{2}CC approach are on average higher than those of ψ-VSP^{2}CC, resulting in an increased THD.

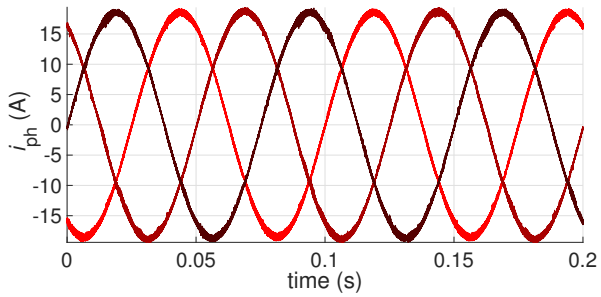
Fig. 7.6 shows the current ripple for both discussed operating points, i.e., low and high speed, when saturation is present. Since motor M3 is only slightly nonlinear, the difference between the two VSP^{2}CC approaches is small.



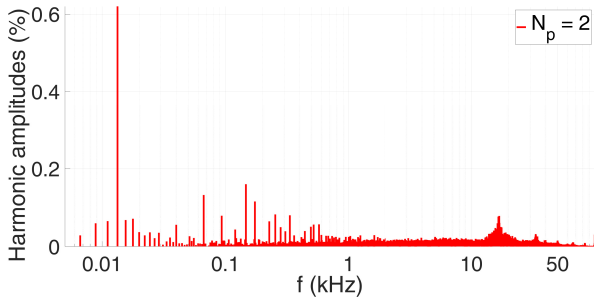
(a) Three-phase stator current for L-VSP^{2}CC with $f_{sw} \approx 10.2\,\text{kHz}$.



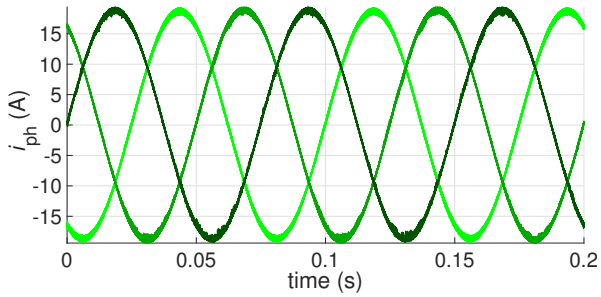
(b) Stator current spectrum for L-VSP^{2}CC with $I_{THD} = 1.85\,\%$.



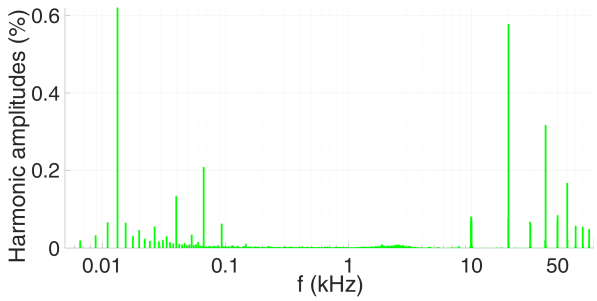
(c) Three-phase stator current for ψ-VSP^{2}CC with $f_{sw} \approx 10.3\,\text{kHz}$.



(d) Stator current spectrum for ψ-VSP^{2}CC with $I_{THD} = 1.48\,\%$.



(e) Three-phase stator current for FOC and SVM with $f_{sw} = 10.0\,\text{kHz}$.

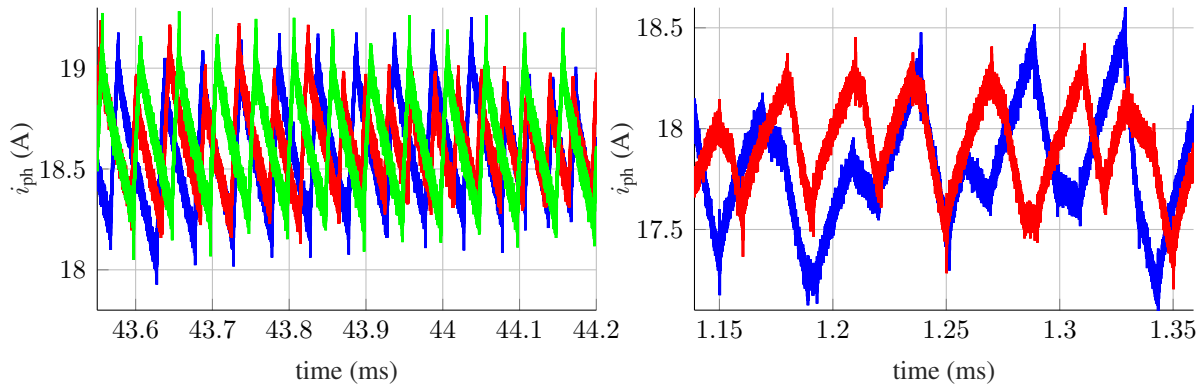


(f) Stator current spectrum for FOC and SVM with $I_{THD} = 1.49\,\%$.

Figure 7.5: Motor M3: Three-phase stator current and spectrum for $i_d^* = -5\,\text{A}$, $i_q^* = 18.03\,\text{A}$, $n_m = 200\,\text{rpm}$ with $N_p = 2$, $f_{sw} \approx 10.0\,\text{kHz}$ (experimental).

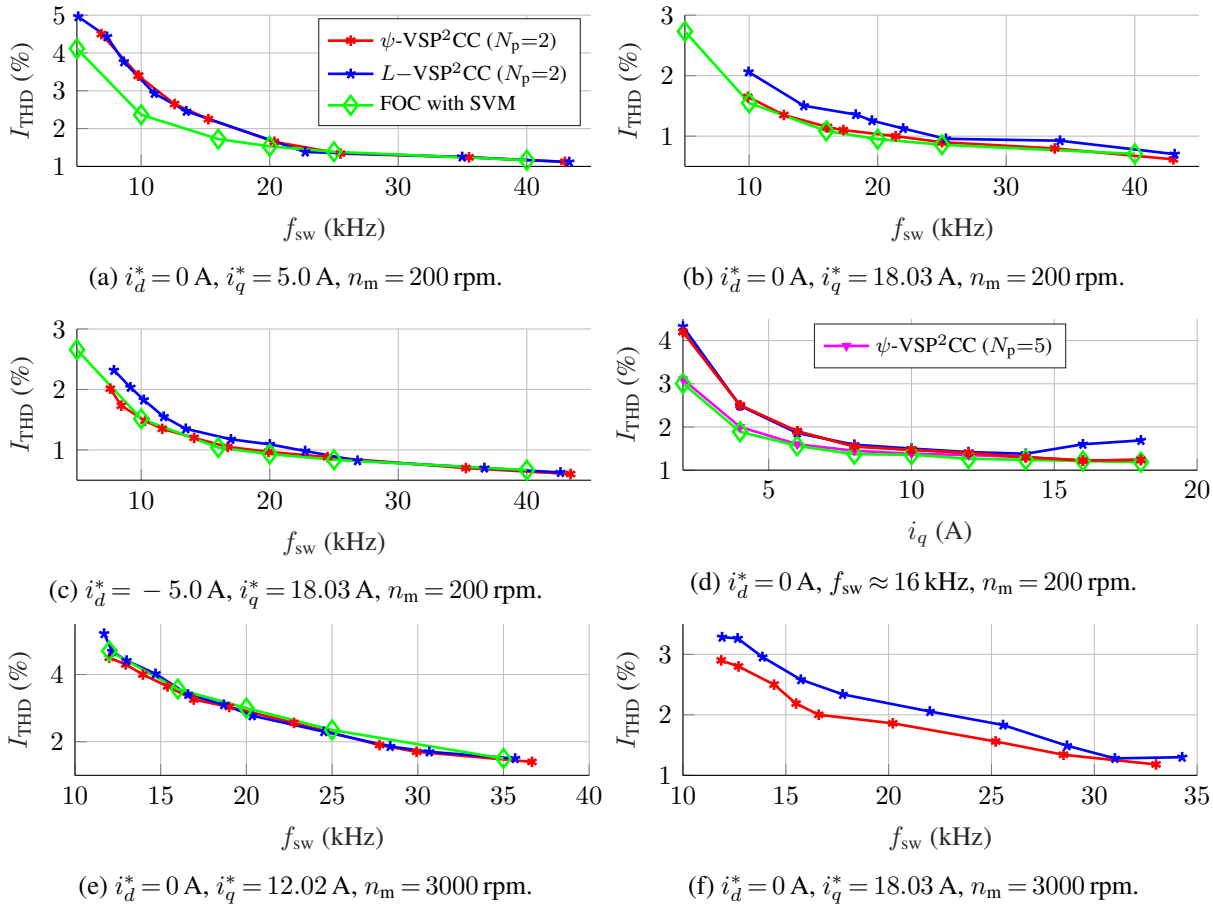
The evaluation in Fig. 7.7 shows that the more heavily the control effort is penalized (i.e., the lower f_{sw} gets by increasing the value of λ_u) the more important the prediction model accuracy becomes. This means that at lower switching frequencies, f_{sw}, the difference in the current THD produced by L- and ψ-VSP^{2}CC gets bigger. In addition, Fig. 7.7(d) shows that a recognizable difference between both VSP^{2}CC approaches—due to the nonlinear magnetic behavior—starts for motor M3 in the range of the nominal current, i.e., $\sqrt{2}I_N = 12.02\,\text{A}$.

Furthermore, as also shown in Fig. 7.7(d), a longer prediction horizon, e.g., $N_p = 5$ is required for an extremely low modulation index, i.e., at very low currents and speeds, to achieve a comparable THD as FOC with SVM. This can be explained by the fact that the time window in which switching should take place is better covered, see also Fig. 6.3.



(a) i_d^*=−5 A, i_q^*=18.03 A, n_m=200 rpm, f_{sw}≈10 kHz. (b) i_d^*=0 A, i_q^*=18.03 A, n_m=3000 rpm, f_{sw}≈16 kHz.

Figure 7.6: Motor M3: Single-phase stator current ripple as close-up view from Fig. 7.5 and Fig. A.14 to show the switching behavior in comparison for operation using VSP^{2}CC with $N_p = 2$ and FOC with SVM in steady state (experimental).



(a) $i_d^* = 0$ A, $i_q^* = 5.0$ A, $n_m = 200$ rpm.

(b) $i_d^* = 0$ A, $i_q^* = 18.03$ A, $n_m = 200$ rpm.

(c) $i_d^* = -5.0$ A, $i_q^* = 18.03$ A, $n_m = 200$ rpm.

(d) $i_d^* = 0$ A, $f_{sw} \approx 16$ kHz, $n_m = 200$ rpm.

(e) $i_d^* = 0$ A, $i_q^* = 12.02$ A, $n_m = 3000$ rpm.

(f) $i_d^* = 0$ A, $i_q^* = 18.03$ A, $n_m = 3000$ rpm.

Figure 7.7: Motor M3: Trade-off between I_{THD} and f_{sw} as well as I_{THD} and i_q for VSP^{2}CC and FOC with SVM (experimental). System two, see Fig. A.2, was used for $N_p = 5$.

Similar performance behavior is observed with the other nonlinear motor, namely motor M4. This is demonstrated in Fig. 7.8, where the phase currents for two different operating points are

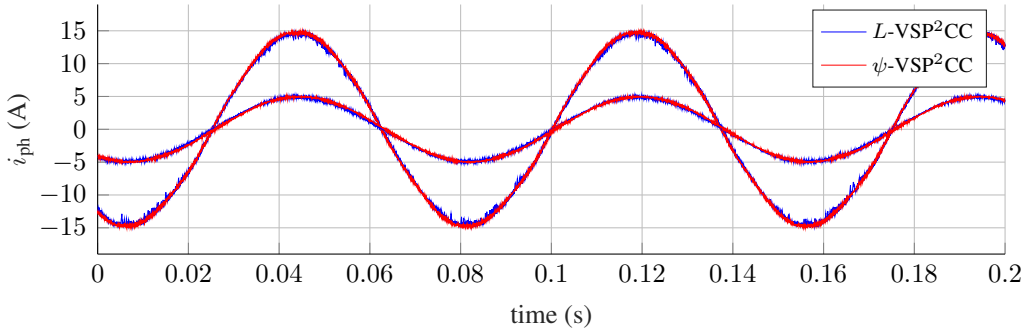


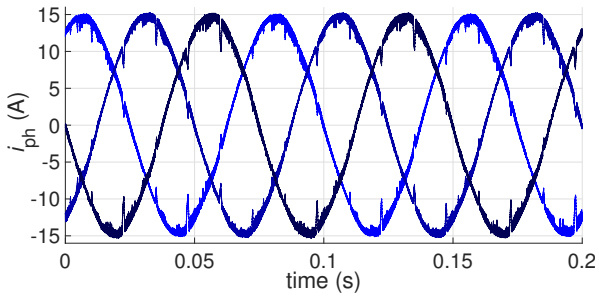
Figure 7.8: Motor M4: Single-phase stator current for L- and ψ-VSP^{2}CC with $i_d^* = -5.0\,\mathrm{A}$, $i_q^* = 14.0\,\mathrm{A}$, $n_\mathrm{m} = 200\,\mathrm{rpm}$, $N_\mathrm{p} = 2$ at $f_\mathrm{sw} \approx 16\,\mathrm{kHz}$. For comparison, an operating point without saturation, i.e., $i_d^* = 0\,\mathrm{A}$, $i_q^* = 5.0\,\mathrm{A}$, is shown (experimental).

shown when using L- and ψ-VSP^{2}CC with $N_\mathrm{p} = 2$. Due to saturation, caused by an increasing reference current, the current ripple and THD of L-VSP^{2}CC also increases.

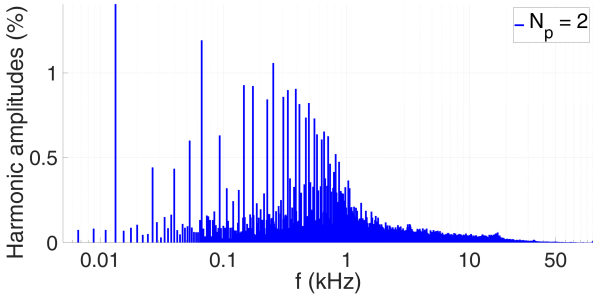
Fig. 7.9 shows the phase current and respective current spectra for motor M4 at low speed. Due to the increased nonlinear magnetic behavior in motor M4, the positive influence of ψ-VSP^{2}CC becomes even more obvious. In addition, Fig. A.15 of Appendix A.5 shows operation at higher speed, i.e., 800 rpm. Despite the good results, it has to be mentioned that the THD of motor M4 shows a difference between simulation and experiment. As can be seen in the spectrum of Fig. 7.9, the prototype M4 has pronounced harmonics, i.e., 5^th and 7^th, while the 2^nd and 4^th also appear due to the fractional slot winding. Such harmonics are not modeled (considered) in the controller, although they exist in the real motor. As shown in Section 4.3.4, the flux linkage maps are identified by averaging over all rotor positions and several turns. Therefore, neither the flux linkage maps nor the inductances take into account the harmonics.[5] Despite the unmodeled harmonics, however, ψ-VSP^{2}CC still achieves a significant reduction of the current THD with respect to that of L-VSP^{2}CC, while it is quite similar to that of FOC with SVM. For the spectrum of the latter, see Fig. 7.9(f), the amplitudes of the harmonic group of the 2^nd multiple of f_sw are higher than the amplitudes of the f_sw group, similar to M3. Whereas in Fig. A.15(f) it is the other way around since a high modulation index is used.

Fig. 7.10 shows the current ripple and Fig. 7.11 the THD of the current over the switching frequency, both for motor M4. Here, nonlinearities are much more pronounced compared to motor M3, i.e., saturation starts already at $4\,\mathrm{A}$, see Fig. 7.11(d). Consequently, the proposed ψ-VSP^{2}CC achieves a more significant reduction in the current THD compared to L-VSP^{2}CC at low modulation indices, i.e, at low speeds. At nominal speeds, ψ-VSP^{2}CC outperforms all discussed control methods (FOC and L-VSP^{2}CC) over the whole range of examined switching frequencies, see, e.g., Figs. 7.11(e) and 7.11(f). Similar to motor M3, Fig. 7.11(d) shows that for

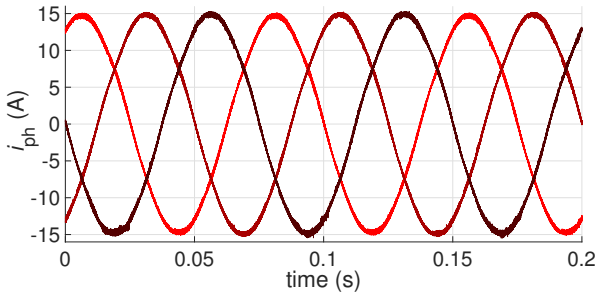
[5] To address this, as mentioned, the flux linkage maps would need one more dimension to depict the rotor position effects. This, however, would increase the associated computational load. To keep the calculation effort modest, an alternative is to use a correction term, see Chapter 8.



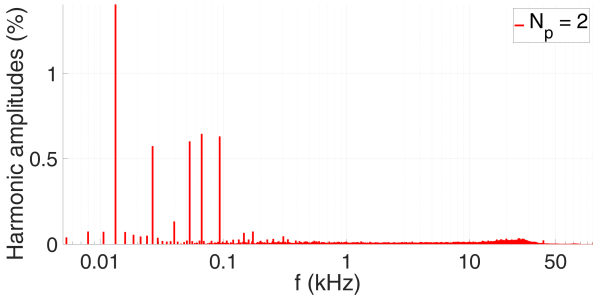
(a) Three-phase stator current for L-VSP^{2}CC with $f_{sw} \approx 13.17\,\text{kHz}$.



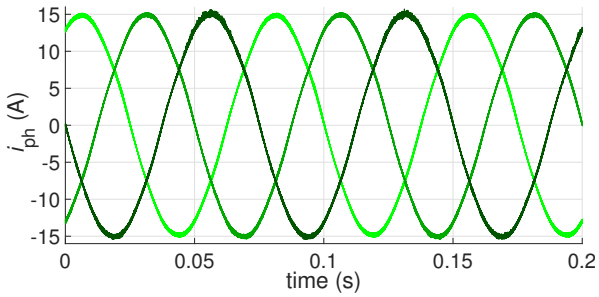
(b) Stator current spectrum for L-VSP^{2}CC with $I_{THD} = 5.28\,\%$.



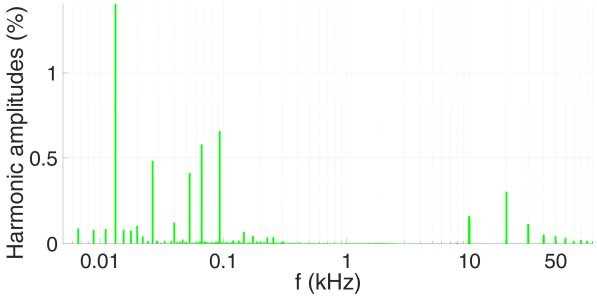
(c) Three-phase stator current for ψ-VSP^{2}CC with $f_{sw} \approx 10.0\,\text{kHz}$.



(d) Stator current spectrum for ψ-VSP^{2}CC with $I_{THD} = 1.90\,\%$.



(e) Three-phase stator current for FOC and SVM with $f_{sw} = 10.0\,\text{kHz}$.

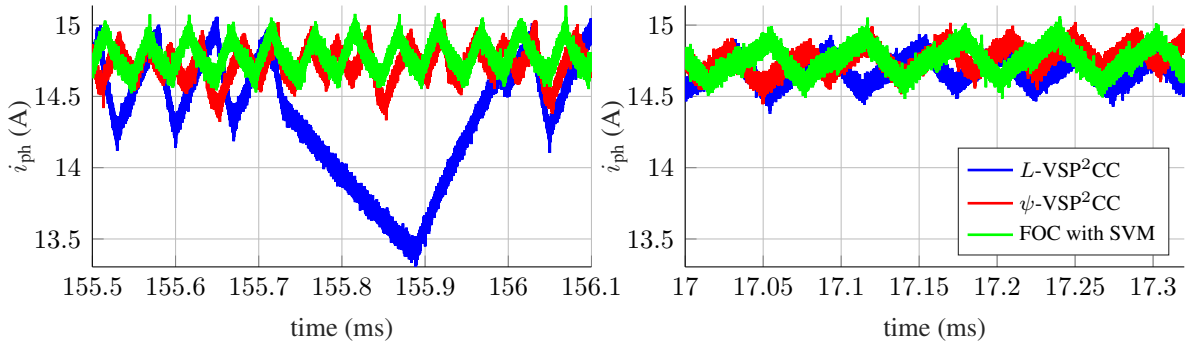


(f) Stator current spectrum for FOC and SVM with $I_{THD} = 1.81\,\%$.

Figure 7.9: Motor M4: Three-phase stator current and spectrum for $i_d^* = -5.0\,\text{A}$, $i_q^* = 14.0\,\text{A}$, $n_m = 200\,\text{rpm}$ with $N_p = 2$, $f_{sw} \approx 10.0\,\text{kHz}$ (experimental).

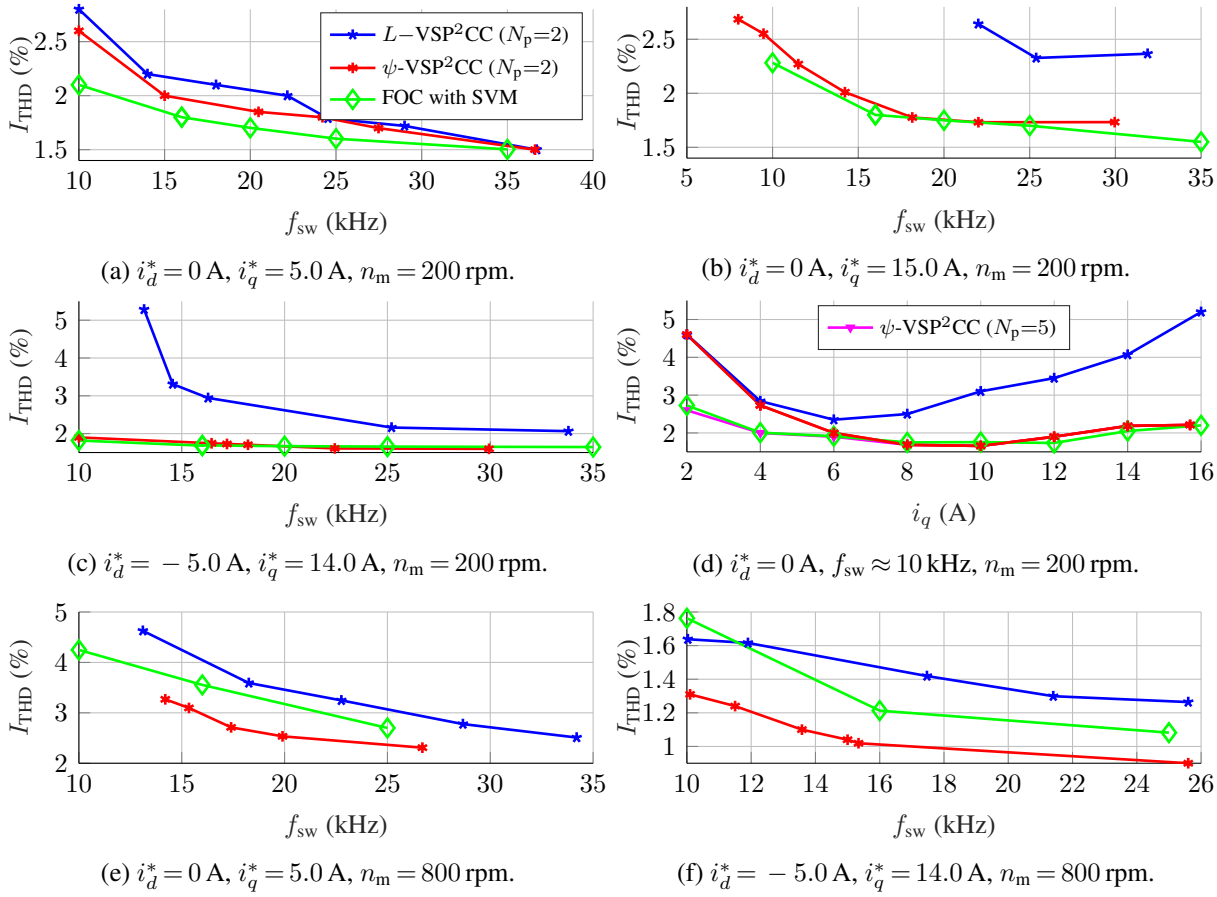
a small modulation index, i.e., current and speed are very small at the same time, an increased horizon helps to improve the THD at low switching frequencies. Apart from the extremely small modulation index, $N_p = 2$ is completely sufficient. However, as bigger the ratio between f_{cf} and the average f_{sw}, i.e., as more switching is penalized, the more beneficial is an increased horizon since the time window in which switching should take place is better covered. In addition, by further increasing the sampling/control frequency so as to increase the ratio between f_{cf} and f_{sw}, the switching granularity can be further improved. This will lead to an even better performance of ψ-VSP^{2}CC, as reported in [KG20, Section V].

Based on the above-mentioned figures, it can be observed that the current ripple is also affected by the position of the rotor flux vector within the triangular sectors formed by the voltage SVs.



(a) $i_d^*=-5$ A, i_q^*=14 A, n_m=200 rpm, $f_{sw}\approx$10 kHz. (b) $i_d^*=-5$ A, i_q^*=14 A, n_m=800 rpm, $f_{sw}\approx$16 kHz.

Figure 7.10: Motor M4: Single-phase stator current ripple as close-up view from Fig. 7.9 and Fig. A.15 to show the switching behavior in comparison for operation using VSP^{2}CC with $N_p=2$ and FOC with SVM in steady state (experimental).



(a) $i_d^*=0$ A, $i_q^*=5.0$ A, $n_m=200$ rpm.

(b) $i_d^*=0$ A, $i_q^*=15.0$ A, $n_m=200$ rpm.

(c) $i_d^*=-5.0$ A, $i_q^*=14.0$ A, $n_m=200$ rpm.

(d) $i_d^*=0$ A, $f_{sw}\approx10$ kHz, $n_m=200$ rpm.

(e) $i_d^*=0$ A, $i_q^*=5.0$ A, $n_m=800$ rpm.

(f) $i_d^*=-5.0$ A, $i_q^*=14.0$ A, $n_m=800$ rpm.

Figure 7.11: Motor M4: Trade-off between I_{THD} and f_{sw} as well as I_{THD} and i_q for VSP^{2}CC and FOC with SVM (experimental). System two, see Fig. A.2, was used for $N_p=5$.

As mentioned e.g. in [HB94], assuming a constant switching frequency, the current ripple is higher when the flux vector is located on the border of a sector, i.e., the induced voltage is in the middle of a sector. This is the case regardless of the modulation/control method used, as it depends on the relative position of the flux vector with respect to the voltage SVs. More

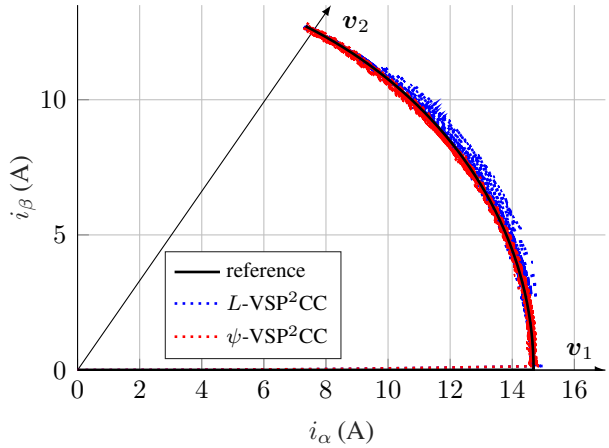


Figure 7.12: Motor M3: Current for the first sector with $i_d^* = -5\,\text{A}$, $i_q^* = 14\,\text{A}$ at steady state with a speed of $n_\text{m} = 100\,\text{rpm}$ (simulation).

precisely, the closer the flux is to the voltage SVs, the bigger the voltage difference between the induced and applied voltage, and, consequently, the bigger the current ripple. Thus, in case of DMPC, where the switching frequency may vary within the sector, more switching is performed in the middle of each sector and less at the sector border to maintain a constant current ripple.

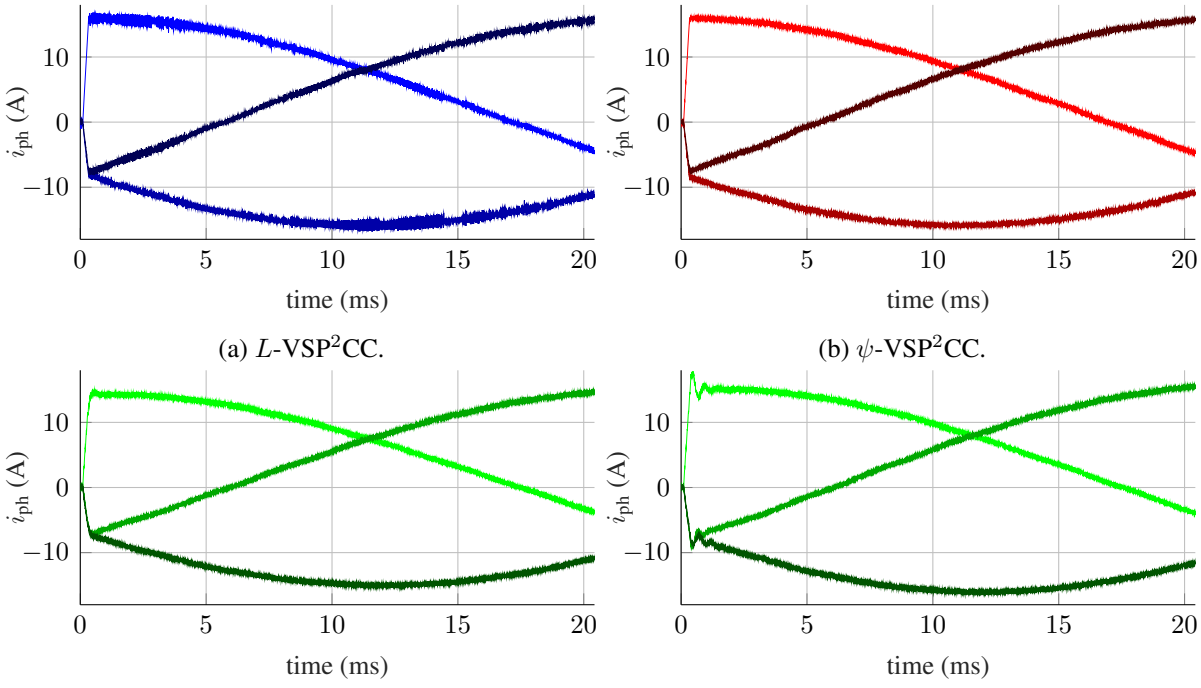
However, as can be seen in Figs. 7.3 and 7.4, there is a significant difference in the current prediction performed by the two VSP^{2}CC methods. For example, in the middle of each sector, e.g., at $\frac{\pi}{6}$ rad, the current error on the q-axis is more dominant for L-VSP^{2}CC, whereas at the sector borders, e.g., at $\frac{\pi}{3}$ rad, it is the other way around, i.e., the d-axis error is bigger. Considering the flux linkage maps in Figs. 4.16(a)-4.16(b), it can be noticed that the saturation effect on the q-axis is more pronounced than that on the d-axis. Since for the chosen case studies the q-component of the reference current prevails, i.e., $|i_q^*| > |i_d^*|$, when it is aligned with a voltage SV then a suboptimal prediction has a stronger adverse effect on the system performance. As a result, L-VSP^{2}CC produces an even higher current ripple at the middle of each sector, see Fig. 7.12. On the other hand, as can be seen in the same figure, ψ-VSP^{2}CC successfully tackles this issue thanks to the more accurate current prediction.

Given the presented results for the steady-state operation of the examined drive systems, it can be claimed that ψ-VSP^{2}CC clearly outperforms L-VSP^{2}CC. The current distortions, as quantified by the current THD, significantly decrease, especially when machines with nonlinear behavior are of concern. Considering saturation and cross-coupling—occurring at higher values of the current reference—ψ-VSP^{2}CC has the biggest advantage at low speed, since at lower modulation index a parameter inaccuracy has a higher impact on the performance of DMPC. Moreover, ψ-VSP^{2}CC achieves similar steady-state behavior as FOC with SVM. However, at high speeds, e.g., the rated speed, and when the ratio between f_sw and the fundamental frequency is small, the MIMO optimization of ψ-VSP^{2}CC even allows a lower THD compared to FOC.[6]

[6]For FOC, as expected, the PI gains have a strong influence on the steady-state performance and must be adjusted carefully depending on the operating point, i.e., the inductances must be updated.

7.6.2 Transient Performance

Besides the steady-state behavior, the transient behavior of the drive systems is also of interest. [Ric16] and [Kel12] showed that the dynamic behavior of nonlinear machines can be improved by compensating at least for the delay time independently of the control approach used. However, by accounting for the nonlinear model in the prediction and optimization processes an even more significant improvement can be achieved, as shown in the following. Fig. 7.13 shows the transient performance for the methods of interest by applying a current reference step. In the close-up view in Fig. 7.14(a) it can be noticed that both VSP^{2}CC approaches show quite similar dynamic behavior. However, when saturation is present, L-VSP^{2}CC calculates suboptimal switching points once the current reaches its reference, resulting in unnecessarily high current ripple—see Fig. 7.14(a)—and torque ripple, see Fig. 7.14(b). Fig. 7.15(a) shows the same behavior for a lower reference current at standstill. At higher speeds, see, e.g., for nominal speed with $i_q^* = 1.5 I_\mathrm{N}$ in Fig. 7.15(b), the voltage amplitude is bigger, so an accurate VSP is less important and the negative influence is smaller. Note that in Figs. 7.13 - 7.15 the d-axis is aligned with the a-phase at the beginning of the transient. Hence, the a-phase current initially rises, while b- and c-phase currents decrease. Moreover, it can be noticed that VSP^{2}CC–regardless of the prediction method—provides also faster current responses than FOC. The latter can be somewhat improved by tuning the PI controller more aggressively, at the expense, however,



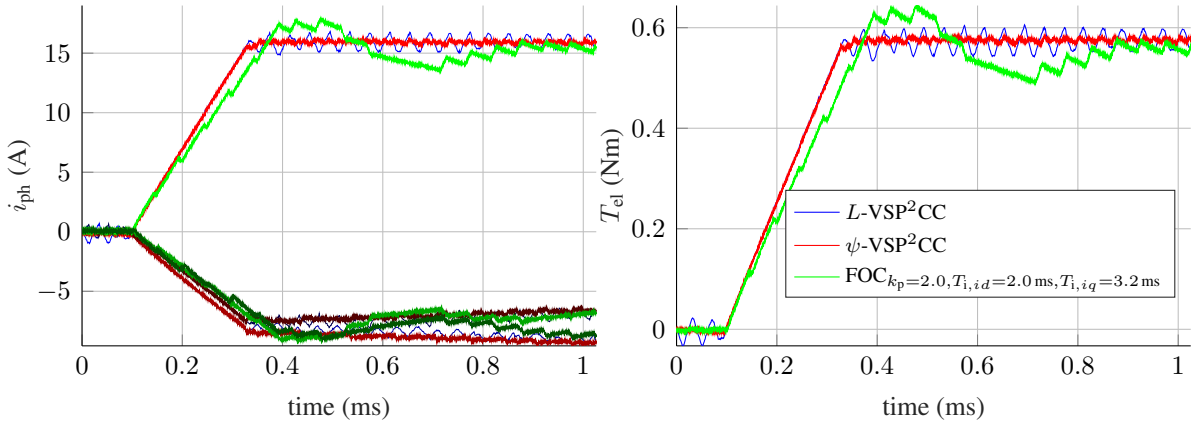
(a) L-VSP^{2}CC.

