

---

# Flatness-based predictive and optimal control for electrical drives

---

Jean-François Stumper



Lehrstuhl für elektrische Antriebssysteme und Leistungselektronik  
der Technischen Universität München

# Flatness-based predictive and optimal control for electrical drives

Jean-François Stumper

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informations-  
technik der Technischen Universität München zur Erlangung des akademischen  
Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Hans-Georg Herzog

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Ralph Kennel
2. Univ.-Prof. Dr.-Ing. Mario Pacas, Universität Siegen
3. Prof. Dr.-Ing. Holger Voos, Universität Luxemburg

Die Dissertation wurde am 28.03.2012 bei der Technischen Universität München  
eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am  
04.12.2012 angenommen.



---

## Vorwort

---

Die vorliegende Arbeit entstand während meiner Tätigkeit am Lehrstuhl für elektrische Antriebssysteme und Leistungselektronik der Technischen Universität München.

Zunächst möchte ich Herrn Prof. Ralph Kennel für die großzügige Unterstützung und die Anregungen während meiner Promotion danken. Besonders dankbar bin ich für das entgegengebrachte Vertrauen und den Freiraum während allen Abschnitten dieser Arbeit. Die vielen ermöglichten Konferenzreisen waren eine zusätzliche Bereicherung mit bleibenden Eindrücken.

Sehr gefreut habe ich mich darüber, dass Herr Prof. Holger Voos von der Universität Luxemburg und Herr Prof. Mario Pacas von der Universität Siegen die Anfertigung der Gutachten übernommen haben. Ich bedanke mich bei Ihnen für Ihr reges Interesse an der Arbeit.

Des Weiteren danke ich dem Fonds National de la Recherche in Luxemburg für die finanzielle Unterstützung dieser Arbeit in Form eines Stipendiums.

Diese Arbeit wäre ohne die tatkräftige Unterstützung meiner Kollegen nicht möglich gewesen. Herrn Dirk Paulus danke ich für die Starthilfe bei den vielen praxisbezogenen Angelegenheiten. Es war eine große Freude gemeinsam dem Labor des Vorgängerlehrstuhls neues Leben einzuhauchen. Weiterhin danke ich Herrn Peter Stolze und Herrn Sascha Kühl für die hervorragende Atmosphäre im Büro sowie für die vielen fachlichen Hilfen und Diskussionen.

All meinen Studenten, insbesondere aber Herrn Alexander Dötlinger und Herrn Janos Jung, möchte ich für ihre arbeitsintensiven Beiträge danken.

Auch außerhalb unseres Lehrstuhls fand ich Unterstützung. So danke ich Herrn Dr. Veit Hagenmeyer für die kritische und zugleich konstruktive Diskussion über die Arbeit in den Kapiteln 3 und 4.

Luxemburg, Frühjahr 2013



---

# Contents

---

<b>Kurzfassung</b>	<b>VII</b>
<b>Abstract</b>	<b>IX</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions . . . . .	2
1.2 Differential flatness . . . . .	5
1.3 Flatness-based control . . . . .	7
<b>2 Machine models and conventional control methods</b>	<b>11</b>
2.1 Representation with space vectors . . . . .	12
2.2 Model of synchronous machines . . . . .	14
2.3 Model of induction machines . . . . .	16
2.4 Systematic constraints . . . . .	21
2.5 Model discretization and delay compensation . . . . .	22
2.6 Limitations of linear flux models . . . . .	25
2.7 Conventional control: Optimized PI controller . . . . .	26
2.8 Flatness of electrical drives . . . . .	30
<b>3 Robust deadbeat current control for AC drives</b>	<b>33</b>
3.1 Preliminaries . . . . .	34
3.1.1 Deadbeat current control for AC drives: State of the art . . .	34
3.1.2 Basic deadbeat current controller design . . . . .	37
3.1.3 Overall control structure . . . . .	37
3.2 Deadbeat control in the flatness-based context . . . . .	40
3.2.1 Principles of deadbeat control and relation to flatness . . . .	40

3.2.2	Conventional deadbeat current controller design . . . . .	43
3.2.3	Deadbeat design using feedforward linearization . . . . .	44
3.2.4	Deadbeat design using mixed feedback and feedforward linearization . . . . .	47
3.3	Analytical robustness considerations . . . . .	48
3.3.1	Transfer function of deadbeat controller . . . . .	49
3.3.2	Transfer function of deadbeat controller including a disturbance estimator . . . . .	49
3.3.3	Interaction of disturbance estimator and deadbeat controller: Low pass filter . . . . .	50
3.3.4	Steady-state accuracy . . . . .	50
3.4	Conclusions . . . . .	51
<b>4</b>	<b>Experimental evaluation of deadbeat control</b>	<b>53</b>
4.1	Comparative evaluation of the proposed controllers . . . . .	54
4.1.1	Evaluation of the conventional deadbeat controller . . . . .	54
4.1.2	Evaluation of the deadbeat controller using feedforward linearization . . . . .	60
4.1.3	Evaluation of the deadbeat controller using mixed feedback and feedforward linearization . . . . .	63
4.1.4	Conclusions on the comparative evaluation . . . . .	68
4.1.5	Structural difference between deadbeat and PI control . . . . .	70
4.2	Implementation and performance evaluation . . . . .	72
4.2.1	Results on a permanent-magnet synchronous machine . . . . .	72
4.2.2	Results on a synchronous reluctance machine . . . . .	74
4.2.3	Results on an induction machine . . . . .	77
4.3	Conclusions . . . . .	83
<b>5</b>	<b>Linear model predictive control for PMSMs</b>	<b>85</b>
5.1	Benefits of optimal control . . . . .	86
5.1.1	Preliminaries: Loss modeling in PMSMs . . . . .	86
5.1.2	Optimization of the power efficiency . . . . .	87
5.1.3	Optimization of the dynamic response . . . . .	88
5.1.4	Problem statement: Receding horizon formulation . . . . .	89
5.2	Linear model predictive control . . . . .	93
5.2.1	Predictive control, MPC, and flatness . . . . .	93
5.2.2	Online optimization: A real-time problem . . . . .	94
5.2.3	A simple and efficient LQ optimization algorithm . . . . .	97
5.2.4	Implementation and control structure . . . . .	102



---

5.3	Results . . . . .	104
5.3.1	Simulation results . . . . .	106
5.3.2	Experimental results . . . . .	107
5.4	Conclusions . . . . .	110
<b>6</b>	<b>Nonlinear optimal control for IMs</b>	<b>111</b>
6.1	Potential of optimal control . . . . .	112
6.1.1	Preliminaries: Loss modeling in IMs . . . . .	112
6.1.2	Efficiency optimization in steady-state . . . . .	113
6.1.3	Efficiency optimization – transient operation . . . . .	115
6.1.4	Problem statement: Optimal torque control for transients . . . . .	117
6.2	Real-time trajectory generation . . . . .	118
6.2.1	Formulation of a real-time optimization problem . . . . .	118
6.2.2	Calculus of variations . . . . .	120
6.2.3	Approximate solution using prototype fitting . . . . .	121
6.2.4	Control structure: Trajectory generation and tracking . . . . .	124
6.3	Experimental results . . . . .	126
6.3.1	Torque response . . . . .	127
6.3.2	Efficiency evaluation: A servo application . . . . .	130
6.4	Conclusions . . . . .	133
<b>7</b>	<b>Conclusions and outlook</b>	<b>135</b>
<b>A</b>	<b>List of symbols and abbreviations</b>	<b>139</b>
A.1	List of symbols . . . . .	139
A.2	Abbreviations . . . . .	142
<b>B</b>	<b>Experimental setup</b>	<b>143</b>
B.1	Parameters of the test machines . . . . .	145
<b>C</b>	<b>Detailed machine model deductions</b>	<b>147</b>
C.1	Synchronous machine . . . . .	147
C.2	Induction machine . . . . .	149
<b>D</b>	<b>Robustness calculations for deadbeat control</b>	<b>155</b>
D.1	Transfer function of deadbeat controller . . . . .	155
D.2	Transfer function including a disturbance estimator . . . . .	156
D.3	Interaction of disturbance estimator and deadbeat control . . . . .	159
D.4	Steady-state accuracy . . . . .	159

---

<b>E Outlook: Feedforward linearization for PTC</b>	<b>161</b>
<b>F Mathematical complements to chapters 5 &amp; 6</b>	<b>167</b>
F.1 Analysis of steady state accuracy of the cost function. . . . .	167
F.2 Convexity of cost function with polynomial parameterization . . . .	169
F.3 Parameterization of the constraints with a polynomial basis: Sufficient affine conditions . . . . .	170
F.4 Analysis: Remainder of the Euler-Lagrange equation . . . . .	171
<b>G Related works</b>	<b>173</b>
G.1 Advised student theses . . . . .	173
G.2 List of publications . . . . .	174
<b>Bibliography</b>	<b>177</b>

---

## Kurzfassung

---

Die vorliegende Arbeit befasst sich mit der modellbasierten Regelung elektrischer Antriebe. Aufgrund der gegenwärtigen hohen Rechenleistung sowie der großen Fortschritte der Regelungstechnik eröffnen sich neue Möglichkeiten, insbesondere zur prädiktiven und optimalen Regelung. Die wesentliche Grundlage der vorgestellten Entwürfe bildet der aus der Regelungstheorie stammende Ansatz der differentiellen Flachheit.

Als prädiktive Regelung wird die deadbeat-Regelung behandelt. Diese Methode findet in hochdynamischen Strom- und Drehmomentregelungen Anwendung, ist jedoch sehr empfindlich gegen Parameterfehler. Des Weiteren bilden konstruktionsbedingte Oberwellenschwingungen ein Problem. Durch einen erneuerten Reglerentwurf, welcher auf der exakten Steuerungslinearisierung basiert, wird die Robustheit der Regelung maßgeblich verbessert. Hohe Parameterabweichungen werden zulässig. Zudem ergibt sich die Möglichkeit, den begleitenden Störgrößenbeobachter deutlich schneller einzustellen, wodurch Oberwellenschwingungen quasi komplett gedämpft werden. Die deadbeat-Regelung wird somit ausreichend zuverlässig und leistungsfähig für einen industriellen Einsatz. Dieses Ergebnis wird analytisch und experimentell bestätigt. Die Methode ist auf andere modellbasierte Regelungsverfahren übertragbar.

Die optimale Regelung stellt die Zustandsgrößen nicht auf vorgegebene Sollwerte, sondern führt sie gemäß Optimalitätskriterien. Ziel ist es, soweit physikalisch möglich, gleichzeitig die Dynamik und die Energieeffizienz zu verbessern. Hierzu können bestehende Freiheitsgrade über die Regelung ausgenutzt werden.

Für die Synchronmaschine wird eine lineare modellprädiktive Regelung (MPC) realisiert. Entscheidend für die praktische Umsetzbarkeit dieser Methode bei dermaßen hohen Abstraten ist ein schneller Algorithmus zum Lösen linear-quadratischer Optimierungsprobleme. Die Regelung nutzt als Freiheitsgrad die Feldschwä-

chung aus, um die Verluste zu minimieren und in einigen Situationen die Dynamik zu verbessern.

Für die Asynchronmaschine muss eine nichtlineare Optimierung durchgeführt werden. Zur besseren Realisierbarkeit wird eine approximative Lösung auf der Grundlage der Variationsrechnung angewandt. Im Gegensatz zu den bekannten Verfahren werden auch Transienten im Drehmoment hinsichtlich der Effizienz optimiert, wodurch sich Vorteile und weitere Einsparungen für dynamische Prozesse ergeben.

Die Anwendung der differentiellen Flachheit in den einzelnen Gebieten ermöglicht demnach die Verbesserung bestehender Lösungen sowie die Einführung neuer Verfahren.

---

## Abstract

---

The underlying thesis deals with model-based control for electrical drives. Due to the availability of extended computational power and to the progress of control systems technology, today, there are new possibilities for predictive and optimal control. Fundamental to the proposed designs is the use of differential flatness, an approach from control systems theory.

Deadbeat control is treated as predictive control scheme. This scheme is used in highly dynamic current and torque controllers, but is very sensitive to parameter errors. Furthermore, harmonic oscillations caused by the machine construction represent a problem. By a design change that applies feedforward linearization, the robustness of the controller is significantly improved. High parameter offsets become tolerable. Additionally, the accompanying disturbance estimator can be tuned up, resulting in almost complete damping of the harmonic oscillations. The control system thereby becomes sufficiently robust and performant for an industrial application. This result is confirmed analytically and experimentally. The method is applicable to other model-based control schemes.

In the optimal control schemes, the state variables are not controlled to predefined references, but according to optimality criteria. The goal is the simultaneous improvement of the dynamics and the energy efficiency, as far as this is physically possible. To do this, the controller can exploit existing degrees of freedom.

For the synchronous machine, a linear model predictive controller (MPC) is realized. The enabling factor for feasibility on such a fast system is the introduction of a fast algorithm to solve linear-quadratic optimization problems. The control system exploits field weakening as degree of freedom to minimize the losses and to improve the dynamics in certain situations.

For the induction machine, a nonlinear optimization has to be performed. In order to realize this in a simple way, an approximate solution is proposed based

on calculus of variations. In contrast to the known methods, torque transients are also considered in the optimization, resulting in further savings and advantages for dynamic processes.

In all these different areas, the use of differential flatness enables the improvement of existing solutions and the introduction of new procedures.

# CHAPTER 1

---

## Introduction

---

Flatness, in the broadest sense, is a simplification. A control designer is not stuck in front of a bunch of equations; using flatness, the problem can be clarified and reduced to one core problem. Flatness is a very fundamental characteristic natural to many systems, and the designer may have come up with the same solution while ignoring this property – but he did not because the original problem has been obscure. Too obscure to figure out where the real problem is, too complicated such that his solution was too demanding. Insight and inventive ideas come if problems are simple. This makes flatness to a powerful design tool.

The term 'flatness' does not describe a specific control algorithm, it is a quite general approach in the analysis and design of dynamical systems. Introduced in 1992, this concept has rapidly become a new and important branch in the design of nonlinear continuous-time control systems. Apart from this academic success, it has found its way into industrial applications as well.

This thesis applies the flatness-based design methods to electrical drive systems. Two high performance control schemes are studied, predictive and optimal control. Existing designs are recasted and improved, and new designs are proposed. It is a difficult task to show that a fundamental theory, such as flatness, gives any merits to the practicing engineer. This is why it has to be put on a practical test stand. The control problems in electrical drives are of physical nature, therefore, from the resulting performance it can be clearly judged whether the proposed designs are good or not.

The behavior of a system can be improved by exploiting the system model more extensively. While feedback controllers use a model in the design to determine the feedback gains, the model is not used explicitly in the control policy. Modern microprocessors provide a great amount of computational capacity and performance that can be used to extract more information online from the model. Furthermore, a number of advanced algorithms is available to either impress a desired behavior or to optimize the future behavior on the system to be controlled. These two tools, computational power and a well-developed control theory, enable further improvements in the field of electrical drive systems.

Predictive control is a family of advanced control strategies that had a tremendous impact on industrial practice in many technical fields. Its main advantage is its simple basic concept. Beyond that, it outperforms classical control methods such as standard PID control even on simple systems. In predictive and optimal control, the best possible operation of an electrical drive according to a defined performance criterion within the physical limits is achieved. Performance criteria may be fastest possible satisfaction of a control target or maximization of power efficiency. Physical limitations are the maximum values for voltage and currents. Using these methods, benefits can be obtained regarding issues such as energy saving and safety.

## 1.1 Contributions

All the works in this thesis consider the design of control systems in the field-oriented frame for rotating three-phase alternating-current (AC) machines. These include permanent-magnet synchronous machines (PMSM), a widespread drive for highly dynamical servo systems, squirrel-cage induction machines (IM), the most common type for reliable and low-cost servo and standard applications, and the synchronous reluctance machine (SynRM), a rather exotic variant of a synchronous machine that does not require permanent-magnets.

The design of model-based predictive and optimal control schemes is carried out for these drive systems. Specific goals are highly performant control, i.e. a fast and precise reference response, and optimal control, i.e. the exploitation of the highest possible energy efficiency. Here, however, some limitations apply. A key issue is the sensitivity to uncertainties, arising from nonlinearities, disturbances, noise, unmodeled interactions of the variables or simply from approximate and inaccurate models. Furthermore, the high computational demands of some methods are an obstacle in the implementation. In this thesis, efficient solutions to these problems are proposed using flatness as design aid in order to make these techniques suitable for industrial practice.



Robustness of predictive control to both model parameter uncertainties and time-varying disturbances is discussed in chapters 3 and 4. The example of deadbeat current control is treated. The methods are assumed to be applicable to many predictive control schemes. Chapter 3 presents the theory and chapter 4 the experimental evaluation and application. An outlook on a different predictive control scheme is presented in appendix E.

The conventional design of deadbeat current control can be classified as flatness-based predictive control. This design relies on exact feedback linearization, a method known to be very sensitive to parametric uncertainties. In flatness-based control, a substitute has been proposed, exact feedforward linearization. This method is an open-loop control method and therefore invariant to parametric uncertainties, however, it also reduces performance. A mix between exact feedback and feedforward linearization is proposed, leading to a tradeoff between both methods. It can improve parametric robustness to a sufficient extent while maintaining good control performance. Furthermore, it robustly decouples the controller from the disturbance estimator, such that the disturbance attenuation mechanism can be tuned extremely fast. Then, time-varying disturbance effects such as those arising from flux harmonics can be attenuated without explicitly identifying them.

The resulting control scheme is very performant, it has a strong tracking capability and at the same time provides good disturbance rejection. Furthermore, it is very robust to parametric uncertainties and less sensitive to measurement noise. These advantages are highlighted by a comparison to a PI feedback controller in chapter 2 and by the experimental evaluation on the three named types of AC machines in chapter 4.

Chapters 5 and 6 discuss optimal control of electrical drives. A focus is given on dynamics, transient operation is central in the design. Both, dynamic operation and energy efficiency, are regarded. As such aims are closely related to the physical principles of the respective machine type, optimal control of PMSM and squirrel-cage IM are separately treated.

Optimal control of permanent-magnet synchronous machines is treated in chapter 5. It is shown that fast torque tracking and energy efficiency are two goals that can be simultaneously obtained, they do not contradict. By formulating an optimal torque control problem, the cost function appears as quadratic type, with a linear system model and linear constraints. A receding horizon problem is formulated, the optimization horizon is set very high, such that closed-loop optimality can be claimed. A long-horizon linear model predictive control scheme (MPC) can therefore provide optimal control.

MPC is based on direct optimal control. The states and inputs are parameterized, and a parametric optimization problem is solved online to meet the actual demands

(torque reference) at the actual condition (speed and initial currents). The major problem in implementing MPC on an electrical drive is the limited computational power, inherited by the high sampling rates. To cope this, a simplified approach is developed based on differential flatness, on continuous-time polynomial parameterization and on linearization and subsequent use of an efficient linear programming solver. This can solve the computation problem, contrary to the standard approach of piecewise constant (discrete-time) parameterization and use of the computationally more intensive quadratic programming solvers. The simplified method is sub-optimal, but still leads to very good results, demonstrated in the experiments.

The experimental implementation can be seen as the first realization of long-range MPC with online-optimization for an electrical drive. There have already been successful implementations, however, they are either with a short (one step) prediction horizon, or based on unconstrained optimization, or an offline calculated explicit solution. Compared to the existing schemes, the experimental results show different and improved behavior, especially regarding the use of dynamic field-weakening – this novelty is to be attached to the high prediction horizon with constraints.

Optimal control of induction machines is treated in chapter 6. Here, fast torque tracking and energy efficiency stand in physical contradiction, however. Reducing losses requires reduction of the flux, preventing a fast torque increase because of the current limitation. The optimization problem has to be defined in a special way, the torque dynamics are fixed by defining a desired trajectory. For this prescribed transient, the optimal flux and current trajectories are determined. The optimization problem is of highly nonlinear nature.

Because of the nonlinearities and the difficulties of nonlinear solvers, indirect optimal control is chosen. Calculus of variations is applied to derive the necessary and sufficient conditions of optimality. A nonlinear two-point boundary value problem (BVP) appears that must be solved. As this is a hard numerical task, a simple heuristic solution is proposed. Based on observations, it is found that the optimal flux resembles an exponential function. The time-dependent behavior of the torque is described as an exponential function from its initial to the desired value with an arbitrary prescribed time constant. This function is fitted such that the BVP is approximately solved, the boundary conditions define the initial and final value, and the optimal time constant is determined numerically using Newton-Raphson search. The computational burden of the method is quite small and the results are fine.

It is the first efficiency optimization scheme that considers losses during torque transients and that operates without a-priori knowledge of speed and load trajectories. Optimization is based solely on actual measurements and references. It is known that flux reduction may result in more losses than operation at rated flux,

however, the proposed method is always more efficient than steady-state optimization. The potential improvements are demonstrated on a flywheel experiment.

The most important aspects of the work are concluded in chapter 7, where also an outlook on possible future works is given.

## 1.2 Differential flatness

Differential flatness is a concept introduced in 1992 by the researchers Fliess, Lévine, Martin and Rouchon [FLMR92,FLMR95]. It is a new approach for the analysis and design of nonlinear continuous-time control systems. From a mathematical perspective, flatness has its roots in differential-algebraic methods, and from control theoretic perspective, it is strongly related to controllability. The general introduction of this theory has had a strong academical impact<sup>1</sup>.

Differential flatness is a property of a nonlinear system. Differential flatness is an extension of the term 'controllability' for nonlinear systems. For linear systems, it is equivalent to controllability, but for nonlinear systems, it is a self-standing property. A given dynamical system is either flat or not, depending on whether it fits the following definition of flatness.

**Definition: Differential flatness.** Assuming a smooth nonlinear continuous-time system is given in state-space form

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (1.1)$$

with state vector  $\mathbf{x} \in \mathbb{R}^n$ , input vector  $\mathbf{u} \in \mathbb{R}^m$  and the system function vector  $\mathbf{f} \in \mathbb{R}^n$ . The system is further assumed to have  $m$  independent inputs, therefore  $\text{rank}\left(\frac{\partial}{\partial \mathbf{u}}\mathbf{f}(\mathbf{x}, \mathbf{u})\right) = m$ . This system is said to be differentially flat if there exists an output function

$$\mathbf{y}_f = \boldsymbol{\lambda}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{x}}, \dots, \mathbf{x}^{(\alpha)}), \quad (1.2)$$

with  $\mathbf{y}_f \in \mathbb{R}^m$  that satisfies the following two conditions:

1) All states  $\mathbf{x}$  and inputs  $\mathbf{u}$  can be expressed in terms of the flat output  $\mathbf{y}_f$  and a finite number of its derivatives:

$$\mathbf{x} = \Gamma_x\left(\mathbf{y}_f, \dot{\mathbf{y}}_f, \dots, \mathbf{y}_f^{(n-1)}\right), \quad (1.3)$$

$$\mathbf{u} = \Gamma_u\left(\mathbf{y}_f, \dot{\mathbf{y}}_f, \dots, \mathbf{y}_f^{(n)}\right). \quad (1.4)$$

---

<sup>1</sup>The Google Scholar entry of the paper [FLMR95], which is the introduction of flatness to the broad audience, lists nearly 1500 citations as of March 2012, a remarkable number for a theoretical control systems contribution.

2) As  $\dim(\mathbf{u}) = \dim(\mathbf{y}_f) = m$ , the components of the flat output  $\mathbf{y}_f$  are differentially independent, meaning, there is no non-trivial differential equation of the form

$$\psi(\mathbf{y}_f, \dot{\mathbf{y}}_f, \dots, \mathbf{y}_f^{(n)}) = 0. \quad (1.5)$$

The set of equations (1.3) and (1.4) is denoted as differential parameterization of the system variables and is used in the controller design.

A necessary condition for flatness is that the system must be controllable. A further necessary and sufficient condition is that the system must be linearizable by endogeneous feedback, i.e. either static or dynamic state feedback. If a controller canonical form (CCF) exists, then the CCF outputs are flat outputs [SRA04]. There are nonlinear systems that are controllable but not flat; no flat output and no differential parameterization can be found [FLMR95]. For electrical drives, however, a deep theoretical study is not required, virtually all electrical drives are controllable and differentially flat, as will be described later in section 2.8.

Differential flatness originates from continuous-time nonlinear systems. Despite this, throughout the thesis, other system classes such as linear and time-discrete systems will be studied. It was shown that flatness is a useful property for continuous-time linear systems as well [FM00a], even though it is equivalent to controllability, the flatness-based approach to a problem inspires different procedures. It was also shown that the theory can be directly applied to discrete-time linear [SRA04, FM00b] and nonlinear systems [Nih07, RCA01]. Chapters 3 and 4 apply flatness to discrete-time systems in both linear and nonlinear form. Chapter 5 is a continuous-time linear method, and chapter 6 continuous-time nonlinear method, the home area of flatness.

To the opinion of the author, the widespread definition already somehow resembles *conditions* for flatness rather than a general definition of that term. Therefore, a generalized definition – in the typical way technical terms are defined – is proposed. It shall give insight to the purpose of flatness.

**Generalized Definition: Differential flatness.** A system is differentially flat if there is a basic variable (a flat output) whose motion describes the complete motion of the system. An explicit invertible transformation leads to an equivalent algebraic system with the basic variable and its derivatives that is free of dynamics.

The connection between the outputs  $\mathbf{y}_f$  and the states  $\mathbf{x}$  or inputs  $\mathbf{u}$  can be done without that differential equations must be integrated or solved, instead, explicit algebraic functions are used. It is the transformation from the original state-space

description to the flat output form that represents the most useful simplification, for instance, in a nonlinear system, it may be very difficult to solve differential equations. In the flat output, the dynamics are nothing but a chain of integrators, all effects are described by straightforward differentiation of  $\mathbf{y}_f$ . So, if an output trajectory is known, by differentiating it, all states and inputs are given in straightforward algebraic manner.

The theory of flatness is well developed and has been made accessible by several books [SRA04, L  v09], in parallel, a high number of applications have been studied, many of them in mechatronics, power electronics and electrical drives. Other application fields are chemical reactors, flight control systems, control and design of mechanical systems, path planning for vehicles, and many more.

### 1.3 Flatness-based control

The following definition is proposed to clarify the flatness-based approach in a general controller design.

**Definition: Flatness-based control.** The term 'flatness based control' denotes that in at least one step in the design of a control algorithm, the explicit algebraic relation between full states or inputs ( $\mathbf{x}$  resp  $\mathbf{u}$ ) and the flat outputs and a number of its derivatives ( $\mathbf{y}_f, \dot{\mathbf{y}}_f, \dots$ ) has been applied.

Differential flatness can be used for control system design in many ways. Differential flatness is conceptually related to system inversion [SDP96], equations (1.3) and (1.4) may also be seen as an explicit inverse system model. In the typical state-space description with differential equations (1.1), with the knowledge of the control input, the states and outputs can be calculated by integrating or solving these equations. It can be very difficult to solve nonlinear differential equations with an a-priori unknown input, and then determine an input that satisfies the control task. In contrast, in an inverse model like (1.3) and (1.4), with knowledge of the output, the corresponding states and inputs are calculated directly in an algebraic manner. Inverse models are more convenient for controller design as they better fit the task, which is, finding an input that will satisfy the control goal. However, as a number of derivatives is required in the design, one must bear in mind that the resulting controllers may be useless, as measurement signals should generally not be differentiated. The problem of derivatives has been solved by the introduction of several general control structures, the most common continuous-time methods for continuous-time systems are presented in the following.

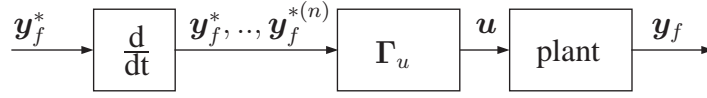


Figure 1.1: Flatness-based feedforward control.

The most straightforward application of flatness in control design is feedforward control, as shown in Fig. 1.1. Given a desired output trajectory  $\mathbf{y}_f^*(t)$  that describes for instance a setpoint change, by differentiating this trajectory and placing it into the input parameterization (1.4), the corresponding control input  $\mathbf{u}(t)$  is calculated directly without solving any equations. The control goal is satisfied only if the plant is stable, if the parameters are exactly known and if there are no disturbances. Therefore, in the practice, this open-loop controller must be extended with an additional feedback mechanism in order to obtain satisfactory results.

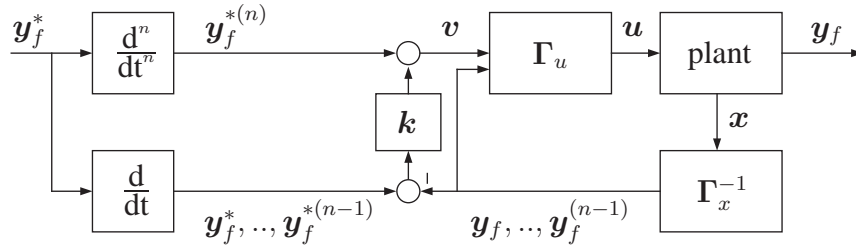


Figure 1.2: Flatness-based trajectory tracking control.

An extension of this design is trajectory tracking control [SL91, SRA04], shown in Fig. 1.2. A 'linearizing input' is implemented using (1.3) and (1.4), this technique is denoted as feedback linearization. The dynamics are compensated such that a new input appears as  $\mathbf{v} = \frac{d^n}{dt^n} \mathbf{y}_f^*$ , this means that the dynamics become an integrator chain. Similar as for feedforward control, based on the  $n$ -th derivative of the reference trajectory, a feedforward signal is calculated. Additionally, the tracking error, which is the control error  $\mathbf{y}_f^* - \mathbf{y}_f$  and a finite number of its derivatives, is weighted with feedback gains  $\mathbf{k}$  and added to the feedforward command. An initial offset from the reference trajectory is asymptotically decreasing with the desired error dynamics, defined by the feedback gain design. Therefore, with feedback linearization, well-known tools from linear control theory can be applied for nonlinear control design. The required output derivatives are not calculated from the measurement derivatives, but with the inverse state parameterization (1.4). A drawback

is the extremely high sensitivity to parametric uncertainties and disturbances. It has prevented the industrial acceptance of this design method.

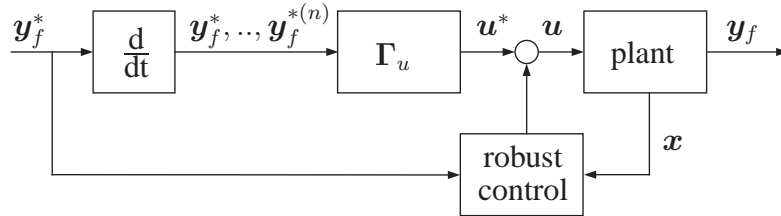


Figure 1.3: Flatness-based two-degree-of-freedom control.

The related two-degree-of-freedom (2DoF) design, shown in Fig. 1.3, has been much more successful [vNM98]. Here, the feedforward command  $u^*$ , calculated like in feedforward control based on a nominal model and references, is combined with a robust feedback controller. An offset of the measured trajectory from the reference trajectory, caused by either parametric uncertainties or disturbances, is robustly compensated. This design has been widely accepted, also as existing PID controllers can be used as robust feedback mechanism. In many applications, the pure extension from existing controllers to a 2DoF controller by adding a feedforward path has shown good performance improvements. This structure will be analyzed for current control in section 2.7.

The mentioned designs all track a predefined trajectory. If the goal is not to track a predefined path precisely, but if there is some freedom on the output trajectory, extending the structure to predictive control as shown in Fig. 1.4 makes sense.

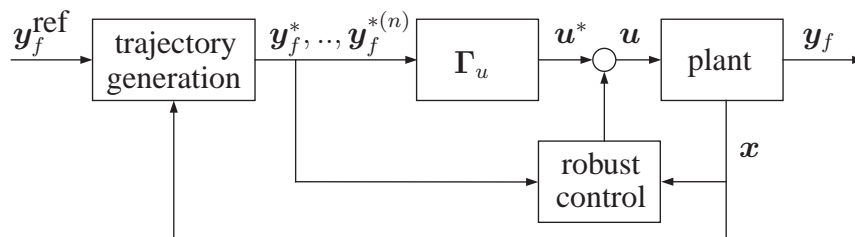


Figure 1.4: Flatness-based predictive control.

Flatness-based methods can be applied to predictive control as well, by using an inverse model such as (1.3) and (1.4) the prediction is inherent [FM00a, HD08]. The reference trajectory  $y_f^*$  is adapted to the measured states  $x$  at every sampling

step, the measurements are the initial conditions for the trajectory generation. The reinitialization of trajectories enables better adaption to uncertainties, disturbances and input saturation. The found trajectory is then applied in (1.4) to find a feed-forward signal. This mechanism can be adopted in a two-degree-of-freedom controller where it is extended with robust feedback control to account for parametric uncertainties and disturbances. The scheme provides both good performance and robustness [FM00a, HD08].

Apart from the trajectory tracking problem, trajectory generation is a major field of flatness-based control [SRA04, Lév09, GF06]. Here, again, the use of the differential parameterization (1.3) and (1.4) simplifies the task. Planning of trajectories is simplified, it can respect the dynamics of the system, but without the need of solving or integrating differential equations. The trajectory generation problem is reduced to an algebraic problem of the output function  $\mathbf{y}_f$  and a finite number of its derivatives.

The most basic method is setpoint interpolation where a predefined path is arbitrarily given, such as a polynomial over a time interval, and the initial and final conditions are imposed. Input saturations can be quite easily accounted, for instance by a time-scaling algorithm that slows down the output [FM00a]. Flatness-based trajectory generation methods come with the advantage of low complexity and computationally efficient implementation. Because of this, they represent an interesting alternative to optimization-based methods, which again involve the solution of (differential) equations. But even if optimization is applied for trajectory generation, which should be done if really good performance is required, the use of flatness is advantageous [GF06]. Especially for nonlinear systems, the reduction of the number of variables to the outputs and the inherited reduction of parameters and equality constraints simplifies the calculations.

To conclude, while flatness-based design could be applied to any control scheme because flatness is such a general property, the two fields that are in the center of *flatness-based control* are *trajectory generation* and *trajectory tracking*. These aspects can be integrated in either simple structures such as feedforward control or in advanced control systems such as predictive and optimal control systems.



## CHAPTER 2

---

### Machine models and conventional control methods

---

A prerequisite for model-based controller design is the availability of a sufficiently accurate model. It should contain all necessary information about the time-dependent behavior of the plant to be controlled. On the other hand, it should not be overloaded with details in order to keep the design procedure simple. Also, only model parameters that are simple to identify should be included.

Therefore, in these design models, secondary effects and exact details are not included. Some simplifications apply regarding the spatially distributed windings and magnetic coupling, foremost:

- The windings are assumed to be spatially sinusoidally and symmetrically distributed.
- Iron saturation and magnetical cross-couplings are neglected, only linear magnetic models are applied ( $\Psi = Li$  with  $L$  as constant).
- The resistance is assumed to be constant (i.e. no skin effect).
- Iron losses, represented by a not necessarily constant 'iron resistance' in the equivalent electric circuit, are neglected.
- The field distribution along the air gap is assumed to be spatially sinusoidal.

Some more advanced models have been designed for applications such as sensorless control, where the rotor position and speed are calculated based on the voltage commands and the current measurements. In such an application, these secondary effects may play a major role. For control, the primary effects are sufficiently well described by the model, undescribed secondary effects are compensated by robust feedback mechanisms.

The machines are modeled as continuous-time systems. Discretization of the models is then detailed, the digital control platform is analyzed in terms of the appearing delays. For the design of predictive controllers, accurate accounting of measurement and control timing is crucial.

Furthermore, the inverter is assumed to operate perfectly, i.e. with a constant source voltage, without conduction losses and without switching nonlinearities. Switching dead-times compensation methods are at a very mature stage and standard in nearly all modern devices.

For the purpose of efficiency optimization, later in chapters 5 and 6, loss models for the machines will be discussed. Inverter losses, however, consisting mostly of switching and conduction losses, will not be discussed as in small-power drives, they only have marginal influence on the results [Abr00, MYKT99].

## 2.1 Representation with space vectors

The key function of space vectors is to simplify analysis and control of the spatially and timely distributed effects in electrical machines.

Regarded are three-phase alternating-current machines. As shown in Fig. 2.1, each of the three phases  $u$ ,  $v$  and  $w$  (in the respective color green, red and blue) constitutes an independent winding in the stator. Each winding is more or less sinusoidally distributed along the air gap. To describe the electrical and electromechanical behavior of the drive, the time-varying current in the three different windings and the respective spatial distribution over the air gap must be respected. As this is a difficult task, the behavior is simplified by introducing space vectors.

First the spatial distribution of a single phase is described by a vector representing the sinusoidal current distribution, defined such that the direction is orthogonal to the maximum magnitude (For  $\mathbf{i}_u(t)$  direction  $\varphi = 0$ ) and such that the magnitude is the phase current provided by the inverter. The space vector is defined as the superposition of the three phase currents [Leo74, Sch09, KKM03],

$$\mathbf{i}_s = \frac{2}{3}(\mathbf{i}_u(t) + \mathbf{i}_v(t) + \mathbf{i}_w(t)), \quad (2.1)$$

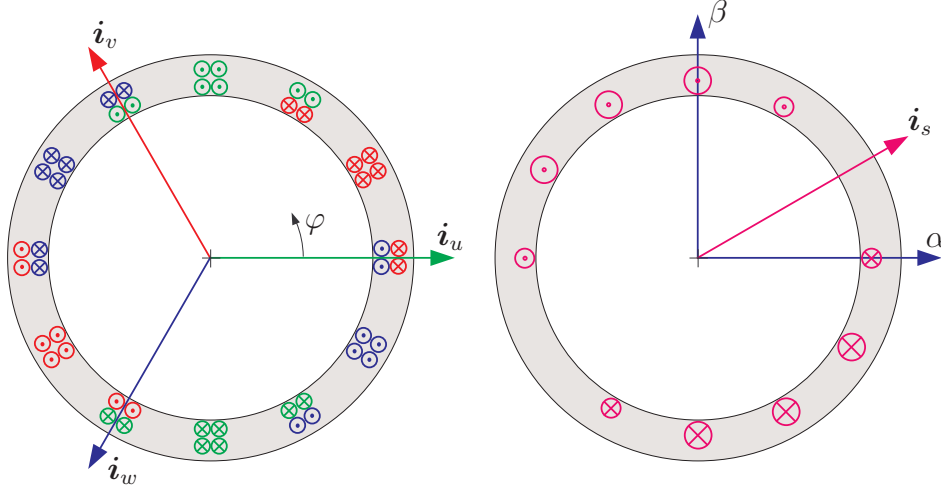


Figure 2.1: Definition of space vectors: Left: Spatial distribution of stator windings in a three-phase stator, Right: Spatial distribution of stator current vector  $i_s$ .

where the factor  $\frac{2}{3}$  is introduced for norming. If the space vector is expressed in the stator fixed  $(\alpha, \beta)$  coordinates, the  $\alpha$  axis is aligned to the  $u$  axis, and knowing that  $i_v$  and  $i_w$  are rotated by  $\pm 120^\circ$  (electrical degrees), the space vector is found as

$$\begin{pmatrix} i_{s\alpha} \\ i_{s\beta} \end{pmatrix} = \frac{2}{3} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix} \begin{pmatrix} i_u \\ i_v \\ i_w \end{pmatrix}. \quad (2.2)$$

This transformation from the three phase components  $(u, v, w)$  to the two space vector components  $(\alpha, \beta)$  is the Clarke transformation and is given by the transformation matrix

$$\mathbf{i}_{s,\alpha\beta} = \mathbf{A}_{\alpha\beta}^{uvw} \mathbf{i}_{uvw} = \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \end{pmatrix} \mathbf{i}_{uvw}. \quad (2.3)$$

Futhermore it is assumed that the three windings are in wye (Y) connection, therefore  $i_u(t) + i_v(t) + i_w(t) = 0$ . With this constraint the transformation from a three-component vector to two-component vector is uniquely invertible and it follows for the back-transformation

$$\mathbf{i}_{uvw} = \mathbf{A}_{uvw}^{\alpha\beta} \mathbf{i}_{s,\alpha\beta} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix} \mathbf{i}_{s,\alpha\beta}, \quad (2.4)$$

therefore  $\mathbf{A}_{\alpha\beta}^{uvw} = \frac{3}{2} (\mathbf{A}_{uvw}^{\alpha\beta})^T$ .

