Data-based approximation of MPC for electrical machines

DIPLOMA THESIS

Conducted in partial fulfillment of the requirements for the degree of a

Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Dipl.-Ing. Georg Janisch BSc
Associate Prof. Dr.-Ing. Wolfgang Kemmetmüller

submitted at the

TU Wien
Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

by
Felix Heidegger
Matriculation number 01527191

Wien, May 2024

Complex Dynamical Systems Group
A-1040 Wien, Gußhausstr. 27–29, Internet: https://www.acin.tuwien.ac.at

Preamble

First and foremost, I would like to express my deep gratitude to my supervisor Associate Prof. Dr.-Ing. Wolfgang Kemmetmüller for the opportunity of conducting the Master Thesis under his supervision and the continuous feedback throughout the conduction of the thesis. I would also like to thank Dipl.-Ing. Georg Janisch BSc for his direct supervision, the ideas and inspirations that he gave me and the availability to my questions at all times. The completion of this Master Thesis would not have been possible without the continuous input, the big support and all the theoretical knowledge that he provided me. Furthermore I would like to thank MAGNA Powertrain GmbH & Co KG for the opportunity to work on this interesting and future oriented topic in a cooperation. My work colleagues, especially Martin Böck, Bence Plenar, Erald Pjetri and Elias Pritzi, helped me out with valuable insights about the model of the electrical machines, the data driven approaches and moral support over the whole course of this thesis.

I am also very grateful for the encouragement and the help from my family over the whole course of my studies. Their belief in my abilities and their support made the studies for me possible at all and they helped me with their kind words in difficult times. I would also like to thank my girlfriend for her constant support, her motivating words and her patience with me over the whole course of the studies.

Finally, I would like to thank my university colleagues, which made the studying with the fruitful discussions and their inputs much easier and more fun. Here I would like to thank especially Georg Feiler, Elias Pritzi and Benjamin Gallauner, who are not only study colleagues but have also become good friends on which I can count at any time.

Wien, May 2024

Abstract

Recent trends in the automotive industry require more dynamic and more efficient performance of electrical machines to satisfy the customer desire of fast acceleration and a wide range. To deal with this requirement, these complex machines must be controlled in an optimal way. For this purpose, **M**odel **P**redictive **C**ontrol (MPC) is increasingly researched in the field of control for electrical motors. It is a well known optimal control method, which minimizes a cost functional over a predicted state trajectory to find the optimal values (with respect to the cost functional) for the control inputs. Since the optimization problem must be solved in each timestep, solving the problem must be fast enough to be finished within the sampling time of the control concept. As the sampling times for current controllers in electrical machines are very low, this asks for high computational power. However, the motor control unit on which the control concept is deployed, offers limited performance and makes the method rather impractical.

For this purpose this work introduces a novel approach of approximating an MPC for the control of electrical machines. It focuses on the control for the two most widespread electrical machine types in the automotive industry, namely the permanent magnet synchronous motor and the induction machine. The data generation step is presented and the computational effort to perform the necessary simulations is estimated. Selecting operating points relevant for the application is indispensable for the achieved control performance. Furthermore, the proposed data-driven approach is evaluated on different MPC formulations and tested thoroughly in simulations. Moreover, the approximation accuracy in the case of parameter deviations and highly dynamical setpoint changes is investigated. Finally, the proposed algorithm is deployed on a dSPACE Microlab Box real-time platform and its runtime is measured and compared to the MPC showing that the proposed method overcomes the biggest disadvantage of MPC control, the runtime.

Kurzzusammenfassung

Jüngste Trends in der Automobilindustrie erfordern eine dynamischere und effizientere Leistung elektrischer Maschinen, um den Kundenwünschen nach hoher Beschleunigung und großer Reichweite gerecht zu werden. Um diese Anforderungen zu erfüllen, müssen diese komplexen Maschinen optimal geregelt werden. Aufgrund dessen wird auf dem Gebiet der Regelung für Elektromotoren zunehmend die modellprädiktive Regelung erforscht. Es handelt sich hierbei um ein bekanntes optimales Regelungskonzept, welches ein Kostenfunktional über eine prädizierte Zustandstrajektorie minimiert, um auf diese Weise die optimalen Werte (in Bezug auf das Kostenfunktional) für den Reglereingang zu finden. Da das Optimierungsproblem in jedem Zeitschritt gelöst werden muss, muss die Lösung des Problems schnell genug sein, um innerhalb der Abtastzeit des Reglers abgeschlossen zu werden. Da die Abtastzeiten für die Stromregelung von Elektromotoren sehr gering sind, erfordert dies eine hohe Rechenleistung. Allerdings bietet das Motorsteuergerät, auf welchem die Regelung eingesetzt wird, nicht die erforderliche Rechenleistung und macht das Konzept für den Einsatz in elektrischen Maschinen unpraktikabel.

Aus diesem Grund wird in dieser Arbeit ein neuartiger Ansatz zur Approximation von modellprädiktiven Reglern vorgestellt. Der Fokus liegt dabei auf der Regelung für die beiden in der Automobilindustrie am weitesten verbreiteten Elektromotoren, nämlich dem Permanentmagnet-Synchronmotor und dem Asynchronmotor. Der Ansatz für die Datengenerierung wird vorgestellt und der Rechenaufwand für die Durchführung der notwendigen Simulationen wird abgeschätzt. Die Auswahl der für die Anwendung relevanten Arbeitspunkte ist für die erzielte Regelgüte essenziell. Darüber hinaus wird der vorgeschlagene datengetriebene Ansatz für verschiedene MPC Formulierungen evaluiert und in Simulationen eingehend getestet. Außerdem wird die Approximationsgenauigkeit im Falle von Parameterabweichungen und hochdynamischen Arbeitspunktwechseln untersucht. Schließlich wird der vorgeschlagene Algorithmus auf einer SPACE Microlab Box Echtzeitplattform eingesetzt, seine Laufzeit gemessen und mit der modellprädiktiven Regelung verglichen. Dieser Vergleich zeigt, dass die vorgeschlagene Methode den größten Nachteil der modellprädiktiven Regelung, nämlich die Laufzeit, beseitigt.

Contents

List of Figures

List of Tables

1 Introduction

In times of escalating climate change concerns, the automotive sector plays a pivotal role in the reduction of its CO_2 emissions. According to [1], road transportation creates 72.9% of the transport emissions in the European Union. One of the main contributors to these high emissions, is the classical internal combustion engine. Combustion engines are run with petrol or diesel, which are fossil fuels that produce CO_2 during their operation. With the introduction of new climate goals stemming from the Paris Agreement, which state that the long-term temperature rise of the global surface temperature should be kept below 2°C compared to pre-industrial levels, the emissions of internal combustion engines are no longer acceptable.

For this purpose, electrical machines in the automotive sector have been investigated over the last decades. As the name suggests, electric vehicles are run with electrical energy, which can be produced by renewable energies. Furthermore, these machines have higher efficiencies than combustion engines, if utilized in the right way. Therefore, these machines show a high potential of decreasing the emissions in the transport sector. There are many different types of electrical machines used nowadays in the automotive industry, where two of the most widespread machine types are the **P**ermanent **M**agnet **S**ynchronos **M**achine (PMSM) and the **I**nduction **M**achine (IM) [2]. In the PMSM the rotor field is generated by permanent magnets, which are part of the rotor. For the induction machine on the other hand, the rotor field is generated via the stator field. Even if these machines differ in their working principle and their constructional point of view, they have in common, that they represent highly nonlinear systems. This is due to the saturation behavior of the iron for the PMSM [3] and the saturation of the mutual inductance for the IM [4]. Furthermore, temperature is also a main influencing factor for these machines. In order to utilize these highly efficient systems and achieve high dynamics, which are important for the use in the automotive sector, well designed control concepts must be developed which are capable of achieving the desired goals.

1.1 Control structures for electrical machines

The main task of the control for electrical machines in the automotive industry is to achieve fast torque dynamics and a high torque accuracy, while still ensuring energy efficiency. Furthermore, safe operation, especially at the voltage and current limit, are of uttermost importance, since the automotive sector is a safety critical field [5].

The two most common used state of the art control strategies are **F**ield **O**riented **C**ontrol (FOC) [6, 7] and **D**irect **T**orque **C**ontrol (DTC) [8]. For the field oriented control, cascaded control loops are used, which consist of a setpoint calculation and a current controller. The setpoint calculation computes the optimal current setpoints according to a cost functional to achieve the desired torque values. The current controller takes these setpoints as input

and outputs the voltages for the inverter. As a result of the high sampling frequencies in automotive applications, the current controller must be evaluated quickly. For this purpose, PI-controllers with a suitable anti-windup method are often used for this task [9]. For the direct torque control, the torque is estimated as a cross product of the estimated stator flux linkage vector and the measured current vector. In this approach, the inverter switches directly control the flux and torque of the motor based on a pre-defined switching table. This method achieves a similar dynamic performance as the FOC, albeit with an increased torque ripple [10]. However, the determination of controller parameters for robust performance in the presence of magnetic saturation behavior, parameter variations and system limits, while still achieving high dynamics, is a challenging task. Furthermore, they only take the current timestep into account and are therefore limited in the performance of dynamical setpoint changes, since they are not able to plan suitable trajectories beforehand. Finally, they struggle with system boundaries, since they are not able to consider the system limits in a systematic way before hitting them. [11]
To counteract these disadvantages, **M**odel **P**redictive **C**ontrol (MPC) for electrical machines is a big research field [12]. MPC is an optimal control strategy, that optimizes the input of the system over a prediction horizon into the future. To predict the state trajectory into the future, the (non)linear model of the system is used. By doing so, system boundaries can be considered in the optimization and thus MPCs do not violate the system limits. However, the optimization problem has to be solved in each time step over the whole prediction horizon. This is computationally expensive and due to the fast sampling rates of control loops for electrical machines, often not real-time capable. To tackle this problem, different MPC concepts are investigated in this work and approximated by data driven approaches.

1.2 Data driven approaches for MPC approximation

To approximate an MPC using a data driven approach, the input output behavior of the system must be learned. In other words, this can be interpreted as a stationary function approximation of the MPC, since the MPC is only dependent on current inputs of the system. Figure 1.1 shows a simplified schematic of this approximation concept. The MPC and the data driven model get the current states $\mathbf{x}$ and the desired states $\mathbf{x}^{\mathrm{d}}$ as input and both predict their outputs $\tilde{\mathbf{u}}$ and $\mathbf{u}^{*}$. These outputs are compared with each other and by some learning action, the data driven model is updated. This is repeated until the MPC and the data driven model output the same values or some maximum number of learning updates has been performed. This approach can be considered as a grey box approach, since the underlying control concept is well known. The approximation itself, however, is a black box model. This is important to mention, because in contrast to reinforcement learning control [13], where the whole control strategy is learned from scratch, the underlying controllers in this work were tested thoroughly and their well known behavior should only be approximated. The advantage of reinforcement learning control is, that no real expert knowledge is needed for the design of the controller, since the whole control action is learned during simulations and testbench measurements. However, this takes a lot of time and computational effort, since all the knowledge about the control

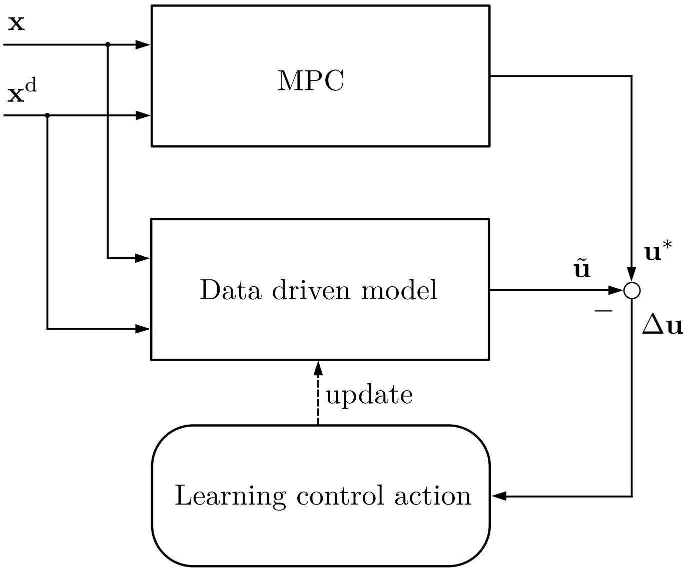


Figure 1.1: Concept of data driven approximation for MPCs.

must be built up by the controller. Furthermore, this is a purely black-box model, since the user does not know anything about the control design in this case, as it is purely data based.

For the approximation of the controller structure, neural networks are widely used. [14] shows the approximation approach on the example of a chemical reactor using Sobolev training. Doing so, the parametric sensitivities of the nonlinear optimization problem are added as extra information for the training of the neural net. For small datasets, this approach delivers better results than the classical neural network training [15]. For a very high dimensional and fine sampled grid, however, this difference gets smaller. Since in this work a lot of simulations have to be performed to cover the whole operating range, the influence on the results might be negligible and therefore this work focuses on the usage of classical fully connected feedforward neural nets.

A big problem arises for the approximation for the performance around the boundaries of the system. Since the approach only approximates the true behavior of the MPC, it is not guaranteed that the boundary conditions can be fulfilled for the approximation. [16] shows an approach for the analytical consideration of boundary conditions with constrained neural networks. However, the paper only shows the approach for explicit MPCs, where the solution of the MPC is computed offline and stored for a certain number of polytopic regions. During operation, it is only necessary to evaluate online in which polytopic region the system state is in. Since in this work only implicit MPC are investigated, which solve the whole optimization problem in each timestep separately, this approach is not further followed. To check if the boundary conditions are met, different dynamic simulations at the voltage and the current boundary are performed and evaluated.

One big caveat in the usage of MPCs is the proof of stability and therefore it is even more difficult to prove the stability of the approximation. An approach to show the stability of an approximated MPC is given by [17]. However, this paper only shows the performance on a rather simplified system. To prove the stability of the developed approach in this work, the MPC is tested for many different use cases including parameter disturbances and the results are evaluated.

1.3 Aim of this work

While the listed publications show different data driven approaches in the control field, they rarely discuss the usage of the proposed concept on a real-world example. Furthermore, many papers rather deal with the usage of reinforcement learning [13] as data driven approach instead of the grey box approach used in this paper. Moreover, no paper was found that deals with this problem in the field of the control of electrical machines. Therefore, the aim of this work is to propose different model predictive control formulations for the current control of PMSMs and the control of IMs and explain the steps that have to be performed to approximate such MPCs. For the PMSM the main goal is to investigate different formulations with different input spaces of the data driven approach to determine the generality of the developed concept. For the IM, the primary objective is to show the performance of the developed concept on a real world MPC, that has been tested thoroughly in [18]. The developed control concepts are compared with the original MPC in closed-loop simulations for steady-state and dynamic scenarios at different driving speeds and their approximation accuracy is investigated. Furthermore, measurements on a target hardware are performed to estimate the runtime of the developed concepts and therefore check if the approach is real-time capable.

1.4 Structure of this work

The remainder of this work is structured as follows: In Chapter 2, mathematical models for the field oriented control of the PMSM and the IM are derived. The MPCs are based on these models and their approximations therefore as well.

Chapter 3 introduces the control concept for the PMSM together with the different MPC formulations, that are investigated in detail in this work and their approximations. The main goal herein is to explain the formulations and analyze their advantages and disadvantages. In Section 3.4 the step of data generation is covered with estimates on the computation time to perform the simulations. The data driven approach is introduced with its necessary inputs and outputs and the performance of the prediction is shown for open loop simulations. Finally, closed loop simulations are performed in Section 3.6 for the developed concepts and the results are compared in detail.

Chapter 4 is structured similarly to Chapter 3, but deals with the IM. The control loop structure is introduced first and the main parts that are approximated, are explained in Section 4.2 and Section 4.3. The data generation is covered in detail in Section 4.5 and an estimate on the computation time is given. 4.6 deals with the data driven approach and the need for an ablation study. Finally, simulation results are shown in Section 4.7 and discussed. Furthermore, a runtime evaluation on a target hardware is performed and the results are displayed.

The work is concluded by summarizing the work and giving outlooks on further improvements and possible extensions in Chapter 5.

2 Mathematical Model

An accurate mathematical model is the prerequisite for model-based control. In this chapter the mathematical models for the PMSM and the IM are derived.

2.1 Permanent magnet synchronous motor

To perform the model-based, field oriented control of the PMSM the description in the dq-coordinate frame is chosen. This description is derived in more detail in [19].
The stator voltage equations of a PMSM in the dq-coordinate frame are given by

$$u_\mathrm{d} = R_\mathrm{s} i_\mathrm{d} + \frac{\mathrm{d}\Psi_\mathrm{d}}{\mathrm{d}t} - \omega \Psi_\mathrm{q} \tag{2.1a}$$

$$u_\mathrm{q} = R_\mathrm{s} i_\mathrm{q} + \frac{\mathrm{d}\Psi_\mathrm{q}}{\mathrm{d}t} + \omega \Psi_\mathrm{d} \tag{2.1b}$$

with the direct- and quadrature-voltages u_d and u_q, the direct- and quadrature currents i_d and i_q, the direct- and quadrature fluxes Ψ_d and Ψ_q and the stator resistance R_s. $\omega = Z_\mathrm{p}\omega_\mathrm{m}$ is the electrical rotor speed and it is computed via the measurable mechanical angular speed ω_m and the number of pole pairs of the machine Z_p. The relationship between the currents and the fluxes in the dq-coordinate frame is defined as

$$\Psi_\mathrm{d} = L_\mathrm{d}(i_\mathrm{d}, i_\mathrm{q}, \vartheta) i_\mathrm{d} + \Psi_\mathrm{PM}(i_\mathrm{q}, \vartheta) \tag{2.2a}$$

$$\Psi_\mathrm{q} = L_\mathrm{q}(i_\mathrm{d}, i_\mathrm{q}, \vartheta) i_\mathrm{q} \ , \tag{2.2b}$$

with the permanent magnet flux Ψ_PM and the direct- and quadrature inductances L_d and L_q. This relationship is nonlinear, since the inductances and the permanent magnet flux can saturate with respect to the currents and the temperature ϑ. In this work, the influence of the temperature on the fluxes and inductances is neglected, since its influence is small compared to the nonlinear saturation behavior. To account for the nonlinear saturation relationship, **L**ook **U**p **T**ables (LUTs) for the inductances and the permanent magnet flux are used. These LUTs are derived from Finite Element simulations. In order to get an even better representation of the true behavior of the electrical machine, these LUTs are further calibrated on the testbench.
By inserting (2.2) in (2.1) the continuous state space formulation

$$\begin{aligned}\dot{\mathbf{x}}_\mathrm{SM} &= \mathbf{A}_\mathrm{SM}\mathbf{x}_\mathrm{SM} + \mathbf{B}_\mathrm{u,SM}\mathbf{u}_\mathrm{SM} + \mathbf{b}_{\mathrm{SM},\Psi}\Psi_\mathrm{SM,PM} = \\ &= \begin{bmatrix} -\frac{R_\mathrm{s}}{L_\mathrm{d}} & \omega \\ -\omega & -\frac{R_\mathrm{s}}{L_\mathrm{q}} \end{bmatrix} \begin{bmatrix} \Psi_\mathrm{d} \\ \Psi_\mathrm{q} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_\mathrm{d} \\ u_\mathrm{q} \end{bmatrix} + \begin{bmatrix} \frac{R_\mathrm{s}}{L_\mathrm{d}} \\ 0 \end{bmatrix} \Psi_\mathrm{PM}\end{aligned} \tag{2.3}$$

with the state space vector $\mathbf{x}_{\mathrm{SM}} = [\Psi_{\mathrm{d}}, \Psi_{\mathrm{q}}]^{\mathrm{T}}$ and the input vector $\mathbf{u}_{\mathrm{SM}} = [u_{\mathrm{d}}, u_{\mathrm{q}}]^{\mathrm{T}}$ is derived. Since in this work different machine models are introduced, the subscript $(*)_{\mathrm{SM}}$ stands for **S**ynchronous **M**achine and refers to the variables for the PMSM. The system (2.3) is nonlinear and thus difficult to discretize. By assuming constant inductances, a linear system results. Since the model error of this approximation is minor due to the high sampling frequency for the controller design, (2.3) is evaluated in each timestep at the current operating point $\mathbf{i}_{\mathrm{SM},m} = [i_{\mathrm{d},m}, i_{\mathrm{q},m}]^{\mathrm{T}}$ and the values for the inductances are kept constant until the next timestep. The resulting time dependent linear system is then discretized using the exact discretization method in each timestep, which yields

$$\mathbf{x}_{\mathrm{SM},m+1} = \mathbf{\Phi}_{\mathrm{SM},m}\mathbf{x}_{\mathrm{SM},m} + \mathbf{\Gamma}_{\mathrm{SM,u},m}\mathbf{u}_{\mathrm{SM},m} + \mathbf{\Gamma}_{\mathrm{SM},\Psi,m}\Psi_{\mathrm{SM,PM},m} \tag{2.4}$$

with

$$\mathbf{\Phi}_{\mathrm{SM},m} = \exp(\mathbf{A}_{\mathrm{SM},m}T_{\mathrm{s}}) \tag{2.5a}$$

$$\mathbf{\Gamma}_{\mathrm{SM,u},m} = \int_0^{T_{\mathrm{s}}} \exp(\mathbf{A}_{\mathrm{SM},m}t)\,\mathrm{d}t\mathbf{B}_{\mathrm{SM,u},m} \tag{2.5b}$$

$$\mathbf{\Gamma}_{\mathrm{SM},\Psi,m} = \int_0^{T_{\mathrm{s}}} \exp(\mathbf{A}_{\mathrm{SM},m}t)\,\mathrm{d}t\mathbf{b}_{\mathrm{SM},\Psi,m}\ , \tag{2.5c}$$

where T_{s} describes the sampling time of the system and $\mathbf{\Phi}_{\mathrm{SM},m}$, $\mathbf{\Gamma}_{\mathrm{SM,u},m}$ and $\mathbf{\Gamma}_{\mathrm{SM},\Psi,m}$ refer to the time discretized dynamic matrix, the time discretized input matrix for the voltage and the time discretized input matrix for the permanent magnet flux respectively. Furthermore, the torque of the PMSM can be computed as

$$\tau(\mathbf{i}_{\mathrm{SM}}) = \frac{3}{2}Z_{\mathrm{p}}(\Psi_{\mathrm{d}}i_{\mathrm{q}} - \Psi_{\mathrm{q}}i_{\mathrm{d}})\ . \tag{2.6}$$

Under the assumption, that the ohmic copper losses are the dominating losses in the machine, the stator losses are furthermore given as

$$P_{\mathrm{s}} = \frac{3}{2}R_{\mathrm{s}}\mathbf{i}_{\mathrm{SM}}^{\mathrm{T}}\mathbf{i}_{\mathrm{SM}}\ . \tag{2.7}$$

These stator losses are later needed in the setpoint computation for the PMSM.

2.2 Induction machine

For the model based, field-oriented control of the IM the equations are derived for the Γ-model, which is extended by stator flux saturation. This model is described in more detail in [4, 20, 21].

According to Figure 2.1 the differential equations of a squirrel cage induction machine in a stator-fixed reference frame (furthermore referenced to as $\alpha\beta$-coordinate frame) are given by

$$\mathbf{u}_{\mathrm{s}} = R_{\mathrm{s}}\mathbf{i}_{\mathrm{s}} + \frac{\mathrm{d}\mathbf{\Psi}_{\mathrm{s}}}{\mathrm{d}t} \tag{2.8a}$$

$$\mathbf{0} = R_{\mathrm{r}}\mathbf{i}_{\mathrm{r}} + \frac{\mathrm{d}\mathbf{\Psi}_{\mathrm{r}}}{\mathrm{d}t} + \omega\mathbf{J}\mathbf{\Psi}_{\mathrm{r}}\ , \tag{2.8b}$$

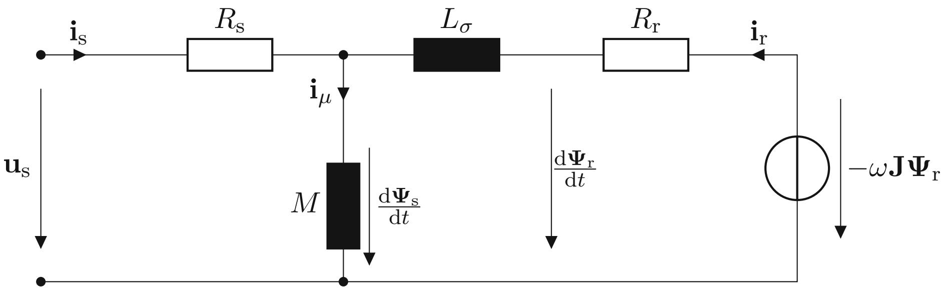


Figure 2.1: Equivalent circuit of the Γ - model. [21]

with the stator- and rotor-fluxes $\mathbf{\Psi}_\mathrm{s} = [\Psi_{\mathrm{s}\alpha}, \Psi_{\mathrm{s}\beta}]^\mathrm{T}$ and $\mathbf{\Psi}_\mathrm{r} = [\Psi_{\mathrm{r}\alpha}, \Psi_{\mathrm{r}\beta}]^\mathrm{T}$, the stator voltage $\mathbf{u}_\mathrm{s} = [u_{\mathrm{s}\alpha}, u_{\mathrm{s}\beta}]^\mathrm{T}$, the stator- and rotor-resistances R_s and R_r and the stator- and rotor-currents $\mathbf{i}_\mathrm{s} = [i_{\mathrm{s}\alpha}, i_{\mathrm{s}\beta}]^\mathrm{T}$ and $\mathbf{i}_\mathrm{r} = [i_{\mathrm{r}\alpha}, i_{\mathrm{r}\beta}]^\mathrm{T}$. Herein, the rotation matrix $\mathbf{J}$ is defined as

$$\mathbf{J} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} . \tag{2.9}$$

The flux linkage equations

$$\mathbf{\Psi}_\mathrm{s} = M(\|\mathbf{\Psi}_\mathrm{s}\|)\mathbf{i}_\mu \tag{2.10a}$$
$$\mathbf{\Psi}_\mathrm{r} = L_\sigma \mathbf{i}_\mathrm{r} + \mathbf{\Psi}_\mathrm{s} \tag{2.10b}$$

relate the currents $\mathbf{i}_\mu = \mathbf{i}_\mathrm{s} + \mathbf{i}_\mathrm{r}$ and $\mathbf{i}_\mathrm{r}$, the mutual inductance $M(\|\mathbf{\Psi}_\mathrm{s}\|)$ and the leakage inductance L_σ with the fluxes $\mathbf{\Psi}_\mathrm{s}$ and $\mathbf{\Psi}_\mathrm{r}$. This relationship is nonlinear, since the mutual inductance $M(\|\mathbf{\Psi}_\mathrm{s}\|)$ saturates with respect to the stator flux $\mathbf{\Psi}_\mathrm{s}$ according to [20]

$$M(\|\mathbf{\Psi}_\mathrm{s}\|) = \frac{L_\mu}{1 + (S\|\mathbf{\Psi}_\mathrm{s}\|)^\beta} , \tag{2.11}$$

with the constant parameters L_μ, β and S.