(b) ψ-VSP^{2}CC.

(c) FOC with $k_\mathrm{p} = 1.0$, $T_{\mathrm{i},id} = 2.0\,\mathrm{ms}$, $T_{\mathrm{i},iq} = 3.2\,\mathrm{ms}$.

(d) FOC with $k_\mathrm{p} = 2.0$, $T_{\mathrm{i},id} = 2.0\,\mathrm{ms}$, $T_{\mathrm{i},iq} = 3.2\,\mathrm{ms}$.

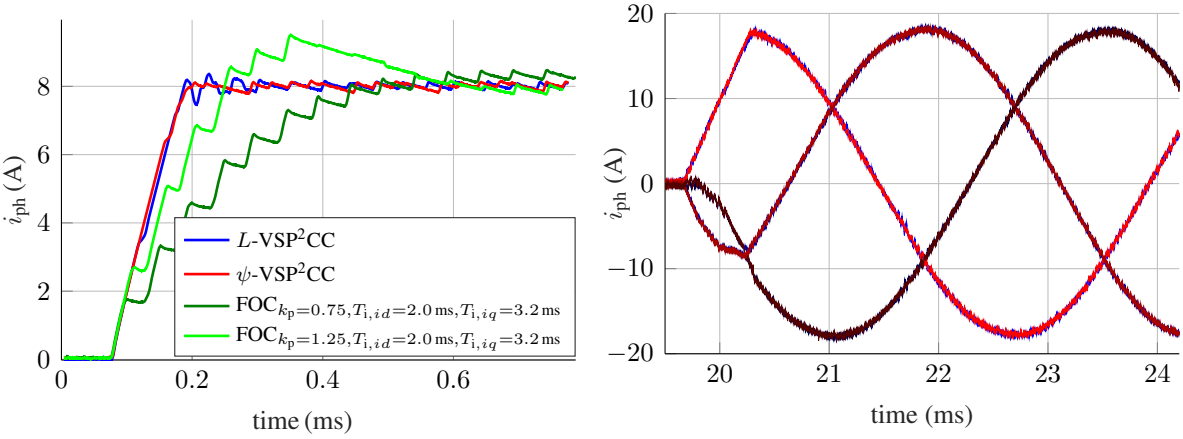
Figure 7.13: Motor M3: Three-phase stator current showing a current reference step from $i_q^* = 0.0\,\mathrm{A}$ to $i_q^* = 16\,\mathrm{A}$ with $i_d^* = 0\,\mathrm{A}$, $n_\mathrm{m} = 200\,\mathrm{rpm}$, $f_\mathrm{sw} \approx 10.0\,\mathrm{kHz}$ after the transient occurred and $N_\mathrm{p} = 2$ (experimental).



(a) Three-phase stator current during the transient.

(b) Electromagnetic torque during the transient.

Figure 7.14: Motor M3: L- and ψ-VSP^{2}CC and FOC for a step-change from $i_q^* = 0\,\mathrm{A}$ to $i_q^* = 16.0\,\mathrm{A}$ with $i_d^* = 0\,\mathrm{A}$, $f_{\mathrm{sw}} \approx 20\,\mathrm{kHz}$, $n_{\mathrm{m}} = 200\,\mathrm{rpm}$, $N_{\mathrm{p}} = 2$. The left hand side is a close-up view from Fig. 7.13 (experimental).



(a) Single-phase current for step-change from $i_q^* = 0\,\mathrm{A}$ to $i_q^* = 8.0\,\mathrm{A}$ at $n_{\mathrm{m}} = 0\,\mathrm{rpm}$ with $f_{\mathrm{sw}} \approx 10\,\mathrm{kHz}$.

(b) Three-phase current for step-change from $i_q^* = 0\,\mathrm{A}$ to $i_q^* = 18.24\,\mathrm{A}$ at $n_{\mathrm{m}} = 3000\,\mathrm{rpm}$ with $f_{\mathrm{sw}} \approx 20\,\mathrm{kHz}$.

Figure 7.15: Motor M3: Stator current for L- and ψ-VSP^{2}CC and FOC during transient operation for an i_q^* step-change with $i_d^* = 0\,\mathrm{A}$, $N_{\mathrm{p}} = 2$ (experimental).

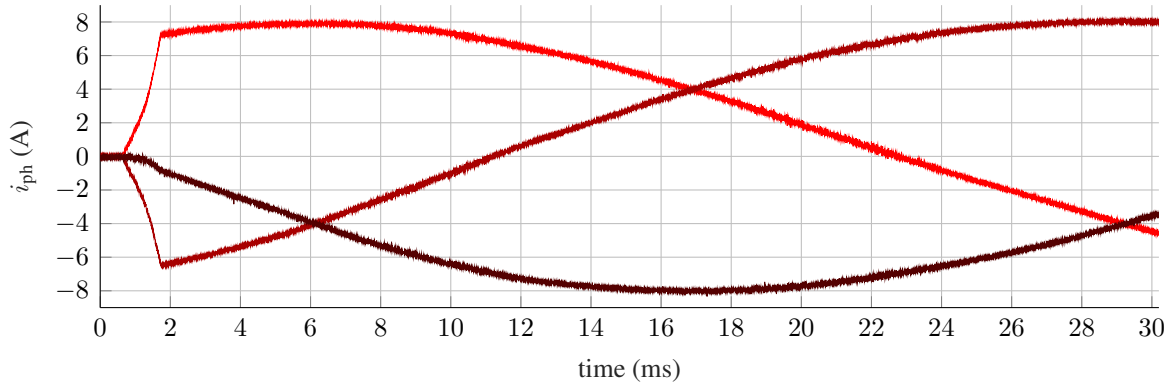
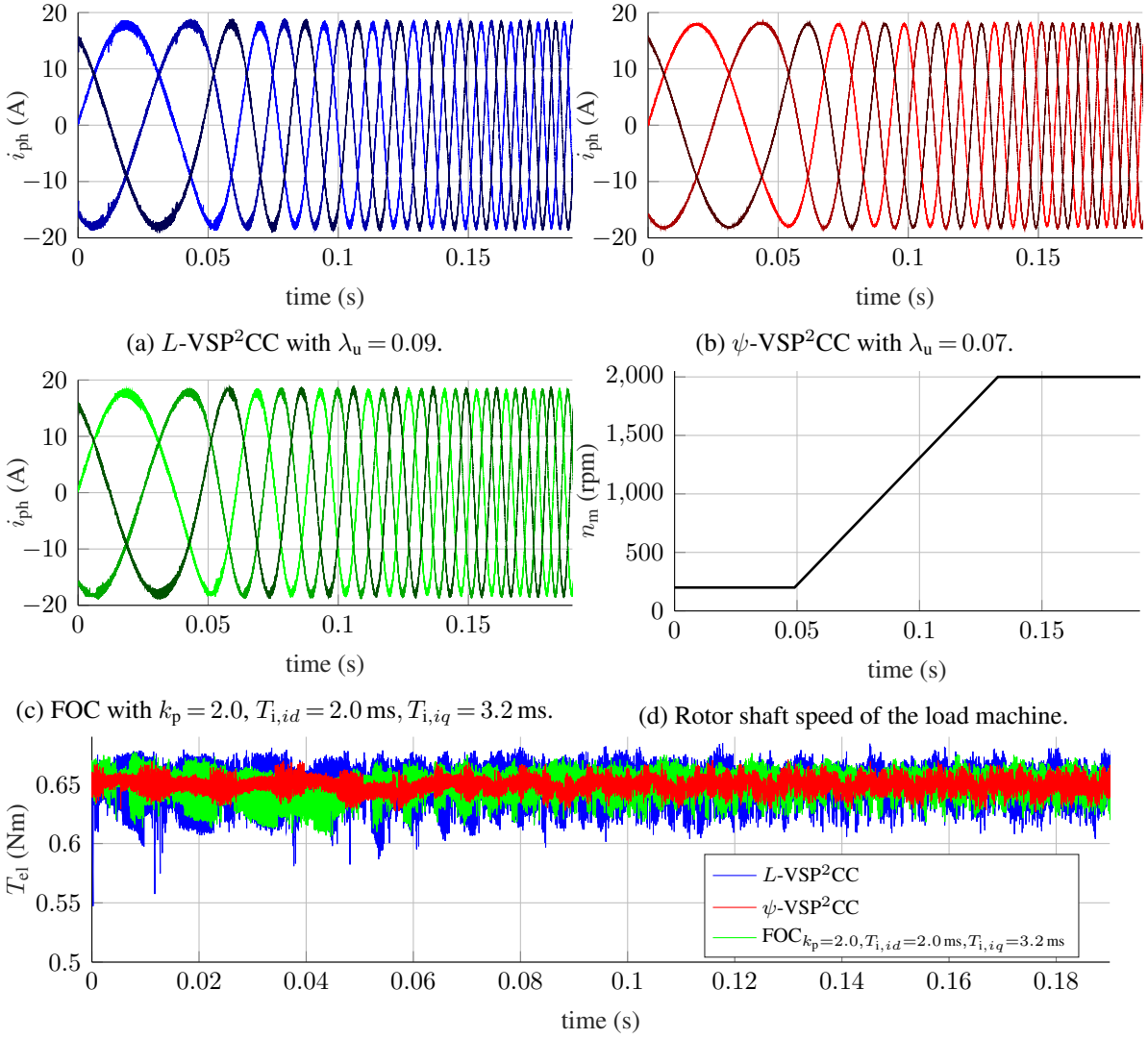


Figure 7.16: Motor M4: Three-phase stator current for ψ-VSP^{2}CC for step-change from $i_q^* = 0\,\mathrm{A}$ to $i_q^* = 8.0\,\mathrm{A}$ with $i_d^* = 0\,\mathrm{A}$, $n_{\mathrm{m}} = 200\,\mathrm{rpm}$, $f_{\mathrm{sw}} \approx 10\,\mathrm{kHz}$ (experimental).



(a) L-VSP^{2}CC with $\lambda_{\mathrm{u}} = 0.09$.

(b) ψ-VSP^{2}CC with $\lambda_{\mathrm{u}} = 0.07$.

(c) FOC with $k_{\mathrm{p}} = 2.0$, $T_{\mathrm{i},id} = 2.0\,\mathrm{ms}$, $T_{\mathrm{i},iq} = 3.2\,\mathrm{ms}$.

(d) Rotor shaft speed of the load machine.

(e) Electromagnetic torque during speed change.

Figure 7.17: Motor M3: Three-phase stator current during a load speed change from $n_{\mathrm{m}} = 200$ to 2000 rpm with $i_q^* = 18.03\,\mathrm{A}$, $i_d^* = 0\,\mathrm{A}$. $f_{\mathrm{sw}} \approx 20.0\,\mathrm{kHz}$ before the transient occurs and $N_{\mathrm{p}} = 2$ (experimental results).

of current overshoots[7]. However, Figs. 7.14(a) and 7.15(a) show that even if an overshoot is accepted, the dynamic of FOC is lower for reasons already discussed in Section 6.3.2. As explained there, both VSP^{2}CC approaches have the freedom to initially use only active voltage SVs, whereas FOC with SVM has to implement zero SVs which detract from the transient response. Moreover, Fig. 7.16 shows the transient behavior of motor M4. Finally, the performance of the proposed and benchmark controllers are tested under changes in the speed. Specifically, Fig. 7.17 shows the phase currents and torque when the load machine changes the speed. As can be seen, all controllers exhibit good behavior, although the quality of the current and torque with ψ-VSP^{2}CC is clearly higher (i.e., less ripples), as also demonstrated in Section 6.3.1.

[7]Decoupling network was active and two sets of gains were set based on the modulus optimum, see Section 2.3.

7.7 Final Assessment

In this chapter, an extension of the VSP^{2}CC algorithm discussed in Chapter 6 is presented in which the flux linkage is used instead of the machine inductances for the stator current prediction. This significantly improves the prediction accuracy and thus the effectiveness of the implemented MPC algorithm when dealing with machines with pronounced nonlinear phenomena such as saturation and cross-coupling, e.g., highly utilized IPMSMs. The more favorable execution of the switching operations is especially supported by the high accuracy of the gradient calculation, which is particularly advantageous when a VSP is used within the control interval. As a result, the advantageous behavior of VSP^{2}CC, see Section 6.4, is maintained even for PMSMs with a strongly nonlinear magnetic circuit. This means that when the flux linkage is used instead of the inductance to predict the current response, the same two main performance criteria, i.e., low THD and fast transient response, are achieved extremely successfully.
More specifically, ψ-VSP^{2}CC can dramatically improve the steady-state performance compared to L-VSP^{2}CC, as quantified by the current distortions (i.e., current THD). Moreover, the improved gradient calculation achieved with ψ-VSP^{2}CC, leads not only to an improved THD at steady state, but also to less oscillations during transients. Thanks to the direct nature of the control scheme, very fast transients with very short settling times are achieved, even when leaving the linear range of the magnetic circuit. Moreover, ψ-VSP^{2}CC achieves similar steady-state performance with FOC while exhibiting superior dynamic behavior.
Even if an effort for the identification of the flux linkage—compared to the identification of absolute inductances—is added, and an increase for the calculation effort can be recognized, the associated benefit outweighs such a cost. The above imply that the proposed ψ-VSP^{2}CC shifts the control effort even more—compared to L-VSP^{2}CC—from the design to the computational stage. However, if the use of the flux linkage is computationally infeasible—since, e.g., a platform with low processing power has to be used—thus making the real-time implementation of the proposed algorithm challenging, an alternative is to use look-up tables for the absolute inductances. In this way, (self-) saturation effects can be—to some extent—respected and the prediction error is—at least partially—minimized.

Finally, it is worth mentioning that thanks to the flux linkage-based control, the identified flux linkage maps can also be used, e.g., for (current) loss minimized torque control, i.e., maximum torque per ampere (MTPA) in the base speed range or maximum torque per volt (MTPV) in the field weakening range, as shown for nonlinear PMSMs, e.g., in [NPRS14, BK18].

8 Repetitive Position Dependent Flux Linkage Correction

As demonstrated in the previous chapter, flux linkage prediction allows for the saturation effects to be taken into account. However, the influences of iron losses and spatial harmonics—described in Section 2.2.1.2—are still either averaged or neglected, see, e.g., Section 4.3.4, thus not considered. Further current harmonics in inverter-fed machines can be caused by nonlinearities of the inverter and switching of the inverter. Even if the above mentioned effects do not have a (significant) impact on the dynamics this may lead to deviations for the steady-state accuracy and performance. In addition, insufficiently compensated parameter mismatches, e.g., the magnet temperature, may still be theoretically possible and thus cause a steady-state error.

The first idea to compensate for model deviations that may still exist is to use model-free MPC as already described in Section 2.6.3. However, model-free MPC is inappropriate for the following two reasons. First, model-free MPC, as the name implies, would lump all nonlinearities present in the model without separating the influences on a parameter-by-parameter basis. Thus, an improved behavior in transients or shortly after transients, as shown in the previous Chapter 7, cannot be achieved, since the change of the inductance, i.e., saturation, is not considered separately. Second, to the best knowledge of the author, the model-free MPC approaches published so far do not take into account rotor-position dependent effects that would require a memory to benefit from the knowledge of deviations from previous rotor revolutions.
Motivated by the above, a flux linkage correction term is introduced in this chapter, which allows to compensate for the remaining sources of possible steady-state errors depending on the rotor angle position and speed. In other words, as mentioned earlier, relevant model and parameter deviations that may adversely affect the prediction are identified and modeled, see Chapter 4 and Chapter 7, and only effects that are less pronounced or insufficiently compensated for—in case of the considered small PMSMs—are addressed in the following.
Therefore, a new extension for the MPC algorithm is introduced, which is based on the approaches of repetitive control (RC) and iterative learning control (ILC). Both methods use previous information to design a new control signal. This means that these methods can *learn* from past experience to improve the control performance in future steps. Since the proposed method is mainly used for tracking and rejecting periodic exogenous signals and does not use a state

space description, it is assigned to the RC family [WGD09]. An overview and further information regarding RC can be found, e.g., in [FW75, HW96, CDX04, WGD09, KCMW14].

In the field of power electronic systems, RC is used to suspress, e.g., voltage harmonics [LJS+17], current harmonics [BPZ08, Ric16] or harmonics caused by the inverter, i.e., interlock time and zero current crossing [BB04]. In [TGFZ16, TFOZ20] torque ripple is suppressed by RC. Here, in the former case a reference current and in the latter case a reference torque is modified. In addition, the former must reliably detect transients and prevent the RC algorithm from learning during this time, which introduces additional complexity. To avoid transient detection and still use RC for steady-state effects without creating instability, an accurate model of the dynamic behavior of the controlled system is required, see, e.g., the torque ripple observer in [TFZ19]. A good overview of further RC approaches is given, e.g., in [Ric16, p. 81]. However, all RC approaches available so far use PI-based or dead-beat controllers, where a compensation of the control voltage—continuous control signal—usually occurs just before the voltage is applied to the modulator, i.e., as feed-forward control, see, e.g., [Ric16, p. 57]. Note that some of them also modify the reference current or torque to compensate for the repetitive disturbances [TGFZ16, TFOZ20]. Nevertheless, a modulator is always present in all the methods mentioned. In contrast to the above, a concept of repetitive control for direct control methods—without modulator—such as DMPC, i.e., ψ-VSP^{2}CC, is proposed in the following.

8.1 Flux Linkage Correction Concept

If all the previous requirements and boundary conditions are combined, an RC method is necessary that primarily compensates for the influences of spatial harmonics (thus avoids harmonic-based flux linkage maps, see Section 4.3.4), harmonics from the inverter and iron losses, but secondarily also for other effects such as undetected parameter and model deviations. Here, the ability to control and damp harmonics is limited only by the switching frequency (or the Nyquist-Shannon sampling theorem [Lun17, p. 427]) and the voltage limit of the inverter. In addition, only the existing measurements, i.e., current, position and speed, should be used.

The proposed algorithm—which belongs to the class of RC methods—takes advantage of the fact that effects describable as flux linkage errors, i.e., the previously mentioned nonlinearities, are periodically repeated in steady-state operation with the electrical rotor angle φ. Even if the electrical angle is sufficient for all electrical periodic signals, the mechanical angle, i.e., φ_m, is chosen in the following to take into account also possible mechanical eccentricities of the shaft. More precisely, the proposed method stores a flux linkage deviation $\boldsymbol{\psi}_{dq,\mathrm{dev}}$ as a function of φ_m

over one mechanical turn. In this process, each position-dependent deviation value is averaged over several mechanical revolutions and is thus robust against noisy measurements. This makes it possible to feed-forward this deviation in the following mechanical revolutions to compensate for periodic and slowly varying model and parameter mismatches. From the control point of view, this corresponds to a rotor angle-dependent integrator component. Thus, angle-dependent effects such as current harmonics are fully compensated for during steady-state operation. Here, the dynamic behavior is not impaired. The latter is also not necessary, since the dynamic behavior is already sufficiently good, see Chapter 7, and angle-dependent effects play a minor role—are acceptable—in dynamic processes anyway. Hence, the flux linkage deviation must be identified fast and precisely to pre-control it correctly, thereby enabling a stable control over the entire operating range without affecting the dynamic behavior.

Therefore, a $\boldsymbol{\psi}_{dq,\mathrm{dev}}(k-1)$ vector is calculated during each control interval. Here, the original predicted flux linkage—using (7.4)—calculated in the previous control step $k-2$ for the delay time compensation, i.e., $\boldsymbol{\psi}_{dq}(k-1|k-2)$, is subtracted from the determined $\boldsymbol{\psi}_{dq}(k-1|k-1)$ based on the measured currents $\boldsymbol{i}_{dq}(k-1)$ by using (7.5a). If the currently determined flux linkage differs from the previously predicted one, a deviation results, i.e.,

$$\boldsymbol{\psi}_{dq,\mathrm{dev}}(k-1) = \boldsymbol{\psi}_{dq}(k-1|k-1) - \boldsymbol{\psi}_{dq}(k-1|k-2)\,, \tag{8.1}$$

which is stored together with the actual rotor position φ_m. The resulting flux linkage deviation $\boldsymbol{\psi}_{dq,\mathrm{dev}}$ cannot be separated into the components of the machine and the inverter. However, this is not necessary for control purposes as long as a single effect does not have a dominant influence on the overall flux linkage. $\boldsymbol{\psi}_{dq,\mathrm{dev}}$ depends on the currents $\boldsymbol{i}_{dq}$, the rotor angle φ_m, the angular frequency ω_m, the dc-link voltage V_dc, the component temperature, i.e., ϑ, ϑ_m, ϑ_inv, and the aging condition of the components, e.g., magnets, MOSFETs, diodes and shunts. Assuming correct V_dc and flux linkage maps—thus correct consideration of saturation effects—in the considered case the main influences on the flux linkage deviation originate from φ_m and ω_m, i.e., from spatial harmonics and iron losses, since temperature and aging affect quite slowly.

In contrast to the flux linkage correction term proposed here, e.g., a voltage correction term is introduced in [Ric16, p. 53]. However, a voltage correction term is particularly useful in a modulator-based method, since the controller output and thus the modulator input represents a voltage, enabling feed-forward control. Whereas with DMPC switch positions are optimized and thus the correction term is more flexible. Moreover, the voltage correction term is valid only for a certain speed, so slowly varying speeds must be assumed. Consequently, the procedure in [Ric16, p. 85] requires at least one electrical period at a new speed to ensure proper error compensation, or even less, if assumptions are used for the occurrence of certain harmonics.

However, for small electric drives, the speed may vary from zero to the rated speed during one electrical period. In addition, the flux linkage change in (7.1) depends on the rotor speed. If, e.g., ψ_{pm} would vary—illustrated in Fig. 2.10—as a function of rotor angle, this would lead not only to a position but also a velocity-dependent flux linkage deviation in (7.4). Thus, the flux linkage deviation is normalized to the speed, i.e., ω_{m}, in the following to decouple it from the speed prevailing at the time of measurement by using

$$\boldsymbol{\xi}_{\psi_{dq},\mathrm{dev}}(k-1) = \frac{\boldsymbol{\psi}_{dq,\mathrm{dev}}(k-1)}{\omega_{\mathrm{m}}(k-1)}\,. \tag{8.2}$$

To store this deviation as a function of the rotor position $\varphi(k-1)$, a suitable granularity for one mechanical revolution is required. The granularity depends on the order of the harmonics to be compensated. As exemplarily shown in Fig. 2.10, the spatial harmonics are dominant up to the 11^{th}-order. Since the 11^{th}- and 13^{th}-order in the abc-system result in the 12^{th}-order in the dq-system, $N_{\mathrm{ord}} = 12$ is chosen. Considering the pole pair number of the examined machines, see Table A.3, and at least 10 samples, i.e., $\frac{\pi}{5}$ rad, resolution for the highest electrical harmonic order, the number of samples per mechanical fundamental period can be calculated by using

$$N_{\mathrm{per}} = 10 N_{\mathrm{ord}}\, p\,. \tag{8.3}$$

After inserting concrete values for e.g. the motors M1 and M4 examined in the following, i.e., $p = 4$ and $N_{\mathrm{ord}} = 12$, this results in $N_{\mathrm{per}} = 480$. Since the measured rotor position $\varphi(k-1)$ will not exactly match one of these discrete position values, the respective deviation $\boldsymbol{\xi}_{\psi_{dq},\mathrm{dev}}(k-1)$ is stored for the nearest discrete position. To avoid the influence of outliers, i.e., to make the approach more robust, and to avoid a negative influence of the nearest position choice, the deviation is additionally averaged over several mechanical rotor revolutions N_{rev}[1]. Thus, to avoid unnecessary memory usage, a weighted averaging method is applied to $\boldsymbol{\xi}_{\psi_{dq},\mathrm{dev}}$, which results in

$$\bar{\boldsymbol{\xi}}_{\psi_{dq},\mathrm{dev}}(k-1) = \frac{N_{\mathrm{rev}}-1}{N_{\mathrm{rev}}}\bar{\boldsymbol{\xi}}_{\psi_{dq},\mathrm{dev}}(k-2) + \frac{1}{N_{\mathrm{rev}}}\boldsymbol{\xi}_{\psi_{dq},\mathrm{dev}}(k-1)\,. \tag{8.4}$$

Since the rotor position changes in each prediction step, i.e., $\varphi(k)\ldots\varphi(k+\mathrm{N_p})$, the corresponding $\bar{\boldsymbol{\xi}}_{\psi_{dq},\mathrm{cor}}(k)\ldots\bar{\boldsymbol{\xi}}_{\psi_{dq},\mathrm{cor}}(k+\mathrm{N_p})$ are calculated by linear interpolation from the two nearest memory values of $\bar{\boldsymbol{\xi}}_{\psi_{dq},\mathrm{dev}}(k)\ldots\bar{\boldsymbol{\xi}}_{\psi_{dq},\mathrm{dev}}(k+\mathrm{N_p})$. This yields

$$\bar{\boldsymbol{\psi}}_{dq,\mathrm{cor}} = \omega_{\mathrm{m}}\bar{\boldsymbol{\xi}}_{\psi_{dq},\mathrm{cor}}\,, \tag{8.5}$$

which is calculated from the stored—speed-independent and averaged—flux linkage deviation as well as the currently prevailing speed. Based on (7.4), the flux linkage prediction equation is

[1] In this work, $N_{\mathrm{rev}} = 6$ was chosen as a compromise between convergence speed and robustness.

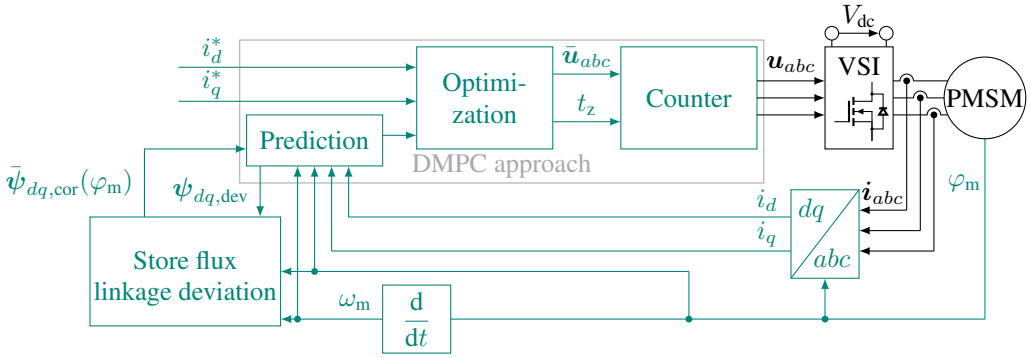


Figure 8.1: ψ-VSP^{2}CC with current reference tracking for a two-level VSI with a PMSM using repetitive flux linkage correction for damping of rotor position dependent effects.

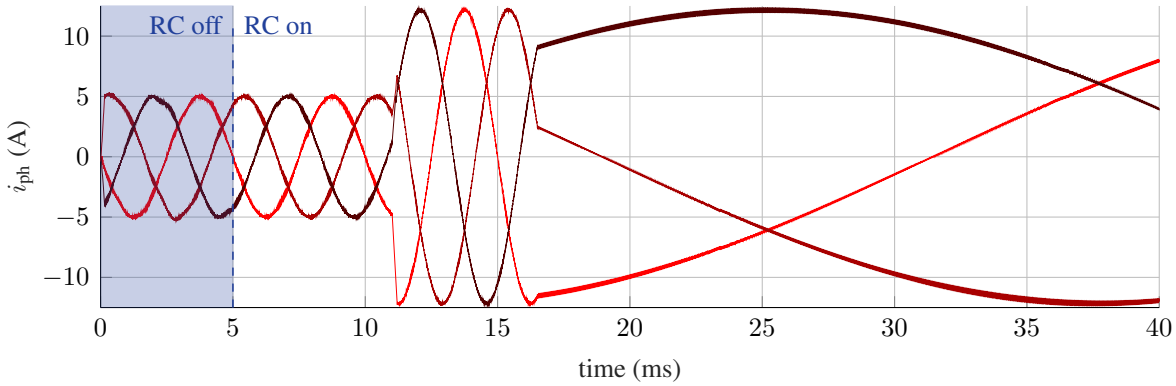
modified to include the flux linkage correction term $\bar{\boldsymbol{\psi}}_{dq,\mathrm{cor}}$, resulting in

$$\boldsymbol{\psi}_{dq}(k{+}1) \approx \boldsymbol{\psi}_{dq}(k) + T_{\mathrm{cf}} \frac{\boldsymbol{v}_{dq}(k) - R_{\mathrm{ph}} \boldsymbol{i}_{dq}(k) - \omega_{\mathrm{el}} \boldsymbol{P} \boldsymbol{\psi}_{dq}(k)}{1 + \frac{1}{4} T_{\mathrm{cf}}^2 \omega_{\mathrm{el}}^2} + \bar{\boldsymbol{\psi}}_{dq,\mathrm{cor}}(k+1) \tag{8.6}$$

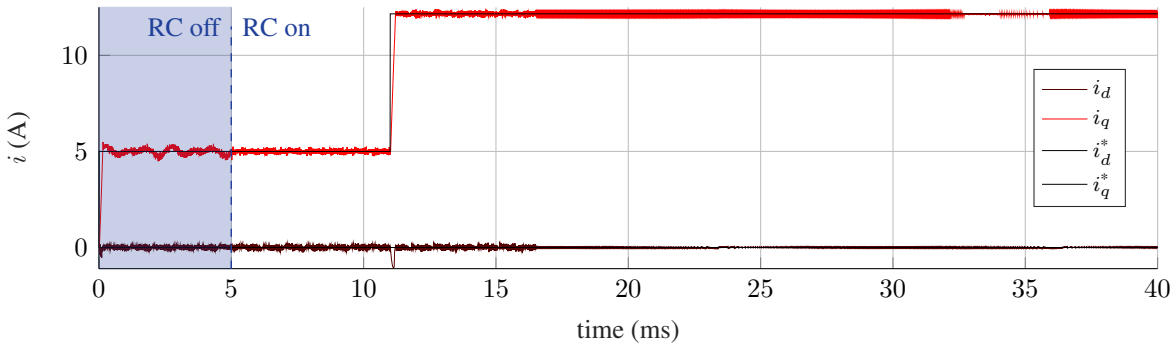
where $\bar{\boldsymbol{\psi}}_{dq,\mathrm{cor}} = [\bar{\psi}_{d,\mathrm{cor}}\ \bar{\psi}_{q,\mathrm{cor}}]^T \in \mathbb{R}^2$. Fig. 8.1 shows the data path for the proposed ψ-VSP^{2}CC with repetitive flux linkage correction. The additional block connected in parallel indicates the position-dependent identification and storage of the flux linkage deviation normalized to the speed. This allows to compensate for deviations of the flux linkage after at least one initial mechanical revolution even if the speed (and reference current) varies afterwards.

8.2 Performance Evaluation

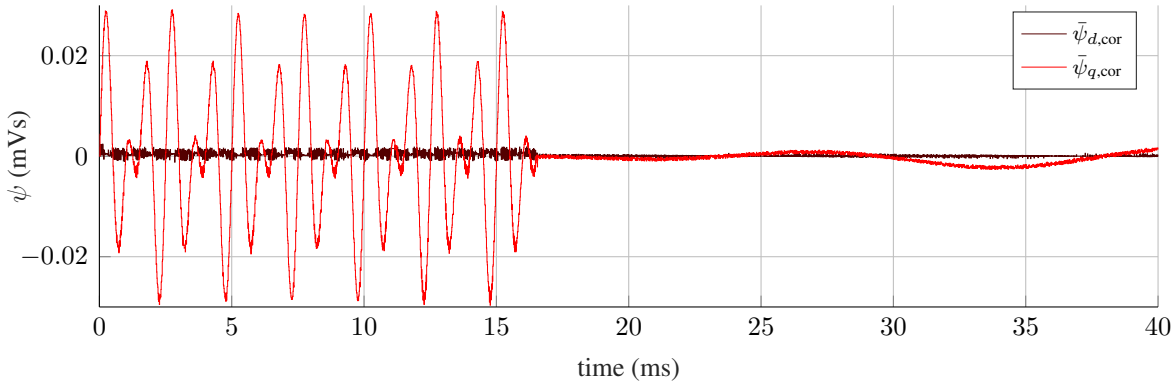
The two effects mainly addressed by RC are the rotor position dependency of the flux linkage and the iron losses. Both lead to a deviation in the prediction accuracy that becomes larger with increasing speed. Moreover, since the iron losses are current-dependent, they can of course only be fully compensated for in steady state, i.e., if $\boldsymbol{i}_{dq}$ remains approximately constant over N_{rev} rotor revolutions. However, the rotor position dependency has the most significant periodic influence in PMSMs and for the small machines considered, see, e.g., motor M4, even the most dominant influence which has not been considered so far. More precisely, for the evaluated machines the iron losses showed less influence than the spatial harmonics, since the former are quite small—see Section 4.4.2—and no steady-state error was evident. In contrast, spatial harmonics caused an unnecessarily high current THD, see, e.g., Fig. 7.9. Thus, the performance is first evaluated by adding in simulation the 4^{th} and 6^{th} spatial harmonic on ψ_{pm} with an amplitude of 20% of ψ_{pm} to emulate a disproportionately varying air gap over the rotor circumference in the dq-plane. Consequently, the proposed repetitive controller attempts to compensate for the



(a) Stator phase currents $\boldsymbol{i}_{abc}$.



(b) i_d and i_q current.



(c) $\bar{\psi}_{d,\mathrm{cor}}$ and $\bar{\psi}_{q,\mathrm{cor}}$ flux linkage correction terms for the prevailing speed.