The transformation is applicable also to the voltage and flux vectors [Leo74, Sch09].

A space vector is a fictitious value for the description of a machine. Some care needs to be applied when referring to physical values. For instance, it can be shown that when power is calculated, the factor  $\frac{3}{2}$  reappears. The power absorbed by the stator windings, for instance, is

$$P_{stator}(t) = \frac{3}{2} \mathbf{u}_s(t) \cdot \mathbf{i}_s(t). \quad (2.5)$$

Similarly, the factor  $\frac{3}{2}$  appears in the description of the output torque and the power losses.

## 2.2 Model of synchronous machines

A surface-mounted permanent-magnet synchronous machine (SMPMSM) is sketched in Fig. 2.2. The permanent magnets are fixed on the rotor such that in the air gap, a more or less sinusoidal magnetic flux appears. The north pole of the rotor is orthogonal to where the highest (positive) flux density appears in the stator. Based on this simplified representation, the field-orientation can be motivated. The rotor reference frame is denoted by the two orthogonal axes  $d$  and  $q$ , where the direct  $d$  axis is aligned to the north pole position. If a current space vector is aligned to this  $d$  axis, based on the previous definition of the spatial current distribution, it can be seen that no torque is generated based on Lorentz's law. If, in contrast, a current vector is aligned to the  $q$  axis, a force is generated on each of the active current windings, and the counteracting force is the generated rotor torque  $\tau_M$ . Therefore an arbitrary stator current vector  $\mathbf{i}_s$  can be decomposed into a direct component  $i_{sd}$ , denoted as field-generating current, and a quadrature component  $i_{sq}$ , denoted as torque-generating current. The decomposition into direct and quadrature current decouples torque and flux control, the transformed dynamics on the  $q$  axis (those of  $i_{sq}$ ) are equal to those of a DC motor if the direct current is constant. As the permanent-magnet rotor flux is mechanically fixed to the rotor, the transformation angle from stator to rotor reference frame  $\varphi$  is determined from the measured rotor position. The detailed derivation of the model equations is printed in appendix C.1.

For the electrical subsystem, the voltage equation of the stator windings is given by

$$\mathbf{u}_{s,\alpha\beta} = R_s \mathbf{i}_{s,\alpha\beta} + \frac{d}{dt} \Psi_{s,\alpha\beta}. \quad (2.6)$$

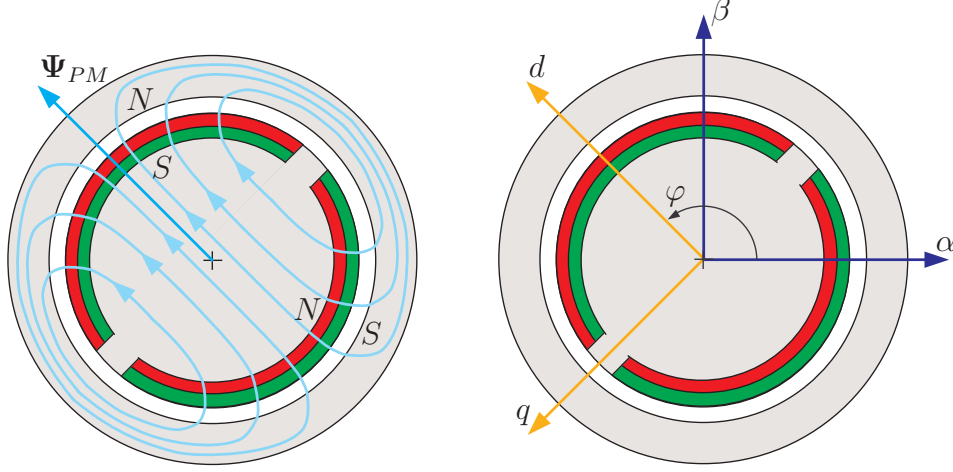


Figure 2.2: Surface-mounted permanent-magnet synchronous machine (SMPMSM): Left: Stator flux generated by PM rotor magnet, Right: direct  $d$  and quadrature  $q$  axis.

The flux linkage equation is applied. The total flux is the sum of the mutual flux generated by the stator windings and the permanent-magnet rotor flux. For this model, leakage inductances and magnetic cross-couplings are neglected. Even though the stator windings are assumed symmetrical, the stator inductance may depend on the rotor position, due to a possibly magnetically unsymmetrical rotor construction and the difference in susceptibility between iron and permanent magnet material, or due to magnetic saturation effects. The stator flux linkage is described for simplicity in the rotor reference frame and given by

$$\Psi_{s,dq} = \begin{pmatrix} L_d & 0 \\ 0 & L_q \end{pmatrix} \mathbf{i}_{s,dq} + \begin{pmatrix} \Psi_{PM} \\ 0 \end{pmatrix}. \quad (2.7)$$

The state-space model of the electrical subsystem follows as

$$\frac{d}{dt} i_{sd} = -\frac{R_s}{L_d} i_{sd} + n_p \omega_M \frac{L_q}{L_d} i_{sq} + \frac{1}{L_d} u_{sd}, \quad (2.8)$$

$$\frac{d}{dt} i_{sq} = -\frac{R_s}{L_q} i_{sq} - n_p \omega_M \frac{L_d}{L_q} i_{sd} - n_p \omega_M \Psi_{PM} + \frac{1}{L_q} u_{sq}. \quad (2.9)$$

The output equation is the generated torque

$$\tau_M = \frac{3}{2} n_p \Psi_{PM} i_{sq} + \frac{3}{2} n_p (L_d - L_q) i_{sd} i_{sq}. \quad (2.10)$$

The first component is the electromagnetic torque, based on the interaction between the permanent-magnet rotor flux and the stator current. The second is the so-called reluctance torque, it is a purely magnetic effect based on the minimum energy principle, the system will tend to the state with minimum potential energy, in this case, the rotor position where minimal magnetical energy is stored. Interestingly, both effects are covered by Lorentz's law.

To complete the state-space model, the mechanical equation

$$\frac{d}{dt}\omega_M = \frac{1}{\Theta}(\tau_M - \tau_L) \quad (2.11)$$

must be included, where  $\Theta$  is the moment of inertia and  $\tau_L$  is the load torque. In the following, this mechanical equation will not be applied for the design of the current or torque controllers. The reason is that in practice, the involved parameters are not known, for instance, the moment of inertia will change whenever a load is attached to the machine, and the load torque can be a time-varying function. Instead, in the current and torque controllers, the speed is measured and regarded as known time-varying parameter, and for speed control, experimentally tuned PI controllers will be used. This somehow corresponds to a linearization of the model.

### 2.3 Model of induction machines

The construction of the induction machine may be compared to a transformer, if the rotor is at standstill, it is nothing but two coupled three-phase windings, a stator current  $\mathbf{i}_s$  can induce a rotor current  $\mathbf{i}_r$ . If the rotor turns synchronously to the excitation field, the rotor windings are subject to a constant magnetic field and so there is no induced current in the rotor,  $\mathbf{i}_r = \mathbf{0}$ , and no torque is generated. If the rotor is rotating with a certain slip versus the excitation frequency, which is the typical case, the rotor turns and simultaneously there is an induction such that torque is generated. The detailed calculation of the model equations is printed in appendix C.2.

There are two windings to be regarded on the stator and the rotor respectively, and the corresponding voltage equations read as

$$\mathbf{u}_{s,\alpha\beta} = R_s \mathbf{i}_{s,\alpha\beta} + \frac{d}{dt} \Psi_{s,\alpha\beta}, \quad (2.12)$$

$$\mathbf{u}_{r,ab} = R_r \mathbf{i}_{r,ab} + \frac{d}{dt} \Psi_{r,ab}, \quad (2.13)$$

where the rotor voltage is  $\mathbf{u}_{r,ab} = \mathbf{0}$  as only the squirrel-cage induction motor is regarded. The index  $(\alpha, \beta)$  denotes the stator-fixed frame and  $(a, b)$  the frame mechanically fixed to the rotor.

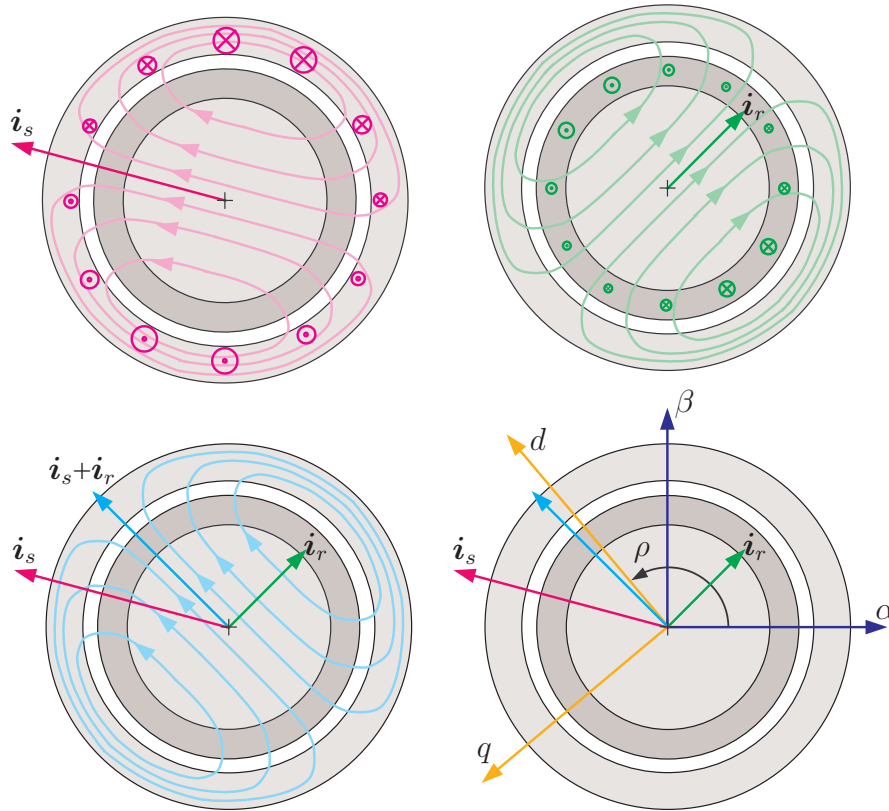


Figure 2.3: Squirrel-cage induction machine (IM): Top left: stator current and corresponding flux, top right: rotor current and corresponding flux, bottom left: Mutual flux generated by superposition, Right: direct  $d$  and quadrature  $q$  axis. The  $d$  axis is not aligned to the mutual flux because of the leakage flux.

The mutual flux in an induction machine consists of the superposition of the stator-current-generated mutual flux  $L_m \mathbf{i}_s$  (Fig. 2.3 top left) and the rotor-current-generated mutual flux  $L_m \mathbf{i}_r$  (Fig. 2.3 top right). The total mutual flux is the total flux passing through both the stator and the rotor windings (Fig. 2.3 bottom left), it is denoted as  $\Psi_m = L_m(\mathbf{i}_s + \mathbf{i}_r) = L_m \mathbf{i}_m$ . The leakage flux in stator and rotor is only passing through the respective winding and is denoted as  $L_{s\sigma} \mathbf{i}_s$  and  $L_{r\sigma} \mathbf{i}_r$ ,

respectively. The flux linkages in the stator and the rotor are therefore

$$\Psi_{s,\alpha\beta} = \Psi_{s\sigma,\alpha\beta} + \Psi_{sm,\alpha\beta} = L_{s\sigma}\mathbf{i}_{s,\alpha\beta} + L_m(\mathbf{i}_{s,\alpha\beta} + \mathbf{i}_{r,\alpha\beta}), \quad (2.14)$$

$$\Psi_{r,ab} = \Psi_{r\sigma,ab} + \Psi_{rm,ab} = L_{r\sigma}\mathbf{i}_{r,ab} + L_m(\mathbf{i}_{r,ab} + \mathbf{i}_{s,ab}). \quad (2.15)$$

The rotor flux  $\Psi_r$  reference frame is used for field-oriented control, it is denoted by the two orthogonal axes  $d$  and  $q$ , the direct  $d$  axis is about orthogonal to where the highest positive rotor flux density appears in the stator. If a current space vector is aligned to this  $d$  axis, based on the previous definition of the spatial current distribution, it can be seen that no torque is generated based on Lorentz's law, however, the flux magnitude is affected. If, in contrast, a current vector is aligned to the  $q$  axis, a force is generated on the respective current windings, and the counteracting force is the generated rotor torque  $\tau_M$ . Therefore an arbitrary stator current vector  $\mathbf{i}_s$  can be decomposed into a direct component  $i_{sd}$ , denoted as field-generating current, it is solely responsible for maintaining the rotor flux at a certain magnitude, and a quadrature component  $i_{sq}$ , denoted as torque-generating current. The decomposition into direct and quadrature current completely decouples torque and flux control, the transformed dynamics on the  $q$  axis can be shown to be equal to those of a DC motor if the rotor flux magnitude is constant.

There are four vectors, stator and rotor flux, and stator and rotor current. The stator current  $\mathbf{i}_{s,\alpha\beta}$  and rotor flux  $\Psi_{r,\alpha\beta}$  are chosen as state variables. This is sensible as  $\mathbf{i}_{s,\alpha\beta}$  is directly measured, and  $\Psi_{r,\alpha\beta}$  is directly used for the field-orientation and flux magnitude control. The stator flux  $\Psi_{s,\alpha\beta}$  and rotor current  $\mathbf{i}_{r,\alpha\beta}$  can easily be replaced by expressions of the two state vectors.

Due to the many interactions in the machine, the model reads as a set of rather complicated equations, which are simplified by some parameter definitions, typical in the literature [KKM03]. The four constants are defined by:

Inverse rotor time constant	$\eta = \frac{R_r}{L_r}$
Leakage (dispersion) coefficient	$\sigma = 1 - \frac{L_m^2}{L_s L_r}$
Coupling factor	$\beta = \frac{L_m}{\sigma L_s L_r}$
Inverse stator time constant	$\gamma = \frac{1}{\sigma L_s} (R_s + \frac{L_m^2}{L_r^2} R_r)$

With these definitions and the two vector differential equations, the state-space



model in stator-fixed frame is found as

$$\frac{d}{dt}i_{s\alpha} = -\gamma i_{s\alpha} + \beta\eta\Psi_{r\alpha} + \beta n_p\omega_M\Psi_{r\beta} + \frac{1}{\sigma L_s}u_{s\alpha}, \quad (2.16)$$

$$\frac{d}{dt}i_{s\beta} = -\gamma i_{s\beta} + \beta\eta\Psi_{r\beta} - \beta n_p\omega_M\Psi_{r\alpha} + \frac{1}{\sigma L_s}u_{s\beta}, \quad (2.17)$$

$$\frac{d}{dt}\Psi_{r\alpha} = -\eta\Psi_{r\alpha} - n_p\omega_M\Psi_{r\beta} + \eta L_m i_{s\alpha}, \quad (2.18)$$

$$\frac{d}{dt}\Psi_{r\beta} = -\eta\Psi_{r\beta} + n_p\omega_M\Psi_{r\alpha} + \eta L_m i_{s\beta}. \quad (2.19)$$

The stator current vector  $\mathbf{i}_{s\alpha\beta}$  and the speed  $\omega_M$  are measured. The voltage commands  $\mathbf{u}_{s\alpha\beta}$  are also known sufficiently well if the inverter nonlinearities are compensated. Only the rotor flux  $\Psi_{r\alpha\beta}$  is unknown, unlike in synchronous motor it is not related to the rotor position and not even to the rotor current, a model-based observer is required. The last equations (2.18) and (2.19) are denoted as 'rotor model' any may be used for open-loop rotor flux estimation [MR96b]. For improved accuracy, the first equations (2.16) and (2.17), denoted as 'stator model', are applied as well for the design of a full-order observer.

The torque is given by

$$\tau_M = \frac{3}{2}n_p\frac{L_m}{L_r}(\Psi_{r\alpha}i_{s\beta} - \Psi_{r\beta}i_{s\alpha}), \quad (2.20)$$

and the mechanical equation is

$$\frac{d}{dt}\omega_M = \frac{1}{\Theta}(\tau_M - \tau_L), \quad (2.21)$$

where  $\Theta$  is the moment of inertia and  $\tau_L$  is the load torque.

In the second step, the model is transformed into the field-oriented frame, also denoted as Park transformation. The transformation into the rotor field oriented frame is a physically motivated coordinate transformation to decouple the torque and flux dynamics. The term 'field-oriented control' denotes that a control system is designed in this reference frame, rather than referring to a specific control method, for instance cascaded PI controllers.

The coordinate frame is rotated to the synchronous frame, where the direct axis  $d$  is aligned to the rotor flux vector  $\Psi_r$ , as shown in (Fig. 2.3 bottom right). Therefore the rotor flux in the transformed coordinates is defined as

$$\Psi_{r,dq} = \begin{pmatrix} \Psi_{rd} \\ 0 \end{pmatrix}, \quad (2.22)$$

where  $\Psi_{rd}$  is the rotor flux magnitude. This is in fact a transformation from cartesian coordinates, where the flux is described by two orthogonal vector components, to polar coordinates, where the flux is described as magnitude  $\Psi_{rd}$  and angle  $\rho$ . This angle  $\rho$  is the rotation angle between the stator fixed frame and the rotor field oriented frame.

The complete model in field-oriented coordinates is given by five differential equations [KKM03, QD08]. The first three equations are explicitly used for the current and flux controller design,

$$\frac{d}{dt}i_{sd} = -\gamma i_{sd} + \beta\eta\Psi_{rd} + n_p\omega_M i_{sq} + \eta L_m \frac{i_{sq}^2}{\Psi_{rd}} + \frac{1}{\sigma L_s} u_{sd}, \quad (2.23)$$

$$\frac{d}{dt}i_{sq} = -\gamma i_{sq} - \beta n_p\omega_M \Psi_{rd} - n_p\omega_M i_{sd} - \eta L_m \frac{i_{sd}i_{sq}}{\Psi_{rd}} + \frac{1}{\sigma L_s} u_{sq}, \quad (2.24)$$

$$\frac{d}{dt}\Psi_{rd} = -\eta\Psi_{rd} + \eta L_m i_{sd}, \quad (2.25)$$

and the fourth equation, which is in general not used for field-oriented controller design, is giving the rotor flux angle derivative, or the electrical excitation frequency,

$$\frac{d}{dt}\rho = n_p\omega_M + \eta L_m \frac{i_{sq}}{\Psi_{rd}}. \quad (2.26)$$

The output equation for the torque simplifies to

$$\tau_M = \frac{3}{2}n_p \frac{L_m}{L_r} \Psi_{rd} i_{sq}, \quad (2.27)$$

and the fifth equation, the mechanical equation, is still

$$\frac{d}{dt}\omega_M = \frac{1}{\Theta}(\tau_M - \tau_L). \quad (2.28)$$

This mechanical equation will not be applied for controller design with the same arguments as in the previous section. The speed is measured and regarded as known time-varying parameter for the electrical subsystem.

Even though the induction machine is a quite complex system, control in the field-oriented frame is simple. Flux and torque can be controlled independently by imposing a current  $i_{sd}$  or  $i_{sq}$ , respectively. Furthermore, for current control, it is seen that  $u_{sd}$  affects  $\frac{d}{dt}i_{sd}$  and that  $u_{sq}$  affects  $\frac{d}{dt}i_{sq}$ , so control of the currents is somehow input independent. However, even though flux and torque are decoupled, the current dynamics are strongly interacting, and a multivariable current controller is desirable for high-performance control.

## 2.4 Systematic constraints

Designing a controller in the field-oriented frame is a fundamental choice, leading to a number systematic characteristics. The main aspects of this design compared to direct and stator-oriented control designs is well described in the literature [Qua93, QD08, Leo74, Sch09].

Control in the field-oriented frame is inherently simple and intuitive. A continuous input voltage is generated by a modulation scheme and an inverter. The key strengths are:

- Simple decoupling of torque and flux magnitude control by dividing the current vector in two orthogonal direct  $d$  and quadrature  $q$  components. Each current component will only influence one output.
- High steady-state accuracy as the controller does not operate in transient operation. In the  $(d, q)$ -frame, at constant speed and torque, all states (voltages, currents, flux magnitude) are constant.
- Independent control of both  $d$  and  $q$  axis currents can be guaranteed by implementing independent saturation functions, such that voltage saturation on one axis shall not influence control of the other axis. In most practical cases, only the voltage in  $q$  axis will be close to its saturation limit because of the back-EMF, and a well-designed saturation function ensures that sufficient voltage is present on the  $d$  axis for good control. This simplifies operation close to the saturation limits, and in high-speed operation, this prevents misorientation and ensures safe torque and flux decoupling. Obviously, for the rare case where voltage saturation appears on both axes, this advantage disappears.
- A constant and well-defined switching frequency ensures low noise in all operation modes.

However, control in the field-oriented frame with continuous voltage commands also has some disadvantages, not appearing for instance in direct control schemes that directly specify the inverter switches without modulation:

- Conventional direct control (DTC) is faster than conventional control with a modulator (cascaded PI control), as 'full' voltage can be applied throughout a sampling interval instead of switching patterns that fall back to zero-voltage.
- While torque and flux are decoupled, the current dynamics themselves are strongly coupled. This coupling may be more intense and more nonlinear in the field-oriented frame, for instance in induction machines.

- Limitations in the measurement circuit, such as analog-digital conversion latency in the current measurements or switching dead times, limit maximum modulation and thereby the maximum applicable voltage. Obtaining full voltage is technically possible, however, hard to implement [QD08]. This problem does not appear in DTC (and PTC).
- The voltage saturation is not naturally included in the controllers. The voltage commands generated by a controller could be higher than the maximum inverter output, as a result, the output does not follow as foreseen, and integrators in the controller further increase the voltage command. This effect results in a controller wind-up. To prevent this, anti-windup mechanisms and artificial voltage command limitations must be implemented.

Here, direct control is not discussed (except in appendix E which is a follow-up to chapters 3 and 4). Each method has its advantages and issues, pointed out for instance in [RKE<sup>+</sup>12]. For small-power drives, however, field-oriented control is the standard vector control principle.

## 2.5 Model discretization and delay compensation

The models derived in the previous sections are continuous-time models. In chapters 3 and 4, however, discrete-time controllers are designed. For this purpose, after an analysis of the digital control platform and measurement and control timing, the models are discretized.

A digital current control loop on the used standard interrupt-synchronized digital control platform has a delay of two sampling steps [MKY03]. This is seen in the timing diagram in Fig. 2.4. The interrupt-based control system triggers an interrupt at every sampling step. The interrupt handling is quite slow due to the system latencies and the slow A/D converter. The actual voltage command  $\mathbf{u}[k]$  is modulated at the marked instants  $k$  and the switching times determined for the power electronics. After that, the interrupt handling software is started and the A/D-conversion of the current measurements  $\mathbf{i}[k]$  is immediately performed to avoid the impact of current ripples on the measurements. Once the current sample  $\mathbf{i}[k]$  is available, the interrupt handler finishes and the control algorithm is started to compute the next voltage command  $\mathbf{u}[k + 1]$ , this signal must, however, wait until the next interrupt  $k + 1$  to be modulated, the switching times cannot be changed between the interrupts.

This means that the voltage  $\mathbf{u}[k + 1]$  has been calculated with an old current sample  $\mathbf{i}[k]$ . This delay of measurement from  $\mathbf{i}[k]$  until the application of control

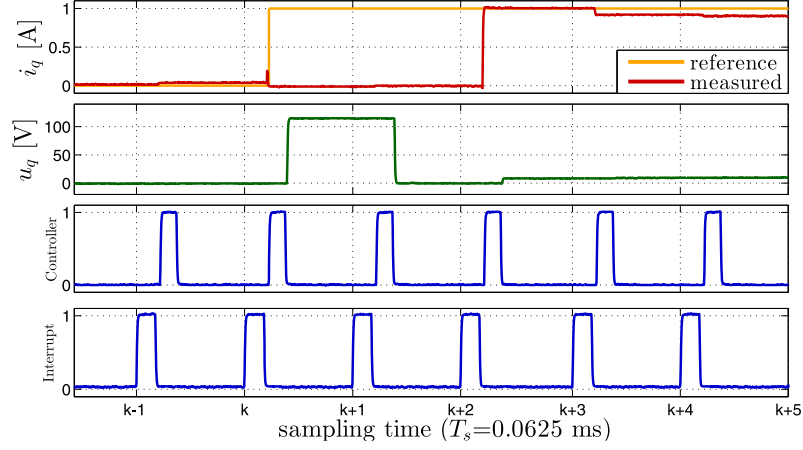


Figure 2.4: Timing of interrupt, control, voltage commands and current measurements of a digital control system.

action  $\mathbf{u}[k + 1]$  to the system is denoted *computational delay*. It has to be compensated, otherwise, oscillations will appear in a closed-loop predictive control system as seen in Fig. 2.5 (a). Here, after the reference step, 140 V are applied, and as the feedback signal shows no change after one step, the voltage is still applied in the next step. The delay leads to the fact that any control error is compensated by a voltage command of correct magnitude, but of double duration than required, this results in an oscillation of the output even though the controller is correctly tuned. The delay must be compensated, the technique is straightforward, the current of the next step  $k$  is predicted at the time  $k - 1$  by integrating the discrete-time system model

$$\mathbf{i}[k|k - 1] = \mathbf{f}(\mathbf{i}[k - 1]) + \mathbf{B}\mathbf{u}[k - 1], \quad (2.29)$$

and this prediction is applied as feedback to the control algorithm to calculate the next control action  $\mathbf{u}[k]$ . This implementation is possible as  $\mathbf{i}[k - 1]$  and  $\mathbf{u}[k - 1]$  are known during the time of execution. In Fig. 2.5 (b) the predicted (blue) current signal  $\mathbf{i}[k|k - 1]$  is applied for feedback instead of the measured current  $\mathbf{i}[k - 1]$  (red) to avoid the mentioned problem.

The technique is known as Smith-predictor. Although the one-step delay is a state, and the system therefore a second-order system, the current control loop is assumed as first-order system for the design of a current controller. With this technique, the delay can be ignored for controller design. In practice, the reference is delayed by one sampling interval, such that two steps delay are still present between reference and output.

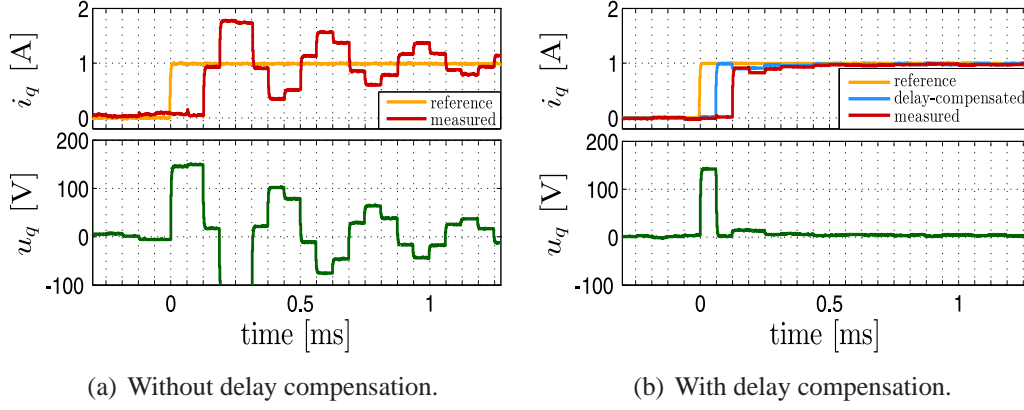


Figure 2.5: Experimental results: Deadbeat control without and with delay compensation.

In Fig. 2.4, it is further seen that the response to a modulated voltage peak  $\mathbf{u}[k + 1]$  is not directly seen at the output but only appears one step later at  $\mathbf{i}[k + 2]$ . This is denoted as *plant delay*, it is respected in the predictive control system if the continuous model is linearized using Euler forward discretization. Accordingly, the continuous time current variation is redefined such that it reflects the physics,

$$L \frac{d}{dt} \mathbf{i}(t) = \mathbf{u}(t) \Rightarrow L(\mathbf{i}[k + 1] - \mathbf{i}[k])/T_s \approx \mathbf{u}[k]. \quad (2.30)$$

The Euler approximation of the continuous derivative generates a lag error that corresponds exactly to the delay in the real system, and the step response of the designed predictive (deadbeat) controller is as desired as seen in Fig. 2.5 (b).

Another assumption that is done for discretization is that the open-loop time constant  $\frac{L}{R_s}$  is much larger than the sampling interval  $T_s$ . During the sampling interval, the response of the current to a constant input voltage is assumed to be of constant slope. The real response is asymptotic, however, assuming the voltage  $\mathbf{u}$  is constant, the continuous-time current response is

$$\mathbf{i}(t) = \mathbf{i}_0 + \left( \frac{\mathbf{u}}{R_s} - \mathbf{i}_0 \right) \left( 1 - e^{-\frac{R_s}{L}t} \right), \quad (2.31)$$

and by assuming  $e^{-\frac{R_s}{L}T_s} \approx (1 - \frac{R_s}{L}T_s)$ , valid if the expression  $\frac{R_s}{L}T_s$  is small, the above described discretization appears,

$$\mathbf{i}[k + 1] = \left( 1 - T_s \frac{R_s}{L} \right) \mathbf{i}[k] + \frac{T_s}{L} \mathbf{u}[k]. \quad (2.32)$$

For systems with low sampling rates or without delay, other techniques exist, foremost based on approximating the continuous-time response analytically [LKKS10].

## 2.6 Limitations of linear flux models

In the deduction of the models in the previous sections, a linear flux model has been assumed. However, due to size and cost constraints in the machine construction, magnetic saturation effects appear and inherit flux nonlinearities. Two sketches are shown in Fig. 2.6 for typical drives, left for a PMSM and right an IM, both in the direct axis.

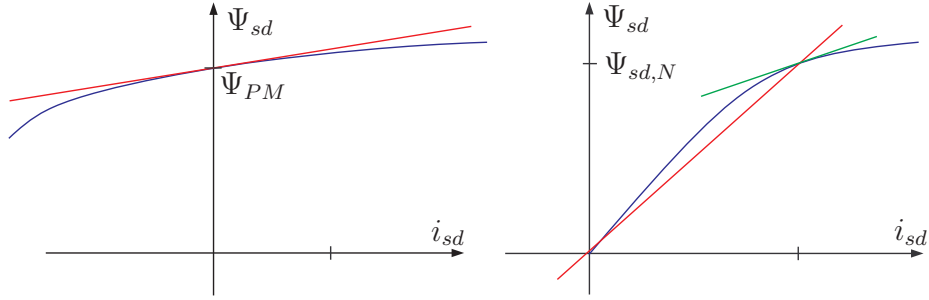


Figure 2.6: Saturation effects in electrical machines, sketch. Left: permanent-magnet synchronous machine, saturation is disregarded in the linear flux model (red). Right: induction machine, saturation is accounted by adjusting the secantial inductance in rated operation (red), the tangential inductance (green) will have a different value.

Saturation effects result in current-dependent inductances [GGPL98]. In drive systems, a convention has appeared that transforms the problem of one nonlinear flux curve in two linear models with setpoint-dependent inductances. These are:

- The tangential inductance (or dynamic inductance)  $L^T$ . It describes the small-signal behavior at a setpoint,

$$L^T = \frac{\partial \Psi}{\partial i} \quad \Rightarrow \quad u = L^T \frac{d}{dt} i. \quad (2.33)$$

- The secantial inductance (or static inductance)  $L^S$ . It describes the large-signal behavior at a setpoint,

$$L^S = \frac{\Psi}{i} \quad \Rightarrow \quad \Psi = L^S \cdot i. \quad (2.34)$$

To account for this saturation, lookup tables must be identified and stored in the real-time system. The model derivation must account for the two different inductance values. This makes controller design complicated and is only done if really necessary, for instance in sensorless control applications, or for synchronous reluctance machines.

In this thesis, saturation effects are not accounted in the design, the controller design is made robust against this uncertainty source. As shown in Fig. 2.6, the inductance values for PMSMs are identified at zero current, thus ignoring any saturation influence. The inductance value for the IM is the secantial inductance at rated flux, such that in rated operation, the secantial inductance is correct. This is important for correct flux calculation by the observer, and is also the industrial standard.

## 2.7 Conventional control: Optimized PI controller

Often it is assumed that a predictive controller is less robust than a feedback controller [PLR05]. This is true for very slow PI controllers, however, as inductance uncertainties result in dynamic gain uncertainty, PI controllers also suffer from such uncertainties. This section analyzes parametric robustness of a fast PI regulator, tuned according to the symmetrical optimum criterion. The tuning rule is quite widespread in field-oriented control and should give a fair comparison to the later presented model-based designs, especially as good disturbance response is claimed [Sch09]. The evaluation is based on the PMSM described in appendix B.1.

### Controller tuning

The continuous design starts by approximating the transfer function of the plant by

$$G(s) = \frac{I(s)}{U(s)} = \frac{\frac{1}{R}}{(1 + sT_\sigma)(1 + s\frac{L}{R})}, \quad (2.35)$$

where  $s$  is the Laplace operator,  $\frac{1}{R}$  the DC gain,  $\frac{L}{R}$  is the open-loop time constant and  $T_\sigma$  is the time constant that approximates the delay consisting of computational and plant delay. In this design, the delay of two sampling steps is thereby approximated as lowpass filter with a time constant  $T_\sigma \approx 2T_s$  as proposed in [Sch09].

Very often, current controllers are tuned according to the magnitude optimum criterion. Both designs can be applied to the underlying dynamics and lead to a PI controller. Which design should be used depends on the machine parameters [Sch09]. For the case that the open-loop time constant is smaller than four times the delay,



$\frac{L}{R} < 4T_\sigma$ , the magnitude optimum criterion is preferable because it does not need a prefilter and has a similar disturbance rejection as the symmetric optimum criterion. In the other case where the open-loop time constant is larger,  $\frac{L}{R} > 4T_\sigma$ , the symmetric optimum criterion will have a considerably better disturbance rejection. For instance, in the presented experimental example, the symmetric optimum criterion leads to a disturbance settling time of 1.25 ms compared to 5.44 ms with the magnitude optimum criterion. The required reference prefilter is not a problem for cascaded control.

As the open-loop time constant is considerably higher than the delay, the resistance is ignored in the design, and the optimum continuous-time controller according to [Sch09] is

$$R(s) = \frac{U(s)}{I^*(s) - I(s)} = V_R \frac{1 + sT_n}{sT_n}, \quad (2.36)$$

where the tuning rules prescribe  $T_n = 8T_s$  and  $V_R = \frac{L}{4T_s}$ . The proportional part is responsible for fast reference tracking, the integrator compensates steady-state offsets, and there is no need for a delay compensation technique. Even though the model parameters are used in the design, there is no explicit usage of the model in the control, the PI controller is not of the class of model-based controllers.

### Robustness analysis

The influence of a difference between correct inductance  $L$  and inductance assumed for gain tuning  $\hat{L}$  is analyzed. The continuous controller is discretized (Euler approximation) and its closed-loop behavior is calculated based on the discrete model. The discrete-time transfer function of the closed loop control system is

$$G(z) = \left\{ \frac{I(z)}{I^*(z)} \right\} = \frac{b_2 z^2 + b_1 z}{a_3 z^3 + a_2 z^2 + a_1 z + a_0}, \quad (2.37)$$

where

$$\begin{aligned} a_3 &= 32, & b_2 &= 9\frac{\hat{L}}{L}, \\ a_2 &= -64 + 32T_s\frac{R}{\hat{L}}, & b_1 &= -8\frac{\hat{L}}{L}, \\ a_1 &= -32T_s\frac{R}{\hat{L}} + 9\frac{\hat{L}}{L} + 32, \\ a_0 &= -8\frac{1}{L}\hat{L}. \end{aligned}$$

Fig. 2.7 shows the  $z$ -space polemap deduced from this transfer function for an inductance uncertainty  $\Delta = \frac{\hat{L}-L}{L}$ . It is already seen that at nominal parameters, the bandwidth of this controller quite low. Sensitivity to  $L$  is high, the stability limit is at  $\Delta = 1.7$ , and any  $\Delta > 0$  inherits oscillations. For  $\Delta < 0$  a slowdown appears.

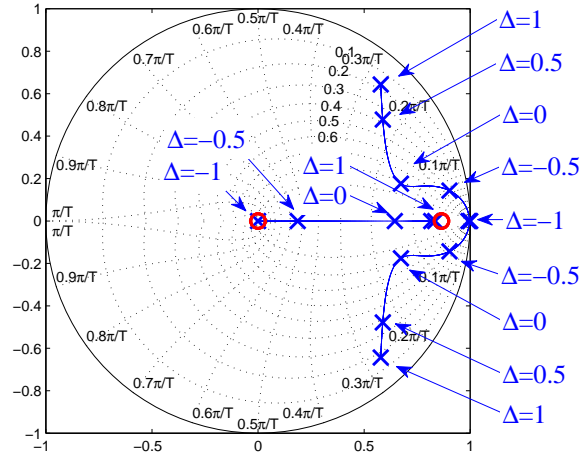


Figure 2.7: Discrete frequency domain polemap of PI controller tuned with symmetric optimum criterion under inductance uncertainty.

If the inductance  $\hat{L}$  is assumed too high, the command voltage is too high as the open-loop dynamic gain of the plant  $\frac{1}{L}$  is assumed too small.