The continuous state space formulation for the equations in the stationary $\alpha\beta$-coordinate frame results from inserting (2.10) into (2.8)

$$\dot{\mathbf{x}}_{\alpha\beta} = \mathbf{A}_{\alpha\beta}(\|\mathbf{\Psi}_\mathrm{s}\|)\mathbf{x}_{\alpha\beta} + \mathbf{B}_{\alpha\beta}\mathbf{u}_\mathrm{s} \tag{2.12}$$

with

$$\mathbf{A}_{\alpha\beta} = \begin{bmatrix} -R_\mathrm{s}\frac{M(\|\mathbf{\Psi}_\mathrm{s}\|) + L_\sigma}{M(\|\mathbf{\Psi}_\mathrm{s}\|)L_\sigma} & 0 & \frac{R_\mathrm{s}}{L_\sigma} & 0 \\ 0 & -R_\mathrm{s}\frac{M(\|\mathbf{\Psi}_\mathrm{s}\|) + L_\sigma}{M(\|\mathbf{\Psi}_\mathrm{s}\|)L_\sigma} & 0 & \frac{R_\mathrm{s}}{L_\sigma} \\ \frac{R_\mathrm{r}}{L_\sigma} & 0 & -\frac{R_\mathrm{r}}{L_\sigma} & -\omega \\ 0 & \frac{R_\mathrm{r}}{L_\sigma} & \omega & -\frac{R_\mathrm{r}}{L_\sigma} \end{bmatrix} , \mathbf{B}_{\alpha\beta} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} , \tag{2.13}$$

and the state space vector $\mathbf{x}_{\alpha\beta} = [\Psi_{\mathrm{s}\alpha}, \Psi_{\mathrm{s}\beta}, \Psi_{\mathrm{r}\alpha}, \Psi_{\mathrm{r}\beta}]^{\mathrm{T}}$. According to [19] the torque for the induction machine is derived as

$$\tau = \frac{3}{2}\frac{Z_{\mathrm{p}}}{L_\sigma}\boldsymbol{\Psi}_{\mathrm{r}}^{\mathrm{T}}\mathbf{J}\boldsymbol{\Psi}_{\mathrm{s}} \ . \tag{2.14}$$

In order to obtain the differential equations of the induction machine in an arbitrary coordinate system, the inverse Park-transformation $\mathbf{x}_{\alpha\beta} = \mathbf{R}(\Theta_{\mathrm{l}})^{-1}\mathbf{x}_{\mathrm{l}}$ with the rotation matrix

$$\mathbf{R}(\Theta_{\mathrm{l}}) = \begin{bmatrix} \cos(\Theta_{\mathrm{l}}) & \sin(\Theta_{\mathrm{l}}) \\ -\sin(\Theta_{\mathrm{l}}) & \cos(\Theta_{\mathrm{l}}) \end{bmatrix} , \tag{2.15}$$

is used [22]. Here, Θ_{l} describes an arbitrary angle between the stator-fixed coordinate frame and an arbitrary coordinate system l. The replacement of the stationary variables in (2.8) with the inverse Park transformed quantities from the rotating reference l yields

$$\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\mathbf{u}_{\mathrm{s,l}} = R_{\mathrm{s}}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\mathbf{i}_{\mathrm{s,l}} + \frac{\mathrm{d}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\boldsymbol{\Psi}_{\mathrm{s,l}}}{\mathrm{d}t} \tag{2.16a}$$

$$\mathbf{0} = R_{\mathrm{r}}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\mathbf{i}_{\mathrm{r,l}} + \frac{\mathrm{d}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\boldsymbol{\Psi}_{\mathrm{r,l}}}{\mathrm{d}t} + \omega\mathbf{J}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\boldsymbol{\Psi}_{\mathrm{r,l}} \ . \tag{2.16b}$$

The time derivative

$$\frac{\mathrm{d}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\boldsymbol{\Psi}_{\mathrm{s/r,l}}}{\mathrm{d}t} = \frac{\mathrm{d}\Theta_{\mathrm{l}}}{\mathrm{d}t}\frac{\mathrm{d}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}}{\mathrm{d}\Theta_{\mathrm{l}}}\boldsymbol{\Psi}_{\mathrm{s/r,l}} + \mathbf{R}(\Theta_{\mathrm{l}})^{-1}\frac{\mathrm{d}\boldsymbol{\Psi}_{\mathrm{s/r,l}}}{\mathrm{d}t} \tag{2.17a}$$

$$= -\frac{\mathrm{d}\Theta_{\mathrm{l}}}{\mathrm{d}t}\mathbf{R}(\Theta_{\mathrm{l}})^{-1}\mathbf{J}\boldsymbol{\Psi}_{\mathrm{s/r,l}} + \mathbf{R}(\Theta_{\mathrm{l}})^{-1}\frac{\mathrm{d}\boldsymbol{\Psi}_{\mathrm{s/r,l}}}{\mathrm{d}t} \tag{2.17b}$$

and a multiplication by the rotation matrix (2.9) from the left side yields

$$\mathbf{u}_{\mathrm{s,l}} = R_{\mathrm{s}}\mathbf{i}_{\mathrm{s,l}} + \frac{\mathrm{d}\boldsymbol{\Psi}_{\mathrm{s,l}}}{\mathrm{d}t} - \omega_{\mathrm{l}}\mathbf{J}\boldsymbol{\Psi}_{\mathrm{s,l}} \tag{2.18a}$$

$$\mathbf{0} = R_{\mathrm{r}}\mathbf{i}_{\mathrm{r,l}} + \frac{\mathrm{d}\boldsymbol{\Psi}_{\mathrm{r,l}}}{\mathrm{d}t} - (\omega_{\mathrm{l}} - \omega)\mathbf{J}\boldsymbol{\Psi}_{\mathrm{r,l}} \tag{2.18b}$$

with $\omega_{\mathrm{l}} = \frac{\mathrm{d}\Theta_{\mathrm{l}}}{\mathrm{d}t}$. Eliminating the currents using (2.10) gives

$$\mathbf{u}_{\mathrm{s,l}} = R_{\mathrm{s}}\frac{L_\sigma + M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)}{L_\sigma M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)}\boldsymbol{\Psi}_{\mathrm{s,l}} - \frac{R_{\mathrm{s}}}{L_\sigma}\boldsymbol{\Psi}_{\mathrm{r,l}} + \frac{\mathrm{d}\boldsymbol{\Psi}_{\mathrm{s,l}}}{\mathrm{d}t} - \omega_{\mathrm{l}}\mathbf{J}\boldsymbol{\Psi}_{\mathrm{s,l}} \tag{2.19a}$$

$$\mathbf{0} = \frac{R_{\mathrm{r}}}{L_\sigma}(\boldsymbol{\Psi}_{\mathrm{r,l}} - \boldsymbol{\Psi}_{\mathrm{s,l}}) + \frac{\mathrm{d}\boldsymbol{\Psi}_{\mathrm{r,l}}}{\mathrm{d}t} - (\omega_{\mathrm{l}} - \omega)\mathbf{J}\boldsymbol{\Psi}_{\mathrm{r,l}} \ . \tag{2.19b}$$

By rearranging (2.19) and evaluating it in the rotor fixed dq-coordinate system, the state space formulation

$$\dot{\mathbf{x}}_{\mathrm{IM}} = \mathbf{A}_{\mathrm{IM}}(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)\mathbf{x}_{\mathrm{IM}} + \mathbf{B}_{\mathrm{IM}}\mathbf{u}_{\mathrm{IM}} \tag{2.20}$$

with

$$\mathbf{A}_{\mathrm{IM}} = \begin{bmatrix} -R_{\mathrm{s}}\dfrac{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)+L_\sigma}{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)L_\sigma} & \omega_{\mathrm{l}} & \dfrac{R_{\mathrm{s}}}{L_\sigma} & 0 \\ -\omega_{\mathrm{l}} & -R_{\mathrm{s}}\dfrac{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)+L_\sigma}{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)L_\sigma} & 0 & \dfrac{R_{\mathrm{s}}}{L_\sigma} \\ \dfrac{R_{\mathrm{r}}}{L_\sigma} & 0 & -\dfrac{R_{\mathrm{r}}}{L_\sigma} & -\omega+\omega_{\mathrm{l}} \\ 0 & \dfrac{R_{\mathrm{r}}}{L_\sigma} & \omega-\omega_{\mathrm{l}} & -\dfrac{R_{\mathrm{r}}}{L_\sigma} \end{bmatrix}, \quad \mathbf{B}_{\mathrm{IM}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.21}$$

the state space vector $\mathbf{x}_{\mathrm{IM}} = [\Psi_{\mathrm{sd}}, \Psi_{\mathrm{sq}}, \Psi_{\mathrm{rd}}, \Psi_{\mathrm{rq}}]^{\mathrm{T}}$ and the input vector $\mathbf{u}_{\mathrm{IM}} = [u_{\mathrm{sd}}, u_{\mathrm{sq}}]^{\mathrm{T}}$. To differentiate between the state space system for the induction machine and the PMSM, the subscript $(*)_{\mathrm{IM}}$ stands for **I**nduction **M**achine and refers to the variables for the IM. These variables are derived in the dq-coordinate frame. To keep the nomenclature as short and simple as possible, the extra subscript $(*)_{\mathrm{dq}}$ are herein omitted.
In this formulation, the rotational speed ω_{l} is completely arbitrary. By using a rotational speed of

$$\omega_{\mathrm{l}} = \omega_{\mathrm{dq}} = \omega + \frac{R_{\mathrm{r}}}{L_\sigma}\frac{\Psi_{\mathrm{sq}}}{\Psi_{\mathrm{rd}}} \tag{2.22}$$

and inserting it into (2.19), the differential equation for the rotor quadrature axis becomes

$$\frac{\mathrm{d}\Psi_{\mathrm{rq}}}{\mathrm{d}t} = -\frac{R_{\mathrm{r}}}{L_\sigma}\Psi_{\mathrm{rq}} \ . \tag{2.23}$$

This formulation represents an autonomous system with the solution

$$\Psi_{\mathrm{rq}} = \mathrm{e}^{-\frac{R_{\mathrm{r}}}{L_\sigma}t} \ , \tag{2.24}$$

which is exponentially decaying with the time constant $\tau = \frac{L_\sigma}{R_{\mathrm{r}}}$. By using this approach, the arbitrary coordinate frame l is oriented into the direct axis of the rotor flux Ψ_{rd}. After the insertion of (2.22) into (2.19), the differential equations of the Γ-model in the flux fixed coordinate frame read as

$$u_{\mathrm{sd}} = R_{\mathrm{s}}\frac{L_\sigma + M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)}{L_\sigma M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)}\Psi_{\mathrm{sd}} - \frac{R_{\mathrm{s}}}{L_\sigma}\Psi_{\mathrm{rd}} + \frac{\mathrm{d}\Psi_{\mathrm{sd}}}{\mathrm{d}t} - \left(\omega + \frac{R_{\mathrm{r}}}{L_\sigma}\frac{\Psi_{\mathrm{sq}}}{\Psi_{\mathrm{rd}}}\right)\Psi_{\mathrm{sq}} \tag{2.25a}$$

$$u_{\mathrm{sq}} = R_{\mathrm{s}}\frac{L_\sigma + M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)}{L_\sigma M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)}\Psi_{\mathrm{sq}} + \frac{\mathrm{d}\Psi_{\mathrm{sq}}}{\mathrm{d}t} + \left(\omega + \frac{R_{\mathrm{r}}}{L_\sigma}\frac{\Psi_{\mathrm{sq}}}{\Psi_{\mathrm{rd}}}\right)\Psi_{\mathrm{sd}} \tag{2.25b}$$

$$0 = \frac{R_{\mathrm{r}}}{L_\sigma}(\Psi_{\mathrm{rd}} - \Psi_{\mathrm{sd}}) + \frac{\mathrm{d}\Psi_{\mathrm{rd}}}{\mathrm{d}t} \tag{2.25c}$$

and the currents are obtained by

$$i_{\mathrm{sd}} = \frac{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)(-\Psi_{\mathrm{rd}} + \Psi_{\mathrm{sd}}) + L_\sigma \Psi_{\mathrm{sd}}}{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)L_\sigma} \tag{2.26a}$$

$$i_{\mathrm{sq}} = \Psi_{\mathrm{sq}} \frac{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|) + L_\sigma}{M(\|\boldsymbol{\Psi}_{\mathrm{s}}\|)L_\sigma} \tag{2.26b}$$

$$i_{\mathrm{rd}} = \frac{\Psi_{\mathrm{rd}} - \Psi_{\mathrm{sd}}}{L_\sigma} \tag{2.26c}$$

$$i_{\mathrm{rq}} = -\frac{\Psi_{\mathrm{sq}}}{L_\sigma} \ , \tag{2.26d}$$

with the equation for the torque

$$\tau = \frac{3}{2} \frac{Z_{\mathrm{p}}}{L_\sigma} \Psi_{\mathrm{rd}} \Psi_{\mathrm{sq}} \ . \tag{2.27}$$

Since [20] uses the power-invariant space vector scaling and this work is based on amplitude-invariant scaling, the factor $\frac{3}{2}$ extends (2.27) compared to the formulation in [20]. The ohmic power loss, that is later needed for the optimal steady-state setpoint calculation and the MPC, is described by

$$P_{\mathrm{l}} = \frac{3}{2}(R_{\mathrm{s}} \mathbf{i}_{\mathrm{s}}^{\mathrm{T}} \mathbf{i}_{\mathrm{s}} + R_{\mathrm{r}} \mathbf{i}_{\mathrm{r}}^{\mathrm{T}} \mathbf{i}_{\mathrm{r}}) \ . \tag{2.28}$$

Furthermore, in the steady state, the time derivatives of the fluxes are set to zero and thus the direct stator flux and the direct rotor flux assume the same value according to (2.25c). By inserting this relation into (2.25a) and (2.25b), the steady-state stator voltages $u_{\mathrm{sd}}^{\mathrm{s}}$ and $u_{\mathrm{sq}}^{\mathrm{s}}$ are computed as

$$u_{\mathrm{sd}}^{\mathrm{s}} = -\omega \Psi_{\mathrm{sq}}^{\mathrm{s}} - \frac{R_{\mathrm{r}} (\Psi_{\mathrm{sq}}^{\mathrm{s}})^2}{L_\sigma \Psi_{\mathrm{sd}}^{\mathrm{s}}} + \frac{R_{\mathrm{s}}}{M} \Psi_{\mathrm{sd}}^{\mathrm{s}} \tag{2.29a}$$

$$u_{\mathrm{sq}}^{\mathrm{s}} = \omega \Psi_{\mathrm{sd}}^{\mathrm{s}} + \frac{R_{\mathrm{r}}}{L_\sigma} \Psi_{\mathrm{sq}}^{\mathrm{s}} + R_{\mathrm{s}} \frac{L_\sigma + M}{L_\sigma M} \Psi_{\mathrm{sq}}^{\mathrm{s}} \tag{2.29b}$$

in which the mutual inductance is evaluated according to (2.11). Additionally, the steady-state currents $i_{\mathrm{sd}}^{\mathrm{s}}$ and $i_{\mathrm{sq}}^{\mathrm{s}}$ are obtained by inserting the steady-state fluxes $\Psi_{\mathrm{sd}}^{\mathrm{s}}$ and $\Psi_{\mathrm{sq}}^{\mathrm{s}}$ into (2.10) and solving this relation for the currents according to

$$i_{\mathrm{sd}}^{\mathrm{s}} = \frac{\Psi_{\mathrm{sd}}^{\mathrm{s}}}{M} \tag{2.30a}$$

$$i_{\mathrm{sq}}^{\mathrm{s}} = \Psi_{\mathrm{sq}}^{\mathrm{s}} \frac{M + L_\sigma}{M L_\sigma} \tag{2.30b}$$

$$i_{\mathrm{rd}}^{\mathrm{s}} = 0 \tag{2.30c}$$

$$i_{\mathrm{rq}}^{\mathrm{s}} = -\frac{\Psi_{\mathrm{sq}}^{\mathrm{s}}}{L_\sigma} \ , \tag{2.30d}$$

in which the mutual inductance is again evaluated according to (2.11).

2.3 Scaling of model parameters

To make the simulations, which are performed in Section 3.6 and Section 4.7, comparable, the machine data and the simulation results in this work are depicted as normalized quantity. To keep the nomenclature as simple as possible, the same variable set of normalization parameters is used for the PMSM and the IM. However it is important to mention, that these parameters do not need to have the same value for the specific machine. The normalization parameters are summarized in Table 2.1.

Normalization Parameter	Description
ω_N	Maximum electrical rotational speed
I_N	Maximum stator current
τ_N	Maximum torque
U_N	Maximum stator voltage
Ψ_N	$\dfrac{U_\mathrm{N}}{\omega_\mathrm{N}}$

Table 2.1: Normalization parameters of the electrical machines.

3 MPC Approximation for PMSM

This chapter deals with the formulation and the approximation of different MPC types for the PMSM. First, the different types of MPCs that are investigated are introduced. In a second step, the data generation and the approximation is explained in more detail and some simulation results are shown.

3.1 Control Loop structure

As already mentioned in Section 1.1, an MPC is a model based control strategy, which optimizes the inputs of the plant over a prediction horizon into the future in order to minimize a certain cost functional [23]. Based on this optimization, the first input value is then applied to the system and the optimization loop starts in the next timestep again. In this work, three different MPC types are investigated:

- MPC without regularization terms
- MPC with a regularization term on the state
- MPC with a regularization term on the state and the input

By choosing these three formulations, the capabilities of the approximation on a set of different input variables of the MPC are shown. This shows the generalizability of the proposed approach to different input variables and formulations.

The control structure for the field-oriented control of the PMSM is shown in Figure 3.1. It consists of a setpoint calculation, which computes the steady-state optimal setpoints with respect to the copper losses, an integrator, which reduces the steady-state error and the MPC itself, which applies finally the computed voltages to the inverter and the motor.

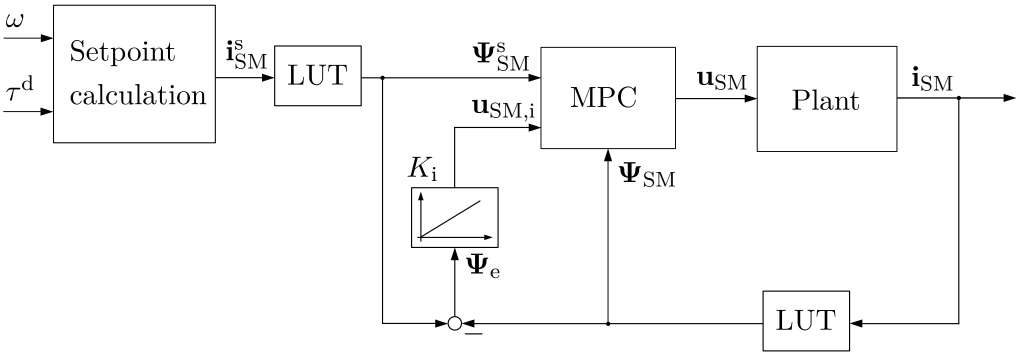


Figure 3.1: Block diagram of the control structure of the PMSM.

In the following the optimal setpoint calculation and the three MPC formulations are introduced and compared to each other in terms of complexity, dynamical characteristics and noise behavior.

3.2 Setpoint calculation

The main task of the setpoint calculation lies in the determination of steady-state, energy optimal setpoints $\boldsymbol{\Psi}^{\mathrm{s}}_{\mathrm{SM}} = [\Psi^{\mathrm{s}}_{\mathrm{d}}, \Psi^{\mathrm{s}}_{\mathrm{q}}]^{\mathrm{T}}$, while still achieving the desired torque. To compute the optimal setpoints, the losses in the machine must be minimized. Since the factor $\frac{3}{2}R_{\mathrm{s}}$ only scales the absolute value of the minimum of the stator losses in (2.7), but does not change the location of the minimum, the nonlinear optimization problem is formulated as

$$
\begin{aligned}
\mathbf{i}^{*}_{\mathrm{SM}} = &\operatorname*{argmin}_{\mathbf{i}_{\mathrm{SM}}} \mathbf{i}^{\mathrm{T}}_{\mathrm{SM}}\mathbf{i}_{\mathrm{SM}} && (3.1\mathrm{a})\\
\text{s.t.} \quad & \tau(\mathbf{i}_{\mathrm{SM}}) - \tau^{\mathrm{d}} = 0 && (3.1\mathrm{b})\\
& u_{\mathrm{d}}^2 + u_{\mathrm{q}}^2 \leq u_{\mathrm{HW}}^2 && (3.1\mathrm{c})\\
& i_{\mathrm{d}}^2 + i_{\mathrm{q}}^2 \leq i_{\mathrm{HW}}^2 && (3.1\mathrm{d})
\end{aligned}
$$

with the reference torque τ^{d} and the current torque from (2.6). In (3.1c) and (3.1d) u_{HW} and i_{HW} refer to the constraints for the voltage and the current respectively. These are the maximum values that can be applied to the hardware in the system. This optimization problem with the optimization vector $\mathbf{i}_{\mathrm{SM}} = [i_{\mathrm{d}}, i_{\mathrm{q}}]^{\mathrm{T}}$ yields the **M**aximum **T**orque **P**er **A**mpere (MTPA) curve in the base speed range. For higher speeds, the voltage constraint becomes active and thus the optimal solution is found on the voltage boundary curve. In the literature this region is often referred to as fieldweakening range. In this equation set, the superscript $(*)^{*}$ refers to the optimal solution and the superscript $(*)^{\mathrm{s}}$ refers to the desired optimal steady state setpoints. Since the states for the MPC are fluxes, the optimal current setpoints have to be transformed using the relationship (2.2) with LUTs for the inductances.

3.3 MPC formulation

3.3.1 MPC without regularization terms

The main goal of the MPC without any regularization terms is to follow a given trajectory over the time horizon N_{ph} as closely as possible while taking the current and voltage constraints into account. For this purpose, the deviation between the predicted states of the model and the optimal setpoints from (3.1a) are used. To improve the readability of this work, the notation $(*)_{m|j}$ is used, which indicates the predicted quantity on the prediction horizon at the time $t = (m + j)T_{\mathrm{s}}$ with the sampling time T_{s} for the MPC evaluated at $t = mT_{\mathrm{s}}$. By taking these considerations into account, the optimization problem and the MPC read as

$$\mathbf{u}^*_{\mathrm{SM},m|j} = \underset{\mathbf{u}_{\mathrm{SM},m|j}}{\operatorname{argmin}} J(\mathbf{u}_{\mathrm{SM},m|j}) =$$

$$= \underset{\mathbf{u}_{\mathrm{SM},m|j}}{\operatorname{argmin}} \sum_{j=1}^{N_{\mathrm{ph}}} \left(\mathbf{x}_{\mathrm{SM},m|j} - \mathbf{x}^{\mathrm{s}}_{\mathrm{SM},m|j}\right)^{\mathrm{T}} \left(\mathbf{x}_{\mathrm{SM},m|j} - \mathbf{x}^{\mathrm{s}}_{\mathrm{SM},m|j}\right) \tag{3.2a}$$

$$\text{s.t.} \quad \mathbf{x}_{\mathrm{SM},m|j+1} = \mathbf{F}_{\mathrm{SM},m|j}\ , \quad j = 0, ..., N_{\mathrm{ph}} - 1 \tag{3.2b}$$

$$u^2_{\mathrm{d},m|j} + u^2_{\mathrm{q},m|j} \leq u^2_{\mathrm{mpc,HW}}\ , \quad j = 0, ..., N_{\mathrm{ph}} - 1 \tag{3.2c}$$

$$i^2_{\mathrm{d},m|j} + i^2_{\mathrm{q},m|j} \leq i^2_{\mathrm{HW}}\ , \quad j = 1, ..., N_{\mathrm{ph}}\ , \tag{3.2d}$$

with the state $\mathbf{x}_{\mathrm{SM},m|j} = [\Psi_{\mathrm{d},m|j}, \Psi_{\mathrm{q},m|j}]^{\mathrm{T}}$ and the input $\mathbf{u}_{\mathrm{SM},m|j} = [u_{\mathrm{d},m|j}, u_{\mathrm{q},m|j}]^{\mathrm{T}}$, which represents also the optimization variable. The function $\mathbf{F}_{\mathrm{SM},m|j}$ stands here as an abbreviation for the evaluation of the state space system in (2.4). The boundary value $u_{\mathrm{mpc,HW}}$ is defined as

$$u_{\mathrm{mpc,HW}} = u_{\mathrm{HW}} - \|\mathbf{u}_{\mathrm{SM,i}}\|_2\ , \tag{3.3}$$

where $\mathbf{u}_{\mathrm{SM,i}}$ defines the voltage that comes from the integrator. This voltage is needed in order to achieve steady state accuracy also in the case of parameter deviations.

The first state for the prediction of the future states is set to be the measured state $\mathbf{x}_{\mathrm{SM},m|0} = \mathbf{x}_{\mathrm{SM},m}$. To save even more computation time for the evaluation of the MPC, the dynamic matrix and the input matrix are only computed once at the current operating point of the machine and kept constant over the prediction horizon. The predicted change of the operating point over the prediction horizon is neglected in the evaluation of the LUTs, since the influence of this effect is minor. This is due to the high sample rate of the controller and thus the small length of the prediction horizon, during which the saturation behavior of the state matrices does not change strongly.

3.3.2 MPC with regularization on the state

The main goal of the MPC with regularization on the state is similar to the MPC without regularization terms- the losses in each optimization step should be reduced, while achieving the desired torque. In contrast to the previous MPC, this MPC has an extra term that penalizes the discrete time derivative (the difference of the state vector between two subsequent states). By adding this term, the state trajectory should become smoother, since its rate of change is penalized. This achieves a better noise suppression, as the state vector can not react as fast to high frequency noise as for the MPC without regularization. However, this comes at the cost of a slower dynamical behavior. Taking these considerations into account, the optimization problem and the MPC read as

$$\begin{aligned}
\mathbf{u}^{*}_{\mathrm{SM},m|j} &= \underset{\mathbf{u}_{\mathrm{SM},m|j}}{\operatorname{argmin}}\, J(\mathbf{u}_{\mathrm{SM},m|j}) = \\
&= \underset{\mathbf{u}_{\mathrm{SM},m|j}}{\operatorname{argmin}} \sum_{j=1}^{N_{\mathrm{ph}}} K_{\mathrm{x}} \left\| \mathbf{x}_{\mathrm{SM},m|j} - \mathbf{x}^{\mathrm{s}}_{\mathrm{SM},m|j} \right\|_2^2 + K_{\Delta\mathrm{x}} \left\| \mathbf{x}_{\mathrm{SM},m|j} - \mathbf{x}_{\mathrm{SM},m|j-1} \right\|_2^2 \qquad (3.4\mathrm{a})\\
\text{s.t.} \quad & \mathbf{x}_{\mathrm{SM},m|j+1} = \mathbf{F}_{\mathrm{SM},m|j} \ , \quad j = 0, ..., N_{\mathrm{ph}} - 1 \qquad (3.4\mathrm{b})\\
& u^2_{\mathrm{d},m|j} + u^2_{\mathrm{q},m|j} \leq u^2_{\mathrm{mpc,HW}} \ , \quad j = 0, ..., N_{\mathrm{ph}} - 1 \qquad (3.4\mathrm{c})\\
& i^2_{\mathrm{d},m|j} + i^2_{\mathrm{q},m|j} \leq i^2_{\mathrm{HW}} \ , \quad j = 1, ..., N_{\mathrm{ph}} \ , \qquad (3.4\mathrm{d})
\end{aligned}$$

with the positive weights K_{x} and $K_{\Delta\mathrm{x}}$. These weights are tuning parameters of the MPC. If the ratio between the weight on the state K_{x} and the weight on the difference of the state $K_{\Delta\mathrm{x}}$ is high, the MPC will behave similarly as the MPC without regularization terms. For a low value of this ratio on the other hand, the MPC will show a slow dynamic behavior with a high noise suppression capability. For the usage on a real machine, these parameters have to be tuned on the testbench to achieve a trade-off between high dynamical behavior and smooth state trajectories. The rest of the MPC is the same as for the MPC without regularization terms. One notable difference is, that the computation time to solve the optimization problem for one timestep is significantly higher than for the MPC without regularization terms. This can be explained by the fact, that the optimization problem got more complex by the introduction of the extra term.

3.3.3 MPC with regularization on the state and on the input

The main goal of the MPC with regularization on the state and on the input is equivalent to the MPC without regularization terms. For this MPC, the regularization is even stronger than for the MPC with regularization on the state, since the input vector and the state vector are penalized by the optimization problem. By adding the penalization term on the discrete time derivative of the input, the input trajectory becomes smoother, since its change rate is penalized. This achieves an even higher noise suppression than for the MPC with regularization on the state. Taking these considerations into account, the optimization problem and the MPC read as

$$\begin{aligned}
\mathbf{u}^{*}_{\mathrm{SM},m|j} &= \underset{\mathbf{u}_{\mathrm{SM},m|j}}{\operatorname{argmin}}\, J(\mathbf{u}_{\mathrm{SM},m|j}) = \\
&= \underset{\mathbf{u}_{\mathrm{SM},m|j}}{\operatorname{argmin}} \sum_{j=1}^{N_{\mathrm{ph}}} K_{\mathrm{x}} \left\| \mathbf{x}_{\mathrm{SM},m|j} - \mathbf{x}^{\mathrm{s}}_{\mathrm{SM},m|j} \right\|_2^2 + K_{\Delta\mathrm{x}} \left\| \mathbf{x}_{\mathrm{SM},m|j} - \mathbf{x}_{\mathrm{SM},m|j-1} \right\|_2^2 + \\
&\quad + K_{\Delta\mathrm{u}} \left\| \mathbf{u}_{\mathrm{SM},m|j} - \mathbf{u}_{\mathrm{SM},m|j-1} \right\|_2^2 \qquad (3.5\mathrm{a})\\
\text{s.t.} \quad & \mathbf{x}_{\mathrm{SM},m|j+1} = \mathbf{F}_{\mathrm{SM},m|j} \ , \quad j = 0, ..., N_{\mathrm{ph}} - 1 \qquad (3.5\mathrm{b})\\
& u^2_{\mathrm{d},m|j} + u^2_{\mathrm{q},m|j} \leq u^2_{\mathrm{mpc,HW}} \ , \quad j = 0, ..., N_{\mathrm{ph}} - 1 \qquad (3.5\mathrm{c})\\
& i^2_{\mathrm{d},m|j} + i^2_{\mathrm{q},m|j} \leq i^2_{\mathrm{HW}} \ , \quad j = 1, ..., N_{\mathrm{ph}} \ , \qquad (3.5\mathrm{d})
\end{aligned}$$

with the positive weight $K_{\Delta\mathrm{u}}$. This weight is an additional tuning parameter of the MPC. If this weight dominates the other weights, the input will become a smooth trajectory at the cost of dynamical behavior. However, since the input has direct influence on the state itself, the smoothing of the input will deliver also a smoother state at the end. For this MPC, the tuning effort is even higher than for the MPC with regularization on the state, since the weight on the state and the weight on the input are not completely independent from each other. This extra term makes the optimization problem more complex and thus its evaluation slower.