Figure 8.2: Motor M1: ψ-VSP^{2}CC with repetitive flux linkage correction for $i_d^* = 0\,\mathrm{A}$, $i_q^* = 5.0\,\mathrm{A}$ at $t = 0{-}11\,\mathrm{ms}$ and $i_q^* = 12.16\,\mathrm{A}$ at $t = 11{-}40\,\mathrm{ms}$, $n_{\mathrm{m}} = 3000\,\mathrm{rpm}$ at $t = 0{-}16.52\,\mathrm{ms}$ and $n_{\mathrm{m}} = 200\,\mathrm{rpm}$ at $t = 16.52{-}40\,\mathrm{ms}$ with $N_{\mathrm{p}} = 2$, $f_{\mathrm{sw}} \approx 20.0\,\mathrm{kHz}$ (simulation).

rotor position dependency of the flux linkage to improve the current steady-state performance, as exemplarily shown for motor M1 in Fig. 8.2. As can be seen, the flux linkage deviation, see Fig. 8.2(c), has been learned in the first periods and can be feed-forwarded after activation both in transients as well as at varying currents and speeds. Without compensation, i.e., $t = 0{-}5\,\mathrm{ms}$ in Fig. 8.2(b), a clear oscillation is visible in the dq-current, even if the current control runs at $f_{\mathrm{cf}} = 100\,\mathrm{kHz}$. As ψ_{pm} was modified, it is obvious that the largest deviation in the predicted flux linkage is in the q-axis and dependent on the prevailing speed, see Fig. 8.2(c).

Considering the results in Fig. 7.9 for motor M4, pronounced low order harmonics, i.e., the 2^{nd}, 4^{th}, 5^{th} and 7^{th}, can be noticed. These are due to the effects mentioned above, which are not known to the prediction model and are also very difficult to model, where the added value of modeling compared to the effort (in terms of computational effort) is questionable. However, using the proposed RC method, these harmonics can be significantly damped while keeping the average switching frequency the same and computational effort low.[2] This advantageous behavior is illustrated in Fig. 8.3, where the measurements without RC are also shown for comparison.

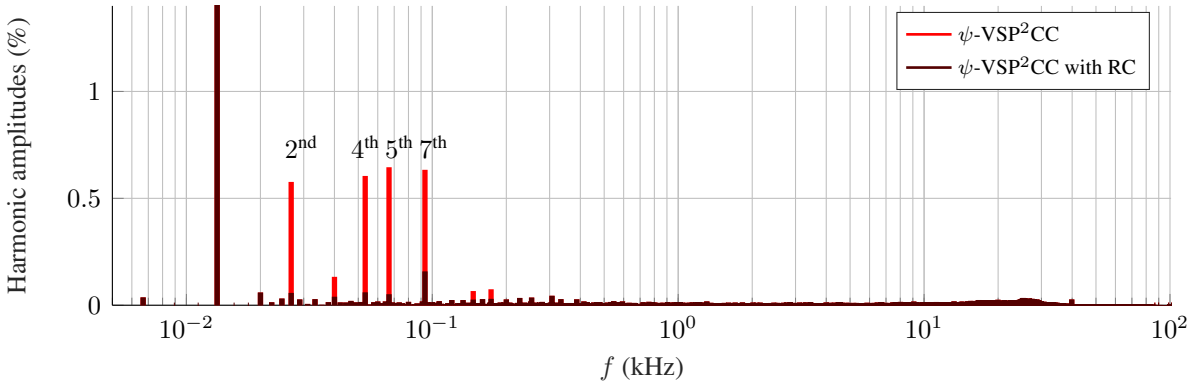


Figure 8.3: Motor M4: Stator current spectrum for ψ-VSP^{2}CC without repetitive correction ($I_{\text{THD}} = 1.90\,\%$) see Fig. 7.9(d) and with repetitive correction ($I_{\text{THD}} = 1.03\,\%$) for $i_d^* = -5.0\,\text{A}$, $i_q^* = 14.0\,\text{A}$, $n_{\text{m}} = 200\,\text{rpm}$ with $N_{\text{p}} = 2$, $f_{\text{sw}} \approx 10.0\,\text{kHz}$ (experimental).

8.3 Final Assessment

In summary, when a flux linkage deviation occurs that varies with the rotor position, the speed (e.g., in case of iron losses) or the temperature, the proposed RC method allows to compensate for these slow varying model and parameter discrepancies when using a direct control approach. This results in an improved steady-state performance, i.e., reference tracking accuracy and minimization of the current THD. Even though the proposed method optimizes the steady-state performance of the current, the same approach can be used to optimize the torque, i.e., antivalent deformation of the current, so that a uniform torque is obtained after taking into account the non-uniform flux linkage, see (2.26).

Although not discussed here, the RC method can also be used for L-VSP^{2}CC or conventional DMPC. More precisely, if saturation and cross-coupling do not dominate, i.e., can be neglected, and thus the absolute inductance can be assumed to be valid for all operating points (linear magnetic circuit, i.e., linear relationship between current and flux linkage), then the current can be used directly instead of the flux linkage for the proposed position-dependent correction.

[2] For comparison, an RC method for a PI controller using a modulator is shown in [Ric16, p. 119].

9 Cascaded Model Predictive Control for Mechatronic Drive Systems

Given the specific application of an electric drive, multiple control objectives are usually required. In addition to the highly dynamic current control described in the previous chapters, for example, speed or position control may also be required. Taking these requirements into account, a mechatronic drive system can be assumed that contains both mechanical and electrical time constants, which are usually in different ranges. More specifically, the ratio between the two is typically in the range of 5-20[1], although in some small electrical drives the mechanical time constant can be almost as small as the electrical one, so that the ratio is nearly one.
Nevertheless, most applications allow the use of cascaded control structures in order to separate mechanical and electrical control objectives. This separation is common, for example, in conventional FOC with cascaded PI controllers, see Section 2.3, or DTC with a superimposed PI controller, see Section 2.4. For these standard methods, the reason for cascading is mainly due to the limited options when using SISO control structures. However, in case of predictive control, the MIMO control structure allows the simultaneous control of multiple control objectives [RKE+13]. Thus, it is theoretically possible to include all control objectives in one single cost function, thereby avoiding a cascaded structure. Since this possibility seems very promising, it was evaluated with numerous predictive control approaches. For example, [PB13b, FKRK14] propose cascade-free current and speed FCS-MPC.

Nevertheless, given the predictive nature and optimization of several criteria with different time constants, it is necessary to predict a noticeable change for each criterion. If no change in the system states can be predicted—because the considered time range in the prediction process is too short—a predictive optimization seems to be of no use. Consequently, the prediction and optimization of current and speed—considering the reduction of mechanical natural oscillations claimed in Section 1.2—require both a very long horizon and a very high granularity in order to keep the current ripple acceptable on one side and to be able to predict a change in speed on the other side.

[1] The electrical time constant of a small electric drive (see Section 2.2.1.3) is usually in the range of a few hundred microseconds to a few milliseconds, while the mechanical time constant (see (2.32)) is in the range of a few tens of milliseconds.

The use of FCS-MPC, as discussed in the previous chapters or as used in the aforementioned cascade-free FCS-MPC approaches, allows direct optimization of switch positions as well as utilization of the maximum available dc-link voltage, thus providing the highest dynamic range, but is also limited by the exponential growth of the computational effort to several discrete time steps into the future, e.g., 5-10. Significantly extended prediction horizons would impede an experimental implementation of FCS-MPC, even if the search space could be reduced as described in Chapter 5. Although ongoing research is dedicated to this problem, in particular through the use of sphere decoding, it is not foreseeable that the implemented FCS-MPC horizon will change significantly in the coming years. In addition, the linearization remains a problem when using the SDA for small drives, as explained in Chapter 5.
Even if the accessible horizons are completely sufficient for the electrical controlled system—see previous chapters or [KG20]—the prediction of mechanical transient processes requires a longer horizon or enlarged time steps, i.e., lower calculation frequency, compared to current control. In other words, if natural oscillations of the mechanical speed are to be damped, this is probably not predictable with a short horizon. For an optimal prediction the whole transient process would be useful. The question therefore arises as to how this can be achieved.

Long prediction intervals In [KGN+14] a review of strategies for long prediction intervals is presented. The review points out three possibilities for FCS-MPC such as the move blocking strategy, the extrapolation strategy and the event-based horizon. Move blocking divides the prediction horizon into two parts with a discrete number of steps N_{p1} and N_{p2}. The first part has a length of $N_{\mathrm{p1}}T_{\mathrm{cf}}$ and the second part of $N_{\mathrm{p2}}n_2T_{\mathrm{cf}}$ where $n_2 \in \mathbb{N}^+$. However, this method uses either FCS-MPC [Aya17, p. 80] or CCS-MPC for the entire prediction. Using the former, with the extrapolation strategy all switching possibilities are calculated only for a short switching horizon (e.g., two) and afterwards a trajectory for each discrete switching possibility, starting from the last predicted step, is calculated and compared to a hysteresis band. The hysteresis bands are considered as soft constraints. The event-based horizon method is a combination of MPC and optimal pulse patterns.
In summary, while the above approaches allow for consideration of a longer time period with the same number of prediction steps, they still optimize discrete switch positions. However, the discrete switch positions have a direct—thus beneficial—influence on the electrical controlled system, but no direct influence on the mechanical system, i.e., speed or position. Thus, FCS-MPC seems unsuitable for the holistic prediction of the entire controlled mechatronic system. Due to the fact that the electrical and mechanical time constants can be clearly separated into two different time domains, a separation of these time domains seems to be useful for MPC despite the possibility of MIMO optimization. For this reason, it seems reasonable to superimpose a more suitable controller to FCS-MPC, i.e., to cascade it, in order to enable predictive control

for the entire mechatronic system. This also avoids the problem that setting weighting factors for criteria from different time domains and magnitudes is very challenging, as can be observed, e.g., in [FSY12].

Cascaded control structures Using the idea of a cascaded control structure, [FRS+09] and [GRS+14] respect a separation of the electrical and mechanical time domains by combining FCS-MPC with a superimposed PI-based and dead-beat speed controller, respectively. Even though dead-beat has a single-stage predictive character, it is not possible to predict the course of oscillations in the mechanical system. Thus, oscillations are not predictable and consequently not avoidable. [WNB19] shows a cascaded long-horizon prediction, where both the mechanical and electrical control loops utilize CCS-MPC. Similarly, [OKBH14, p. 249 ff.] also uses CCS-MPC for speed control, but with an underlying PI current control. Unfortunately, both methods do not take advantage of the direct nature of FCS-MPC. [HYZ+15] shows a hierarchical MPC with a CCS-MPC power controller and an FCS-MPC current controller for a multilevel matrix converter. However, no mechanical controlled system is considered here. In [ABY18] a cascaded structure of CCS-MPC and FCS-MPC—but with programmed modulator—is proposed, however, since the modulator stage is separated, direct control behavior is not used. A general discussion of hierarchical MPC structures is presented, e.g., in [SC07], which recommends multiple layers to account for the different time constants in the controlled system.

9.1 Cascaded Continuous and Finite Control Set-MPC

As mentioned in [Sch15, p. 422], predictive current control enables fast current dynamics and thus allows a highly dynamic superimposed speed or position control. However, none of the aforementioned cascaded concepts allows the simultaneous use of FCS-MPC and a long prediction horizon for the entire mechatronic system.

To accomplish this, a novel cascaded continuous and finite control set-model predictive control (CCF-MPC) algorithm is described and experimentally verified for mechatronic drive systems. This approach is advantageous for speed control of electrical drives in mechatronic systems with equally high requirements on the electrical and mechanical controlled system. CCF-MPC combines FCS-MPC and CCS-MPC in a cascaded structure regarding the electrical and mechanical controlled system and their respective time constants. In this way, CCF-MPC allows a compromise between the discrete granularity of the prediction step size and the possible length of the prediction horizon—see Fig. 9.1—considering only a slightly increasing computational load. This is in contrast to the previously mentioned approaches.

First, CCF-MPC takes advantage of the direct nature of FCS-MPC. More precisely, it enables high current dynamics by exploiting the dead-beat behavior, direct consideration of constraints,

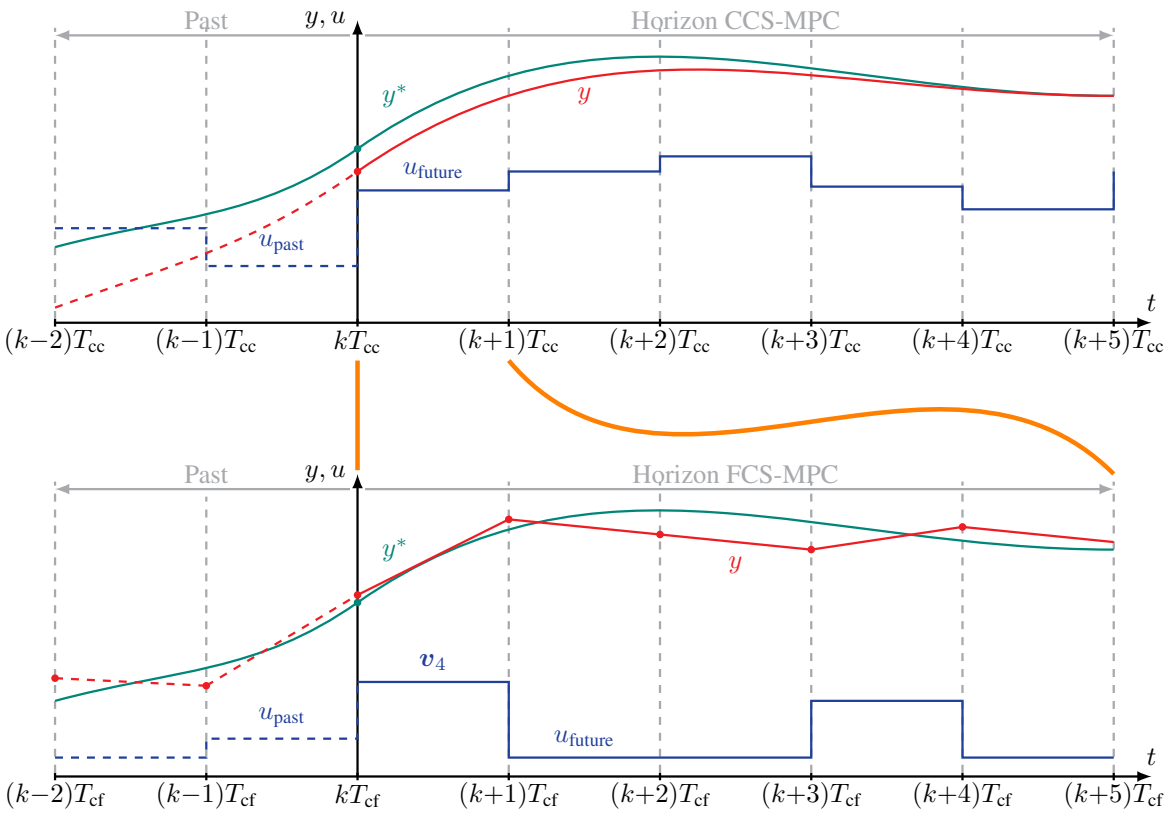


Figure 9.1: CCF-MPC with receding horizon policy for a five-step prediction horizon, i.e., $N_{\text{p}} = 5$, respectively. y are the resulting controlled signals, y^* are the reference signals and u are the discrete-time control signals, i.e., T_{el}^* for CCS-MPC and the switch positions for FCS-MPC. For the latter, v_4 is chosen exemplarily at the beginning.

optimization of the steady-state current behavior indicated by the current THD, and a minimized control deviation by using a long horizon (see Chapter 7).[2]

Second, by integrating CCS-MPC, CCF-MPC enables a predictive calculation of changes in the mechanical controlled system and thus reduced speed or position overshoots as well as an active damping of mechanical oscillations in the load speed, i.e., torsional oscillations. This is beneficial for the predictive overall optimization of mechatronic systems, i.e., two-mass systems, that require highly dynamic but also highly precise position and speed control, e.g., for electrical drives in machine tools or for end-of-line test systems in the field of window regulator motors.

Since CCS-MPC does not have the direct capability of FCS-MPC, but allows a longer prediction horizon due to the tendency of lower computational effort (see Section 2.6.4), cascading both schemes seems reasonable. This allows the best of both MPC methods—see Section 2.5—to be used, i.e., combined. In addition, as shown in Chapter 3, the use of heterogeneous computation platforms is advantageous for the implementation of the heterogeneous CCF-MPC approach. More specifically, FCS-MPC is computed on the FPGA with f_{cf}—see previous chapters—and

[2]Note that due to the high time granularity of the control interval (e.g., $T_{\text{cf}} = 10\,\mu\text{s}$), a linear current gradient is assumed for each individual prediction step of FCS-MPC, see Section 6.1.

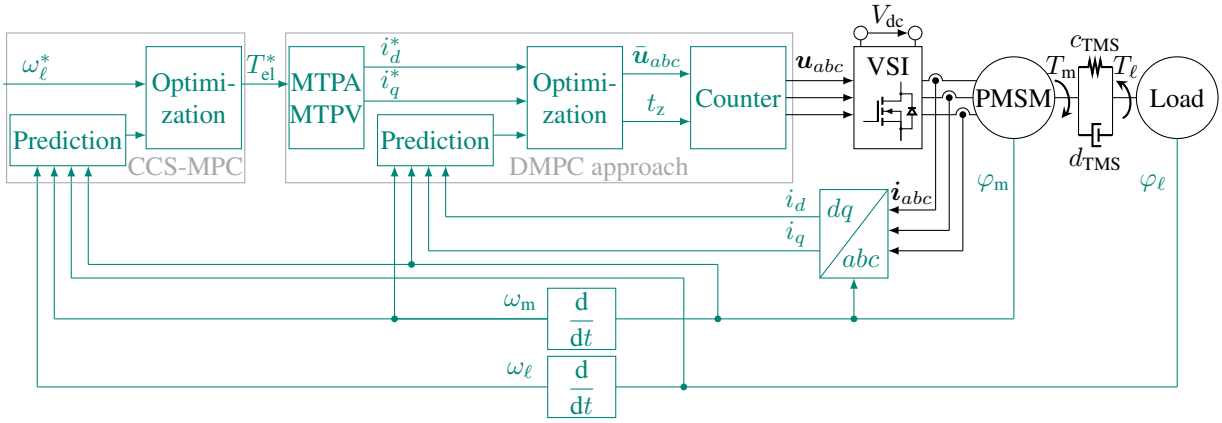


Figure 9.2: Control loop structure of the proposed CCF-MPC.

CCS-MPC is computed on the processor, since a lower control frequency, i.e., f_{cc}, can be used. This algorithm splitting is illustrated in Fig. 9.2. More in detail, Fig. 3.1 shows the implementation and Fig. 3.2 the corresponding temporal interaction. The basic idea and princible of CCF-MPC was first published in [WHKD20, WHKDK18, WHKDK19].

9.2 Definition of Superimposed CCS-MPC

In this thesis, as described in Section 2.2.2, a TMS is considered for the mechanical controlled system. Moreover, the focus is on speed control with ω^*, i.e., ω_{m}^* or ω_ℓ^*, rather than position control. Using CCS-MPC—see Section 2.5.1—for the optimization of the TMS, the required LTI system is described by

$$\underbrace{\begin{bmatrix} \frac{\mathrm{d}\omega_{\text{m}}(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}\Delta\varphi(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}\omega_\ell(t)}{\mathrm{d}t} \end{bmatrix}}_{\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t}} = \underbrace{\begin{bmatrix} -\frac{d_{\text{TMS}}+d_{\text{fric}}}{J_{\text{m}}} & -\frac{c_{\text{TMS}}}{J_{\text{m}}} & \frac{d_{\text{TMS}}}{J_{\text{m}}} \\ 1 & 0 & -1 \\ \frac{d_{\text{TMS}}}{J_\ell} & \frac{c_{\text{TMS}}}{J_\ell} & -\frac{d_{\text{TMS}}}{J_\ell} \end{bmatrix}}_{\boldsymbol{A}_{\text{c}}} \underbrace{\begin{bmatrix} \omega_{\text{m}}(t) \\ \Delta\varphi(t) \\ \omega_\ell(t) \end{bmatrix}}_{\boldsymbol{x}(t)} + \underbrace{\begin{bmatrix} \frac{1}{J_{\text{m}}} \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{B}_{\text{c}}} \underbrace{T_{\text{el}}(t)}_{u(t)} \tag{9.1a}$$

$$\boldsymbol{y}(t) = \begin{bmatrix} \Delta\varphi(t) \\ \omega_\ell(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{C}_{\text{c}}} \boldsymbol{x}(t)\,, \tag{9.1b}$$

where the torque is the decision variable, i.e, control signal. The latter serves as reference for the subordinate controller of the electrical system. More in detail, as already mentioned, this chapter deals with the optimization of the load speed, i.e., ω_ℓ, in order to control highly dynamic speed

changes with low overshoots. Hence, in Fig. 9.2 ω_ℓ^* is given, where ω_m^* could be used in the same way. On top of that, an active damping of possible mechanical oscillations in the drive train is achieved by minimizing $\Delta\varphi = \varphi_\mathrm{m} - \varphi_\ell$, where φ_m is the angle of the motor shaft and φ_ℓ the angle of the load. This suppresses mechanical natural oscillations in the TMS. Considering the augmented model in Section 2.5.1.3 and the goal to optimize the load speed ω_ℓ and minimize the angle deflection $\Delta\varphi$, the weighting matrix is chosen with

$$\boldsymbol{C}_\lambda = \begin{bmatrix} 0 & 0 & 0 & \lambda_{\Delta\varphi} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{\omega_\ell} \end{bmatrix} \tag{9.2}$$

where λ defines which criterion of the TMS is prioritized.

9.2.1 Defining Constraints

As explained in Section 2.5.1.1, constraints can be included directly in the cost function. The definition of these constraints is explained in Section 2.5.1.4. Depending on the application, e.g., the maximum speed can be limited as an output constraint or the change in torque as an input constraint. In the case of classical SISO controllers, e.g., FOC, if numerous constraints have to be considered, this can often become very challenging since several feedbacks are present and thus cases and prioritizations have to be covered. For CCS-MPC, it is significantly easier to take such constraints into account. Notwithstanding this, just the maximum torque, $T_\mathrm{el,max}$, is limited in the following. This constraint can be defined by a rough calculation in terms of the maximum allowable current, i.e., $T_\mathrm{el,max} = \frac{3}{2} p \psi_\mathrm{pm} \sqrt{2} I_\mathrm{max}$, neglecting the reluctance. Alternatively, the maximum permissible torque of the weakest component in the drive train can be used to take into account mechanical limitations of individual components, such as the torque limit of a measuring shaft.
Consequently the constraints can be described based on (2.61) with

$$-\boldsymbol{U}_\mathrm{min}(k) = \boldsymbol{U}_\mathrm{max}(k) = \left[T_\mathrm{el,max}(k)\, T_\mathrm{el,max}(k+1) \ldots T_\mathrm{el,max}(k+N_\mathrm{c}-1)\right]^T \in \mathbb{R}^{N_\mathrm{c}\times 1} \tag{9.3}$$

and the lower triangular matrix

$$\boldsymbol{M}_{I,ij} = \begin{cases} 1 & \text{if } i \geq j \\ 0 & \text{if } i < j \end{cases} \in \mathbb{Z}^{N_\mathrm{c}\times N_\mathrm{c}} \tag{9.4}$$

with i, j counting up until the size of N_c. In doing so, the control signal $u(k)$ is not simply clamped to the value $T_\mathrm{el,max}$ as in the case of a PI-based controller or even just saturated as in the

case of LQR and the unconstrained solution of CCS-MPC (see Section 2.5.1.3), but rather considered as a constraint during optimization. Consequently, a predicted excitation of a possible oscillation in the controlled system is not even excited. This means that, in comparison to the unconstrained solution, the constrained solution already takes into account during optimization (planning) that limitations must be met and therefore certain control signals cannot or should not be produced. For example, the constrained solution is aware of the fact that a supposedly infinite torque cannot be output in order to later compensate for an initially excited oscillation. Therefore, the constrained solution already excites this oscillation less, because it knows that it cannot damp it sufficiently later.
In the end, the reference can be reached faster and with less oscillations in the TMS by using a better—in this case reduced—planned control signal, i.e., T_{el}^{*}, as will be shown in the following sections. Thus, the distinction made in Section 2.5.1 between $\mathrm{CCS}_{\mathrm{unc}}$-MPC, which does not consider constraints, and $\mathrm{CCS}_{\mathrm{con}}$-MPC, which considers constraints and must be solved numerically, is of great importance.

9.2.2 Defining Hildreth's Method

Considering the controlled mechanical system, i.e., (9.1), the weighting factors (9.2) and the defined constraints, i.e., (9.3), a constrained solution, i.e., $\mathrm{CCS}_{\mathrm{con}}$-MPC, must be determined. Hence, based on the analytically solved unconstrained solution in (2.58), a numerical solution must be calculated, as explained in Section 2.5.1.4. FGM [PBD13] and Hildreth's Method—see Section 2.5.1.4—have been considered for this purpose. However, FGM has some disadvantages compared to Hildreth's method in case of speed control. Due to the lack of an integrator, FGM relies on using an observer or other methods of parameter tracking to compensate for parameter deviations and disturbances. Furthermore, in the ideal case, a reference value must be specified for each state, which is not possible for the control of elastically coupled TMS. Consequently, Hildreth's method is applied in the following to numerically calculate the constrained optimal torque.
To implement this general approach in real time, the numerical method in (2.72) is limited to $k = 10$ iterations. In doing so, the ISR runtime is only moderately loaded (see Section 9.4.2), since additional computing capacity is required for updating the state matrices in Section 9.2.3, i.e., $\boldsymbol{\Upsilon}_{\mathrm{c}}$ and $\boldsymbol{\Gamma}_{\mathrm{c}}$, as well as further time-uncritical tasks, see, e.g., Chapter 4.
The optimization provides the constrained solution for the torque optimization problem, i.e., $\Delta \boldsymbol{u}_{\mathrm{con}}$, where $\boldsymbol{u}_{\mathrm{con}}$ is limited to $T_{\mathrm{el,max}}$.
As a side comment, as can be seen from the defined system, i.e., (9.1), and the constraints, i.e., (9.3), the number of constraints may be equal or larger than the number of decision variables,

i.e., $\Delta \boldsymbol{u}$. Thus, the variant shown in Section 2.5.1.4 to solve the quadratic problem analytically in the Lagrange representation with equality constraints is unsuitable for two reasons. First, having more constraints than decision variables does not lead to a feasible solution that satisfies the constraints. Second, if the number of equality constraints equals the number of decision variables, the only feasible solution is the one that satisfies the constraints, hence there are no degrees of freedom left that can be used to minimize the cost function. For the presented use case of speed control, where the torque is limited by $|T_{\mathrm{el,max}}|$, this would mean that all $\Delta \boldsymbol{U}_{\mathrm{con}}$ (based on ρ) are optimized to the maximum permissible torque. As a result, the torque would no longer fulfill the actual goals of the cost function, namely reference tracking.
Consequently, to avoid feasibility problems, it is justified and reasonable to use inequality constraints for the proposed control target, although this involves a numerical optimization.

9.2.3 Online Recalculation of Control Matrices

As explained in Section 2.5.1, the use of CCS-MPC requires an LTI system, which is not a problem as long as the controlled system can be assumed to be (partially) linear. For the TMS considered in this chapter, this can also be assumed for the spring constant and damping. However, the moment of inertia can vary significantly under different load conditions. In addition, friction varies with temperature. Nevertheless, the resulting mechanical nonlinearities have quite slow time constants. Consequently, the idea of [Rau03] can be taken up and adapted.

In doing so, the slow time-varying parameters, e.g., friction, can be identified and adapted as explained in Section 4.2. Consequently, the system and input matrix in (9.1a) can be updated during operation. Afterwards, $\boldsymbol{\Upsilon}_{\mathrm{c}}$ and $\boldsymbol{\Gamma}_{\mathrm{c}}$ are recalculated—but in a time-uncritical manner—based on the procedure described in Section 2.5.1.3. In addition, the weighting matrix $\boldsymbol{R}_{\mathrm{v}}$ can be modified during this recalculation. Finally, the new matrices are stored in the RAM. The prediction matrices are then replaced at the beginning of the following ISR call.[3]

9.2.4 Selection of Weighting Factors and Prediction Horizon

Regarding the weighting factors, a heuristic adjustment method is chosen, which is used for both $\mathrm{CCS_{con}}$-MPC and $\mathrm{CCS_{unc}}$-MPC. First, the weighting factor of the manipulated variable, i.e., r_{v}, is chosen so that the controllers are stable at the maximum speed reference used. If the

[3] In the currently implemented algorithm, this recalculation and update occurs every 60 s, where this time period was chosen heuristically. Considering the slowly changing mechanical parameters, e.g., friction, this update rate has proven to be completely sufficient in the application demonstrated.

maximum speed reference leads to an oscillation, i.e., possible instability, r_v must be increased. Afterwards, the weighting factors of the output matrix $\boldsymbol{C}_\lambda$ are chosen. Since only the ratio between the different optimization criteria is relevant, i.e., determines how strongly which of the optimized criteria is prioritized, $\lambda_{\omega_\ell} = 1$ is set first. In a second step, $\lambda_{\Delta\varphi}$ is increased. If a too large $\lambda_{\Delta\varphi}$ is chosen, hardly any oscillations occur, but this can lead to a permanent tracking error as soon as the drive system is loaded. As mentioned in the previous section, the weighting factors and thus their ratio can be changed during operation if necessary. However, this is not done in the following.

$N_\mathrm{c} = 10$ and $N_\mathrm{p} = 35$ are determined heuristically as described in [WHKDK19]. Here, the ITAE criterion—see (2.90)—was determined for a variety of possibilities with $N_\mathrm{p} \in [10, 50] \subset \mathbb{N}$ and $N_\mathrm{c} \in [1, 50] \subset \mathbb{N}$, where the minimum was selected afterwards. Moreover, due to the limited computational capacity in one ISR cycle, the online adjustable horizon is limited to $N_\mathrm{c} \leq N_\mathrm{p} \leq 10$ and $N_\mathrm{p} \leq 35$ for the experimental evaluation, see Section 9.4.2. Thus, $N_\mathrm{c} = 10$, $N_\mathrm{p} = 35$, $r_\mathrm{v} = 100000$, $\lambda_{\omega_\ell} = 1$, and $\lambda_{\Delta\varphi} = 300$ are chosen for the following evaluations.

9.3 Transition between the Cascaded Methods

Basically, two possibilities were considered to implement the transition between CCS-MPC and FCS-MPC.

1. CCS-MPC optimizes and outputs a torque that is directly used as reference for a subordinate FCS-MPC torque control. Thus MTPA/MTPV is an inherent part of the FCS-MPC cost function.
2. CCS-MPC optimizes and outputs a torque, which is forwarded to an independent MTPA/MTPV optimization. The latter provides the reference currents for a subordinate FCS-MPC current control.

In both cases, however, MTPA/MTPV must be taken into account so that a loss-minimized torque can be generated, i.e., only the current heat losses are minimized here.
The first possibility was examined, e.g., in [PB13c] with MTPA and in [PB13d] with MTPV. In case of MTPA, the author has two criteria in the cost function. The first relates to the difference between the measured and the reference torque

$$c_\mathrm{T}(k) = \Big(T_\mathrm{el}(k) - T_\mathrm{el}^*(k)\Big)^2 . \tag{9.5}$$

The second is about an MTPA attraction term, i.e.,

$$c_{\mathrm{A}}(k) = \left(i_d + \frac{L_d - L_q}{\psi_{\mathrm{pm}}}(i_d^2 - i_q^2)\right)^2, \tag{9.6}$$

which selects the switching state with minimal total current. However, this concept has two terms and the choice of the weighting factor between them is quite challenging. Especially when electrical drives with different torque and current ranges are considered, the weighting factors have to be adjusted frequently.
This problem was also examined in own considerations, see [Sch19]. Furthermore, the mentioned approach assumes a linear magnetic circuit, which, as shown in Chapter 7, is mostly not valid and also has a noticeable impact on loss minimization, as evaluated, e.g., in [RTAG16, BK18]. Finally, the introduction of a variable switching point, see Chapters 6-7, that minimizes the current ripple is completely impossible, since a reference current—and not the reference torque—must be optimized. If the torque ripple would be minimized, see, e.g., [KSK+14, AKMK15, KAK18], this does not automatically mean that the current ripple is also minimized. On the other hand, the reverse can be guaranteed [Hol16]. Since the focus is not only on a smooth torque but also on minimizing current heat losses, the harmonics of the current—indicated by the THD—must be reduced, as shown in Chapters 6-7.
Therefore, the second possibility is chosen, i.e., the use of an independent MTPA/MTPV, to calculate the reference currents based on the reference torque with the help of (2.26).

Reference currents based on the linear torque equation When assuming a linear magnetic circuit, the reference currents in d- and q-axis can be calculated—see, e.g., [RTAG16], [Sch15, p. 1095 ff.]—by using

$$i_d = -\frac{\psi_{\mathrm{pm}}}{2(L_d - L_q)} - \sqrt{\frac{\psi_{\mathrm{pm}}^2}{4(L_d - L_q)^2} + {i_q}^2}\,. \tag{9.7}$$

As shown, e.g., in [RTAG16, BK18], this is only valid if the inductances do not exhibit saturation and cross-coupling effects and the flux constant has not been changed by temperature.

Reference currents based on the nonlinear torque equation As highlighted in the previous chapters, magnetic nonlinearities can no longer be neglected due to the increasing use of highly utilized synchronous machines. Thus, in order to avoid time-varying nonlinearities for L_d, L_q and ψ_{pm}, [RTAG16] shows an extended MTPA (EMTPA) calculation by using the Lagrangian function. Other approaches that consider magnetic nonlinearities for MTPA can be found, e.g., in [BK18] or, if iron losses are also to be considered, in [HKM20].

Since the flux linkage maps are present anyway—see Section 4.3.4—it is recommended to use the latter, i.e., EMTPA, to allow for a minimization of the copper heat losses even for nonlinear magnetic circuits.
However, since this was not the research focus of this work, only the standard MTPA will be used in the following, which is valid since the evaluated motor, i.e., motor M5, has a fairly linear magnetic behavior.

For the transition between CCS-MPC and FCS-MPC, it is also of interest how the concept behaves as soon as the mechanical time constant is approximately as small as the electrical time constant, i.e., $\tau_{\mathrm{el}} \approx \tau_{\mathrm{m}}$. In this case, the control frequency f_{cc} can be similar to f_{cf}.
For comparison, in conventional PI-based control schemes, a ratio of $f_{\mathrm{cf}}/f_{\mathrm{cc}} = 5-10$ is usually recommended to avoid stability problems, so that the discrete-time current controller can quasi-continuously follow the torque reference provided by the discrete-time speed controller. This is mainly due to the fact that the symmetrical optimum and modulus optimum methods mentioned in Section 2.3 are based on continuous-time.
For CCF-MPC, since there is no physical integrator present, the constraints are already part of the optimization, and the controller design is performed in the discrete-time domain, no such stability problems have been encountered even if $f_{\mathrm{cc}} \approx f_{\mathrm{cf}}$. This is mainly due to the direct consideration of the constraints, i.e., in the specific case the reference torque, when using $\mathrm{CCS}_{\mathrm{con}}$-MPC. However, since none of the experimental test setups had such close time constants, this could only be validated in simulation.