Fig. 2.8 shows experimental results. In subfigure (a) the response with nominal parameters is shown, the characteristic overshoot of 40.4% appears as no reference prefilter was applied. Subfigure (b) shows the response with  $\Delta = 1$ , or  $\hat{L} = 2L$ , a badly damped oscillation and a much stronger overshoot appear. In subfigure (c), an underestimated inductance  $\Delta = 0.5$ , or  $\hat{L} = \frac{1}{2}L$ , is applied, the settling time is thereby doubled. These results are all a zero speed, where no disturbance appears. Subfigure (d) shows the response at 2000 rpm. The current has a strong ripple of 600 Hz, these are harmonic effects caused by the non perfectly sinusoidal distribution of the stator windings, six times the excitation frequency which is 100 Hz. The magnitude of the ripple is about 140 mA (RMS) on the  $q$ -axis and 50 mA on the  $d$ -axis.

Thus the fast PI controller is sensitive to modeling errors. Furthermore the disturbance bandwidth of the PI controller is insufficient to compensate flux harmonics, as already remarked in [SH98]. Both characteristics, tracking performance and disturbance rejection, are not really satisfactory and could be improved.

### Performance improvement: Two-degree-of-freedom control

The tracking performance can be improved by using a two-degree-of-freedom control structure. As described in Fig. 1.3, the tracking performance is given by a

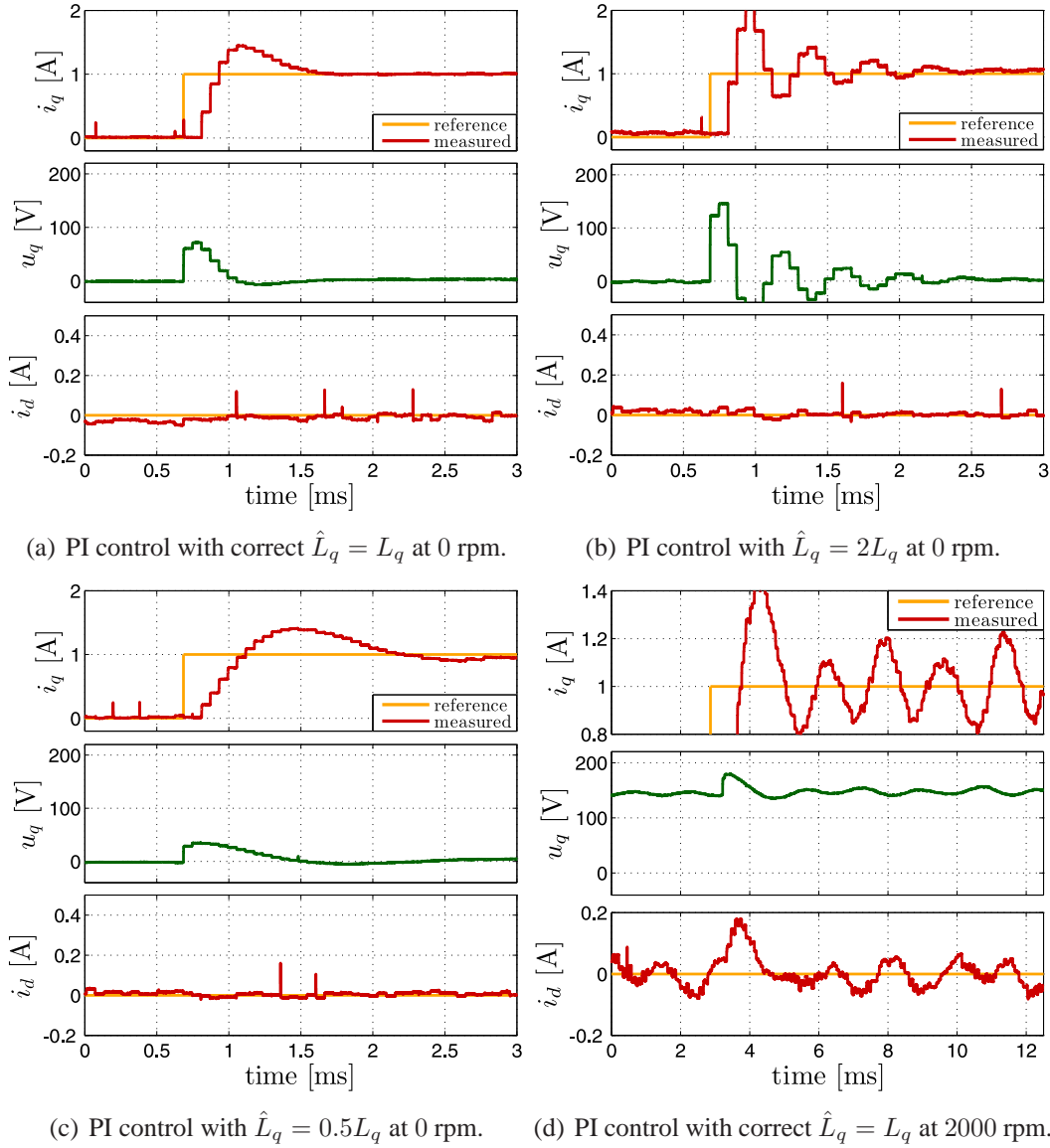


Figure 2.8: Experimental results: PI controller tuned with symmetric optimum criterion.

feedforward controller. The PI controller compensates the difference between the reference and measured trajectory, knowing that the system has a delay of two steps, the reference is delayed accordingly for the PI controller. The experimental results are shown in Fig. 2.9, the tests are identical to those of the PI controller. Subfigure (a) shows a very satisfactory reference response, the feedforward path is good

such that the PI controller only has a small task. Under uncertainties, however, the feedforward controller does not operate precisely, and the PI controller, also subject to the uncertainty, has to compensate the offset. In subfigure (b) the error of the feedforward controller is very high, and the PI controller does not operate safely as the stability limit is very close. The response is subject to the same oscillations as without feedforward control. In subfigure (c), both control paths have a too small action. Subfigure (d) shows that the disturbance response is identical to the sole PI controller.

To conclude, the reference tracking performance of the two-degree-of-freedom controller is improved compared to the sole PI controller. However, the results are still not satisfactory, especially under uncertainties. The parametric sensitivity is high and the disturbance rejection capability is low. This motivates the following work, which aims at obtaining a good reference tracking, even under uncertainties, and simultaneously, a strong disturbance rejection to compensate the effects of the flux harmonics.

## 2.8 Flatness of electrical drives

Virtually all electrical drives are differentially flat systems. This includes the DC machine [SRA04], PMSMs [SRA04], induction machines [MR96b], and some more exotic designs such as linear torquers or magnetic bearings [vL02].

Control of electrical drives is of high practical importance, and more than that, they are interesting from theoretical point of view. This has motivated a number of research projects in both the electrical drives and the control systems communities. The induction machine is a high-order nonlinear multivariable system. The two outputs strongly interact and nonlinear terms appear, and the flux dynamics are of second order. Additionally, one of the outputs, the flux, can not be measured and must be calculated with an observer. As multivariable system, its flat output vector is not unique but there is a certain choice, for instance:

- The rotor flux magnitude  $\Psi_{rd}$  and the torque  $\tau_M$  (or the speed  $\omega_M$  or rotor position  $\varphi_M$ , depending whether one discusses torque, speed or position control).
- The rotor position  $\varphi_M$  and the 'slip' angle  $\delta = n_p \varphi_M - \rho$ , i.e. the angle of the rotor flux with respect to the rotor-fixed  $(a, b)$ -frame [MR96b, DLO01].
- The output to be controlled (either  $\tau_M$ ,  $\omega_M$  or  $\varphi_M$ ) and the 'controllable' losses  $P_{loss}$  (defined in chapter 6).

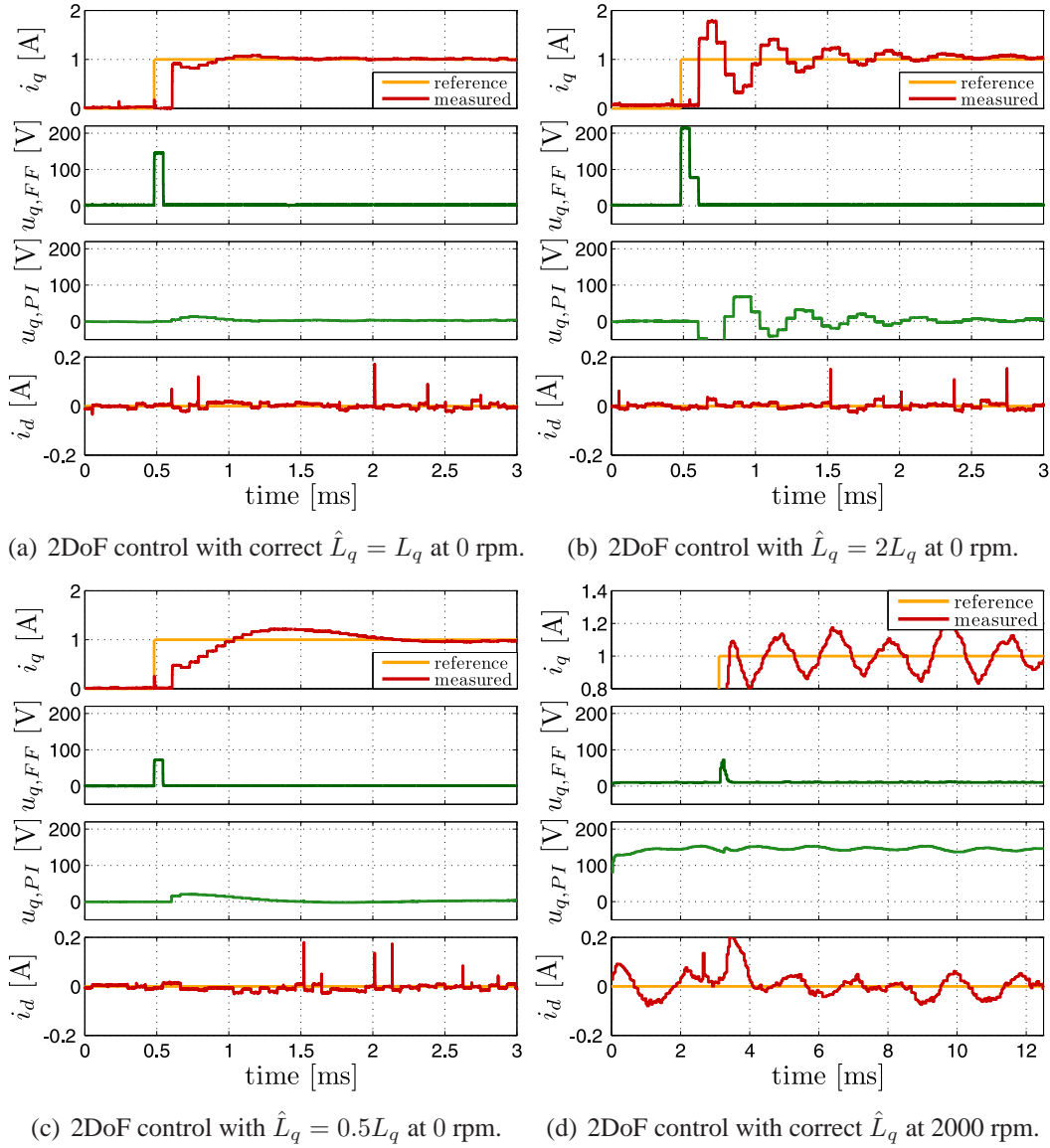


Figure 2.9: Experimental results: Two-degree-of-freedom (2DoF) controller using a PI controller tuned with symmetric optimum criterion.

Which flat output is the best choice is mostly matter of the control goal. It may be possible to avoid nonlinearities in the design by choosing an appropriate output. In most cases, however, one should choose a vector with physical meaning to simplify the design task.

Many flatness-based control schemes were proposed for the induction machine.

A general exposition of its flatness is found in [DLO01]. Some works were done on discretization and discrete-time control of the nonlinear model [MR96a], as well as on improved rotor flux observers [MR96b]. Efficiency-optimal control of the squirrel-cage IM is described in [HRD03] and of the doubly-fed induction generator in [Gen08]. It also served as benchmark example to study robustness of predictive control in [HD08].

The synchronous machine is a much simpler system, all states can be measured if a position encoder is available. Again, as multivariable system, its flat output vector is not unique but can be chosen from:

- The field-generating stator current  $i_{sd}$  and the torque-generating stator current  $i_{sq}$  (or the speed  $\omega_M$  or rotor position  $\varphi_M$ ) [SRA04].
- The field-generating stator current  $i_{sd}$  and the torque  $\tau_M$  as sum of electromagnetic and reluctance torque (or the speed  $\omega_M$  or rotor position  $\varphi_M$ ) [DS04].
- The stator current magnitude  $\sqrt{i_{s\alpha}^2 + i_{s\beta}^2}$  and the stator current angle  $\arctan 2(i_{s\beta}/i_{s\alpha})$  [SRA04] in the case of torque or current control.

The first choice is the simplest because it relies on the variables applied in any field-oriented controller. Latter choice of stator current magnitude led to the successful design of a 'nonlinear position estimator' for back-EMF-based sensorless control.

For the synchronous machine, a quite small number of flatness-based controller designs have been proposed. It seems as the simplicity of this drive prevents the interest of control engineers with a more theoretical orientation. Nevertheless, PMSM control design can be challenging, especially for advanced and robust algorithms. A maximum-torque-per-ampere controller accounting for reluctance torque is presented in [DS04] and robust tracking control has been studied in [LML03].

Apart from current and torque control, higher-order systems have been studied, for instance resonant loads [TF11] or the combination of drives and boost converters [AD06].

## CHAPTER 3

---

### **Robust flatness-based predictive control: Deadbeat current control for AC drives**

---

Predictive control is a powerful control method that makes extensive use of the model information. The closed-loop performance can be designed to be close to the physical limits of the controlled system. A major restriction, however, is the high sensitivity to errors in the model. This chapter demonstrates a method inspired from flatness-based control which improves robustness without decreasing control performance. The scheme is applicable to many other predictive control systems.

Deadbeat control is chosen to present the method. Deadbeat control is considered as one of the fastest current control schemes, it provides extremely fast control and simple straightforward design. It only requires a simple model of the current dynamics, consisting of the time constant, the gain and cross-coupling. Problems in the design are, however, delays, disturbances, and high sensitivity to modeling errors and noise. Since deadbeat control is a very aggressive scheme, these problems require more consideration than if conventional control schemes are applied. Indeed, these disturbances prevent an industrial application of deadbeat control. So far it is only used in few applications such as active power filters or uninterruptible power supplies, but in industrial drives, although it is useful, it is, with just a few exceptions [Nus99], mostly used in laboratories. Improving robustness of predictive controllers and more specifically of deadbeat control is a major point in transferring this theory to practice.

## 3.1 Preliminaries

### 3.1.1 Deadbeat current control for AC drives: State of the art

The first designs of deadbeat current control have been published already around 40 years ago, but research is still active. Problems of the design are, as mentioned, delays in the digital system, leading to oscillations, modeling errors, leading to performance reduction or instability, and the high sensitivity to noise. The deadbeat controller has been sequentially improved, step by step.

While the first works were designed on special controller hardware, in order to use more standard hardware, the delay problem was addressed. A flux and current observer with adaptive poles was proposed in [BBK92] to avoid the oscillations caused by the delay, shown in Fig. 2.5. As alternative, model-free delay compensation was proposed, the current is predicted based on second-order interpolation [KMY90] or on Lagrange interpolation [Kuk96] using the past current measurements. The interpolation method is robust to parameter uncertainties and solves the problem in steady-state, however, not in transient operation. The model-based delay compensation was finally proposed in [MKY03], already presented in section 2.5. It is the most simple, performant and most common method, today applied in almost all deadbeat and predictive control schemes. Any well-known delays in the system can be compensated by prediction, including for instance the position measurements for the synchronous frame transformation [MKY03], further improving performance.

The sensitivity to model errors was addressed as well, mostly aiming at removing steady-state offsets. The introduction of a disturbance estimator, named 'input delay approach' and acting like an integral term that compensates the model errors and disturbances, is of very high relevance. The original presentation was on special hardware without delay [KY01], but the same design can be easily repeated to account for the delay, and steady-state accuracy is improved. The system with disturbance estimator provides good parametric robustness to the resistance and the back-EMF, however, inductance uncertainty remains a problem [YL02], the stability limit is at 100% parameter error. This means if the estimated inductance has double value than the real inductance, the system is unstable. Other problems are the line voltage, as the stability limit was reported at 20%, an estimation method that additionally accounts inductance and inverter effects was proposed [MMB99]. Using such disturbance estimators, the steady-state error is practically zero.

Time-varying disturbances, such as those arising from flux linkage harmonics, are not well treated with the disturbance estimator. The disturbance terms are assumed constant in the disturbance estimator design (random walk model) [KY01],



furthermore, the estimated disturbance is lowpass-filtered to reduce dependence on model parameters (time-scale decoupling). For instance in [KY01] the disturbance cutoff frequency is at 318 Hz for a PMSM excited at 100 Hz, such cutoff frequencies are typical in the respective literature. In this case, the bandwidth of the disturbance estimator becomes too small and time-varying disturbances cannot be compensated. Such effects arise from nonperfect stator construction, mainly the non-perfectly sinusoidal distribution of the windings. In the stator frame, these imply a third, fifth and seventh harmonic on the respective phase current or voltage [ML92]. In the space vector representation in the synchronously rotating  $dq$ -frame, it can be shown by calculation [Wal01] and measurement [HS96, SH98] that the main components are 6 and 12 times the fundamental excitation frequency. It is possible to measure this high-frequency disturbance and compensate it [SH98], an additional voltage signal dependent on the speed and the rotor position is applied. To bypass the limitations of this off-line method with fixed parameters, on-line torque estimation combined with fast torque control has been proposed [CKKY98]. If the mechanical system is well known and the sensors are sufficiently good, the torque ripple is considerably suppressed.

Steady state accuracy and ripple can be compensated, however, the problem of sensitivity to inductance remains very strong. The deadbeat condition states that the control error should vanish in the next sampling step, making this controller very aggressive. The objective can be reformulated to make the controller less aggressive and therefore also less sensitive. In [BLNH05] the design condition is that the control error is reduced by 50% in the next sampling step, resulting in asymptotic convergence. In [Qua93] a finite settling time of not one, but two or three steps is proposed. It is well known that the robustness is increased by making the deadbeat interval longer than the order of the system [Föl85]. Therefore, both methods render the system less sensitive, and errors around 50% in the inductance are no problem.

Furthermore, depending on the operation condition, due to the voltage saturation, the deadbeat condition of zero error in the next sampling step may not be practicable. Various voltage saturation methods are discussed in [OS02]. It is important to regard the voltage saturation for the delay compensation algorithm, any difference between voltage command and applied voltage leads to errors. Several methods are discussed regarding the two main aspects, which are independent control of both current components and fast response. Simple saturation of the axis guarantees independent control, but less restrictive methods exist, namely dynamic vector limitation or methods regarding the multivariable dynamics of the drive.

Interestingly, there are only few works on online parameter identification. Although deadbeat control and online identification is possible [JP97], simultaneous estimation of disturbance and parameters is not directly possible. A heuristic

method was proposed, in [MES07] the deadbeat controller is started with nominal parameters and the correct parameters estimated offline, then, at a certain instant, the parameter set is updated. The absence of real online-identification on the given system, however, strengthens the motivation to improve robustness.

Alternative fast control schemes to deadbeat control were and are still developed. Using specialized hardware, such as field programmable gate array (FPGA) and an extremely fast A/D converter that samples 10 times faster than the controller, the inductance and back-EMF can be identified online from the measured current slopes [WB08, FBB12]. There, it was shown that simultaneous estimation of parameters and disturbances is possible, a fundamental limitation of deadbeat control when implemented on standard hardware [JP97]. However, apart from the expensive electronics, this interesting scheme cannot yet handle rotor anisotropies or non-linear cross-coupling terms, therefore it cannot handle interior permanent-magnet SMs or induction motors in field-oriented frame. Furthermore, the maximum output voltage is limited to the range 10 – 90% as a rising and falling slope is required within each sampling interval. Nevertheless, online parameter identification is an upcoming topic.

Deadbeat direct torque control [KL03] was proposed, the machine equations are solved in similar way as for the deadbeat design, the analytic version of the well-known direct torque control method (DTC) simplifies the incorporation of current and voltage constraints and gives the possibility to combine injection-based sensorless methods. Indeed, the deadbeat design method, because of its simple and straightforward analytic design, is the core of many control systems, such as adaptive controllers etc.

Because of the simple design, however, deadbeat control is still popular. Simplicity and the fact that the method is easy to embed in existing field-oriented controllers on an arbitrary drive might be the major advantages of the conventional deadbeat controller.

The goal of this chapter is to solve the delay, noise and parameter sensitivity problem. The standard delay compensation [MKY03] and disturbance estimator [KY01] will be applied as they are simple to implement. Performance should be maximum, the deadbeat condition that control error vanishes at the next step is maintained. Furthermore, the disturbance estimator is tuned so fast that it also compensates the named harmonics and current ripples additionally to the steady-state offset. These criteria lead to an extremely fast but also sensitive control system. Using the results from flatness-based control on robustness, however, the deadbeat controller can be made considerably more robust while maintaining the superior control performance.

### 3.1.2 Basic deadbeat current controller design

A discrete-time model of the current dynamics is written as

$$\mathbf{i}[k+1] = \mathbf{f}(\mathbf{i}[k]) + \mathbf{B}\mathbf{u}[k], \quad (3.1)$$

where  $\mathbf{i}[k]$  is the current at instant  $k$ ,  $\mathbf{u}[k]$  the voltage,  $\mathbf{f}(\mathbf{i}[k])$  the current dynamic vector and  $\mathbf{B}$  the input matrix. The output is the current  $\mathbf{i}[k]$ . The deadbeat condition is

$$\mathbf{i}[k+1] = \mathbf{i}^*[k], \quad (3.2)$$

which defines the desired closed-loop performance. The linearizing input  $\mathbf{v}[k]$  is found as  $\mathbf{v}[k] = \mathbf{i}[k+1]$ , or, with (3.1),

$$\mathbf{v}[k] = \mathbf{f}(\mathbf{i}[k]) + \mathbf{B}\mathbf{u}[k], \quad (3.3)$$

where the deadbeat condition leads to the control law

$$\mathbf{v}[k] = \mathbf{i}^*[k]. \quad (3.4)$$

With the deadbeat condition and by rearranging (3.1), the physical realization of the deadbeat control law is

$$\mathbf{u}[k] = \mathbf{B}^+ \mathbf{i}^*[k] - \mathbf{B}^+ \mathbf{f}(\mathbf{i}[k]), \quad (3.5)$$

where  $\mathbf{B}^+$  is the pseudo-inverse of the input matrix  $\mathbf{B}$ . With the delay compensation technique of section 2.5, as  $\mathbf{i}[k]$  is not available, it becomes

$$\mathbf{u}[k] = \mathbf{B}^{-1} \mathbf{i}^*[k] - \mathbf{B}^{-1} \mathbf{f}(\mathbf{i}[k|k-1]). \quad (3.6)$$

Using this controller, the control error is eliminated in one step.

### 3.1.3 Overall control structure

The structure of the state-of-the-art deadbeat controller with delay compensation, disturbance estimator and saturation is sketched in Fig. 3.1. All variables such as  $\mathbf{i}[k]$  and  $\mathbf{u}[k]$  are referred to in field-oriented coordinates. The deadbeat controller is a predictive controller, as such, it does not have an integral action. Unmodeled disturbances lead to steady-state offsets, because of this, the extension with a disturbance estimator is necessary. The delay compensation technique has been presented in section 2.5, the disturbance estimator and saturation are presented in the following.

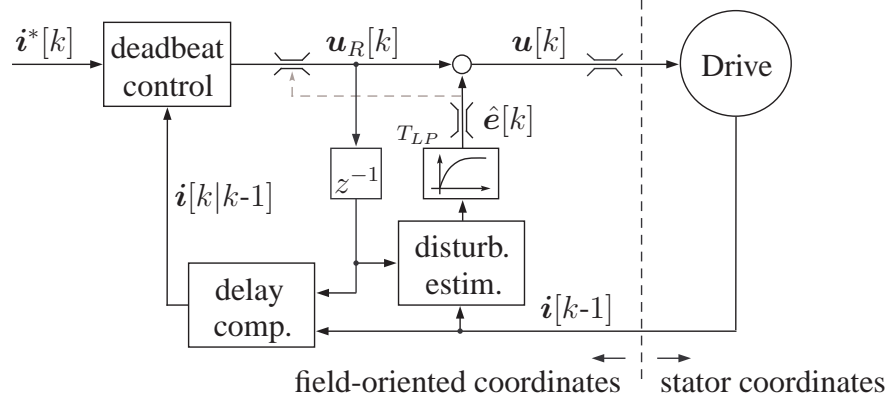


Figure 3.1: Control structure of a conventional deadbeat controller.

### Disturbance estimator

The current control loop is subject to a number of disturbances. To the major part, this is the induced voltage, consisting of back-EMF terms. Additionally, if the cross-coupling and voltage drop terms are not correctly modeled due to modeling uncertainties, these terms are also seen as a disturbance, just like any unmodeled terms in the right hand side of the model. In section 2.7 it turned out that flux harmonics lead to a time-varying back-EMF and also represent an important disturbance. Generally, such disturbances do not lead to stability problems, but to offsets between the measurements and the references, and to current ripples.

The full model of the current dynamics is written as

$$\mathbf{i}[k+1] = \mathbf{f}(\mathbf{i}[k]) + \mathbf{B}\mathbf{u}[k] - \mathbf{B}\mathbf{e}[k], \quad (3.7)$$

where the disturbance  $\mathbf{e}[k]$  is assumed quasi-constant. The control input is divided in two parts

$$\mathbf{u}[k] = \mathbf{u}_R[k] + \hat{\mathbf{e}}[k], \quad (3.8)$$

where  $\mathbf{u}_R[k]$  is the control input of the deadbeat controller and  $\hat{\mathbf{e}}[k]$  the estimated disturbance. The disturbance estimator of choice is based on the well-known input delay approach [KY01], redesigned accounting for the computational delay. The disturbance is calculated from the model with available past measurement and control values, therefore, as  $\mathbf{i}[k-1]$  and  $\mathbf{u}[k-1]$  are the most recent available values, the equation follows from (3.7), (3.8) as

$$\hat{\mathbf{e}}[k] = \hat{\mathbf{e}}[k-1] + \mathbf{u}_R[k-2] - \mathbf{B}^+ \mathbf{i}[k-1] + \mathbf{B}^+ \mathbf{f}(\mathbf{i}[k-2]). \quad (3.9)$$

If the disturbance  $e[k]$  and the current  $i[k]$  are constant, the influence of parameter errors is minimal. Inductance values are only relevant if the current changes significantly. To attenuate this, a lowpass filter is added to the estimated disturbance; if it is slow enough, it is extremely robust and can be seen as quasi-invariant to inductance uncertainties. The inductance is only required if the current is time-varying, for constant current it cancels, therefore the low-pass filter enforces a time-scale decoupling. This will be demonstrated in the experiments later. The disturbance estimator with first-order lowpass-filter reads as

$$\hat{e}[k] = \hat{e}[k-1] + \alpha (\mathbf{u}_R[k-2] - \mathbf{B}^+ \mathbf{i}[k-1] + \mathbf{B}^+ \mathbf{f}(\mathbf{i}[k-2])), \quad (3.10)$$

where  $\alpha = \frac{T_s}{T_s + T_{LP}}$  with  $T_{LP}$  as time-constant of the lowpass filter and  $T_s$  as sampling time. On the other hand, if disturbances such as flux harmonics should be compensated too, the lowpass filter must be very fast, close to the sampling frequency. The assumption that the controller and disturbance estimator do not interact is then not satisfied under uncertain parameters, this is analyzed in the following developments.

### Control input saturation

As mentioned, a careful saturation of the voltage commands must be implemented, otherwise errors appear in the delay compensation.

In this chapter, this task is kept simple to guarantee independent control of the two current components  $i[k]$ . A rectangular saturation is implemented as

$$-U_q \leq u_{sq}[k] \leq U_q, \quad (3.11)$$

$$-U_d \leq u_{sd}[k] \leq U_d. \quad (3.12)$$

More advanced methods are available, for instance, circular limitations  $u_{sd}^2 + u_{sq}^2 \leq U^2$ , but then, however, the current controllers interact [Qua93]. Further improvements are to respect the dynamics and the multivariable structure [OS02].

It is clear that for the delay compensation and disturbance estimation algorithm, the saturated voltage is applied. Furthermore, the estimated disturbance must be respected.

A sum of two commands,  $\mathbf{u}_R[k]$  and  $\hat{e}[k]$ , must be implemented. To assure independence of the disturbance compensation from the control, the method implements saturation in the disturbance estimator first, saturating  $\hat{e}[k]$ . In the second step, according to  $\mathbf{u}[k] = \mathbf{u}_R[k] + \hat{e}[k]$ ,  $\mathbf{u}_R[k]$  is saturated such that the output voltage  $\mathbf{u}[k]$  is within the limits.

## 3.2 Deadbeat control in the flatness-based context

### 3.2.1 Principles of deadbeat control and relation to flatness

Deadbeat control is from a theoretic viewpoint a *linear state feedback control* for discrete-time systems. Unlike continuous-time linear feedback controllers, a discrete time linear controller can be designed for finite settling time [Ack88]. The step response of a deadbeat controlled system is defined as  $n$  delays for a degree  $n$  system (with full relative degree), otherwise said, the control error should vanish after  $n$  steps. For the current control loop, which is a first-order plant, the control error is eliminated after  $n = 1$  step.

#### Deadbeat control as linear feedback controller

The discrete-time linear single-input single output (SISO) system is described by

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k], \quad (3.13)$$

$$y[k] = \mathbf{C}\mathbf{x}[k], \quad (3.14)$$

where  $\mathbf{x}[k] \in \mathbb{R}^n$  is the state at instant  $k$ ,  $u[k] \in \mathbb{R}$  the control input,  $y[k] \in \mathbb{R}$  the output,  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are the system matrix, input vector and output vector, respectively,  $\mathbf{x}[0] \in \mathbb{R}^n$  is an arbitrary initial state and the goal is  $\mathbf{x}[n] = \mathbf{0}$ . This is the typical regulator problem. The linear state controller is

$$u[k] = -\mathbf{K}\mathbf{x}[k], \quad (3.15)$$

where vector  $\mathbf{K}$  is designed such that the closed-loop dynamics

$$\mathbf{x}[k+1] = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}[k] \quad (3.16)$$

lead - after  $n$  steps - to

$$\mathbf{x}[n] = (\mathbf{A} - \mathbf{B}\mathbf{K})^n \mathbf{x}[0] = \mathbf{0} \quad \forall \mathbf{x}[0] \in \mathbb{R}^n. \quad (3.17)$$

For the design, the equation  $(\mathbf{A} - \mathbf{B}\mathbf{K})^n = \mathbf{0}$  must be solved. The simplicity of deadbeat control only becomes clear if the system (3.13) is in controller canonical form, i.e. the system matrix  $\mathbf{A}$  and input vector  $\mathbf{B}$  are in the form

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \vdots \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ b \end{pmatrix}, \quad (3.18)$$

and the output is  $y[k] = x_1[k]$ . Then, the deadbeat controller gains are

$$\mathbf{K} = -[a_0, a_1, \dots, a_{n-1}]/b. \quad (3.19)$$

The controller design then follows straightforward from the model, and no solving of equations is necessary for the design.

This linear state feedback control is a predictive controller. The control law  $\mathbf{u}[k] = -\mathbf{K}\mathbf{x}[k]$  includes the output and a number of predictions, in the case of a system with flat output  $y$ , the state vector is  $\mathbf{x}[k] = (y[k], y[k+1], \dots, y[k+n-1])^T$ , see [SRA04]. Whether the controller is categorized as feedback or predictive controller is purely a matter of implementation, but most engineers nowadays clearly classify it as predictive controller [JP97], as the predicted outputs are used in the control law. Obtaining those predictions is an application-specific problem that is addressed in predictive control implementations.

### Extension to nonzero reference

In the case of nonzero reference, just as for any linear state controller, the design is applied by redefining the state vector as 'error state'  $\mathbf{x}[k] = \boldsymbol{\xi}^*[k] - \boldsymbol{\xi}[k]$ , where  $\boldsymbol{\xi}^*[k]$  is the reference and  $\boldsymbol{\xi}[k]$  the state. The results from the regulator problem which had  $\mathbf{x} = \mathbf{0}$  as goal can be reused for the special control problem  $\boldsymbol{\xi} = \boldsymbol{\xi}^*$ . The reference is assumed to be known  $n - 1$  steps in advance as  $\boldsymbol{\xi}^*[k] = (\xi^*[k], \xi^*[k+1], \dots, \xi^*[k+n-1])$ . Then the system is

$$\boldsymbol{\xi}[k+1] = \mathbf{A}\boldsymbol{\xi}[k] + \mathbf{B}w[k], \quad (3.20)$$

where the input must be rewritten as

$$w[k] = -u[k] - \mathbf{B}^+\mathbf{A}\boldsymbol{\xi}^*[k] + \mathbf{B}^+\boldsymbol{\xi}^*[k+1]. \quad (3.21)$$

Here  $w[k]$  is the physical control input and  $u[k]$  a 'fictitious' input merely used for the control design. The pseudo-inverse  $\mathbf{B}^+$ , following (3.18), is the vector  $[0, 0, \dots, \frac{1}{b}]$ . This way, the design procedure can be repeated for the transformed system to design a deadbeat controller for a nonzero reference, the control law (3.15) can be applied. This is the classical control method, in the following, a flatness-based design is presented.

### Design with feedback linearization

This alternative design method is very close to feedback linearization based *tracking control*, typically applied in continuous-time systems [SL91] but also applicable to

discrete-time systems [RCA01, FM00b]. The underlying deadbeat controller has the control goal  $y[k+n] = y^*[k]$ , whereas a real trajectory tracking controller has the goal  $y[k] = y^*[k]$ , but in discrete time, the design is absolutely equal except that the reference should be predicted for tracking control whereas the reference is directly overtaken for deadbeat control.

Starting from the system description (3.20), the system is linearized, in a sense that a new input is defined such that the system results in a delay chain

$$\xi_1[k+n] = v[k], \quad (3.22)$$

called the 'linearized system'. This 'linearizing input'  $v[k]$  is realized by

$$w[k] = \frac{1}{b}v[k] - \mathbf{B}^+ \mathbf{A}\xi[k]. \quad (3.23)$$

The procedure originates from nonlinear continuous-time control systems and is called feedback linearization (dt. *exakte Zustandslinearisierung*), the new input  $v[k]$  is called Brunovsky - or linearizing - input. It is applied in linear systems as well, here, the term means not only compensating nonlinear terms, but transforming the dynamical system to a chain of delays, therefore to the simplest possible dynamics.

The controller consists of a feedforward part  $v_{ff}[k]$  and an error dynamics feedback part  $v_{fb}[k]$ , given by

$$v_{ff}[k] = \xi^*[k+n], \quad (3.24)$$

$$v_{fb}[k] = -\mathbf{M}(\xi^*[k] - \xi[k]). \quad (3.25)$$

The feedback gain vector  $\mathbf{M}$  results from a direct pole placement. The feedback part is only active if the initial state does not correspond to the reference, i.e. if there is an initial control error. The resulting controller is therefore  $v[k] = v_{ff}[k] + v_{fb}[k]$  and the physical implementation reads as

$$w[k] = \frac{1}{b}\xi^*[k+n] - \frac{1}{b}\mathbf{M}(\xi^*[k] - \xi[k]) - \mathbf{B}^+ \mathbf{A}\xi[k]. \quad (3.26)$$

### Equivalence of the classical and the feedback-linearization-based design

For the special case that the desired error dynamics are minimum-step (deadbeat),  $\mathbf{M}$  is given by  $\mathbf{M} = b\mathbf{B}^{-1}\mathbf{A} - b\mathbf{K}$ , the resulting controller (3.26) is absolutely equal to the deadbeat controller for an arbitrary reference (3.21). More specifically, the result is  $\mathbf{M} = \mathbf{0}$ . This shows that deadbeat control design and feedback-linearization based tracking control design are strongly related. While deadbeat control is rather a feedback gain tuning method, whereas tracking control is rather



a way to account time-varying references, the equations and controllers gains are equivalent.

The mentioned difference in the control goal is based on the practical fact that trajectory tracking control, the reference is predicted to obtain the same output as reference, whereas in deadbeat control, it is directly overtaken, leading to a fixed delay of  $n$  steps between the reference trajectory and the output.

Parametric robustness is a well known problem of feedback linearization. Consequently, the vast literature does not coincide with an appropriate number of industrial applications. A new concept for a more robust design that stems from flatness-based control is feedforward linearization (dt. *exakte Steuerungslinearisierung*) [HD03]. Flatness is somehow related to feedback linearization, but the linearization (transformation from  $u[k]$  to  $v[k]$ ) can be performed in an open-loop control manner, thus, as feedforward control. The advantages of the method have already been shown in continuous-time formulation on power electronics systems [Gen08] and drive system motion control [HD08].

In this chapter the concept is used in discrete time to improve deadbeat control. For application to current control, a first-order plant, the deadbeat condition in the design is that the control error is zero in the next time step, leading to the design rule  $\mathbf{i}[k+1] = \mathbf{i}^*[k]$ . Deadbeat current control is classified as a one-step predictive controller [CKK<sup>+</sup>08], the discrete-time model is used explicitly to calculate the voltage command, so the voltage command is based on a model-based prediction rather than on a compensation based on previous control errors.

### 3.2.2 Conventional deadbeat current controller design

The classical design follows the basic description of the previous section, and includes delay compensation and disturbance estimator. The model of the current dynamics, with compensated delay and disturbance, is given as

$$\mathbf{i}[k+1] = \mathbf{f}(\mathbf{i}[k|k-1]) + \mathbf{B}\mathbf{u}_R[k]. \quad (3.27)$$

The deadbeat condition describes the desired closed-loop performance

$$\mathbf{i}[k+1] = \mathbf{i}^*[k]. \quad (3.28)$$

Defining a new control input

$$\mathbf{v}[k] = \mathbf{f}(\mathbf{i}[k|k-1]) + \mathbf{B}\mathbf{u}_R[k], \quad (3.29)$$

which is the 'linearizing input', the deadbeat control law is

$$\mathbf{v}[k] = \mathbf{i}^*[k]. \quad (3.30)$$

Rearranging these equations with the model, the physical implementation of the deadbeat control law is

$$\mathbf{u}_R[k] = \mathbf{B}^+ \mathbf{i}^*[k] - \mathbf{B}^+ \mathbf{f}(\mathbf{i}[k|k-1]). \quad (3.31)$$

### Characteristics of conventional deadbeat control

The control law compensates the dynamics of the system, with  $\mathbf{v}[k]$ , the system dynamics are made equal to a simple delay

$$\mathbf{i}[k+1] = \mathbf{f}(\mathbf{i}[k|k-1]) + \mathbf{B}\mathbf{u}_R[k] = \mathbf{v}[k]. \quad (3.32)$$

The arising dynamics  $\mathbf{i}[k] = z^{-1}\mathbf{v}[k]$  are called the linearized dynamics, in a sense that the output is a chain of  $n$  delays of the linearizing input  $\mathbf{v}[k]$ . The compensation of the system dynamics is done using measurements and the model, illustrated in Fig. 3.2. This technique is known as feedback linearization. The new input  $\mathbf{v}[k]$  is the Brunovsky input. The parametric sensitivity of this feedback compensation is well-known and the concept has difficulties in industrial applications [SL91].