3.4 Data generation

3.4.1 MPC without regularization and MPC with regularization on the state

In order to perform an approximation of the control action of the MPC, simulations have to be performed, which cover the whole operation space of the electrical machine. The MPC formulation in (3.2) and (3.4) and the equation for the maximum voltage in (3.3) show, that seven input variables are needed in order to evaluate these MPCs in a specific operation point. The input variables for the evaluation of the MPCs are summarized in

$$\zeta_{\mathrm{mpc,SM}} = [i_{\mathrm{d}}, i_{\mathrm{q}}, i_{\mathrm{d}}^{\mathrm{s}}, i_{\mathrm{q}}^{\mathrm{s}}, u_{\mathrm{d,i}}, u_{\mathrm{q,i}}, \omega]^{\mathrm{T}} \ . \tag{3.6}$$

From the seven variables in (3.6), the integrator voltages in the direct and quadrature direction $u_{\mathrm{d,i}}$ and $u_{\mathrm{q,i}}$ are important to determine the true voltage boundary in (3.3). The electrical speed is necessary to evaluate the dynamic matrix $\boldsymbol{\Phi}_{\mathrm{SM}}$ in (2.5). The other four variables are directly representing the optimization goal of the MPC in (3.2b), namely the two true currents that represent the current state i_{d}, i_{q} and the two setpoint currents $i_{\mathrm{d}}^{\mathrm{s}}$, $i_{\mathrm{q}}^{\mathrm{s}}$. Here the currents are used as inputs, since the definition of a maximum current for the machine is a safety boundary for the possible operation range. Therefore, the input currents in the two respective coordinate system directions are used as inputs for the MPC and its approximation in Section 3.5 instead of defining a simulation grid directly for the fluxes. The corresponding fluxes to a set of input currents is obtained by evaluating the LUTs according to (2.2). The advantage of this approach is, that the input grid for the simulations of the MPC can be defined very easily. Furthermore, the computational effort for the evaluation of the approximation in a later step is lower, since the LUT evaluation has not to be performed for the approximation. The downside of this approach is, that the LUTs can not be exchanged at a later stage, since they are part of the approximation itself. Furthermore, the approximation of the nonlinearity of the LUTs adds another layer of complexity for the approximation of the MPC behavior. Since the main goal of this work is to decrease the computational effort, however, this approach yields the smallest evaluation time, since only the data driven approach is evaluated and the additional evaluation of a LUT is not needed.

In order to perform suitable simulations from which the behavior of the MPC can be learned, the parameter space for the inputs has to be defined. For the usage in a car, simulations for the motor and generator mode of the electrical machine are of big importance for the approximation of the MPC. The reason for this is, that the voltage drop over the stator resistance R_s in the quadrature direction changes its sign when the current becomes negative. This behavior finally leads to different setpoints for the MTPA calculation. In order to show a proof of concept in this work, only the motoric region is considered in the parameter ranges of this PMSM. For the following considerations, an electrical machine with the input parameter space in Table 3.1 is chosen. The input space for the electrical speed covers the entire positive speed range of this electrical machine. Since this machine is symmetric, simulations over the positive and the negative speed range would be redundant, since the same results are expected for positive and negative speeds. Dynamic setpoint changes at different parameter deviations of the MPC define the input parameter space for the integrator voltages by simulations. For this purpose, the integrator weights are tuned in order to ensure less than 5% overshoot in the whole operating range including parameter deviations in the flux LUTs of ±30%.

Parameter	Minimum	Maximum
i_d	$-I_\mathrm{N}$	0
i_q	0	I_N
i_d^s	$-I_\mathrm{N}$	0
i_q^s	0	I_N
$u_\mathrm{d,i}$	$-0.04U_\mathrm{N}$	$0.04U_\mathrm{N}$
$u_\mathrm{q,i}$	$-0.04U_\mathrm{N}$	$0.04U_\mathrm{N}$
ω	0	ω_N

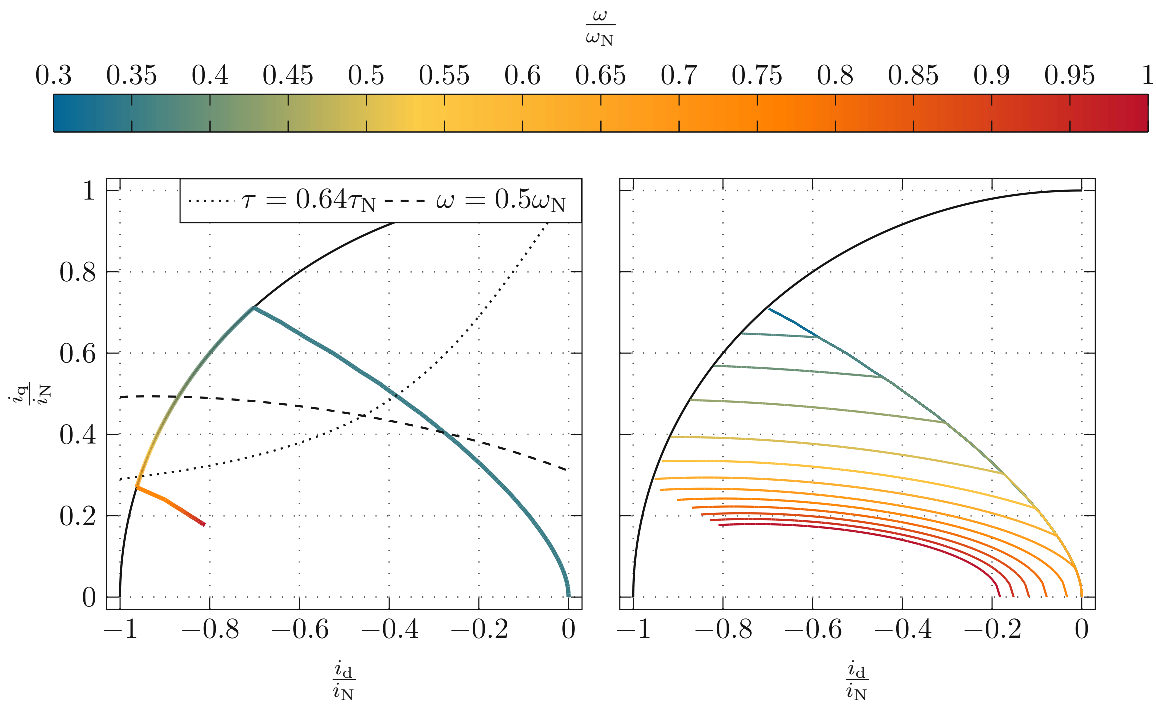
Table 3.1: Input parameter space of the PMSM.

To accomplish a high approximation accuracy, the parameter ranges from Table 3.1 have to be covered by a fine grid, because the nonlinear nature of the control structure and the system is sensible to big interpolation errors by the data driven approach. This need for a fine simulation grid leads to a high number of simulations to be performed and thus to a high computational effort. Since the input variable space is seven dimensional, the so called curse of dimensionality has also to be considered. This concept was introduced first by Richard E. Bellman in [24] and says, that with increasing dimensionality, the volume of the space increases so fast, that the available points become sparse. If the parameter ranges in Table 3.1 are for example sampled by a grid with $n_\mathrm{i} = 41$ elements for i_d, i_q, i_d^s and i_q^s respectively, a grid with $n_\mathrm{ui} = 5$ elements for the integrator voltages in direct and quadrature direction and a grid with $n_\omega = 17$ speedpoints for the electrical speed, this would correspond to

$$n_\mathrm{sim} = n_\mathrm{i}^4 \cdot n_\mathrm{u,i}^2 \cdot n_\omega = 41^4 \cdot 5^2 \cdot 17 \approx 1.2 \cdot 10^9 \tag{3.7}$$

simulations. The reason for this high number of simulations is, that every possible permutation of inputs has to be simulated. By assuming an average simulation time of 5ms for each simulation, this would finally lead for this 1.2 billion simulations to a computation time of 6 million seconds or approximately 70 days on a single core. Since the computations are independent from each other, this problem is very well parallelizable, but even on 8 cores it would still take up to 9 days to perform all these simulations.

For this purpose a different sampling technique is chosen in this work. PMSMs are usually driven in the base speed region along the MTPA curve, at higher speeds as soon as the voltage boundary is active along the voltage boundary in the so called field weakening region and at the highest speeds and high torques along the MTPV (**M**aximum **T**orque **P**er **V**olt) curve. The optimal operating strategy for an increasing speed is shown in Figure 3.2a. This figure shows a torque sweep for the blue depicted MTPA curve for small speeds. As soon as the machine reaches its maximum torque, the speed is increased. The figure shows, that the machine is driven explicitly in the MTPA region until a speed of $0.3\omega_\mathrm{N}$ is reached. For higher speeds however, the voltage boundary gets more restrictive, since the Back EMF increases. This results in a lower maximal achievable torque in the field weakening region, where the electrical machine is controlled at the voltage boundary and the current boundary for the highest torques– this results in movement among the current boundary curve, where the maximum achievable torque is decreased. By increasing the speed above $0.6\omega_\mathrm{N}$ it can be observed, that the operating strategy detaches from the current boundary circle. This can be explained by the fact, that the voltage boundary has an elliptical form in the plot over the direct and quadrature current. Since the isotorque lines are hyperbolas in the second quadrant, one reaches the maximum achievable torque in the high speed region by finding the point where the isotorque line is exactly tangent to the elliptical voltage boundary constraint. This can only be the case, if the semimajor



(a) Operating strategy over speed. (b) Operating strategy for discrete speeds.

Figure 3.2: Optimal operating strategy for the PMSM.

of the ellipse lies on the right of the current boundary, since otherwise there is no tangent point in the allowed region where the current and the voltage boundary are met. In Figure 3.2a, an arbitrary isotorque line for a value of $\tau = 0.64\tau_{\mathrm{N}}$ and the voltage constraint for an arbitrary speed of $\omega = 0.5\omega_{\mathrm{N}}$ are shown.

Figure 3.2b shows the operating strategy for some discrete electrical speeds, where the torque is increased from 0Nm to the maximum achievable torque at each speed. Here the transition between the MTPA controlled region and the fieldweakening region can be observed by the fact, that for smaller torques the curves move along the MTPA curve and for higher torques and higher speeds the elliptical voltage boundary gets active and the curve changes to proceed on the path of the voltage boundary in order to achieve higher torques. For speeds above $0.65\omega_{\mathrm{N}}$ it can also be observed, that the machine can not be controlled in the MTPA region for 0Nm anymore, since the voltage boundary is active at all times. For this high speed region a negative direct current has to be set even at 0Nm in order to weaken the field, since the Back EMF has become already so strong, that it creates more counter voltage than the inverter can apply to the system.

By taking these considerations into account, the sampling can be adapted in such a way, that the behavior of the electrical machine around these operation regions is sampled in a fine fashion without an exploding computational time effort. Therefore the setpoint currents and after the LUT evaluation the setpoint fluxes are sampled around the optimal operating curves of the PMSM. This sampling is shown in Figure 3.3. In this figure all simulation points for the different speeds are plotted. For a representative speed of $0.5\omega_{\mathrm{N}}$, the sampling is shown by the red dots. For the generation of this simulation

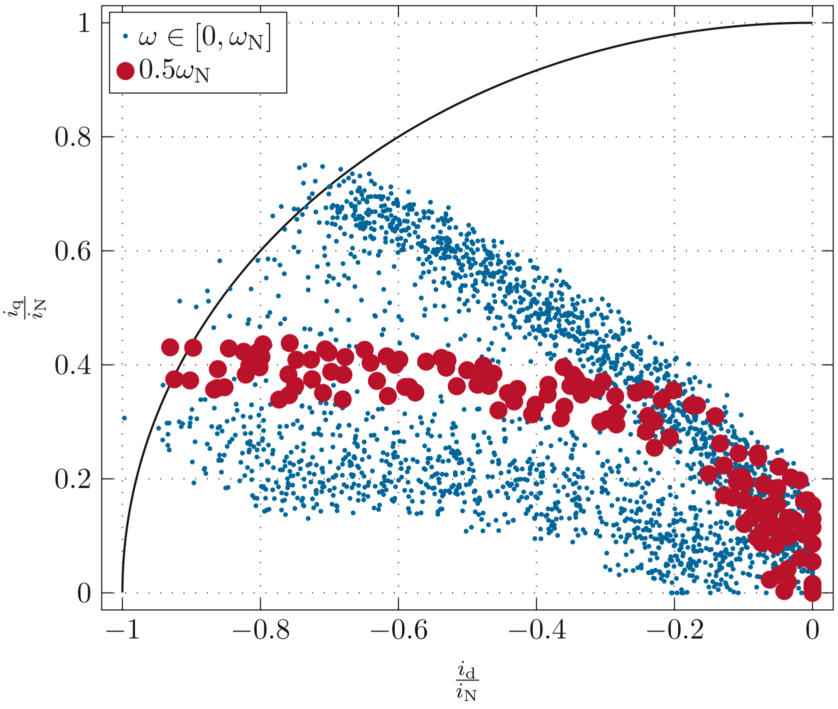
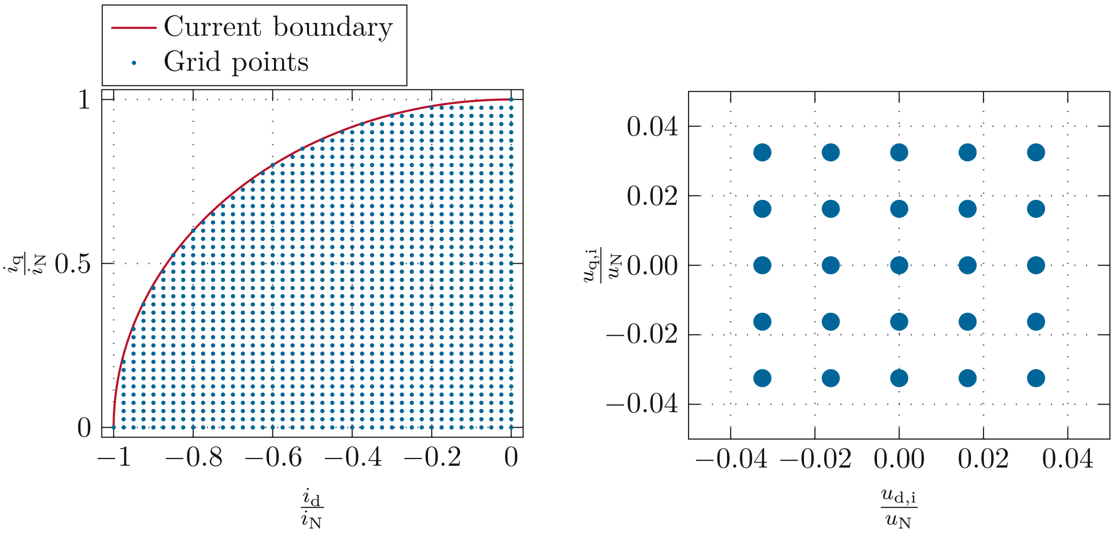


Figure 3.3: Simulation grid for the optimal setpoint currents in the machine.



(a) Grid for the actual currents. (b) Grid for the integrator voltages.

Figure 3.4: Simulation grids for the actual currents and the integrator voltages in the machine.

grid, each curve of Figure 3.2b is sampled by $n_{\mathrm{i,SP}} = 150$ points. For each point two random values are picked from a uniform distribution with a maximum value of $0.05 I_{\mathrm{N}}$ and these values are then added to the direct and quadrature component of the current respectively. By choosing this sampling, the optimal operating strategy for the ideal system for each speed is contained in the dataset and furthermore the randomness can help against parameter deviations, which would lead to slightly different curves. This helps the data driven approach by making the dataset more representative and thus potentially increasing the performance of the interpolation. In principle, every speed interval is captured by the sampling seperately from all other speed intervals. Since the region around $\mathbf{i}_{\mathrm{SM}} = [0.7 I_{\mathrm{N}}, 0.5 I_{\mathrm{N}}]^{\mathrm{T}}$ is sparsely populated with simulation points by this sampling scheme, another speed interval is added to the sampling to generate more simulation points in this region and increase the interpolation capability of the approach in this region. However, this increases the dimension of the speed vector by one element and thus the computational simulation effort.

Figure 3.4 shows the sampling scheme in detail for the current states of the electrical machine and the integrator voltages. For the currents in Figure 3.4a the uniform grid that was described before was used, since the electrical machine can principally be operated in each of these current combinations and therefore every combination should be resolved by the sampling in a fine grid. The currents with a bigger magnitude than the current boundary are omitted here, since the electrical machine should not be operated in these points under normal conditions. This yields $n_{\mathrm{i,states}} = 1297$ points for the actual currents in the machine. For the integrator voltages a uniform grid as shown in Figure 3.4b is used. This finally results in a simulation effort of

$$n_{\mathrm{SM}} = n_{\mathrm{i,states}} \cdot n_{\mathrm{i,SP}} \cdot n_{\mathrm{u,i}}^2 \cdot n_\omega = 1297 \cdot 150 \cdot 5^2 \cdot 18 \approx 8.8 \cdot 10^7 \tag{3.8}$$

points, which is by a factor ≈ 13 smaller than the result that would be obtained by the sampling in (3.7). The computation time could therefore also be brought down by a factor of approximately 13 and therefore these simulation can be run over one day on 8 cores.

3.4.2 MPC with regularization on the state and on the input

For the MPC with regularization on the state and on the input, the input parameter space increases by two, since the voltage vector in the first timestep is not known at the beginning of the evaluation. To approximate the behavior in a meaningful way, the starting voltages in the direct and quadrature direction must be provided as two extra inputs to the system. Therefore the input vector for this MPC extends to

$$\zeta^{\mathrm{ext}}_{\mathrm{mpc,SM}} = \left[i_{\mathrm{d}}, i_{\mathrm{q}}, i^{\mathrm{s}}_{\mathrm{d}}, i^{\mathrm{s}}_{\mathrm{q}}, u_{\mathrm{d,i}}, u_{\mathrm{q,i}}, \omega, u_{\mathrm{d},0}, u_{\mathrm{q},0}\right]^{\mathrm{T}} , \tag{3.9}$$

with the starting values for the voltage $u_{\mathrm{d},0}$ and $u_{\mathrm{q},0}$. By adding these two variables, the machine is able to start in every dynamical operating point for the evaluation of the MPC and the smoothing term can be evaluated accordingly.

As already mentioned in Section 3.4.1, the curse of dimensionality plays a pivotal role in the generation of the input data. For this MPC, the effect is even stronger, since the input dimension increases by two. Furthermore the evaluation time of the MPC is longer than for the other MPCs, which makes a sampling as in Section 3.4.1 not feasible.

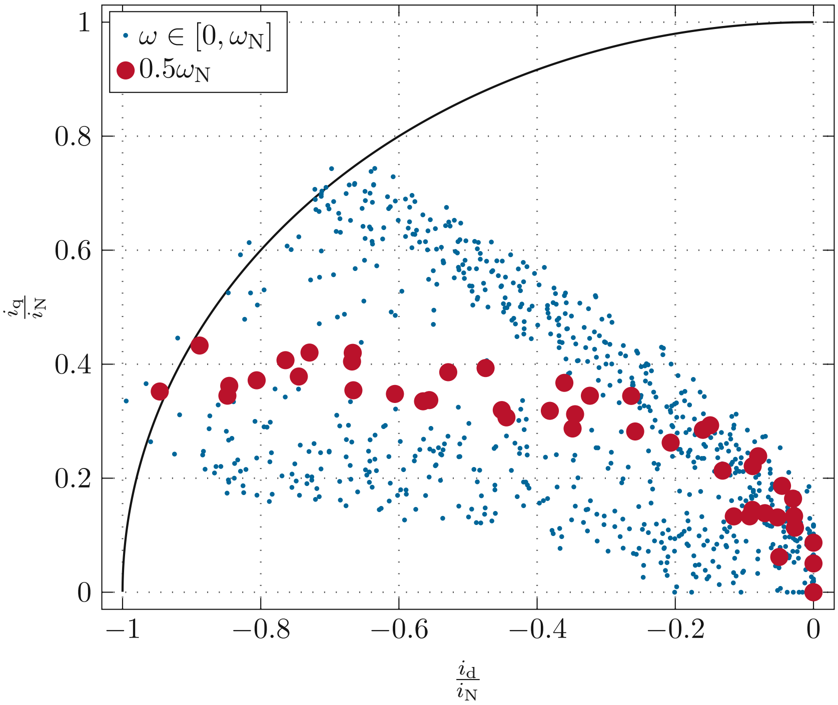
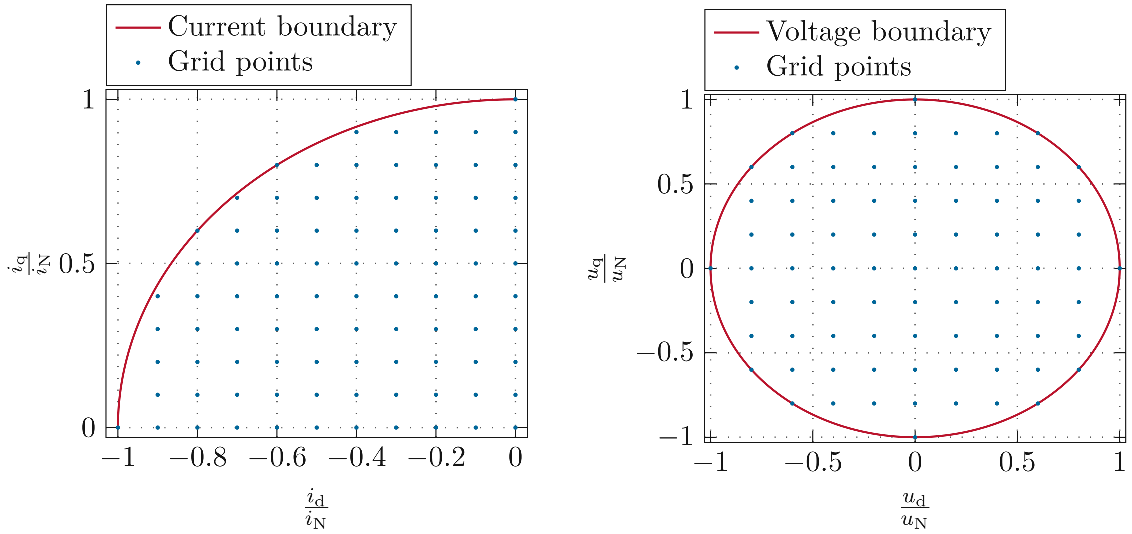


Figure 3.5: Simulation grid for the optimal setpoint currents in the machine for the MPC with regularization on the state.

For this purpose, the sampling is chosen differently and much coarser for this MPC. The



(a) Grid for the actual currents.

(b) Sampling grid for the voltages.

Figure 3.6: Simulation grids for the actual currents and the voltages from the last timestep.

actual currents in the machine and starting voltages are sampled according to Figure 3.6, with $n_{\mathrm{i,states}} = 90$ points for the currents and $n_{\mathrm{u,0}} = 81$ points for the starting voltages. Furthermore, only $n_{\mathrm{i,SP}} = 45$ instead of 150 points are used for each speed interval for the current setpoint values. The integrator voltages are finally sampled in a grid with $n_{\mathrm{u,i}} = 3$ samples per dimension instead of 5 samples per dimension. For the speed grid the same resolution as for the MPC without regularization on the input is used. The resolution of these grids is kept this low to be able to perform the simulations on a normal hardware on a short time frame. For the here defined parameter ranges

$$n_{\mathrm{SM,ext}} = n_{\mathrm{i,states}} \cdot n_{\mathrm{i,SP}} \cdot n_{\mathrm{u,i}}^2 \cdot n_\omega \cdot n_{\mathrm{u,0}} = 90 \cdot 45 \cdot 3^2 \cdot 18 \cdot 81 \approx 5.3 \cdot 10^7 \tag{3.10}$$

simulations have to be performed to evaluate every permutation of inputs. This shows the curse of dimensionality, since the resolution of most of the input variables was decreased drastically, while reducing the number of simulations only by a factor of 1.6. Since the simulations take approximately six times as long (≈ 30ms) as for the MPC without regularization, the overall computation time for the generation of the simulation data increases to 18 days on a single core or approximately 2.5 days on eight cores. This means, that in total the simulations run 2.5 times as long as the before mentioned simulations for the other MPCs. However, this coarse grid has also a benefit to it, since the approximation accuracy on a smaller dataset can be evaluated and compared to the finer grids.

3.5 Data driven approach

After the simulations have been performed, the approximation of the MPC takes place. For this purpose the *Python Programming Language* was chosen, since it contains many different powerful libraries in the field of machine learning.

In order to check the correlations between the individual inputs and the outputs, the so

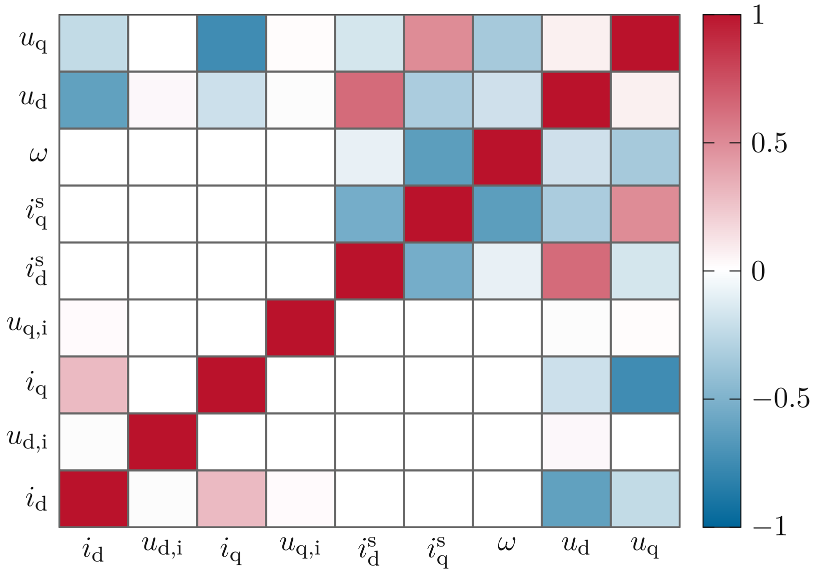


Figure 3.7: Correlation matrix for the input and output variables for the MPC without regularization.

called correlation matrix [25] is computed and shown in Figure 3.7. This matrix shows in the diagonal the autocorrelation and in all the other elements the cross correlation between the different inputs and outputs. This matrix gives a first indication about the capability of learning the regression policy, since it shows the linear interconnections between the different variables and especially if the output is correlated to the input. Figure 3.7 shows, that the output variables u_d and u_q are correlated with all input variables (either by a negative correlation or a positive correlation) except for the integrator voltages. This can be explained by the fact, that the underlying differential equations show a direct connection between the inputs and the outputs, but the integrator voltages are only there in order to limit the maximum voltage and don't correlate linearly with the output variables. However through the capability of data driven approaches to find nonlinear correlations, these variables have also an impact on the output and help to achieve stationary accuracy in the case of parameter deviations.

For the MPC with regularization on the input and the state, the same correlations are observed. The two extra inputs, namely the starting voltages, correlate only strongly with the output. This behavior is expected, since the regularization on the input smoothens the ouput trajectory and does not allow for steep changes. Therefore, these variables are strongly positively correlated with the output voltages. The correlation matrix is shown in Figure 3.8.