9.4 Performance Evaluation

In the following, CCF-MPC is examined using the example of motor M5, see Table A.3, on the test bench described in Section 3.3, whose TMS parameters are shown in Table A.4 of Appendix A.3. Moreover, for a meaningful comparison, similar to the previous chapters, several comparable approaches for the cascaded control structure are considered. For the current control, the PI-based control (Section 2.3), DTC (Section 2.4)—as another direct control approach—and the proposed VSP^{2}CC (Chapter 7) are compared.[4] For the superimposed speed control, the PI-based control[5] (Section 2.3), LQR[6] and both of the proposed unconstrained

[4]Even though the conventional FCS-MPC (Section 2.5.2) shows similar transient behavior, due to the clearly better steady-state behavior of VSP^{2}CC—as shown in the previous chapters—only the latter is shown in the following. However, both have been validated and compared in [HK20].

[5]The setting of the controller gains in case of DTC is explained in Appendix A.6.

[6]The LQR design is based on [Lun17, p. 304], with adjustments for the specific use case of a TMS from [JS95], adding an additional integrator for steady-state accuracy.

Table 9.1: Evaluated cascaded control structures and their respective control frequencies.

Acronym	Speed control	Current / torque control	f_a (kHz)	f_{cf} (kHz)	f_{cc} (kHz)
FOC	PI	PI	20	20	10
PI-DTC	PI	DTC	100	100	10
PI-DMPC	PI	VSP^{2}CC	100	100	10
LQR	LQR	VSP^{2}CC	100	100	10
CCF$_{unc}$-MPC	CCS$_{unc}$-MPC	VSP^{2}CC	100	100	10
CCF$_{con}$-MPC	CCS$_{con}$-MPC	VSP^{2}CC	100	100	10

CCS$_{unc}$- and constrained CCS$_{con}$-MPC are compared. Since in the field of MIMO optimization problems LQR—as a method of optimal control—is very common [Lun17, p. 291 ff.], especially for mechanical controlled systems, it is used as a reference for the desired highly dynamic speed control. The investigated combinations of the mentioned approaches and the respective sampling and control frequencies are summarized in Table 9.1. Furthermore, for the comparison of the TMS speed controller it must be noted, that LQR and CCS-MPC as MIMO controllers optimize the load speed, minimize the angle difference between motor and load shaft and respect the motor speed. On the other hand, the three approaches with PI-based speed controller—since they are SISO controllers—optimize just the motor speed.[7]

Parameter selection For CCS-MPC, it is necessary to parameterize the drive system as a TMS, requiring a spring stiffness c_{TMS}, a damping d_{TMS} and the moments of inertia, as outlined in Section 9.2. The same parameters are required for LQR, whereas the moments of inertia are sufficient for FOC, PI-DTC and PI-DMPC. All parameters have been calculated analytically and identified using the offline ID described in Section 4.2, where the results were quite similar in both cases. Regarding the spring stiffness c, it should be noted that although in a drive train each component has its own spring stiffness, in a TMS one component is the dominant elastic one. Thus, the spring stiffness of the shafts can be considered rigid, i.e., $c \to \infty$, compared to the spring stiffnesses of the couplings, and the latter in turn can be considered rigid compared to the torque measurement shaft. Hence, c of the torque measurement shaft is approximately equal to c_{TMS}. The individual c values can be found in Appendix A.3.2.
With the assumption of a TMS, the moments of inertia on the right and left side of the measuring shaft can be summarized, see Appendix A.3.2 for details. The damping of the TMS has been calculated analytically with (2.35) and identified by means of the parameter ID in Section 4.2.2. The parameter ID procedure in Section 4.2.1 was used to identify the friction. All TMS parameters are summarized in Table A.4 of Appendix A.3.2.

[7] It has also been investigated that the PI-based controller directly controls the load speed instead of the motor speed. Unfortunately, this caused strong oscillations and eventually unstable behavior.

Evaluation As mentioned in Section 9.2, in case of TMS speed control, the speed of the load, i.e., n_ℓ, is the state of interest, i.e., is optimized, while the motor speed is not of interest. A uniform test profile for n_ℓ^* is used for the subsequent evaluation of the control methods, see Fig. A.16 of Appendix A.6. First, at time $t = 0.01\,\mathrm{s}$, a speed reference step is applied from zero to the desired reference speed. Second, at $t = 0.2\,\mathrm{s}$ the drive is loaded with a load torque $T_\mathrm{L} = 0.2\,\mathrm{Nm}$, i.e., almost half of the nominal torque, and is removed at $t = 0.3\,\mathrm{s}$. Third, at $t = 0.4\,\mathrm{s}$ a speed reversal is performed, i.e., $-n_\ell^*$ is set as reference speed. For n_ℓ^*, speeds were evaluated in a range of $500\,\mathrm{rpm}$ up to $2000\,\mathrm{rpm}$ with $500\,\mathrm{rpm}$ steps. However, since the oscillatory behavior is more pronounced in the part-load range than, e.g., at the nominal operating point, such a part-load operating point, i.e., $n_\ell^* = 500\,\mathrm{rpm}$, is evaluated in the following. In addition, a relatively low speed and torque constraint is chosen to focus on the advantage of the superimposed CCS-MPC approach instead of the more dynamic behavior of the subordinate DMPC current control. In doing so, a torque constraint of T_N, i.e., 0.47 Nm in case of motor M5, is chosen for the control output, i.e., the control signal of the superimposed speed controller. In case of $\mathrm{CCF_{con}}$-MPC this constraint is part of the optimization—see Section 9.2.1—where for the control output of the PI-based controller clamping is used and for $\mathrm{CCF_{unc}}$-MPC and LQR[8] the control signal is simply saturated, i.e., truncated. The comparison of the control methods of interest is shown in Fig. 9.3-9.5. More concretely, Fig. 9.3 shows the speed of the load side. The associated torque T_el is shown in Fig. 9.4. For $\mathrm{CCF_{con}}$-MPC, the associated phase currents are additionally shown in Fig. A.17 of Appendix A.6. As can be seen in Fig. 9.3, $\mathrm{CCF_{con}}$-MPC reaches the speed reference much faster and with less overshoot. The same beneficial behavior appears for the load step. In order to interpret the results correctly, a more detailed analysis of the procedures and results is necessary.

From a theoretical point of view, it is assumed that LQR is in most cases preferable to CCS-MPC [SR98]. This is based on the fact that, first, LQR does not need to define a horizon, since the horizon in LQR is infinite. Second, similar to $\mathrm{CCS_{con}}$-MPC, constraints can be theoretically considered [SR98].[8] Third, the computational effort is less than for CCS-MPC. Although in simulation the LQR results were quite promising, in the experimental evaluation the gains of the controller had to be reduced to avoid oscillations at the controller output.[9] In addition, CCS-MPC settled faster during reference tracking, with similar observations shown in [SB17, p. 839]. Finally, the superimposed LQR algorithm requires an additional pre-filter or a tracking integrator to guarantee steady-state accuracy [JS95], [SB17, p. 837]. For these reasons, CCS-MPC is preferred in this work.

[8] Even though there are approaches for LQR that can take constraints into account, e.g., [SR98], this is not common and is not considered in this work.

[9] Using the LQR method as descriped in [JS95], the LQR gains are chosen to be $\alpha_\mathrm{LQR} = 500$, $\beta_\mathrm{LQR} = 500$, $\delta_\mathrm{LQR} = 10^7$ and $\gamma_\mathrm{LQR} = 1.3 \cdot 10^5$, where the feedback factor is $\boldsymbol{K}_\mathrm{LQR} = [0.13 \;\; 27.67 \;\; -0.047 \;\; 8.41]^T$.

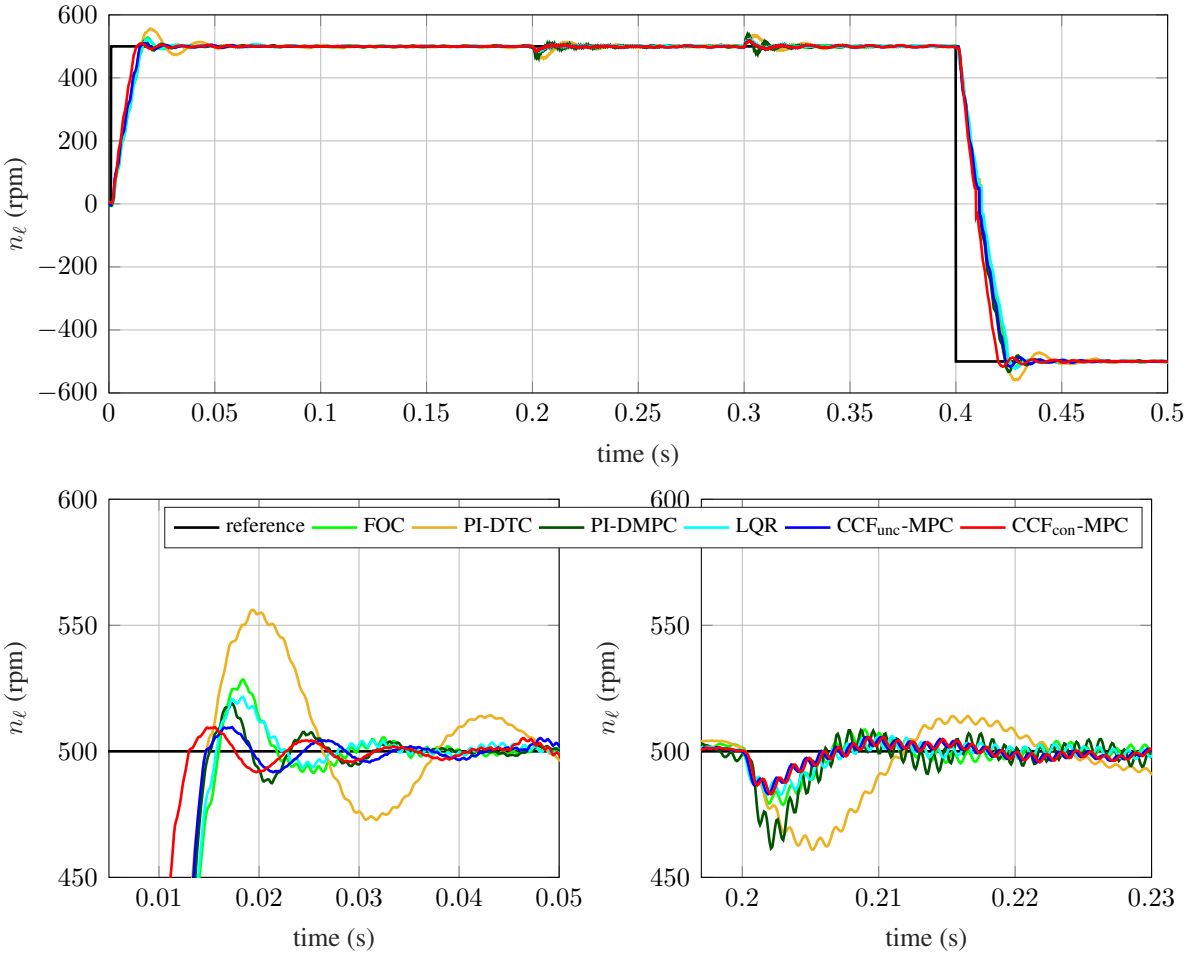


Figure 9.3: Motor M5: Comparison of the considered methods for a reference speed step from $n_\ell^* = 0$ rpm to $n_\ell^* = 500$ rpm using the test profile in Fig. A.16 (experimental).

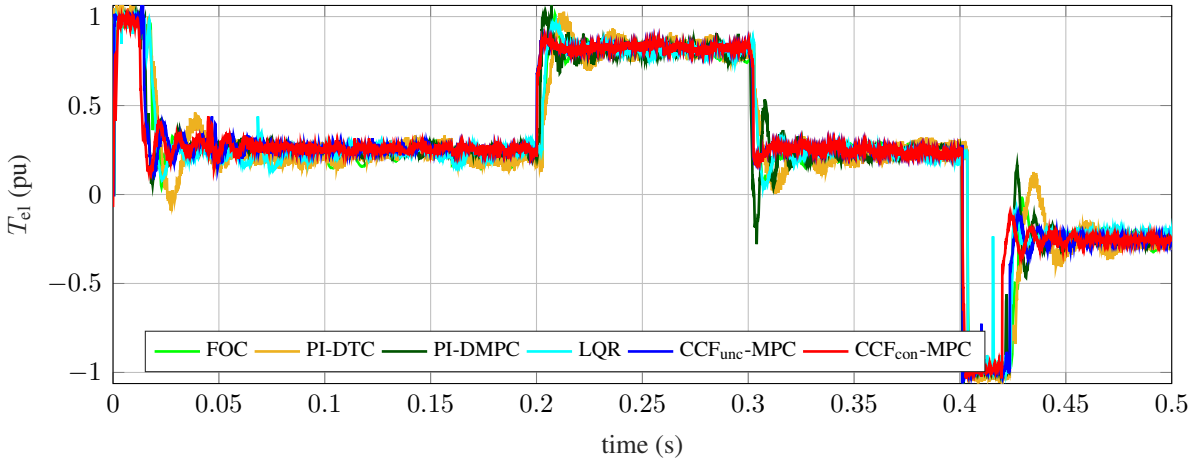


Figure 9.4: Motor M5: T_{el} of the considered methods for a reference speed step from $n_\ell^* = 0$ rpm to $n_\ell^* = 500$ rpm using the test profile in Fig. A.16 (experimental).

With respect to the superimposed PI-based controller, for each combination, i.e., PI-DTC, PI-DMPC, FOC, the SISO controller optimizes only the motor speed n_m without considering the angular displacement $\Delta\varphi$ and the load speed n_ℓ, the latter two being the actual optimization objectives. However, if the PI-based controller is fed with the load speed, disproportionate oscillations appear, making it unsuitable for stable operation.

With regard to the behavior of CCF$_\mathrm{unc}$-MPC and CCF$_\mathrm{con}$-MPC, however, a clear difference becomes apparent. As expected, both methods behave equally well for the load step shown in Fig. 9.3, since no constraint is active there. However, in case of the speed reference step, where the torque constraint is active, CCF$_\mathrm{con}$-MPC can directly take the limitations into account during the optimization and thus suppress the mechanical oscillations that occur much better compared to the other approaches. For a more detailed analysis of the oscillations, Fig. 9.5 shows the reference speed step of Fig. 9.3 in a close-up view. In addition, the corresponding deflection and oscillation of $\Delta\varphi$ is shown in Fig. 9.6(a), where the associated control output, i.e., T_el^* and the resulting measured torque T_el—using (2.26)—, are shown in Fig. 9.6(b).

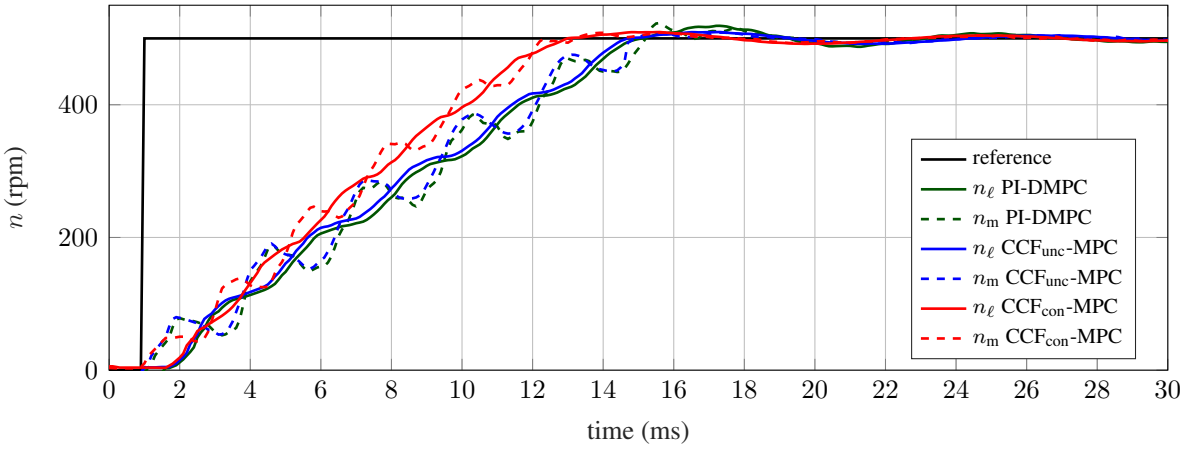
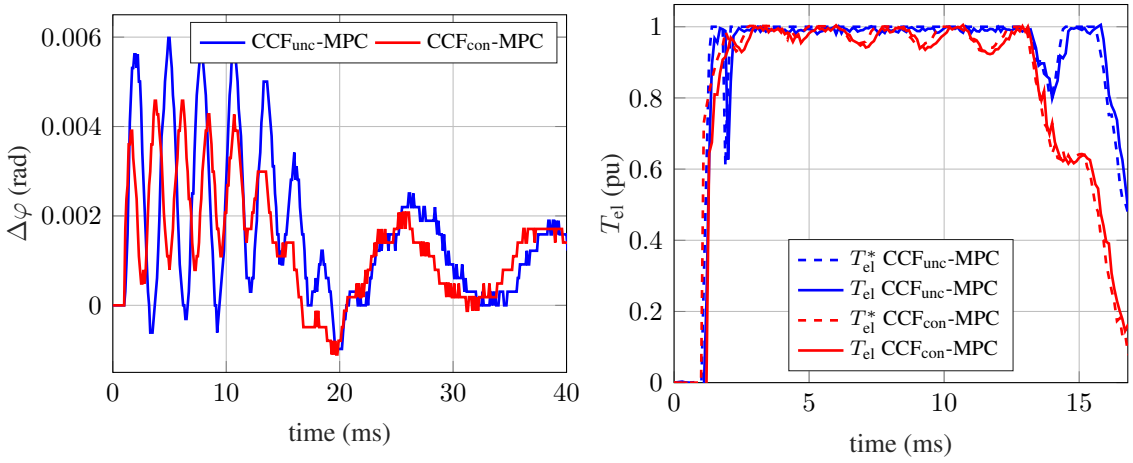


Figure 9.5: Motor M5: Speed for CCF$_\mathrm{con}$-MPC for a reference speed step from $n_\ell^* = 0$ rpm to $n_\ell^* = 500$ rpm (experimental).

The main benefit of the proposed CCF$_\mathrm{con}$-MPC can be clearly recognized in Fig. 9.6(a). As mentioned before, the CCS$_\mathrm{con}$-MPC suppresses the oscillations much more or excites them less and thus reaches the reference speed much faster than the other controllers. In case of CCF$_\mathrm{unc}$-MPC, the torsional moment resulting from twisting the TMS, see peak-to-peak in Fig. 9.6(a), can be calculated with

$$T_\mathrm{TMS} = c_\mathrm{TMS} \Delta\varphi \,. \tag{9.8}$$

Thus, using $c_\mathrm{TMS} = 100 \, \dfrac{\mathrm{Nm}}{\mathrm{rad}}$, the total change of $\Delta\varphi = 0.0065$ rad results in a torque of 0.65 Nm. This corresponds to almost 1.5 of the nominal torque of the considered drive. Even though—similar to a capacitor—the energy present during the oscillation is first stored in one oscillation



(a) Displacement angle $\Delta\varphi$.

(b) Control output T_{el}^* and measured T_{el}.

Figure 9.6: Motor M5: Transient comparison for CCS$_{unc}$-MPC and CCS$_{con}$-MPC for a reference speed step from $n_\ell^* = 0$ rpm to $n_\ell^* = 500$ rpm (experimental).

half-period and second released in the other half-period, this process is subject to losses, e.g., due to the damping d_{TMS}. More in detail, the dissipation, i.e., the irreversible conversion of kinetic energy into thermal energy, leads to the fact that for control methods with a significantly higher oscillation—the energy supplied is assumed to be the same for all methods due to the constraints—more energy is dissipated and is therefore not available for acceleration. Considering Fig. 9.5, the energy loses by dissipation in the TMS can be approximated from $t = 1{-}16$ ms by using

$$E_{dis}(t) = d_{TMS} \int (\omega_m(t) - \omega_\ell(t))^2 \mathrm{d}t \,. \tag{9.9}$$

Further details on dissipation can be found, e.g., in [Zel18, p. 56 ff.]. Additional damping effects occur due to electromagnetic damping and damping by the controller. As a result, CCF$_{con}$-MPC reaches the reference speed earlier thanks to the reduced oscillations.

Examining Fig. 9.6(b) more closely, it is noticeable that the clearly more pronounced oscillatory behavior with the other control methods, e.g., CCF$_{unc}$-MPC, is due, first, to the steeper increase of T_{el}^* and, second, to the non-existent reaction to the oscillations, i.e., the uncompensated oscillation amplitudes. CCF$_{unc}$-MPC still demands almost infinite torque to reach the reference speed as fast as possible, even if the torque limit has already been reached and thus the control signal is saturated. If the controller subsequently wants to reduce the control signal in order to minimize $\Delta\varphi$ for a predicted oscillation, the internally reduced control signal is still far above the torque constraint $T_{el,max}$, i.e., still saturated, due to the previously demanded high torque. As a result, the reduction has no effect. Contrary to this, CCF$_{con}$-MPC has a kind of integrated reference filter, whereby this results from the control signal constraint, i.e., $T_{el,max}$ (see (9.3)), in the cost function. On the other hand, a reference filter would have to be added separately for the

other methods. This avoids in case of CCF_{con}-MPC steep reference changes—if they cannot be followed due to the system limits—and thereby excites less oscillations. In addition, the control signal does not increase further inside the controller after reaching the constraint, so in case of oscillations these can be damped directly, see the red reductions in Fig. 9.6(b).

Evaluation with laser vibrometer To conclude the evaluation of CCF-MPC, an external measurement is performed using a Polytec rotational laser vibrometer RLV-5500. External means that the measured values are acquired with an external measuring unit instead of using the measured values of the controller, i.e., SoC FPGA. The vibrometer gives a detailed analysis of the different frequency components in the speed. The experimental setup is shown in Fig. A.3 of Appendix A.3. For the evaluation of CCF-MPC, the change in load speed during a load step of $T_\ell = 0.1\,\text{Nm}$ is measured. This change is represented by the angle difference between two measurement points of the load speed, i.e., $\Delta\varphi_\ell$. Fig. 9.7 shows a significant reduction of the deflection during the load step when using CCF-MPC. This confirms the results of the previous measurements. In case of CCF-MPC, no noticeable difference is apparent between CCF_{unc}-MPC and CCF_{con}-MPC. This is to be expected since the applied torque is not close to the torque limit, i.e., both control signals are identical.

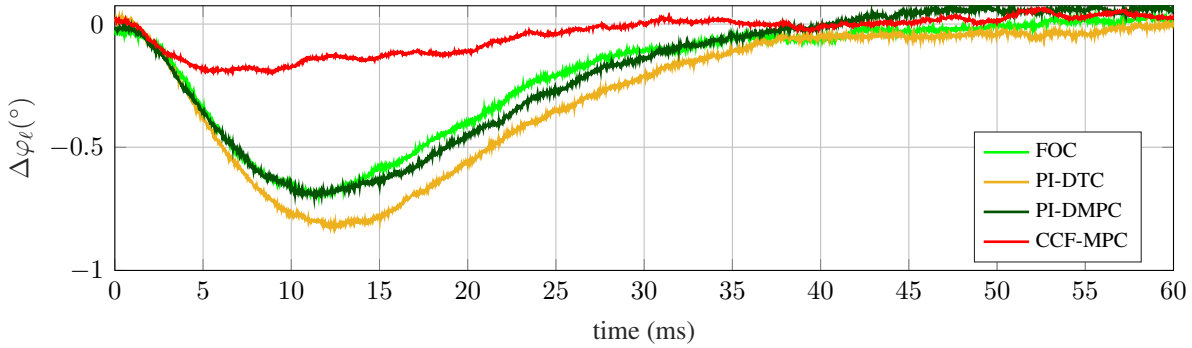
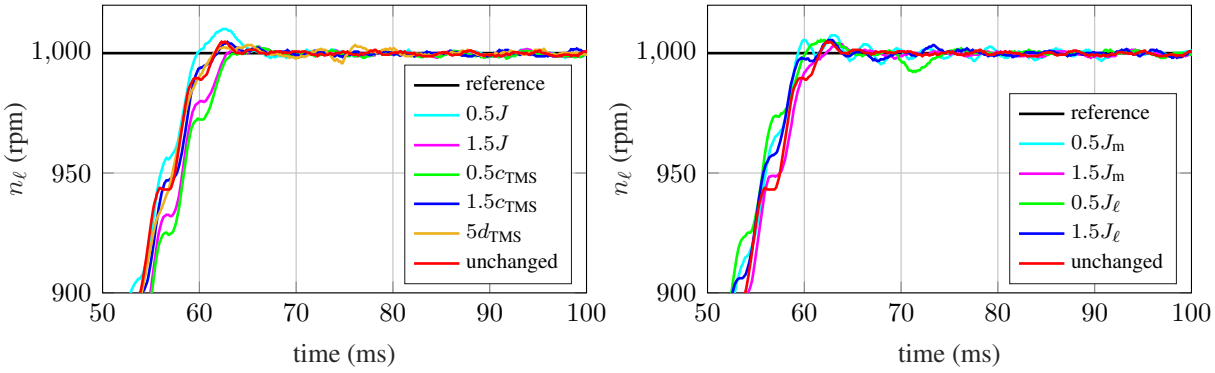


Figure 9.7: Motor M1: Displacement angle $\Delta\varphi_\ell$ of the load shaft during a load step $T_\ell = 0.1\,\text{Nm}$ (experimental).

9.4.1 Influence of Model and Parameter Inaccuracies

Similar to the previous chapters, the robustness of the model-based controller to parameter variations is of interest. Hence, in the following, a variation of the inertia, the torsional stiffness and the damping factor is added to the model. The test bench is unchanged. However, Fig. 9.8 shows almost similar control behavior under parameter mismatches compared to the unchanged model, even with significant deviations of 50 %. In Fig. 9.8(a), J represents the simultaneous variation of J_m and J_ℓ.



(a) Transient response after speed step.

(b) Transient response after speed step.

Figure 9.8: Motor M1: CCF-MPC with parameter variations for a reference speed step from $n_\ell^* = 0$ rpm to $n_\ell^* = 1000$ rpm (experimental).

In summary, the most important influence on the control behavior of the mechanical controlled system—similar to the inductance for the electrical controlled system, see Sections 2.6.3 and 6.3.3—is a parameter uncertainty for one of the moments of inertia, i.e., J_{m} or J_ℓ. The damping constant has the least influence.

9.4.2 Resource and Timing Evaluation

The following evaluation is done by using the first system, see Table A.2 of Appendix A.3, while the second system behaves similar.

Processor Table 9.2 shows the change of the ISR runtime, i.e., ΔT_{ISR}, which represents the computation time when using the individual speed control methods in steady state. It can be noted that the computation time of CCS_{con}-MPC is significantly increased compared to the other methods. Especially in transients, when constraints are (almost) active, the computation time of CCS_{con}-MPC can be even longer due to the slower converging problem. In steady state, however, it can be assumed that the numerical optimization converges in time since the set of active constraints are zero, i.e., $\gamma_{\text{act}} = \mathbf{0}$. The maximum ISR runtime, considering all tasks, was $T_{\text{ISR}} \approx 90\,\mu\text{s}$, which is near the ISR limit of $T_{\text{cc}} = 100\,\mu\text{s}$. However, it should be noted that during this maximum ISR runtime, i.e., when the maximum computation time is required by a slowly converging optimization problem, an online recalculation and replacement of the control matrices—see Section 9.2.3—has additionally taken place.

Table 9.2: Required computation time for the speed controller in steady state.

Required computation time	PI	LQR	Unconstrained CCS-MPC	Constrained CCS-MPC
ΔT_{ISR}	9 µs	9 µs	17 µs	27 µs

FPGA Table 9.3 shows the required total resources and the achieved timing when implementing the current control algorithm on an FPGA with a basic clock frequency of $100\,\mathrm{MHz}$. Even if the resources depend on the specific way of implementation (sharing and streaming), the provided information provides insight since it can indicate the required resources and/or computational load for an increased horizon and/or more complex cost functions. The fixed-point value range is chosen based on a normalized (pu) scaling. It should be mentioned that for the comparison of the two predictive approaches, apart from the prediction process itself (i.e., (2.73) as opposed to (7.4)), all IP cores are identical and use the same clocks. This means that, e.g., the ADC readout, the cost function and the dead-beat solution are completely identical. The main difference between the two VSP^{2}CC approaches is the use of flux linkage maps in case of the ψ-based method.

At this point, it is worth mentioning that flux linkage maps are preferred to polynomial solutions for reasons related to resource efficiency, see Section 4.3.4. Specifically, for the bivariate extension, two terms (one for self- and one for cross-saturation) and, therefore, two multiplications are required for each polynomial order. Moreover, a third term describes the impact of the permanent magnets. Therefore, it can be concluded that when comparing the calculation effort of both options and taking into account the subsequent FPGA implementation, the flux linkage maps constitute a computationally cheaper method.

Finally, FOC and DTC require only about half of the resources needed for the proposed direct MPC algorithms.

Table 9.3: Resource and timing evaluation for the FPGA using the first system in Table A.2.

Resources	LUT as logic	LUT as memory	Block RAMs	Slice registers (flip-flops)	DSP slices
Available in first system	53200	17400	140	106400	220
Available in second system	274080	144000	912	548160	2520
L-VSP^{2}CC (N_p=2)	30983	1134	32	30310	119
ψ-VSP^{2}CC (N_p=2)	39776	1470	37	36723	135
FOC	21652	1114	32	21675	52
DTC	22187	1214	32	23602	47

Timing	Sampling & dq-trans.	Dead-beat preselection	Prediction & VSP^{2}CC	Optimization & minimization	Total
L-VSP^{2}CC (N_p=2)	1.6 µs	0.3 µs	0.9 µs	0.38 µs	3.18 µs
ψ-VSP^{2}CC (N_p=2)	1.6 µs	0.3 µs	1.9 µs	0.38 µs	4.18 µs
FOC	1.6 µs	—	—	—	2.21 µs
DTC	1.6 µs	—	—	—	1.72 µs

9.5 Final Assessment

In this chapter, the advantageous behavior of MIMO control, i.e., LQR and CCS-MPC, over SISO control, i.e., PI, is demonstrated using an elastically coupled TMS as an example. Although LQR is a very capable MIMO method, CCS_{con}-MPC has shown better performance in the applications and experiments considered here, especially for reference tracking. Depending on the speed reference, the latter can be reached faster as well as the settling time of the closed loop can be reduced up to $50\,\%$. However, this may be different for other use cases.
Moreover, the proposed CCS_{con}-MPC showed the benefit over CCS_{unc}-MPC by considering the constraints directly during optimization instead of saturating the control signal afterwards. As demonstrated, in the case of an elastically coupled TMS, this leads to improved dynamics and a smoother speed behavior, which can lead to improved positioning accuracy. Moreover, with the presented method, it is possible to adjust the system matrices in the model of CCS-MPC during operation. In terms of usability, this is beneficial compared to the state of the art, as no precise knowledge of the TMS setup, i.e., the parameters, is required in advance. More precisely, the parameters are identified directly in the application by executing the offline parameter ID in Section 4.2. The same offline ID can be re-executed as soon as parameter changes are expected, e.g., during operating breaks.
Even though the proposed CCF_{con}-MPC implies an increased computational load compared to the other methods, this is addressed by the platform shown in Chapter 3. Consequently, as demonstrated in Section 9.4.2, the computational load is moderate.

In summary, CCF_{con}-MPC, as stated at the beginning of the thesis in the requirements of Section 1.2, enables the fastest possible dynamics while taking into account the minimization of energy losses. More specifically, the subordinate DMPC, i.e., ψ-VSP^{2}CC, follows ideally and provides exactly the required current as illustrated in Fig. 9.6(b). Moreover, as shown, a delay results only from the electrical time constant and the system delay, i.e., discrete sampling and control interval. Consequently, in case of Fig. 9.6(b), where $f_{\text{cc}} = 10\,\text{kHz}$ is used for the speed controller, the requested current is reached at most after $100\,\mu\text{s}$.

One possibility for improving the method is to observe the load speed n_ℓ, i.e., encoderless operation, which could eliminate the second incremental encoder on the load side. A second option is to extend the speed controller with a position controller that is as jerk-free as possible. Third, funnel control [SB17, p. 739 ff.]—which avoids a knowledge of the model parameters—is a promising alternative for the TMS speed control. This could potentially be attractive in combination with a subordinate DMPC. Here, a comparison with CCF_{con}-MPC would be interesting.

10 Conclusion and Outlook

Current and speed control of elastically coupled mechatronic drive systems using small—possibly magnetically nonlinear—PMSMs have been addressed. The main aim is to achieve very fast closed-loop dynamics while minimizing ripple and overshoots, with low-loss operation as a secondary condition. For this purpose, it is beneficial to consider the numerous system constraints such as maximum current, torque, and the natural behavior of the controlled system, i.e., the discrete switching states of the VSI, directly during optimization. It has been shown that classical SISO controllers such as FOC with SVM or DTC have their difficulties in considering several objectives simultaneously. For this reason, a combination of two MPC methods is formulated that achieves the above objectives in a significantly superior manner.

10.1 Conclusion

To accomplish the aforementioned claims—see also Section 1.1—this thesis addresses five open problems in the field of MPC. These issues are pointed out and possible solutions are proposed, theoretically analyzed and experimentally validated.

First, MPC requires increased processing power compared to the state of the art, which is addressed by introducing heterogeneous calculation platforms. These enable more efficient solving of control algorithms by performing distributed computation on processors and FPGA using low latency data exchange. Contrary to previous research, this makes it possible to solve FCS-MPC—and even ψ-VSP^{2}CC—with a long horizon and control frequencies of several hundred kHz.

Second, since the switching behavior of FCS-MPC for electrical machines with small electrical time constants has insufficient switching granularity for an acceptable current ripple, a variable switching point based on the calculation of current gradients is provided, i.e., VSP^{2}CC. Here, high current dynamics and current quality, i.e., low THD, are achieved simultaneously. This enables DMPC to be used not only for large drives but also for small drives, i.e., those with small electrical time constants, as demonstrated for the first time in this work.

Third, this thesis considers the possible impact of nonlinearities, i.e., parameter and model inaccuracies, on the control performance along with compensation solutions that are suitable for ensuring the best possible control behavior. More specifically, the inductance was identified

as the parameter with the most significant and non-negligible influence on the performance, compared to all other electrical parameters (while it was the moment of inertia in the case of mechanical parameters). Hence, a flux linkage-based prediction rather than an inductance-based one has been established, which enables accurate calculation of the current behavior even for PMSMs with a nonlinear magnetic circuit, i.e., when saturation and cross-coupling are pronounced. In the case of VSP^{2}CC, i.e., when using the current gradients, the results demonstrated the potential benefits of the proposed ψ-VSP^{2}CC. As is shown, the latter clearly outperforms L-VSP^{2}CC—which bases its prediction on inductances—when magnetic nonlinearities are of concern. Moreover, it achieves similar steady-state performance with FOC while exhibiting superior dynamic behavior.