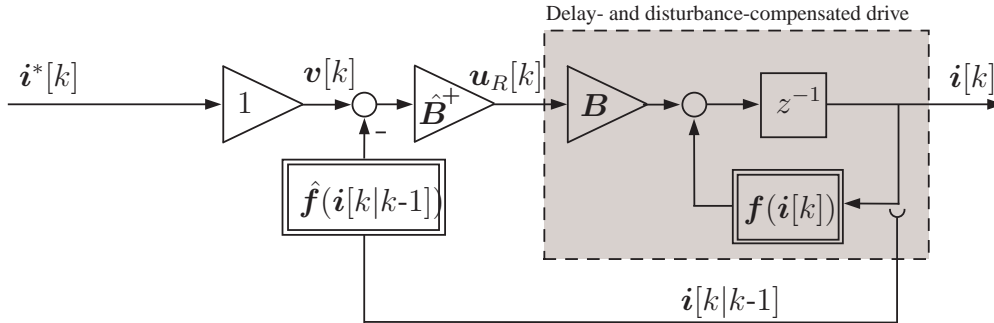


Figure 3.2: Structural diagram of standard deadbeat with feedback linearization.

The notions of feedback linearization and Brunovsky input are generally applied in continuous control. The notion can, however, be directly extended to the discrete-time case [SRA04].

### 3.2.3 Deadbeat design using feedforward linearization

The new design follows the basic description of the previous section, thus includes delay compensation and disturbance estimator, and only has a small difference in

the design. It will be shown that, however, this small difference has an important impact on behavior and performance. The model of the current dynamics is given as

$$\mathbf{i}[k+1] = \mathbf{f}(\mathbf{i}[k|k-1]) + \mathbf{B}\mathbf{u}_R[k]. \quad (3.33)$$

The deadbeat condition describes the desired closed-loop performance

$$\mathbf{i}[k+1] = \mathbf{i}^*[k]. \quad (3.34)$$

Defining a new control input

$$\mathbf{v}[k] = \mathbf{f}(\mathbf{i}^*[k-1]) + \mathbf{B}\mathbf{u}_R[k], \quad (3.35)$$

which only differs from the previous design by using  $\mathbf{i}^*[k-1]$  instead  $\mathbf{i}[k|k-1]$  to implement the 'linearizing input'. Note that both values  $\mathbf{i}^*[k-1]$  and  $\mathbf{i}[k]$  (respectively  $\mathbf{i}[k|k-1]$ ) are equal in the ideal case. The deadbeat control law is

$$\mathbf{v}[k] = \mathbf{i}^*[k]. \quad (3.36)$$

These equations are rearranged with the model, and the physical implementation of the feedforward-linearization based deadbeat control law is

$$\mathbf{u}_R[k] = \mathbf{B}^+\mathbf{i}^*[k] - \mathbf{B}^+\mathbf{f}(\mathbf{i}^*[k-1]). \quad (3.37)$$

A reference governor must be implemented. In the case that the reference  $\mathbf{i}^*[k]$  can not be imposed because of voltage saturation in  $\mathbf{u}_R[k]$ , the situation that the feedforward controller generates one single infeasible voltage peak which is simply cut off by the saturation must be prevented. If it is detected that  $\mathbf{i}[k+1]$  can not (based on the model prediction, independently from any uncertainty) reach  $\mathbf{i}^*[k]$ , the reference  $\mathbf{i}^*[k]$  is reduced accordingly. This way, the feedforward voltage command generated from a large reference step is distributed over multiple sampling intervals. This algorithm is very simple to implement and not further discussed.

### Characteristics of deadbeat design using feedforward linearization

Again, the control law compensates the dynamics of the system, with the definition of the new input  $\mathbf{v}[k]$ , the system is transformed into a simple delay

$$\mathbf{i}[k+1] = \mathbf{f}(\mathbf{i}^*[k-1]) + \mathbf{B}\mathbf{u}_R[k] = \mathbf{v}[k]. \quad (3.38)$$

The compensation of the system dynamics respectively the linearization is done using the model, but is based on references instead of measurements. The new input

$v[k]$  is again a Brunovsky input. The method is denoted as feedforward linearization, and was introduced in the context of flatness-based control as a substitute for feedback linearization to reduce parametric sensitivity [HD03]. The robustness improvement was demonstrated on continuous-time tracking controllers as well as on flatness-based predictive controllers [HD08]. As illustrated in Fig. 3.3, the resulting controllers are feedforward controllers, feedback is not taken into account at all.

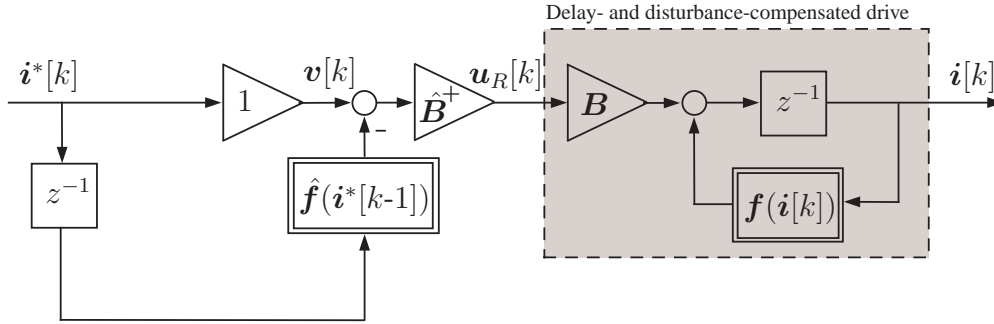


Figure 3.3: Structural diagram of deadbeat using feedforward linearization.

However, one issue is obvious, as there is no feedback, the feedforward controller may not satisfy the control objective. An additional controller is required. Two choices are possible, the disturbance estimator typical for predictive controllers, or as proposed in the literature, the well-established PI controller which is integrated in a two-degree-of-freedom control structure [HD03]. The difference between the delayed reference and the measured output is controlled by the additional feedback mechanism. The resulting control structure is still a predictive controller, even though the predictive part is implemented as feedforward [HD08].

It will be shown later that while the robustness problem is solved, quite some performance is lost. To help out this issue, an intermediate solution of feedback and feedforward linearization is proposed in the following.

### 3.2.4 Deadbeat design using mixed feedback and feedforward linearization

The proposed design again includes the delay compensation and the disturbance estimator. The model of the current dynamics is given as

$$\mathbf{i}[k+1] = \mathbf{f}(\mathbf{i}[k|k-1]) + \mathbf{B}\mathbf{u}_R[k]. \quad (3.39)$$

Defining an initial condition for the current

$$\mathbf{i}_{FB}[k] = q \mathbf{i}[k|k-1] + (1-q) \mathbf{i}^*[k-1], \quad (3.40)$$

where  $0 \leq q \leq 1$ , the value  $\mathbf{i}_{FB}[k]$  is a weighted sum of the measurement and the reference. The weight is  $q$ , setting  $q = 1$  equals the original deadbeat controller, and  $q = 0$  equals the deadbeat controller based on feedforward linearization.

The deadbeat condition describes the desired closed-loop performance

$$\mathbf{i}[k+1] = \mathbf{i}^*[k]. \quad (3.41)$$

Defining a new control input

$$\mathbf{v}[k] = \mathbf{f}(\mathbf{i}_{FB}[k-1]) + \mathbf{B}\mathbf{u}_R[k], \quad (3.42)$$

the deadbeat control law is

$$\mathbf{v}[k] = \mathbf{i}^*[k], \quad (3.43)$$

Rearranging these equations, the physical implementation of the deadbeat control law is

$$\mathbf{u}_R[k] = \mathbf{B}^+ \mathbf{i}^*[k] - \mathbf{B}^+ \mathbf{f}(q\mathbf{i}[k|k-1] + (1-q)\mathbf{i}^*[k-1]). \quad (3.44)$$

Same as for the feedforward linearization, a reference governor must be implemented, to account for the voltage saturation in the feedforward path by adjusting  $\mathbf{i}^*[k]$ .

### Characteristics of deadbeat design using mixed feedback and feedforward linearization

The mix aims at obtaining the robustness advantage of feedforward linearization, but maintaining feedback control. As the feedback of the deadbeat controller is still partially active, the task of the disturbance estimator is simplified as it is not the only feedback mechanism, as shown in Fig. 3.4.

For an open-loop stable system, the feedforward control approach is more robust, the feedback terms are simply less affected by uncertainties. The basic idea is that in an open-loop stable system, the destabilizing mechanism is erroneous feedback.

One question is how to tune the 'mix' parameter  $q$ . It is the only tuning parameter of this design, apart from the time constant of the disturbance estimator  $T_{LP}$ , all other parameters are physical parameters of the drive.

The robustness improvement of this design could, to some extent, be compared to [BLNH05], where the feedback gain of the deadbeat controller has been reduced by replacing the deadbeat condition (3.41) by the condition that the control error is halved in the next sampling step. However, in this design, the tracking performance remains untouched, the deadbeat condition is still active.

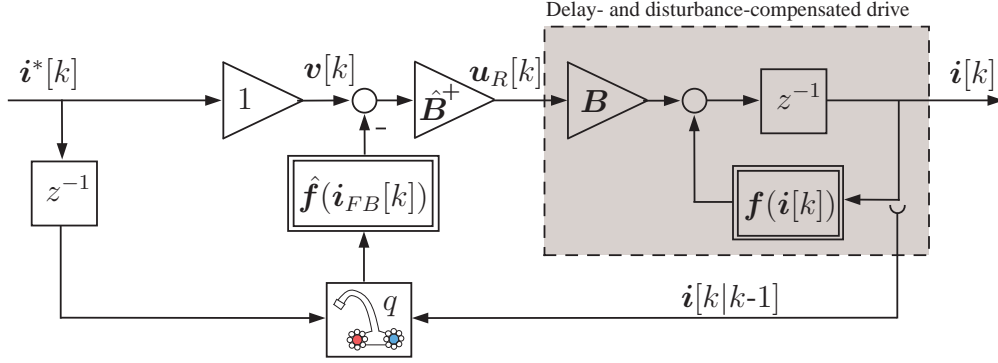


Figure 3.4: Structural diagram of deadbeat using mixed feedforward and feedback linearization.

### 3.3 Analytical robustness considerations

In this section, the robustness regarding disturbances and parameter uncertainties is analyzed. As simplification, a linear single-input single-output (SISO) analysis is performed for a single RL-load. For this simplified analysis, the results already describe the advantages and issues of the three proposed methods. Of course, in the practical application to a drive, there is an interaction of the orthogonal components (cross-coupling), and nonlinearities appear for instance in the induction motor. Here, in a simple way, the results are presented and interpreted which are also found later in the experiments.

The model is given as

$$i[k+1] = (1-a)i[k] + bu[k] - be[k], \quad (3.45)$$

where the parameters are

$$a = T_s \frac{R}{L}, \quad b = T_s \frac{1}{L}, \quad (3.46)$$

and  $e[k]$  is a possibly time-varying disturbance signal. For a perfectly sinusoidal stator winding, the disturbance is constant, but harmonic effects, transients on the orthogonal axis, etc., generate a time-varying disturbance  $e[k]$ .

This section derives transfer functions that account for constant parametric offsets. The detailed calculations are found in appendix D. Their evaluation and interpretation will be given in the next chapter.

### 3.3.1 Transfer function of deadbeat controller

First, the transfer function of a deadbeat controller with delay compensation but without disturbance estimator is calculated. This neglect is valid when the output is lowpass filtered with a high  $T_{LP}$ . The simplicity of the sole deadbeat controller enables more insight. Parametric offsets in the controller and the plant are regarded, the estimated parameters are denoted as  $\hat{a}$  and  $\hat{b}$ , respectively.

The resulting closed-loop dynamics in the discrete-time frequency domain is calculated in appendix D.1 and the transfer function appears as

$$G_{db}(z) = \frac{I(z)}{I^*(z)}. \quad (3.47)$$

Under correct parameters and with  $q = 1$ , the transfer function is  $G_{db}(z) = z^{-1}$ , the deadbeat condition  $i[k+1] = i^*[k]$  is therefore satisfied. If the parameters do not match, there are additional dynamics, as it is a second-order plant consisting of delay plus current dynamics. Two poles and one transfer zero determine the dynamics.

### 3.3.2 Transfer function of deadbeat controller including a disturbance estimator

Now the complete transfer function including the disturbance estimator is calculated. It is necessary to analyze the behavior under a fast disturbance estimator (i.e. small  $T_{LP}$ ) to see the interaction under parameter faults. The closed-loop system has one reference input  $i^*[k]$ , one disturbance input  $e[k]$  and one output  $i[k]$ .

Two transfer functions of the closed-loop system appear, the first one is the response from the reference to the output and the second one is from the disturbance to the output,

$$I(z) = G_{i^*i}I^*(z) + G_{ei}E(z). \quad (3.48)$$

The calculations are again shown in appendix D.2.

### 3.3.3 Interaction of disturbance estimator and deadbeat controller: Low pass filter

Under ideal conditions  $\hat{a} = a$ ,  $\hat{b} = b$  and if  $e[k]$  is constant, the deadbeat controller and the disturbance estimator are decoupled. Under uncertainties, however, decoupling is affected, especially in transient operation. The lowpass filter for  $\hat{e}$  attenuates

the effect and considerably increases parametric robustness of the disturbance estimator. This is known as time-scale decoupling.

The closed-loop transfer function of the disturbance response  $G_{ei}(z)$  shows that the poles under ideal conditions are real for values  $T_{LP} \geq 3T_s$ , as calculated in appendix D.3. If the lowpass has a smaller time constant than this, the poles become complex and the step response is subject to oscillations, furthermore, the magnitude of the poles approaches the stability limit and parametric sensitivity becomes higher. Therefore, a senseful condition to the lowpass filter is

$$T_{LP} \geq 3T_s, \quad (3.49)$$

respectively  $\alpha \leq \frac{1}{4}$ . In standard deadbeat control, if the system is subject to strong uncertainties, the filter must be set even slower, however, at the cost of performance.

In the literature, it is claimed the sole purpose is to attenuate parametric uncertainty influence [KY01]. This time-scale decoupling makes sense as the inductance sensitivity only appears in current transients. A comparison is difficult as the analyzed systems do not have a computational delay, they are not standard DSP platforms. If the two delays are present in the system, however, it becomes clear that the filter is required for stability and for an oscillation-free step response even in the case of perfect parameter knowledge.

### 3.3.4 Steady-state accuracy

Steady-state accuracy of the deadbeat controllers with delay compensation and disturbance estimator is analyzed. In steady-state, the reference, disturbance and the output are constant. The steady-state offset is calculated in appendix D.4 and found to be

$$\frac{i^*[k]}{i[k]} = \frac{\hat{a} + q}{\hat{a} + q - \hat{a}q} = \frac{1}{1 - \frac{\hat{a}q}{\hat{a} + q}} \approx 1 + \frac{\hat{a}q}{\hat{a} + q}. \quad (3.50)$$

This shows that the errors arising from uncertainties in all parameters are mostly compensated, to one part as some parameters do not affect steady-state values, to the other part as the disturbance estimator compensates these errors. Noting that  $\hat{a} \ll 1$ , only a small prediction error remains. For fast sampling systems with large open-loop time constants, the steady-state offset is negligible.

Interestingly, in the feedforward linearization approach  $q = 0$ , there is no more steady-state offset, so  $q < 1$  is, even though a small, improvement.



## 3.4 Conclusions

This chapter has exposed the state-of-the-art and the detailed design of deadbeat control, a very performant control method. However, it suffers from some limitations. Problems in the design are delays, disturbances, and high sensitivity to modeling errors and noise. Therefore, in the literature, the controller was extended with a model-based delay compensation technique and a disturbance estimator to avoid current ripples and to improve steady-state accuracy, respectively. However, the sensitivity to model parameters, especially with respect to the inductance, has only been addressed by tuning the deadbeat controller slower.

It is shown that the deadbeat current controller is a flatness-based predictive controller, as presented in [FM00a], but designed in the discrete-time domain. The sensitivity of the deadbeat controller is caused by the use of feedback linearization. In flatness, an open-loop substitute for feedback linearization has been proposed to solve the robustness problem, feedforward linearization [HD08]. On the other hand, this method as open-loop controller comes with some performance loss.

Three variants are presented, the conventional deadbeat design, then, a deadbeat design based on feedforward linearization, and as a tradeoff of the respective issues and advantages, a mix between both methods.

The analytical fundamentals have been derived to study the robustness, respectively the influence of parameter errors and disturbances. As first results, steady-state accuracy and tuning limits for the disturbance estimator have been found out. The analysis of these three methods is completed in the following chapter by providing experimental results.



## CHAPTER 4

---

### **Robust flatness-based predictive control: Experimental evaluation of deadbeat control**

---

This chapter completes the developments of the previous chapter by an experimental evaluation to obtain a clear result on robustness and performance. The three presented designs of conventional deadbeat control, deadbeat control with feedforward linearization, and deadbeat control using mixed feedback and feedforward linearization, are studied.

In the first part, successively, each design is analyzed in terms of tracking performance under both correct parameters and uncertainties, an analytic robustness consideration is discussed with pole maps, and the disturbance rejection capability is tested. The calculations and experiments are based on the parameters of a PMSM, described in appendix B.1.

In the second part, the deadbeat controller is evaluated on three different AC drives, the PMSM, the SynRM and the IM. While the previous experiments limit to a step response analysis, here, a more serious application scenario is analyzed. Different speeds and full load are studied. Some problems appear, namely magnetic saturation effects, general parameter deflection effects, control input saturation and flux harmonics.

## 4.1 Comparative evaluation of the proposed controllers

### 4.1.1 Evaluation of the conventional deadbeat controller

#### Resistance uncertainty

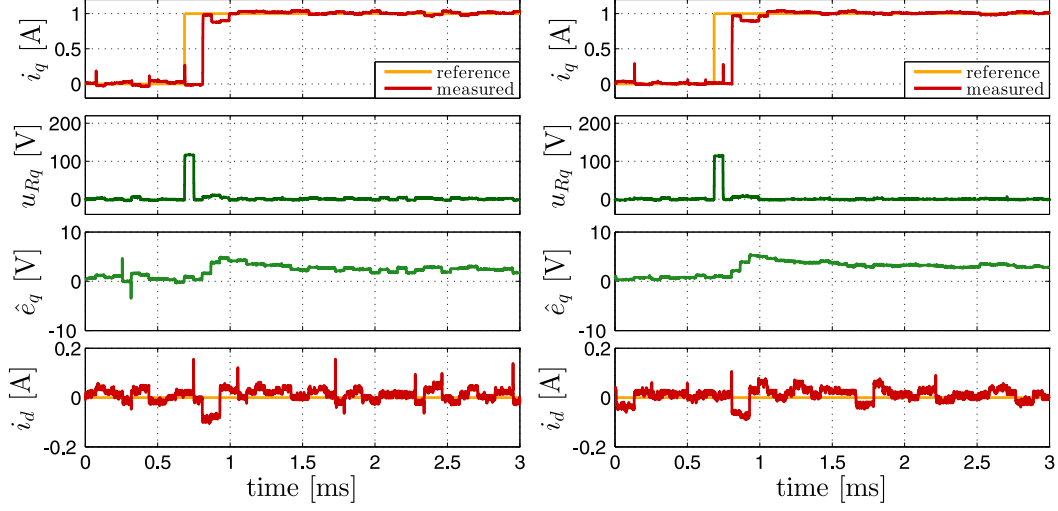
It is known from the literature that the influence of resistance uncertainties is neglectable [YL02]. Two arguments support this: The first one is that the open-loop time constant of a drive is much higher than the closed-loop one, in this case approximately two times the sampling interval. The influence on dynamical behavior is therefore marginal. From the transfer functions of section 3.3 it can be seen that the resistance influence applies to  $\hat{a}$  and is marginal as  $\hat{a} \ll 1$ . The second argument is that the resulting steady-state error is almost completely compensated by the disturbance estimator.

Experimental results are shown in Fig. 4.1. Subfigure (a) is the response with nominal parameters, the response is almost perfect and the disturbance estimator only has a small reaction. In (b) the resistance is assumed one tenth of the nominal value. The steady-state value of  $u_R$  is slightly too small and consequently the current offset is compensated by a higher disturbance estimator output  $\hat{e}$ . In (c) the resistance is assumed 10 times higher than the nominal value. The deadbeat controller calculates a too high voltage  $u_R$  for the steady-state, the resulting offset is compensated by a lower disturbance estimator output  $\hat{e}$ .

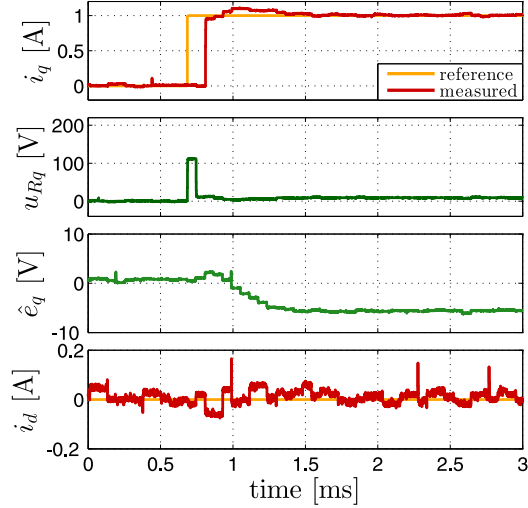
This shows that the errors arising from resistance uncertainty in both the disturbance estimator and in the predictive controller compensate.

#### Inductance uncertainty

Contrary to resistance, the influence of inductance uncertainty is very important and deserves more discussion. Uncertainties can arise from identification errors or from unmodeled magnetic saturation effects, amongst others. They result in two effects. The first one is the influence on the magnetic flux and the induced voltage, it appears on the right hand side of the system equation, subsequently, it is compensated by the disturbance estimator. This effect is only indirectly related to the dynamics and therefore rather decreases accuracy than stability. The second effect is the influence on the dynamic gain of the system, in the model the current derivatives are multiplied by the inductance, this can lead to either a slowdown, badly damped oscillations, or instability.



(a) Correct resistance  $\hat{R} = R$  and  $T_{LP} = 3T_s$ . (b) Underestimated  $\hat{R} = 0.1R$  and  $T_{LP} = 3T_s$ .



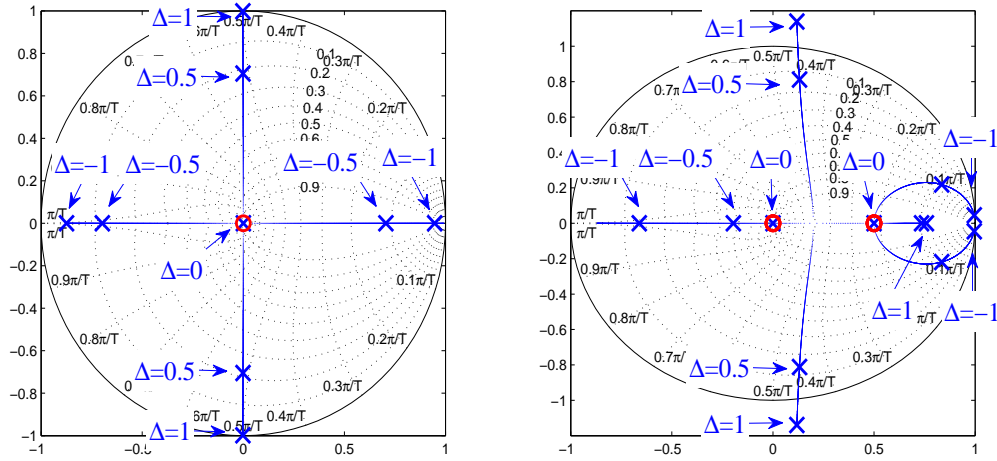
(c) Overestimated  $\hat{R} = 10R$  and  $T_{LP} = 3T_s$ .

Figure 4.1: Experimental results of conventional deadbeat control under resistance uncertainty.

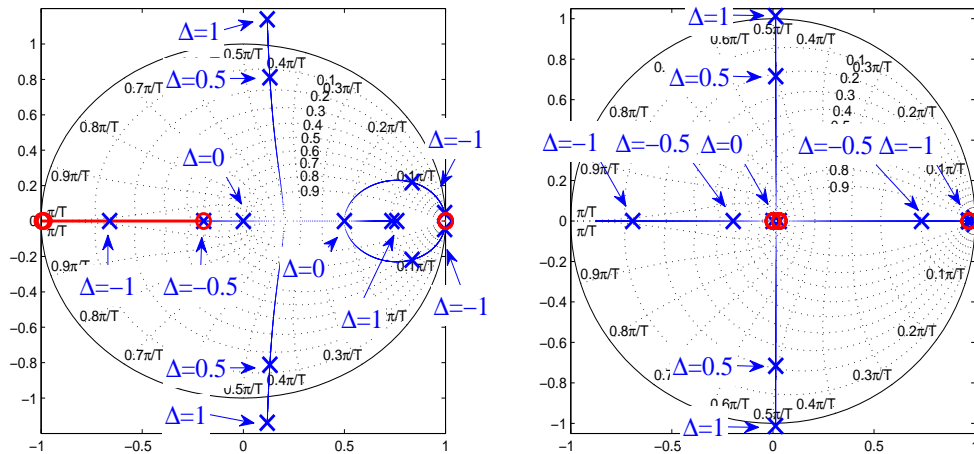
To analyze the effect on the gain, a constant inductance error  $\Delta$  with

$$\Delta = \frac{\hat{L} - L}{L} \quad (4.1)$$

is assumed to simplify the analysis, however, the results are also valid for a setpoint-



(a) Reference response without disturbance estimator  $G_{db}(z)$ . (b) Reference response with fast disturbance estimator  $G_{i^*i}(z)$ .



(c) Disturbance response with fast disturbance estimator  $G_{ei}(z)$ . (d) Reference response with slow disturbance estimator  $G_{i^*i}(z)$ .

Figure 4.2: Discrete frequency domain polemaps of conventional deadbeat control under inductance uncertainty.

dependent inductance error  $\Delta L(i)$  as it will be shown later. Using the transfer functions presented in section 3.3, the effect of inductance variations can be studied.

In Fig. 4.2 the  $z$  space pole maps are plotted. Subfigure (a) is the deadbeat con-

troller without disturbance estimator  $G_{db}(z)$ . For correct inductance, the transfer function is one zero and two poles at the origin. The two poles are the reference delay and the closed-loop deadbeat response, the zero arises from the delay compensation (Smith predictor). For overestimated inductances  $\Delta > 0$ , a pair of imaginary poles appears while the zero remains in the origin. The delay compensation does not operate correctly and the deadbeat controller generates too high control actions, resulting in overshoots and oscillations. The stability limit is reached once  $\Delta = 1$  or  $\hat{L} = 2L$ . For underestimated inductances, a pair of real poles appears, resulting in a slowdown of the reference response. The stability limit is reached once  $\Delta = -1$  or  $\hat{L} = 0$ , which does never appear in practice.

Subfigure (b) is the deadbeat controller with a fast disturbance estimator  $T_{LP} = 3T_s$ ,  $G_{i^*i}(z)$ . There is one additional pole and zero, as the order of the closed-loop dynamics is increased by the disturbance estimator. Apart from that, the characteristics are similar to (a), the sensitivity is slightly increased as for overestimated inductances the stability limit is now  $\hat{L} = 1.7L$ , and for underestimated inductances, there are some additional slow oscillations.

Subfigure (c) is the disturbance response  $G_{ei}(z)$ . It looks very similar to the reference response, except for the zeros. This also means that the sensitivity is the same, logically, if either controller or estimator destabilize, the complete system is unstable. Subfigure (d) is the deadbeat controller with a slow disturbance estimator  $T_{LP} = 2$  ms. If the pole-zero cancelation at  $z = 1$  is ignored, it is equal to the deadbeat controller without disturbance estimator: The disturbance estimator is fully decoupled from the reference response if it is sufficiently slowed down.

To clarify these results, experimental results are shown in Fig. 4.3. Subfigure (a) is the response with nominal inductance. The reference is reached after two sampling steps, the delay of the control system. Subfigure (b) shows the stability limit at  $\hat{L} = 1.7L$  with a fast disturbance estimator. The overestimated inductance results in a too high control voltage. Two sampling steps after the reference step, feedback is available and the controller counteracts the overshoot, however, because of the model error, with a too strong action. This faulty behavior repeats and the parameter leads to an undamped oscillation. Subfigure (c) shows the response at  $\hat{L} = 0.5L$ , the control voltage is too low, after two sampling steps the controller repeats the action, resulting in a series of peaks on  $u_q$ . The slow overshoot is caused by the disturbance estimator, the error is interpreted as disturbance, such an overshoot does not appear for a slow estimator. Subfigure (d) finally shows the response at an overestimated inductance  $\hat{L} = 1.7L$  with a slow disturbance estimator  $T_{LP} = 2$  ms. As the stability limit is at  $\Delta \approx 1$ , a badly damped oscillation appears. These experimental results confirm the correctness and explain the characteristics of the transfer function and of the polemaps.

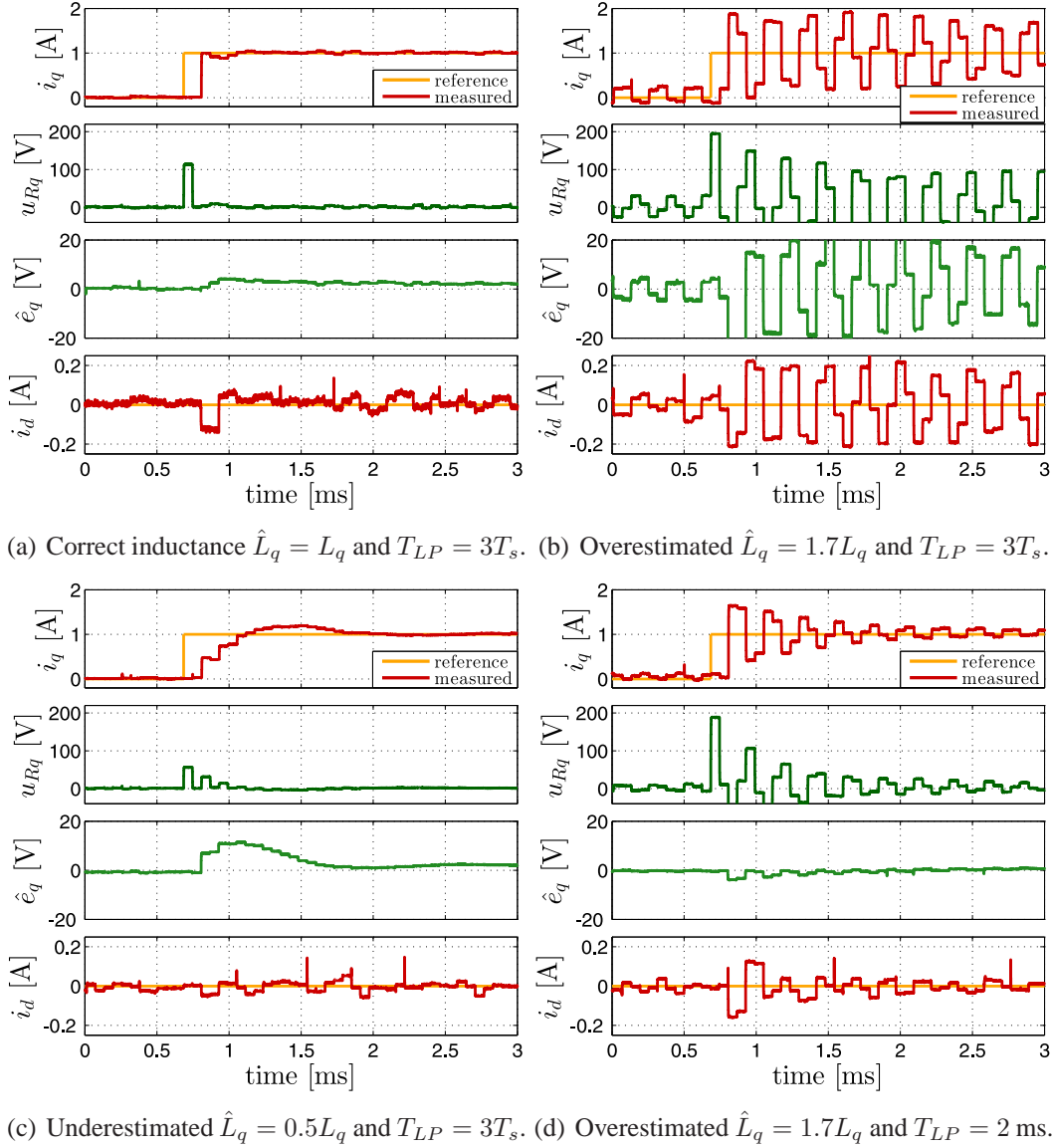


Figure 4.3: Experimental results of conventional deadbeat control under inductance uncertainty.

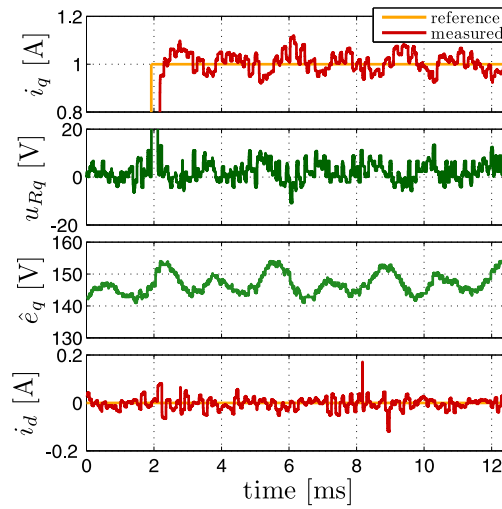
### High speed effects

Operation in high speed leads to two disturbing effects. The first is that the cross-coupling between the two current components is increased. Cross-coupling is modeled for deadbeat design, however, under uncertainties, the decoupling is not pre-



cise. This effect can be seen as fast time-varying disturbance on the respective axis.

The second effect in high speed is that flux harmonics appear. These are caused by the non perfectly sinusoidal distribution of the stator windings, the effect is position dependent with six periods (and integer multiples) per electrical rotation. At high speed, if the controller cannot compensate these harmonic effects, the harmonics will be seen in the currents causing a torque ripple.



(a) Correct  $\hat{L}$  and  $\hat{R}$  and  $T_{LP} = 3T_s$ .

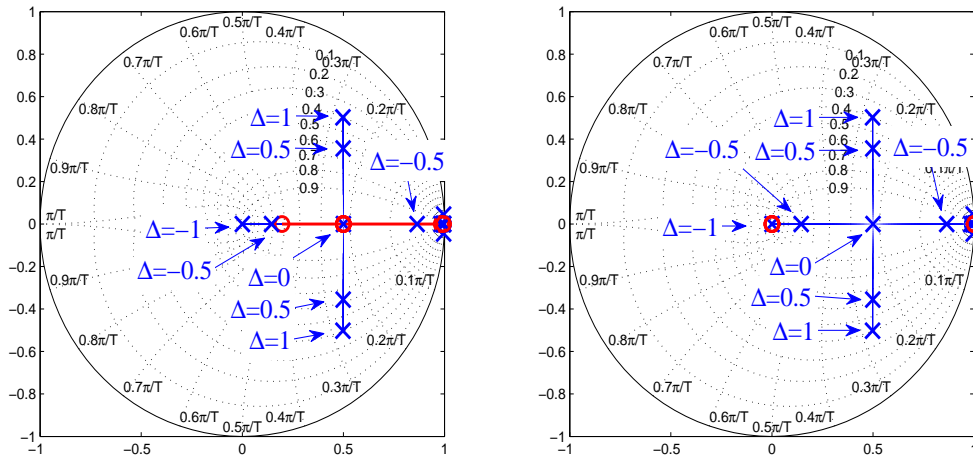
Figure 4.4: Experimental results of conventional deadbeat control in high speed (2000 rpm).

In Fig. 4.4 the performance is evaluated experimentally at 2000 rpm for the conventional deadbeat design. Current tracking is very good and in the estimated disturbance, a frequency component of 600 Hz is seen, which is the compensated flux harmonics as the fundamental excitation frequency at 2000 rpm is 100 Hz. During the quadrature current transient, the direct current is well maintained at 0 A. It is obvious that compensating harmonics is a matter of the disturbance estimator, the deadbeat controller is a linear feedback controller without integral action, it cannot fully reject a disturbance. Still, the deadbeat controller shows some reaction. Compared to the PI controller of Fig. 2.8, the disturbance estimator is a remarkable improvement if  $T_{LP}$  is small.

## 4.1.2 Evaluation of the deadbeat controller using feedforward linearization

### Inductance uncertainty

As mentioned, when using feedforward linearization, the deadbeat controller turns into a feedforward controller. Interestingly, parameter faults do not impact stability. The only feedback mechanism that counteracts uncertainty effects by feedback is then the disturbance estimator. The disturbance estimator is assumed to be sufficient in order to compensate for the lack of feedback.



(a) Reference response with fast disturbance estimator  $G_{i^*i}(z)$ . (b) Disturbance response with fast disturbance estimator  $G_{ei}(z)$ .

Figure 4.5: Discrete frequency domain polemaps of deadbeat control using feedforward linearization under inductance uncertainty.