In order to perform the approximation of the input-output behavior of the neural net, a fully connected feedforward neural net is chosen. The inputs for the data driven approach are the variable inputs for the evaluation of the MPC $\boldsymbol{\zeta}_\mathrm{mpc,SM}$ and $\boldsymbol{\zeta}_\mathrm{mpc,SM}^\mathrm{ext}$ defined in (3.6) and (3.9) respectively and the outputs are the two voltages $\mathbf{u}_\mathrm{SM} = [u_\mathrm{d}, u_\mathrm{q}]^\mathrm{T}$ that are

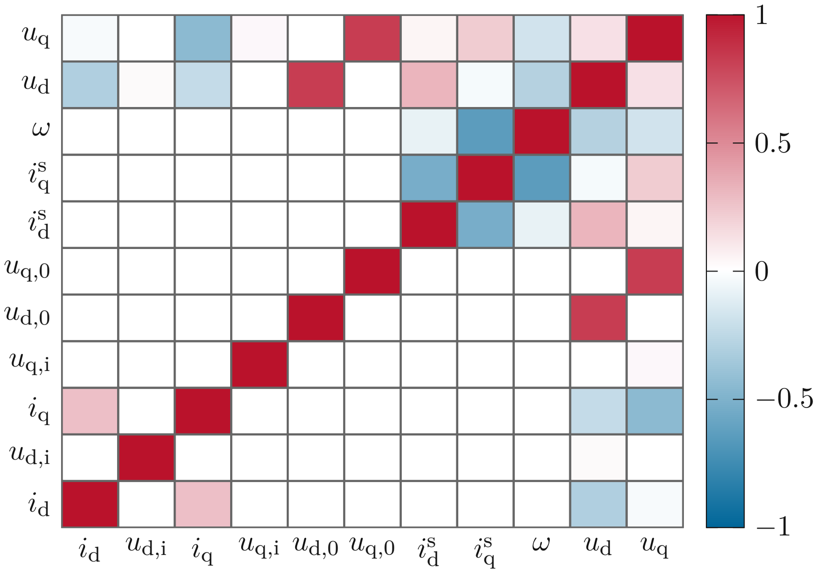


Figure 3.8: Correlation matrix for the input and output variables for MPC with regularization on input and state.

applied to the inverter. The performed simulations are collected in the matrices

$$\mathbf{X}_{\mathrm{SM}} = \begin{bmatrix} \zeta^{\mathrm{T}}_{\mathrm{mpc,SM},1} \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,SM},2} \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,SM},3} \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,SM},4} \\ \vdots \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,SM},n_{\mathrm{SM}}} \end{bmatrix} , \qquad \mathbf{Y}_{\mathrm{SM}} = \begin{bmatrix} \mathbf{u}^{\mathrm{T}}_{\mathrm{SM},1} \\ \mathbf{u}^{\mathrm{T}}_{\mathrm{SM},2} \\ \mathbf{u}^{\mathrm{T}}_{\mathrm{SM},3} \\ \mathbf{u}^{\mathrm{T}}_{\mathrm{SM},4} \\ \vdots \\ \mathbf{u}^{\mathrm{T}}_{\mathrm{SM},n_{\mathrm{SM}}} \end{bmatrix} , \tag{3.11}$$

where $\mathbf{X}_{\mathrm{SM}}$ contains all the inputs and $\mathbf{Y}_{\mathrm{SM}}$ contains all the outputs. Here it is shown only for the inputs and outputs of the MPC without regularization and the MPC with regularization on the state. However, this transfers in the same way to the MPC with regularization on the input and the state with the extra two inputs. To check the fitting performance of the neural net, the inputs and the outputs are split up into a training set consisting of $\mathbf{X}_{\mathrm{train}}$ and $\mathbf{Y}_{\mathrm{train}}$, which contains 80% of the points, and a validation set consisting of $\mathbf{X}_{\mathrm{test}}$ and $\mathbf{Y}_{\mathrm{test}}$, which contains 20% of the points. Furthermore the two sets are shuffled to minimize the correlation between two subsequent simulation sets.

The inputs show big differences in their respective magnitude. This can lead to numerical problems for the fitting of the neural network and thus deteriorate the approximation. Therefore the inputs were scaled by a minimum maximum scaler, given by

$$\overline{\mathbf{X}}_{\mathrm{train},m} = \frac{\mathbf{X}_{\mathrm{train},m} - \min(\mathbf{X}_{\mathrm{train},m})}{\max(\mathbf{X}_{\mathrm{train},m}) - \min(\mathbf{X}_{\mathrm{train},m})} , \qquad m = 1, 2, 3, ..., 7 , \tag{3.12}$$

where each column is normalized into the range between zero and one. It is important to mention, that the scaling is only performed on the training set, since for the training no information about the validation should be used in order to prevent overfitting.

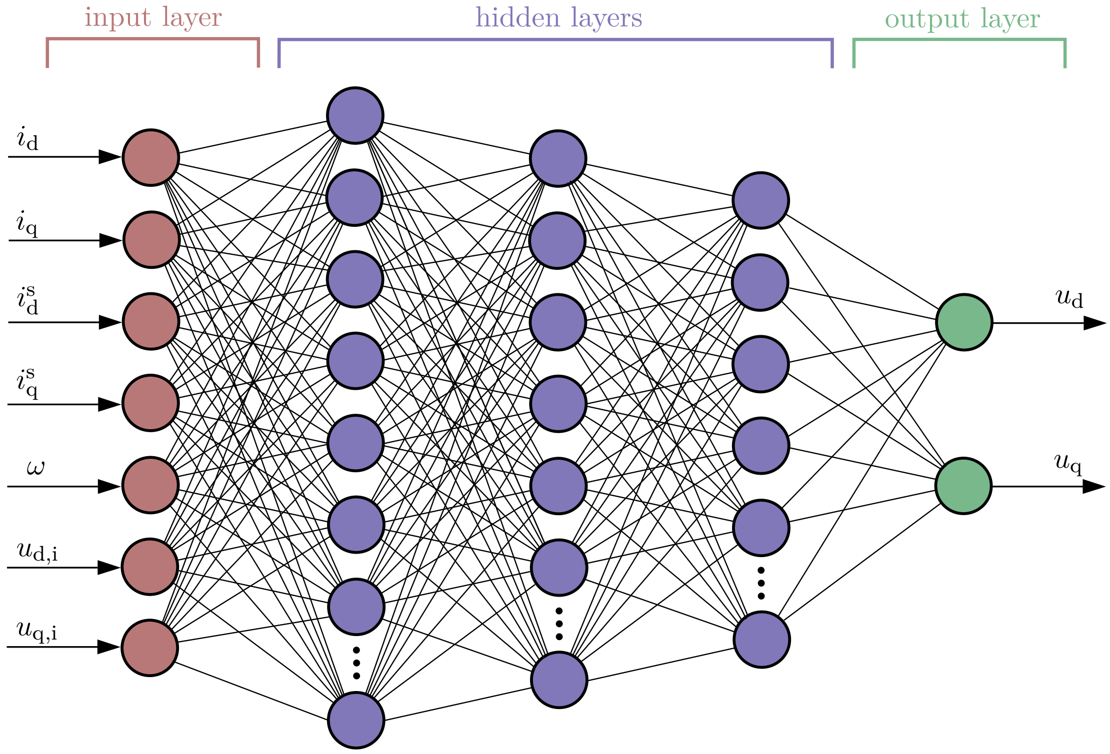


Figure 3.9: Structure of the feedforward neural net for the MPC without regularization and the MPC with regularization on the state [26].

The net structure for the fitting is shown in Figure 3.9 and it consists of a seven (or nine for the MPC with regularization on state and input) dimensional input layer, three hidden layers with 100, 70 and 50 nodes respectively and a two dimensional output layer. As activation function for the hidden layers ReLU (**Re**ctified **L**inear **U**nit) is chosen, because it introduces nonlinearity to the neural network and enables therefore the network to learn more complex relationship in the dataset [27]. In contrast to the classical activation functions, like sigmoid or tanh, it has no problem with vanishing gradients since the gradient of the activation function is constant for positive values of the input [28]. Furthermore it is often faster to compute, since the activation function

$$h(x) = \max(0, x) \tag{3.13}$$

is easier to compute than the sigmoid function or the tanh function.

One drawback of the ReLU activation function is, that it can only predict positive values, since the activation function clips all negative values to 0. If the neural network should be able to predict positive and negative values, the output layer has therefore to be able to predict values in the positive and negative range. Therefore, the linear activation function was used for the output layer. It introduces no extra nonlinearity, since this is already covered by the ReLU, but it extends the prediction range of the neural network to the full operating range of the electrical machine.

The size of the input and the output is predefined by the simulation results of the MPC itself. The size of the hidden layers on the other hand can be designed according to the problem. The dimensions that are specified here show a good tradeoff between approximation accuracy and computational effort. By making the neural network deeper (adding extra hidden layers), the approximation quality improves marginally, however the computation time increases since the last layer can only be evaluated after the other layers. By increasing the width of the neural network and thus increasing the number of nodes, no real benefit in the approximation accuracy can be identified. This means, that the here shown neural network is wide enough in order to approximate the desired behavior of the MPC. However, an ablation study could be performed in order to further reduce the execution time of the neural net under the condition to keep the prediction performance as it is [29]. The procedure of the ablation study is shown for the case of the induction machine in Section 4.6.

For the optimizer of the neural network Adamax [30] is chosen, because it shows a faster convergence for this problem than SGD (**S**tochastic **G**radient **D**escent) [31]. For the learning rate of the optimizer a learning rate scheduler was used, which starts at a value of $\lambda = 0.01$ and reduces its learning rate based on the evolution of the training loss. If the training loss remains too long on a plateau and does not decrease further, the learning rate might be too big and is thus divided by a preset factor of 10. This enables a faster learning with bigger steps in the beginning and more precise results towards the end, when the learning rate decreases. Furthermore *Early Stopping* was used in order to mitigate the problem of overfitting on the training dataset. For the loss function itself the MSE (**M**ean **S**quared **E**rror) loss was used, since it performs well if the dataset is in the same order of magnitude, which is the case. Furthermore it penalizes outliers in the prediction heavily, which is important for the approximation of a control scheme that should work in all operating regions.

To check the approximation accuracy of the approach, Figure 3.10 shows a Q-Q plot (also called 45° plot) of the prediction of the MPC without regularization [32]. For this plot, the actual output values of the MPC $\mathbf{u}_{\mathrm{SM}}^{\mathrm{mpc}}$ are plotted dependent of the prediction values $\mathbf{u}_{\mathrm{SM}}^{\mathrm{app}}$. The prediction values are very close to the 45° line, which would correspond to a perfect prediction of the true output values. Since for the other two approaches the plot looks almost identical, they are not shown in this work. The maximum voltage error is equal to 3% of the maximum voltage of the system, which indicates a good prediction accuracy. The root mean squared error scaled by the maximum voltage for this prediction is equal to 0.0041, which corresponds to 0.4% of the maximum voltage in the system. Furthermore, approximately 97% of all points lie in a region of three times the standard deviation between prediction and actual values around the 45° line. Therefore, this plot shows the high approximation accuracy and that the approximation performs well in the whole operating range of the prediction without outliers for higher or lower voltages.

3.6 Simulation Results

The developed approximation approach is checked on its feasibility for stationary and dynamic simulations. These simulations are closed-loop simulations with an inverter

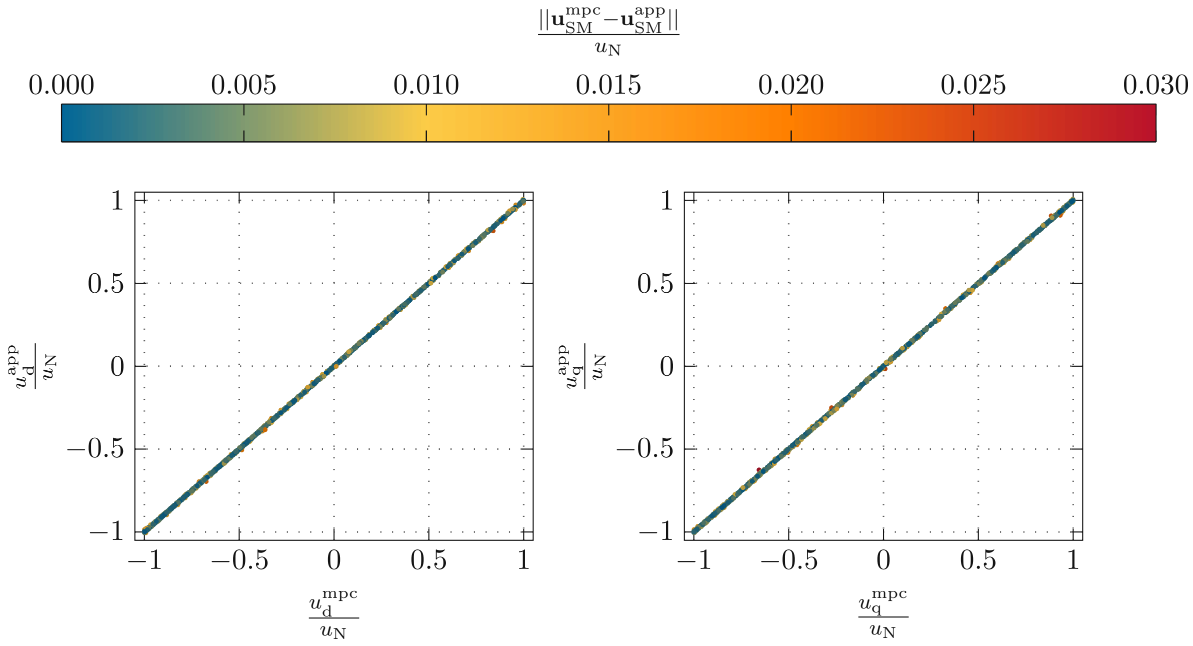
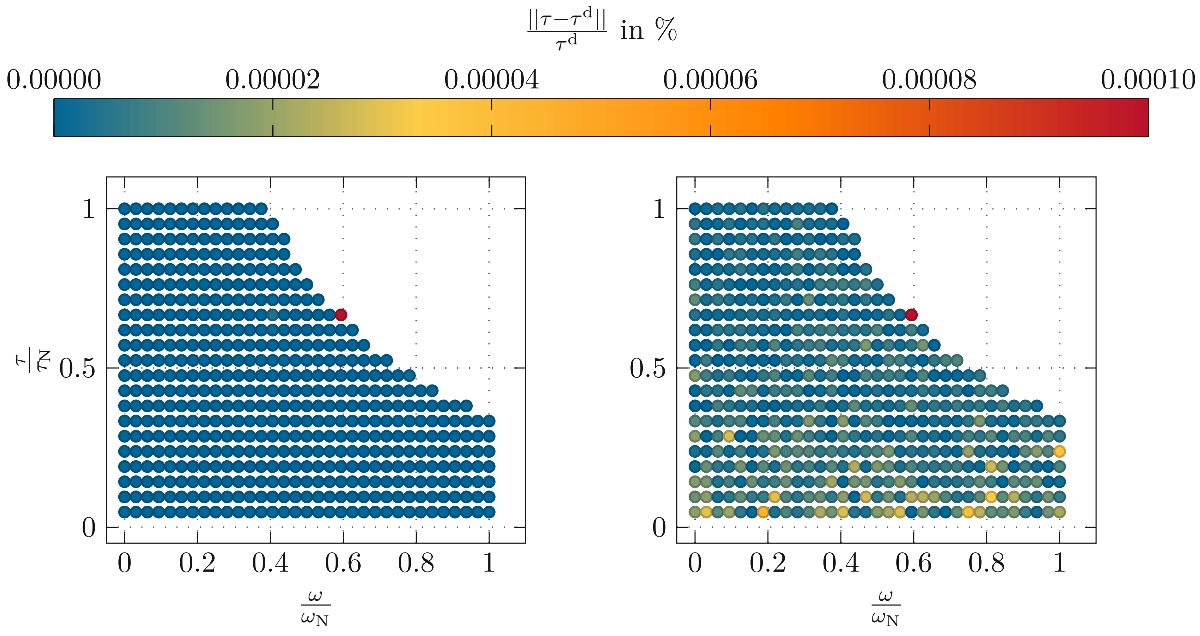


Figure 3.10: Q-Q plot for the approximation of the MPC without regularization.

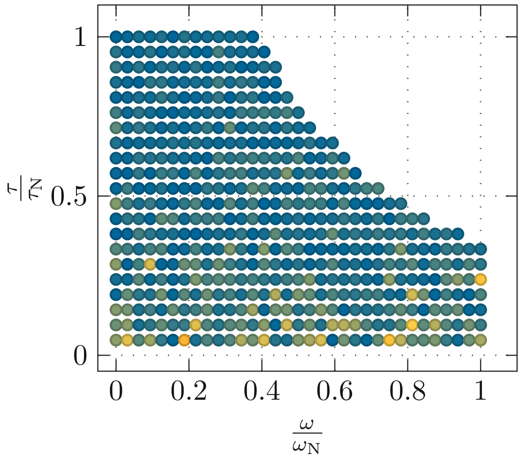
model and an angle independent machine model. The loop is herein closed either by the MPC or the approximation of the MPC. To identify the influence of parameter deviations, the simulations were carried out for the nominal parameter set and a parameter set, where the direct flux of the machine was increased by 10% modeling a parameter mismatch between the simulated LUTs in the FEM and the true machine behavior or a temperature error in the thermal model.

3.6.1 Stationary simulations

Figure 3.11 shows the results of the approximation on the torque accuracy for a steady-state torque speed map. In these simulations, different combinations of desired torques and speeds over the whole operating region are set and the stationary results of the comparison on these points are compared to the MPC results and the setpoint values. The simulations are performed without the influence of measurement noise since it is investigated in the following section. The figure shows, that the relative error is smaller than 0.0001% normalized by the nominal torque τ_{N} over the whole operating region for the nominal system. This means, that steady-state accuracy can be achieved with the here proposed approximation of the MPC and that the approximation is able to reach stationary endpoints. The reason for this is, that the integrator integrates the error until stationary accuracy is achieved. It can be observed, that the steady-state accuracy is slightly worse for the approximation. However, the error still remains very small and this behavior does not influence the overall steady-state performance of the control algorithm. These plots look very similar for all three approximation approaches. Therefore only this



(a) Error between MPC and setpoint. (b) Error between approximation and setpoint.



(c) Error between approximation and MPC.

Figure 3.11: Absolute relative error for stationary simulations without parameter deviations for the MPC without regularization.

plot is shown.

3.6.2 Transient simulations

MPC without regularization terms

A dynamic setpoint profile including steep (almost steplike) ramps, sinusoidal signals at different torque levels and a slower ramp is simulated. Figure 3.12 to Figure 3.17 show the results of the dynamic simulations for this dataset. For each MPC there exist two plots- one plot shows the overall transient for the given input signal and the other plot shows a close-up of the first sinusoidal signal. White noise is applied to the currents to show the noise surpressing capabilities of the developed approaches. The setpoint torque, the torque controlled by the MPC and the torque controlled by the MPC approximation are shown in the first subfigure. The absolute error, scaled by the maximum achievable torque are shown in the second subfigure. The approximation error between the MPC and its approximation is shown in the third subfigure. The last two subfigures show the voltages and the currents for the MPC and the approximation with their respective boundaries.
All plots show, that the approximation accuracy of the approach is rather high, with only some marginal exceptions. The robustness against noise of the two MPCs with regularization is clearly visible in Figure 3.14 and Figure 3.16, especially at the steady state points. However, the control error of the MPC is significantly higher than the approximation error in all the scenarios. For the MPCs with the regularization terms, the approximation behavior itself could be improved. This can be explained by the fact, that the MPCs with regularization terms might be badly tuned. It seems like the weights on the regularization terms are chosen too high. This is especially well shown in the close-up sinusoidal signal at high speeds without parameter deviations when comparing the MPC results without regularization and the MPC results with regularization term on the state in the first subplot in Figure 3.13c and Figure 3.15c. The voltages for the regularized MPCs, shown for example in the last subplot in Figure 3.15a and Figure 3.17a, are much smoother than the voltage for the MPC without regularization shown in Figure 3.13a. However, this leads to a slower dynamical behavior and for this fast change rate of the sinus, this causes inaccuracies in the control. Nevertheless, the results of the approximation are in good alignment with results of the MPC itself. The voltage and the current boundary are respected at all times and the approximation error is pretty low compared to the error of the MPC itself. For these dynamic changes, the parameter deviation on the direct flux $\tilde{\Psi}_\mathrm{d} = 1.1\Psi_\mathrm{d}$ seems to have an influence on the approximation accuracy, especially for the case of the MPC with regularization on the input shown in Figure 3.17b and Figure 3.17d. The results for the parameter deviation show here a much bigger deviation than for the simulations for the nominal case. The reason for this is, that for the MPC with regularization on the state and the input, the grid for the integrator voltages is chosen pretty coarse in order to save computation time. By refining this grid, a higher approximation accuracy is achievable also for the case of parameter deviations. However, these results show overall, that even for high deviations from the nominal value for the direct flux Ψ_d, the approach works still well in most of the cases. Furthermore, it is pretty impressive, that the MPC with regularization on the input and the state approximates

the MPC in such a fine manner, especially for the nominal case, given the limited size of the dataset.

To quantify the prediction performance of the MPC, the root mean squared error, the mean absolute error and the maximum absolute error between the MPC without regularization terms, its approximation and the setpoint values for the simulations are exemplary summarized in Table 3.2 for the simulation without parameter deviations at $0.15\omega_\mathrm{N}$. This table shows, that the approximation is following the MPC closely also in dynamic setpoint changes, and that the approximation error is by an order of magnitude smaller than the control error of the MPC itself.

error	RMSE	MAE	Max
$\dfrac{\tau^\mathrm{mpc} - \tau^\mathrm{d}}{\tau_\mathrm{N}}$	0.0177	0.0104	0.0794
$\dfrac{\tau^\mathrm{app} - \tau^\mathrm{d}}{\tau_\mathrm{N}}$	0.0179	0.0106	0.0776
$\dfrac{\tau^\mathrm{mpc} - \tau^\mathrm{app}}{\tau_\mathrm{N}}$	0.0025	0.0019	0.0071

Table 3.2: Error measures for the approximation of the MPC without regularization terms.

In this chapter, the basic idea of the approximation was demonstrated for the case of current control using MPC for a PMSM. The next chapter discusses a practical example of a complex nonlinear MPC for the flux control of an IM and the feasibility of the developed approach is examined using this example.

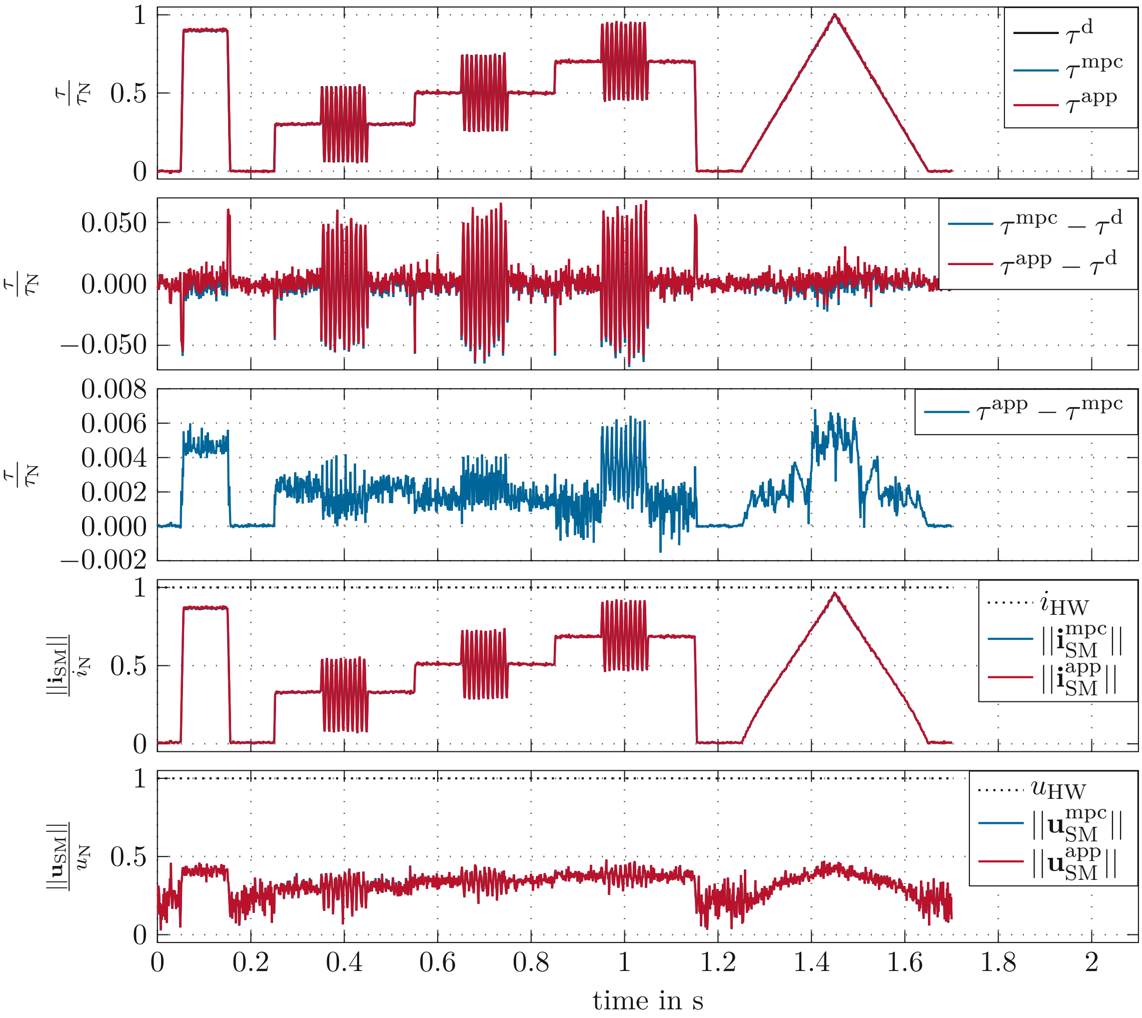
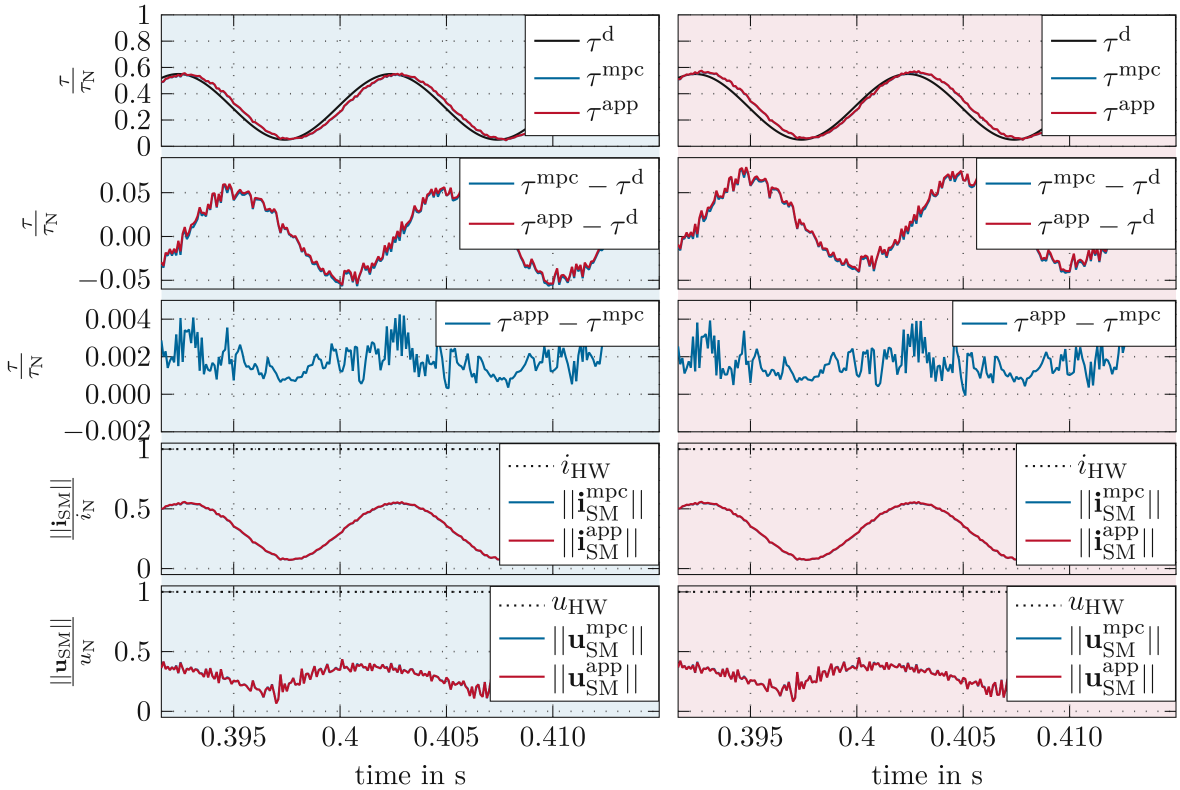
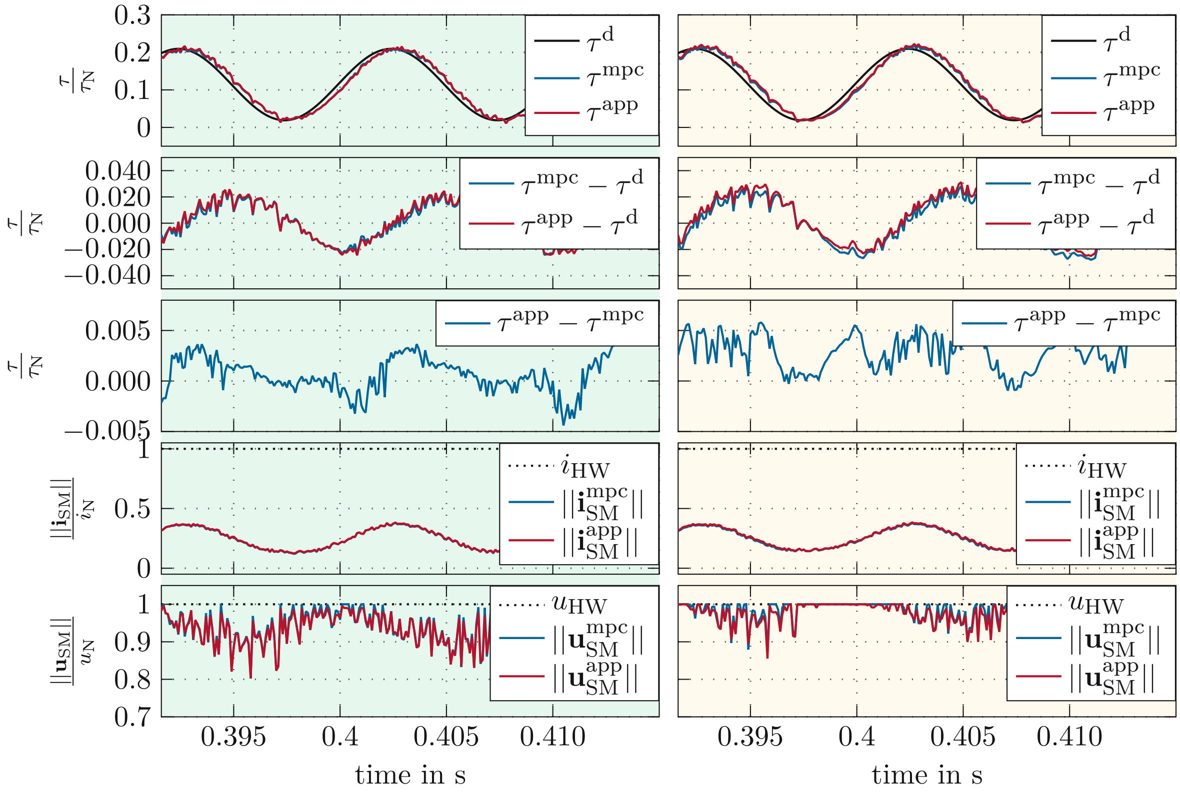


Figure 3.12: Transient simulations at $\omega = 0.15\omega_N$ without parameter deviations for MPC without regularization.