Fourth, to compensate for rotor position-dependent as well as cyclically repeating model and parameter deviations, a repetitive correction term for direct control methods, i.e., DMPC, is introduced. This compensates for the negative influences of spatial harmonics, for example, that may be present in the PMSM design, resulting in increased current and, in turn, enhanced torque performance.

Fifth, when the mechanical controlled system is additionally considered, a cascaded MPC structure for mechatronic drive systems is presented, i.e., CCF-MPC. As in conventional FOC, it separates electrical and mechanical controlled systems rather than optimizing the entire system in a single MPC cost function. The novelty is that DMPC is used for the current control loop and CCS-MPC for the superimposed mechanical control loop. This separation facilitates both the fine granularity necessary for the prediction steps in current optimization and a discernible change in the states for the prediction of the mechanical behavior. This decouples the usually larger mechanical time constant, i.e., τ_m, from the electrical time constant τ_el. Thus, DMPC enables highly dynamic current optimization as described, and with superimposed CCS-MPC—which is comparatively computationally cheaper—a long prediction horizon for the entire drive system is still achieved. The demonstrated measurements show that the presented MIMO optimization, i.e., CCF-MPC, provides improved dynamics and shorter settling times in transients, as well as active damping of natural oscillations in the flexible shaft in steady state, in contrast to PI (SISO) and LQR (MIMO).

In addition, an inherent method for parameter identification—especially for the flux linkage—and online adaptation is elaborated, which enables high usability by parameterizing MPC models during operation, thus ensuring the best possible control behavior over the entire operating range.

In conclusion, the answers to the research question can now be assessed. As claimed, the proposed ψ-VSP^{2}CC achieves the fastest possible dynamic current response while avoiding overshoots, even for PMSMs with nonlinear magnetic behavior. Thus, the overall benefit of ψ-

VSP^{2}CC can be seen in terms of enhanced dynamics and increased usability in comparison to FOC with SVM and improved THD in comparison to conventional FCS-MPC, DTC and the L-VSP^{2}CC approach. More specifically, in the former case, i.e., FOC with SVM, the PI gains and pulse pattern must be adjusted as a function of the operating point to achieve acceptable performance over the entire operating range. In contrast, ψ-VSP^{2}CC inherently decides to switch more or less frequently depending on the operating point, while always ensuring maximum dynamics. Only a single weighting factor, i.e., λ_u, needs to be adjusted, which defines the acceptable current ripple.
Moreover, DMPC produces a spread spectrum—comparable to white noise—with low amplitudes, in contrast to SVM, which has a spectrum with certain orders of harmonics and comparatively high amplitudes. For drives, the spread spectrum, i.e., the distribution of harmonic energy over a wide frequency range, can actually be advantageous for two reasons. First, a variable switching frequency reduces mechanical excitation at certain frequencies, which is in contrast to the concentrated frequencies around the carrier frequency in the case of SVM [Hol94]. Second, depending on the operating point, it avoids unnecessary switching operations.
In addition, constraints are inherently accounted for in MPC, avoiding complex feedback paths and prioritization in the case of PI-typical clamping, i.e., when the current or voltage limit is reached.
To sum up, it can be concluded that CCF-MPC with subordinate ψ-VSP^{2}CC is an interesting alternative to established control methods and is useful for mechatronic controlled systems with small electrical drives.

10.2 Outlook

Iron losses due to the fundamental frequency cannot be mitigated by the controller, since the flux density B and fundamental frequency are due to the operating point. However, the inverter-related iron losses can potentially be beneficially affected. In SVM, the switching frequency, the multiples thereof, and the respective side harmonics each cause their own iron losses, which depend on the amplitude of the respective flux density with B^2. In the case of DMPC, a spread spectrum is given. Thus, considering all the harmonic orders generated by the inverter switching, a lower amplitude of B for the respective orders—even if more harmonic orders are present in the spread spectrum—can potentially reduce the overall inverter-related iron losses of the motor. Even though the iron losses turned out to be relatively low for the small drives under consideration, the reduction of inverter-related iron losses could be an extremely promising criterion for PMSMs in higher power classes.

Moreover, with the proposed repetitive extension in Chapter 8, the torque rather than the current can be optimized directly. Thus, for geometrically induced spatial harmonics in the flux linkage, the current can be bent in the opposite direction—instead of being corrected to an ideal fundamental—which, due to the multiplication of current and flux linkage in the torque equation, i.e., (2.26), can provide a smooth torque.

Considering its ability to directly account for constraints during optimization, DMPC is especially promising at the modulation limit, i.e., voltage limit, see, among others, [PB13d]. As shown by the first self-conducted investigations [Sch19], the voltage reserve, i.e., the reserve for the compensation of dynamic processes at the modulation limit, can be kept significantly smaller and thus the available dc-link voltage can be better utilized. However, this requires further investigation.

Finally, this thesis provides a basis for more complex systems such as multi-phase PMSMs, where significantly more states must be optimized simultaneously due to the increased complexity.

Nomenclature

Abbreviations

ADC	**A**nalog-to-**D**igital **C**onverter
CCF-MPC	**C**ascaded **C**ontinuous and **F**inite control set-**M**odel **P**redictive **C**ontrol
CCS-MPC	**C**ontinuous **C**ontrol **S**et-**M**odel **P**redictive **C**ontrol
DFT	**D**iscrete **F**ourier **T**ransform
DMPC	**D**irect **M**odel **P**redictive **C**ontrol
DSP	**D**igital **S**ignal **P**rocessor/**D**igital **S**ignal **P**rocessing
DT	**D**rive **T**rain
DTC	**D**irect **T**orque **C**ontrol
FCS-MPC	**F**inite **C**ontrol **S**et-**M**odel **P**redictive **C**ontrol
FEM	**F**inite **E**lement **M**ethod
FGM	**F**ast **G**radient **M**ethod
FOC	**F**ield **O**riented **C**ontrol
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
GUI	**G**raphical **U**ser **I**nterface
HDL	**H**ardware **D**escription **L**anguage
ID	**ID**entification
IDW	**I**nverse **D**istance **W**eighting
IGBT	**I**nsulated **G**ate **B**ipolar **T**ransistor
IP	**I**ntellectual **P**roperty
IPMSM	**I**nterior **P**ermanent **M**agnet **S**ynchronous **M**achine
ISR	**I**nterrupt **S**ervice **R**outine
ITAE	**I**ntegral of **T**ime multiplied by **A**bsolute value of **E**rror
LM	**L**evenberg-**M**arquardt algorithm

LMG	**P**ower **M**easuring **D**evice
LQR	**L**inear **Q**uadratic **R**egulator
MIMO	**M**ultiple-**I**nput **M**ultiple-**O**utput
MOSFET	**M**etal **O**xide **S**emiconductor **F**ield **E**ffect **T**ransistor
MPC	**M**odel **P**redictive **C**ontrol
NdFeB	**N**eo**d**ymium iron **B**oron
OCM	**O**n-**C**hip **M**emory
PI	**P**roportional-**I**ntegral
PMASynRM	**P**ermanent **M**agnet **A**ssisted **Syn**chronous **R**eluctance **M**achine
PMSM	**P**ermanent **M**agnet **S**ynchronous **M**aschine
PRBS	**P**seudo **R**andom **B**inary **S**equence
PWM	**P**ulse **W**idth **M**odulation
QP	**Q**uadratic **P**rogramming
RCP	**R**apid **C**ontrol **P**rototyping
RSM	**R**eluctance **S**ynchronous **M**achine
SAR	**S**uccessive **A**pproximation **R**egister
SISO	**S**ingle-**I**nput **S**ingle-**O**utput
SoC FPGA	**S**ystem **o**n **C**hip **F**ield **P**rogrammable **G**ate **A**rray
SPI	**S**erial **P**eripheral **I**nterface
SPMSM	**S**urface **P**ermanent **M**agnet **S**ynchronous **M**achine
SV	discrete voltage **S**pace **V**ector
SVM	**S**pace **V**ector **M**odulation
SynRM	**Syn**chronous **R**eluctance **M**achine
THD	**T**otal **H**armonic **D**istortion
TMS	**T**wo-**M**ass **S**ystem
VHDL	**V**ery high speed integrated circuit **H**ardware **D**escription **L**anguage
VSI	**V**oltage **S**ource **I**nverter
VSP	**V**ariable **S**witching **P**oint
VSP^{2}CC	**V**ariable **S**witching **P**oint **P**redictive **C**urrent **C**ontrol
VSP^{2}TC	**V**ariable **S**witching **P**oint **P**redictive **T**orque **C**ontrol

Symbols

α_{R}	temperature coefficient of winding material in 1/K
α_{ψ}	temperature coefficient of flux linkage in 1/K
$\bar{\boldsymbol{\Psi}}$	measurement array for the scattered flux linkage maps
$\mathbf{0}_{m\times n}$	zero matrix of dimension $m \times n$ with $\mathbf{0}_{m\times n} \in \mathbb{R}^{m\times n}$
$\boldsymbol{\Gamma}$	predictive state matrix
$\hat{\boldsymbol{\Theta}}_{\mathrm{LM}}$	parameter estimation vector for Levenberg-Marquardt algorithm
$\boldsymbol{\Psi}$	measurement array for the instantaneous and scattered flux linkage maps
$\boldsymbol{\Upsilon}$	predictive input matrix
$\boldsymbol{A}$, $\boldsymbol{A}_{\mathrm{c}}$	state (or system) matrix in discrete and continuous time respectively
$\boldsymbol{B}$, $\boldsymbol{B}_{\mathrm{c}}$	input matrix in discrete and continuous time respectively
$\boldsymbol{C}$, $\boldsymbol{C}_{\mathrm{c}}$	output matrix in discrete and continuous time respectively
$\boldsymbol{H}$	Hessian matrix
$\boldsymbol{I}_n$	identity matrix of dimension $n \times n$, $\boldsymbol{I}_n = \mathrm{diag}(1, ..., 1) \in \mathbb{R}^{n\times n}$
$\boldsymbol{J}_{\mathrm{J}}$, $\boldsymbol{J}_{\mathrm{J,c}}$	Jacobian matrix in discrete and continuous time respectively
$\Delta\boldsymbol{\Psi}_d$	measurement array for the temperature-dependent flux deviation
$\Delta\omega$	deviation of the electrical rotor angular velocity in rad/s
$\Delta\psi_d$, $\Delta\psi_q$, $\Delta\boldsymbol{\psi}_{dq}$	deviation of the flux linkage in the rotor-fixed coordinate system in Vs
$\Delta\varphi$	displacement angle between motor and load shaft position in rad
$\Delta\varphi_{\Delta}$	deviation for the rotor position encoder offset in rad
$\Delta\vartheta$	temperature difference in °C
$\Delta c_{\mathrm{TMS,rel}}$	relative deviation of the estimated torsional stiffness in %
$\Delta d_{\mathrm{TMS,rel}}$	relative deviation of the estimated rotational damping in %
Δi_a, Δi_b, Δi_c	deviation of the phase currents in A
Δi_d, Δi_q, $\Delta\boldsymbol{i}_{dq}$	deviation of the current in the rotor-fixed coordinate system in A
ΔR_{ph}	deviation of the phase (winding) resistance in Ω
ΔT	torque deviation in Nm
Δu, $\Delta\boldsymbol{u}$, $\Delta\boldsymbol{U}$	change in switch position
Δv_a, Δv_b, Δv_c	deviation of the phase voltages in V
$\Delta v_{d,\mathrm{Fe}}$, $\Delta v_{q,\mathrm{Fe}}$	iron loss error voltages in the rotor-fixed coordinate system in V

Δv_d, Δv_q, $\Delta \boldsymbol{v}_{dq}$	deviation of the voltage in the rotor-fixed coordinate system in V
Δx, Δy, Δz	absolute deviation of a variable
Δ_{step}	relative measurement uncertainty as a result of step size for ID in $\%$
ℓ	discrete time step relative to k
γ, $\boldsymbol{\gamma}$, γ_{act}	constraint when using CCS-MPC
γ_{db}	angle of the dead-beat solution in rad
γ_{svm}	angle of the space vector modulation in rad
κ	number of discrete samples for one ID step/number of parameters in the system
λ	weighting factor
λ_{LM}	weighting factor for scaling Levenberg-Marquardt algorithm
$\lambda_{\Delta\varphi}$	weighting factor for the angle deflection when using CCS-MPC
λ_{ω_ℓ}	weighting factor for the load speed when using CCS-MPC
λ_{u}	weighting factor for penalizing switching when using FCS-MPC
Λ_d, Λ_q, $\boldsymbol{\Lambda}_{dq}$	differential angular dependencies of the flux linkage in Vs/rad
$\mathcal{P}_{\omega_{\text{m}}}$	spectral speed power density in dBm
$\mathcal{P}_{T_{\text{m}}}$	spectral torque power density in dBm
ω_{dam}	measured damper frequency in rad/s
ω_{el}	electrical rotor angular velocity in rad/s
ω_{m}	mechanical rotor angular velocity on the motor side in rad/s
$\omega_{\text{res,ana}}$	analytic resonance frequency in rad/s
ω_{res}	measured resonance frequency in rad/s
ω_ℓ	mechanical rotor angular velocity on the load side in rad/s
ψ_α, ψ_β, $\boldsymbol{\psi}_{\alpha\beta}$	flux linkage in the stator-fixed coordinate system in Vs
$\psi_{\text{pm,ref}}$	reference direct-axis flux linkage component caused by the permanent magnets per pole pair in Vs (Wb)
ψ_{pm}	direct-axis flux linkage component caused by the permanent magnets per pole pair in Vs (Wb)
ψ_a, ψ_b, ψ_c, $\boldsymbol{\psi}_{abc}$	flux linkages of each phase in Vs

ψ_d, ψ_q, $\boldsymbol{\psi}_{dq}$	direct- and quadrature-axis flux linkage in the rotor-fixed coordinate system in Vs
$\psi_d^{\circledast}$	direct-axis flux linkage due to the current excitation without the permanent magnets in Vs
ρ, ρ_{act}, ρ_{con}, $\rho^{\circledast}$	Lagrange multiplier when using CCS-MPC
ρ_{sph}	radius of sphere when using FCS-MPC
σ	standard deviation of the measurements
τ_{el}	electrical time constant of the motor in s
τ_{m}	mechanical time constant of the motor in s
τ_{ϑ}	thermal time constant of the motor in s
ε	distance between two measurements in the flux linkage map in A (for IDW)
ε_{min}	minimum distance between two measurements in the flux linkage map in A (for IDW)
φ	electrical rotor angle in rad
φ_{m}	mechanical rotor angle on the motor side in rad
φ_{Δ}	rotor position encoder offset in rad
φ_{ℓ}	mechanical rotor angle on the load side in rad
ϑ	temperature of the motor winding in °C
ϑ_{inv}	temperature of the semiconductors of the inverter in °C
$\vartheta_{\text{m,ref}}$	reference temperature of the permanent magnets in °C
ϑ_{m}	temperature of the permanent magnets in °C
ϑ_{ref}	reference temperature of the motor winding in °C
$\widetilde{\omega}_{\text{m}}$	mean of speed measurement values for viscous friction ID in rad/s
$\widetilde{T}_{\text{m}}$	mean of torque measurement values for viscous friction ID in Nm
ξ, $\boldsymbol{\xi}$	control deviation, output tracking term
ζ, $\boldsymbol{\zeta}$	general state
B	flux density in T (Vs/m^2)
c	torsional stiffness in Nm/rad
c_{cou}	torsional stiffness of the coupling in Nm/rad
$c_{\text{TMS,0}}$	torsional stiffness of the undamped system in Nm/rad

$c_{\text{TMS,ideal}}$	reference for torsional stiffness in Nm/rad
c_{TMS}	torsional stiffness of the whole TMS in Nm/rad
D	damping ratio in $\%$
d	rotational damping in Nms/rad
d_{fric}	viscous friction constant in Nms/rad
$d_{\text{TMS,ideal}}$	reference for rotational damping in Nms/rad
d_{TMS}	rotational damping of the whole TMS in Nms/rad
e	Euler's number
f_{a}	sampling frequency of the measurements in Hz
f_{cc}	control frequency of the mechanical control loop in Hz
f_{cf}	control frequency of the electrical control loop in Hz
f_{c}	control frequency in Hz
f_{dam}	measured damper frequency in Hz
$f_{\text{ID, min}}$, $f_{\text{ID, max}}$	smallest and largest identified frequency during ID in Hz
f_{res}	measured resonance frequency in Hz
$f_{\text{S, PRBS}}$	sampling rate of the Welch algorithm in Hz
f_{sw}	(average) switching frequency in Hz
G_{ana}	analytic transfer function for ID in rad/(Nms)
G_{exp}	measured transfer function for ID in rad/(Nms)
G_{S}	shear modulus in Nm/mm^2
i, $\boldsymbol{i}$, I	current in A or pu
i, j, l, m, n, o	arbitrary counting index
i_0	no load RMS phase current in A
I_{N}	rated current in A
i_α, i_β, $\boldsymbol{i}_{\alpha\beta}$	currents in the stator-fixed coordinate system in A
i_a, i_b, i_c, $\boldsymbol{i}_{abc}$	phase currents in A
i_d, i_q, $\boldsymbol{i}_{dq}$	currents in the rotor-fixed coordination system in A
J	moment of inertia in kgm^2
j	imaginary unit of complex numbers $\sqrt{-1}$
J_{cou}	moment of inertia of the coupling in kgm^2

J_{c}, $J_{\mathrm{c,unc}}$, $J_{\mathrm{c,con}}$	cost function of CCS-MPC
$J_{\mathrm{e,m}}$, $J_{\mathrm{e},\ell}$	moment of inertia of the encoder on the motor and load side respectively in kgm^2
J_{f}	cost function of FCS-MPC
J_{LM}	cost function of Levenberg-Marquardt algorithm
$J_{\mathrm{ms,m}}$, $J_{\mathrm{ms},\ell}$	moment of inertia of the torque measuring shaft on the motor and load side respectively in kgm^2
J_{sm}	moment of inertia of the motor in kgm^2
J_{ℓ}	moment of inertia on the load side in kgm^2
J_{m}	moment of inertia on the motor side in kgm^2
$J_{\mathrm{s}\ell}$	moment of inertia of the load in kgm^2
k	discrete time instants, $k \in \mathbb{N}_0$
k_{p}	proportional tuning factor
$k_{\mathrm{p},\omega}$	proportional tuning factor for speed controller
$k_{\mathrm{p},id}$, $k_{\mathrm{p},iq}$	proportional tuning factor for current controller in the respective axis
$L_{\text{ph-ph}}$	phase-phase inductance between two terminals in H
L_{ph}, L_a, L_b, L_c	phase (winding) inductance in H
L_{dd}	partial (differential) d-inductance of the d-axis (self-coupling inductance in the direct-axis) in the rotor-fixed coordinate system in H
L_{dq}	partial (differential) q-inductance of the d-axis (cross-coupling inductance in the direct-axis) in the rotor-fixed coordinate system in H
L_d, L_q, $\boldsymbol{L}_{dq}$	absolute inductances in the rotor-fixed coordinate system in H
L_{qd}	partial (differential) d-inductance of the q-axis (cross-coupling inductance in the quadrature-axis) in the rotor-fixed coordinate system in H
L_{qq}	partial (differential) q-inductance of the q-axis (self-coupling inductance in the quadrature-axis) in the rotor-fixed coordinate system in H
m_d, m_q, $\boldsymbol{m}_{dq}$	current gradient (slope) in A/s
n	motor speed in rpm
N_{brk}	number of measurements during breakaway torque ID
n_{max}	maximum allowed speed in rpm
n_{min}	minimum speed representable by the encoder in rpm

n_{N}	rated motor speed in rpm
N_{PRBS}	number of switching operations in a PRBS interval
n_{PRBS}	number of shift register stages
N_{sty}	number of control steps to guarantee steady state for breakaway torque ID
N_{visco}	number of measurement steps for viscous friction ID
p	number of pole pairs
r	weighting factor of the distance influence when using IDW
$R_{\mathrm{DS(on)}}$	drain-source on-state resistance of the MOSFET in Ω
R_{Fe}	iron loss resistance Ω
$R_{\mathrm{ph,ref}}$	reference phase (winding) resistance in Ω
$R_{\mathrm{ph\text{-}ph}}$	phase-phase resistance between two terminals in Ω
R_{ph}, R_a, R_b, R_c	phase (winding) resistance in Ω
R_{sh}	resistance of the shunt in Ω
s	Laplace operator
s_{step}	scaling factor of current steps during ID
$S_{\omega T}$	variance of measured speed and torque values in Nm rad/s
S_{ω^2}	variance of measured speed values in rad^2/s^2
t	time in s
T_{a}	sampling interval of the measurements in s
$t_{\mathrm{brk,ID}}$	duration of breakaway torque ID in s
T_{brk}	breakaway friction torque in Nm
T_{cc}	control interval of the mechanical control loop in s
T_{cf}	control interval of the electrical control loop in s
T_{coul}	coulomb friction torque (speed independent) in Nm
T_{c}	control interval in s
T_{el}	electromagnetic (electromechanical or inner) torque in Nm
T_{fpga}	period of the FPGA clock in s
$T_{\mathrm{fric,k}}$	single friction torque of one arbitrary friction contact in Nm
T_{fric}	total torque through friction in Nm
T_{f}	the torque applied by the spring to the load side (spring torque) in Nm

T_{ISR}	runtime of the ISR in s
T_{i}	integral tuning factor
T_{ms}	torque of the measuring shaft in Nm
T_{m}	mechanical torque on the motor side in Nm
T_{N}	rated torque in Nm
$T_{\mathrm{P,PRBS}}$	period of the PRBS in s
$T_{\mathrm{S,PRBS}}$	cycle time of the PRBS in s
T_{sw}	modulation and switching interval in s
t_{z}	variable switching point in s or pu
T_{ℓ}	mechanical torque on the load side in Nm
$T_{\mathrm{i},\omega}$	integral tuning factor for speed controller
$T_{\mathrm{i},id}$, $T_{\mathrm{i},iq}$	integral tuning factor for current controller in the respective axis
U, $\boldsymbol{U}$	sequence of switch positions (switching sequence)
u, $\boldsymbol{u}$	switch position, control signal, input signal
v, $\boldsymbol{v}$, V	voltage in V or pu
V_{dc}	dc-link voltage in V or pu
V_{N}	rated voltage in V
v_{α}, v_{β}, $\boldsymbol{v}_{\alpha\beta}$	voltages in the stator-fixed coordination system in V
v_a, v_b, v_c, $\boldsymbol{v}_{abc}$	phase voltages in V
v_d, v_q, $\boldsymbol{v}_{dq}$	voltages in the rotor-fixed coordination system in V
x, $\boldsymbol{x}$, $\boldsymbol{X}$	state signal or x-axis in the coordinate system
$x^{(1)}$	measurement for the test signal with positive half-wave
$x^{(2)}$	measurement for the test signal with negative half-wave
y, $\boldsymbol{y}$, $\boldsymbol{Y}$	controlled signal, output signal or y-axis in the coordinate system
z	disturbance signal or z-axis in the coordinate system

A Appendix

A.1 Mathematical Definitions

In this work the following conventions for mathematical symbols and definitions are used:

x	scalars are written in italic letters
$\boldsymbol{x}$	vectors are bold lower case letters
$\boldsymbol{X}$	matrices are bold upper case letters
x^*	references are marked with a star superscript
$\lvert x \rvert$	absolute value of the scalar x (abs(x))
$\lvert\lvert \boldsymbol{x} \rvert\rvert_1$	1-norm of the vector $\boldsymbol{x}$, i.e., sum of the absolute values ($\sum$ abs($\boldsymbol{x}$))
$\lvert\lvert \boldsymbol{x} \rvert\rvert_2$	2-norm or length of the vector $\boldsymbol{x}$, i.e., square root of the sum of the squared values also named Euclidian norm ($\sqrt{\sum \boldsymbol{x}^2}$)
$\lvert\lvert \boldsymbol{x} \rvert\rvert_2^2$	sum of the squared values ($\sum \boldsymbol{x}^2$)

A.2 Derivations

A.2.1 Space Vector Modulation - Quadrant and Sector Selection

For an SVM implementation which avoids the additional trigonometric functions in (2.8a)-(2.8b), the existing sin(φ) and cos(φ) values of the current dq-transformation are used to convert $\boldsymbol{v}^*_{dq}$ to $\boldsymbol{v}^*_{\alpha\beta}$, see Section 2.1.1. Afterwards, by using

$$\mathrm{a} = |v^*_\alpha| + \frac{1}{\sqrt{3}} \left|v^*_\beta\right| \tag{A.1a}$$

$$\mathrm{b} = |v^*_\alpha| - \frac{1}{\sqrt{3}} \left|v^*_\beta\right| \tag{A.1b}$$

$$\mathrm{c} = \frac{2}{\sqrt{3}} \left|v^*_\beta\right| \ , \tag{A.1c}$$

the sector and quadrant of $\boldsymbol{v}^*_{\alpha\beta}$ can be determined with Fig. A.1 by using the signs of v^*_α and v^*_β as well as b. In doing so, $|\boldsymbol{v}_\mathrm{R}|$ and $|\boldsymbol{v}_\mathrm{L}|$ can be determined with the help of Table A.1.

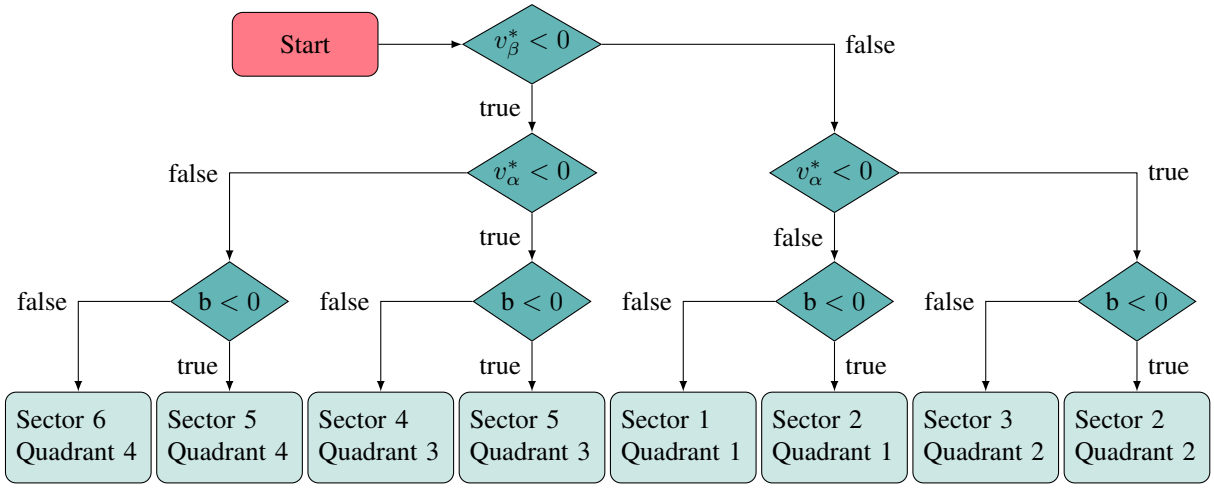


Figure A.1: Space vector modulation - flowchart to determine quadrant and sector [QD15, p. 32].

Table A.1: Space vector modulation - table for calculation of the voltage vectors $|\boldsymbol{v}_\mathrm{R}|$ and $|\boldsymbol{v}_\mathrm{L}|$ based on the current sector and quadrant [QD15, p. 23 ff.].

		$\lvert\boldsymbol{v}_\mathrm{R}\rvert$	$\lvert\boldsymbol{v}_\mathrm{L}\rvert$
Sector 1	Quadrant 1	$\lvert v_\alpha^*\rvert - \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$	$\frac{2}{\sqrt{3}}\lvert v_\beta^*\rvert$
Sector 2	Quadrant 1	$\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$	$-\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$
	Quadrant 2	$-\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$	$\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$
Sector 3	Quadrant 2	$\frac{2}{\sqrt{3}}\lvert v_\beta^*\rvert$	$\lvert v_\alpha^*\rvert - \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$
Sector 4	Quadrant 3	$\lvert v_\alpha^*\rvert - \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$	$\frac{2}{\sqrt{3}}\lvert v_\beta^*\rvert$
Sector 5	Quadrant 3	$\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$	$-\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$
	Quadrant 4	$-\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$	$\lvert v_\alpha^*\rvert + \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$
Sector 6	Quadrant 4	$\frac{2}{\sqrt{3}}\lvert v_\beta^*\rvert$	$\lvert v_\alpha^*\rvert - \frac{1}{\sqrt{3}}\lvert v_\beta^*\rvert$

Finally, with the help of $\boldsymbol{v}_\mathrm{R}$, $\boldsymbol{v}_\mathrm{L}$ and $V_\mathrm{max} = V_\mathrm{dc}/\sqrt{3}$ the activation times of the discrete SVs are determined without additional trigonometric function by using

$$t_\mathrm{R} = T_\mathrm{sw}\frac{|\boldsymbol{v}_\mathrm{R}|}{V_\mathrm{max}} \tag{A.2a}$$

$$t_\mathrm{L} = T_\mathrm{sw}\frac{|\boldsymbol{v}_\mathrm{L}|}{V_\mathrm{max}} \tag{A.2b}$$

$$t_0 = T_\mathrm{sw} - t_\mathrm{R} - t_\mathrm{L}\ . \tag{A.2c}$$

A.2.2 Derivation of the Discretization Error Corrected Flux Linkage Prediction

A detailed derivation of the flux linkage change is provided hereafter. Using (7.2), the flux linkage change at time step $k+1$ is [RGD14]

$$\psi_d(k{+}1) = \psi_d(k) + T_{\text{cf}}\Bigg(v_d(k) - R_{\text{ph}} i_d(k) + \frac{\omega_{\text{el}}(k)}{2}\Big(\psi_q(k){+} \underbrace{\psi_q(k) + T_{\text{cf}}\big(v_q(k) - R_{\text{ph}} i_q(k) - \frac{\omega_{\text{el}}(k)}{2}\Sigma\psi_d(k{+}1)\big)}_{\psi_q(k+1)}\Big)\Bigg) \tag{A.3a}$$

$$\psi_q(k{+}1) = \psi_q(k) + T_{\text{cf}}\Bigg(v_q(k) - R_{\text{ph}} i_q(k) - \frac{\omega_{\text{el}}(k)}{2}\Big(\psi_d(k){+} \underbrace{\psi_d(k) + T_{\text{cf}}\big(v_d(k) - R_{\text{ph}} i_d(k) + \frac{\omega_{\text{el}}(k)}{2}\Sigma\psi_q(k{+}1)\big)}_{\psi_d(k+1)}\Big)\Bigg)\,. \tag{A.3b}$$

After rearranging and collecting terms, it yields

$$\begin{aligned}\psi_d(k{+}1) = \psi_d(k) &+ \frac{T_{\text{cf}}(v_d(k) - R_{\text{ph}} i_d(k) + \omega_{\text{el}}(k)\psi_q(k))}{1 + \frac{1}{4}T_{\text{cf}}^2\omega_{\text{el}}^2(k)} \\ &+ \frac{T_{\text{cf}}^2\frac{\omega_{\text{el}}(k)}{2}(v_q(k) - R_{\text{ph}} i_q(k) - \omega_{\text{el}}(k)\psi_d(k))}{1 + \frac{1}{4}T_{\text{cf}}^2\omega_{\text{el}}^2(k)}\end{aligned} \tag{A.4a}$$

$$\begin{aligned}\psi_q(k{+}1) = \psi_q(k) &+ \frac{T_{\text{cf}}(v_q(k) - R_{\text{ph}} i_q(k) - \omega_{\text{el}}(k)\psi_d(k))}{1 + \frac{1}{4}T_{\text{cf}}^2\omega_{\text{el}}^2(k)} \\ &+ \frac{T_{\text{cf}}^2\frac{\omega_{\text{el}}(k)}{2}(-v_d(k) + R_{\text{ph}} i_d(k) - \omega_{\text{el}}(k)\psi_q(k))}{1 + \frac{1}{4}T_{\text{cf}}^2\omega_{\text{el}}^2(k)}\,.\end{aligned} \tag{A.4b}$$

Further simplifications are valid. First, given that $T_{\text{cf}} \ll 1$, it holds that $\omega_{\text{el,max}} T_{\text{cf}} \ll 1$, where $\omega_{\text{el,max}}$ is the maximum electrical speed. In light of this, the last term of (A.4a) and (A.4b) can be neglected. Second, the denominator of the first term can be assumed to be constant and equal to one, since $\omega_{\text{el,max}}^2 T_{\text{cf}}^2 \approx 0$. However, since the formula should be valid in general, only the first assumption is implemented, whereas the second is not. Due to extreme conditions such as high

speed drives at low sampling and control frequencies (e.g., $\omega_{\mathrm{el}} = 1000\,\mathrm{rad/s}$ and $T_{\mathrm{cf}} = 1\,\mathrm{ms}$), the aforementioned denominator can be greater than one. For this reason, the denominator is calculated on the processor and is kept constant for one complete prediction process, since the mechanical angular velocity can be assumed to be constant during the prediction process of the electrical model. With the above manipulations, assumptions, and simplifications (7.4) results.

A.3 Test Bench

Two systems are used for the evaluation, as mentioned in Chapter 3. The first one is an industry-oriented solution and is more cost-effective. The second one is a high-end platform that serves as a feasibility study. Both systems and their core components are summarized in Table A.2.

Table A.2: Core components of the two control platforms used.

Component	First system **Ohmrichter**	Second system **UltraZohm**
Calculation unit (SoC FPGA)	XA7Z020	XCZU9EG
ADC	AD7980 (16-bit, 1 MSPS)	LTC2311-16 (16-bit, 5 MSPS)
Current sensor	CSS4527FT10L0	
Op-amp	INA240A1PWR	
MOSFET	BSC094N06LS5	
Gate driver	UCC27211A	

As an alternative to the first system shown in Fig. 3.4, the second real-time calculation system, i.e., the UltraZohm, first presented in [WGL+19], is used for longer horizons. The UltraZohm is shown in Fig. A.2. The idea, concept, hardware and software have been shared as open source and are continuously developed further by a community, see [WSL+].

In addition, a laser vibrometer is used as an external device to benchmark CCF-MPC and the other control approaches in Section 9.4 regarding the reduction of mechanical oscillations. The alignment of the laser vibrometer on the test bench is shown in Fig. A.3.