In Fig. 4.5 the  $z$  space pole maps are plotted. The analysis of the deadbeat controller without disturbance estimator is skipped, as the poles of the transfer function  $G_{db}(z)$  are unaffected at  $q = 0$ . The analysis with a slow estimator is also skipped as performance is too low. Subfigure (a) shows the controller with a fast disturbance estimator  $G_{i^*i}(z)$ , and (b) is the corresponding disturbance response  $G_{ei}(z)$ . The disturbance response is slowed down, even though the lowpass filter is untouched. However, it is also seen that both the controller and the estimator remain stable even for strong inductance errors. The stability limit is reached at  $\hat{L} = 4L$ , even though  $T_{LP}$  is kept extremely low. Hence, not only the deadbeat controller becomes more robust, but also the disturbance estimator. Feedforward linearization

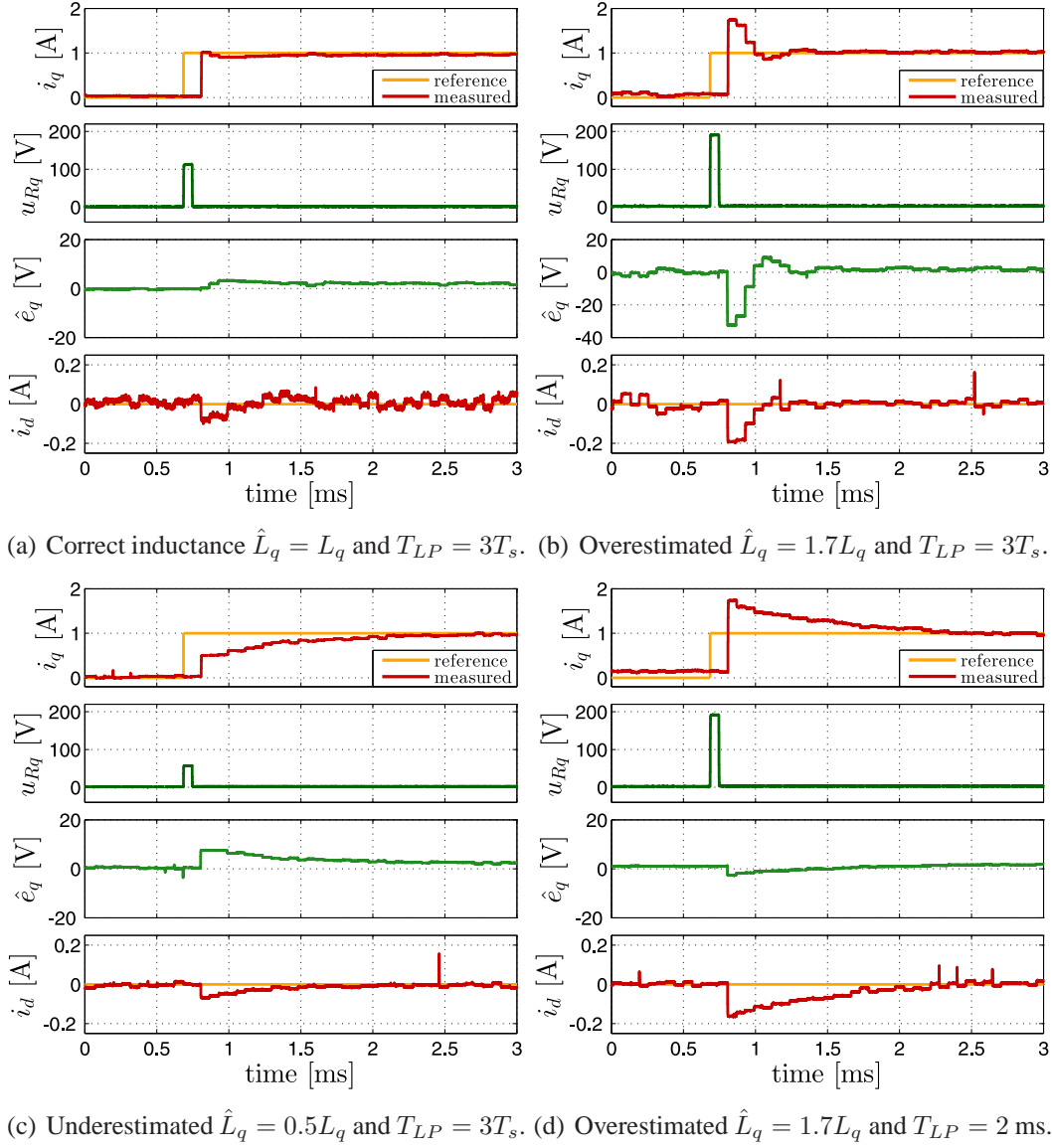


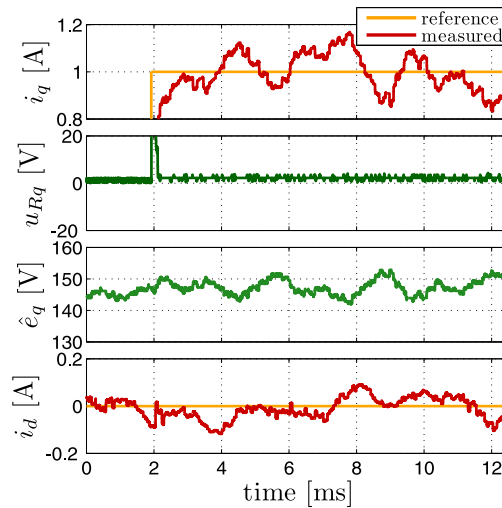
Figure 4.6: Experimental results of deadbeat control using feedforward linearization under inductance uncertainty.

decouples deadbeat control and disturbance estimation / compensation. Even higher stability limits are obtained by increasing  $T_{LP}$ .

In Fig. 4.6 the sensitivity is analyzed experimentally. Fig. 4.6 (a) shows the response with correct inductance, there is not much difference to the conventional deadbeat controller. Subfigure (b) shows the response for an overestimated induc-

tance  $\hat{L} = 1.7L$ , there is an overshoot which is corrected by the disturbance estimator. Two sampling steps after the reference step feedback is available and the disturbance estimator commands a negative voltage, the three negative steps on  $\hat{e}$  compensate the excessive initial voltage on  $u_{Rq}$ . In contrast to the conventional controller, there is no oscillation, so this is a remarkably good response regarding that the overshoot of +70% cannot be reduced due to the system delay of two steps. In (c), for an underestimated inductance  $\hat{L} = 0.5L$ , the disturbance estimator also compensates the too small control action, this time there is no overshoot and the response is too slow. Subfigure (d) shows the same scenario as (b) but with a slow disturbance estimator  $T_{LP} = 2$  ms, obviously the response is slowed down too much.

### High speed effects



(a) Correct  $\hat{L}$  and  $\hat{R}$  and  $T_{LP} = 3T_s$ .

Figure 4.7: Experimental results of deadbeat control using feedforward linearization in high speed (2000 rpm).

Fig. 4.7 shows the behavior of the deadbeat controller with feedforward linearization at 2000 rpm. While the disturbance estimator compensates the flux harmonics less well, i.e. with a smaller voltage than in Fig. 4.4, also, current tracking becomes very poor. The disturbance estimator, as only feedback mechanism, cannot compensate the cross coupling and the disturbance simultaneously. These considerable current excursions only appear in high speed operation. The lack of feedback

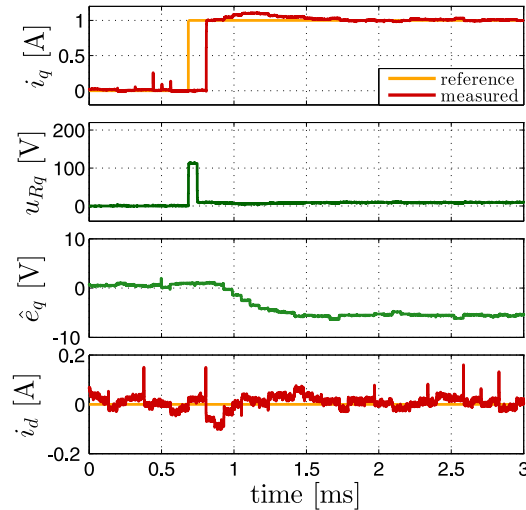
in this deadbeat controller reduces performance too much, this experimental test disqualifies the approach of pure feedforward linearization.

To conclude the experiments, pure feedforward linearization comes with considerable robustness improvements. Uncertainties only provoke a steady-state error, but no oscillations. This steady-state offset must be compensated by a feedback mechanism, namely the disturbance estimator. However, in high speed, where cross-coupling effects and time-varying disturbances are present, the performance loss is apparently too strong.

### 4.1.3 Evaluation of the deadbeat controller using mixed feedback and feedforward linearization

As feedback linearization is performant but sensitive, and feedforward linearization is robust but less performant under disturbances, a mix between both methods has been proposed. The new tuning parameter which describes the tradeoff between both methods is set to  $q = 0.5$ .

#### Resistance uncertainty



(a) Overestimated  $\hat{R} = 10R$  and  $T_{LP} = 3T_s$ .

Figure 4.8: Experimental results of deadbeat control using mixed feedback and feedforward linearization under resistance uncertainty.

Experimental results of a resistance error test are shown in Fig. 4.8. The response

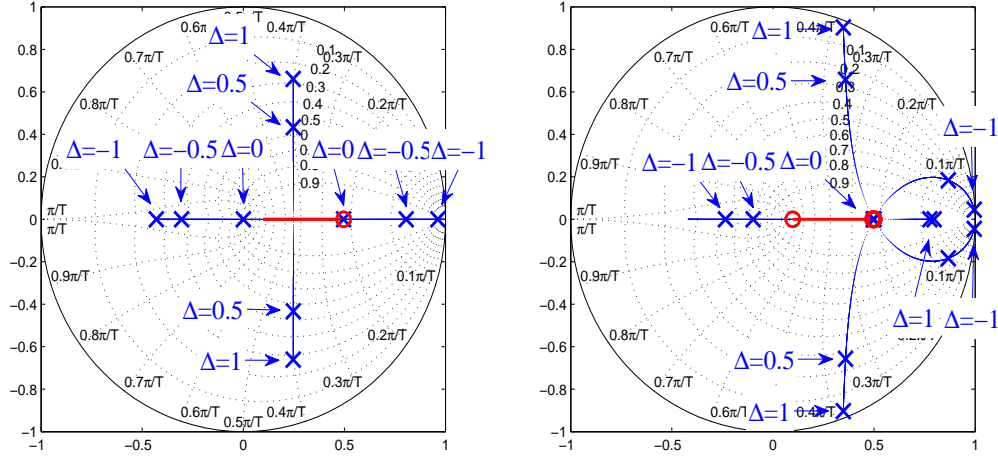
is identical than with conventional deadbeat in Fig. 4.1 (c), the resistance error is compensated by the disturbance estimator. The test is just for the sake of completeness, to show that the resistance is no problem for the proposed method just like as for the conventional method.

### Inductance uncertainty

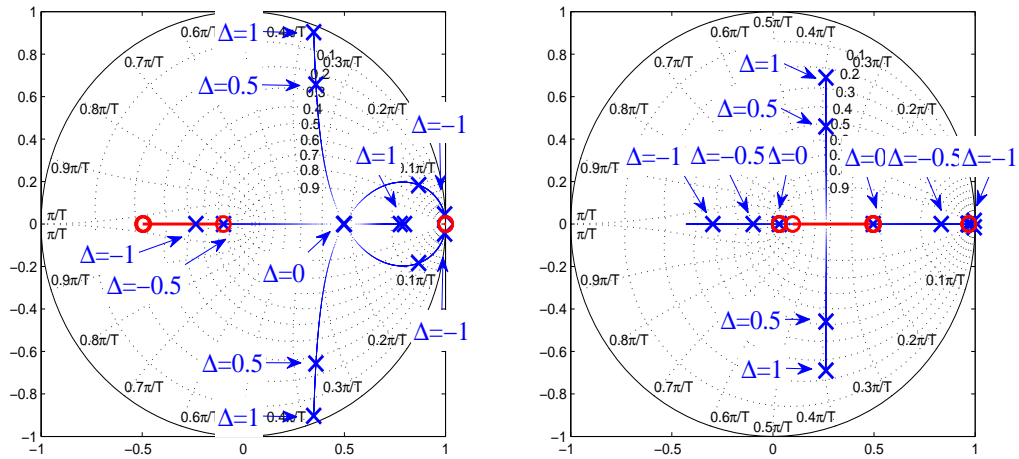
To analyze the influence of an inductance error, in Fig. 4.10, the  $z$  space pole maps are plotted. Subfigure (a) is again the deadbeat controller without disturbance estimator  $G_{db}(z)$ . Compared to the conventional deadbeat controller, for overestimated inductances, damping is improved absolutely by about 0.3 for the same uncertainties. From the transfer function follows that the stability boundary is now  $\hat{L} = 3L$  instead of  $\hat{L} = 2L$ , in fact the stability limit depends on  $q$  and, when neglecting the estimator, it is at  $\Delta = \frac{1}{q}$  for inductance uncertainties. Subfigure (b) represents the results of the controller with a fast disturbance estimator  $G_{i^*i}(z)$ . Here, damping is also higher for the same error, compared to the conventional deadbeat controller, and the disturbance estimator generates some additional slow dynamics for underestimated inductances. Subfigure (c) shows the disturbance response with a fast lowpass filter  $G_{ei}(z)$ . Disturbances are compensated quickly and robustness is same as in (b). Subfigure (d) then shows the controller with a very slow disturbance estimator, it is almost identical to (a) except for an additional pole and zero around  $z = 1$ .

In Fig. 4.10 the sensitivity is shown by experiments. Subfigure (a) shows the response with correct inductance, there is not much difference to the conventional deadbeat controller. Subfigure (b) shows the response for an overestimated inductance  $\hat{L} = 1.7L$ , the current overshoot remains for two sampling steps, but is then controlled to its reference value by combined action of the deadbeat controller and the disturbance estimator. In contrast to the conventional controller, there is nearly no oscillation. In (c) the response for an underestimated inductance  $\hat{L} = 0.5L$  is shown, the initial voltage peak is too small and both the disturbance estimator and the deadbeat controller compensate the remaining control error. The slow overshoot is caused by the disturbance estimator, as it was shown in the transfer function analysis. Subfigure (d) shows the response with a slow disturbance estimator. The system response is excellent, as robustness of the disturbance estimator is increased for higher  $T_{LP}$ , and the overshoot error almost completely vanishes after two sampling steps.

This shows that the gain errors arising from inductance uncertainty still result in stability problems, but the problem is visibly attenuated with the help of feedforward linearization. By choosing an intermediate between feedback and feedforward



(a) Reference response without disturbance estimator  $G_{db}(z)$ . (b) Reference response with fast disturbance estimator  $G_{i^*i}(z)$ .



(c) Disturbance response with fast disturbance estimator  $G_{ei}(z)$ . (d) Reference response with slow disturbance estimator  $G_{i^*i}(z)$ .

Figure 4.9: Discrete frequency domain polemaps of deadbeat control using mixed feedback and feedforward linearization under inductance uncertainty.

linearization with  $0 < q < 1$ , the robustness advantages of feedforward linearization are obtained to some extent without the performance disadvantage that the feedback control task is left fully to the disturbance estimator.

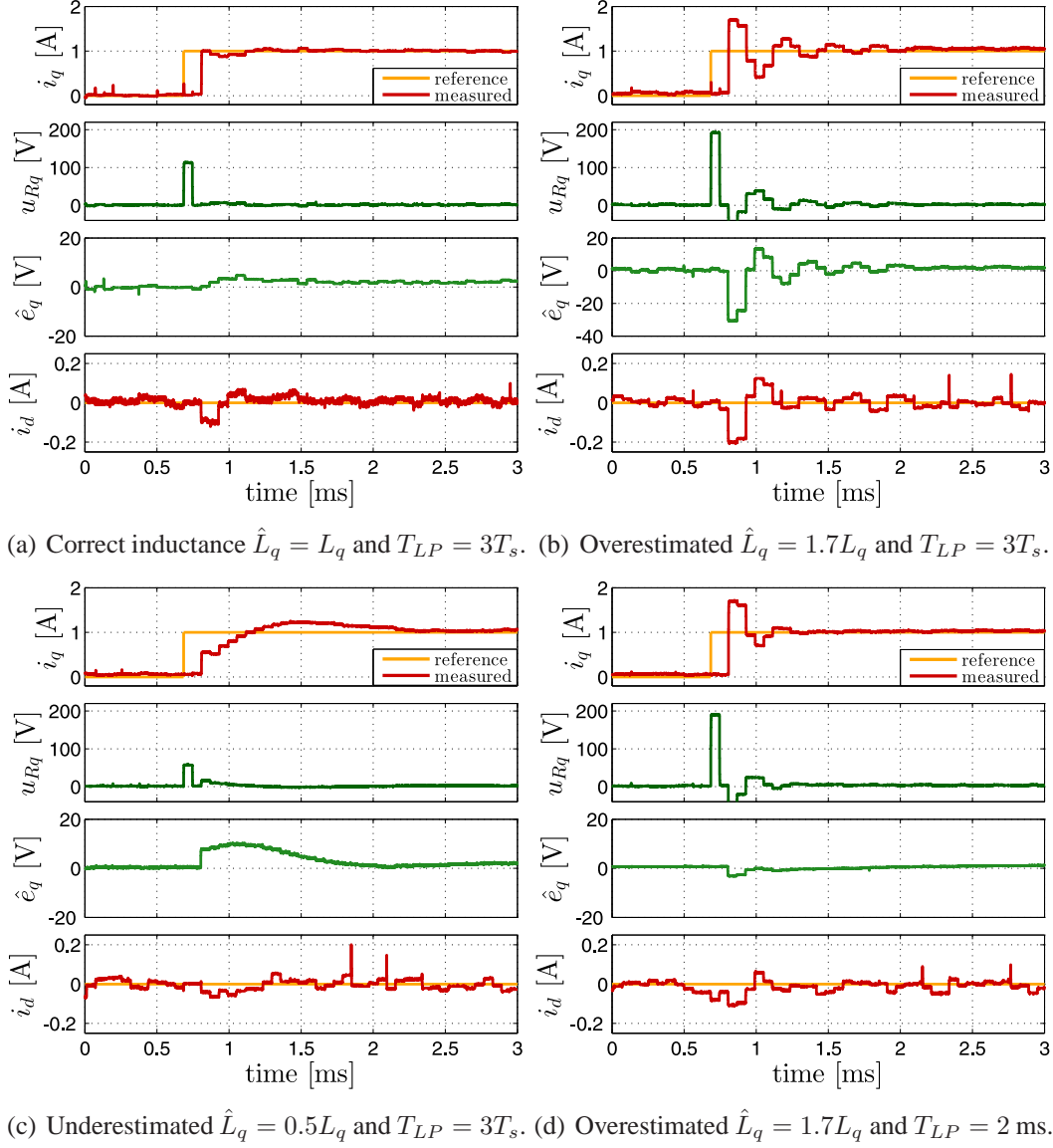


Figure 4.10: Experimental results of deadbeat control using mixed feedback and feedforward linearization under inductance uncertainty.

Following the theoretical results, it can be shown that the stability limit becomes  $\Delta = \frac{1}{q}$  without disturbance estimator, therefore, for smaller  $q$ , robustness increases. On the other hand, performance decreases with  $q$ , as more work is left to the disturbance estimator. The new parameter  $q$  describes a compromise between robustness and control performance. However, for  $q \approx 0.5$ , all characteristics such as tracking



performance, disturbance rejection and robustness are all simultaneously at a very good level.

### High speed effects

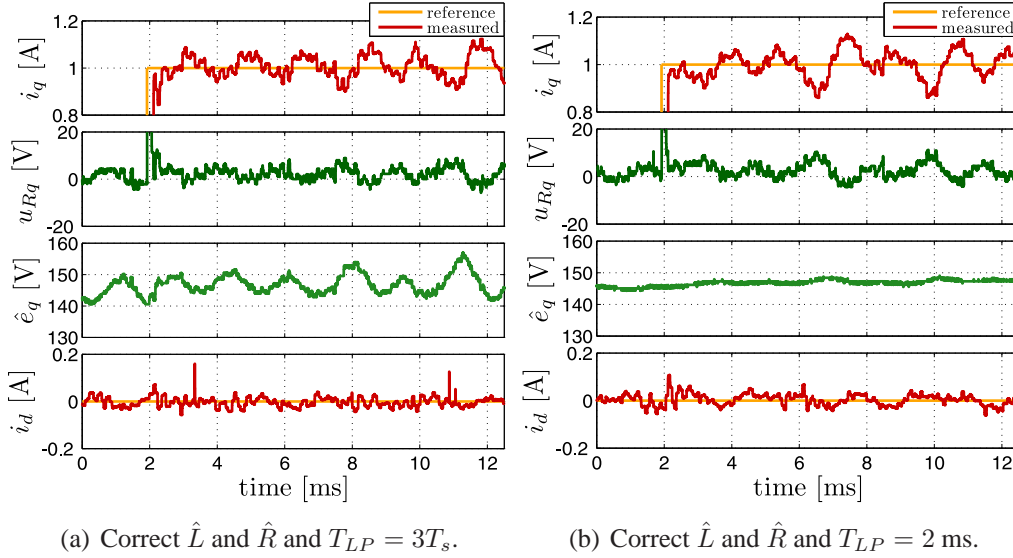


Figure 4.11: Experimental results of deadbeat control using mixed feedback and feedforward linearization in high speed (2000 rpm).

Fig. 4.11 shows the behavior of the deadbeat controller with mixed feedforward and feedback linearization at 2000 rpm. In (a), where a fast disturbance estimator is applied, the harmonics are compensated almost as well as with the conventional deadbeat controller in Fig. 4.4, at the same time, current tracking is very good. The difference is very small, so that can be assumed that the performance loss of the mixed method is marginal compared to the conventional deadbeat controller.

Subfigure (b) shows the same controller with a slow disturbance estimator  $T_{LP} = 2$  ms. Obviously, the bandwidth of the estimator is not sufficient to compensate the flux harmonics, resulting in a current ripple. This proves that harmonics are compensated by the disturbance estimator to the major part, not by the deadbeat controller as sometimes assumed in the literature [YL02].

These experiments demonstrate that the proposed method of mixed feedforward and feedback linearization shows good tracking performance and disturbance rejection capability.

#### 4.1.4 Conclusions on the comparative evaluation

The major sources of uncertainty, consisting of resistance and inductance uncertainty, as well as the influence of flux harmonics, have been investigated.

The resistance uncertainty, as known, is not a problem for any of the controllers.

The inductance uncertainty, on the other hand, turns out as a key stability issue. The conventional deadbeat controller destabilizes at  $\hat{L} = 1.7L$ . The controller design based on feedforward linearization could solve this problem, even with the fast disturbance estimator, the limit is shifted to  $\hat{L} = 4L$ , and by tuning the estimator slower with  $T_{LP}$ , it can be made even more robust. The mixed approach is a tradeoff and the limit is shifted to  $\hat{L} = (1 + \frac{1}{q})L$ , here with  $q = 0.5$ , it is at  $\hat{L} = 3L$ . Therefore, the inductance uncertainty problem can now be handled.

The analytical study of the previous chapter leads to a couple of pole maps that clarify and support these results for the inductance uncertainty. Even though this study was performed on a simplified model, the obtained information describes the advantages and problems of the controllers sufficiently well. The general validity of these comparative results is therefore given.

The disturbance rejection capability is important to obtain good results in high speed operation. Here, flux harmonics appear, and as the analyzed servo drive has a small inductance, these cause a considerable current ripple. To compensate this fast disturbance, the estimator must be sufficiently fast, the proposed setting was  $T_{LP} = 3T_s$ . The conventional deadbeat controller can handle this uncertainty well and only leaves a ripple of about 50 mA (RMS) magnitude. The mixed approach comes with a similar performance. The design based on feedforward linearization, however, leaves a ripple of more than 100 mA (RMS), one part is caused by the flux harmonics and the other part by the cross-coupling, where also an oscillating interaction between the current components  $i_{sd}$  and  $i_{sq}$  appears.

So it can be concluded that the conventional deadbeat controller is highly performant but too sensitive. On the other hand feedforward linearization is highly robust but suffers from reduced performance, especially regarding disturbance rejection. The approach of mixed feedforward and feedback linearization is a compromise between both methods and turns out to be both performant and robust. It has an excellent tracking and disturbance rejection capability while it can handle parametric uncertainty.

Another important aspect, not yet discussed, is measurement noise. Because deadbeat is a very aggressive scheme and tries to immediately compensate output offsets, noise may render the system unstable. In Fig. 4.12, the standard deadbeat controller in (a) is compared to the mixed feedforward-feedback linearization approach in (b). The current measurement signal lines were unshielded and placed

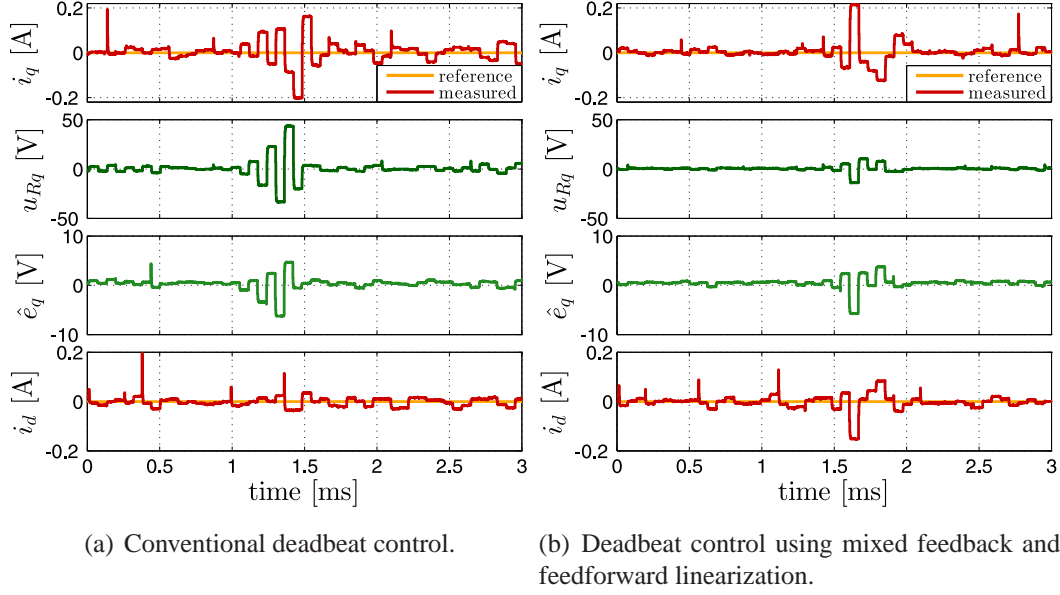


Figure 4.12: Experimental results under measurement noise.

nearby a device that generates an identical electromagnetic ping regularly. As result, the current measurement signal is subjected to noise. While the deadbeat controller reacts with about 40% of available voltage to this ping, the mixed approach only reacts with about 15% and does not influence the real current signal that much. This is quite logical as the noise is amplified less in the feedback loop ( $q < 1$ ) of the proposed control system.

The analyzed error sources are the major problems in a modern drive system. Apart from them, some other problems may appear.

The scheme is unable to compensate sensor errors, for instance current sensor offsets or absolute and differential position sensor errors. The outputs are enforced to the measured outputs.

Voltage modulation also deserves some discussion. Using space vector modulation techniques, a linear space vector can be modulated until 0.866 of the maximum voltage, which is  $\frac{2}{3}$  of the DC-link voltage. This means any voltage vector  $\mathbf{u}_{dq}$  with magnitude lower than  $0.866\frac{2}{3}U_{DC}$  and arbitrary angle can be generated. Beyond this magnitude, the inverter operates in the overmodulation regime resulting in harmonics. Because of the physical limitation of the voltage, such harmonics cannot be compensated. Furthermore, if the switching frequency is slow compared to the fundamental excitation frequency, harmonic effects with the switching frequency appear.

Voltage variations on the DC link have been disregarded during this work. In typical servo inverters, the DC link voltage is measured and the saturation limit and the switching times can be adjusted. However, this error source seems to be of secondary relevance.

Furthermore, the switching dead times of the inverter are assumed to be compensated, as this technique is established in most industrial servo drives nowadays. Without dead time compensation, a 'zero-current' oscillation problem appears, as reported in the literature [KMY90].

### 4.1.5 Structural difference between deadbeat and PI control

Formally, a deadbeat controller might be compared to a PI controller, as it consists of a linear component, the actual deadbeat control law, for instance (3.15), and an integrating component, the disturbance estimator. However, there are some key differences.

The performance and robustness of the PI controller with symmetrical optimum criterion has been shown in section 2.7. The stability limit was at  $\hat{L} = 2.7L$ . The PI controller also had a large ripple cause by the flux harmonics, about 140 mA (RMS). While the reference tracking was fine in the 2DoF extension with a feedforward controller, robustness and disturbance rejection are not sufficient. The PI controller has been outperformed by all three discussed deadbeat controller designs. This can be traced back to some systematic limitations of feedback compensators.

A classical PI controller has one single input, the control error  $i^* - i$ . The controller commands are generated from the error signal, however, this controller does not make a difference between a reference change or a disturbance, therefore it does not know about the cause of a control error. The PI controller reacts with two compensation mechanisms, the P- and the I-part. Many tuning rules exist, for instance the magnitude optimum criterion, which has good reference tracking performance as it has a quite high proportional gain, but a less good disturbance rejection. The implemented symmetric optimum criterion, on the other hand, has good disturbance rejection as the integrator gain is higher, but less good reference tracking. While it must be noticed that an experimentally tuned PI controller, that accounts saturation by anti-windup techniques, may have better performance than these algebraic tuning rules, generally, a PI regulator has to trade off reference tracking versus disturbance rejection. This gives rise to a fundamental limitation.

Two-degree-of-freedom extensions exist which aim at improving this trade off, by splitting the controller in two different systems, either a feedforward and a feedback path, or two cascaded systems. However, the tuning effort becomes complex and the decoupling is somehow limited.

Contrary to this, a deadbeat controller has a more general controller structure with two different inputs, the reference  $i^*$  and the measurement feedback  $i$ . The deadbeat controller is responsible for tracking performance and is designed ignoring any disturbance. The tracking performance is the best possible, only limited by the voltage saturation. For disturbance rejection, a disturbance estimator is added to the system. It is decoupled explicitly from the deadbeat controller and therefore has, in the case of good parameter knowledge, a completely independent action. This additional feedback mechanism with integral action is also close to the best possible physical dynamics – only limited by the delays in the feedback loop. The controller can make a difference between a reference change and a disturbance, and has independent and decoupled mechanisms for each purpose that operate each at the physical system limit.

Therefore the deadbeat controller can break the fundamental limitation of classical linear controllers. This fact has been stated for many model-based and predictive control designs [PLR05].

## 4.2 Implementation and performance evaluation

In this section, the proposed deadbeat controller is evaluated on three types of AC drives, namely a permanent magnet synchronous machine (PMSM) servo drive, a synchronous reluctance machine (SynRM) and an induction machine (IM).

The controller is based on the classical deadbeat design, combined with a disturbance estimator [KY01] and delay compensation [MKY03]. The only difference is that feedforward linearization is partially applied to increase robustness to parametric uncertainties, and the lowpass filter for the disturbance estimator is kept extremely fast.

### 4.2.1 Results on a permanent-magnet synchronous machine

The model applied for the deadbeat current controller is based on Euler forward discretization of the continuous model and is

$$\dot{\mathbf{i}}_{s,dq}[k+1] = \mathbf{f}(\dot{\mathbf{i}}_{s,dq}[k]) + B\mathbf{u}_{s,dq}[k], \quad (4.2)$$

where the system function vector is

$$\mathbf{f}(\dot{\mathbf{i}}_{s,dq}[k]) = \begin{pmatrix} (1 - T_s \frac{R_s}{L_d})i_{sd}[k] + T_s n_p \omega_M \frac{L_q}{L_d} i_{sq}[k] \\ (1 - T_s \frac{R_s}{L_q})i_{sq}[k] - T_s n_p \omega_M \frac{L_d}{L_q} i_{sd}[k] \end{pmatrix}, \quad (4.3)$$

and the input vector is

$$B = T_s \begin{pmatrix} \frac{1}{L_d} \\ \frac{1}{L_q} \end{pmatrix}. \quad (4.4)$$

The induced voltage term on the  $q$ -axis ( $e = K n_p \omega_M$ ) is neglected, it is compensated by the disturbance estimator. Apart from that, the major task of the disturbance estimator is to compensate the flux harmonics. As the inductances are very small, the flux harmonics result in a quite considerable current (and torque) ripple at high speed, the compensation of this effect is a significant quality improvement.

For the controller, the nominal model parameters from appendix B.1 are applied. The only two tuning parameters are  $q = 0.5$  and  $T_{LP} = 3T_s = 0.1875$  ms.

The deadbeat controller therefore consists of 50% feedback and 50% feedforward linearization, to reduce the influence of parametric uncertainties and the sensitivity to noise. The closed-loop cut-off frequency of the disturbance estimator is sufficiently high. Therefore, the major part of the flux harmonics (6 times the excitation frequency, or 600 Hz at 2000 rpm) is canceled.

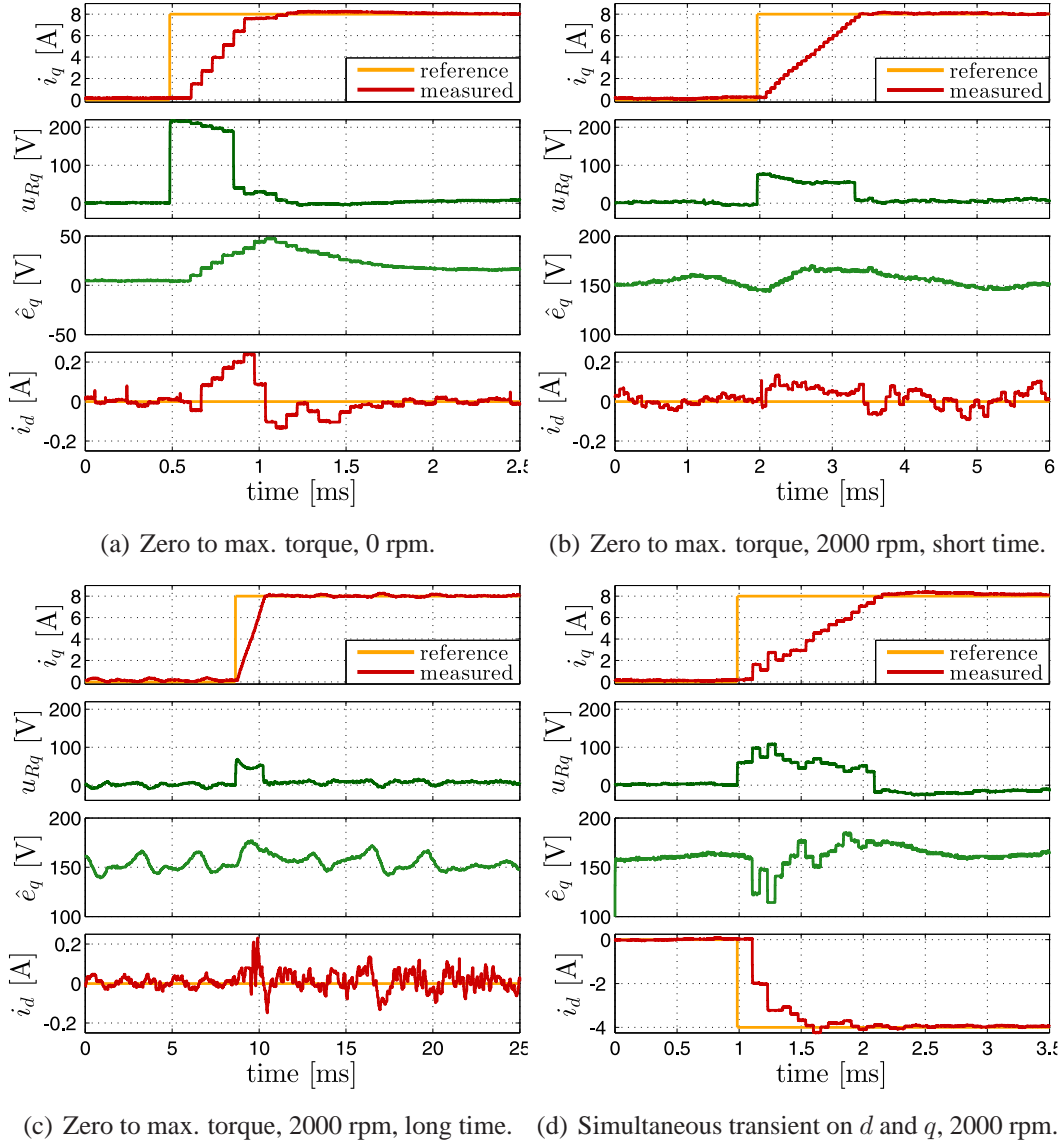


Figure 4.13: Proposed deadbeat control ( $q = 0.5$ ,  $T_{LP} = 3T_s = 0.1875$  ms) on a surface-mounted permanent magnet synchronous machine (PMSM).

Fig. 4.13 shows the experimental results. In (a) a reference step from zero to nominal torque is applied at zero speed. Because of some small parameter uncertainties and as the speed is not perfectly 0, the disturbance estimator is active. The sum of the control voltage  $u_{Rq}$  and the estimated disturbance  $\hat{e}_q$  is at the maximum voltage during the transient. In (b) the same scenario is repeated at 2000 rpm. Be-

cause of the induced voltage (back-EMF), which is about 150 V, the control voltage  $u_{Rq}$  is smaller and the transient is slower. A slow oscillation appears on  $\hat{e}_q$  which is the compensation of the flux harmonics. In (c) the same situation is shown over a longer time, it is seen that the oscillation on  $\hat{e}_q$  is about 600 Hz (superimposed by about 300 Hz), which indicates the compensation of the harmonics, the remaining current ripple is less than about 100 mA or 1.25% of the rated current. Finally, (d) shows a simultaneous transient from zero to nominal torque and from  $i_{sd} = 0$  to  $-4$  A that is performed at 2000 rpm. Because of imperfect parameters and the nonconstant disturbance, performance is deteriorated during the simultaneous transient. This could only be improved by accounting for the multivariable structure of the system, however, as only two current components are available, the information for the (simple) estimator is also limited to two disturbances, but not the interaction. Performance is still convincing.

To conclude, the deadbeat current controller is a very powerful method on a PMSM. Parameter knowledge of this machine type is very reliable, although even small uncertainties and secondary effects have a high influence on the output as the inductances are very small. The fast disturbance estimator proves as powerful tool to compensate such errors.

### 4.2.2 Results on a synchronous reluctance machine

The model applied for the deadbeat current controller is the same as for the PMSM. As the SynRM has no permanent magnet, there is no back-EMF and no electromagnetic torque. The generated torque is purely the so-called reluctance torque, as described in section 2.2.

In Fig. 4.14 the flux curves of the applied motor are shown. The inductances are subject to very strong saturation effects, and additionally, to magnetic cross-coupling effects. Furthermore the SynRM has quite strong flux harmonics, as the stator is originally from a norm induction motor. The harmonics, however, generate only a small current ripple because of the high inductance.

So far, only few works are available on high-performance control of this very special AC drive. Model-based designs which take saturation effects into account by advanced modeling exist and obviously have a much higher performance [KP02, MCP07], such as extremely low overshoot and short settling time. However, the presented work should serve as benchmark for robustness of a deadbeat controller with a linear flux model, for standard control schemes these simple models lead to bad performance and instabilities [KP02].