(a) $\omega = 0.15\omega_N$, no parameter deviations.

(b) $\omega = 0.15\omega_N$, $\tilde{\Psi}_d = 1.1\Psi_d$.



(c) $\omega = 0.85\omega_N$, no parameter deviations.

(d) $\omega = 0.85\omega_N$, $\tilde{\Psi}_d = 1.1\Psi_d$.

Figure 3.13: Zoomed detail of sine signal for different operating ranges of the MPC without regularization.

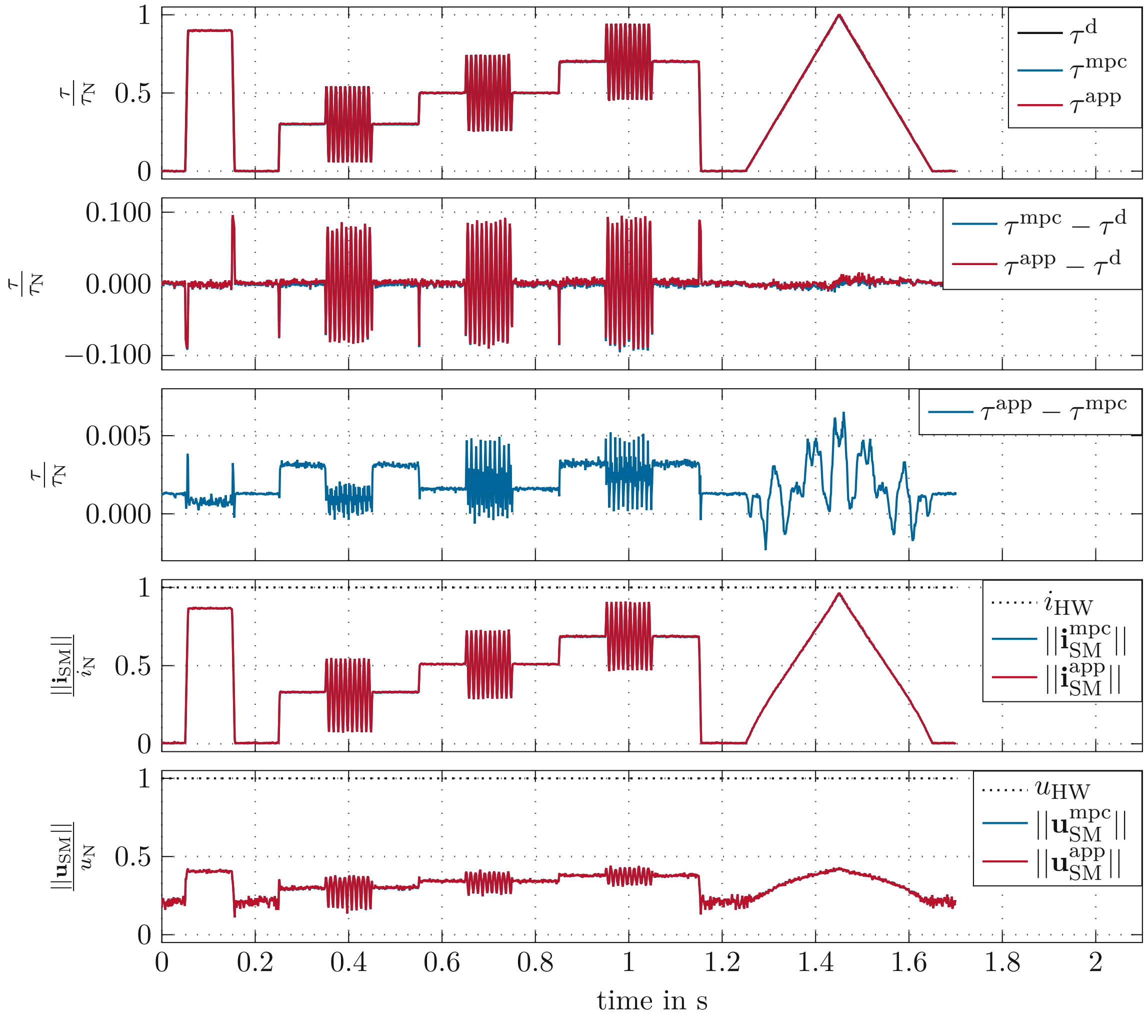
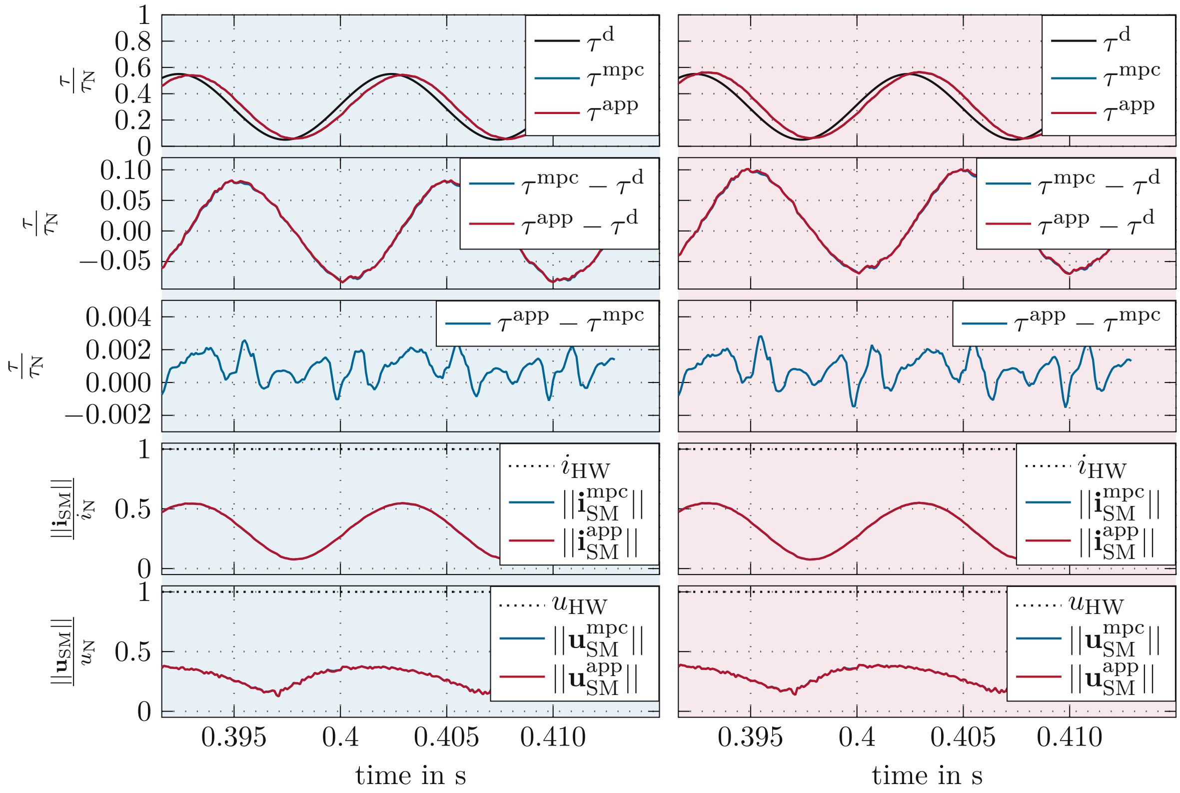
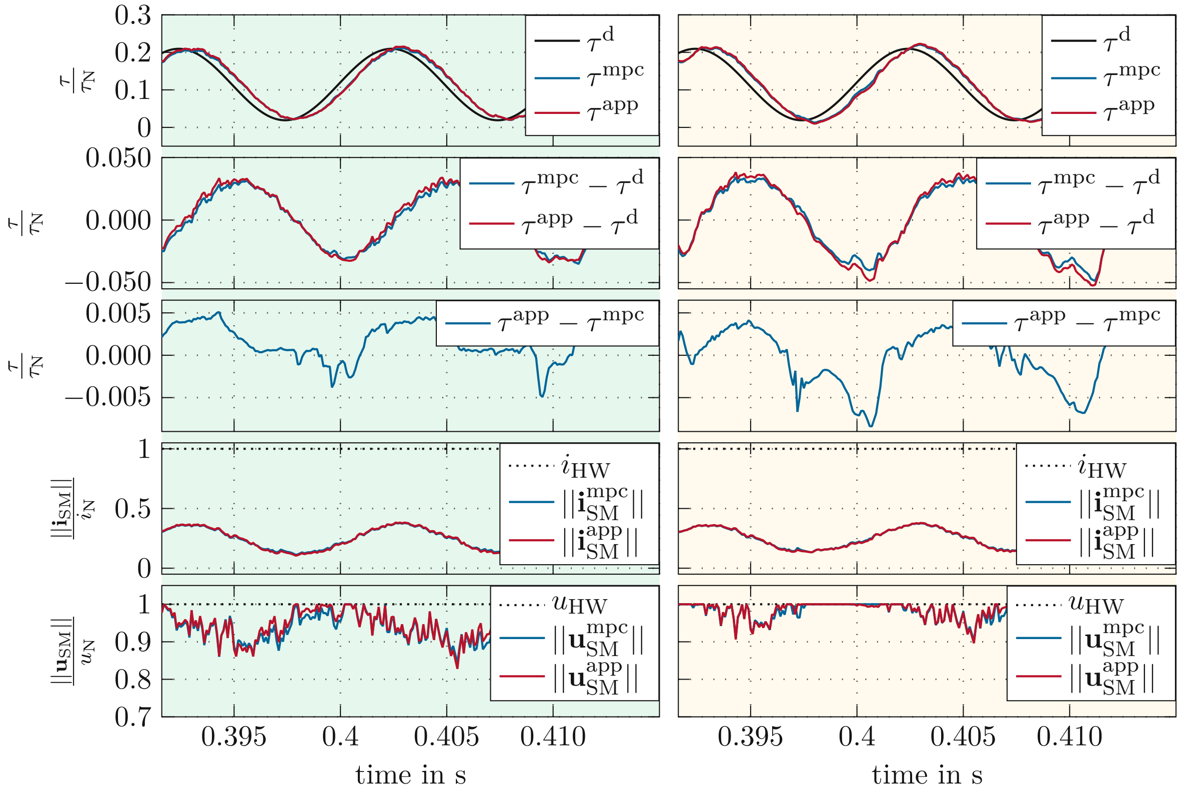


Figure 3.14: Transient simulations at $\omega = 0.15\omega_N$ without parameter deviations for MPC with regularization on state.



(a) $\omega = 0.15\omega_N$, no parameter deviations.

(b) $\omega = 0.15\omega_N$, $\tilde{\Psi}_d = 1.1\Psi_d$.



(c) $\omega = 0.85\omega_N$, no parameter deviations.

(d) $\omega = 0.85\omega_N$, $\tilde{\Psi}_d = 1.1\Psi_d$.

Figure 3.15: Zoomed detail of sine signal for different operating ranges of the MPC with regularization on state.

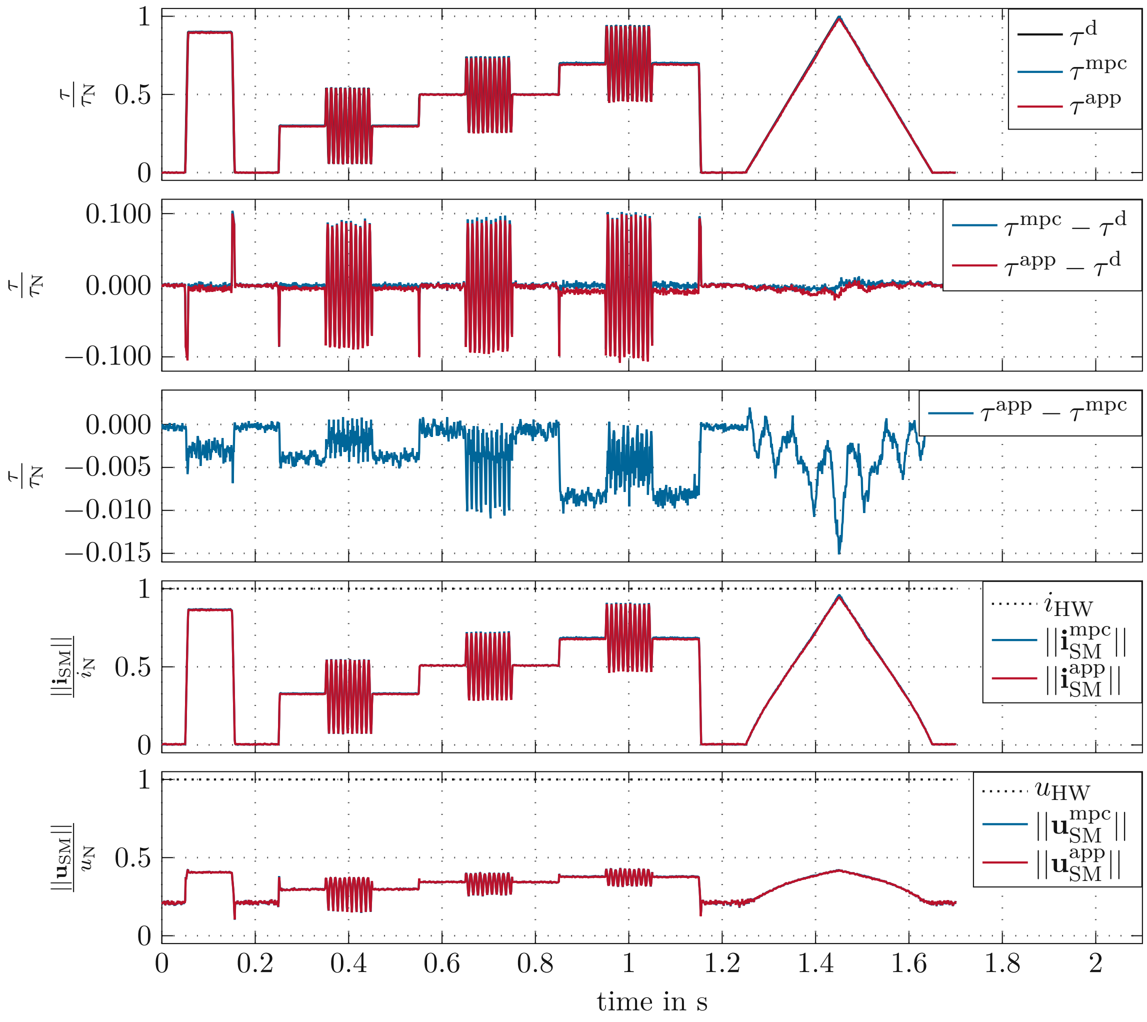
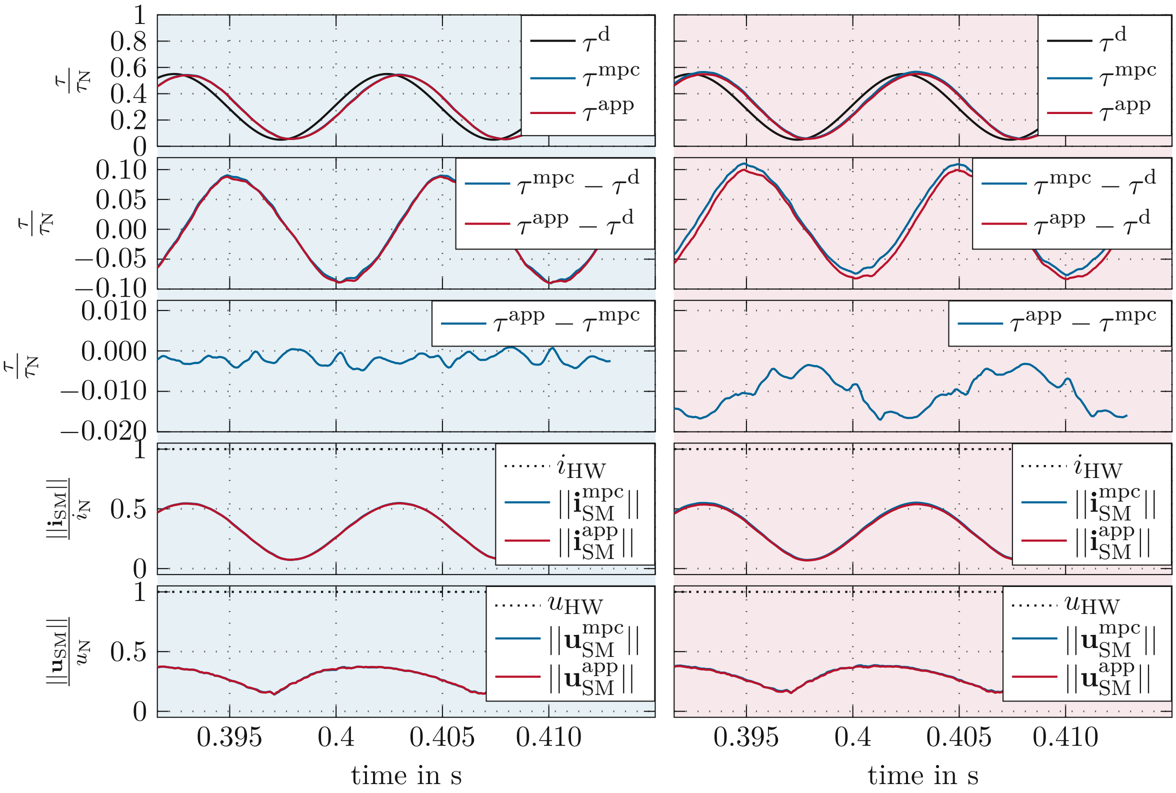
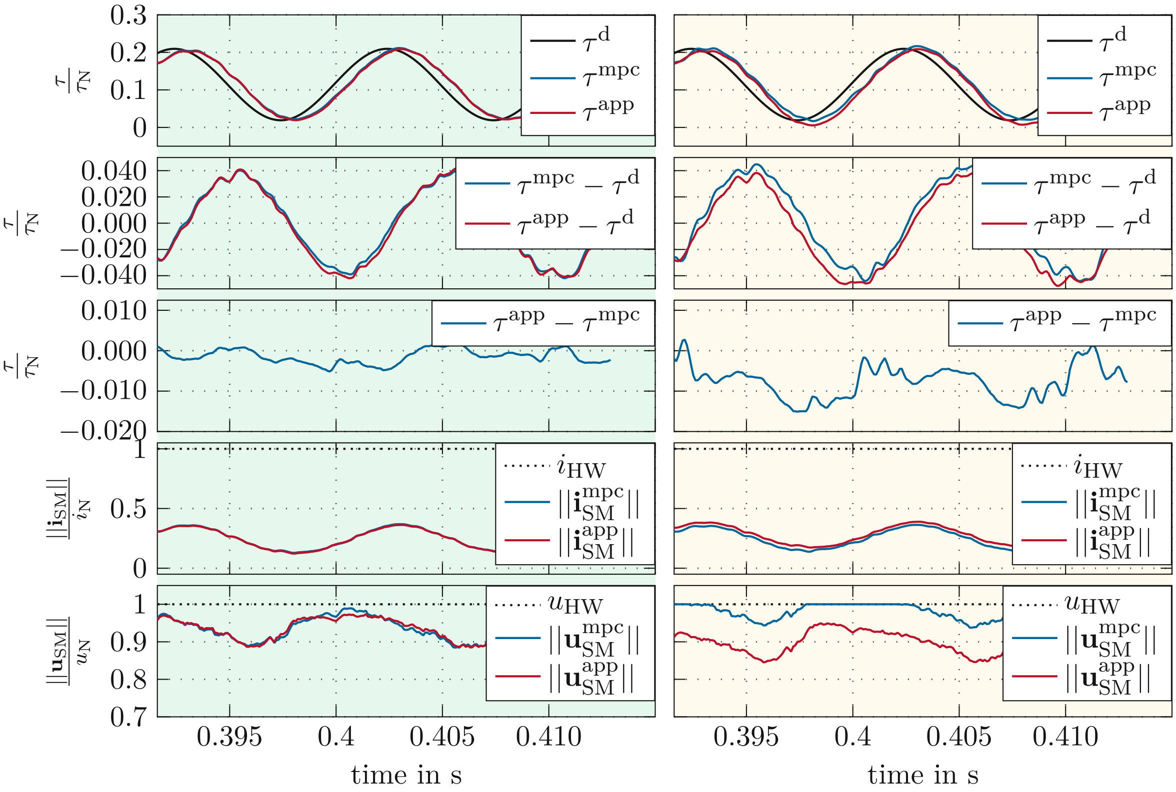


Figure 3.16: Transient simulations at $\omega = 0.15\omega_N$ without parameter deviations for MPC with regularization on state and input



(a) $\omega = 0.15\omega_N$, no parameter deviations.

(b) $\omega = 0.15\omega_N$, $\tilde{\Psi}_d = 1.1\Psi_d$.



(c) $\omega = 0.85\omega_N$, no parameter deviations.

(d) $\omega = 0.85\omega_N$, $\tilde{\Psi}_d = 1.1\Psi_d$.

Figure 3.17: Zoomed detail of sine signal for different operating ranges of the MPC with regularization on state and input.

4 MPC Approximation for IM

This chapter discusses the formulation and the approximation of an MPC for an induction machine. First, the investigated MPC is introduced. In a second step, the data generation and the approximation is explained in more detail and some simulation results are shown. Lastly a runtime analysis on a dSPACE Microlab Box real-time platform is performed.

4.1 Control loop structure

The control structure for the control of the IM is shown in Figure 4.1. It consists of a setpoint calculation, which computes the energy-optimal steady-state setpoints, an observer, which estimates the states and the not well known parameters of the system, the MPC itself, which acts as a dynamic trajectory generator and finally the flux state controller, which applies the voltages to the system. The usage of the subordinate flux controller is necessary due to the low sampling times in the control of electrical machines and the rather high computation time for the MPC. Therefore, the MPC and the steady-state optimal point calculation act in a slower task with a higher sampling time, while the observer and the flux controller run in a faster task. This is also indicated by the colors in Figure 4.1, where red indicates the higher sampling time of the signals and the blocks, and blue indicates the slower sampling time. The green background shows the parts of the control loop that are approximated.
In the following the setpoint calculation, the MPC and the state-space controller are introduced and analyzed on a high level, as these two are the most crucial parts for the approximation of the MPC. Furthermore the necessity of the flux controller is described. More details about the single components of the control structure are given in [18].

4.2 Setpoint calculation

The main task of the setpoint calculation consists of computing steady-state energy-optimal setpoints for the operation of the IM. It takes as inputs the desired torque τ^{d} and the electrical speed ω and outputs the energy-optimal setpoints $\mathbf{\Psi}_{\mathrm{IM}}^{\mathrm{s}} = [\Psi_{\mathrm{sd}}^{\mathrm{s}}, \Psi_{\mathrm{sq}}^{\mathrm{s}}, \Psi_{\mathrm{rd}}^{\mathrm{s}}]^{\mathrm{T}}$. These stationary optimal setpoints form three inputs for the MPC, since the MPC should be able to follow the generated setpoints under steady-state conditions.
For the steady-state setpoint calculation the main goal is to achieve a desired stationary torque τ^{d} while minimizing the power losses P_{l} given by (2.28). Furthermore, the stator current and stator voltage constraints

$$\left\|\mathbf{i}_{\mathrm{s,dq}}^{\mathrm{s}}\left(\mathbf{\Psi}_{\mathrm{s,dq}}^{\mathrm{s}}\right)\right\|_2 \leq i_{\mathrm{max}}^{\mathrm{s}} \tag{4.1a}$$

$$\left\|\mathbf{u}_{\mathrm{s,dq}}^{\mathrm{s}}\left(\mathbf{\Psi}_{\mathrm{s,dq}}^{\mathrm{s}}\right)\right\|_2 \leq u_{\mathrm{max}}^{\mathrm{s}} \tag{4.1b}$$

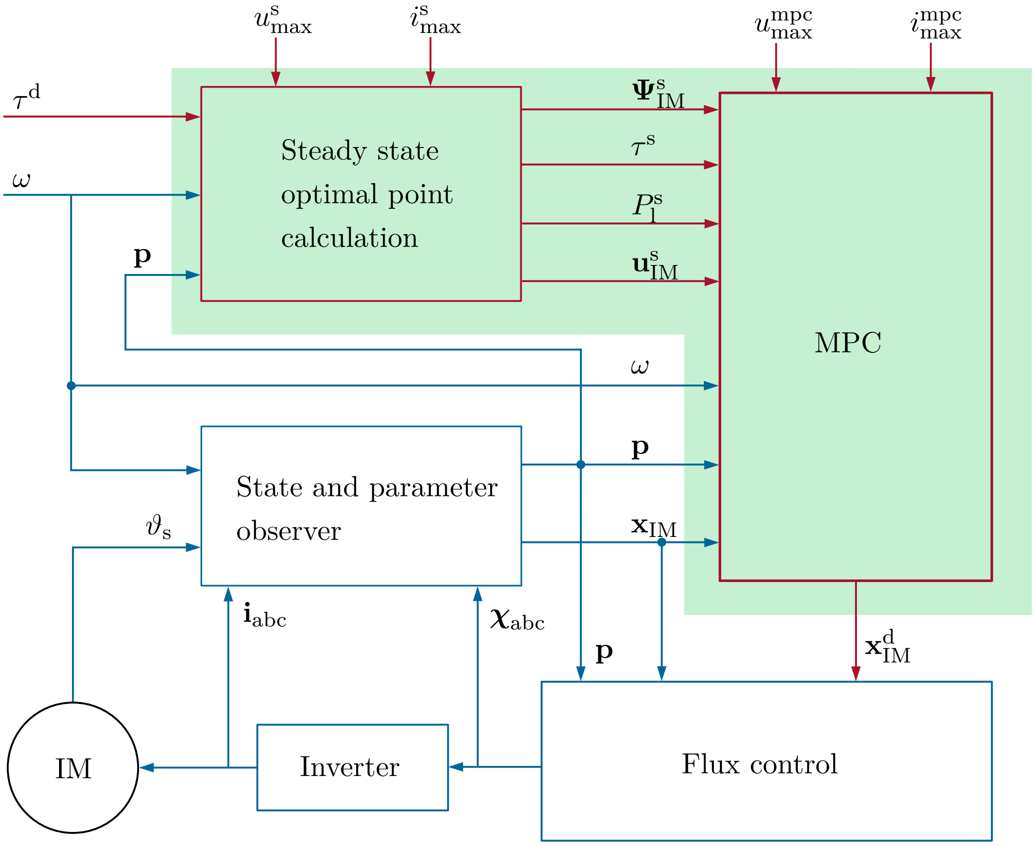


Figure 4.1: Block diagram of the control structure of the IM.

with the steady-state stator current vector $\mathbf{i}^{\mathrm{s}}_{\mathrm{s,dq}} = [i^{\mathrm{s}}_{\mathrm{sd}}, i^{\mathrm{s}}_{\mathrm{sq}}]^{\mathrm{T}}$, the steady-state stator flux vector $\boldsymbol{\Psi}^{\mathrm{s}}_{\mathrm{s,dq}} = [\Psi^{\mathrm{s}}_{\mathrm{sd}}, \Psi^{\mathrm{s}}_{\mathrm{sq}}]^{\mathrm{T}}$, the steady-state stator voltage vector $\mathbf{u}^{\mathrm{s}}_{\mathrm{s,dq}} = [u^{\mathrm{s}}_{\mathrm{sd}}, u^{\mathrm{s}}_{\mathrm{sq}}]^{\mathrm{T}}$ and the respective current and voltage limits $i^{\mathrm{s}}_{\max}$ and $u^{\mathrm{s}}_{\max}$ must be met at all times. The relationship between the flux linkages and the steady state voltages and currents is given in (2.29) and (2.30). In case that the desired torque is not reachable due to the constraints, the maximum achievable torque should be achieved (MTPA).