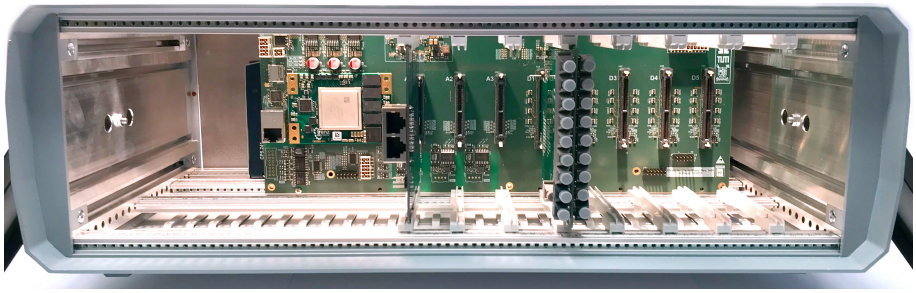


Figure A.2: UltraZohm real-time computation system with analog and digital adapter board.

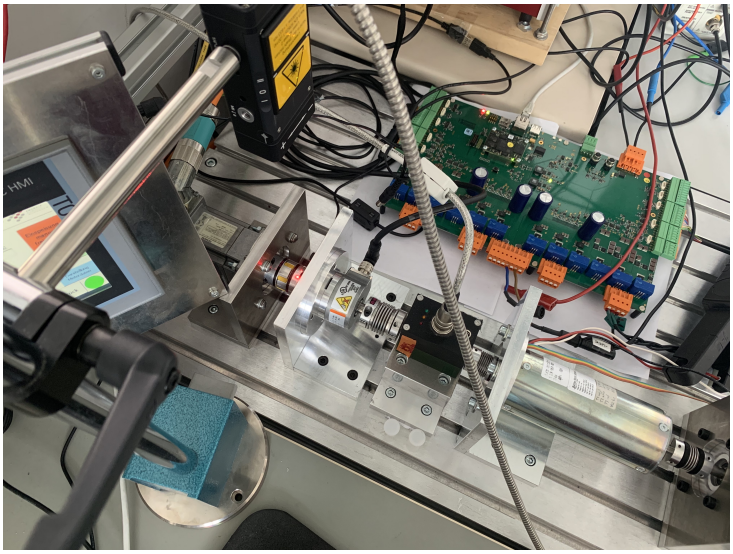


Figure A.3: Experimental setup for the evaluation of CCF-MPC via the Polytec rotational laser vibrometer RLV-5500.

A.3.1 Investigated Synchronous Machines

The following Table A.3 shows the investigated PMSMs. Note that for R_{ph}, L_d, L_q and ψ_{pm} the identified parameters are listed in the linear region of the machine. Additionally, the data sheet values are given in brackets. It is worth mentioning that the identified parameters also include the respective cables and connectors.

An incremental quadrature encoder with 5000 lines per revolution (Sendix 8.5000.8344.5000) is selected to determine the rotor position of the respective motor under test. An RS422 voltage signal is selected to ensure robust operation. In general, there a two possibilities to evaluate the

Table A.3: Motor and system parameters in the linear region of the magnetic circuit.

Description	Symbol	Motor M1 Bühler 1.25.058.401	Motor M2 ebm-papst ECI-63.20-K1-D00	Motor M3 ebm-papst ECI-63.20-K1-B00	Motor M4 Prototype —	Motor M5 Bühler 1.25.058.201
Circuit	—	star	star	star	star	star
Rotor PM	—	surface	interior	interior	interior	surface
Winding	—	non-overlapping concentrated	non-overlapping concentrated	non-overlapping concentrated	non-overlapping concentrated	non-overlapping concentrated
Pole pair number	p	4	4	4	4	4
Rated power	P_{N}	170 W	150 W	150 W	—	157 W
Rated voltage	V_{N}	24 V	48 V	24 V	24 V	24 V
Rated current	I_{N}	8.6 A	4.5 A	8.5 A	—	8.0 A
Rated torque	T_{N}	40 Ncm	36 Ncm	36 Ncm	90 Ncm	47 Ncm
Rated speed	n_{N}	4000 rpm	4000 rpm	4000 rpm	—	3200 rpm
Winding resistance	R_{ph}	107(85) mΩ	220 (210) mΩ	90 (70) mΩ	290 mΩ	86 (70) mΩ
d-axis inductance	L_d	0.26(0.25) mH	0.47 (0.44) mH	0.14 (0.13) mH	0.49 mH	0.29 (0.38) mH
q-axis inductance	L_q	0.26(0.25) mH	0.77 (0.44) mH	0.21 (0.13) mH	2.10 mH	0.32 (0.38) mH
PM flux constant	ψ_{pm}	5.9 mVs	10(11.6) mVs	6.0 (6.6) mVs	20 mVs	7.5 (7.0) mVs
dc-link voltage	V_{dc}	24 V	24 V	24 V	24 V	24 V

A, B, I-lines. First, the time between two—either rising or falling—edges of the same line, i.e., A or B, is measured, or, second, by counting the edges of both lines (quadrature) in a given time window. Combining these two possibilities, an IP core in the FPGA evaluates the lines with a clock frequency of 100 MHz and outputs ω_m, φ_m and φ, see Figs. 3.1-3.3.

A.3.2 Investigated Two-Mass System

Next to the motor parameters, the system parameters of the test bench—see Chapter 3—, i.e., load machine, measuring shaft (torque transducer), speed encoder, couplings are shown in Table A.4. The parameters were identified by the offline ID, see Section 4.2. In addition, all parameters of the test bench shown in Fig. 3.4 were calculated and compared to the identified values for validation.
The spring stiffness of the drive shaft and the other components is usually in a range that can be considered to be rigid, i.e., $c \to \infty$, compared to the spring stiffness of the couplings and torque measuring shaft. Comparing the test bench in Fig. 3.4, the spring stiffnesses of all couplings are identically, i.e., $c_\mathrm{cou,1\text{-}2} = c_\mathrm{cou,2\text{-}3} = c_\mathrm{cou,4\text{-}5} = 1200\,\mathrm{Nm/rad}$ [Coub], with one exception, the coupling between the measuring shaft and incremental encoder of the load side, which has the spring stiffness $c_\mathrm{cou,3\text{-}4} = 7000\,\mathrm{Nm/rad}$ [Coua]. In relation to this, the torsional stiffness of the measuring shaft is significantly lower. Therefore, the torsional stiffness of the couplings are considered torsionally stiff and the spring stiffness of the measuring shaft, i.e., $c_\mathrm{TMS} = 100\,\mathrm{Nm/rad}$ [Pra], can be used for modeling the TMS.

The summarized moment of inertia on the motor side, i.e., J_m, results from the incremental encoder of the motor under test ($J_\mathrm{e,m}$), the used motor—motor M5 is used in Table A.4—(J_sm), the drive side of the torque measuring shaft ($J_\mathrm{ms,m}$) and two couplings ($J_\mathrm{cou,1\text{-}2}$, $J_\mathrm{cou,2\text{-}3}$). The moment of inertia on the load side, i.e., J_ℓ, is a combination of the load side of the torque measuring shaft ($J_\mathrm{ms,\ell}$), the incremental encoder of the load side ($J_\mathrm{e,\ell}$), the load machine ($J_\mathrm{s\ell}$) and the remaining two couplings ($J_\mathrm{cou,3\text{-}4}$, $J_\mathrm{cou,4\text{-}5}$).

Table A.4: Parameter of the two-mass system under investigation using motor M5.

Description	Symbol	Value
Spring constant of the shaft (torsional stiffness)	c_TMS	100 Nm/rad
Viscous friction	d_fric	$1.71 \cdot 10^{-4}$ Nms/rad
Viscous damping of the TMS	d_TMS	$0.2 \cdot 10^{-3}$ Nms/rad
Mass inertia of the motor side	J_m	$4.61 \cdot 10^{-5}$ kgm^2
Mass inertia of the load side	J_ℓ	$5.66 \cdot 10^{-5}$ kgm^2
Mechanical time constant	τ_m	18.3 ms

A.4 Supplementary Measurements for Offline Parameter Identification

A.4.1 Results for Electrical Offline Parameter Identification

For the method presented in Section 4.1.1, Fig. A.4 exemplarily shows the identification of the resistance and d-axis inductance for motor M1, see Table A.3 of Appendix. It is worth mentioning that due to the switch positions, i.e., $\boldsymbol{u}_{abc} = [1 \ -1 \ -1]^T$, the terminal values are measured and thus a conversion to phase values, i.e., $\hat{R}_{\text{ph}}$ and $\hat{L}_{\text{d}}$, must be carried out.

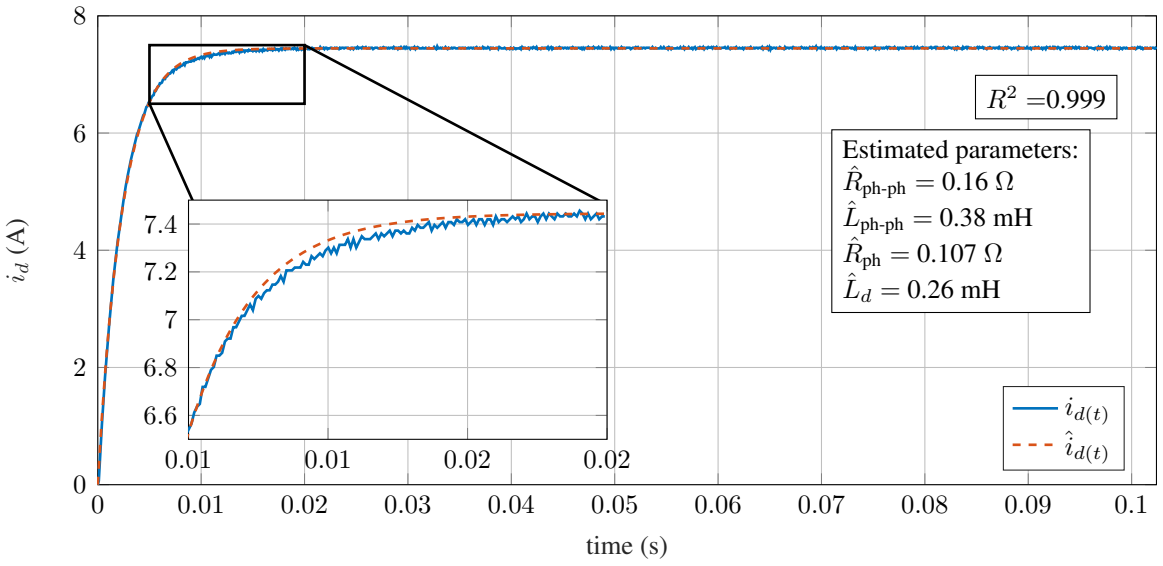
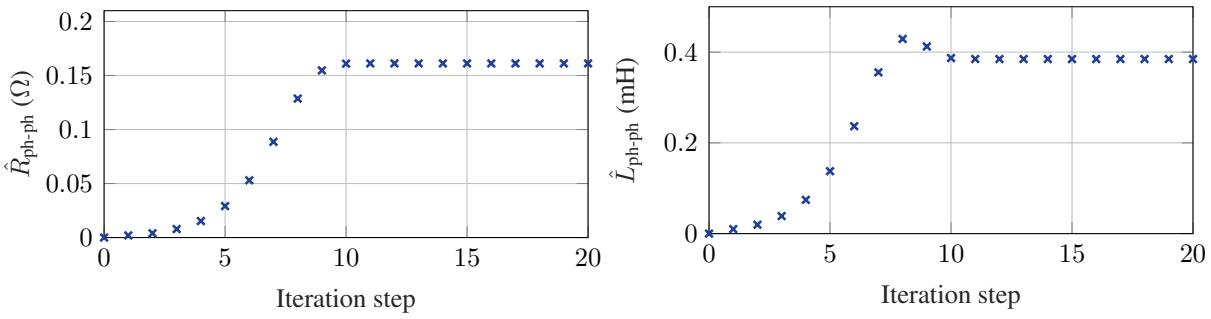


Figure A.4: Motor M1: Current response in d-axis (experimental).

The LM algorithm shows a high coefficient of determination and thus a good adaptation of the analytical to the measured current response. Fig. A.5 shows the convergence of the parameters during the optimization process of the LM algorithm. As can be seen, the parameters have already converged to an optimum after 11 iteration steps.



(a) Convergence of $\hat{R}_{\text{ph-ph}}$.

(b) Convergence of $\hat{L}_{\text{ph-ph}}$.

Figure A.5: Motor M1: Convergence of the LM algorithm (experimental).

A.4.2 Results for Friction Identification

In the following, the results of the breakaway torque followed by the results of the viscous and coulomb friction are discussed both in general and on the basis of specific experimental measurements.

Since the subsequent identification of T_{brk} and T_{fric} are based on T_{el}, i.e., (2.28), and thus calculated using the electrical measured quantities and variables, their accuracy is particularly important. The accuracy of current and angle described in Section 3.3 is used for this purpose. In addition, the values given in Table 4.3 are used for the flux linkage. Since only i_q is used during the ID, i.e, $i_d \approx 0$, the inductances are irrelevant.

Using a multiple of the standard deviation, i.e., 3σ, as discussed in Section 4.4.1, Table A.5 summarizes the resulting estimation uncertainty of the experimental identification procedure.
Here, the accuracy of the current sensor and encoder remains negligible, since averaging over 512 data values during the identification allows mutual compensation of the random deviations. Therefore, the uncertainty in the estimation of the flux linkage constant is responsible for the dominant error in the torque calculation.
The measurement uncertainty for the reference torque T_{ms}, determined by the torque measuring shaft, is mainly distorted by the measuring amplifier. Nevertheless, it is still about ten times more accurate than the identified torque.

Table A.5: Measurement uncertainty for the experimental ID process.

	$\boldsymbol{i}_{abc}$	$\Delta\varphi_{\Delta}$	ψ_{pm}	T_{el}	Torque measuring shaft	Torque measuring amplifier	T_{ms}
$\pm 3\sigma$	$\pm\, 0.04\,\%$	$\pm\, \pi/3200\,\text{rad}$	$2.94\,\%$	$\pm\, 2.98\,\%$	$\pm\, 0.05\,\%$	$\pm\, 0.3\,\%$	$\pm\, 0.35\,\%$

A.4.2.1 Breakaway Torque Identification

The identification of the breakaway torque is part of the offline parameter identification in Section 4.2.1. However, since this parameter is not used for the control of the mechanical controlled system in Chapter 9, but rather for condition monitoring, it is described in the following.

The identification process of the breakaway torque starts, as shown in Fig. A.6, by increasing the reference current, i.e., i_q^*, (independent of saliency) with discrete current steps i_{evastep} until rotation occurs. The step size depends on the rated motor current I_{N} and the step scaling factor s_{step}. It can be calculated by $i_{\text{evastep}} = I_{\text{N}} s_{\text{step}}$. To ensure completely development of motor torque

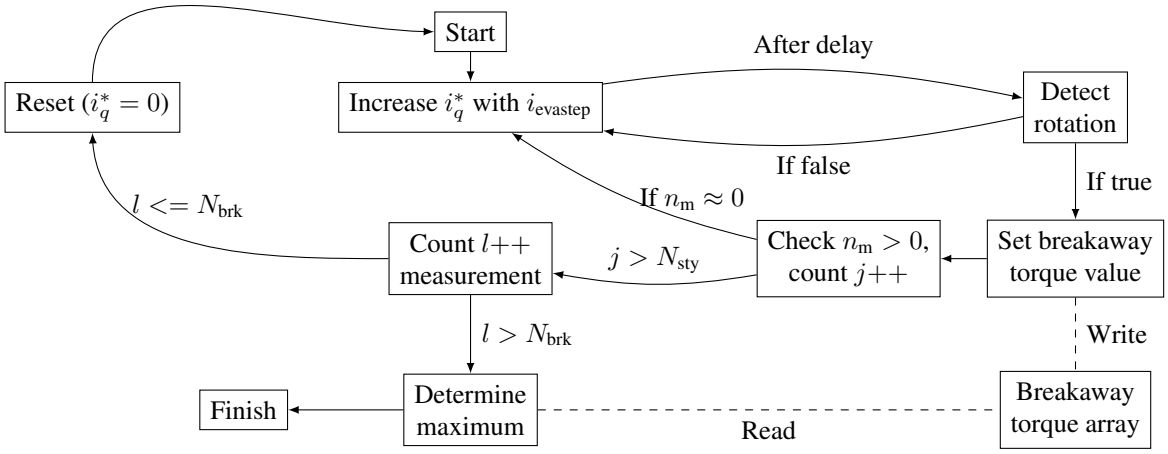


Figure A.6: Breakaway torque ID flow chart.

as a result of i_q current, the delay time between two current steps must be greater than $5\tau_{\text{el}}$. Due to the discrete iteration process, the accuracy of the identified torque arise from the current step size i_{evastep}. To achieve the most accurate torque, s_{step} must be chosen as small as possible. However, for small values of s_{step} the ID duration increases considerably. In order to be able to assess a compromise solution between small measurement uncertainty Δ_{step} and a short ID duration $t_{\text{brk,ID}}$, the two process parameters are evaluated in dependence of s_{step} (see Fig. A.7). The process parameters correspond to

$$t_{\text{brk,ID}} = \frac{2T_{\text{brk}}5\tau_{\text{el}}}{3p\psi_{\text{pm}}I_{\text{N}}s_{\text{step}}} \tag{A.5}$$

$$\Delta_{\text{step}} = \frac{3I_{\text{N}}s_{\text{step}}p\psi_{\text{pm}}}{2T_{\text{brk}}}\,. \tag{A.6}$$

As depicted in Fig. A.7, for small electrical drives a compromise between Δ_{step} and t_{brkID} can be achieved by choosing $s_{\text{step}} \approx 0.001$. However, s_{step} is a modifiable variable during operation. Due to cogging torque and presliding effects, such as frictional hysteresis and stick slip effects, the breakaway torque is a varying measurand, which fluctuats in dependence on the rotor position [Rud12]. For a reduction of the sensitivity and in order to prevent the release of the algorithm with a poorly identified T_{brk}, a counter needs to exceed the heuristically chosen threshold N_{sty}. If the rotor shaft stops while counting, i_{evastep} will be increased. This enables the speed to be debounced, which may be necessary due to the cogging torque and the pre-sliding effects. If the amount of measurements exceeds N_{brk}, the maximum existing T_{brk} is calculated based on the maximum i_q measurement by using (2.28). It must be ensured that $i_d \approx 0$.

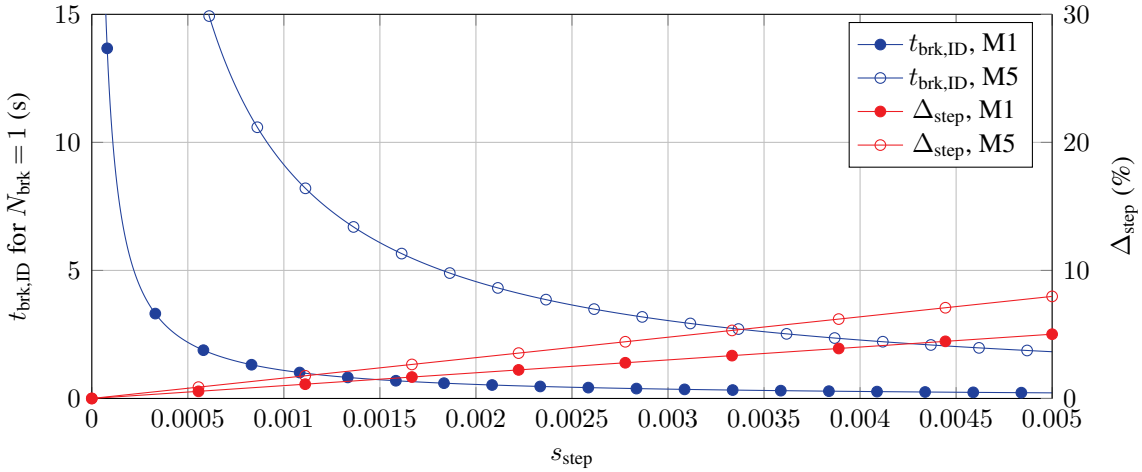


Figure A.7: Impact of the scaling factor s_{step} on the breakaway torque ID process.

A.4.2.2 Results for Breakaway Torque Identification

For the breakaway torque identification, the validity of the results is first verified by the relative standard deviation, which is determined for two motors, see Table A.6. The breakaway torque ID has a low variance overall. However, there is a trend towards higher variance the smaller the motor or its inertia.

Figs. A.8-A.9 show the torque and speed measurements exemplarily for the two test motors. The breakaway torques experimentally identified by the offline ID are shown here in comparison to the reference measurement—torque measuring shaft—and in comparison to the simulation.

For motor M1, the results of experiment and simulation agree well in general. Due to the dependence of the breakaway torque on the rotor position, the experiment shows lower peak speed values compared to the simulation. Nevertheless, the maximum breakaway torque measured with the torque measuring shaft agrees sufficiently well with the experimentally identified, i.e., offline ID, values.

Motor M5 behaves similarly to motor M1, where the identified breakaway torque also depends on the rotor position. In the simulation, however, such dependencies on the rotor position are not modeled, so they do not appear in the simulation results. However, the maximum torque of the torque measuring shaft also agrees very well with the identified maximum torque of the offline ID.

It can be noticed that, compared to motor M1, the overshoot of the speed is lower for motor

Table A.6: Relative standard deviation of breakaway torque ID.

	Motor M1	Motor M5
$\sigma_{T_{\text{brk}}}/\widetilde{T}_{\text{brk}}$	1.76 %	2.61 %

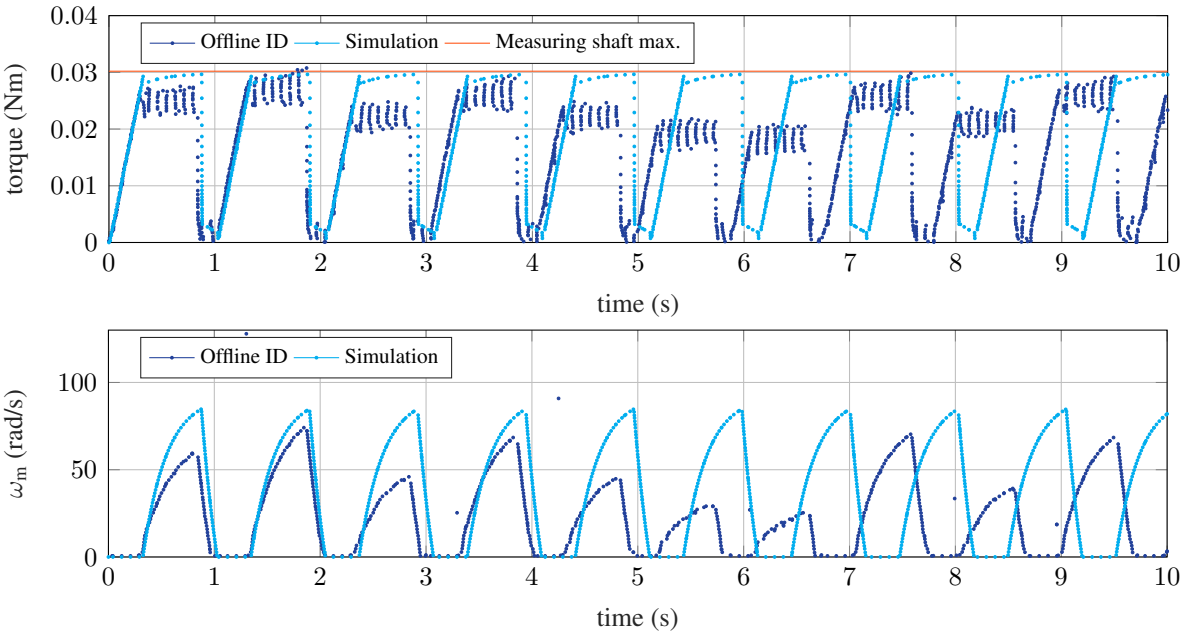


Figure A.8: Motor M1: Timeseries of breakaway torque ID with $N_{sty} = 50$ and $s_{step} = 0.5\,\%$.

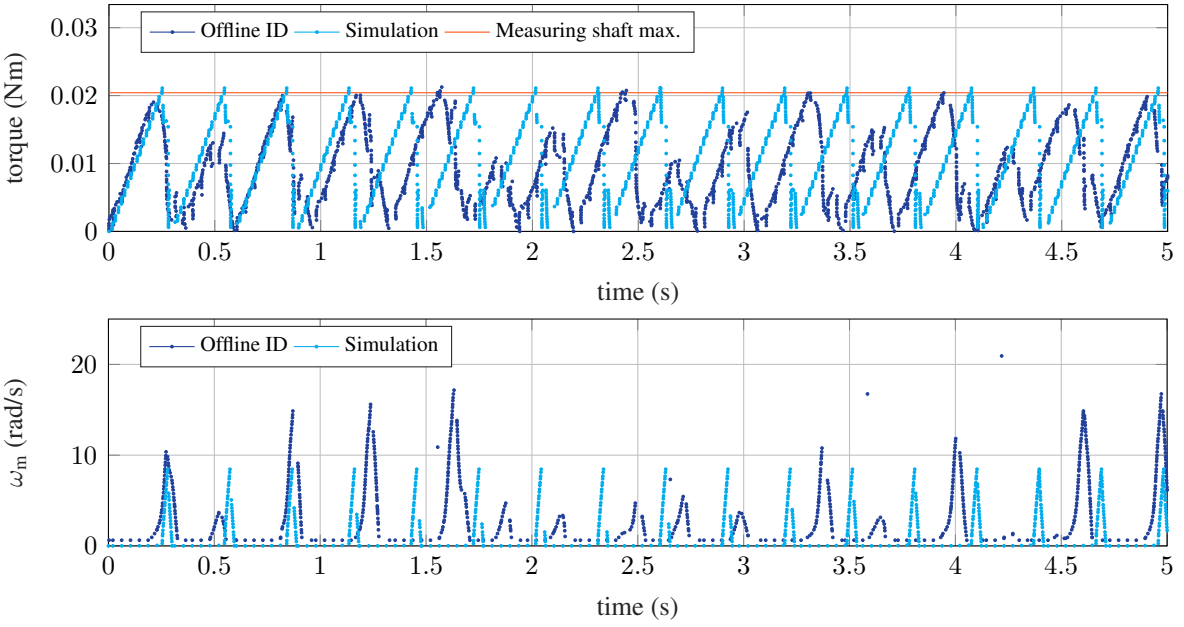
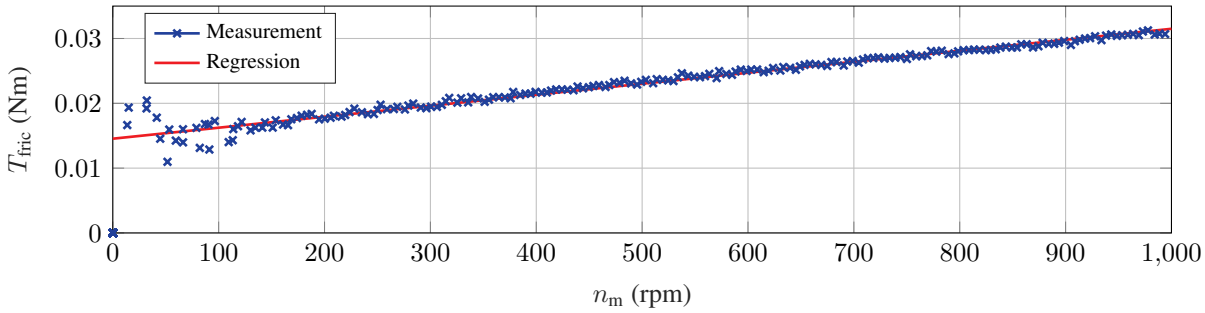


Figure A.9: Motor M5: Timeseries of breakaway torque ID with $N_{sty} = 2$ and $s_{step} = 0.5\,\%$.

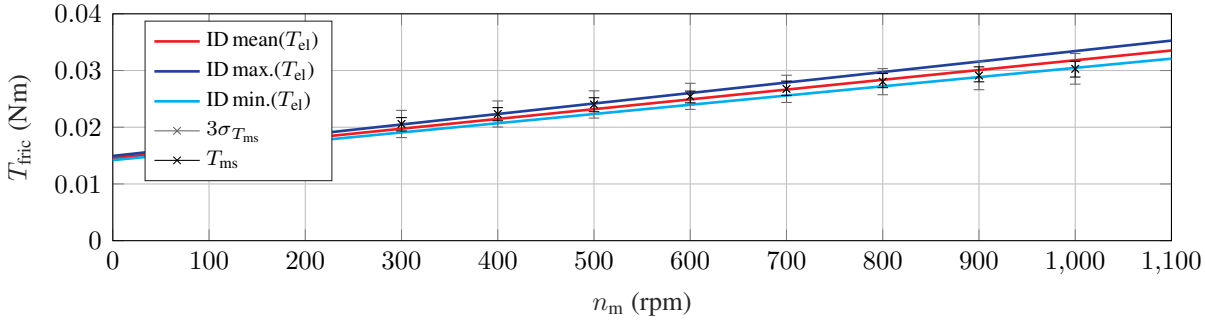
M5 due to the higher sensitivity of the algorithm by using a smaller N_{sty} value. However, N_{sty} is chosen large enough to ensure smooth rotation.

A.4.2.3 Results for Viscous and Coulomb Friction Identification

In the following, the offline ID discussed in Section 4.2.1 is verified. Similar to the breakaway torque, the validity of the results is first verified by the relative standard deviation, which is



(a) Measured values of one offline ID iteration and the corresponding regression line.



(b) Offline ID compared to a reference, i.e., T_{ms} of the measuring shaft, over 15 iterations.

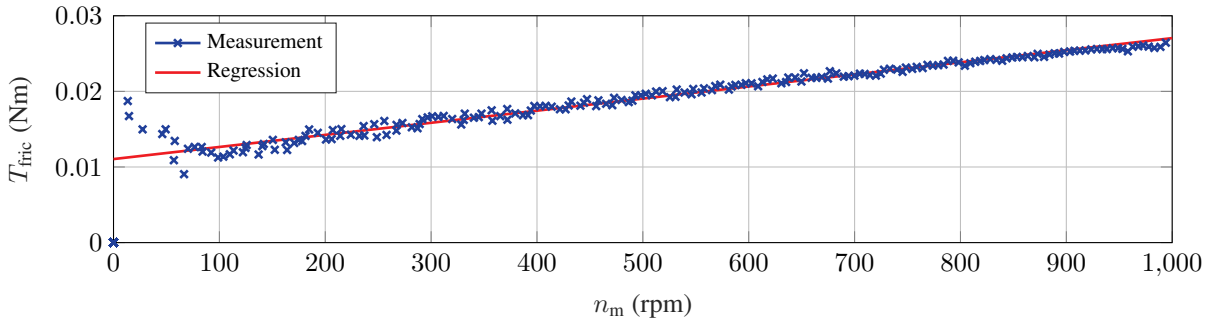
Figure A.10: Motor M1: Offline ID for friction.

determined for two motors, see Table A.7. When evaluating the results, the coulomb friction shows a lower variance than the viscous friction. This observation is consistent with more complex friction models in the literature, such as [AG20], where viscous friction depends on the viscosity and loading condition of the lubricant and is thus more affected by influences such as temperature variations.

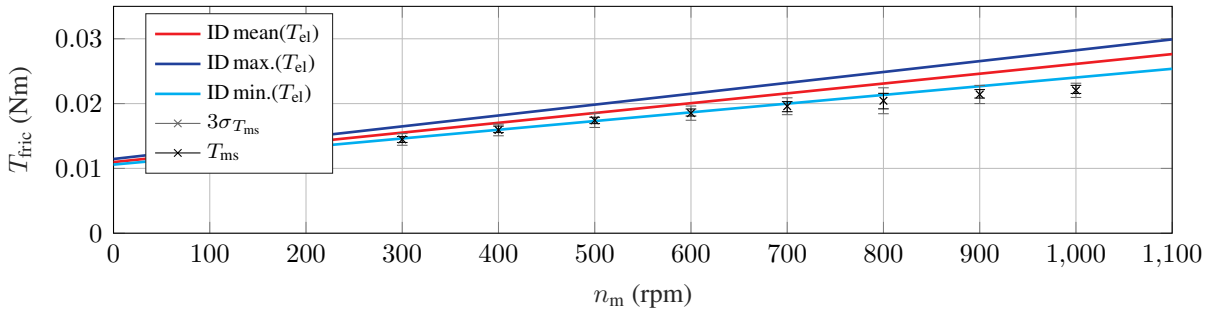
Figs. A.10(a) and A.11(a) show for motor M1 and M5 respectively the identified friction behavior over speed, using the offline ID of Section 4.2.1. Here, for both motors, the measurements are compared with the respective regression line to examine the agreement with the linear friction simplification made in Section 2.2.2.1. In Figs. A.10(b) and A.11(b) the friction from the linear regression is compared with a reference, i.e., torque of the measuring shaft T_{ms}.
As can be noticed, for the considered motors and speed range, the simplification of linear friction fits very well for motor M1 and M5. Here, a good agreement is achieved respectively in the speed range from $200\,\text{rpm}$ to $1000\,\text{rpm}$. However, in the speed range below $200\,\text{rpm}$, the rotation becomes unstable and the torque values cannot contribute to the linear regression.

Table A.7: Relative standard deviation of viscous- and coulomb friction offline ID.

$\sigma_{\text{x}}/\widetilde{x}$	Motor M1	Motor M5
$x = T_{\text{coul}}$	$1.38\,\%$	$1.82\,\%$
$x = d_{\text{fric}}$	$2.44\,\%$	$4.66\,\%$



(a) Measured values of one offline ID iteration and the corresponding regression line.



(b) Offline ID compared to a reference, i.e., T_{ms} of the measuring shaft, over 10 iterations.

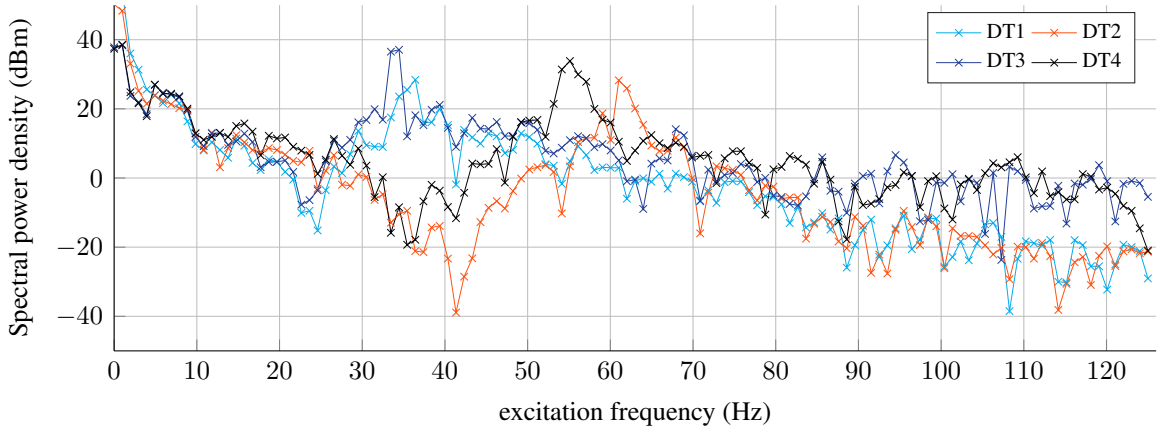
Figure A.11: Motor M5: Offline ID for friction.