Identification of these nonlinear flux curves is sometimes performed for sensorless control schemes, however, this procedure is time-consuming. For current con-



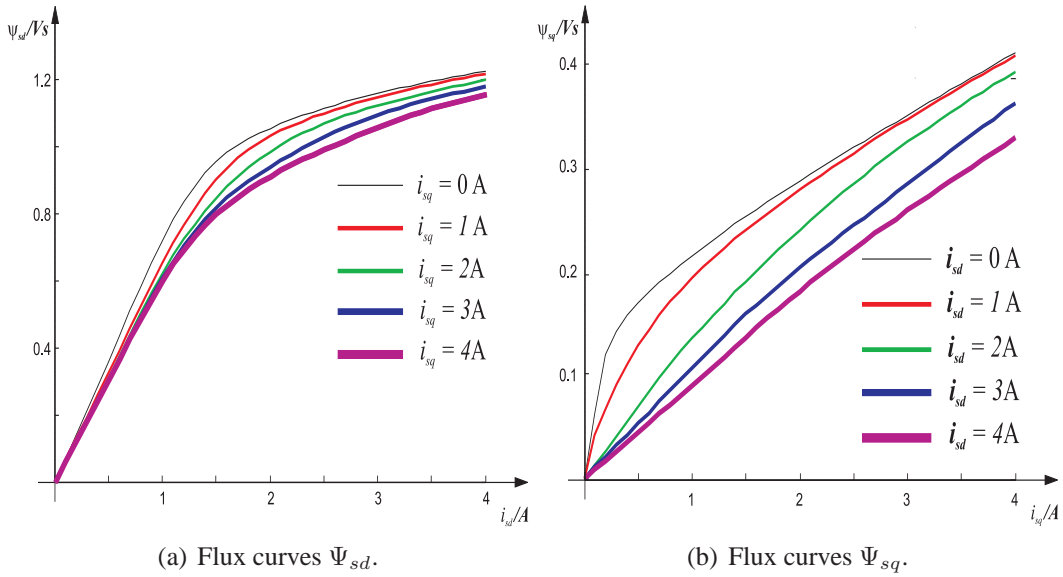


Figure 4.14: Nonlinear flux curves of a synchronous reluctance machine (SynRM). (Source: D. Paulus, *Optimierung eines drehgeberlosen auf Hochfrequenzinjektion basierenden Regelverfahrens*, Diploma thesis, University of Wuppertal, Germany, 2006.)

trol, the identification should be as simple as possible. For the design, the unsaturated inductance is identified, meaning,  $L_q$  is identified at  $i_{sd} = 0$  A and from the step response of  $i_{sq}$  from 0 to 0.1 A. The inductance is then halved to somehow account for the saturation, the values are  $L_d = 350$  mH and  $L_q = 280$  mH, however, depending on the setpoint, this leads to a considerable inductance uncertainty as seen on the flux curves.

In order to still obtain reliable deadbeat control,  $q = 0.25$  is set on the  $i_{sd}$ -controller and  $q = 0.125$  is set on the  $i_{sq}$ -controller. The lowpass filter of the disturbance estimator is slightly enlarged to  $T_{LP} = 6T_s$ . These means will allow a considerable improvement in robustness, with a rather moderate impact on performance. It is not possible to implement such a fast disturbance estimator with standard deadbeat, also it is not possible to run the conventional deadbeat controller, only the introduction of feedforward linearization and the parameter  $q \ll 1$  makes this possible.

Figs. 4.15 show the experimental results. Subfigures (a) and (b) show the small-signal step response from  $i_{sq} = 0$  to 0.5 A in order to show the impact of saturation. The first step is unsaturated at  $i_{sd} = 0$  A, the second step is saturated at  $i_{sd} = 2$  A. The response is very fast and reliable. While (a) is at standstill, (b) is at 1000 rpm, the estimated disturbance is average about zero while the steady-state voltage  $u_{sq}$

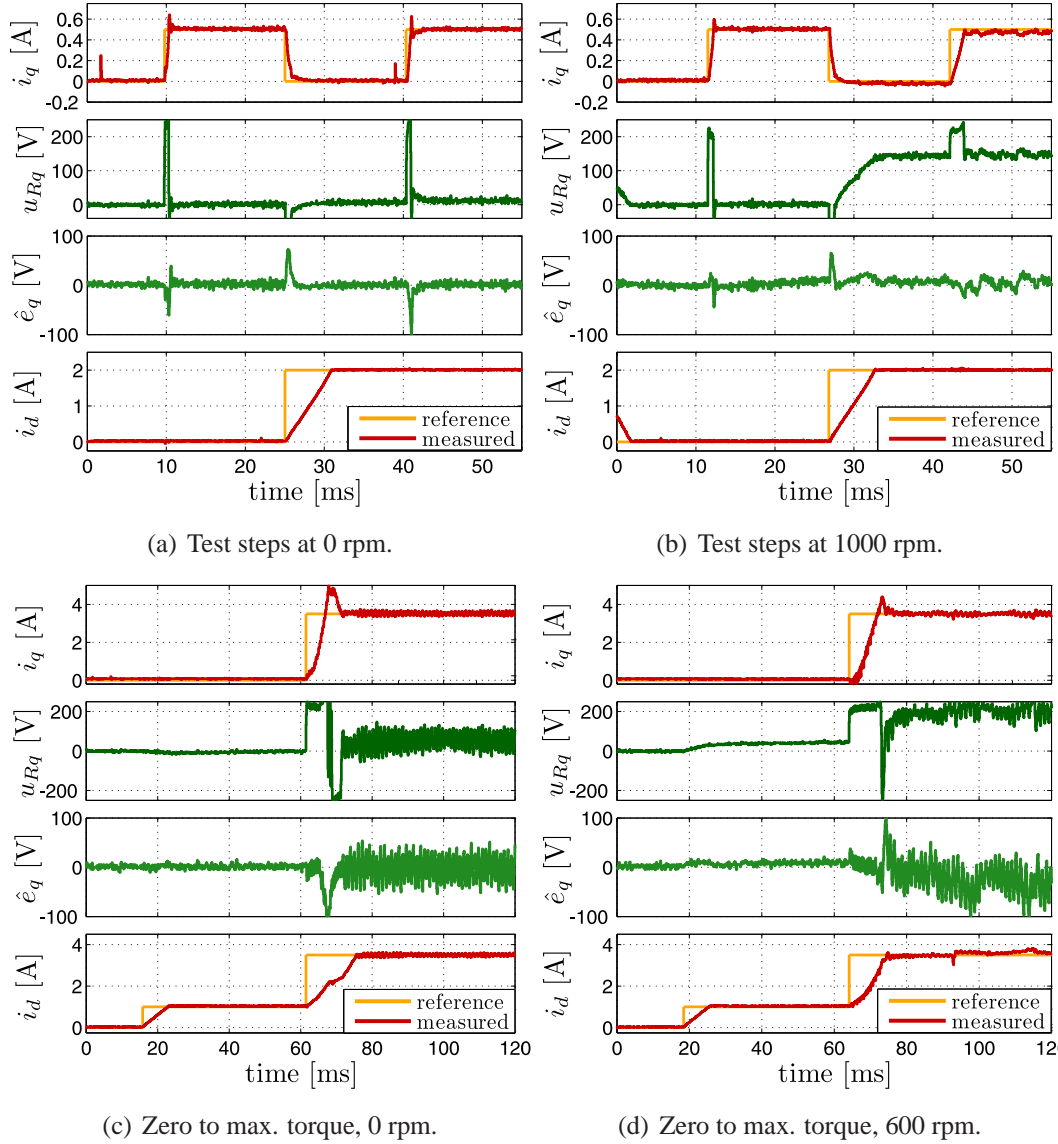


Figure 4.15: Proposed deadbeat control ( $q = 0.25/0.125$ ,  $T_{LP} = 6T_s = 0.375$  ms) on a synchronous reluctance machine (SynRM). Top: test of saturation influence, bottom: from zero to maximum torque.

becomes considerable, this is specific for the SynRM which has high cross-coupling terms because of the high inductance, but no back-EMF.

Subfigures (c) and (d) show the more typical application scenario where the SynRM is operated in maximum-torque-per-ampere (MTPA) mode. First a step

from zero current to  $i_{sd} = 1$  A is performed, which is called premagnetization. Then the torque is increased to its rated value by adjusting an optimal current angle of  $60^\circ$ . Because of the strong saturation at rated current, the command voltage oscillates slightly. Simultaneous change on both axes is not a problem this time, as the inductance values are higher and the transient is slower.

To conclude, the deadbeat current controller has some problems on the SynRM. Saturation effects are extreme on the chosen motor, resulting in a very high inductance uncertainty, and the controller based on a linear flux model suffers of reduced performance. However, the proposed deadbeat controller operates reliable and with nice performance over the complete current range.

### 4.2.3 Results on an induction machine

The deadbeat controller is somewhat more involved for the induction machine, as a rotor flux observer is necessary and a number of nonlinearities appear for field-oriented control.

As rotor flux observer, a full-order observer in stator reference frame is designed. Even though simple integration of the rotor model represents a stable observer [MR96b], a full-order observer is required for precise flux estimation under uncertainties, appearing for instance in field-weakenig operation. The full-order rotor flux observer in the stator reference frame [MR00] is a Luenberger observer and is given by the nonlinear discrete-time equations

$$\begin{aligned}\tilde{\Psi}_{r\alpha}[k] = & (1 - T_s\eta)\tilde{\Psi}_{r\alpha}[k-1] - T_s n_p \omega_M \tilde{\Psi}_{r\beta}[k-1] + T_s \eta L_m i_{s\alpha}[k-1] \\ & + T_s \xi (\tilde{i}_{s\alpha}[k-1] - i_{s\alpha}[k-1]),\end{aligned}\quad (4.5)$$

$$\begin{aligned}\tilde{\Psi}_{r\beta}[k] = & (1 - T_s\eta)\tilde{\Psi}_{r\beta}[k-1] + T_s n_p \omega_M \tilde{\Psi}_{r\alpha}[k-1] + T_s \eta L_m i_{s\beta}[k-1] \\ & + T_s \xi (\tilde{i}_{s\beta}[k-1] - i_{s\beta}[k-1]),\end{aligned}\quad (4.6)$$

$$\begin{aligned}\tilde{i}_{s\alpha}[k] = & (1 - T_s\gamma)\tilde{i}_{s\alpha}[k-1] + T_s \beta \eta \tilde{\Psi}_{r\alpha}[k-1] + T_s \beta n_p \omega_M \tilde{\Psi}_{r\beta}[k-1] \\ & + \frac{1}{\sigma L_s} u_{s\alpha}[k-1] + T_s \xi (\tilde{i}_{s\alpha}[k-1] - i_{s\alpha}[k-1]),\end{aligned}\quad (4.7)$$

$$\begin{aligned}\tilde{i}_{s\beta}[k] = & (1 - T_s\gamma)\tilde{i}_{s\beta}[k-1] + T_s \beta \eta \tilde{\Psi}_{r\beta}[k-1] - T_s \beta n_p \omega_M \tilde{\Psi}_{r\alpha}[k-1] \\ & + \frac{1}{\sigma L_s} u_{s\beta}[k-1] + T_s \xi (\tilde{i}_{s\beta}[k-1] - i_{s\beta}[k-1]).\end{aligned}\quad (4.8)$$

If the observer feedback parameter  $\xi$  is set zero, the eigenvalues of this observer are  $-\eta \pm j n_p \omega_M$ , meaning it converges with the rotor time constant  $\frac{1}{\eta}$ . The integration of the rotor model leads to an open-loop value. For positive values of  $\xi$ , this information is corrected by the stator model (4.7) and (4.8), where the rotor flux is

generating a voltage component useful for feedback. The observer will converge faster while parametric robustness will increase. The chosen feedback parameter value is  $\xi = 4$  for the machine described in appendix B.1. Timing is very important if the observer shall be used for deadbeat control – it is noted that the input signals  $\mathbf{i}_{s,\alpha\beta}[k-1]$  and  $\mathbf{u}_{s,\alpha\beta}[k-1]$  are all available at instant  $k$ . Furthermore, it is seen that, as the controller design is based on Euler forward discretization, the rotor model in the flux observer acts as delay compensation (or prediction) algorithm for the rotor flux signal  $\Psi_{r,\alpha\beta}$ . Therefore, if the feedback parameter  $\xi$  is not too large, no additional delay compensation system is required for deadbeat flux control.

### Current control

The model used for deadbeat current control is based on Euler forward discretization of the continuous model and is

$$\mathbf{i}_{s,dq}[k+1] = \mathbf{f}(\mathbf{i}_{s,dq}[k]) + B\mathbf{u}_{s,dq}[k], \quad (4.9)$$

where the nonlinear system function vector is

$$\mathbf{f}(\mathbf{i}_{s,dq}) = \begin{pmatrix} (1 - T_s\gamma)i_{sd} + T_s n_p \omega_M i_{sq} + T_s \eta L_m \frac{i_{sq}^2}{\Psi_{rd}} \\ (1 - T_s\gamma)i_{sq} - T_s n_p \omega_M i_{sd} - T_s \eta L_m \frac{i_{sd} i_{sq}}{\Psi_{rd}} \end{pmatrix}, \quad (4.10)$$

and the input vector is

$$B = T_s \begin{pmatrix} \frac{1}{\sigma L_s} \\ \frac{1}{\sigma L_s} \end{pmatrix}. \quad (4.11)$$

All terms independent of the current  $\mathbf{i}_{s,dq}$ , meaning the induced voltage and the flux coupling, are classified as disturbance and ignored in the model. They will be compensated by the disturbance estimator. The terms described by the fundamental model are quite slow. The disturbance generated by the flux harmonics, however, which are not described in the fundamental model, represent a fast disturbance and therefore a challenge for the disturbance attenuation mechanism.

The nominal machine parameters are found in appendix B.1. Just like for the PMSM, the two tuning parameters are set  $q = 0.5$  and  $T_{LP} = 3T_s$ . It should be remarked that the chosen motor is of low efficiency ('eff3') and of rather compact size. This implies that secondary effects, such as saturation, are considerable. The inductance applied for deadbeat control,  $L_m$ , is the fundamental inductance, as described in section 2.6, the automatic parameter identification from the inverter

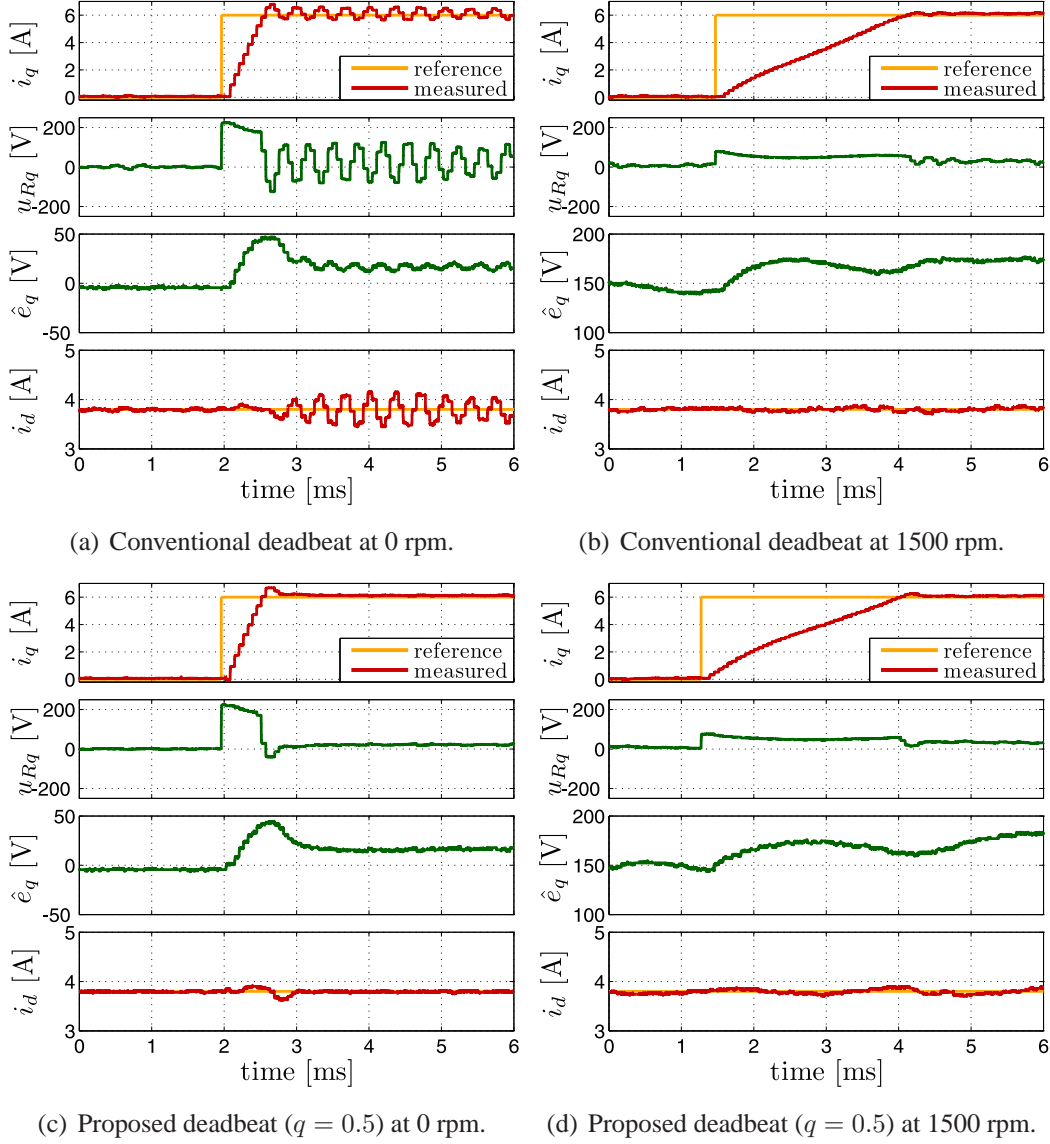
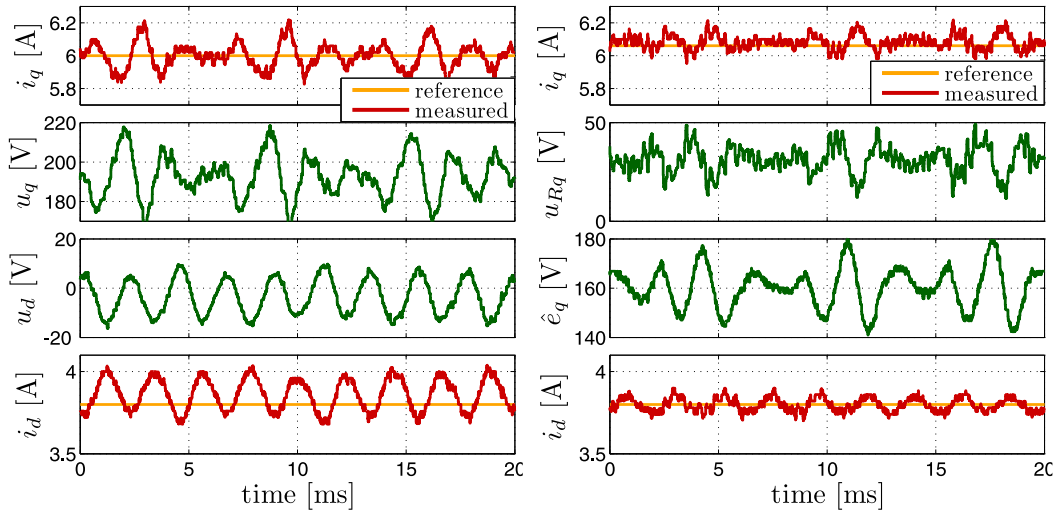


Figure 4.16: Deadbeat control on an induction machine. Top: conventional deadbeat control, bottom: proposed method ( $q = 0.5$ ,  $T_{LP} = 3T_s = 0.1875$  ms).

compensates the saturation. However, the parameter  $L_m$  is higher than the tangential inductance (which is the dynamic gain), therefore the deadbeat controller will be in the situation  $\hat{L} > L$  where the stability problems appear.

In Fig. 4.16 a torque step from zero to nominal torque is analyzed. The top two figures (a) and (b) are the results of a conventional deadbeat controller ( $q = 1$ ).

Clearly, in (a) the parametric uncertainty leads to an instability problem resulting in an undamped oscillation of the torque. An oscillation is also seen on the disturbance estimator, it tries to compensate the oscillation but does not manage to. In (b) in high-speed operation this instability phenomenon is reduced as the available voltage is quite small, but still, an oscillation is seen after the end of the transient. The lower two figures (c) and (d) are the results in identical situation of the proposed deadbeat controller ( $q = 0.5$ ). The instability phenomenon does not appear at all in both cases. Apart from that, performance is identical.



(a) PI controller (symetric optimum) at 1500 rpm. (b) Proposed deadbeat ( $q = 0.5$ ,  $T_{LP} = 3T_s$ ) at 1500 rpm.

Figure 4.17: High-speed operation and harmonic effects on an induction machine.

The low-frequency oscillation of the estimated disturbance  $\hat{e}_q$  in subfigures (b) and (d) is the compensation of flux harmonics. This effect is further studied in Fig. 4.17, where (a) is the response of a PI controller and (b) the response of the proposed deadbeat controller. The PI controller is again tuned according to the symmetric optimum condition as discussed in section 2.7, so the tuning parameters are  $V_R = \frac{\sigma L_s}{4T_s}$  and  $T_n = 8T_s$ . The flux harmonics have a frequency of 150 Hz at 25 Hz excitation, respectively 1500 rpm. Although they are of low frequency and the inductance is high, they result in a current ripple of 0.2 A with the PI controller, but less than 0.1 A with the deadbeat controller.

To conclude on current control, while the conventional deadbeat controller is unstable on the induction machine, the proposed method is well stable and shows good tracking performance and disturbance rejection.

## Rotor flux control

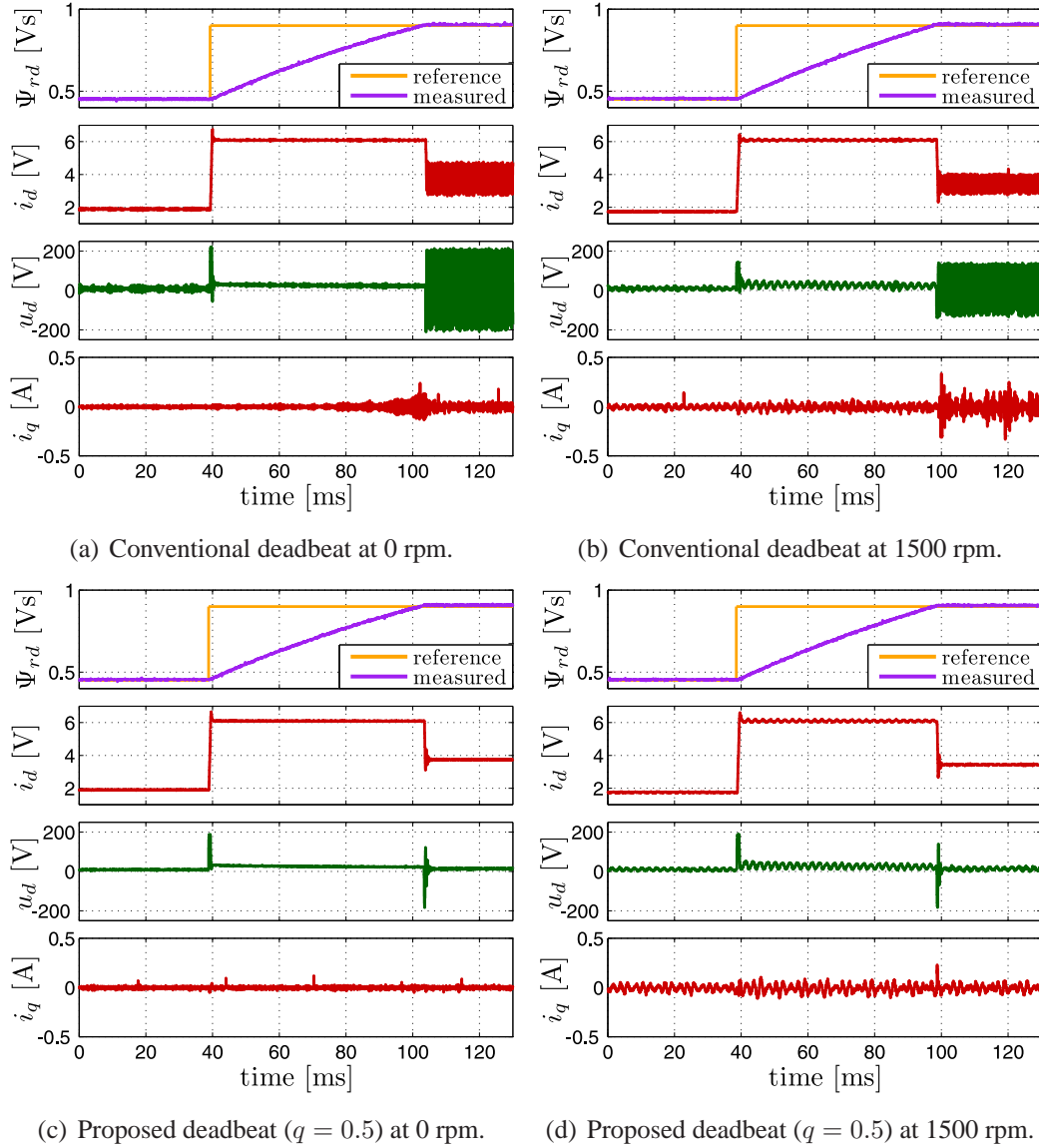


Figure 4.18: Deadbeat flux control on an induction machine. Top: conventional deadbeat control, bottom: proposed deadbeat control ( $q = 0.5$ ,  $T_{LP} = 3T_s = 0.1875$  ms).

The induction motor further offers the chance to analyze second-order dynamics by using flux control in the field-oriented frame. The design of a deadbeat flux con-

troller is very simple because of the flatness of the dynamics. The design is equivalent to two cascaded first-order deadbeat controllers. Furthermore it is strongly simplified as the flux observer, designed based on Euler forward discretization of the rotor model, already acts as delay compensation technique – the value of the flux magnitude  $\Psi_{rd}[k]$  is available at the instant where the reference  $i_{sd}^*[k]$  is computed. The model for flux control is

$$\Psi_{rd}[k+1] = (1 - T_s\eta)\Psi_{rd}[k] + (T_s\eta L_m)i_{sd}[k], \quad (4.12)$$

from which the deadbeat control law is derived as

$$i_{sd}^*[k] = \frac{1}{\eta L_m T_s} \Psi_{rd}^*[k] - \frac{1 - \eta T_s}{\eta L_m T_s} \Psi_{rd}[k]. \quad (4.13)$$

The disturbance estimator is designed using exactly the same procedure as for the current controller, but knowing that the latest available value is  $\Psi_{rd}[k]$  and therefore only one single sampling step delay is involved, it reads as

$$\hat{e}[k] = \hat{e}[k-1] + \alpha \left( i_{sd}^*[k-1] - \frac{1}{\eta L_m T_s} \Psi_{rd}[k] + \frac{1 - \eta T_s}{\eta L_m T_s} \Psi_{rd}[k-1] \right), \quad (4.14)$$

with  $\alpha = \frac{T_s}{T_s + T_{LP}}$  and  $T_{LP} = 3T_s$  as lowpass filter time constant. Saturation is implemented by limiting  $i_{sd}^*[k]$ .

In Fig. 4.18, deadbeat flux control is analyzed. Again the top two subfigures (a) and (b) are the results of a regular deadbeat controller ( $q = 1$ ). At a flux level of  $\Psi_{rd} = 0.45$  Vs, the controller is stable, but once at  $\Psi_{rd} = 0.9$  Vs, saturation is so strong that the current and voltage are subject to undamped oscillations. This appears for zero and high speed, furthermore, in both situations the instability transfers to the  $q$  axis. The lower two figures (c) and (d) are the results of the proposed deadbeat controller ( $q = 0.5$ ). Again there is no more instability while the good performance is untouched.

To conclude, the proposed deadbeat current controller is a very powerful method on the induction motor. Surprisingly, although induction motors are very common, the standard deadbeat design cannot be applied, it is unstable. Even though parameter knowledge is pretty good, the omnipresent saturation effect prevents the standard method to work. The proposed, only slightly modified deadbeat controller, however, works fine and shows excellent performance, in both current and flux control.

The fast disturbance estimator again proves to be a powerful tool to compensate harmonic effects. It is therefore a competitor for the conventional scheme which consists of modeling, identification and compensation of this position- and speed-dependent disturbance [SH98]. Its simplicity also makes it interesting regarding the more involved online methods for this purpose [CKKY98].



## 4.3 Conclusions

This chapter has provided the experimental results and interpretations necessary for a through study of deadbeat control.

Robustness of the controller was studied analytically and experimentally. While the conventional deadbeat scheme works well on easily modelable drives, such as surface-mounted PMSMs, it is too sensitive for application to more complex drives such as the SynRM or the IM. The limitations arise from the omnipresent magnetic saturation effects and the high sensitivity to inductance uncertainty.

A more robust design variant for deadbeat control has been proposed based on feedforward linearization. This concept has been introduced in flatness-based control to improve robustness of model-based control schemes. It basically transforms the model-based controller into a feedforward controller, then, differences between the estimated and the real parameters do not destabilize the system anymore. However, the scheme must be fitted with an additional robust feedback controller to compensate offsets between the predicted and the measured output. In the experiments it turns out that robustness is outstanding, however, available feedback mechanisms such as the disturbance estimator or a PI controller are not sufficiently performant to compensate the time-varying disturbances present in AC drive systems. The performance loss is strong compared to the conventional deadbeat controller.

To help out, an intermediate between the conventional feedback-linearization and feedforward linearization based design was proposed. It aims at giving the robustness advantage but without losing control performance. A 'mix' parameter  $q$  was introduced, which can be seen as trade-off parameter between conventional feedback linearization and the more robust feedforward linearization, or differently said, a trade-off between control performance and robustness.

The proposed deadbeat controller is sufficiently robust to handle any AC machine type, including PMSM, SynRM or IM. The new deadbeat controller is considerably less sensitive to errors, stability range is extended and oscillations caused by errors are well damped. It can handle extreme inductance uncertainties, which appear for instance from saturation effects. Furthermore, the control performance is good enough to reject time-varying disturbances, such as flux harmonics, which result in current and torque ripples when using conventional feedback controllers. A nice side-effect of the proposed method is that the disturbance estimator is better decoupled from the deadbeat controller, especially under parametric uncertainties. This makes the disturbance estimator more robust and the typical lowpass filter, necessary for robustness and time-scale decoupling of the estimator in conventional deadbeat control, can be tuned very fast – at a time constant of only three sampling intervals. Furthermore, sensitivity to measurement noise is considerably improved.



## CHAPTER 5

---

# Optimal control of permanent-magnet synchronous machines using linear model predictive control

---

The control principle of the previous chapter is intended for single-variable systems. If the design is applied to an electrical drive, the two outputs are decoupled and controlled independently. While this is fine in the unconstrained case, one might encounter restrictions if the voltage limitation is active. To avoid performance deterioration in this case, control of the two outputs should not be independent.

While the speed and torque are controlled via the quadrature current  $i_{sq}$ , the direct current  $i_{sd}$  is more an internal variable. The direct current is a degree of freedom whose precise value is less important. A non-salient PMSM is generally controlled with  $i_{sd} = 0$ , however, the direct current can be used to improve power efficiency [MXM00, CTM<sup>+</sup>05] or to improve torque dynamics [CKS95, CS98]. Dedicated control algorithms have been developed in the past for these respective purposes, each algorithm is a solution to the respective optimization problem.

Linear model predictive control is a technique that can combine different operator requirements, such as efficiency and dynamics, respect limitations such as voltage and current constraints, and find an optimal system behavior while regarding the complete multivariable system model, including the interaction of the outputs. In this chapter this advanced technique will be applied to improve the efficiency and the dynamic responses of a PMSM.

## 5.1 Benefits of optimal control

### 5.1.1 Preliminaries: Loss modeling in PMSMs

The losses of an electrical drive system can be divided into controllable and non-controllable losses [KG83]. Controllable losses are, for instance, the ohmic and iron losses, and can be influenced by the controlled variables. By optimizing voltage and current, the controllable losses can be minimized for an arbitrary speed and torque setpoint.

Uncontrollable losses are friction and windage losses, but also stray load losses. Stray load losses describe losses caused by the non-sinusoidal field distribution in the air gap, stator slotting effects and conductor skin effects, they primarily depend on torque, secondarily on the stator voltage frequency and temperature [KG83]. The uncontrollable losses can only be influenced by changing the motor design or by additional hardware, such methods are disregarded in this work.

A simple empirical model of the controllable losses is derived. The ohmic losses are the heating losses of the stator windings and are described by

$$P_{Ohm,S} = \frac{3}{2}R_s(i_{sd}^2 + i_{sq}^2). \quad (5.1)$$

The factor  $\frac{3}{2}$  appears as  $i_{sd}$  and  $i_{sq}$  are described in peak values in space vector notation as described in section 2.1.

The iron losses are mainly caused by the time-varying magnetic flux in the stator [CTM<sup>+</sup>05]. The hysteresis losses are generally proportional to the area enclosed by the characteristic hysteresis curve and the frequency in which that curve is periodically cycled. They are described by

$$P_{Hyst} = \frac{3}{2}k_h n_p \omega_M (\Psi_{sd}^2 + \Psi_{sq}^2) = \frac{3}{2}k_h n_p \omega_M ((L_d i_{sd} + \Psi_{PM})^2 + (L_q i_{sq})^2), \quad (5.2)$$

where  $k_h$  is a constant. The eddy currents are induced electric currents in the stator iron, the inherent losses are described by

$$P_{Eddy} = \frac{3}{2}k_e (n_p \omega_M)^2 (\Psi_{sd}^2 + \Psi_{sq}^2) = \frac{3}{2}k_e (n_p \omega_M)^2 ((L_d i_{sd} + \Psi_{PM})^2 + (L_q i_{sq})^2), \quad (5.3)$$

where  $k_e$  is a constant. The two constants  $k_h$  and  $k_e$  are not part of standard motor parameters, thus they have to be identified. The regular determination of power efficiency of a drive is performed with a wattmeter to determine the input power, and a torque and speed meter to measure the mechanical output power. With such a

measurement, all losses, controllable and non-controllable, are respected, therefore they are not suitable to determine the iron loss coefficients. Instead the method in [USU03] is applied, it leads to the insight that the eddy current losses are neglectable compared to the hysteresis losses. This is due to the optimized stator design of the PMSM which suppresses eddy currents (unlike the low-cost stator of a norm induction motor, for instance). The hysteresis losses, however, are considerable, they are much higher than the ohmic losses. For the PMSM described in appendix B.1, at rated speed and torque, one has  $P_{Ohm,S} = 47$  W and  $P_{Hyst} = 200$  W. This is typical for PMSMs, and also motivates the use of control methods to minimize these losses. Still, possible identification errors must be noted as any errors in the DC link voltage, inverter nonlinearities or machine parameters influence the measurements, which are sensitive because of the quite small value of  $L_d$ .

The model of the controllable losses applied for the controller design is

$$P_{Loss} = \frac{3}{2}R_s(i_{sd}^2 + i_{sq}^2) + \frac{3}{2}k_h n_p \omega_M ((L_d i_{sd} + \Psi_{PM})^2 + (L_q i_{sq})^2). \quad (5.4)$$

### 5.1.2 Optimization of the power efficiency

The direct current  $i_{sd}$  is a degree of freedom that can minimize the losses [CTM<sup>+</sup>05]. For a given setpoint, defined by speed and torque, the quadrature current  $i_{sq}$  and the iron losses caused by the rotating PM flux are fixed. If  $i_{sd}$  is now decreased, the ohmic losses increase quadratically, as shown in Fig. 5.1 (red). At the same time, however, the flux magnitude in the air gap is decreased, thereby the cycle in the hysteresis curve is smaller and the hysteresis losses reduce slightly (black). The total losses (green) decrease to some extent. This field weakening has no influence on the torque, assuming the reluctance torque is negligible ( $\Psi_{PM} \gg (L_d - L_q)i_{sd}$ ).

The efficiency improvement on the analyzed drive is quite small, about 4% loss reduction is obtained at rated speed resulting in an efficiency improvement of only 0.5%. If a PMSM with higher inductance is chosen, less current is needed to weaken the field, and higher improvements are possible. Additionally, if the drive is subject to rotor anisotropy  $L_d < L_q$ , the occurring reluctance torque supports the electromagnetic torque [DS04]. For instance in [CTM<sup>+</sup>05] the efficiency improvement at rated speed and torque is 3.5%. The physical design of the analyzed drive, which can be seen as typical cost-efficient industrial servo drive, do not allow such good results. Still, as there are no additional costs to this method of efficiency improvement, the method is of interest.

However, inverter losses must be discussed. Efficiency improvement of a PMSM implies a higher stator current magnitude and therefore increases the inverter losses. As the switching frequency is fixed, in this case one has 8 kHz sampling frequency

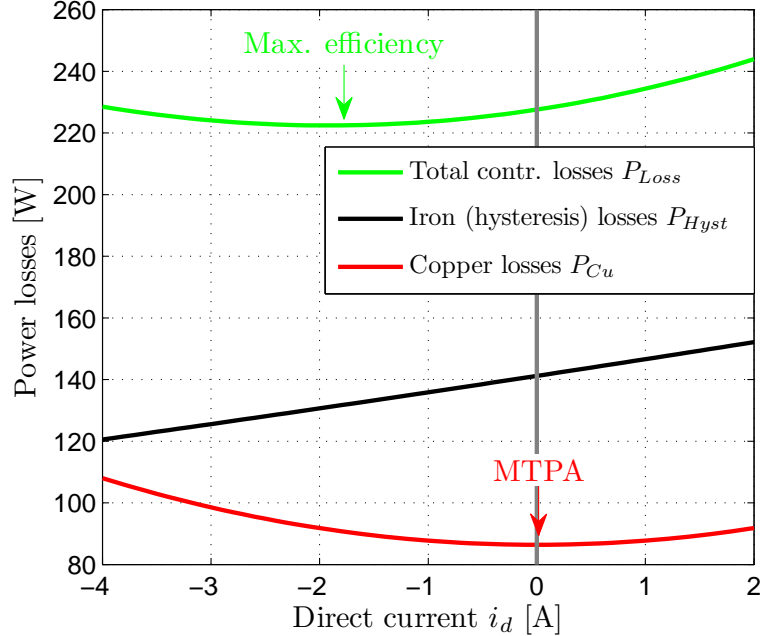


Figure 5.1: Ohmic and iron losses as function of direct current  $i_{sd}$  of a PMSM at 2000 rpm and rated torque.

and a 4 kHz switching frequency, a simple loss model of the inverter may be given as  $P_{Loss,Inv} = k_1|i| + k_2|i|^2$ . The coefficients  $k_1$  and  $k_2$  are determined from the parameters of the inverter IGBTs. While  $k_2$  is negligible and a similar term already included with  $P_{Ohm,S}$ ,  $k_1$  stems from the constant voltage drop of these IGBTs, assuming a constant 1.5 V drop, it is  $k_1 = 3$ . Then the additional losses from the nonzero  $i_{sd}$  are 6 W in no-load full speed, and 1.5 W in full load full speed operation. Even though the loss improvement is extremely small for the chosen machine, the additional inverter losses are even smaller. For small power servo drives these losses are not relevant and can be neglected [Abr00]. In mid-voltage applications, however, these losses should be considered and it was shown that reducing switching frequency is advantageous.