To compute the optimal steady state setpoints, a nonlinear constrained optimization problem has to be solved in real time. Since the solution of this nonlinear optimization problem is computationally demanding, [18] describes a more efficient method to solve this optimization problem by splitting it up in smaller sub-problems, which makes the solution computationally more efficient.

4.3 MPC formulation

Based on the steady-state setpoints from Section 4.2, the main goal of this MPC is to derive dynamic trajectories that enable an optimal dynamic operation of the motor. In contrast to the MPC from Section 3.3, this MPC does not directly output the control action itself, but acts as a dynamic trajectory generator for the subordinate flux controller.

These trajectories should ensure two factors- on the one hand a loss-optimal operation of the motor in stationary or quasi-stationary operating points and on the other hand guarantee that during high dynamic setpoint changes the system constraints are met at any time. During these high dynamic setpoint changes the loss optimality plays therefore a subordinate role. For this purpose, the cost functional must be designed properly in order to enable this operation [18].
The input variables for the MPC are the optimal steady-state setpoints $\Psi^{\mathrm{s}}_{\mathrm{sd},m}$, $\Psi^{\mathrm{s}}_{\mathrm{sq},m}$ and $\Psi^{\mathrm{s}}_{\mathrm{rd},m}$, the steady-state voltages $u^{\mathrm{s}}_{\mathrm{sd},m}$ and $u^{\mathrm{s}}_{\mathrm{sq},m}$ and the stationary torque τ^{s} defined at the time $t = mT_{\mathrm{mpc}}$ with the sampling time T_{mpc} of the MPC. The prediction horizon of the MPC is discretized in $N_{\mathrm{ph}} - 1$ steps with non-uniform time intervals $\Delta T_{\mathrm{ph},j}$ with a time increment over the horizon of $\Delta T_{\mathrm{ph},j} = T_{\mathrm{mpc}} \cdot 1.151^{j-1}$. The usage of a non-uniform time grid achieves a longer prediction horizon, since the size of the time intervals is growing with increasing j while still obtaining the solution for the next time step directly as the length of the first time interval is T_{mpc}. For example, for a prediction horizon of $N_{\mathrm{ph}} = 5$ timesteps this would yield a prediction length of $6.755T_{\mathrm{mpc}}$, which is a 35% increase of the horizon length compared to a uniform spaced sampling grid with T_{mpc}. Similar to Section 3.3 the nomenclature $(*)_{m|j}$ is used in the formulation of the MPC, which indicates the predicted quantity on the prediction horizon at time $t = mT_{\mathrm{mpc}} + T_{\mathrm{ph},j}$, with $T_{\mathrm{ph},j} = \sum_{\mathrm{k}=1}^{j-1} \Delta T_{\mathrm{ph},k}$ [18].
To achieve the loss-optimal operation of the MPC, deviations from the stationary optimal values for the rotor flux setpoint $\Psi^{\mathrm{s}}_{\mathrm{rd}}$, the torque τ^{s} and the power losses $P^{\mathrm{s}}_{\mathrm{l}}$ are penalized in the cost function. Furthermore, the trajectories of the stator currents should be as smooth as possible. This results in the cost function of the MPC [18]

$$
\begin{aligned}
J(\mathbf{w}_{\mathrm{m}}) = \sum_{j=1}^{N_{\mathrm{ph}}} W_{\tau}(\tau(\mathbf{x}_{\mathrm{IM},m|j}) - \tau^{\mathrm{s}}_{m})^2 + W_{\mathrm{P}}(P_{\mathrm{l}}(\mathbf{x}_{\mathrm{IM},m|j}) - P^{\mathrm{s}}_{\mathrm{l},m})^2 + \\
+ W_{\Psi}(\Psi_{\mathrm{rd}}(\mathbf{x}_{\mathrm{IM},m|j}) - \Psi^{\mathrm{s}}_{\mathrm{rd},m})^2 + W_{\Delta}\left\|\mathbf{i}_{\mathrm{s},m|j}(\mathbf{x}_{\mathrm{IM},m|j}) - \mathbf{i}_{\mathrm{s},m|j-1}(\mathbf{x}_{\mathrm{IM},m|j-1})\right\|_2^2,
\end{aligned}
\tag{4.2}
$$

with the state vector $\mathbf{x}_{\mathrm{IM},m|j} = [\Psi_{\mathrm{sd},m|j}, \Psi_{\mathrm{sq},m|j}, \Psi_{\mathrm{rd},m|j}]^{\mathrm{T}}$ and the (positive) optimization weights W_{τ}, W_{P}, W_{Ψ} and W_{Δ}. The prediction horizon and the optimization weights are the tuning factors of the MPC. By increasing the prediction horizon, the trajectory is predicted over a longer time and has thus a higher capability of dealing with system boundaries. However, with an increasing prediction horizon, the number of optimization variables increases as well, which in turn increases the computational effort and decreases the potential of real time capability. The weights on the other hand are important to define the main optimization goal of the cost function. By changing these, more focus can be given to the fast torque dynamics or on robust operation at high efficiency. Therefore, it is important to find a good trade-off between the optimization goals by tuning these parameters in simulations and on the testbench [18].
The optimization variables are summarized in

$$
\mathbf{w}_{\mathrm{m}}^{\mathrm{T}} = [\mathbf{u}^{\mathrm{T}}_{\mathrm{IM},m|1}, \mathbf{x}^{\mathrm{T}}_{\mathrm{IM},m|2}, \mathbf{u}^{\mathrm{T}}_{\mathrm{IM},m|2}, \ldots, \mathbf{x}^{\mathrm{T}}_{\mathrm{IM},m|N_{ph}-1}, \mathbf{u}^{\mathrm{T}}_{\mathrm{IM},m|N_{ph}-1}, \mathbf{x}^{\mathrm{T}}_{\mathrm{IM},m|N_{ph}}] \tag{4.3}
$$

with the voltage vector $\mathbf{u}_{\mathrm{IM},m|j} = [u_{\mathrm{sd},m|j}, u_{\mathrm{sq},m|j}]^{\mathrm{T}}$. To predict the state evolution over the prediction horizon into the future, (2.25) is discretized by an implicit Euler method.

The current and the voltage constraints are finally given as

$$h_{\mathrm{v},m|j}(\mathbf{u}_{\mathrm{IM},m|j}) = \left\|\mathbf{u}_{\mathrm{IM},m|j}\right\| - u_{\max}^{\mathrm{mpc}} \leq 0 \tag{4.4a}$$

$$h_{\mathrm{i},m|j}(\mathbf{x}_{\mathrm{IM},m|j}) = \left\|\mathbf{i}_{\mathrm{IM},m|j}\right\| - i_{\max}^{\mathrm{mpc}} \leq 0 \ , \tag{4.4b}$$

with the respective maximum current $i_{\max}^{\mathrm{mpc}}$ and the maximum voltage $u_{\max}^{\mathrm{mpc}}$ of the MPC. These limits can be set different from the limits for the steady state setpoint calculation in (4.1). By setting the current limit $i_{\max}^{\mathrm{mpc}}$ to higher values than $i_{\max}^{\mathrm{s}}$, the short time thermal overload capability of the machine can be utilized in order to achieve higher dynamical behavior [18]. However, for stationary points the maximum current of the MPC must be consistent with the maximum steady-state current to meet the thermal limits of the motor. Furthermore, $u_{\max}^{\mathrm{mpc}}$ should be set to a slightly higher value than $u_{\max}^{\mathrm{s}}$ to allow the MPC and the subordinate controller to compensate for model-plant mismatch [18].
By combining (4.2) and (4.4) and adding the implicit Euler prediction, the optimization problem for the MPC is formulated as [18]

$$\mathbf{w}_m^* = \underset{\mathbf{w}_m}{\operatorname{argmin}} \, J(\mathbf{w}_m) \tag{4.5a}$$

$$\text{s.t.} \quad \mathbf{x}_{\mathrm{IM},m|j+1} = \mathbf{F}_{\mathrm{IM},m|j+1} \ , \quad j = 1, ..., N_{\mathrm{ph}} - 1 \tag{4.5b}$$

$$h_{\mathrm{v},m|j}(\mathbf{u}_{\mathrm{IM},m|j}) \leq 0 \ , \quad j = 1, ..., N_{\mathrm{ph}} - 1 \tag{4.5c}$$

$$h_{\mathrm{i},m|j}(\mathbf{x}_{\mathrm{IM},m|j}) \leq 0 \ , \quad j = 2, ..., N_{\mathrm{ph}} \quad , \tag{4.5d}$$

where $\mathbf{F}_{\mathrm{IM},m|j+1}$ stands for the implicit Euler step [18].
This discretization approach of the optimal control problem of the MPC is known in the literature as direct method. By using this method, the infinite-dimensional optimal control problem is discretized on a specific timegrid and solved by using a numerical solver which can deal with constrained optimization problems. The chosen discretization approach is known as full discretization, since the optimization vector $\mathbf{w}_m$ consists of the state variables $\mathbf{x}_{\mathrm{IM},m|j}$ and the input variables $\mathbf{u}_{\mathrm{IM},m|j}$, both of which are discretized [33]. Using the discretized system dynamics using the implicit Euler, the optimal states could be computed by using the initial point and the input vector only– this approach is known as subordinate time integration [34]. The usage of the full discretization results in a higher dimensional optimization vector. However, for this small dimensional optimization problem the chosen approach is fast to compute and delivers directly the discretized state space vectors together with the voltages [33].
To approximately compensate for the calculation time of the MPC, the first state $x_{\mathrm{IM},m|1}$ is computed by the optimal stator voltage of the last optimization step $u_{\mathrm{IM},m|j-1}$ and the actual (estimated) state $\mathbf{x}_{\mathrm{IM},m}$. To solve the optimization problem in real time, it is reformulated and approximated by a quadratic program. This enables to reach very fast computation times. For more details regarding the implementation, see [18].

4.4 Flux controller

The MPC in the previous section computes optimal trajectories for the dynamic operation of the IM. It optimizes the flux linkages $\mathbf{\Psi}_{\mathrm{IM}}^{\mathrm{d}}$ and the stator voltages $\mathbf{u}_{\mathrm{IM}}$, which are the

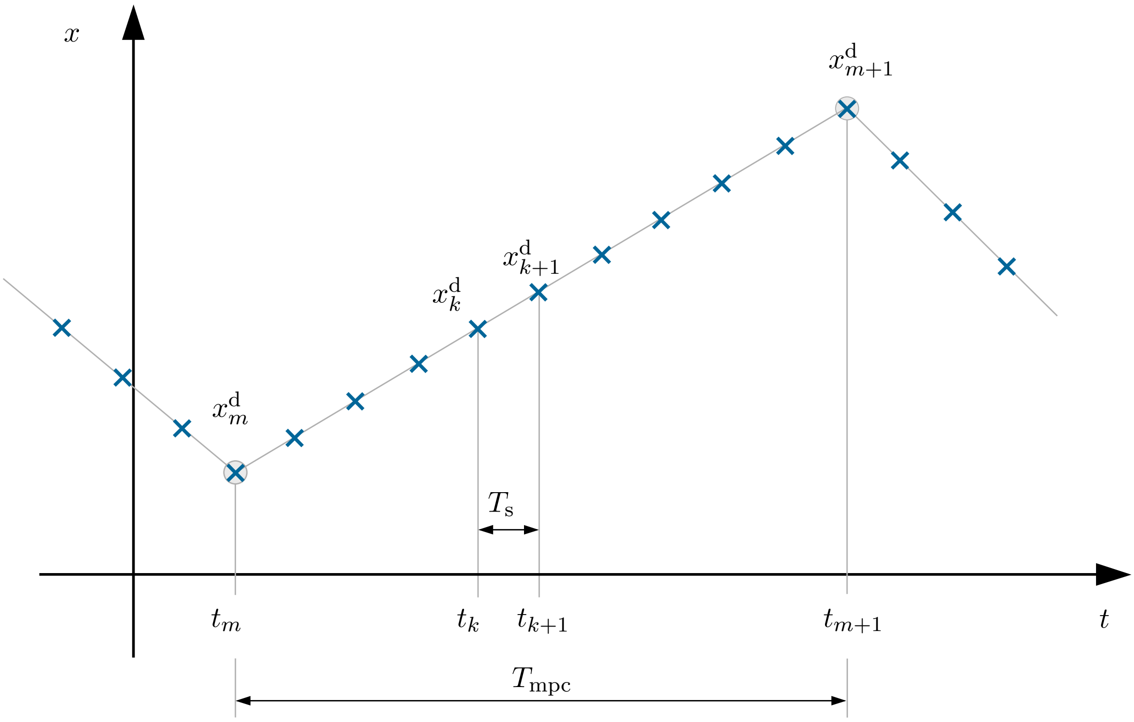


Figure 4.2: Sampling grid for the MPC and the subordinate flux controller.

inputs for the inverter. This computation is performed at the sampling time of the MPC T_{mpc}, which is bigger than the sampling time of the control loop T_{s}. The reason for this is the high computational effort for the evaluation of the optimization problem of the MPC. Therefore, a subordinate state control scheme is utilized, which runs at the fast sampling time T_{s}. Figure 4.2 shows the timescale for the MPC updates (labeled with the subscript m) and the timescale on which the flux controller runs (labeled with the subscript k). Since the state space controller runs in this short sampling time, the trajectory between the desired points $\mathbf{x}_{\mathrm{IM},m}^{\mathrm{d}}$ and $\mathbf{x}_{\mathrm{IM},m+1}^{\mathrm{d}}$ must be interpolated onto the finer grid $\mathbf{x}_{k+1}^{\mathrm{d}}$ to get the next desired setpoint. Herein k stands for the current timestep for the flux controller inside the timegrid of the MPC [18].

This flux controller consists of a feedforward and a feedback part. More details on the implementation is found in [18]. The main goal of the flux controller is to follow the given interpolated trajectory as fast and as smooth as possible. Furthermore, the noise behavior of the control loop is strongly influenced by the tuning of this controller. As the MPC acts as trajectory planning algorithm and does not apply the optimized voltages directly, the controller itself is mainly responsible for the noise suppression. Since the goal of this work is to approximate the setpoint calculation and the MPC, but does not replace the flux controller, the simulations in Section 4.7 are performed without noise, as this noise would mainly arise from the controller performance and tuning.

4.5 Data generation

As already discussed in Section 3.4, simulations have to be performed that cover the whole operation space of the IM. The block diagram of the control loop in Figure 4.1 shows, that ten input variables are necessary to evaluate the MPC in an arbitrary operating point. These variables can be divided into the following categories:

- Three variables define the current state of the machine $\mathbf{x}_{\mathrm{IM},m} = [\Psi_{\mathrm{sd},m}, \Psi_{\mathrm{sq},m}, \Psi_{\mathrm{rd},m}]^{\mathrm{T}}$, which come from the observer.
- Three variables define the computed optimal steady state setpoints of the machine $\mathbf{x}^{\mathrm{s}}_{\mathrm{IM},m} = [\Psi^{\mathrm{s}}_{\mathrm{sd},m}, \Psi^{\mathrm{s}}_{\mathrm{sq},m}, \Psi^{\mathrm{s}}_{\mathrm{rd},m}]^{\mathrm{T}}$, which come from the steady state optimal setpoint calculation.
- Three variables define the varying parameters of the machine $\mathbf{p} = [R_{\mathrm{s}}, R_{\mathrm{r}}, L_{\mu}]^{\mathrm{T}}$, which can either change during operation because of temperature changes (stator and rotor resistance) or are not accurately known in the modeling phase (the saturation of the machine) and come thus from the state space observer.
- One variable defines the current electrical speed of the machine ω, which comes from the resolver measurement in the operation of the electrical machine.

The other three outputs of the steady-state optimal point calculation τ^{s}, $P^{\mathrm{s}}_{\mathrm{l}}$ and $\mathbf{u}^{\mathrm{s}}_{\mathrm{IM}}$ are dependent on the steady-state flux vector $\boldsymbol{\Psi}^{\mathrm{s}}_{\mathrm{IM}}$ and therefore do not represent extra independent inputs of the system. This input space must be resolved in high resolution in order to achieve high approximation accuracy. However, as discussed already in Section 3.4, this input space is very high dimensional and thus the curse of dimensionality is even more prominent than in the case of the PMSM.
Since the approximation of this input space and the performance of the necessary simulations would need a lot of computational effort, the system boundary of the approximation was changed by including the steady state optimal point calculation into the approximation. This approach decreases the input dimension, because the three optimal steady state setpoints can be computed by the desired torque of the machine τ^{d} and the current electrical speed ω. Since the electrical speed is also already an input for the MPC, the input dimension for the simulations can be reduced by two. This further reduces the computational effort for the evaluation of this approach, since the steady state optimal point calculation is now also part of the approximation and does not need to be evaluated separately in each time step during the operation. This approach finally yields

$$\boldsymbol{\zeta}_{\mathrm{mpc,IM}} = [\Psi_{\mathrm{sd}}, \Psi_{\mathrm{sq}}, \Psi_{\mathrm{rd}}, \tau^{\mathrm{d}}, \omega, R_{\mathrm{s}}, R_{\mathrm{r}}, L_{\mu}]^{\mathrm{T}} \tag{4.6}$$

with the input vector $\boldsymbol{\xi}_{\mathrm{mpc,IM}} \in \mathbb{R}^{8}$ of the simulation and the approximation. The parameter ranges for the variables in the input vector are summarized in Table 4.1. The three flux inputs are herein set to cover the whole operating range of the electrical machine. For this purpose, multiple dynamic simulations (e.g. torque ramps with different torque rates between different positive and negative values, sinusoidal torque trajectories, speed ramps with different speed rates,...) were performed on the MPC and the maximum and

minimum values of the states was captured. The torque range corresponds to the whole operating range of the electrical machine, such that the machine can be operated in motoric and generatoric points. The speed range corresponds to the forward driving operating range of the machine and since the machine is symmetrical, its behavior for positive and negative speeds is the same. Therefore, only simulations for positive speeds are performed. For the machine parameters, values were used from testbench measurements with the MPC. In these measurements the stator resistance value was never around its nominal value at 20° because the machine was operated at a different stator temperature, which is the main influencing factor for the stator resistance. This parameter range could be extended easily by changing the minimum and maximum values for the resistances and the inductance. However, for a proof of concept and a comparison with the measurement results of the electrical machine, these parameter ranges are sufficient.
To achieve a high approximation accuracy for this approach, the eight input variables must be sampled on fine grids. Therefore, the fluxes are sampled with a resolution of $0.025\Psi_{\mathrm{N}}$, the torque with a resolution of $0.025\tau_{\mathrm{N}}$ and the speed with a resolution of $0.0625\omega_{\mathrm{N}}$. For the parameters, 3 uniformly sampled points for the stator resistance, 4 uniformly sampled points for the rotor resistance and 5 uniformly sampled points for the main inductance are used. This results in a total number of

Parameter	Minimum	Maximum
Ψ_{sd}	$-0.25\Psi_{\mathrm{N}}$	Ψ_{N}
Ψ_{sq}	$-0.5\Psi_{\mathrm{N}}$	$0.5\Psi_{\mathrm{N}}$
Ψ_{rd}	$0.03\Psi_{\mathrm{N}}$	Ψ_{N}
τ^{d}	$-\tau_{\mathrm{N}}$	τ_{N}
ω	0	ω_{N}
R_{s}	$1.12R_{\mathrm{s},20}$	$1.22R_{\mathrm{s},20}$
R_{r}	$0.85R_{\mathrm{r},20}$	$1.48R_{\mathrm{r},20}$
L_{μ}	$0.74L_{\mu,0}$	$1.2L_{\mu,0}$

Table 4.1: Input parameter space for the IM.

$$n_{\Psi_{\mathrm{sd}}} \cdot n_{\Psi_{\mathrm{sq}}} \cdot n_{\Psi_{\mathrm{rd}}} \cdot n_{\tau} \cdot n_{\omega} \cdot n_{\mathrm{R_s}} \cdot n_{\mathrm{R_r}} \cdot n_{\mathrm{L}_\mu} = 51 \cdot 41 \cdot 39 \cdot 41 \cdot 17 \cdot 3 \cdot 4 \cdot 5 \approx 3.4 \cdot 10^9 \quad (4.7)$$

simulations. Herein $n_{\Psi_{\mathrm{sd}}}$, $n_{\Psi_{\mathrm{sq}}}$, $n_{\Psi_{\mathrm{rd}}}$, n_{τ}, n_{ω}, $n_{\mathrm{R_s}}$, $n_{\mathrm{R_r}}$ and n_{L_μ} describe the number of simulation points per parameter. The simulations are performed on the real time implementation of the MPC. The optimization is executed until it converges to the optimal points. By doing so, only fully converged points are used for the approximation itself. By assuming a computation time of approximately 5ms on average per simulation, this would correspond to an overall simulation time of approximately 195 days on a single core or approximately 25 days on 8 cores.
As discussed already in Section 3.4, this sampling is not feasible for the approximation. Therefore, some restrictions are used in the sampling to increase the efficiency and make

the simulation effort feasible. To begin, only points are considered, for which the torque according to (2.27) differs less than a factor of 1.5 of the maximum torque rate $\Delta\tau_{\mathrm{max}}$ times the sampling time of the MPC T_{mpc} from the desired torque τ^{d}

$$|\tau^{\mathrm{d}} - \tau(\Psi_{\mathrm{rd}}, \Psi_{\mathrm{sq}})| \leq 1.5\Delta\tau_{\mathrm{max}}T_{\mathrm{mpc}} \ . \tag{4.8}$$

This assumption holds as long as the control is able to follow the desired torque limited by the maximum torque rate with a deviation smaller than one and a half of the maximum torque rate multiplied by the sampling time. Since the sampling time of the MPC has a rather high value compared to the sampling time of the subordinate controller, the machine can follow the planned trajectory from the MPC quite well and thus this approach is feasible.

Secondly, only points are used in the sampling scheme, which do not violate the system boundaries. This means, that the stationary stator currents $\mathbf{i}^{\mathrm{s}}_{\mathrm{s,dq}}$ from (2.30) and the stationary stator voltages $\mathbf{u}^{\mathrm{s}}_{\mathrm{s,dq}}$ from (2.29) for the current simulation are limited by the current and the voltage boundary given by (4.1). This relationship is evaluated for each permutation of the uniformly sampled input grid and only the points that fulfill this relationship are kept.

By the application of these two assumptions on the dataset, the number of necessary simulations decreases from 3.4 billion to approximately $7.5 \cdot 10^7$ necessary simulations, which is by a factor of 45 smaller than the initial dataset would have been. Therefore, the simulations can be performed in the span of approximately 4.5 days on a single core or in approximately half a day on eight cores.

The evaluation of a training on this dataset, which is described in Section 4.7, showed, that the approximation worked well under dynamic and stationary scenarios, however, a large mismatch in the region around 0Nm was observed. The root cause for this behavior is, that the machine is not controllable, if the rotor direct flux gets to a value of 0Wb. The reason for this is, that in this scenario there is no field in the machine's rotor and thus no torque is generated by the stator field. In this condition, a rotor field is first built up by the control and only afterwards the machine is controlled to follow the desired torque trajectory. In this region, the IM is very sensitive to state changes. This effect was mainly observed in the predicted quadrature flux of the electrical machine. Therefore, the simulation grid is refined in this region by having a resolution of $0.0125\Psi_{\mathrm{N}}$ in the region between $-0.025\Psi_{\mathrm{N}}$ and $0.025\Psi_{\mathrm{N}}$ for the quadrature flux. Furthermore, extra desired torque points at the lowest torques are introduced in order to achieve a finer resolution and a better approximation accuracy. The torque grid is refined in the region between $-0.025\tau_{\mathrm{N}}$ and $0.025\tau_{\mathrm{N}}$ by adding six extra grid points at $\pm0.0175\tau_{\mathrm{N}}$,$\pm0.01\tau_{\mathrm{N}}$ and $\pm0.005\tau_{\mathrm{N}}$.

By using these extra points, the simulation effort increases again to $n_{\mathrm{IM}} = 1.1 \cdot 10^8$ simulations, which is by a factor of approximately 31 smaller than the initial dataset. These simulations correspond to a time of approximately 6.5 days on a single core or one day on eight cores.

This approach helps to perform the simulations and confirm the proof of concept in a fast and pragmatic way. However, for the application in a safety critical control regime all points should be simulated in a fine fashion and then be used for the training on a powerful hardware.

4.6 Data driven approach

The data driven approach to approximate the MPC for the IM is very similar to the approach described in Section 3.5. For the approximation, the *Python Programming Language* is also chosen here.

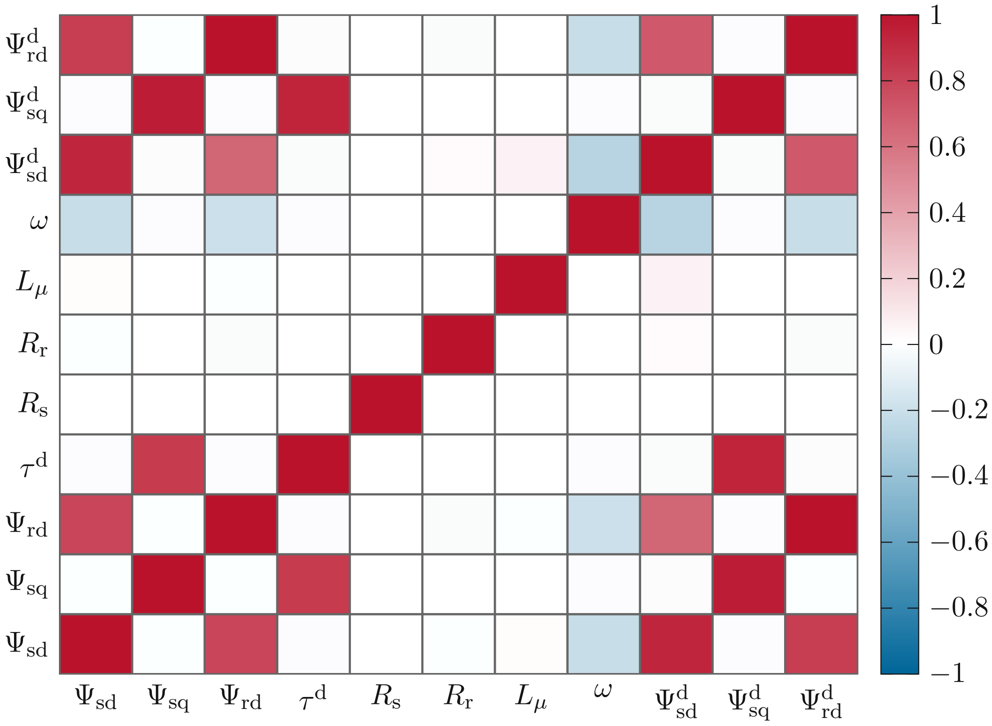


Figure 4.3: Correlation matrix for the input and output variables for the IM.

The correlation matrix between the individual inputs and outputs is shown in Figure 4.3. The correlation between the input states Ψ_{sd}, Ψ_{sq} and Ψ_{rd} and the output states of the MPC Ψ_{sd}^d, Ψ_{sq}^d and Ψ_{rd}^d show highly positive correlation coefficients, since the output of the MPC is strongly coupled to the current state. Furthermore, the cross-correlation between the stator direct flux Ψ_{sd} and the rotor direct flux Ψ_{rd} is high for the input state and the output state. This is because the rotor direct flux is built up by the stator direct flux according to (2.25c). For stationary points, the two flux values have identical values. The torque and the stator quadrature flux Ψ_{sd} correlate strongly as well, since the torque is directly proportional to the q-component of the stator flux, which can be changed by the input voltage in a fast manner because of the low time constant of the stator. The rotor d-component of the flux does not correlate as strongly linear with the torque, since the relationship herein is more complex, due to the high time constant of the rotor and the time that it takes to build up the field in the rotor. Furthermore, the speed correlates negatively with the direct component of the stator flux and the rotor flux. The reason for this is that, similar to the PMSM, the direct stator flux has to be reduced for highly dynamical setpoint changes, when the machine hits the voltage boundary. Since the rotor direct flux is strongly correlated with the stator direct flux, the correlation with the speed is high as well. The parameter values of the observer are not correlated with the other variables. For the stator and the rotor resistance, this behavior is explained by the fact that these parameters are mainly dependent on the temperature and not by the other quantities. The mutual inductance is not correlated in a linear fashion with the other parameters, since its saturation behavior is strongly nonlinear. However, this nonlinear behavior is captured by the neural network and the parameters are important for the control in the case of parameter mismatch and changing environmental conditions (e.g. when the rotor or the stator heats up).