More specifically, the results of the torque measuring shaft and the offline ID match very well for motor M1. When comparing the friction curves in Fig. A.10(b), no significant difference can be observed. Only a small difference at higher speeds can be seen, which must be caused by fundamental wave iron losses that corrupt the offline ID measurement.
For motor M5, first, the offline identified friction curves in Fig. A.11(b) show a somewhat larger scatter compared to motor M1 as well as, second, a significantly larger deviation compared to the torque of the measuring shaft. For the latter, the difference between them becomes even bigger at higher speeds. This again indicates iron losses.
Comparing the standard deviation of the offline ID and that of the torque measuring shaft for motor M1 in Fig. A.10(b), it is noticeable that the latter is higher than the former. This can mainly be attributed to the extremely poorly utilized measuring range of the torque measuring shaft and offsets that occur when the direction of rotation changes around zero position.
A significant difference between motor M1 and M5 can be observed by comparing the standard deviations of the torque measuring shaft measurements, i.e., $3\sigma_{T_{\text{ms}}}$, in Figs. A.10(b) and A.11(b). Here, $3\sigma_{T_{\text{ms}}}$ of motor M5 is considerably smaller than that of motor M1. The reason for this is thought to be the lower number of measurements for motor M5.

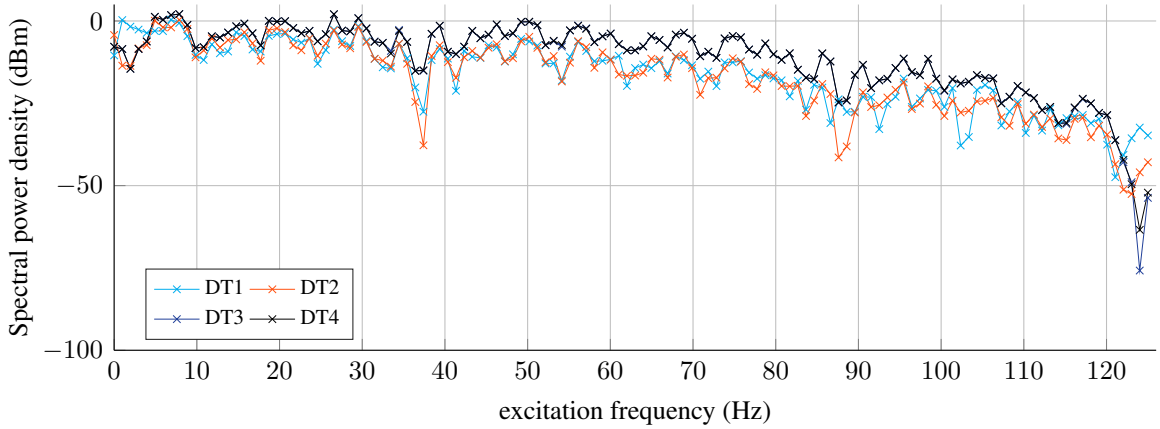
Future work could consider nonlinear friction curves, since the linear simplification does not correspond to reality at high speeds. In this case, a numerical optimization method is required for the regression calculation.

A.4.3 Results for Two-mass System Identification

Fig. A.12 shows the experimental speed and torque measurements for four different drive trains when excited by pseudo-binary signals, see Section 4.2.2.3. This measurements are used to determine the transfer function in Fig. 4.8.



(a) Speed signal.

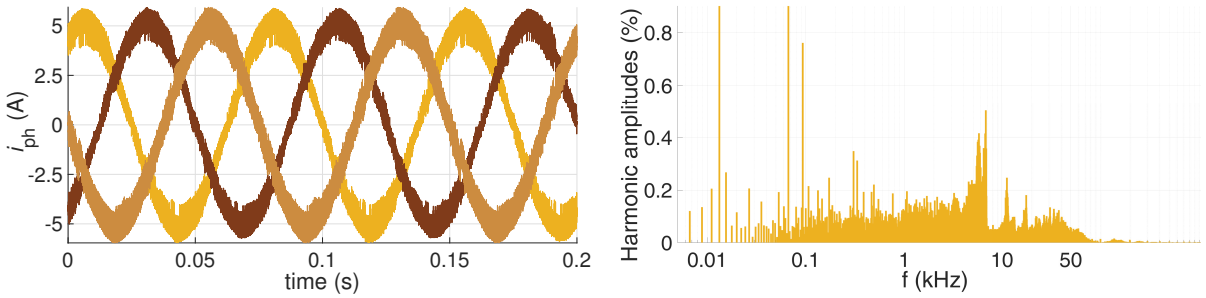


(b) Torque signal.

Figure A.12: Measured signals for the identification of the transfer function (experimental).

A.5 Supplementary Measurements for Current Control

Motor M1 As a comparison to FOC with SVM and FCS-MPC in Fig. 2.26, Fig. A.13 shows the phase currents and spectrum at the same operating point when using DTC.

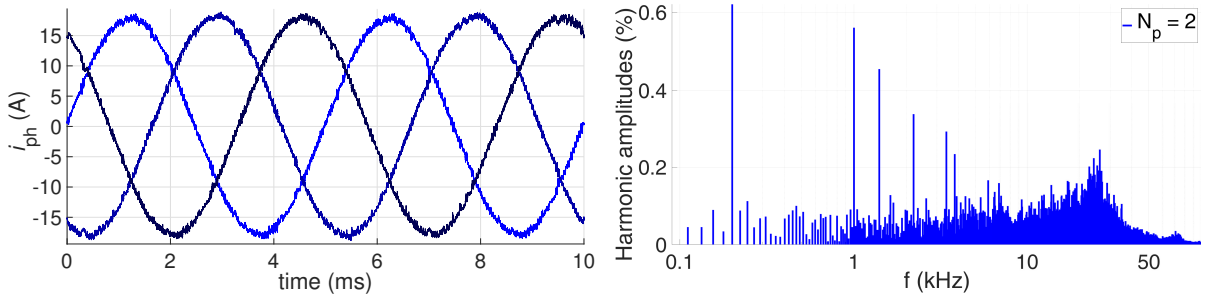


(a) Three-phase stator current for conventional DTC with $|\Delta T_{el}| = 0.011$, $f_{sw} \approx 10.0\,\text{kHz}$.

(b) Stator current spectrum for conventional DTC with $I_{THD} = 7.48\,\%$.

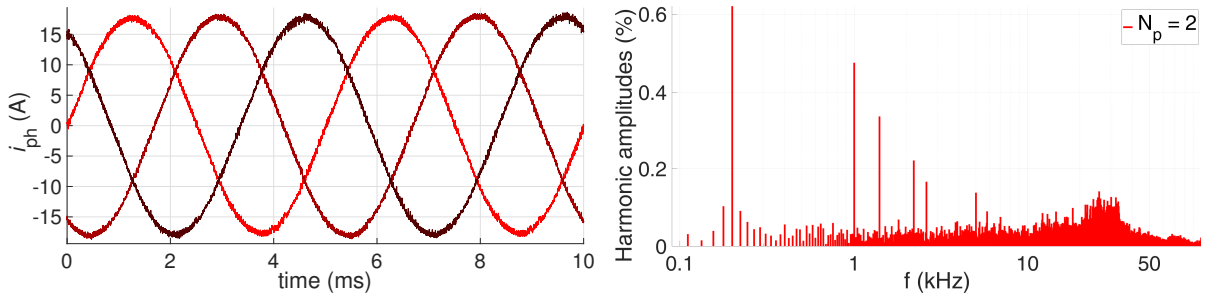
Figure A.13: Motor M1: Three-phase stator current and spectrum for $i_d^* = 0\,\text{A}$, $i_q^* = 5.0\,\text{A}$, $n_m = 200\,\text{rpm}$ with $f_{sw} \approx 10.0\,\text{kHz}$, where the 5^{th} harmonic has $1.36\,\%$ (experimental).

Motor M3 In addition to the measurements in Section 7.6.1, Fig. A.14 shows the phase currents for steady-state operation when operating with 3000 rpm.



(a) Three-phase stator current for L-VSP^{2}CC with $f_{sw} \approx 16.4\,\text{kHz}$.

(b) Stator current spectrum for L-VSP^{2}CC with $I_{THD} = 2.58\,\%$.

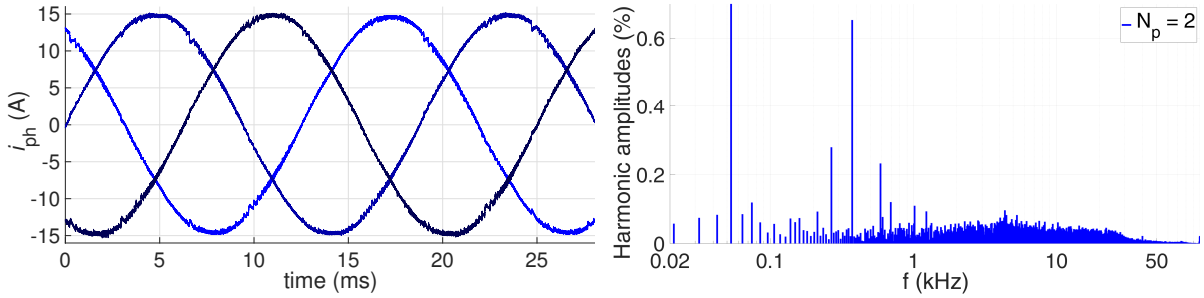


(c) Three-phase stator current for ψ-VSP^{2}CC with $f_{sw} \approx 16.6\,\text{kHz}$.

(d) Stator current spectrum for ψ-VSP^{2}CC with $I_{THD} = 2.01\,\%$.

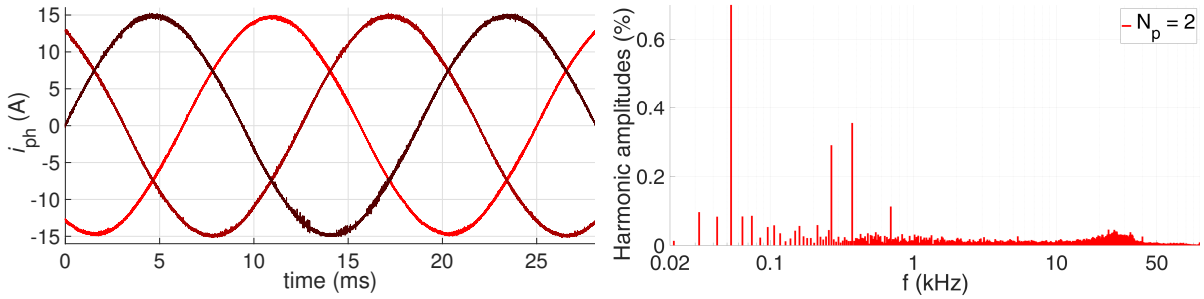
Figure A.14: Motor M3: Three-phase stator current and spectrum for $i_d^* = -5\,\text{A}$, $i_q^* = 18.03\,\text{A}$, $n_m = 3000\,\text{rpm}$ with $N_p = 2$, $f_{sw} \approx 16.0\,\text{kHz}$ (experimental).

Motor M4 In addition to the measurements in Section 7.6.1, Fig. A.15 shows the phase currents for steady-state operation when operating with 800 rpm.



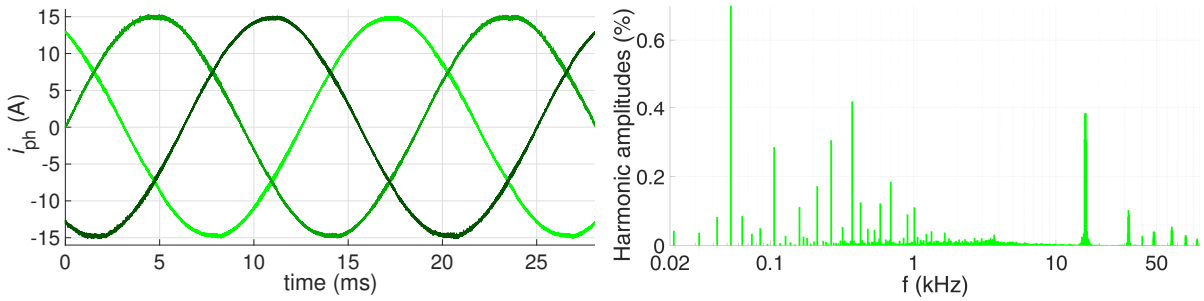
(a) Three-phase stator current for L-VSP^{2}CC with $f_{sw} \approx 15.65\,\text{kHz}$.

(b) Stator current spectrum for L-VSP^{2}CC with $I_{THD} = 1.55\,\%$.



(c) Three-phase stator current for ψ-VSP^{2}CC with $f_{sw} \approx 15.35\,\text{kHz}$.

(d) Stator current spectrum for ψ-VSP^{2}CC with $I_{THD} = 1.02\,\%$.



(e) Three-phase stator current for FOC and SVM with $f_{sw} = 16.0\,\text{kHz}$.

(f) Stator current spectrum for FOC and SVM with $I_{THD} = 1.22\,\%$.

Figure A.15: Motor M4: Three-phase stator current and spectrum for $i_d^* = -5.0\,\text{A}$, $i_q^* = 14.0\,\text{A}$, $n_m = 800\,\text{rpm}$ with $N_p = 2$, $f_{sw} \approx 16.0\,\text{kHz}$ (experimental).

A.6 Supplementary Measurements for Speed Control

DTC with PI-based speed controller As mentioned in Section 2.4, setting the PI gains of the speed controller, that is superimposed on DTC, can be challenging. Under certain circumstances, inadequate gains can damage the drive system due to strong oscillating currents at the stability limit. PI gains are therefore chosen similar to FOC speed control, but reduced by a

factor of 10, since $I_\mathrm{N}/T_\mathrm{N}$ has approximately the same ratio.
Thus, the PI gains of the speed controller correspond to $T_{\mathrm{i},\omega} = 2\,\mathrm{ms}$ and $k_{\mathrm{p},\omega} = 0.03$. For DTC, $\left|\Delta\boldsymbol{\psi}_{dq}\right| = 0$ was chosen to achieve the lowest possible current and torque ripple. A torque hysteresis not equal to zero, on the other hand, has a positive effect on the ripple of current and torque, since this allows zero pointers to be switched in the partial load range of the drive. This hysteresis is chosen as $|\Delta T_\mathrm{el}| = 0.02\,T_\mathrm{N}$. DTC is executed with a frequency of $T_\mathrm{cf} = 100\,\mathrm{kHz}$ since the control algorithm switches the semiconductors directly. However, for a fair comparison, the resulting average switching frequency should be in a similar range as SVM.

Test profile for the evaluation For the evaluation in Section 9.4, the test profile used is shown exemplarily for $n_\ell^* = 500\,\mathrm{rpm}$ in Fig. A.16.

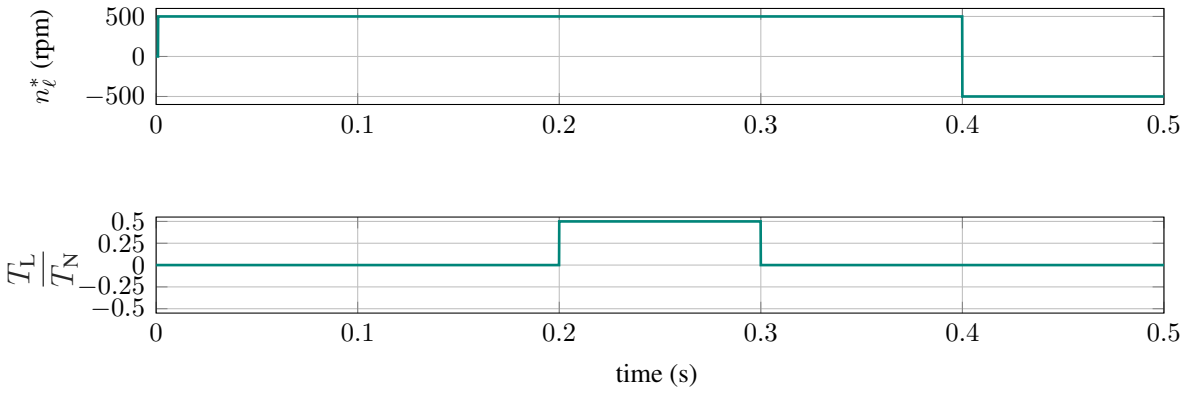


Figure A.16: Test profile for evaluation.

Motor M5 For CCF$_\mathrm{con}$-MPC, Fig. A.17 shows the associated phase currents for the speed and torque shown in Fig. 9.3 and Fig. 9.4 of Section 9.4.

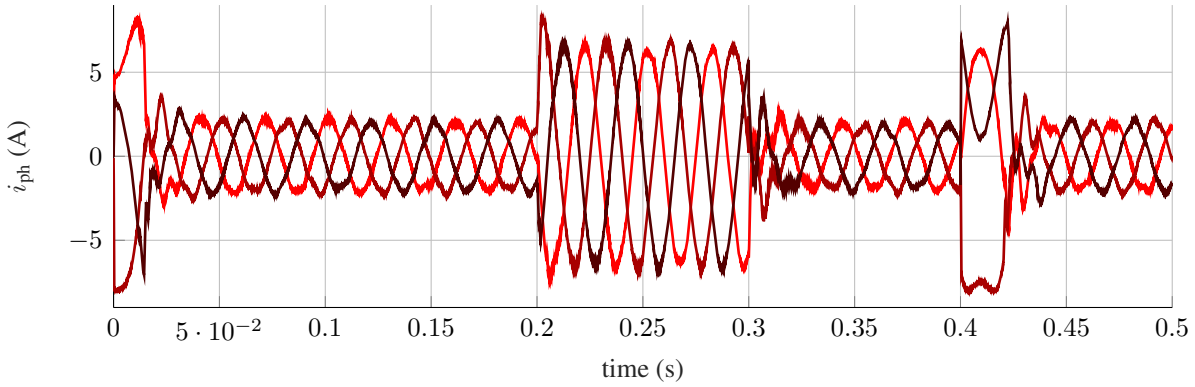


Figure A.17: Motor M5: Phase currents for CCF$_\mathrm{con}$-MPC for a reference speed step from $n_\ell^* = 0\,\mathrm{rpm}$ to $n_\ell^* = 500\,\mathrm{rpm}$ and afterwards to $n_\ell^* = -500\,\mathrm{rpm}$. Moreover, a load step $T_\ell = 0.2\,\mathrm{Nm}$ is shown, see the test profile in Fig. A.16 (experimental).

Bibliography

[ABY18] A. A. Abdelsalam, K. K. Byung, and I. L. Young. A comparison of finite control set and continuous control set model predictive control schemes for speed control of induction motors. IEEE Trans. on Industrial Informatics, 14(4):1334–1346, Aug. 2018.

[Ach11] G. Achilleos. The inverse distance weighted interpolation method and error propagation mechanism - creating a DEM from an analogue topographical map. Journal of Spatial Science, 56(2):283–304, Dec. 2011.

[AG20] Schaeffler AG. Reibung und Erwärmung. Schaeffler AG, Feb. 2020.

[Aki74] H. Akima. A method of bivariate interpolation and smooth surface fitting based on local procedures. Communications of the ACM, 17(1):18–20, Jan. 1974.

[AKMK15] I. Alevras, P. Karamanakos, S. Manias, and R. Kennel. Variable switching point predictive torque control with extended prediction horizon. Seville, Spain, Mar. 2015. IEEE Int. Conf. on Industrial Technology.

[AQ11] R. P. Aguilera and D. E. Quevedo. On stability and performance of finite control set MPC for power converters. Munich, Germany, Oct. 2011. Workshop on Predictive Control of Electrical Drives and Power Electronics.

[AQ13] R. P. Aguilera and D. E. Quevedo. Stability analysis of quadratic MPC with a discrete input alphabet. IEEE Trans. on Automatic Control, 58(12):3190–3196, Dec. 2013.

[AQ15] R. P. Aguilera and D. E. Quevedo. Predictive control of power converters: Designs with guaranteed performance. IEEE Trans. on Industrial Informatics, 11(1):53–63, Feb. 2015.

[ARB+19] P. Acuna, C. A. Rojas, R. Baidya, R. P. Aguilera, and J. E. Fletcher. On the impact of transients on multistep model predictive control for medium-voltage drives. IEEE Trans. on Power Electronics, 34(9):8342–8355, Sep. 2019.

[Aya17] A. N. F. Ayad. Advanced control techniques of impedance source inverters for distributed generation applications. PhD thesis, Chair of Electrical Drive Systems and Power Electronics, Technical University of Munich, 2017.

[BÏ14] S. Büchner. Modellierung und Identifikation elektromechanischer Aktoren am Beispiel einer Dieselregelklappe. PhD thesis, Fakultät für Mathematik und Naturwissenschaften, Technische Universität Ilmenau, 2014.

[BAA+20] R. Baidya, R. P. Aguilera, P. Acuna, T. Geyer, R. A. Delgado, D. E. Quevedo, and H. du T. Mouton. Enabling multistep model predictive control for transient operation of power converters. IEEE Open Journal of the Industrial Electronics Society, 1:284–297, Oct. 2020.

[BAC+19] F. Bertele, U. Ammann, C. Cheshire, M. Neuburger, S. Pirienko, and T. Roeser. Interpretation of measured IPMSM flux tables for parameter identification. Genova, Italy, Sep. 2019. European Conf. on Power Electronics and Applications (EPE'19 ECCE Europe).

[BB98] N. Bianchi and S. Bolognani. Magnetic models of saturated interior permanent magnet motors based on finite element analysis. St. Louis, MO, USA, Oct. 1998. IEEE Industry Applications Conf.

[BB04] L. Ben-Brahim. On the compensation of dead time and zero-current crossing for a PWM-inverter-controlled AC servo drive. IEEE Trans. on Industrial Electronics, 51(5):1113–1118, Oct. 2004.

[Bei00] S. Beineke. Online-Schätzung von mechanischen Parametern, Kennlinien und Zustandsgrößen geregelter elektrischer Antriebe. PhD thesis, Faculty of Computer Science, Electrical Engineering and Mathematics, Paderborn University, 2000.

[Bem] A. Bemporad. Model predictive control: Quadratic programming and explicit MPC. `http://cse.lab.imtlucca.it/~bemporad/teaching/mpc/imt/3-qp_explicit.pdf`. Access: Oct. 2020.

[Ber98] G. Bertotti. Hysteresis in magnetism: For physicists, materials scientists, and engineers. Academic Press, 1st Edition, 1998.

[BIFfMS14] D. Beckmann, J. Immel, O. Frölich, and Leibniz Universität Hannover Institut für Mechatronische Systeme. Online-Parameteridentifikation: Online-Identifikation und Beobachtung von Systemparametern elektrischer

Antriebssysteme zur Nachführung von regelungstechnisch relevanten Parametern: Abschlussbericht; Forschungsvorhaben Nr. 669 I. FVA-Heft. FVA, Forschungsvereinigung Antriebstechnik e.V., Arbeitskreis Geregelte Elektroantriebe, 2014.

[Bin17] A. Binder. Elektrische Maschinen und Antriebe. 2. Auflage, Springer Vieweg, 2017.

[BK18] J. Bonifacio and R. Kennel. On considering saturation and cross-coupling effects for copper loss minimization on highly anisotropic synchronous machines. IEEE Trans. on Industry Applications, 54(5):4177–4185, Oct. 2018.

[Bla72] F. Blaschke. The principle of field orientation as applied to the new transvector closed-loop control system for rotating-field machines. Siemens Review 34(5), 217-219, 1972.

[BLNMMT11] T. Boileau, N. Leboeuf, B. Nahid-Mobarakeh, and F. Meibody-Tabar. Online identification of PMSM parameters: Parameter identifiability and estimator comparative study. IEEE Trans. on Industry Applications, 47(4):1944–1957, Aug. 2011.

[BM15] C. Bordons and C. Montero. Basic principles of MPC for power converters: Bridging the gap between theory and practice. IEEE Industrial Electronics Magazine, 9(3):31–43, Sep. 2015.

[BNMMT08] T. Boileau, B. Nahid-Mobarakeh, and F. Meibody-Tabar. On-line identification of PMSM parameters: Model-reference vs EKF. Edmonton, AB, Canada, Oct. 2008. IEEE Industry Applications Society Annual Meeting.

[BPZ08] S. Bolognani, L. Peretti, and M. Zigliotto. Repetitive-control-based self-commissioning procedure for inverter nonidealities compensation. IEEE Trans. on Industry Applications, 44(5):1587–1596, Sep./Oct. 2008.

[Bri17] M. Brinkhaus. Untersuchung und Modellierung der Istwerterfassung des Stroms für einen Drehmomentbeobachter. Master thesis, Institute ELSYS, Technische Hochschule Nürnberg, 2017.

[CBLB20] G. Cimini, D. Bernardini, S. Levijoki, and A. Bemporad. Embedded model predictive control with certified real-time optimization for synchronous motors. IEEE Trans. on Control Systems Technology, 29(2):893–900, Mar. 2020.

[CDX04] L. Cuiyan, Z. Dongchun, and Z. Xianyi. A survey of repetitive control. Sendai, Japan, Sep./Oct. 2004. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.

[CKK+08] P. Cortes, M. Kazmierkowski, R. Kennel, D. Quevedo, and J. Rodriguez. Predictive control in power electronics and drives. IEEE Trans. on Industrial Electronics, 55(12):4312–4324, Oct. 2008.

[Coua] R+W Couplings. Specification of MK2 miniature bellows coupling with clamping hub series 45. Access: Jan. 2021.

[Coub] R+W Couplings. Specification of MKH miniature coupling with fully split hubs series 20. Access: Jan. 2021.

[CR79] C. R. Cutler and B. L. Ramaker. Dynamic matrix control - A computer control algorithm. Houston, TX, USA, Apr. 1979. AIChE 86th National Meeting.

[CTBZ19] P. G. Carlet, F. Tinazzi, S. Bolognani, and M. Zigliotto. An effective model-free predictive current control for synchronous reluctance motor drives. IEEE Trans. on Industry Applications, 55(4):3781–3790, Jul./Aug. 2019.

[Dep84] M. Depenbrock. Verfahren und Einrichtung zur Regelung einer Drehfeldmaschine, Patent DE3438504A1, Oct. 1984.

[Dep85] M. Depenbrock. Direct self-control of the flux and rotary moment of a rotary-field machine, Patent US4678248A, Oct. 1985.

[DH16] H. Dresig and F. Holzweißig. Maschienendynamik. 12. Auflage, Springer Berlin Heidelberg, 2016.

[DLS+18] S. Decker, A. Liske, D. Schweiker, J. Kolb, and M. Braun. Measurement of two-level inverter induced current slopes at high switching frequencies for control and identification algorithms of electrical machines. Niigata, Japan, May 2018. Int. Conf. on Power Electronics (ICPE-ECCE Asia).

[DMGK20] M. Dorfling, H. du T. Mouton, T. Geyer, and P. Karamanakos. Long-horizon finite-control-set model predictive control with non-recursive sphere decoding on an FPGA. IEEE Trans. on Power Electronics, 35(7):7520–7531, Jul. 2020.

[DMKG17] M. Dorfling, H. du T. Mouton, P. Karamanakos, and T. Geyer. Implementation of a sphere decoder in an FPGA for direct model predictive control with long horizons. Southern Africa, Jan. 2017. Southern African Universities Power Engineering Conf.

[DP04] R. Dittmar and B. Pfeiffer. Modellbasierte prädiktive Regelung: Eine Einführung für Ingenieure. Oldenbourg Verlag, 2004.

[DRB16] S. Decker, J. Richter, and M. Braun. Predictive current control and online parameter identification of interior permanent magnet synchronous machines. Karlsruhe, Germany, Sep. 2016. European Conf. on Power Electronics and Applications (EPE'16 ECCE Europe).

[DSL+19] S. Decker, J. Stoss, A. Liske, M. Brodatzki, J. Kolb, and M. Braun. Online parameter identification of permanent magnet synchronous machines with nonlinear magnetics based on the inverter induced current slopes and the dq-system equations. Genova, Italy, Sep. 2019. European Conf. on Power Electronics and Applications (EPE'19 ECCE Europe).

[Eme05] X. Emery. Simple and ordinary multigaussian Kriging for estimating recoverable reserves. Mathematical Geology, 37(3):295–319, Apr. 2005.

[FHH10] T. Finken, M. Hombitzer, and K. Hameyer. Study and comparison of several permanent-magnet excited rotor types regarding their applicability in electric vehicles. Leipzig, Germany, Nov. 2010. Emobility - Electrical Power Train.

[FKP+14] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. qpOASES: a parametric active-set algorithm for quadratic programming. Math. Prog. Comp., (6):327–363, Apr. 2014.

[FKRK14] E. J. Fuentes, D. Kalise, J. Rodriguez, and R. Kennel. Cascade-free predictive speed control for electrical drives. IEEE Trans. on Industrial Electronics, 61(5):2176–2184, May 2014.

[FRS+09] E. J. Fuentes, J. Rodriguez, C. Silva, S. Diaz, and D. E. Quevedo. Speed control of a permanent magnet synchronous motor using predictive current control. Wuhan, China, May 2009. IEEE Int. Power Electronics and Motion Control Conf.

[FSY12] E. J. Fuentes, C. A. Silva, and J. I. Yuz. Predictive speed control of a two-mass system driven by a permanent magnet synchronous motor. IEEE Trans. on Industrial Electronics, 59(7):2840–2848, Jul. 2012.

[FW75] B. A. Francis and W. M. Wonham. The internal model principle for linear multivariable regulators. Applied Mathematics and Optimization, 2(2):170–194, Jun. 1975.

[Geb18] P. Gebhardt. Automatic identification of the frictional behaviour of permanent magnet synchronous machines. Project work, Institute ELSYS, Technische Hochschule Nürnberg, 2018.

[Geb19] P. Gebhardt. Automatisierte Identifikation mechanischer Parameter in elektrisch angetriebenen Antriebssträngen. Project work, Institute ELSYS, Technische Hochschule Nürnberg, 2019.

[Geb20] P. Gebhardt. Automatisierte Online-Identifikation von Systemparametern permanenterregter Synchronmaschinen zur Nachführung im geregelten Betrieb. Master thesis, Institute ELSYS, Technische Hochschule Nürnberg, 2020.

[Gem15] T. Gemassmer. Effiziente und dynamische Drehmomenteinprägung in hoch ausgenutzten Synchronmaschinen mit eingebetteten Magneten. PhD thesis, Fakultät für Elektrotechnik und Informationstechnik, Karlsruhe Institute of Technology, 2015.

[Gey11] T. Geyer. A comparison of control and modulation schemes for medium-voltage drives: Emerging predictive control concepts versus PWM-based schemes. IEEE Trans. on Industry Applications, 47(3):1380–1389, Mar. 2011.

[Gey17] T. Geyer. Model predictive control of high power converters and industrial drives. John Wiley & Sons, Inc., 2017.

[GKK14] T. Geyer, P. Karamanakos, and R. Kennel. On the benefit of long-horizon direct model predictive control for drives with LC filters. Pittsburgh, PA, USA, Sep. 2014. IEEE Energy Conversion Congress and Exposition.

[GQ14] T. Geyer and D. E. Quevedo. Multistep finite control set model predictive control for power electronics. IEEE Trans. on Power Electronics, 29(12):6836–6846, Feb. 2014.

[GQ15] T. Geyer and D. E. Quevedo. Performance of multistep finite control set model predictive control for power electronics. IEEE Trans. on Power Electronics, 30(3):1633–1644, Mar. 2015.

[GRS+14] C. Garcia, J. Rodriguez, C. Silva, C. Rojas, P. Zanchetta, and H. Abu-Rub. Cascaded predictive speed control. Dallas, TX, USA, Oct. 2014. Annual Conf. of the IEEE Industrial Electronics Society.

[GT18] J. Germishuizen and R. Tanner. Stepped versus fixed rotor position FEA solutions for 2D flux linkage maps in machine design. Alexandroupoli, Greece, Sep. 2018. Int. Conf. on Electrical Machines.

[GVR21] Inc. Grand View Research. Global electric motor sales market size report, 2021-2028. Technical report, https://www.grandviewresearch.com/industry-analysis/electric-motor-market, Jan. 2021.

[GWB+11] O. Gutfleisch, M. Willard, E. Brück, C. H. Chen, S. G. Sankar, and J. Ping Liu. Magnetic materials and devices for the 21st century: Stronger, lighter, and more energy efficient. Advanced Materials - Wiley Online Library, 23(7):821–842, Feb. 2011.

[Hac12] C. M. Hackl. Contributions to high-gain adaptive control in mechatronics. PhD thesis, Chair of Electrical Drive Systems and Power Electronics, Technical University of Munich, 2012.

[Han12] D. Hanselman. Brushless motors - Magnetic design, performance and control. E-Man Press LLC, 2012.

[Has69] K. Hasse. On the dynamics of speed control of a static AC drive with a squirrel-cage induction machine. PhD thesis, Hochschule Darmstadt, 1969.

[HB94] J. Holtz and B. Beyer. Optimal pulsewidth modulation for ac servos and low-cost industrial drives. IEEE Trans. on Industry Applications, 30(4):1039–1047, Jul./Aug. 1994.

[HG15] D. Huger and D. Gerling. The effects of thermal cycling on aging of neodymium-iron-boron magnets. Sydney, NSW, Australia, Jun. 2015. IEEE Int. Conf. on Power Electronics and Drive Systems.

[Hil57] C. Hildreth. A quadratic programming procedure. Naval Research Logistics Quarterly, 4(1):79–85, Mar. 1957.

[HK20] B. Haucke-Korber. Modellbasierte prädiktive Regelung von weich gekoppelten elektrischen Antriebssystemen. Master thesis, Institute ELSYS, Technische Hochschule Nürnberg, 2020.

[HKL98] A. M. Hava, R. J. Kerkman, and T. A. Lipo. A high-performance generalized discontinuous PWM algorithm. IEEE Trans. on Industry Applications, 34(5):1059–1071, Sep./Oct. 1998.

[HKM20] 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. TechRxiv Power, Energy and Industry Applications, DOI: 10.36227/techrxiv.13270949.v1:1–8, Nov. 2020.

[HL03] D. G. Holmes and T. A. Lipo. Pulse width modulation for power converters. Principles and practice. Wiley-IEEE Press, 2003.

[Hoe17] M. Hoerner. Automated parameter identification of fractional horsepower permanent magnet synchronous motors. Project work, Institute ELSYS, Technische Hochschule Nürnberg, 2017.

[Hoe18] M. Hoerner. Automatisierte Identifikation nichtlinearer Flusskennfelder permanenterregter Synchronmaschinen. Master thesis, Institute ELSYS, Technische Hochschule Nürnberg, 2018.

[Hol92] J. Holtz. Pulsewidth modulation - A survey. IEEE Trans. on Industrial Electronics, 39(5):410–420, Dec. 1992.

[Hol94] J. Holtz. Pulsewidth modulation for electronic power conversion. Proceedings of the IEEE, 82(8):1194–1214, Aug. 1994.