### 5.1.3 Optimization of the dynamic response

Apart from efficiency, dynamics are an important feature of servo drives. A high torque bandwidth can be advantageous, for example for compensation of mechanical vibrations, or for special applications. In low speed, this is never a problem, but

in high speed, where the induced voltage is close to the maximum voltage, torque dynamics are considerably slowed down. In the model equation

$$L_q \frac{d}{dt} i_{sq} = -R i_{sq} + u_{sq} - n_p \Psi_{PM} \omega_M - L_d i_{sd} n_p \omega_M, \quad (5.5)$$

the term  $n_p \Psi_{PM} \omega_M$  dominates in high speed. If it comes close to the maximum value for the control input  $u_{sq}$ , the quadrature current  $i_{sq}$  can only be increased slowly. The deadbeat controllers proposed in the previous chapter are fast controllers, but as they are SISO controllers they cannot bypass this physical limit. Also controllers operating purely on the MTPA trajectory are generally slower in transient operation.

To help out, the direct current  $i_{sd}$  can be exploited as a degree of freedom to improve dynamics if the setpoint is close to maximum voltage. The disturbance voltage is  $n_p \omega_M (\Psi_{PM} + L_d i_{sd})$ , and in high speed, it can be reduced by imposing field-weakening  $i_{sd} < 0$ . The gap between disturbance voltage and maximum voltage increases, therefore  $\frac{d}{dt} i_{sq}$  increases and the torque dynamics are improved. Thus, by exploiting cross coupling in high speed operation, optimal adjustment of  $i_{sd}$  can improve dynamics. A minimum-time control algorithm based on calculus of variations was proposed in [CKS95, CS98], this controller enforces  $i_{sd} = 0$  in steady state but an appropriate  $i_{sd} \neq 0$  during torque transients.

#### 5.1.4 Problem statement: Receding horizon formulation

The stated goals for the optimal controller are formulated as a receding-horizon optimization problem, which is solved at every sampling step. Both goals are obtained by field-weakening, therefore, qualitatively, both goals can be obtained.

To obtain the optimization benefits, a torque controller is sufficient. Although efforts were done to design predictive controllers without a cascaded structure [BBPZ09], this is not necessary. The degree of freedom  $i_{sd}$  is fully available at the torque control level. Furthermore, the torque controller formulation enables some key simplifications to formulate a linear-quadratic optimization problem.

#### Linearization of the machine model

As linear-quadratic optimization problems with linear constraints are simpler to solve in real-time, the machine model is linearized. Assuming that the rotor speed does not change too much over the optimization horizon  $T$ ,

$$\frac{d}{dt} \omega_M(t) \approx 0 \Rightarrow \omega_M(t) = \text{const.} \quad \forall t \in [0, T], \quad (5.6)$$

the PMSM model and the voltage equations become linear. The assumption is justified if the current control loop is faster than the speed control loop. The electrical subsystem of the machine, consisting of the quadrature and direct currents  $i_{sq}$  and  $i_{sd}$ , is given as

$$L_d \frac{d}{dt} i_{sd} = -R i_{sd} + n_p \omega_M L_q i_{sq} + u_{sd}, \quad (5.7)$$

$$L_q \frac{d}{dt} i_{sq} = -R i_{sq} - n_p \omega_M L_d i_{sd} - n_p \omega_M \Psi_{PM} + u_{sq}, \quad (5.8)$$

$$\tau_M = \frac{3}{2} n_p \Psi_{PM} i_{sq}. \quad (5.9)$$

The nomenclature and parameters are found in appendix B.1. It is noted that the reluctance torque  $\tau_M^R = \frac{3}{2} n_p (L_d - L_q) i_{sd} i_{sq}$  is neglected, as this term is very small compared to the electromagnetic torque in surface-mounted PMSMs or in machines with small saliency. Furthermore, it would render the model nonlinear, requiring either nonlinear optimization methods [DS04] or more extensive linearization by defining the nonlinear terms as external disturbances [BBPZ09, BKKP11].

### Cost functional and tuning

The formulation of a suitable cost functional is a key point in predictive control, as it is the only tuning possibility of the control scheme. Both goals, the control error for good dynamical performance and the machine losses for better efficiency, are included in the cost functional. By choosing the cost functional and weights well, it is possible to find a good trade-off between both goals during transients, and eventually to fulfill both goals in steady-state. The cost functional for the predictive torque controller is

$$J = \int_0^t (P_{Ctrl}(t) + w_L P_{Loss}(t)) dt + T P_{Loss}(T), \quad (5.10)$$

which trades off the squared control error from the constant torque reference  $\tau_M^*$ ,

$$P_{Ctrl}(t) = (\tau_M(t) - \tau_M^*)^2, \quad (5.11)$$

with the machine losses  $P_{Loss}(t)$  defined above. The last term in  $J$  is the end-weight of the control error, it enforces a better convergence in some operation points. It is weighted with the horizon  $T$  such that it has a similar weight as the control error term in the integral; assuming the control error decreases, however, the end-weight term is of course much smaller. In appendix F.1 an analysis of steady-state accuracy is shown, while efficiency optimization is guaranteed, a negligible steady-state



offset appears on the torque. More important, however, is behavior during transient operation.

There are two tuning parameters, the weight  $w_L$  and the prediction (and control) horizon  $T$ . The weight  $w_L$  trades off losses and torque tracking, it was set 0.05, the value was determined heuristically. The optimization horizon is set to  $T = 2$  ms such that the cost functional  $J$  includes the complete torque setpoint change if speed is not too high. It is important that the optimization horizon is high enough, otherwise the open-loop and closed-loop trajectories differ and the behavior is strongly suboptimal. This is illustrated in Fig. 5.2. The optimized open-loop trajectory differs from the closed-loop trajectory, latter one results from regeneration at every sampling step. If the horizon is too small, due to the end-weight of the control error, a significant difference appears. Then, the closed-loop trajectories simply don't fit the cost functional anymore and are suboptimal. For a horizon higher than required for the setpoint change, the difference between open- and closed-loop trajectories becomes smaller, and the closed-loop trajectories can be assumed optimal regarding the open-loop cost functional  $J$ .

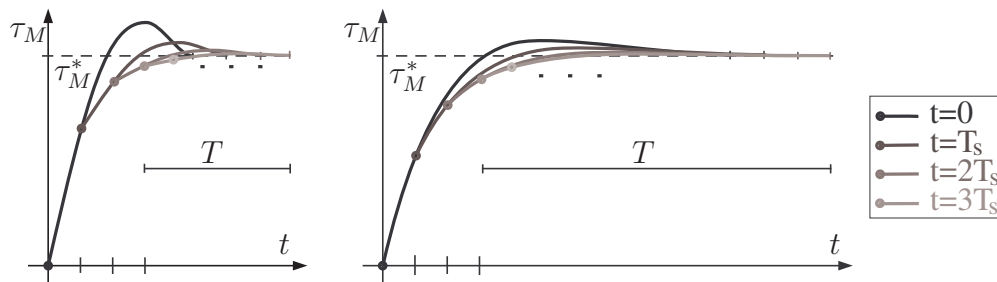


Figure 5.2: Exemplary torque setpoint change to describe open- and closed-loop trajectories in model predictive control. Left: small horizon, right: high horizon. Circles: reinitialization points of each trajectory planning iteration.

### System constraints

The most important nonlinearities of a PMSM, in view of control, are the voltage and current limitations. The current constraints prevent overheating of the machine and protect the inverter, and the voltage is limited by the maximum output voltage of the voltage source inverter. The voltage constraints limit rotor speed as well as current dynamics in high-speed operation. Both constraints are linearized, in order to be computationally efficiently treated.

The current range for the direct current  $i_{sd}$  is limited to  $i_{sd}^{min} \leq i_{sd} \leq 0$ . Only negative values of  $i_{sd}$  are desirable, as they improve power efficiency and reduce the induced voltage by weakening the flux magnitude in the stator [CTM<sup>+</sup>05,BBPZ08]. The lowest value  $i_{sd}^{min}$  is the optimum value at rated speed ( $\frac{\partial}{\partial i_{sd}} P_{loss} = 0$ ) and is given as

$$i_{sd}^{min} = -\frac{L_d \Psi_{PM}}{L_d^2 + \frac{R}{n_p \omega_{MN} k_{Fe}}}, \quad (5.12)$$

which is independent of the quadrature current  $i_{sq}$  if the reluctance torque (the saliency in the torque equation [MKM04]) is neglected. The value is doubled to enable further field-weakening to improve dynamics in high speed. For the quadrature current  $i_{sq}$ , the largest possible range of values should be available. The resulting linear constraints in trapezoid form are shown in Fig. 5.3, they almost completely fill the current region of interest. A linearization in the synchronously rotating frame is thus acceptable, and is also used in other MPC implementations [BBPZ09].

The approximation of the voltage constraints is a bit more difficult. The  $q$ -axis should not be restricted, as the induced voltage is aligned to it and is the largest value that will appear. A steady-state analysis of the system equations (5.7), (5.8) shows that a rectangular voltage area results

$$R i_{sd}^{min} - n_p L_q \omega_M^{max} i_{sq}^{max} \leq u_{sd} \leq n_p L_q \omega_M^{max} i_{sq}^{max}, \quad (5.13)$$

$$-R i_{sq}^{max} + n_p L_d \omega_M^{max} i_{sd}^{min} - n_p \Psi_{PM} \omega_M^{max} \leq u_{sq} \leq R i_{sq}^{max} + n_p \Psi_{PM} \omega_M^{max}. \quad (5.14)$$

This rectangle (dark gray on Fig. 5.3) is expanded such that the outer circle of the voltage limitation is hit (light gray on Fig. 5.3). During dynamical transients, the voltage vector points to one of the outer corners, subsequently touching the outer limiting circle. Therefore, a linear approximation of the voltage limits as a rectangle in field-oriented frame by the presented method, as shown on Fig. 5.3, does not limit the steady-state operational range and only marginally affects dynamics. A less restrictive method is presented in [RMM10], where a time-varying constraint in form of a hexagon in stator frame is proposed. While such a linearization is possible with the underlying predictive control algorithm, the method in the  $(d, q)$ -frame is chosen for simplicity and to prevent possible current ripples.

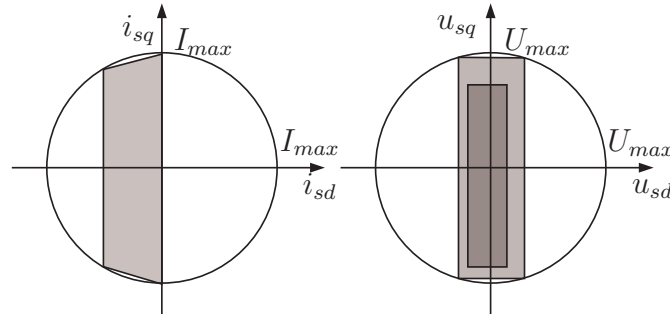


Figure 5.3: Affine approximation of the current and voltage constraints. Circle: feasible set of current and voltage vectors, gray: feasible set after linearization of the constraints.

## 5.2 Linear model predictive control

### 5.2.1 Predictive control, MPC, and flatness

The term 'predictive control' denotes a large class of controllers rather than a specific algorithm, with the sole common characteristic that there is a prediction in some sense. Especially in electrical drives, a vast variety of predictive controllers exist. These include direct torque control (DTC), predictive torque control (PTC), generalized predictive control (GPC), deadbeat control, and many more, overviews are found in [LKKS10, CKK<sup>+</sup>08]. The algorithms can be divided by a continuous or discrete control set, their ability to handle constraints, and more.

Even though some authors call their works model predictive control because they include model-based predictions in some sense, the term model predictive control (MPC) is a well-defined controller design in the control literature [ML99, Lee11]. MPC is therefore a specialized variant of predictive control, and only a handful of recent works study MPC for electrical drives [LK05, BBPZ09, BKKP11, RMM10, LKKS10]. The reason is that MPC comes with a high computational burden and is hard to implement due to the high sampling rates, furthermore, interest has been low as it was long unclear what the advantages of such a design could be. So far, the discovered advantages are the possibility of cascade-free speed control [BBPZ09] and the simplified inclusion of sensor filters [LKKS10].

In [FM00a] three salient features that characterize MPC uniquely amongst the many optimal or predictive control methods are proposed:

- The system *behavior is predicted by a model* and the prediction is explicitly

included in the control.

- The problem includes *input and output constraints* which are treated in the controller design.
- A *feedback mechanism* deals with perturbations and uncertainties.

The categorization makes sense and is both precise and abstract, although it is not accurate, for instance, generalized predictive control (GPC) is also seen as MPC even though it is unconstrained.

The definition in [ML99] is more technical. Accordingly, linear model predictive control is a control method which is based on formulating and solving repeatedly at *every* sampling step an open-loop optimization problem. It includes constraints by defining inequality constraints in the optimization problem, differing it from optimum LQ state controller design.

With the definition of [FM00a], however, the optimization problem may be seen as one mean amongst many alternatives to generate trajectories by predictions that also account for the constraints, the optimizer as such is not the core of MPC. The generalized definition and the mathematical framework in [FM00a] leaves more freedom and extends flatness-based control, which, in the classical sense, means trajectory generation and trajectory tracking [Lév09], such that continuous-time flatness-based methods can be applied for MPC. Several computationally efficient continuous-time trajectory generation methods exist [vNM98, vL02, GF06], incorporating prediction, optimality, constraints and setpoint changes in some way, but not necessarily quadratic programming (QP) and the related high computational burden.

Full-fledged MPC design based on flatness therefore consists of an extended trajectory generation stage that includes prediction based on the model (not just setpoint interpolation) and the explicit respect of constraints (not just saturation), combined with a robust tracking controller.

### 5.2.2 Online optimization: A real-time problem

The major obstacle in implementing predictive control schemes is the limited computational power, inherited by the high sampling rates. The most widespread schemes trade computational feasibility against compromises in the problem formulation, for instance, generalized predictive control (GPC) has a high prediction horizon but is unconstrained [LKKS10], whereas predictive torque control (PTC) is constrained but so far only reaches 2 steps of prediction [CKK<sup>+</sup>08]. To obtain the advantages claimed by the classical MPC formulation on constrained multivariable systems,

both, inclusion of constraints and a high prediction horizon are required. In this application, these two requirements also make physical sense, as discussed in the previous section. The problem formulation shall not be simplified, but the solution of it.

Using the continuous control set in the field-oriented frame, the analytical problem description enables the use of efficient optimization algorithms to maximize the obtained information for a given computational power. The online solution of the linearly constrained linear-quadratic problem, typical for MPC, requires quadratic programming (QP) algorithms, which are, however, computationally too expensive for drive systems.

In the efforts performed to apply MPC to mechatronical systems, the many extensions and advances of optimizers (warm start, etc.) were recombined and it was shown that QP can be improved to operate at sampling rates up to 200 Hz [WB10]. This also means that even though all available means were implemented on the best computers, the sampling rates for electrical drives are out of sight for QP.

A recent development is the use of explicit MPC, where an offline solution is computed and stored as look-up table in the real-time controller [BBPZ09]. The scheme reaches 5 prediction steps with constraints. However, the design is subject to some issues. The complexity of the controller increases exponentially with the number of constraints and state variables, so that the problem formulation must be kept compact. A high complexity increases the size of the look-up table and leads to real-time problems. Furthermore, from a practical perspective, the explicit solution does not allow fine tuning of weights and model parameters during commissioning, the lengthy offline precalculations must be repeated, a major limitation to explicit MPC [Lee11].

Development of fast online MPC algorithms and their application to new systems, including mechatronic and power electronic systems, has been declared as major research objective of the decade for MPC [Lee11].

Recently, an online algorithm based on a fast-gradient method (FGM) was proposed for reference tracking control of a grid-connected inverter [RMM10]. The optimizer is of a different class than QP. In [RMM10], open-loop software tests show that the runtime of the controller is acceptable, and simulations indicate superior performance compared to explicit MPC. An advantage of online optimization is the possibility to manipulate or adapt parameters, which results in simplified commissioning. Another advantage are time-varying constraints, which are better adopted to the physical voltage limitation.

Some other QP-like algorithms that are specifically designed for MPC applications are in research. The main assumption therein is that the solution of the actual optimization problem is not much different from that of the previous sampling step,

and only few additional iterations are required at each sampling step. This makes sense for the typical MPC tracking formulation [ML99], but for electrical drives, it turns out the physically motivated optimization problem could change drastically at a reference step, so that a warm-start technique only has marginal improvement.

Before discussing in detail the real-time feasible optimization algorithm, the runtime of available optimizers is discussed. This is necessary to determine the highest possible amount of optimization parameters. The fastest optimizer with constraints is still the good old and widely known linear programming (LP) method. Table 5.1 shows some worst-case computational results of LP (simplex method from [PTVF92]) as a function of the number of free parameters (CPU: 1.4 GHz Pentium M in an industrial PC104+ computer). More parameters lead to a higher number of iterations which are also more complex; the worst-case number of iterations is the number of parameters plus the number of constraints [Pie69]. As in the underlying application, the constraints are decoupled, however, this worst-case is not to be expected. The maximum runtime is given by the sampling rate minus latency of input/output, therefore at 8 kHz sampling rate, it must be less than about 110  $\mu$ s. Thus, at best, 12 parameters can be optimized if an LP method is used.

Table 5.1: Runtime of a linear program for some worst-case problems on a 1.4 GHz CPU

Parameters	Constraints	Iterations	Runtime [ $\mu$ s]
20	44	67	769
12	28	34	165
8	20	10	35

As comparison, the results of a quadratic programming solver are shown in table 5.2. The QP algorithm is from the open-source mathematics library 'GSL GNU Scientific Library', version 1.15, released in spring 2011. The results with 6 parameters compare to those of 12 parameters in the LP, it is a similar optimization problem and in QP the variables can be positive and negative, while in the LP these are limited to positive values [Pie69]. Clearly, the runtime is too high for real-time application, the LP solver shows to be about 20 to 30 times faster than the QP solver.

By solving the computational problem without compromising the MPC formulation, meaning with both a high horizon and constraints, the merits of this controller can be studied. The main contribution of the chapter is a fast near-optimal trajectory generator, which, embedded in the framework of [FM00a], leads to a full-fledged MPC. This leads to the first MPC with online optimization experimentally applied to a drive system.

Table 5.2: Runtime of a quadratic program on a 1.4 GHz CPU

Parameters	Constraints	Iterations	Runtime [ $\mu$ s]
6	28	19	3621
4	20	10	1151

### 5.2.3 A simple and efficient LQ optimization algorithm

A trajectory generation algorithm related to flatness-based methods [GF06] is developed. It can optimize a quadratical cost function like (5.10) with linear constraints like in Fig. 5.3. As major differences to standard algorithms, it is applying a continuous parameterization instead discretization, and the computationally efficient linear programming solver is used instead of quadratic programming or iterative gradient search. Even though LP is used, it is still quadratic optimization; the unconstrained solution to the quadratic cost is calculated first, then, constraints are included using the LP solver.

As a first simplification the trajectories for the current are defined as degree  $n$  power series with undetermined coefficients  $\alpha_{ij}$ ,

$$i_{sd}(t) = \sum_{k=0}^n \alpha_{dk} \frac{t^k}{T^k}, \quad i_{sq}(t) = \sum_{k=0}^n \alpha_{qk} \frac{t^k}{T^k}, \quad t \in [0, T]. \quad (5.15)$$

This definition reduces the dimensionality of the generated trajectories rather than their length and is referred to as Ritz parameterization [Pie69]. It is an alternative parameterization to the typical Euler discretization. While some applications use more sophisticated basis functions, for instance higher-order polynomials (Laguerre and Legendre polynomials), because of higher numerical stability. The simple choice of power series is adequate for this application. Moreover, in [BS97] it was shown that only the polynomial degree but not the type of polynomials is important for convergence.

The first coefficients  $\alpha_{d0}$  and  $\alpha_{q0}$  are the initial conditions, and the remaining 6 coefficients are determined by optimization. A high prediction horizon is obtained for a relatively small number of parameters. Due to the analyzed computational limitations,

$$n = 3 \quad (5.16)$$

is chosen as polynomial degree. Fig. 5.4 illustrates the computational advantage, with 3 parameters at 8 kHz sampling rate, using a discrete description, the prediction horizon is 0.375 ms, but with a degree 3 polynomial, a well-conditioned setpoint

change can basically be described over the desired prediction horizon of 2 ms. More complex trajectories may not be expected for the underlying application.

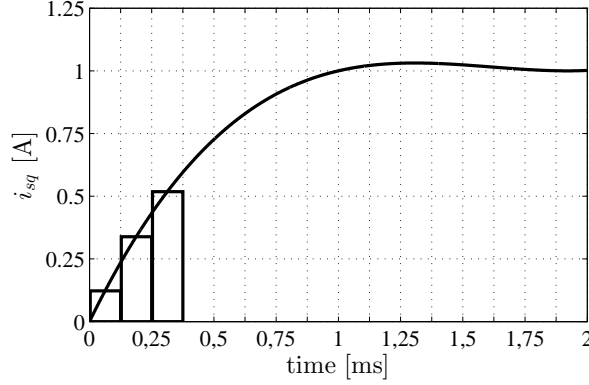


Figure 5.4: A trajectory described by 3 free parameters. Discrete-time horizon is 0.375 ms, continuous polynomial-based trajectory is well-conditioned at the desired horizon length 2 ms.

As second simplification, the voltages  $\mathbf{u}_{s,dq}(t)$  are substituted in the optimization problem. The model equations (5.7) and (5.8) are equality constraints in the optimization problem, they describe the relationship between the outputs and the inputs, and in the case of a flat system, these equality constraints can be substituted directly without solving a differential equation [SRA04]. By algebraically differentiating the currents (5.15), the expression for the current derivatives is

$$\frac{d}{dt}i_{sd}(t) = \sum_{k=1}^n k\alpha_{dk} \frac{t^{k-1}}{T^k}, \quad \frac{d}{dt}i_{sq}(t) = \sum_{k=1}^n k\alpha_{qk} \frac{t^{k-1}}{T^k}. \quad (5.17)$$

Substituting the algebraic expressions of  $i_{s,dq}$  and  $\frac{d}{dt}i_{s,dq}$  in the model equations (5.7) and (5.8), the expressions for the voltage

$$u_{sd} = Ri_{sd} + L_d \frac{d}{dt}i_{sd} - n_p \omega_M L_q i_{sq}, \quad (5.18)$$

$$u_{sq} = Ri_{sq} + L_q \frac{d}{dt}i_{sq} + n_p \omega_M L_d i_{sd} + n_p \omega_M \Psi_{PM}, \quad (5.19)$$

are used directly to replace these two variables in the constraints by functions of the undetermined coefficients  $\alpha$ . This way the voltages do not need to be represented by additional parameters, and equality constraints are avoided in the optimization.



The cost functional  $J$  is then a quadratic function of the unknown parameters  $\alpha$  and of the parameters which were assumed constant, namely the motor parameters, the measured currents, the speed  $\omega_M$  and the torque reference  $\tau_M^*$ . Defining the vector of undetermined coefficients as  $\alpha = (\alpha_{d1}, \alpha_{d2}, \alpha_{d3}, \alpha_{q1}, \alpha_{q2}, \alpha_{q3})^T$ , it is rewritten as

$$J = \alpha^T Q \alpha + q^T \alpha + q_0. \quad (5.20)$$

Because of the parameterization with a polynomial basis, convexity must be discussed [GF06], in (5.10) the weight matrix was  $\mathbb{R}^{2 \times 2}$  whereas now it is extended to  $Q \in \mathbb{R}^{6 \times 6}$ . The proof that convexity is maintained with this transformation is given in appendix F.2 for the unconstrained case. As only linear inequality constraints are regarded, which always form a convex set, the result is also valid in the constrained case. To illustrate the further developments,  $J$  is represented graphically in Fig. 5.5 (left).

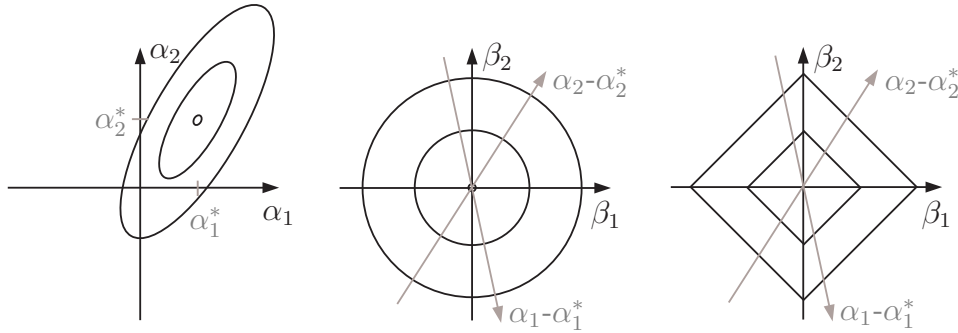


Figure 5.5: Transformation and linearization of the optimization problem. Lines are equimagnitude contours of the cost function. Left: original problem, middle: transformed least-distance problem, right: transformed and linearized problem.

The inequality constraints are also parameterized with the polynomial. Exact parameterization requires linear matrix inequality (LMI) methods, which are, however, computationally too demanding for this application. Necessary and sufficient conditions for such univariate polynomials over an interval  $t \in [0, T]$  exist, but require numerically intensive algorithms to establish corresponding linear matrix inequalities [HL03]. A computationally more efficient way is to sample the trajectories for  $i_{s,dq}(t)$  and  $u_{s,dq}(t)$  at an interval  $\frac{T}{n}$  as shown in Fig. 5.6. A degree 3 polynomial is simply constrained at 4 points to be negative. In appendix F.3 it is

proven that if an additional interlay  $\Delta$  is added, the transformation guarantees maintenance of the original constraints. For instance, a constraint  $i(t) \leq I \quad \forall t \in [0, T]$  is parameterized as the four inequality constraints

$$i(0) \leq I, \quad (5.21)$$

$$i\left(k\frac{T}{n}\right) - \Delta i(0) \leq I, \quad k = 1..3, \quad (5.22)$$

where  $\Delta = 0.064$  is the respective interlay constant for  $n = 3$  (as calculated in the appendix F.3). While the transformed constraints are sufficient, meaning guaranteed to be maintained, they are not necessary, therefore a bit too restrictive. The restriction of 6.4% of the initial distance to the constraint is acceptable, however, also as it shrinks as  $i$  converges to its boundary  $I$ . The first constraint on the initial condition does not need to be included, and the remaining 3 conditions are affine functions of  $\alpha$ , but not of  $t$ , such that they can directly be included in linear-quadratic optimization in the parameter space  $\alpha$ .

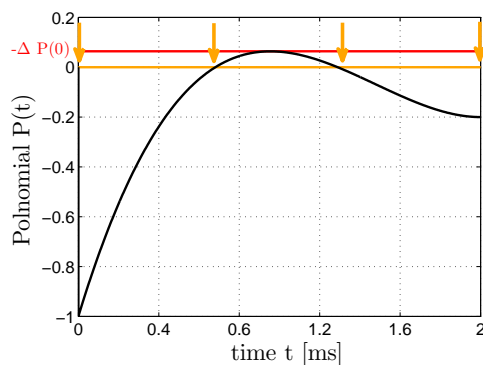


Figure 5.6: A degree  $n$  polynomial trajectory (black) constrained at  $n + 1$  points to be nonpositive (orange arrows) can not exceed the upper bound of  $-\Delta \cdot P(0)$  (red line) (Proof: appendix F.3). Here,  $n = 3$ ,  $P(0) = -1$  and  $\Delta = 0.064$ .

As  $J$  is convex, the unconstrained optimum  $\alpha_0^*$  is found algebraically by solving first-order necessary conditions,

$$\alpha_0^* = -\frac{1}{2}Q^{-1}q. \quad (5.23)$$

Then, by an affine coordinate transformation to a new coefficient vector

$$\beta = A(\alpha - \alpha_0^*), \quad (5.24)$$

the problem can be reformulated as least-distance problem, i.e. a quadratical cost describing the distance to the unconstrained optimum. The linear transformation includes both a coordinate rotation as well as scaling, and is found with

$$\mathbf{A}^T \mathbf{A} = \mathbf{Q}, \quad (5.25)$$

which can be solved with the Cholesky decomposition  $\mathbf{A}^T = \text{Cholesky}(\mathbf{Q}^T)$ . As result, the cost functional looks much simpler and reduces to a sum of squares

$$J = \boldsymbol{\beta}^T \boldsymbol{\beta}, \quad (5.26)$$

and can be represented as in Fig. 5.5 (middle). This constrained least-distance problem is already simpler to solve than the original problem. In the next step, the least-distance problem is linearized around the unconstrained optimum, see Fig. 5.5 (right). The squares in the cost function are replaced by absolute values

$$J = \boldsymbol{\beta}^T \boldsymbol{\beta} = \sum_i \beta_i^2 \approx \sum_i |\beta_i| = J'. \quad (5.27)$$

In the LP standard form, furthermore, only positive parameters are possible, therefore the variables are replaced by  $\beta_i = \beta_{ip} - \beta_{in}$ , with  $\beta_{ip}, \beta_{in} \geq 0$ . If  $\beta_i$  is positive,  $\beta_{ip} \geq 0$  and  $\beta_{in} = 0$ , and vice versa [Pie69]. The absolute value can then be replaced by  $|\beta_i| = \beta_{ip} + \beta_{in}$ . Equivalence is guaranteed by minimizing the (positive) sum, such that at least one variable of each pair  $(\beta_{ip}, \beta_{in})$  will be zero [Pie69]. The linearization of the cost function inherits a large difference (and error) in the value of  $J$  [RR00], but the values of the coefficients  $\boldsymbol{\beta}$  are not affected that much: the least-distance problem is not so different in the linear form as it would have been in the quadratical form. Furthermore, a difference only appears if a constraint is active, the unconstrained optimum is the same. It can be shown that the resulting cost inherited by the linearization is

$$J' = J_0 + 2n \cdot J_C \quad (5.28)$$

in the worst case, where  $J_0$  is the unconstrained cost, and  $J_C$  the extra cost when considering constraints. The suboptimality is therefore bounded.

The linear constraints on  $\boldsymbol{\alpha}$  are as well transformed to the new variables  $\boldsymbol{\beta}$  with (5.24). Therefore, after all the transformations, the problem is available in standard form for linear programming, say a linear cost with linear inequality constraints in the parameter space  $(\boldsymbol{\beta}_p, \boldsymbol{\beta}_n)$ , and a simplex solver [PTVF92] can be run. The optimal solution  $\boldsymbol{\beta}^*$ , the output of the LP, has to be retransformed to find the optimum in the original coordinates

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_0^* + \mathbf{A}^{-1} \boldsymbol{\beta}^*. \quad (5.29)$$

The optimal result  $\alpha$  is used in (5.15) and the motor model to obtain the optimal trajectories for the currents and voltages.

To conclude, the algorithm consists of a continuous parameterization using a degree 3 power series with undetermined coefficients  $\alpha$ . The problem is transformed to a constrained least-distance problem using the Cholesky decomposition and the unconstrained optimum. Then, the quadratic cost is approximated by a linear cost to apply linear programming (LP). The constraints become univariate polynomials because of the transformation, they were sampled and rewritten as parametric inequalities.

## 5.2.4 Implementation and control structure

### Software implementation aspects

All computations of the previous section are done using a computer algebra tool (Maplesoft's<sup>®</sup> Maple<sup>™</sup>). From the initial optimization problem and the transformation, the matrices (initialization of floating-point arrays) for the LP solver are generated using the C code generation toolbox. The real-time software thus consists of array assignments, which is automatically generated code to initialize one matrix and two vectors of floating-point variables, a simplex LP solver from [PTVF92], and some post-processing again from algebraic math. As the assignment is based on symbolic calculations, the motor parameters can be changed online. However, the majority of parameters is defined as constant, the compiler can then optimize code size as well as runtime of the initialization stage. The implementation flow is sketched in Fig. 5.7.

Alternatives to this procedure were evaluated. The direct use of a quadratic program (QP) is too slow, as shown in table 5.2, with 6 parameters, 28 constraints and 19 iterations it requires 3621  $\mu s$ . The proposed method is therefore considerably faster than QP. If the problem is first transformed to least-distance, the number of iterations and the runtime reduce by about 50%. It should be remarked that due to the more specific cost function and the polynomial parameterization, the resulting optimization problem is more complex than the regular MPC problem with Euler parameterization, which is a major reason why the good timing results from QP found in the literature cannot be obtained here.

Online calculation of the trajectory generation algorithm instead of computer algebra generated code was also analyzed. The online calculated matrix inversion with 6 parameters takes 13  $\mu s$  and the Cholesky decomposition 40  $\mu s$ . Additionally, this requires the use of a scientific mathematics library. The runtime of the initialization stage is thereby expanded from about 10  $\mu s$  for the proposed computer

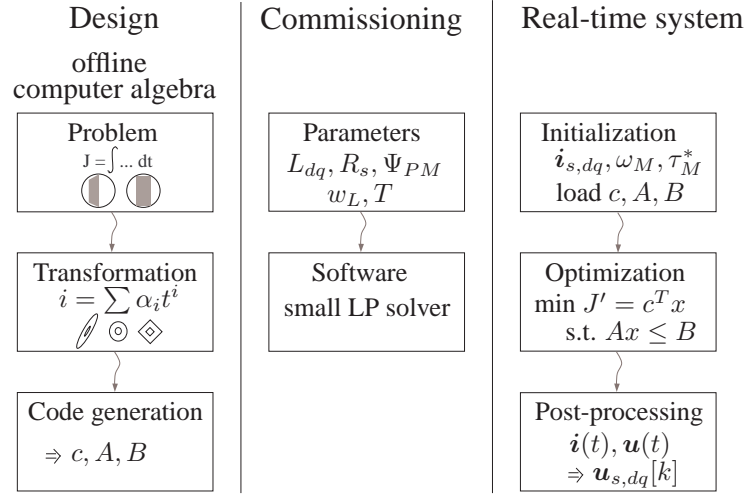


Figure 5.7: Implementation of the proposed trajectory generation resp. MPC algorithm. The tasks in the left column are done once, those in the middle column are done for every new drive system, and those in the right column are carried out in real-time at each sampling step.

algebra method to more than  $50\mu s$ .

The choice to use a computer algebra tool and automatic code generation turns out as the easiest way to implement and as well as the computationally most efficient way. The automatically generated code is copy-and-pasted to the real-time software, only a small LP solver like [PTVF92] has to be added. The chosen LP implementation, however, seems to be a somewhat slow implementation, runtime improvements are possible.

### Predictive control structure

The control structure is shown in Fig. 5.8. A cascaded control structure is chosen as speed is assumed constant for trajectory generation. As the mechanical plant is generally only roughly known, this overlying speed controller is advantageous. An optimizing model-based controller is used for the electrical subsystem of the motor as the parameters are known, but for the mechanical part, any robust feedback controller can be chosen.

The trajectory generation algorithm is embedded in the framework proposed in [FM00a] to realize continuous-time flatness-based MPC. As described in section 2.5, the current measurements are delay-compensated [MKY03]. Then they are

used as initial conditions, along with the updated speed and reference, to generate a new trajectory at every sampling step. The optimal control input trajectory  $\mathbf{u}_{s,dq}(t)$  is discretized and  $\mathbf{u}_{s,dq}[k]$  is modulated at the next interrupt.

Additionally, for steady-state accuracy, a disturbance observer is necessary, for instance to compensate modeling errors of the parameters and harmonic effects. Even though the induced voltage is modeled, variations in  $\Psi_{PM}$  caused by either heating or identification errors cause an offset of the current  $i_{sq}$ . The design is identical as in the previous chapter [KY01]. Furthermore, robustness to parametric uncertainties can be improved in exactly the same way as proposed in the previous chapter, i.e. by replacing  $\mathbf{i}_{s,dq}[k-1]$  by a weighted sum of measurements and reference trajectories.

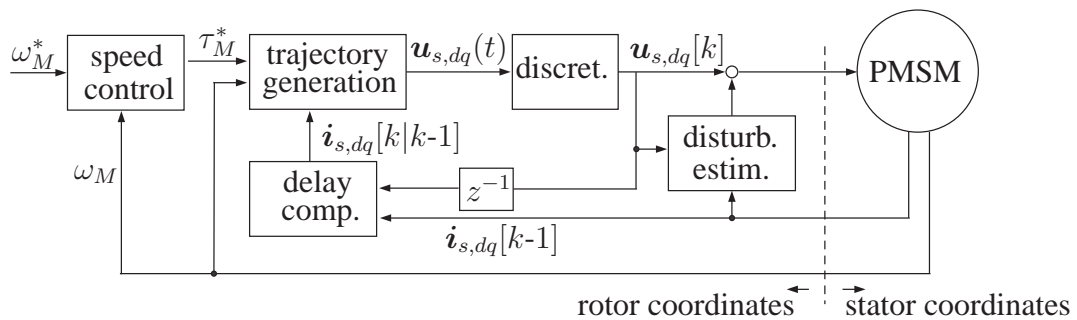


Figure 5.8: Control structure of the predictive torque controller cascaded by PI speed control.

### 5.3 Results

The proposed control scheme was implemented in numerical simulations and in experiments on the test bench. The numerical simulations shall give insight in both the trajectory generation algorithm and in the suboptimality of the method. The experimental results demonstrate the performance of the scheme. The PMSM is described in appendix B.1, tuning is as described in the developments of this chapter. The sampling rate is reduced to 8 kHz (compared to 16 kHz in the previous chapter) to have a longer sampling interval.