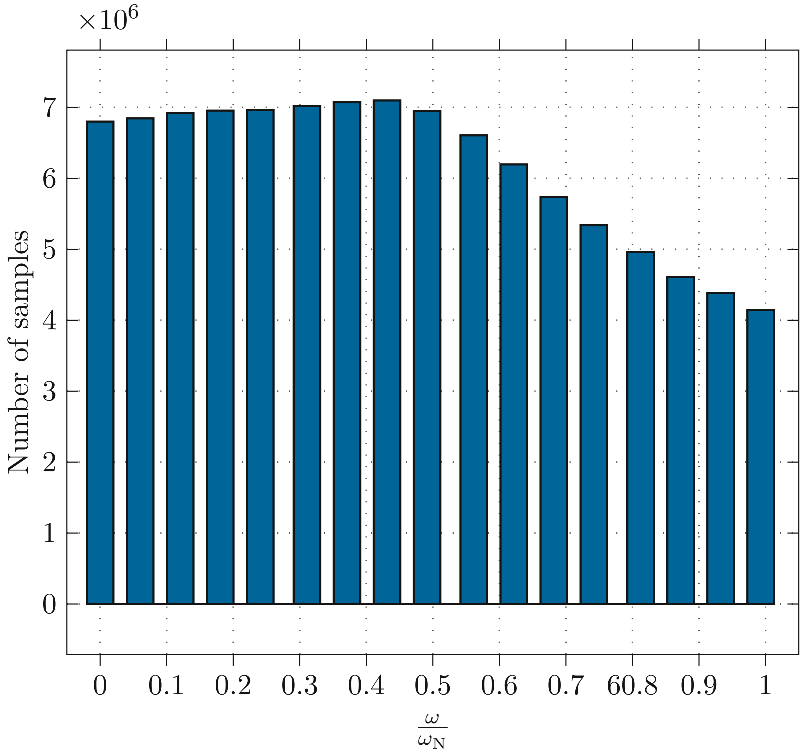
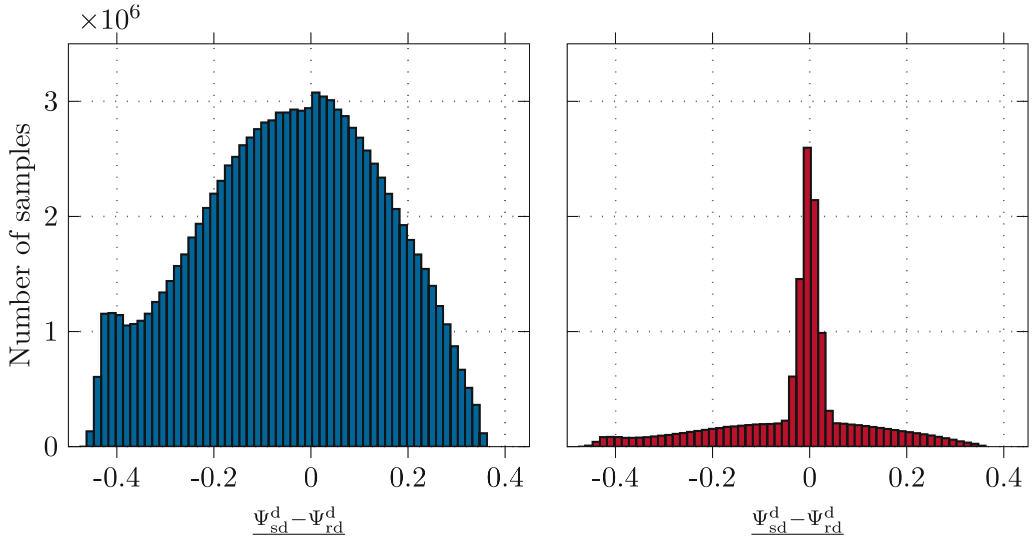


Figure 4.4: Histogram of the speed distribution in the original dataset.

The usage of the dataset described in Section 4.5 for the training and the prediction yields good results for dynamic setpoint changes, however, it performs rather poorly on stationary setpoints. This behavior is explained by the fact, that the dataset contains a lot of dynamic setpoint changes and is thus overfitting in this region. Furthermore, it performs better for smaller speeds than for high speeds. The reason for this is, that the dataset is limited by the voltage constraint of the system, which gets more restrictive for higher speeds. This behavior is shown in Figure 4.4. Therefore, another sampling step is performed on this dataset to counter these problems.

To mitigate the problem of the high speed approximation accuracy, the maximum number of sample points per speed $n_{\mathrm{max},\omega_\mathrm{N}}$ is limited by the number of simulation points at the highest speed ω_N. This maximum number of sample points is used in the next step as



(a) Uniform sampling. (b) Steady-state focused sampling.

Figure 4.5: Comparison of the sampling schemes.

an upper boundary of sampling points per speed interval. To mitigate the problem of the stationary accuracy, the difference $e_{\Psi_{\mathrm{d}}}$ between the stator direct flux and the rotor direct flux is computed and the number of samples with a specific error is displayed as a histogram in Figure 4.5a. This shows, that a large amount of the dataset is not close to zero, where the stationary setpoints of the induction machine lie ($\Psi^{\mathrm{s}}_{\mathrm{sd}} = \Psi^{\mathrm{s}}_{\mathrm{rd}}$). For this purpose, a threshold of 3% around the stationary point is applied and all points that lie in this region are used for the dataset (if this number is smaller than $n_{\mathrm{max},\omega_{\mathrm{N}}}$). These points should represent the stationary points in the dataset. Additionally, for each simulated speed from all the other simulation points with more than ±2% difference from the stationary point, 10% of $n_{\mathrm{max},\omega_{\mathrm{N}}}$ is used to randomly pick simulation points. This finally results in approximately twice as many points outside the 2% threshold compared to the points inside this threshold. These adapted sampling points are shown as a histogram in Figure 4.5b. By performing this, the number of points in the dataset decreases drastically by still achieving good results. This is a further advantage of this approach, since the neural network is able to be trained faster.

Similar to (4.9) the inputs and the outputs of the system that are approximated are summarized in

$$\mathbf{X}_{\mathrm{IM}} = \begin{bmatrix} \zeta^{\mathrm{T}}_{\mathrm{mpc,IM,1}} \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,IM,2}} \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,IM,3}} \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,IM,4}} \\ \vdots \\ \zeta^{\mathrm{T}}_{\mathrm{mpc,IM,n_{IM}}} \end{bmatrix}, \qquad \mathbf{Y}_{\mathrm{IM}} = \begin{bmatrix} \mathbf{\Psi}^{\mathrm{T}}_{\mathrm{IM,1}} \\ \mathbf{\Psi}^{\mathrm{T}}_{\mathrm{IM,2}} \\ \mathbf{\Psi}^{\mathrm{T}}_{\mathrm{IM,3}} \\ \mathbf{\Psi}^{\mathrm{T}}_{\mathrm{IM,4}} \\ \vdots \\ \mathbf{\Psi}^{\mathrm{T}}_{\mathrm{IM,n_{IM}}} \end{bmatrix}, \tag{4.9}$$

with the input matrix $\mathbf{X}_{\mathrm{IM}}$ and the output matrix $\mathbf{Y}_{\mathrm{IM}}$ of the neural network and the flux vector $\mathbf{\Psi}_{\mathrm{IM,j}} = [\Psi^{\mathrm{d}}_{\mathrm{sd,j}}, \Psi^{\mathrm{d}}_{\mathrm{sq,j}}, \Psi^{\mathrm{d}}_{\mathrm{rd,j}}]^{\mathrm{T}}$, which summarizes the three output states of the MPC. These matrices are split up in a training and a test set, where the training set consists of 85% of the data and the validation set consists of 15% of the data. These sets are shuffled in order to minimize the correlation between subsequent simulations.

The input scaler is the same minimum maximum scaler as described in (3.12) with 8 inputs. The fully connected neural network used for the approximation consists of an eight dimensional input layer, four hidden layers with 70, 60, 50 and 50 nodes respectively and a three dimensional output layer. The activation function of the hidden layers is set to ReLU and for the output layer to linear. All the other parameters of the neural net are set equivalent to the net in Section 3.5. The shape of the hidden layers is chosen after an ablation study is performed. In this ablation study, the size of the neural network is reduced in width and depth to save computational effort. To perform the ablation study, first a high dimensional neural network with 300, 150, 75 and 50 nodes is trained. After the training of this network, it creates the baseline (BL) for the ablation study. Subsequently, the number of hidden layers and the width of the hidden layers is varied and the performance of the networks is checked against the baseline network by using the Q-Q plot and a dynamic simulation environment, described in more detail in Section 4.7. Based on these simulations, the network size is reduced until the performance of the ablation study (AS) network is significantly worse than the baseline performance. This approach represents a tradeoff between approximation accuracy and computational effort. By doing so, the number of weights n_{w} and biases n_{b} is reduced from

$$n_{\mathrm{w,BL}} = 8 \cdot 300 + 300 \cdot 150 + 150 \cdot 75 + 75 \cdot 50 + 50 \cdot 3 = 62550 \tag{4.10a}$$

$$n_{\mathrm{b,BL}} = 300 + 150 + 75 + 50 + 3 = 578 \tag{4.10b}$$

to

$$n_{\mathrm{w,AS}} = 8 \cdot 70 + 70 \cdot 60 + 60 \cdot 50 + 50 \cdot 50 + 50 \cdot 3 = 10410 \tag{4.11a}$$

$$n_{\mathrm{b,AS}} = 70 + 60 + 50 + 50 + 3 = 233 \;, \tag{4.11b}$$

so by a factor of ≈ 6 for the weights and by a factor of ≈ 2.5 for the biases. This approach saves computation time, since less multiplication and addition operations have to be performed. Furthermore, it also saves memory on the hardware, since less parameters have to be saved. Both factors are crucial for the usage in the automotive control field, since the reduction of computation time enables the real time capability and the reduction of memory, which is crucial for the limited storage on the hardware.

The Q-Q plot shows the approximation accuracy of the neural network after the ablation study for each of the outputs. The approximation shows herein a high accuracy for all three output variables except for a few outliers. The root mean square error scaled by the nominal flux Ψ_{N} for the three approximations is equal to 0.0066 for the stator direct flux, 0.0047 for the stator quadrature flux and 0.0018 for the rotor direct flux. For the approximation, 97% of all datapoints for the stator direct flux, 97% of all datapoints for the stator quadrature flux and 99.5% of all datapoints for the rotor direct flux lie in the region of three times the standard deviation between prediction and actual values. This shows a high approximation accuracy for all three parameters over the whole operating range of the IM.

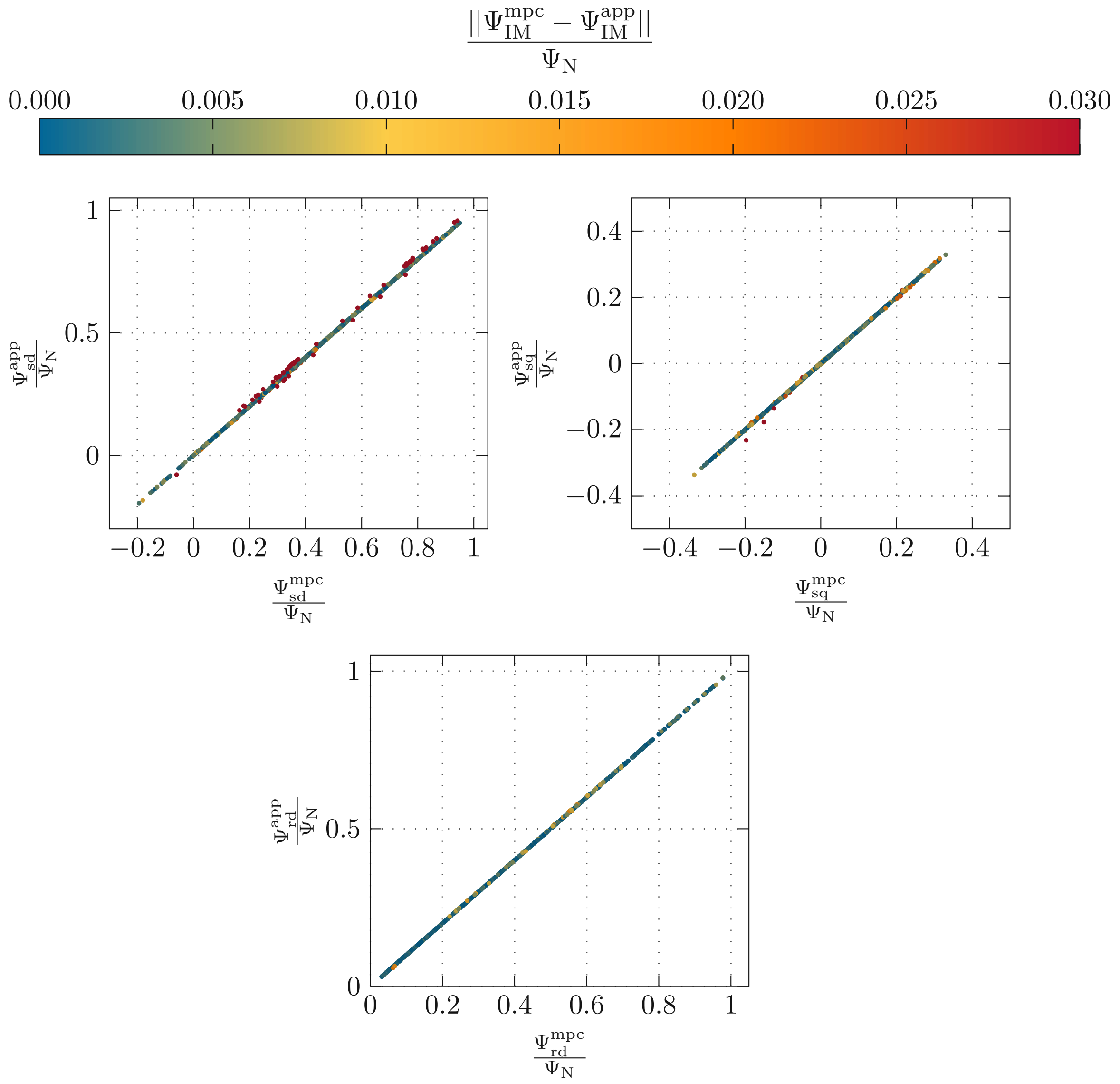


Figure 4.6: Q-Q plot for the approximation of the MPC for the IM.

4.7 Simulation results

Similar to Section 3.6, the developed approximation approach is checked on its feasibility for stationary and dynamic simulations. These simulations are closed-loop simulations with the observer, the flux state space controller, an inverter model and an angle independent machine model. To compare the two approaches, the loop is closed either by the MPC or by its approximation. Both control concepts are fed with the same input parameters and their outputs are compared. All shown simulations are performed at a constant temperature of 60°C. At this temperature, the stator and rotor resistance R_s, R_r are inside their ranges from Table 4.1 for their nominal and the disturbed values.

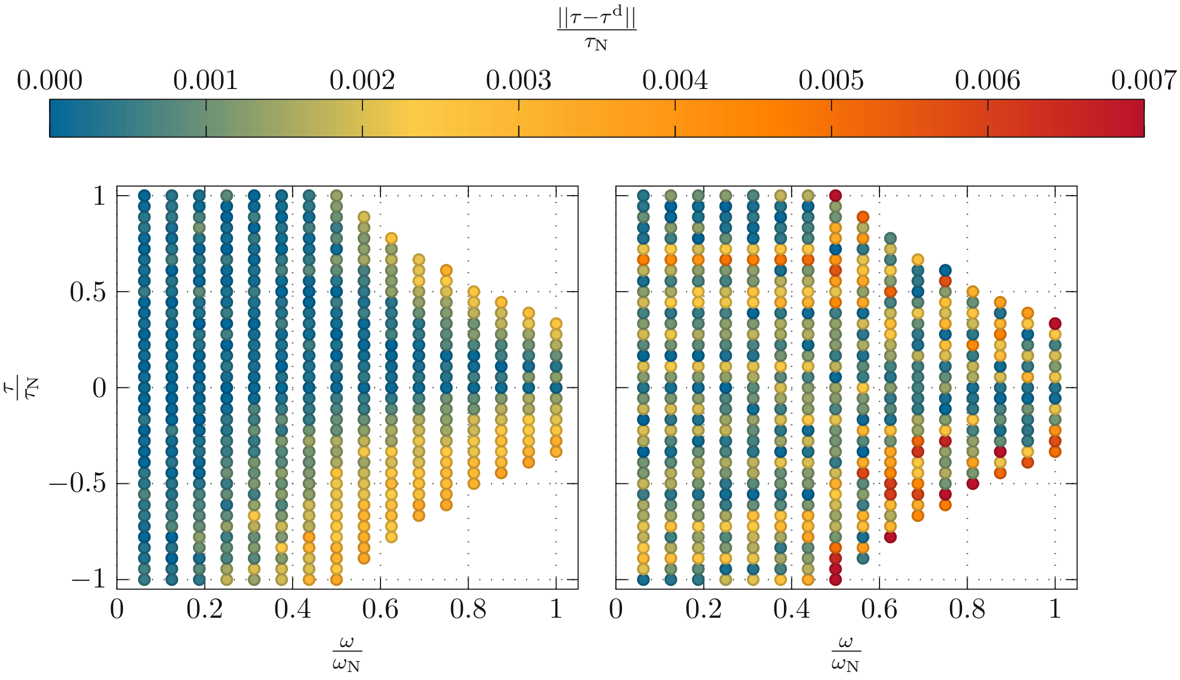
To identify the performance and the robustness of the newly developed concept, the following simulations are performed on the nominal dataset and a disturbed dataset, where the rotor resistance is increased by 30%. This error in the input parameter space of the approximation could come from a parameter mismatch because of component tolerances or from an error in the thermal model.

4.7.1 Stationary simulation results

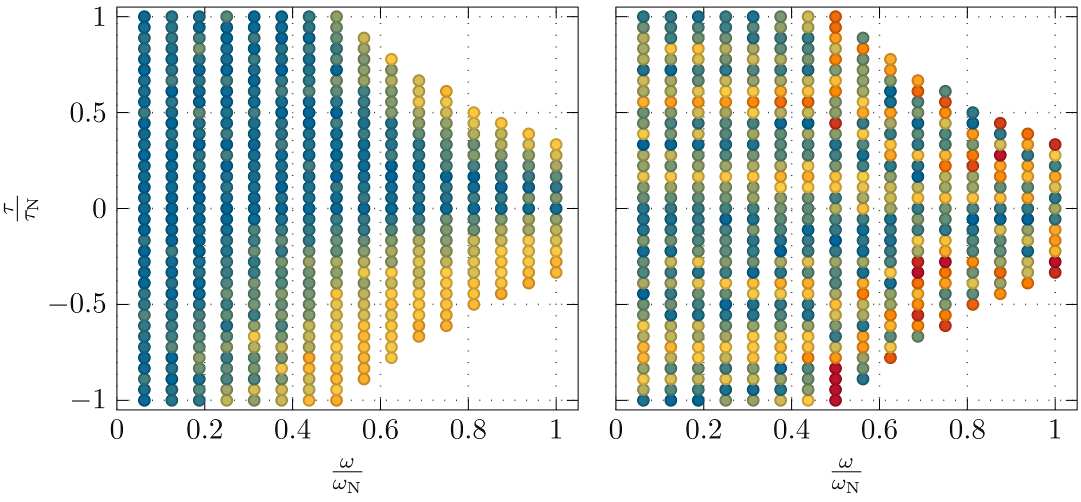
One main performance indicator of a control scheme is the stationary accuracy. For this purpose, Figure 4.7 and Figure 4.8 show the accuracy of the MPC and its approximation over steady-state torque speed maps for the nominal (with the blue background) and the disturbed (with the red background) case. It is important to mention, that compared to the stationary results in Figure 3.11, the mean absolute error is significantly higher. The reason for this is, that for this control loop, no integrator is used, which should achieve the stationary accuracy. However, this does not affect the evaluation of the performance of the approximation, since the main goal in this section is to compare the results between the control with the MPC and the control with its approximation.

Figure 4.7 shows the absolute error for all points in the torque speed map including 0Nm. This region is crucial for dynamic setpoint changes and yields the biggest error in the approximation of the MPC. However, for the stationary simulations this behavior is not observed. The error between the MPC and the desired torque value in Figure 4.7a and Figure 4.7c show low errors for the small speeds. The nominal system and the disturbed system with the increased rotor resistance yield similar results. This means, that the disturbance on this system parameter does not influence its performance. Especially in the motoric range the MPC yields good results. For higher speeds above the corner point and higher torque values (generatoric and motoric), the performance of the MPC on the stationary map falls off. The reason for this is, that with increasing speed the voltage boundary gets active, and this influences the performance of the MPC, since the machine is driven at the voltage boundary. Since this is a restriction on the input value of the dynamic system (the voltage $\mathbf{u}_\mathrm{IM}$), the trajectory generation and the solution of the optimization problem gets more complex. Therefore, bigger differences are expected here than in the base speed range. However, these absolute error values are still small compared to the maximum achievable torque of this machine.

Figure 4.7b and Figure 4.7d show the results for the approximation of the MPC. A similar behavior as for the MPC is found, since it performs better for smaller speeds than for

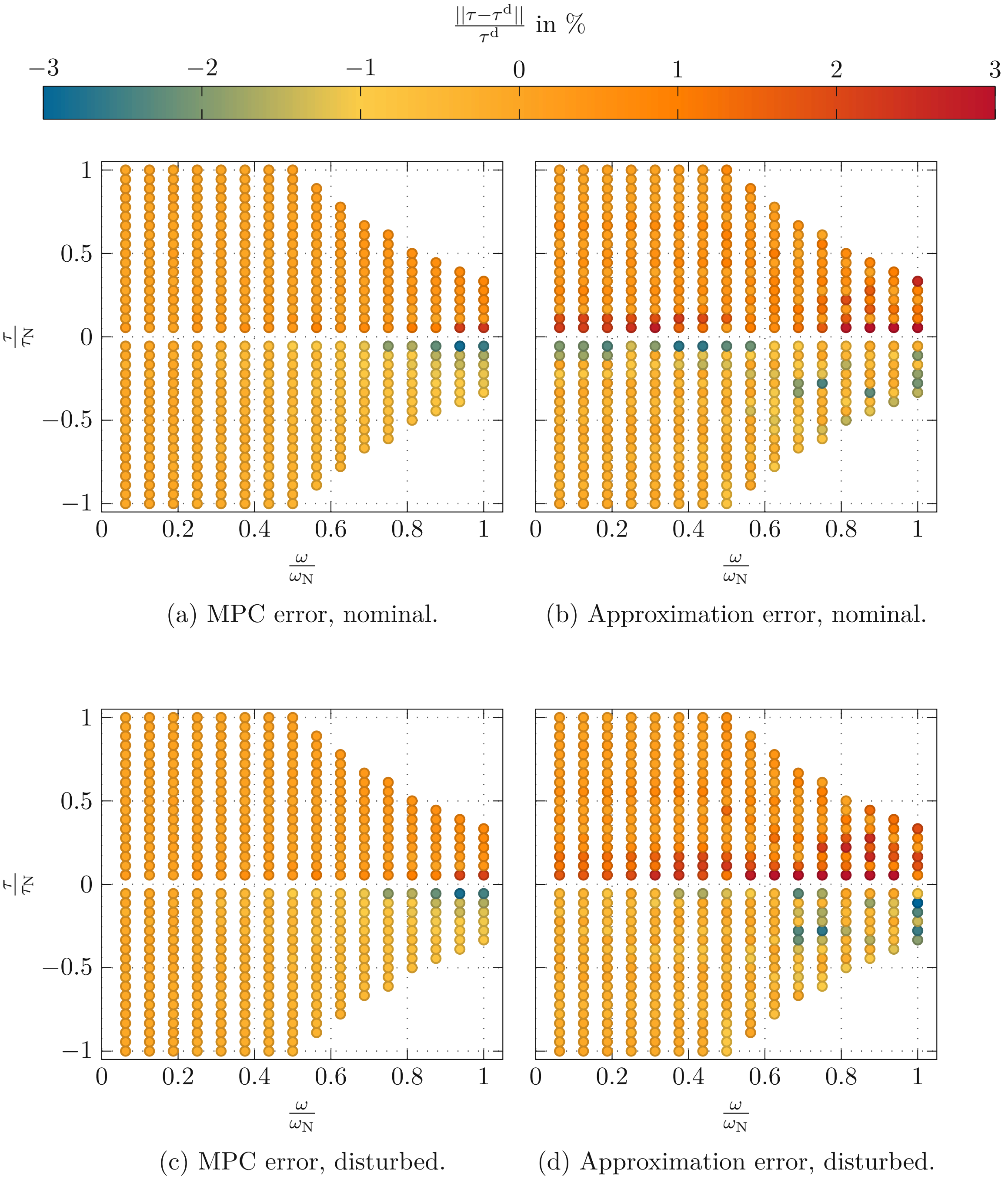


(a) MPC error, nominal. (b) Approximation error, nominal.



(c) MPC error, disturbed. (d) Approximation error, disturbed.

Figure 4.7: Absolute error for the stationary simulations for the IM.



(a) MPC error, nominal. (b) Approximation error, nominal.

(c) MPC error, disturbed. (d) Approximation error, disturbed.

Figure 4.8: Absolute relative error for stationary simulations for the IM.

higher speeds. The approximation also deals well with the parameter disturbance, since the two maps deliver similar results. The approximation behavior is not as smooth as for the MPC, since there are no clusters where many points have the same errors. This is explained by the nature of the approximation itself. Since the approximation does not know anything about the system that it should approximate, it tries to fit the data driven approach as well as possible to the datapoints. The evaluation on the input vector yields finally the interpolated behavior by the neural net of the training dataset. If there were error clusters in this map, this would mean, that there is some systematic error in the approximation which is not captured by the input parameters. Furthermore, the error is in general bigger than for the MPC itself. This is an expected behavior, since the approximation tries to approach the MPC as closely as possible, but it is still an approximation. However, the maximum error that is found in the torque speed maps is again small for the machine and shows a good result, taking the nonlinear behavior of the system into account. It is necessary to mention, that by increasing the dataset or by using a more complex neural network, the approximation accuracy can still be increased. However, for the usage on a target hardware, the neural net from Section 4.6 after the ablation study is used, since it is a good tradeoff between accuracy and computational performance.

To put the things into perspective of the currently driven torque, Figure 4.8 shows the relative errors of the two control approaches. For this comparison, the values around 0Nm are removed from the plot to compute the relative error for each point in the dataset. Compared to the absolute error, a different picture emerges, which shows very small relative errors for the approaches for high torques and high relative errors for small torques. This is explained by the fact, that small deviations from the setpoint cause for small torques bigger relative errors. The performance of the approximation shows also here good results with a relative error in the range of $\pm 3\%$. The approximation shows a similar error distribution as the MPC for high speeds. For low speeds and low torques, the approach shows the biggest differences. The reason for this is that the induction machine is difficult to control at these low torque values and the approximation has still problems in this region, even after the addition of the extra torque points in the sampling scheme in Section 4.5. To improve the performance, the parameter sensitivity in this region could be increased even further, which might lead to a higher approximation. Overall, these results show, that for the stationary case, the approximation performs very well in the whole operating range.

4.7.2 Dynamic simulation results

In order to evaluate the dynamic behavior of the proposed control concept, transient closed-loop simulations are performed. For this purpose, a dynamic profile including steps limited by the torque rate, sinusoidal signals at different torque levels and a slower torque ramp is used as input for the control loop. The result of this simulation is shown in Figure 4.9 for a low speed of $\omega = 0.15\omega_{\mathrm{N}}$. The first subplot shows herein the torque setpoint, the torque of the MPC and the torque of the approximation. The evaluation of the errors between the MPC, the approximation and the desired torque are shown in the second and the third subplot. The second subfigure shows in detail, that the error to the desired torque setpoint is small during steady-state and slowly changing operation, however it increases under fast dynamic operation. This behaviour is mainly caused by two factors: On the one hand, the control needs some time to settle during fast dynamical setpoint changes. On the other hand, the desired setpoint and the response of the system are not synchronized, since the evaluation of the control concept needs some time and is thus always time delayed with respect to the setpoint change. The third subplot shows, that the error of the approximation is by approximately a factor 10 smaller during the whole simulation. This means, that the approximation performs well and that the performance of the developed control scheme is mainly dependent on the underlying MPC that needs to be approximated. In the fourth and fifth subplot, the currents and the voltages are shown during these transients together with their boundary values. This plots show, that the approximation is able to fulfill the boundary conditions at low speeds. In the last subfigure the direct rotor flux, that the MPC and the approximation output as dynamic setpoint, is plotted. This shows, that the approximation and the MPC yield the same states as outputs of the system, since they are almost on top of each other.