[Hol16] J. Holtz. Advanced PWM and predictive control—An overview. IEEE Trans. on Industrial Electronics, 63(6):3837–3844, Jun. 2016.

[Huf21] D. Hufnagel. Optimierte Online-Identifizierung von Flusskennfeldern bei nichtlinearen PMSM über einen weiten Betriebsbereich. Master thesis, Institute ELSYS, Technische Hochschule Nürnberg, 2021.

[HW96] G. Hillerström and K. Walgama. Repetitive control theory and applications - A survey. IFAC Proceedings Volumes, 29(1):1446–1451, Jun./Jul. 1996.

[HWD17] M. Hoerner, S. Wendel, and A. Dietz. Automated parameter identification of fractional horse power permanent magnet synchronous motors. Saarbruecken, Germany, Sep. 2017. GMM/ETG-Symposium on Innovative Small Drives and Micro-Motor Systems.

[HYZ+15] L. Huang, X. Yang, B. Zhang, L. Qiao, H. Li, and M. Tian. Hierarchical model predictive control of modular multilevel matrix converter for low frequency AC transmission. Seoul, South Korea, Jun. 2015. Int. Conf. on Power Electronics (ICPE-ECCE Asia).

[IEE14] IEEE Std 519-2014, Revision of IEEE Std 519-1992. IEEE recommended practices and requirements for harmonic control in electrical power systems, Jun. 2014.

[IM11] R. Isermann and M. Münchhof. Identification of dynamic systems. Springer Berlin Heidelberg, 2011.

[Inc] Xilinx Inc. AXI reference guide - UG1037. Access: Jan. 2021.

[Ins19] Texas Instruments. InstaSPIN-FOC and InstaSPIN-MOTION user's guide (SPRUHJ1F), Jun. 2019.

[ITF+12] A. Imura, T. Takahashi, M. Fujitsuna, T. Zanma, and S. Doki. Dead-time compensation in model predictive instantaneous-current control. Montreal, QC, Canada, Oct. 2012. Annual Conf. of the IEEE Industrial Electronics Society.

[JHH+09] S. Jacobs, D. Hectors, F. Henrotte, M. Hafner, M. H. Gracia, K. Hameyer, P. Goes, D. R. Romera, E. Attrazic, and S. Paolinelli. Magnetic material optimization for hybrid vehicle PMSM drives. Stavanger, Norway, May 2009. The Int. Battery, Hybrid and Fuel Cell Electric Vehicle Symposium.

[Jor24] H. Jordan. Die ferromagnetischen Konstanten für schwache Wechselfelder. Eletr. Nachr. Techn., 1:7–29, Sep. 1924.

[JS95] J. K. Ji and S. K. Sul. Kalman filter and LQ based speed controller for torsional vibration suppression in a 2-mass motor drive system. IEEE Trans. on Industrial Electronics, 42(6):564–571, Dec. 1995.

[KAK18] P. Karamanakos, A Ayad, and R. Kennel. A variable switching point predictive current control strategy for Quasi-Z-source inverters. IEEE Trans. on Industry Applications, 54(2):1469–1480, Apr. 2018.

[Kar13] P. Karamanakos. Model predictive control strategies for power electronics converters and AC drives. PhD thesis, School of Electrical and Computer Engineering, National Technical University of Athens, 2013.

[KBY18] R. Kerid, H. Bourouina, and R. Yahiaoui. Parameter identification of PMSM using EKF with temperature variation tracking in automotive applications. Periodicals of Engineering and Natural Sciences, 6(2):109–119, Nov. 2018.

[KCMW14] E. Kurniawan, Z. Cao, O. Mahendra, and R. Wardoyo. A survey on robust repetitive control and applications. Batu Ferringhi, Malaysia, Nov. 2014. IEEE Int. Conf. on Control System, Computing and Engineering.

[Kel12] S. L. Kellner. Parameteridentifikation bei permanenterregten Synchronmaschinen. PhD thesis, Faculty of Engineering, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2012.

[KG20] P. Karamanakos and T. Geyer. Guidelines for the design of finite control set model predictive controllers. IEEE Trans. on Power Electronics, 35(7):7434–7450, Jul. 2020.

[KGA18] P. Karamanakos, T. Geyer, and R. P. Aguilera. Long-horizon direct model predictive control: Modified sphere decoding for transient operation. IIEEE Trans. on Industry Applications, 54(6):6060–6070, Nov./Dec. 2018.

[KGK15] P. Karamanakos, T. Geyer, and R. Kennel. Suboptimal search strategies with bounded computational complexity to solve long-horizon direct model predictive control problems. Montreal, QC, Canada, Sep. 2015. IEEE Energy Conversion Congress and Exposition.

[KGK18] P. Karamanakos, T. Geyer, and R. Kennel. On the choice of norm in finite control set model predictive control. IEEE Trans. on Power Electronics, 33(8):7105–7117, Aug. 2018.

[KGMK16] P. Karamanakos, T. Geyer, H. du T. Mouton, and R. Kennel. Computationally efficient sphere decoding for long-horizon direct model predictive control. Milwaukee, WI, USA, Sep. 2016. IEEE Energy Conversion Congress and Exposition.

[KGN+14] P. Karamanakos, T. Geyer, Oikonomou N., Kieferndorf F. D., and Manias S. Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics. IEEE Industrial Electronics Magazine, 8(1):32–43, Mar. 2014.

[KHK11] J. B. Kim, K. Y. Hwang, and B. I. Kwon. Optimization of two-phase in-wheel IPMSM for wide speed range by using the Kriging model based on Latin hypercube sampling. IEEE Trans. on Magnetics, 47(5):1078–1081, May 2011.

[KJP11] O. König, S. Jakubek, and G. Prochart. Model predictive control of a battery emulator for testing of hybrid and electric powertrains. Chicago, IL, USA, Sep. 2011. IEEE Vehicle Power and Propulsion Conf.

[KK13] S. Kuehl and R. Kennel. Measuring magnetic properties of synchronous machines by applying angle estimation techniques known from sensorless control. Denver, CO, USA, Sep. 2013. IEEE Energy Conversion Congress and Exposition.

[Kle05] R. Klein. Algorithmische Geometrie: Grundlagen, Methoden, Anwendungen. Springer Berlin Heidelberg, 2005.

[KLGK20] 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, 1:95–114, Aug. 2020.

[KLK12] S. Kuehl, P. Landsmann, and R. Kennel. Bivariate polynomial approximation of cross-saturated flux curves in synchronous machine models. Florence, Italy, Sep. 2012. IEEE Int. Energy Conf. and Exhibition.

[KMG18] P. Karamanakos, R. Mattila, and T. Geyer. Fixed switching frequency direct model predictive control based on output current gradients. Washington, DC, USA, Oct. 2018. Annual Conf. of the IEEE Industrial Electronics Society.

[KNG20] P. Karamanakos, M. Nahalparvari, and T. Geyer. Fixed switching frequency direct model predictive control with continuous and discontinuous modulation for grid-tied converters with LCL filters. IEEE Trans. on Control Systems Technology, DOI: 10.1109/TCST.2020.3008030:1–16, Jul. 2020.

[KPR+15] S. Kouro, M. A. Perez, J. Rodriguez, A. M. Llor, and H. A. Young. Model predictive control - MPC's role in the evolution of power electronics. IEEE Industrial Electronics Magazine, 9(7):8–21, Dec. 2015.

[KR59] K. P. Kovács and I. Rácz. Transiente Vorgänge in Wechselstrommaschinen. Bd. 2. Verlag der Ungarischen Akademie der Wissenschaften, 1959.

[KSK+14] P. Karamanakos, P. Stolze, R. Kennel, S. Manias, and H. du T. Mouton. Variable switching point predictive torque control of induction machines. IEEE Journal of Emerging and Selected Topics in Power Electronics, 2(2):285–295, Jun. 2014.

[KSP11] S. L. Kellner, M. Seilmeier, and B. Piepenbreier. Impact of iron losses on parameter identification of permanent magnet synchronous machines. Nuremberg, Germany, Sep. 2011. Int. Electric Drives Production Conf.

[KWB21] W. Kirchgässner, O. Wallscheid, and J. Böcker. Estimating electric motor temperatures with deep residual machine learning. IEEE Trans. on Power Electronics, 36(7):7480–7488, Jul. 2021.

[KZPF15] D. Kouzoupis, A. Zanelli, H. Peyrl, and H. J. Ferreau. Towards proper assessment of QP algorithms for embedded model predictive control. Linz, Austria, Jul. 2015. European Control Conf.

[Lan14] P. Landsmann. Sensorless control of synchronous machines by linear approximation of oversampled current. PhD thesis, Chair of Electrical Drive Systems and Power Electronics, Technical University of Munich, 2014.

[LB13] M. Leuer and J. Böcker. Fast online model predictive control of IPMSM using parallel computing on FPGA. Chicago, IL, USA, May 2013. IEEE Int. Electric Machines & Drives Conf.

[LB14] M. Leuer and J. Böcker. Real-time implementation of an online model predictive control for IPMSM using parallel computing on FPGA. Hiroshima, Japan, May 2014. Int. Conf. on Power Electronics (ICPE-ECCE Asia).

[Leo96] W. Leonhard. Control of electrical drives. 2nd Completely Revised and Enlarged Edition, Springer Berlin Heidelberg, 1996.

[Leu14] M. Leuer. Modellprädiktive Regelung permanent erregter Synchronmotoren im Antriebsstrang von Automobilen. PhD thesis, Institute for Power Electronics and Electrical Drives, Paderborn University, 2014.

[Lin10] A. Linder. Model-based predictive control of electric drives. PhD thesis, Cuvillier Verlag Göttingen, 2010.

[LJS+17] A. Lidozzi, C. Ji, L. Solero, P. Zanchetta, and F. Crescimbini. Digital deadbeat and repetitive combined control for a stand-alone four-leg VSI. IEEE Trans. on Industry Applications, 53(6):5624–5633, Nov./Dec. 2017.

[LK12] P. Landsmann and Kennel. Saliency-based sensorless predictive torque control with reduced torque ripple. IEEE Trans. on Power Electronics, 27(10):4311–4320, Oct. 2012.

[LKF+16] J. I. Leon, S. Kouro, L. G. Franquelo, J. Rodriguez, and B. Wu. The essential role and the continuous evolution of modulation techniques for voltage-source inverters in the past, present, and future power electronics. IEEE Trans. on Industrial Electronics, 63(5):2688–2701, Jan. 2016.

[LKG+17] E. Liegmann, P. Karamanakos, T. Geyer, H. du T. Mouton, and R. Kennel. Long-horizon direct model predictive control with active balancing of the neutral point potential. Pilsen, Czech Republic, Sep. 2017. IEEE Int. Symposium on Predictive Control of Electrical Drives and Power Electronics.

[LL00] E. Levi and V. A. Levi. Impact of dynamic cross-saturation on accuracy of saturated synchronous machine models. IEEE Trans. on Energy Conversion, 15(2):224–230, Jun. 2000.

[Lue69] D. G. Luenberger. Optimization by vector space methods. John Wiley & Sons, Inc., 1969.

[Lun17] J. Lunze. Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung. 9. Auflage, Springer Berlin Heidelberg, 2017.

[LW14] H. Lutz and W. Wendt. Taschenbuch der Regelungstechnik: Mit MATLAB und Simulink, 11. ergänzte Auflage. Europa-Lehrmittel, 2014.

[LZC+11] K. Liu, Q. Zhang, J. Chen, Z.Q. Zhu, and J. Zhang. Online multiparameter estimation of nonsalient-pole PM synchronous machines with temperature variation tracking. IEEE Trans. on Industrial Electronics, 58(5):1776–1788, May 2011.

[LZW+20] 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 Trans. on Power Electronics, 35(10):10778–10788, Oct. 2020.

[Mat] The MathWorks. HDL coder. `https://uk.mathworks.com/products/hdl-coder.html`. Access: Sep. 2020.

[MCDA+19] A. Mora, R. Cárdenas-Dobson, R. P. Aguilera, A. Angulo, F. Donoso, and J. Rodríguez. Computationally efficient cascaded optimal switching sequence MPC for grid-connected three-level NPC converters. IEEE Trans. on Power Electronics, 34(12):12464–12475, Dec. 2019.

[MCK18] I. McInerney, G. A. Constantinides, and E. C. Kerrigan. A survey of the implementation of linear model predictive control on FPGAs. IFAC Conf. on Nonlinear Model Predictive Control NMPC, 51(20):381–387, Aug. 2018.

[MD17] R. Marchtaler and S. Dingler. Kalman-Filter: Einführung in die Zustandsschätzung und ihre Anwendung für eingebettete Systeme. 1. Auflage, Springer Fachmedien Wiesbaden GmbH, 2017.

[Mey10] M. Meyer. Wirkungsgradoptimierte Regelung hoch ausgenutzter Permanentmagnet-Synchronmaschinen im Antriebsstrang von Automobilen. PhD thesis, Faculty of Computer Science, Electrical Engineering and Mathematics, Paderborn University, 2010.

[Min13] F. Mink. Modellierung und hochdynamische Stromregelung von PM-Synchronmaschinen unter Berücksichtigung von Sättigungseffekten. PhD thesis, Fachbereich Elektrotechnik und Informationstechnik, Technische Universität Darmstadt, 2013.

[MM98] J. Maes and J. Melkebeek. Discrete time direct torque control of induction motors using back-EMF measurement. St. Louis, MO, USA, Oct. 1998. IEEE Industry Applications Society Annual Meeting.

[Mor09] M. Morari. Predicting the future of model predictive control. Automatic Control Laboratory, ETH Zürich, 2009.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. Automatica, 36(6):789–814, Jun. 2000.

[MTSL08] K. J. Meessen, P. Thelin, J. Soulard, and E. A. Lomonova. Inductance calculations of permanent-magnet synchronous machines including flux change and self- and cross-saturations. IEEE Trans. on Magnetics, 44(10):2324–2331, Oct. 2008.

[MUC+19] A. Mora, M. Urrutia, R. Cárdenas, A. Angulo, M. Espinoza, M. Díaz, and P. Lezana. Model-predictive-control-based capacitor voltage balancing strategies for modular multilevel converters. IEEE Trans. on Industrial Electronics, 66(3):2432–2443, Mar. 2019.

[MWB11] J. Mattingley, Y. Wang, and S. Boyd. Receding horizon control: Automatic generation of high-speed solvers. IEEE Control Systems Magazine, 31(3):52–65, Jun. 2011.

[Nes83] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. Soviet Math. Dokl., 27(2):372–376, 1983.

[NKG19] M. Nahalparvari, P. Karamanakos, and T. Geyer. Gradient-based fixed switching frequency direct model predictive control for grid-connected converters with LCL filters. Genova, Italy, Sep. 2019. European Conf. on Power Electronics and Applications (EPE'19 ECCE Europe).

[Now16] S. Nowoisky. Verfahren zur Identifikation nichtlinearer dynamischer Getriebemodelle. PhD thesis, Universitätsverlag der TU Berlin, Technische Universität Berlin, 2016.

[NPRS14] Q. K. Nguyen, M. Petrich, and J. Roth-Stielow. Implementation of the MTPA and MTPV control with online parameter identification for a high speed IPMSM used as traction drive. Hiroshima, Japan, May 2014. Int. Conf. on Power Electronics (ICPE-ECCE Asia).

[NT84] T. Noguchi and I. Takahashi. Quick torque response control of an induction motor based on a new concept. IEEJ Tech. Meeting Rotating Mach. (in Japanese), RM84-76:61–70, Sep. 1984.

[OKBH14] T. Orlowska-Kowalska, F. Blaabjerg, and Rodriguez J. (Hg.). Advanced and intelligent control in power electronics and drives. Studies in Computational Intelligence, volume 531, 2014.

[OKZ20] L. Ortombina, P. Karamanakos, and M. Zigliotto. Robustness analysis of long-horizon direct model predictive control: Permanent magnet synchronous motor drives. Aalborg, Denmark, Nov. 2020. Workshop on Control and Modeling for Power Electronics.

[PB13a] M. Preindl and S. Bolognani. Comparison of direct and PWM model predictive control for power electronic and drive systems. Long Beach, CA, USA, Mar. 2013. IEEE Applied Power Electronics Conf. and Exposition.

[PB13b] M. Preindl and S. Bolognani. Model predictive direct speed control with finite control set of PMSM drive systems. IEEE Trans. on Power Electronics, 28(2):1007–1015, Feb. 2013.

[PB13c] M. Preindl and S. Bolognani. Model predictive direct torque control with finite control set for PMSM drive systems, Part 1: Maximum torque per ampere operation. IEEE Trans. on Industrial Informatics, 9(4):1912–1921, Nov. 2013.

[PB13d] M. Preindl and S. Bolognani. Model predictive direct torque control with finite control set for PMSM drive systems, Part 2: Field weakening operation. IEEE Trans. on Industrial Informatics, 9(2):648–657, May 2013.

[PBD13] M. Preindl, S. Bolognani, and C. Danielson. Model predictive torque control with PWM using fast gradient method. Long Beach, CA, USA, Mar. 2013. IEEE Applied Power Electronics Conf. and Exposition.

[PPS+16] C. Pedret, A. Poncet, K. Stadler, A. Toller, A. Glattfelder, A. Bemporad, and M. Morari. Model-varying predictive control of a nonlinear system. Barcelona, Spain, 2016. Internal report in Computer Science Dept. ETSE de la Universitat Autónoma de Barcelona.

[Pra] Burster Praezisionsmesstechnik. Datasheet - Burster precision torque sensor model 8661-5001-v0xxx. Access: Jan. 2021.

[Pre13] M. Preindl. Novel model predictive control of a PM synchronous motor drive. PhD thesis, Department of Industrial Engineering, University of Padua, 2013.

[QD15] N. P. Quang and J.-A. Dittrich. Vector control of three-phase AC machines: System development in the practice. 2. Edition, Springer Berlin Heidelberg, 2015.

[Rau03] M. Rau. Nichtlineare modellbasierte prädiktive Regelung auf Basis lernfähiger Zustansraummodelle. PhD thesis, Chair of Electrical Drive Systems and Power Electronics, Technical University of Munich, 2003.

[RC12] J. Rodriguez and P. Cortes. Predictive control of power converters and electrical drives. Wiley-IEEE Press, 2012.

[RDD14] J. Richter, A. Dollinger, and M. Doppelbauer. Iron loss and parameter measurement of permanent magnet synchronous machines. Berlin, Germany, Nov. 2014. Int. Conf. on Electrical Machines.

[RFT+16] D. D. Reigosa, D. Fernandez, T. Tanimoto, T. Kato, and F. Briz. Permanent-magnet temperature distribution estimation in permanent-magnet synchronous

machines using back electromotive force harmonics. IEEE Trans. on Industry Applications, 52(4):3093–3103, Jul./Aug. 2016.

[RGD14] J. Richter, T. Gemaßmer, and M. Doppelbauer. Predictive current control of saturated cross-coupled permanent magnet synchronous machines. Ischia, Italy, Jun. 2014. Int. Symposium on Power Electronics, Electrical Drives, Automation and Motion.

[Ric16] J. Richter. Modellbildung, Parameteridentifikation und Regelung hoch ausgenutzter Synchronmaschinen. PhD thesis, Fakultät für Elektrotechnik und Informationstechnik, Karlsruhe Institute of Technology, 2016.

[Rie] D. Riedl. Inverse distance weighting. `https://homepage.univie.ac.at/doris.riedl/model0910/tut_geostat/IDW.htm`. Access: Apr. 2020.

[RJM09] S. Richter, C. N. Jones, and Morari M. Real-time input-constrained MPC using fast gradient methods. Shanghai, China, Dec. 2009. IEEE Conf. on Decision and Control.

[RKCD20] M. Rossi, P. Karamanakos, and F. Castelli-Dezza. Indirect model predictive control for a grid-tied three-level neutral point clamped converter with an LCL filter. Detroit, MI, USA, Oct. 2020. IEEE Energy Conversion Congress and Exposition.

[RKE+13] 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 Trans. on Industrial Informatics, 9(2):1003–1016, May 2013.

[RM09] J. B. Rawlings and D. Q. Mayne. Model predictive control: Theory and design. Madison, WI: Nob Hill, 2009.

[RMM10] S. Richter, S. Mariéthoz, and Morari M. High-speed online MPC based on a fast gradient method applied to power converter control. Baltimore, MD, USA, Jul. 2010. Proceedings of the 2010 American Control Conf.

[RSF+22] A. Rüetschi, P. Syrpas, B. Flak, K. Tomzik, and P. K. Steimer. Heterogeneous control platform design for power conversion systems. IEEE Trans. on Industrial Informatics, 18(5):2934–2942, May 2022.

[RTAG16] A. Rabiei, T. Thiringer, M. Alatalo, and E. A. Grunditz. Improved maximum-torque-per-ampere algorithm accounting for core saturation, cross-coupling effect, and temperature for a PMSM intended for vehicular applications. IEEE Trans. on Transportation Electrification, 2(2):150–159, Jun. 2016.

[Rud12] M. Rudermann. Zur Modellierung und Kompensation dynamischer Reibung in Aktuatorsystemen. PhD thesis, Lehrstuhl für Regelungssystemtechnik, Technical University of Dortmund, 2012.

[RWR98] C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. Journal of Optimization Theory and Applications, 99(3):723–757, Dec. 1998.

[SA09] R. C. Stevenson and A. Arbor. The role of coenergy & the development of a comprehensive analytical model for a PM motor. Dearborn, MI, USA, Sep. 2009. IEEE Vehicle Power and Propulsion Conf.

[Sae15] S. Saeidi. FPGA-based nonlinear model predictive control of electric drives. PhD thesis, Chair of Electrical Drive Systems and Power Electronics, Technical University of Munich, 2015.

[Sah08] B. Sahhary. Elektrische Antriebe mit dauermagneterregten Maschinen im dynamischen sensorlosen Betrieb. PhD thesis, Faculty of Electrical Engineering, University of the Federal Armed Forces Hamburg, 2008.

[SB17] D. Schröder and M. Buss. Intelligente Verfahren - Identifikation und Regelung nichtlinearer Systeme. 2. Auflage, Springer Vieweg, 2017.

[SBGB20] B. Stellato, G. Banjac, P. Goulart, and A. Bemporad. OSQP: an operator splitting solver for quadratic programs. Mathematical Programming Computation, 12(4):637–672, Feb. 2020.

[SC07] R. Scattolini and P. Colaneri. Hierarchical model predictive control. New Orleans, LA, USA, Dec. 2007. IEEE Conf. on Decision and Control.

[Sch02] F. Schütte. Automatisierte Reglerinbetriebnahme für elektrische Antriebe mit schwingungsfähiger Mechanik. PhD thesis, Faculty of Computer Science, Electrical Engineering and Mathematics, Paderborn University, 2002.

[Sch13] D. Schröder. Elektrische Antriebe - Grundlagen. 5. Auflage, Springer Vieweg, 2013.

[Sch15] D. Schröder. Elektrische Antriebe - Regelung von Antriebssystemen. 4. Auflage, Springer Vieweg, 2015.

[Sch19] A. Schmidt. Modellprädiktive Regelung einer permanenterregten Synchronmaschine an der Aussteuergrenze. Bachelor thesis, Institute ELSYS, Technische Hochschule Nürnberg, 2019.

[SGG17] B. Stellato, T. Geyer, and P. J. Goulart. High-speed finite control set model predictive control for power electronics. IEEE Trans. on Power Electronics, 32(5):4007–4020, May 2017.

[SKK+15] P. Stolze, P. Karamanakos, R. Kennel, S. Manias, and C. Endisch. Effective variable switching point predictive current control for ac low-voltage drives. Int. Journal of Control, 88(7):1366–1378, Jul. 2015.

[SKR17] M. Siami, D. A. Khaburi, and J. Rodríguez. Torque ripple reduction of predictive torque control for PMSM drives with parameter mismatch. IEEE Trans. on Power Electronics, 32(9):7160–7168, Sep. 2017.

[SKT+13] P. Stolze, P. Karamanakos, M. Tomlinson, R. Kennel, H. du T. Mouton, and S. Manias. Heuristic variable switching point predictive current control for the three-level neutral point clamped inverter. Munich, Germany, Oct. 2013. IEEE Int. Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics.

[SP14] M. Seilmeier and B. Piepenbreier. Impact of iron losses and parameter errors on back-EMF based sensorless control of PMSM. Dallas, TX, USA, Oct./Nov. 2014. Annual Conf. of the IEEE Industrial Electronics Society.

[Spe14] A. Specht. Ermittlung der Rotortemperatur einer Synchronmaschine mit eingebetteten Permanentmagneten für einen automobilen Traktionsantrieb mittels Beobachter basierend auf elektrischen Größen. PhD thesis, Faculty of Computer Science, Electrical Engineering and Mathematics, Paderborn University, 2014.

[SR98] P. O. M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. IEEE Trans. on Automatic Control, 43(8):1163–1169, Aug. 1998.

[Sto14] P. J. Stolze. Advanced finite-set model predictive control for power electronics and electrical drives. PhD thesis, Chair of Electrical Drive Systems and Power Electronics, Technical University of Munich, 2014.

[TFG+17] L. Tarisciotti, A. Formentini, A. Gaeta, M. Degano, P. Zanchetta, R. Rabbeni, and M. Pucci. Model predictive control for shunt active filters with fixed switching frequency. IEEE Trans. on Industry Applications, 53(1):296–304, Jan./Feb. 2017.

[TFOZ20] M. Tang, A. Formentini, S. A. Odhano, and P. Zanchetta. Torque ripple reduction of PMSMs using a novel angle-based repetitive observer. IEEE Trans. on Industrial Electronics, 67(4):2689–2699, Apr. 2020.

[TFZ19] M. Tang, A. Formentini, and P. Zanchetta. Repetitive observer design for torque ripple reduction in PMSM drives. Genova, Italy, Sep. 2019. European Conf. on Power Electronics and Applications (EPE'19 ECCE Europe).

[TGFB19] F. Toso, P. Gherardo, A. Favato, and S. Bolognani. On-line continuous control set MPC for PMSM drives current loops at high sampling rate using qpOASES. Baltimore, MD, USA, Sep./Oct. 2019. IEEE Energy Conversion Congress and Exposition.

[TGFZ16] M. Tang, A. Gaeta, A. Formentini, and P. Zanchetta. A variable frequency angle-based repetitive control for torque ripple reduction in PMSMs. Glasgow, UK, Apr. 2016. IET Int. Conf. on Power Electronics, Machines and Drives.

[TJB03] P. Tondel, T. A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. Automatica, 2003.

[TN86] I. Takahashi and T. Noguchi. A new quick-response and high-efficiency control strategy of an induction motor. IEEE Trans. on Industry Applications, 22(5):820–827, Sep. 1986.

[TZW+15] L. Tarisciotti, P. Zanchetta, A. Watson, J. C. Clare, M. Degano, and S. Bifaretti. Modulated model predictive control for a three-phase active rectifier. IEEE Trans. on Industry Applications, 51(2):1610–1620, Apr. 2015.

[UH09] S. J. Underwood and I. Husain. Online parameter estimation and adaptive control of permanent-magnet synchronous machines. IEEE Trans. on Industrial Electronics, 57(7):2435–2443, Nov. 2009.

[Vas98] P. Vas. Sensorless vector and direct torque control. Oxford University Press, 1998.

[Vil07] S. Villwock. Identifikationsmethoden für die automatisierte Inbetriebnahme und Zustandsüberwachung elektrischer Antriebe. PhD thesis, Fachbereich Elektrotechnik und Informatik, University of Siegen, 2007.

[VPdD16] A. Veltman, D. W. J. Pulle, and R. W. de Doncker. Fundamentals of electrical drives. 2. Edition, Springer Nature, 2016.

[vvD+03] B. Štumberger, G. Štumberger, D. Dolinar, A. Hamler, and M. Trlep. Evaluation of saturation and cross-magnetization effects in interior permanent-magnet synchronous motor. IEEE Trans. on Industry Applications, 39(5):1264–1271, Sep. 2003.

[WAB16] O. Wallscheid, U. Ammann, and J. Böcker. Real-time capable model predictive control of permanent magnet synchronous motors using particle swarm optimisation. Nuremberg, Germany, May 2016. Int. Exhibition and Conf. for Power Electronics.

[Wan09] L. Wang. Model predictive control system design and implementation using MATLAB®. Advances in Industrial Control, Springer Berlin Heidelberg, 2009.

[WB05] H. Wöhl-Bruhn. Vergleich von Rotoren mit Oberflächenmagneten und eingebetteten Magneten. Jahresbericht 2005 von Wöhl-Bruhn, 2005.

[WDK17a] S. Wendel, A. Dietz, and R. Kennel. Area-efficient FPGA implementation of finite control set model predictive current control. Puerto Varas, Chile, Dec. 2017. IEEE Southern Power Electronics Conf.

[WDK17b] S. Wendel, A. Dietz, and R. Kennel. FPGA based finite-set model predictive current control for small PMSM drives with efficient resource streaming. Pilsen, Czech Republic, Sep. 2017. IEEE Int. Symposium on Predictive Control of Electrical Drives and Power Electronics.

[WDK17c] S. Wendel, A. Dietz, and R. Kennel. Model predictive position control for permanent magnet synchronous linear motors. Saarbruecken, Germany, Sep. 2017. GMM/ETG-Symposium on Innovative Small Drives and Micro-Motor Systems.

[WGD09] Y. Wang, F. Gao, and F. J. Doyle. Survey on iterative learning control, repetitive control, and run-to-run control. Journal of Process Control, 19(10):1589–1600, Sep. 2009.

[WGL+19] S. Wendel, A. Geiger, E. Liegmann, D. Arancibia, E. Durán, T. Kreppel, F. Rojas, F. Popp-Nowak, M. Diaz, A. Dietz, R. Kennel, and B. Wagner. UltraZohm - a powerful real-time computation platform for MPC and multi-level inverters. Quanzhou, China, May 2019. IEEE Int. Symposium on Predictive Control of Electrical Drives and Power Electronics.

[WHKD20] S. Wendel, B. Haucke-Korber, and A. Dietz. Cascaded continuous and finite model predictive control for mechatronic systems. `https://register.epo.org/application?number=EP18190294`, Patent EP3614218, Feb. 2020.

[WHKDK18] S. Wendel, B. Haucke-Korber, A. Dietz, and R. Kennel. Cascaded continuous and finite model predictive speed control for electrical drives. Riga, Latvia, Sep. 2018. European Conf. on Power Electronics and Applications (EPE'18 ECCE Europe).

[WHKDK19] S. Wendel, B. Haucke-Korber, A. Dietz, and R. Kennel. Experimental evaluation of cascaded continuous and finite set model predictive speed control for electrical drives. Genova, Italy, Sep. 2019. European Conf. on Power Electronics and Applications (EPE'19 ECCE Europe).

[WHPB14] O. Wallscheid, T. Huber, W. Peters, and J. Böcker. Real-time capable methods to determine the magnet temperature of permanent magnet synchronous motors - A review. Dallas, TX, USA, Oct./Nov. 2014. Annual Conf. of the IEEE Industrial Electronics Society.

[WKDK19] S. Wendel, P. Karamanakos, A. Dietz, and R. Kennel. Operating point dependent variable switching point predictive current control for PMSM drives. Quanzhou, China, May 2019. IEEE Int. Symposium on Predictive Control of Electrical Drives and Power Electronics.

[WKDK20] S. Wendel, P. Karamanakos, A. Dietz, and R. Kennel. Flux linkage-based model predictive current control for nonlinear PMSM drives. Singapore, Singapore, Oct. 2020. Annual Conf. of the IEEE Industrial Electronics Society.

[WKG+21] S. Wendel, P. Karamanakos, P. Gebhardt, A. Dietz, and R. Kennel. Flux linkage-based direct model predictive current control for synchronous machines. IEEE Trans. on Power Electronics, 36(12):14237–14256, May 2021.

[WLH+18] S. Wendel, P. Löhdefink, M. Hoerner, A. Dietz, and R. Kennel. Dynamic model predictive position control for linear actuators in automotive applications. Monte-Carlo, Monaco, Apr. 2018. Int. Conf. on Ecological Vehicles and Renewable Energies.

[WNB19] O. Wallscheid, E. F. B. Ngoumtsa, and J. Böcker. Hierarchical model predictive speed and current control of an induction machine drive with moving-horizon load torque estimator. San Diego, CA, USA, May 2019. IEEE Int. Electric Machines & Drives Conf.

[WP84] D. F. Watson and G. M. Philip. Triangle based interpolation. Mathematical Geology, 16(8):779–795, Nov. 1984.

[WSJ+13] G. Weidenholzer, S. Silber, G. Jungmayr, G. Bramerdorfer, H. Grabner, and W. Amrhein. A flux-based PMSM motor model using RBF interpolation for time-stepping simulations. Chicago, IL, USA, May 2013. Int. Electric Machines & Drives Conf.

[WSL+] S. Wendel, S. Schindler, E. Liegmann, M. Hoerner, A. Geiger, and UltraZohm community. UltraZohm real-time calculation platform. `https://docs.ultrazohm.com`. Access: May 2021.

[YZZ20] X. Yuan, S. Zhang, and C. Zhang. Nonparametric predictive current control for pmsm. IEEE Trans. on Power Electronics, 35(9):9332–9341, Sep. 2020.

[ZBY18] Y. Zhang, Y. Bai, and H. Yang. A universal multiple-vector-based model predictive control of induction motor drives. IEEE Trans. on Power Electronics, 33(8):6957–6969, Aug. 2018.

[ZDMB20] C. Zheng, T. Dragičević, B. Majmunović, and F. Blaabjerg. Constrained modulated-model predictive control of an LC-filtered voltage source converter. IEEE Trans. on Power Electronics, 35(2):1967–1977, Feb. 2020.

[Zel18] P. Zeller. Handbuch Fahrzeugakustik. 3. Auflage, Springer Vieweg, 2018.

[Zou17] H. Zoubek. Drehgeberlose Identifikation schwingungsfähiger Mehrmassensysteme und Diagnose von Lagerfehlern im Antriebsstrang durch Frequenzgangmessung. PhD thesis, Fachbereich Elektrotechnik und Informatik, University of Siegen, 2017.

[ZPY16] Y. Zhang, Y. Peng, and H. Yang. Performance improvement of two-vectors-based model predictive control of PWM rectifier. IEEE Trans. on Power Electronics, 31(8):6016–6030, Aug. 2016.

[ZXL+14] 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 Trans. on Industry Applications, 53(4):3683–3695, Jul./Aug. 2014.

[ZXLZ14] Y. Zhang, W. Xie, Z. Li, and Y. Zhang. Low-complexity model predictive power control: Double-vector-based approach. IEEE Trans. on Industrial Electronics, 61(11):5871–5880, Nov. 2014.

[ZZZ19] X. Zhang, L. Zhang, and Y. Zhang. Model predictive current control for PMSM drives with parameter robustness improvement. IEEE Trans. on Power Electronics, 34(2):1645–1657, Feb. 2019.