In addition to this, a close-up view of the first sinusoidal signal at around six seconds is shown in Figure 4.10 for low and high speed and for the case of parameter deviations and no parameter deviations. For the case of parameter deviations, the rotor resistance R_r is increased by 30%. The first row shows the comparison for low speeds. This zoomed view shows, that the error of the approximation is very small. The error curves lie almost on top of each other and it makes no difference if the parameters are disturbed or not. The second row shows the results at high speeds. The overall error of the MPC decreases for both cases. However, the difference between the MPC and the approximation increases. The reason for this is, that the voltage constraint is active in this region. This makes the solution of the optimization problem more complex and also the approximation via the Neural Network gets more complex, since this introduces an extra nonlinear behavior into the system. Nevertheless, the resulting error between the approaches remains still small compared to the error of the error between the MPC and the desired setpoints. The voltage in the fourth plot shows, that for this operating range the system boundaries are met. The maximum stationary voltage is exceeded for this fast transient, since additional voltage helps in this dynamic scenarios to build the flux faster up. Anyhow, the voltage boundary is still met for this operating range. The fluxes are also again very close to each other with a small error between them. The non-smooth trajectory of the fluxes is explained by the voltage trajectory- since it is very difficult for a neural net to incorporate boundaries in a smooth way, the voltage is less smooth than for the MPC and this is directly also visible in the flux. On the overall torque approximation accuracy this effect plays however a minor role.

Figure 4.11 shows the behaviour on two additional transient signals at a speed of $\omega = 0.85\omega_N$ without any parameter deviations. Figure 4.11a shows the response on a very fast (almost steplike) ramp from a positive torque setpoint to a negative torque setpoint. The resulting torque profile shows some oscillations in the region around 0Nm, where the MPC has its biggest problems. However, for this torque ramp, the approximation can follow the behavior of the MPC very well with only little difference in the rotor direct flux. Figure 4.11b shows on the other hand a fast sinusoidal signal around 0Nm. Here the approximation around 0Nm shows bigger deviations from the MPC and the setpoint.

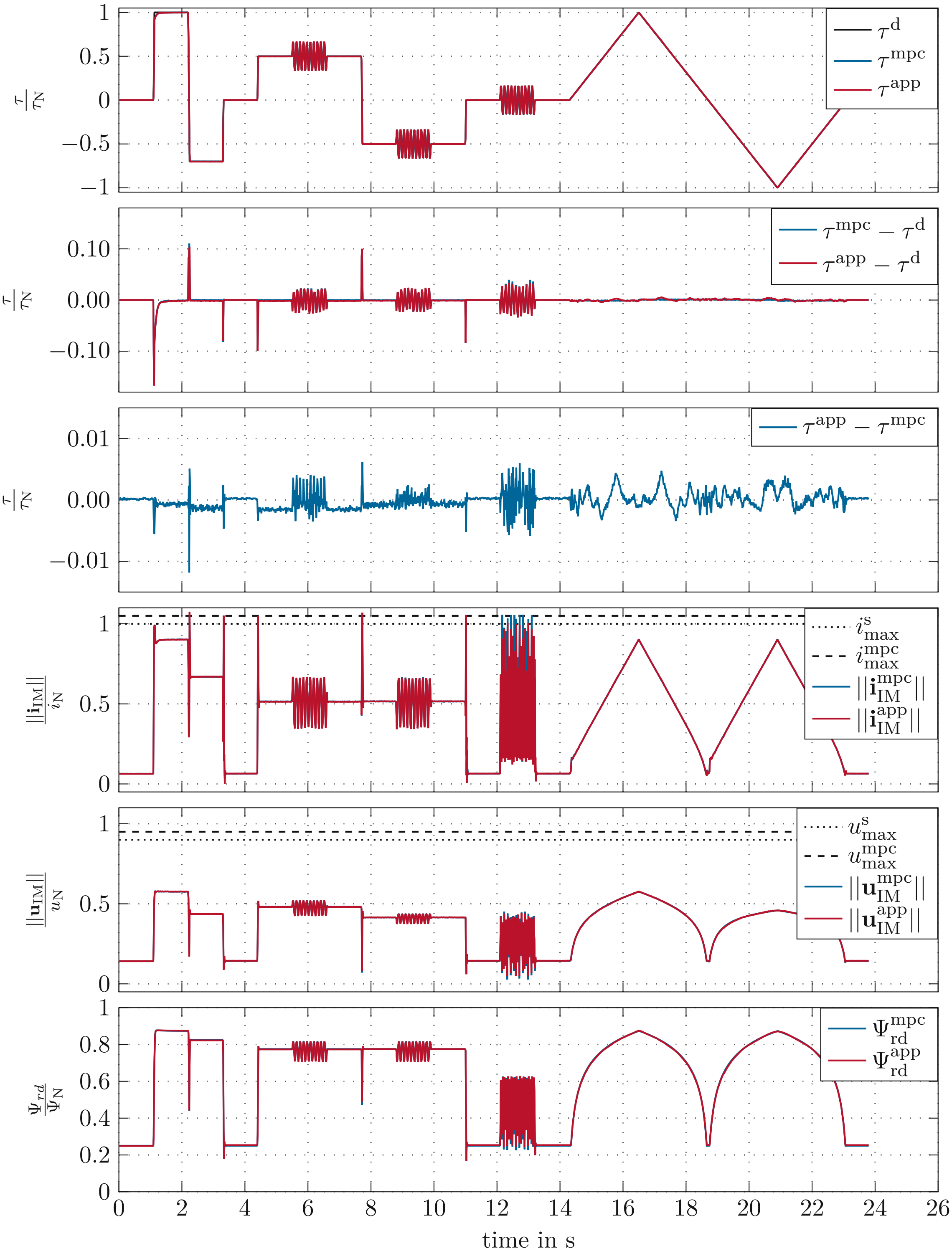
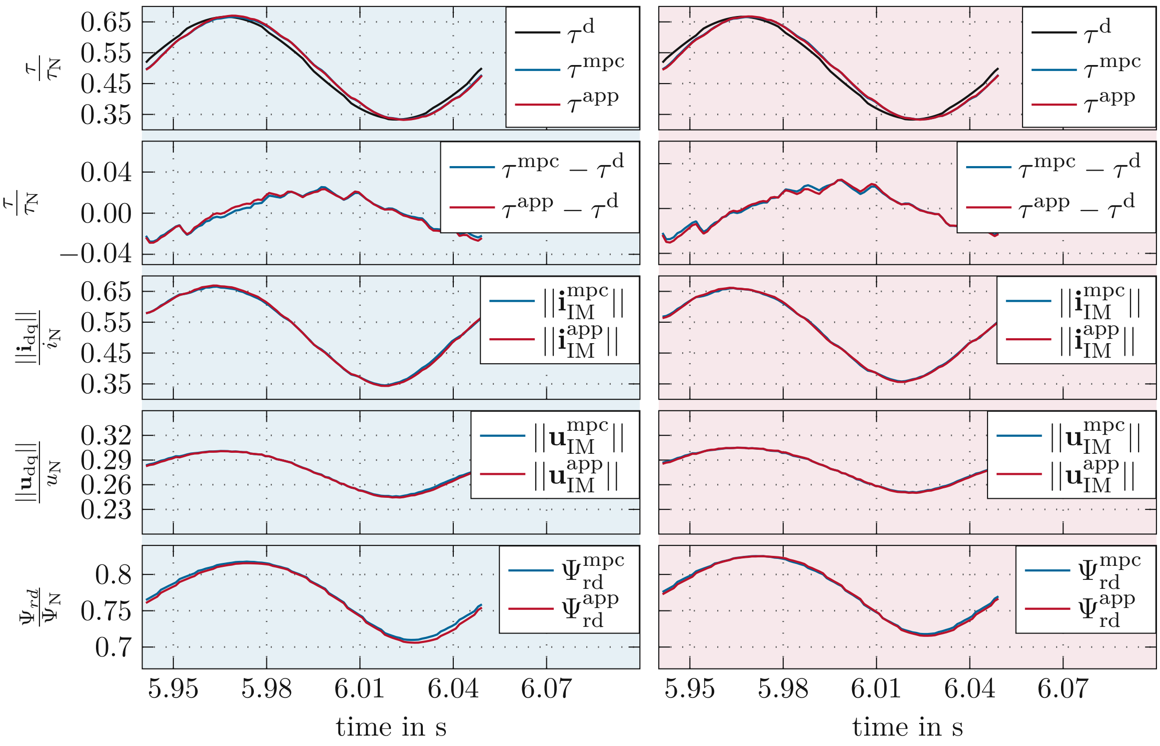
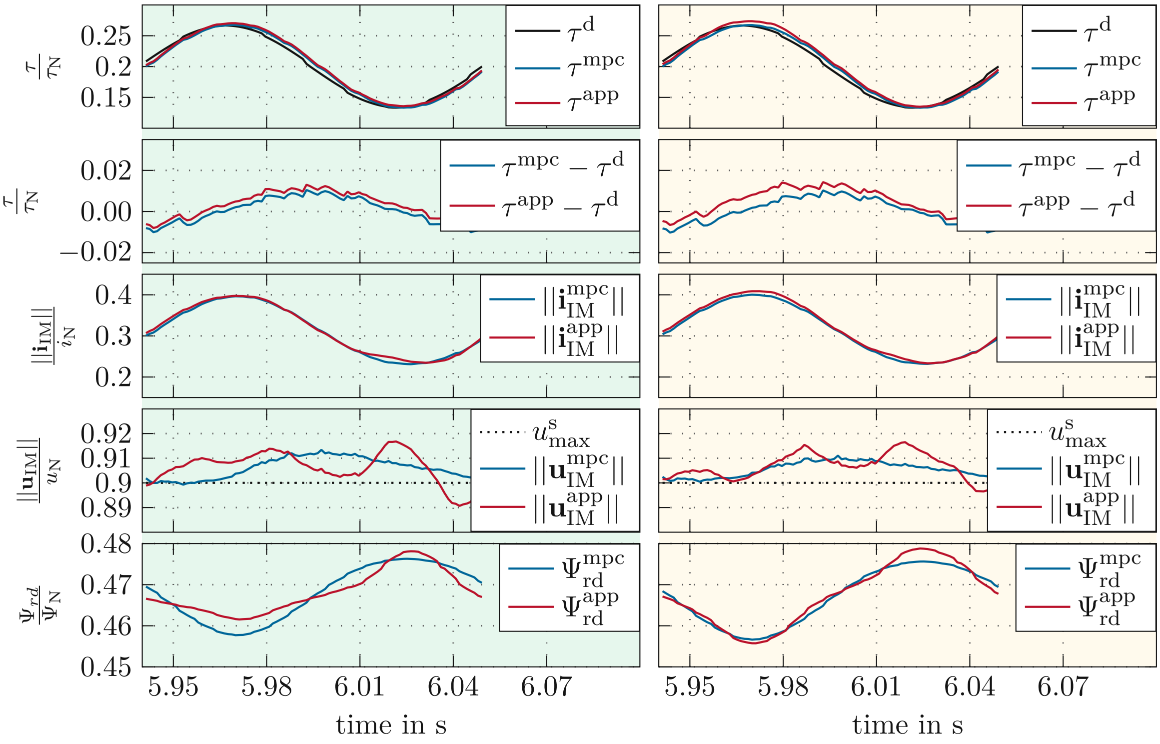


Figure 4.9: Transient simulations at $\omega = 0.15\omega_N$ without parameter deviations for the IM.

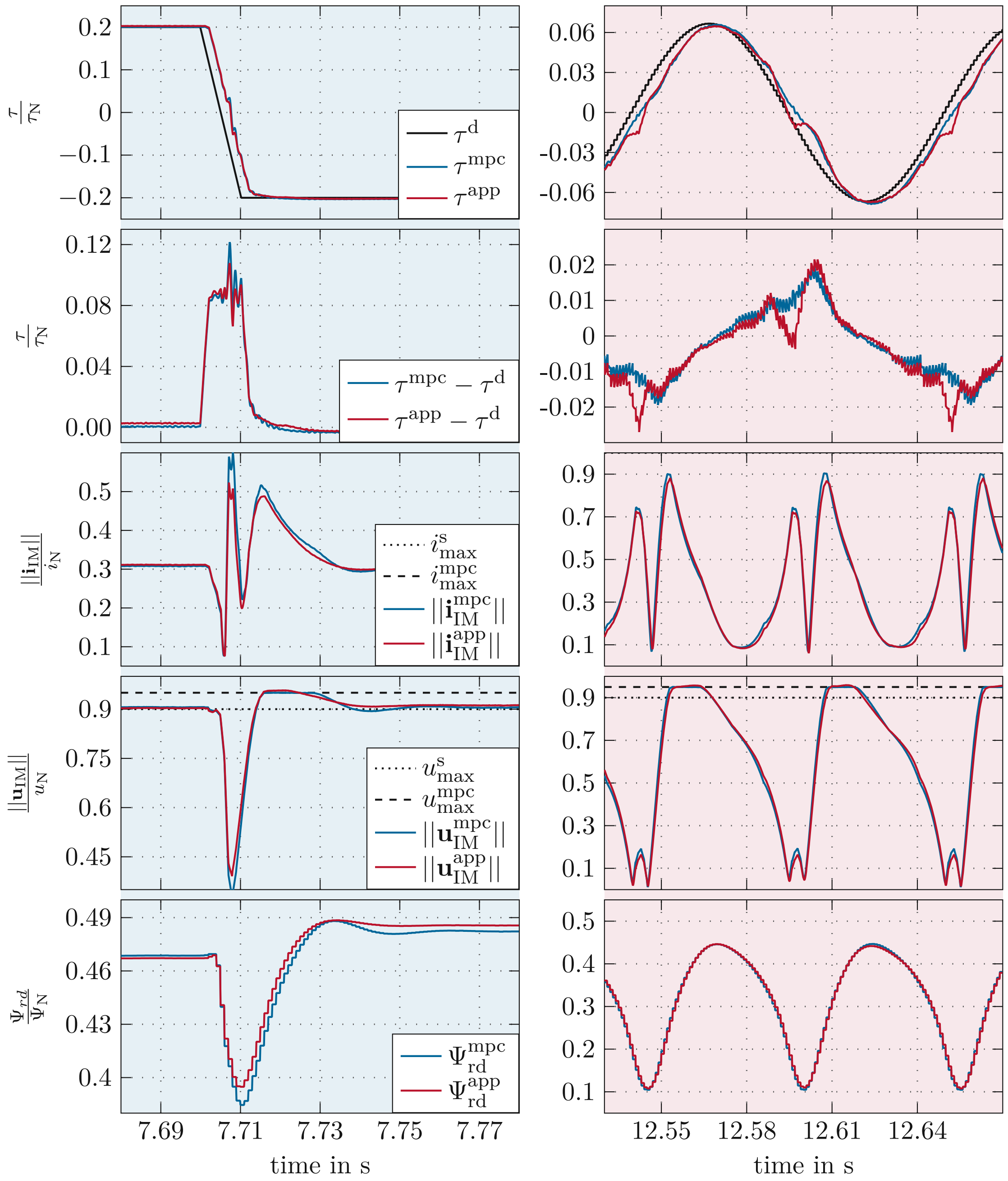


(a) $\omega = 0.15\omega_N$, nominal. (b)$\omega = 0.15\omega_N$, $\tilde{R}_r = 1.3R_r$.



(c) $\omega = 0.85\omega_N$, nominal. (d) $\omega = 0.85\omega_N$, $\tilde{R}_r = 1.3R_r$.

Figure 4.10: Zoomed detail of sine signal for different operating ranges of the MPC for the IM.



(a) Step response at $\omega = 0.85\omega_N$. (b) Sine around 0Nm at $\omega = 0.85\omega_N$.

Figure 4.11: Zoomed detail of fast ramp and sine signal around 0Nm without parameter deviations for the IM.

Since this behavior was seen in an amplified form for the original dataset, the dataset was changed as described in Section 4.5. This improved the torque error between the MPC and its approximation around 0Nm by a factor of 5. It is important to mention, that this behavior is not seen in the direct rotor flux, but it seems to come from the approximation of the stator quadrature flux. However, after adding the points and the retraining, the performance for this sinusoidal signal is acceptable also in this region. It is important to mention, that the MPC and its approximation meet the current and the voltage boundary in both of these transient scenarios. For the highly dynamic change it can furthermore be observed, that both approaches utilize the transient voltage constraint $u_{\mathrm{max}}^{\mathrm{mpc}}$ in order to achieve the highest possible dynamics before settling on the stationary voltage constraint $u_{\mathrm{max}}^{\mathrm{s}}$.

Finally, the prediction performance of the approximation on this dynamic trajectory for a low speed of $0.15\omega_{\mathrm{N}}$ is quantified by the means of the root mean squared error, the mean absolute error and the maximum absolute error between the MPC, its prediction and the setpoint values. These results are summarized in Table 4.2. All the values in this table were scaled by the nominal torque value τ_{N}. The table shows, that the error between the MPC and its approximation is by a factor of ≈ 10 smaller than the approximation error of the MPC to the setpoint. The performance of the MPC and its approximation is herein similar.

error	RMSE	MAE	Max
$\frac{\tau^{\mathrm{mpc}} - \tau^{\mathrm{d}}}{\tau_{\mathrm{N}}}$	0.0107	0.0033	0.1626
$\frac{\tau^{\mathrm{app}} - \tau^{\mathrm{d}}}{\tau_{\mathrm{N}}}$	0.0110	0.0039	0.1690
$\frac{\tau^{\mathrm{mpc}} - \tau^{\mathrm{app}}}{\tau_{\mathrm{N}}}$	0.0014	0.0010	0.0141

Table 4.2: Error measures for the approximation of the MPC.

4.7.3 Runtime evaluation

A runtime evaluation was performed on a dSPACE Microlab Box real-time platform. For this runtime evaluation, different neural net structures are compared with each other with respect to the computation time. The compared nets have the following structures:

- Net 1: 4 hidden layers, with 300, 150, 75, 50 nodes respectively
- Net 2: 4 hidden layers, with 50, 40, 30, 20 nodes respectively
- Net 3: 3 hidden layers, with 70, 50, 30 nodes respectively

The results of this measurement are summarized in Table 4.3. The runtime for the MPC in this table is given as a range, since the time duration of the optimal setpoint calculation

	MPC	Net 1	Net 2	Net 3
Runtime in μs	240-360	60	8	7

Table 4.3: Runtime evaluation of the developed approximation approach.

is dependent on the setpoint. The cascaded calculation of the optimal setpoint values yields thus for points in the voltage boundary the highest values, while the lowest runtime is achieved for the unconstrained case. This behavior is more thoroughly analyzed in [18]. However, this behavior is not seen for the approximation of the MPC, since for the approximation the neural network is evaluated for each setpoint in the same way.
The shown nets were all trained on the dataset from Section 4.5. Net 1 is the neural net structure before the ablation study was performed. Net 2 and Net 3 are two minimal examples of neural nets, which achieved an acceptable approximation accuracy. For the presented evaluations, however, a slightly more complex net than Net 2 was used. This means, that the runtime of this neural net should be in the range of 10 to 15μs. This means, that by the approximation of the MPC the runtime could be decreased on this hardware by a factor of approximately 20 (depending on the setpoint). However, if a more specific hardware is used, which is able to perform matrix multiplications in a fast manner as for example a GPU or a TPU, this factor could be increased drastically, since the necessary operations are not complex and no optimization problem has to be solved online. Therefore, this approach shows high potential to make MPCs real time capable in systems with high sampling frequencies.

5 Discussion and further outlook

In this work, three different MPCs for the PMSM and one complex MPC for the IM are approximated via data driven methods. The overall approach is introduced and explained in detail, from the data generation up to the evaluation in simulation. The reduction of simulation time by choosing a suitable simulation dataset is presented, as well as the used data driven approach using a feedforward neural net. It could be shown, that the approximation accuracy is high over the whole operating range, even in the case of parameter deviations. The MPCs performance is evaluated in simulations on steady-state and dynamic test cycles and performs overall very similar to the true MPC. Furthermore, the runtime evaluation shows a big potential to save computational time on a target hardware and make the approximation thus real-time capable. This would be a big breakthrough in the field of MPCs, since the real-time capability of different MPCs is often a hard task to achieve, especially if the sampling frequencies are as high as in the here discussed case.

Even if the shown approximation accuracy is very good over large areas, there are still some open topics to be further analyzed. The approximation of the MPC for the IM in the region of 0Nm should be investigated in more detail in order to ensure a smooth and safe operation of the electrical machine. Furthermore, simulations could be performed on a finer grid with a higher resolution in order to get an even better approximation accuracy. For the IM, the setpoint calculation and the MPC approximation could be split up in two different neural networks and trained separately. By doing so, the setpoint calculation method could be exchanged without the need of a retraining of the MPC approximation. Furthermore, as the runtime evaluation shows, the approximation of the MPC is fast enough to ensure real-time capability for the sampling time of the state space controller. Therefore it could be examined, how the performance of the overall control concept would change by omitting the state space controller and using directly the optimized voltage from the MPC output. Another topic that could be investigated, is the analysis of a different data driven approach. This work focuses only on the usage of neural networks for the approximation. However, a regression technique as for example XGBoost or a Random Forest Regressor might work in a similar fashion with even lower computation times or higher accuracies. Furthermore, the operating range of the MPC for the PMSM could be extended to the generator region and there the behavior could be examined. Finally, the developed concept could be used for measurements on the testbench and the performance could be evaluated against the MPC.

Overall, this work creates the basis for many other interesting projects. It shows that the field of control and the machine learning field can go hand in hand to achieve the highest performances after which the industry strives. The approximation concepts that are here presented, are only shown for the usecase of flux control. However, the developed data driven approach can be easily used on another dataset for a completely different task, as long as the input-output behavior is only stationary and not dependent on other time series data.

Bibliography

[1] Y. Van Fan, S. Perry, J. J. Klemeš, and C. T. Lee, "A review on air emissions assessment: Transportation," *Journal of cleaner production*, vol. 194, pp. 673–684, 2018. DOI: `10.1016/j.jclepro.2018.05.151`.

[2] E. Agamloh, A. von Jouanne, and A. Yokochi, "An overview of electric machine trends in modern electric vehicles," *Machines*, vol. 8, no. 2, 2020. DOI: `10.3390/machines8020020`.

[3] C. Lai, G. Feng, K. Mukherjee, V. Loukanov, and N. C. Kar, "Torque ripple modeling and minimization for interior pmsm considering magnetic saturation," *IEEE Transactions on Power Electronics*, vol. 33, no. 3, pp. 2417–2429, 2017. DOI: `10.1109/TPEL.2017.2695440`.

[4] T. Tuovinen, M. Hinkkanen, and J. Luomi, "Modeling of saturation due to main and leakage flux interaction in induction machines," *IEEE Transactions on industry applications*, vol. 46, no. 3, pp. 937–945, 2010. DOI: `10.1109/TIA.2010.2045210`.

[5] P. Mauborgne, S. Deniaud, E. Levrat, E. Bonjour, J.-P. Micaëlli, and D. Loise, "Operational and system hazard analysis in a safe systems requirement engineering process–application to automotive industry," *Safety science*, vol. 87, pp. 256–268, 2016. DOI: `10.1016/j.ssci.2016.04.011`.

[6] R. Marino, P. Tomei, and C. M. Verrelli, *Induction motor control design*. Heidelberg: Springer Science & Business Media, 2010.

[7] D. Schröder and J. Böcker, *Elektrische Antriebe-Regelung von Antriebssystemen*. Heidelberg: Springer, 2009.

[8] G. Buja and M. Kazmierkowski, "Direct torque control of pwm inverter-fed ac motors - a survey," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 4, pp. 744–757, 2004. DOI: `10.1109/TIE.2004.831717`.

[9] L. Harnefors and H.-P. Nee, "Model-based current control of ac machines using the internal model control method," *IEEE Transactions on Industry Applications*, vol. 34, no. 1, pp. 133–141, 1998. DOI: `10.1109/28.658735`.

[10] M. Merzoug and F. Naceri, "Comparison of field-oriented control and direct torque control for permanent magnet synchronous motor (pmsm)," *International Journal of Electrical and Computer Engineering*, vol. 2, no. 9, pp. 1797–1802, 2008. DOI: `10.36227/techrxiv.171332371.13141782/v1`.

[11] J. Rodriguez *et al.*, "Latest advances of model predictive control in electrical drives—part ii: Applications and benchmarking with classical control methods," *IEEE Transactions on Power Electronics*, vol. 37, no. 5, pp. 5047–5061, 2021. DOI: `10.1109/TPEL.2021.3121589`.

[12] F. Wang, X. Mei, J. Rodriguez, and R. Kennel, "Model predictive control for electrical drive systems-an overview," *CES Transactions on Electrical Machines and Systems*, vol. 1, no. 3, pp. 219–230, 2017. DOI: `10.23919/TEMS.2017.8086100`.

[13] M. Schenke, W. Kirchgässner, and O. Wallscheid, "Controller design for electrical drives by deep reinforcement learning: A proof of concept," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4650–4658, 2019. DOI: `10.1109/TII.2019.2948387`.

[14] L. Lüken, D. Brandner, and S. Lucia, "Sobolev training for data-efficient approximate nonlinear mpc," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5765–5772, 2023. DOI: `10.1016/j.ifacol.2023.10.545`.

[15] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu, "Sobolev training for neural networks," *Advances in neural information processing systems*, vol. 30, 2017. DOI: `10.48550/arXiv.1706.04859`.

[16] S. Chen *et al.*, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American control conference (ACC)*, IEEE, Milwaukee, 2018, pp. 1520–1527. DOI: `10.23919/ACC.2018.8431275`.

[17] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018. DOI: `10.1109/LCSYS.2018.2843682`.

[18] G. Janisch, W. Kemmetmüller, and A. Kugi, "A high-performance model predictive torque control concept for induction machines for electric vehicle applications," *Control Engineering Practice*, vol. submitted, 2024.

[19] D. Schröder, *Elektrische Antriebe: Grundlagen.* Heidelberg: Springer, 1994.

[20] G. Janisch, A. Kugi, and W. Kemmetmüller, "Model calibration strategy for energy-efficient operation of induction machines," *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 307–312, 2022. DOI: `10.1016/j.ifacol.2022.09.113`.

[21] S. Fligl, J. Bauer, M. Vlcek, and J. Lettl, "Analysis of induction machine T and Γ circuit coequality for use in electric drive controllers," in *2012 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, 2012, pp. 659–664. DOI: `10.1109/OPTIM.2012.6231923`.

[22] R. H. Park, "Two-reaction theory of synchronous machines generalized method of analysis-part i," *Transactions of the American Institute of Electrical Engineers*, vol. 48, no. 3, pp. 716–727, 1929. DOI: `10.1109/T-AIEE.1929.5055275`.

[23] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989, ISSN: 0005-1098. DOI: `10.1016/0005-1098(89)90002-2`.

[24] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966. DOI: `10.1126/science.153.3731.34`.

[25] A. Sen and M. Srivastava, *Regression analysis: theory, methods, and applications.* Heidelberg: Springer Science & Business Media, 2012.

[26] R. Dastres and M. Soori, "Artificial neural network systems," *International Journal of Imaging and Robotics (IJIR)*, vol. 21, no. 2, pp. 13–25, 2021.

[27] A. F. Agarap, "Deep learning using rectified linear units (relu)," *CoRR*, vol. abs/1803.08375, 2018. DOI: `10.48550/arXiv.1803.08375`.

[28] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998. DOI: `10.1142/S0218488598000094`.

[29] R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen, "Ablation studies in artificial neural networks," *CoRR*, vol. abs/1901.08644, 2019. DOI: `10.48550/arXiv.1901.08644`.

[30] R. Llugsi, S. El Yacoubi, A. Fontaine, and P. Lupera, "Comparison between adam, adamax and adam w optimizers to implement a weather forecast based on neural networks for the andean city of quito," in *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, IEEE, 2021, pp. 1–6. DOI: `10.1109/ETCM53643.2021.9590681`.

[31] S.-I. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4, pp. 185–196, 1993, ISSN: 0925-2312. DOI: `https://doi.org/10.1016/0925-2312(93)90006-O`.

[32] M. B. Wilk and R. Gnanadesikan, "Probability plotting methods for the analysis for the analysis of data," *Biometrika*, vol. 55, no. 1, pp. 1–17, 1968. DOI: `10.2307/2334448`.

[33] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear model predictive control*. Heidelberg: Springer, 2017.

[34] M. Gerdts, *Optimal control of ODEs and DAEs*. Berlin: Walter de Gruyter GmbH & Co KG, 2023.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct - Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im Mai 2024

Felix

Felix Heidegger