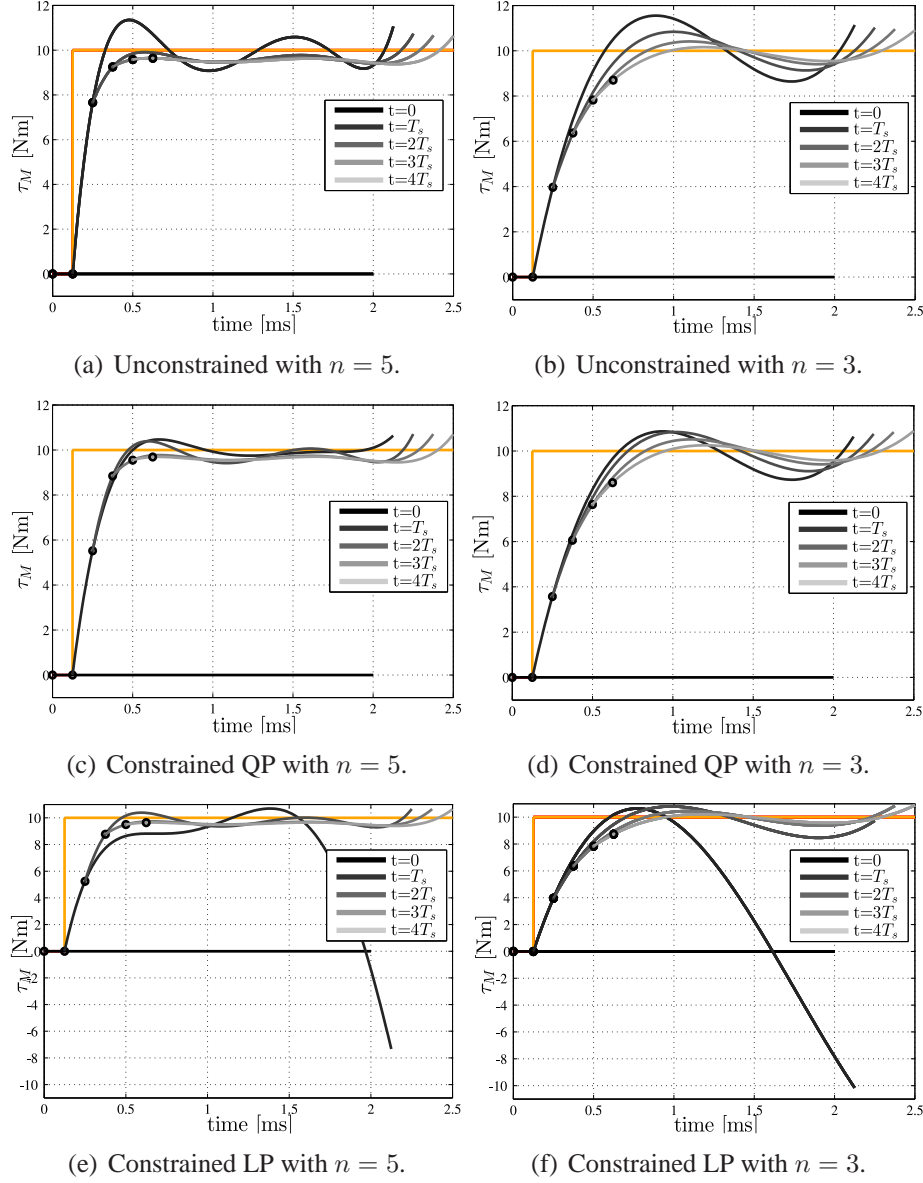


Figure 5.9: Simulation results: Open- and closed-loop trajectories, with unconstrained optimization (top), quadratic programming (QP, middle) and linear programming (LP, bottom), and different polynomial degrees ( $n = 5$  on the left resp.  $n = 3$  on the right side). Orange: reference, Black/gray: Current, Black circles: re-initialization points. Operation point  $\omega_M = 0$  rpm, 8 kHz sampling rate. Constraints:  $\tau_M \leq 11$  Nm,  $|\mathbf{u}_{s,dq}| \leq 330$  V.

### 5.3.1 Simulation results

Fig. 5.9 shows simulations to demonstrate the behavior of the polynomial-based parameterization, especially regarding the influence of the low polynomial degree, the linearization and the difference between open- and closed-loop trajectories. A torque reference step is represented at  $\omega_M = 0$ . The black line is the first trajectory, the circles are the reinitialization points and the lighter lines are the recalculated trajectories at the respective later instants.

Subfigures (a) and (b) show the step response with deactivated voltage constraints. (a) is a degree  $n = 5$  polynomial, and (b) is a degree  $n = 3$  polynomial like in the implementation. In (a) the response is very fast, but slower than deadbeat as a tradeoff of control error and the losses are included in the planning. In (b) the response is considerably slowed down because of the low degree, even though there is no constraint.

Subfigures (c) and (d) are the response with QP. The response is slower as voltage limitations are respected. Furthermore, oscillations are smaller in the predicted open-loop trajectories. Again, (d) is slower than (c) because of the low-order polynomials.

Subfigures (e) and (f) are the response with the simplification and the LP, which is also implemented in the experiments. Interestingly, there is virtually no difference in the closed-loop response compared to the QP results. There is a considerable difference in the predicted open-loop trajectories, however, especially at the end of the prediction horizon. The linearization results in a certain suboptimality on the parameters  $\alpha$ . This parameter variation  $\Delta\alpha = \alpha^{opt} - \alpha^{subopt}$  influences the polynomial by an additional error  $\Delta P(t)$ , as

$$P^{subopt}(t) = \sum \alpha_i^{subopt} t^i = \underbrace{\left( \sum \alpha_i^{opt} t^i \right)}_{P^{opt}(t)} - \underbrace{\left( \sum \Delta\alpha_i t^i \right)}_{\Delta P(t)}, \quad \forall t \in [0, T]. \quad (5.30)$$

In the polynomial, this suboptimality only has a small influence at the beginning of the trajectory as the parameters  $\Delta\alpha_i$  in the error  $\Delta P(t)$  are multiplied by  $t^i$ , at the end, however, this influence can become very large.

To conclude, the results in Fig. 5.9 indicate that the influence of the polynomial order 3 is quite considerable, but it was shown that the computational restrictions prevent higher orders. Furthermore, it was shown that the influence of the cost function linearization is quite small.



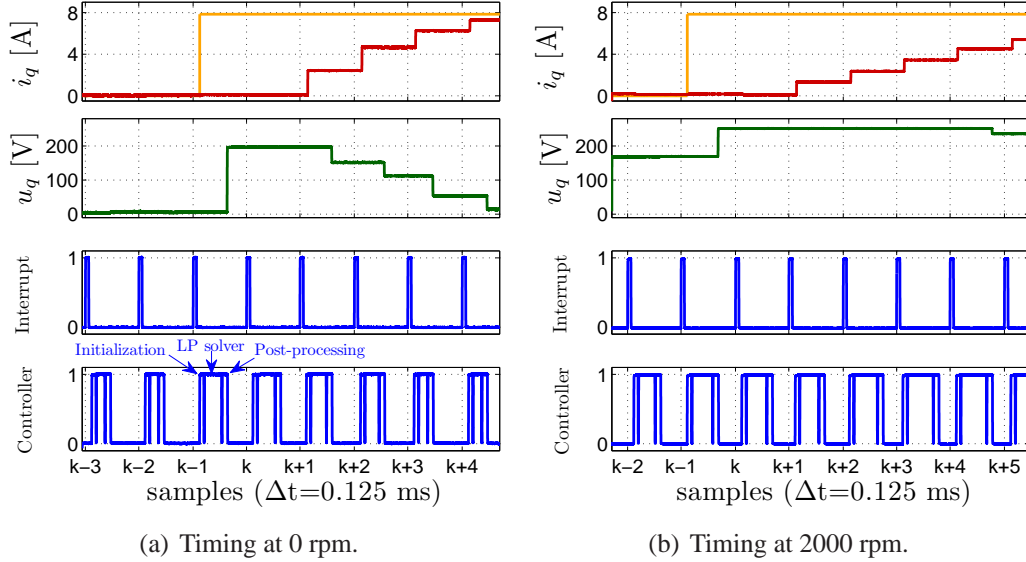


Figure 5.10: Experimental results on timing of the predictive controller.

### 5.3.2 Experimental results

From the timing sequences in Fig. 5.10, interesting insight into the computational demands of the algorithm is gained. The figures are generated in the same way as already for Fig. 2.1. The first part of the controller signal (bottom signal, blue) in (a) shows the calculation time for the simplex tableau initialization, it takes about  $10\mu\text{s}$ . Included in these calculations, which is automatically generated code resulting from symbolical calculations, is a calculation of the unconstrained optimum and the linearization of the problem. The second and biggest part of the controller signal is the runtime of the linear program (LP solver from [PTVF92]). At the beginning of (a), where voltage and current are both zero, it is only about  $20\mu\text{s}$ , but to calculate the voltage step at 2000 rpm shown in (b), more iterations are involved as many constraints are active, and the computation time rises to almost  $60\mu\text{s}$ . The total time of the interrupt handling, latency, the simplex initialization, the LP solver and the post-processing sum up to almost  $100\mu\text{s}$  in the worst case, therefore up to 80% of the available time is used.

Experimental results of the proposed scheme are shown in Fig. 5.11. Subfigure (a) shows the response to two subsequent speed reference steps, the load drive is deactivated. The cascaded PI speed controller is very fast, the speed reference step results in a voltage peak, and the torque is increased rapidly. The direct current  $i_{sd}$  depends on the speed and thereby reduces iron losses which are considerable at

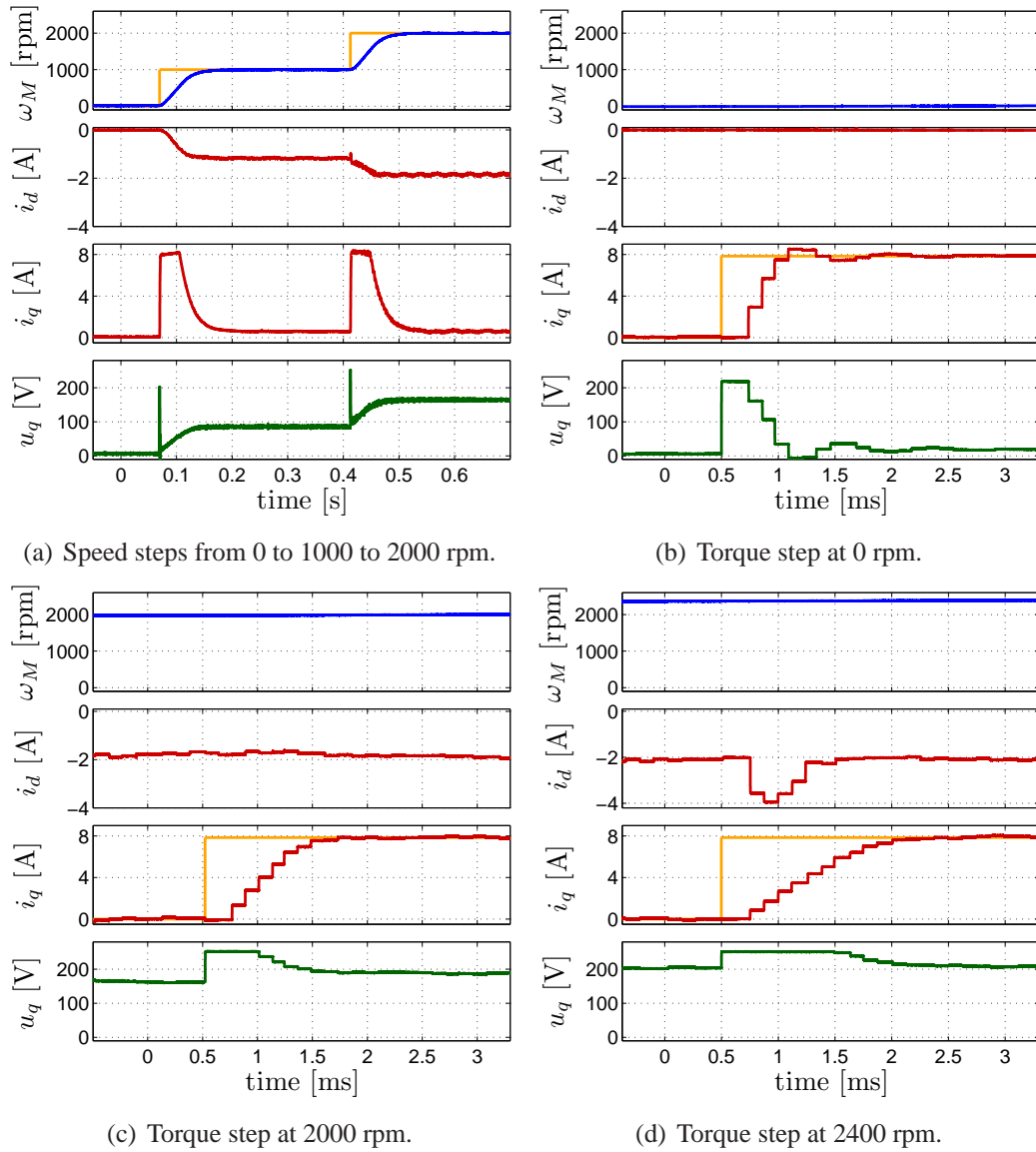


Figure 5.11: Experimental results: (a): Response of speed controller, (b)-(d): response of torque controller.

high speeds. At 2000 rpm, losses are decreased by about 2.5%, and the efficiency is improved by about 0.25%, as already indicated in Fig. 5.1. Even though these improvements are small, modeling the losses is mandatory to design a convex cost function, both currents  $i_{sd}$  and  $i_{sq}$  must be weighted. Better results are only obtained on motors with higher inductances [CTM<sup>+</sup>05].

The next three subfigures (b), (c) and (d) show fast torque transients at zero, medium and high speed, respectively. The PMSM is in torque control mode while the load drive keeps speed constant at 0, 2000 or 2400 rpm, respectively. The current components are well decoupled, a fast current change on the quadrature axis does not affect the direct axis in (b) and (c). Again, the current on the direct axis  $i_{sd}$  is dependent on the speed. Furthermore, the torque change is fast and at the same time smooth, the voltage becomes smoothly smaller for smaller control errors – a nice characteristic of quadratic cost functions with a high horizon, compared to linear cost functions which result in deadbeat behavior.

On subfigures (b) and (c), the behavior with active voltage constraint is the same as when using standard saturation or anti-windup strategies. On subfigure (d), however, a different behavior is seen, the direct current  $i_{sd}$  is reduced during the torque transient to perform field-weakening. This implies that the stator induced voltage is reduced on the quadrature axis, see eq. (5.8). Thereby the gap between induced and maximum voltage increases, the derivative of the quadrature current  $\frac{d}{dt}i_{sq}$  is higher and the torque-generation dynamics are increased, at the cost of higher ohmic losses on the direct axis. Without additional field-weakening, the reference torque would not be reached after the prediction horizon of 2 ms, thereby the end-weight of the control error in  $J$  oversized the loss term. Therefore, in this predictive control implementation, field-weakening not only improves efficiency, but also improves dynamics by exploiting the cross coupling between the orthogonal current components to optimally bypass the voltage saturation [CKS95, CS98]. The situation described above is the one where the highest number of constraints is active, and thus, also where the suboptimality of the proposed trajectory generation method is the highest, but still, behavior is very good.

It is also possible to operate the PMSM beyond rated speed with steady field-weakening to bypass the voltage saturation on the quadrature axis, as shown in [BBPZ08]. Some safety concerns have been mentioned, a PMSM operating under field-weakening beyond rated speed will generate a beyond-rated voltage if the inverter fails, however, the safety buffer is sufficiently high on standard servo drives to prevent destruction.

It should be highlighted that the current on the direct axis  $i_{sd}$  has no reference, its value is obtained from the optimization of the cost functional. Therefore, the method works well and is numerically stable; the optimal value follows inherently.

## 5.4 Conclusions

Two questions are central to this chapter:

- Is it possible to implement linear model predictive control (MPC) with online optimization, a promising advanced control algorithm?
- Does this controller give any merit that justifies the additional design and implementation effort?

A model predictive control scheme for torque control of a PMSM was introduced. The core differences to existing MPC schemes are the online solver which simplifies commissioning, the specific cost function formulation that aims at optimal efficiency and dynamics, and the high prediction horizon necessary to claim optimality. Based on suboptimal real-time optimization, the currents and voltages are computed according to a cost functional at every sampling step. The scheme is very close to the MPC formulation applied in process control, only two differences apply, the cost function is physically motivated to obtain advantages, and a continuous-parameterization-based optimization was applied. The prediction horizon is 2 ms at a sampling rate of 8 kHz, and voltage and current constraints are respected. Therefore it is very well possible to implement long-range MPC with constrained online-optimization even on fast-sampling systems such as electrical drives. According to the definition in [FM00a], the proposed scheme is the first realization of long-range online MPC for a drive, apart from the explicit solution.

The merits of the control scheme are seen in the experimental results. The advantages of the scheme are these of any predictive controller, namely precise accounting for timing of measurement and control, decoupling, respecting current and voltage constraints safely and very fast control. More specifically, long-range constrained predictive MIMO control leads to improved accounting of cross coupling, fast and smooth dynamical behavior, improved power efficiency by field weakening, and improved dynamics close to voltage saturation by additional short-time field-weakening.

As negative point, the high demands to computational power must be named, and the rather involved implementation.

## CHAPTER 6

---

### **Optimal control of induction machines using nonlinear trajectory optimization**

---

Induction machines are the most widely used type of AC drives. Their biggest advantage is the low price and the possibility to apply the very simple V/f-control scheme. These two characteristics made it the favorite drive in the industry, and it is responsible for about 60% of the total electric power consumption in the industry.

However, the power efficiency of an induction machine is lower than that of a permanent-magnet synchronous machine, especially in the small power area below 20 kW. In operation below rated load, the efficiency is decreasing even more. In many applications, the drive operates on rated torque only during limited time, but most of the time in partial load, such drives are running below their rated efficiency.

To bypass this drawback, the power efficiency can be improved in partial load by control means [KG83]. The established efficiency improvement methods are designed for steady-state operation, meaning if the drive is operating in constant speed and torque. This chapter introduces and discusses a power efficiency improvement method designed for dynamic operation, meaning torque transients.

A scheme including behavior in dynamic operation is helpful to obtain improvements in servo applications, it will be shown in the experiments that a steady-state optimization scheme can lead to higher losses than a drive without an additional optimization scheme.

## 6.1 Potential of optimal control

### 6.1.1 Preliminaries: Loss modeling in IMs

Again, according to [KG83], the machine losses are divided into uncontrollable and controllable losses. The controllable losses, manipulable by the controlled variables voltage, flux and current, are then minimized by control methods. The uncontrollable losses are basically only accessible by the induction motor design, and not further discussed.

The ohmic losses are the heating losses of the copper stator windings

$$P_{Ohm,S} = \frac{3}{2}R_s(i_{sd}^2 + i_{sq}^2), \quad (6.1)$$

and the heating losses of the aluminium rotor bars of the squirrel cage

$$P_{Ohm,R} = \frac{3}{2}R_r(i_{rd}^2 + i_{rq}^2). \quad (6.2)$$

The rotor current vector  $\dot{i}_{r,dq}$  is not represented as a state vector, but can be substituted by the expression (C.21) as function of stator current and rotor flux.

The iron losses basically consist of 4 terms, the eddy current and hysteresis losses in both the stator and rotor [LN04]. They are given by

$$P_{Eddy,S} = \frac{3}{2}k_e^{stator}(\omega_s)^2\Psi_s^2, \quad (6.3)$$

$$P_{Hyst,S} = \frac{3}{2}k_h^{stator}\omega_s\Psi_s^2, \quad (6.4)$$

$$P_{Eddy,R} = \frac{3}{2}k_e^{rotor}(\omega_s^{rotor})^2\Psi_r^2, \quad (6.5)$$

$$P_{Hyst,R} = \frac{3}{2}k_h^{rotor}\omega_s^{rotor}\Psi_r^2. \quad (6.6)$$

However, knowing that the iron losses depend on the frequency, and that for a typical design the stator excitation frequency  $\omega_s = p\omega_M + \eta L_m \frac{i_{sq}}{\Psi_{rd}}$  is much larger than the frequency of flux in the rotor, which is the slip frequency  $\omega_s^{rotor} = \eta L_m \frac{i_{sq}}{\Psi_{rd}}$ , the iron losses in the rotor can be neglected. The slip speed is generally less than about 5% of the rated speed.

Furthermore, due to the unoptimized low-cost stator design, the eddy current losses are much larger than the hysteresis losses once the excitation frequency is higher than few Hertz. Hysteresis losses are still present, but lower than for instance in PMSMs as the frequency is lower. The iron loss coefficients  $k_{es}$  and  $k_{hs}$  are

about the same order of magnitude in SI units (although they have different units), but the eddy currents rise quadratically with the frequency [LY92b]. Therefore, the hysteresis losses in the stator,  $P_{Hyst,S}$ , are also negligible.

Only the stator eddy current losses  $P_{Eddy,S}$  are of interest. As simplification, the losses caused by the leakage inductances can be neglected [LY92b] and the stator flux magnitude is approximated as  $\Psi_s = \frac{L_m}{L_r} \Psi_{rd}$ . Furthermore, the influence of slip frequency is neglected for the loss model, and the flux frequency is assumed as  $n_p \omega_M$ . The loss coefficient  $k_e^s$  is part of standard motor parameters and is modeled as 'iron loss resistance'  $R_m$ , thus  $k_e^s = \frac{1}{R_m}$ . This parameter  $R_m$  can be identified with sufficient accuracy by any upper class servo inverter. In the 'T' equivalent circuit, the iron loss resistance is parallel to the mutual inductance  $L_m$ , therefore the voltage drop is about  $\frac{3}{2} \omega_s \frac{L_m}{L_r} \Psi_{rd}$ , the losses dissipated in this resistance are  $P_{R_m} = \frac{3}{2} \omega_s^2 \frac{L_m^2}{L_r^2} \frac{\Psi_{rd}^2}{R_m}$  which fully correlate with the stator current eddy losses under the named simplifications.

As in several existing works [LY92b, LN04, QD08], the iron losses are described by the simplified model

$$P_{Iron} = \frac{3}{2} n_p^2 \omega_M^2 \frac{L_m^2}{L_r^2} \frac{\Psi_{rd}^2}{R_m}. \quad (6.7)$$

To conclude, the model of the controllable losses applied for the control-based loss minimization is

$$\begin{aligned} P_{Loss} &= P_{Ohm,S} + P_{Ohm,R} + P_{Iron} \\ &= \frac{3}{2} \left( R_s + R_r \frac{L_m^2}{L_r^2} \right) i_{sd}^2 - 3 R_r \frac{L_m}{L_r^2} \Psi_{rd} i_{sd} + \frac{3}{2} \left( R_s + R_r \frac{L_m^2}{L_r^2} \right) i_{sq}^2 \\ &\quad + \frac{3}{2} \left( \frac{R_r}{L_r^2} + n_p^2 \omega_M^2 \frac{L_m^2}{R_m L_r^2} \right) \Psi_{rd}^2. \end{aligned} \quad (6.8)$$

Inverter losses are not discussed as they only marginally influence the results [Abr00, MYKT99]. Technically it is possible to include a loss model as well. As the switching frequency is fixed, this is quite simple. However, for small drives, the converter losses are unimportant compared to the motor losses (in contrast to medium-size drives), especially as IM optimization inherits a reduced stator current magnitude.

## 6.1.2 Efficiency optimization in steady-state

In steady-state, optimization of power efficiency is quite simple, especially in the field-oriented frame. The basic principle is to optimize the currents and flux magnitude so that a minimum of losses is dissipated while still maintaining the desired

torque at the given speed. In steady-state, the relationship between the outputs and the current is

$$\tau_M = \frac{3}{2} n_p \frac{L_m}{L_r} \Psi_{rd} i_{sq}, \quad (6.9)$$

$$\Psi_{rd} = L_m i_{sd}, \quad (6.10)$$

and it is seen that the flux magnitude  $\Psi_{rd}$  is actually a degree of freedom. The flux magnitude  $\Psi_{rd}$  can be decreased and the torque-generating current  $i_{sq}$  increased while the same torque  $\tau_M$  is generated. This is illustrated in Fig. 6.1. On the left side are the power losses as function of rotor flux magnitude, and at the right the corresponding power efficiency calculation regarding only controllable losses, each curve for a specific load torque. It is seen that no improvement is possible at rated torque and speed, as the rated flux value  $\Psi_{rd} = 0.9$  Vs is about the optimal value. However, for operation at half rated torque, already, a slight improvement is possible by reducing the flux to  $\Psi_{rd} = 0.75$  Vs. For even lower torque, further reduction of the rotor flux is possible and leads to better results, for instance at 10% load torque, the efficiency can be controlled to 90% with an optimal flux compared to about 60% at rated flux.

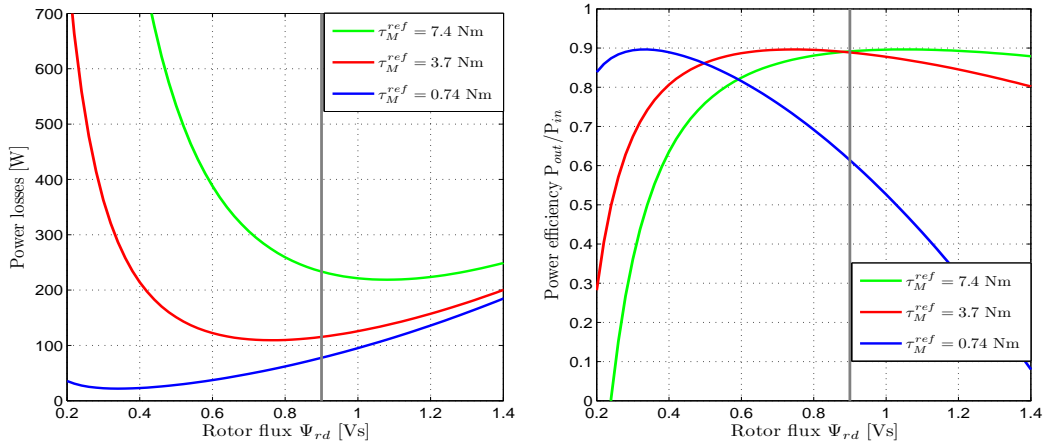


Figure 6.1: Controllable losses and efficiency of an induction machine in steady-state as function of rotor flux at rated speed for differend loads. Left: Ohmic and iron losses, Right: Power efficiency.

From the loss model (6.8), the optimal steady-state value of the flux is found by solving

$$\frac{\partial P_{loss}}{\partial \Psi_{rd}} = 0, \quad (6.11)$$



by substituting the currents with expressions (6.9) and (6.10), the optimal flux is a function of torque, speed and the machine parameters,

$$\Psi_{rd,Opt} = \sqrt{\tau_M^{ref} \frac{2 L_r}{3 n_p} \sqrt{\frac{R_s L_r^2 + R_r L_m^2}{R_s L_r^2 + L_m^4 \frac{n_p^2 \omega_M^2}{R_m}}}}. \quad (6.12)$$

To account for saturation effects, the applied value of the flux is limited to

$$\Psi_{rd}^{Sat} = \begin{cases} \Psi_{rd,N}, & \text{if } \Psi_{rd,N} \leq \Psi_{rd,Opt} \\ \Psi_{rd,Opt}, & \text{if } \Psi_{rd,Min} < \Psi_{rd,Opt} < \Psi_{rd,N} \\ \Psi_{rd,Min}, & \text{if } \Psi_{rd,Opt} \leq \Psi_{rd,Min} \end{cases}, \quad (6.13)$$

where the limits are the rated value  $\Psi_{rd,N} = 0.9$  Vs and an arbitrarily chosen minimum value  $\Psi_{rd,Min} = 0.2$  Vs. Imposing a minimum value is necessary to maintain a certain flux level, otherwise the field-orientation might be disarranged. Furthermore it prevents an excessive current demand in high torque steps. To account for voltage saturation in operation beyond rated speed, the rated value  $\Psi_{rd,N}$  is further adapted according to a flux-weakening curve.

The existing methods to implement such a flux-adaptation can be divided into three classes: model-based methods which directly apply an equation such as (6.12) [GLSW94, LN04], self-optimizing (physics-based) methods which measure the input power and minimize it by adapting the flux online [KNL85], and so-called hybrid methods which combine both methods and try to bypass the disadvantages of the respective methods. A complete coverage of existing steady-state methods, with experimental validation including converter losses, is found in [Abr00]. A major disadvantage of model-based methods is that they do not account for magnetic saturation effects, an error on the mutual inductance  $L_m$  has a direct influence on the calculated torque and flux. It is nevertheless possible to identify these parameters online. Physics-based methods, on the other hand, do not rely that much on parameters, but may have poor performance and may require experimental determination of tuning parameters.

The proposed method is a model-based method, thus also sensitive to parametric uncertainties. The results, however, will show that the error arising from parameter errors are acceptable, especially if a good rotor flux observer is applied. Furthermore, the scheme could be extended with parameter adaptation methods.

### 6.1.3 Efficiency optimization – transient operation

The previous discussion was on steady-state operation, meaning constant torque and speed. According to (6.12), the optimal flux is directly proportional to the square

root of the torque. However, it is seen from the motor model that

$$i_{sd} = \frac{1}{L_m} \Psi_{rd} + \frac{1}{\eta L_m} \frac{d}{dt} \Psi_{rd}, \quad (6.14)$$

therefore the expression (6.10) is not valid for a time-varying flux. To change the flux fast, a strong field-generating current  $i_{sd}$  is required. A fast flux variation also inherits a strong current  $i_{rd}$  in the rotor, what is seen in (C.21). In the steady-state efficiency considerations, these two effects have been ignored. Therefore, if a torque transient is controlled with a steady-state-optimized flux, a fast flux variation is imposed, and high losses appear in both stator and rotor on the  $d$  axis. Steady-state optimization is therefore inefficient during torque transients.

Furthermore, the torque response is slowed down. The generation of the torque is the product of flux and torque-generating current (6.9). If the flux is at a quite low level, the torque  $\tau_M$  is limited if the current saturation on  $i_{sq}$  is respected. Then, the torque cannot increase faster than the flux. However, it is clear that the flux is always slower than the current, and therefore an efficiency-improved induction machine has a slower response than if it is operating at rated flux.

Combining efficiency optimization and fast flux control makes not much sense for efficiency. Quite some work exists for the purpose of optimized dynamics respectively torque regarding the current and voltage limitations [BCN95]. Some publications claim simultaneous optimization of efficiency and dynamics, these works implement the flux optimization (6.12) with somewhat more performant model-based controllers [KHK92]. However, the term 'simultaneous optimization' is not correct as one has to decide for either efficiency or dynamics. As fast flux control generates losses any implementation cannot be more than a tradeoff. Both goals are in a physical contradiction.

The only combination of efficiency and dynamics that is sensible is to switch between optimal efficiency and minimum time control [MYKT99, VL03]. If a torque transient is detected, efficiency optimization according to (6.12) is deactivated, and a minimum-time controller (similar to deadbeat control) is activated. This attenuates the drawback of slow response to torque steps. However efficiency optimality is never claimed for the transients.

So the state of the art is that efficiency optimization is well given at steady state, especially if simultaneous parameter identification is performed [KHK92], but in transient, these works only consider the time response. Efficiency is disregarded in transient operation. So, induction machines used in servo drives are not yet efficiency optimized.

#### 6.1.4 Problem statement: Optimal torque control for transients

The problem addressed in this chapter is how optimal efficiency control can be obtained during a torque transient. This problem is discussed quite rarely in the literature, as in the typical optimization problem, the load torque and flux are supposed to be constant. Furthermore, induction machines in servo drives are operated with constant norm flux to bypass the current limitation problem.

The first treatment of this problem in [LY92a, LY92b] assumed full knowledge of the speed and torque trajectories of a periodic process. Offline optimization on a PC is used to calculate a time-varying rotor flux that minimizes the losses (6.8) regarding the full induction motor model. These optimized trajectories are then uploaded to a real-time controller. Foremost the experimental study shows that this method, in the case of a time-varying torque reference, obtains considerable loss improvements compared to constant norm flux operation. However, the offline optimization limits applicability, as the optimal flux trajectories are only valid for one specific predefined load and speed curve.

An analytic study of the problem of optimal efficiency during transient operation is presented in [dWR99]. Conditions of optimality are calculated for minimizing the total energy losses while satisfying torque tracking constraints. As the non-linear optimization is too difficult to solve analytically, only analytic results for the steady-state are given. The dynamic operation problem, where an optimal flux trajectory is found for a given torque transient, requires numerical algorithms. Today, however, computational power has expanded such that a numerical scheme can be applied online. The works have been recasted in the context of flatness-based control [HRD03], where the results are put into practice using trajectory tracking control.

These works are used as fundament for the underlying chapter, foremost, the analytical results from [dWR99]. An asymptotic torque transient with a desired dynamical behavior (i.e. a desired time constant) is prescribed. Based on the analytical results on optimality conditions, a simple numerical algorithm is proposed to find a good approximation of the optimal flux trajectory. The algorithm is sufficiently fast to recalculate the optimal trajectory at every sampling step, and trajectory tracking control is applied to impose the optimal behavior to the machine.

This problem is well-suited for flatness-based control. Trajectory generation and trajectory tracking are two key strengths of this control method, especially for continuous-time nonlinear systems. Furthermore, several procedures were proposed for computationally efficient trajectory generation. As the optimization has to be repeated at every sampling step, computational burden is quite a problem. For instance, the implementation of a full-fledged nonlinear model predictive control

(NMPC) scheme (a nonlinear counterpart to the scheme of the previous chapter) is out of sight for the sampling rates.

## 6.2 Real-time trajectory generation

### 6.2.1 Formulation of a real-time optimization problem

The currents – regarded as internal states rather than outputs – are replaced by the differential parameterizations of the flat outputs torque  $\tau_M$  and rotor flux  $\Psi_{rd}$ ,

$$i_{sq} = \frac{\tau_M}{\Psi_{rd}} \frac{L_r}{L_m} \frac{1}{n_p} \frac{2}{3}, \quad (6.15)$$

$$i_{sd} = \frac{1}{L_m} \Psi_{rd} + \frac{1}{\eta L_m} \frac{d}{dt} \Psi_{rd}. \quad (6.16)$$

Then the loss function (6.8) can be rewritten as

$$P_{Loss} = \frac{3}{2} \left( k_1 \Psi_{rd}^2 + k_2 \Psi_{rd} \left( \frac{d}{dt} \Psi_{rd} \right) + k_3 \left( \frac{d}{dt} \Psi_{rd} \right)^2 + k_4 \frac{\tau_M^2}{\Psi_{rd}^2} \right), \quad (6.17)$$

where

$$k_1 = \frac{R_s}{L_m^2} + \frac{n_p^2 \omega_M^2 L_m^2}{R_m L_r^2}, \quad (6.18)$$

$$k_2 = 2 \frac{R_s}{\eta L_m^2}, \quad (6.19)$$

$$k_3 = \frac{R_r}{\eta^2 L_r^2} + \frac{R_s}{\eta^2 L_m^2}, \quad (6.20)$$

$$k_4 = \frac{4}{9n_p^2} \left( \frac{R_s L_r^2}{L_m^2} + R_r \right). \quad (6.21)$$

It is further assumed that speed is constant, such that  $k_1$  can be treated as a constant. This cost function is then a function of the outputs  $\tau_M$  and  $\Psi_{rd}$  and their derivatives.

The outputs of the system are the torque and the flux, for now two free functions  $\tau_M(t)$  and  $\Psi_{rd}(t)$ . They have to be found such that the losses are minimized, the minimum-energy problem is described by the cost function

$$J = \int_0^T P_{loss}(t) dt, \quad (6.22)$$

which is constrained with the tracking objectives, i.e. the desired torque.

This general problem has too many degrees of freedom. To cast the problem into a simpler optimization problem with boundary conditions, the desired torque transient is described as

$$\tau_M(t) = m_0 + (m_1 - m_0)(1 - e^{-\lambda t}), \quad (6.23)$$

where  $m_0$  is the initial (measured) torque,  $m_1 = \tau_M^*$  is the desired final value of the torque and  $\lambda$  the eigenvalue of the transient, i.e. this parameter defines the dynamics.

Defining  $\lambda$  is left up to the commissioning engineer. However, it is not completely free but subject to some limitations. If it is set too low, the torque response might be too slow for a given application, as it is proportional to the torque bandwidth. On the other hand, if it is too high, the trajectory can possibly not be tracked if the current limitation is reached. Experience shows that a good choice is to orient with the open-loop rotor eigenvalue by about  $\lambda \approx 10..20 \cdot \eta$ , i.e. to make the torque transient 10 to 20 times faster than the open-loop flux transient.

The simple definition (6.23) is especially well suited if the resulting controller is embedded in a cascaded control structure. It is clear that field-weakening influences the gain of the torque control loop and makes the torque control loop nonlinear, cascading with a linear speed controller becomes more difficult. However, if the torque response is asymptotic, the nonlinearity is compensated and the closed-loop torque control system acts like a linear system. Another advantage is that, in the case of constant reference  $\tau_M^*$ , the open- and closed-loop trajectories are the same. This is very important to preserve the optimality from the open-loop calculation in the closed-loop behavior, as already discussed in section 3.1.4.

Now that the torque trajectory is defined, two things are known of the flux trajectory  $\Psi_{rd}(t)$ . First, obviously the initial condition  $\Psi_{rd}(0)$ , which is the observed flux magnitude. Second, as the torque transient was defined with the assumption that the system will go to steady-state, the final value of  $\Psi_{rd}(t \rightarrow \infty)$  can be defined by the steady-state optimum  $\Psi_{rd,Opt}$ , as calculated in (6.13). Other initial conditions, such as the initial current  $i_{sd}$ , are not regarded, they would overconstrain the problem. Actually, the induction machine is treated as it is fed by a current source inverter [dWR99], the currents  $i_{s,dq}$  are the control inputs, they will be impressed on a cascaded control structure if a voltage-source-inverter is applied.

With knowledge of the torque trajectory  $\tau_M(t)$  and the initial and final value of the rotor flux, the general problem is transformed and simplified into an optimization problem with one free function  $\Psi_{rd}(t)$  and two-point boundary conditions  $\Psi_{rd}(0)$  and  $\Psi_{rd}(T)$ .

## 6.2.2 Calculus of variations

The fact that the cost function and the involved functions  $\tau_m(t)$  and  $\Psi_{rd}(t)$  are continuous allow to use a famous result from calculus of variations [Pie69], the Euler-Lagrange-equation. This equation gives the necessary condition on optimality of the flux trajectory  $\Psi_{rd}(t)$  regarding the afore defined optimization problem. The flux trajectory has to satisfy the condition

$$\frac{\partial}{\partial \Psi_{rd}} P_{Loss} - \frac{d}{dt} \left( \frac{\partial}{\partial (\frac{d}{dt} \Psi_{rd})} P_{Loss} \right) = 0, \quad (6.24)$$

which, with the intermediate calculation steps

$$\frac{\partial}{\partial \Psi_{rd}} P_{Loss} = \frac{2}{3} \left( 2k_1 \Psi_{rd} + k_2 \frac{d}{dt} \Psi_{rd} - k_4 \frac{2\tau_M^2}{\Psi_{rd}^3} \right), \quad (6.25)$$

$$\frac{\partial}{\partial (\frac{d}{dt} \Psi_{rd})} P_{Loss} = \frac{2}{3} \left( k_2 \Psi_{rd} + 2k_3 \frac{d}{dt} \Psi_{rd} \right), \quad (6.26)$$

reads as

$$k_1 \Psi_{rd}^4(t) - k_3 \Psi_{rd}^3(t) \left( \frac{d^2}{dt^2} \Psi_{rd}(t) \right) = k_4 \tau_M^2(t). \quad (6.27)$$

The condition is constrained to the torque trajectory  $\tau_M(t)$  in (6.23) as well as to the initial and final value of the flux trajectory  $\Psi_{rd}(0)$  and  $\Psi_{rd}(T)$ . The 'horizon'  $T$  is assumed sufficiently high to have quasi steady-state at  $t = T$ . This is a two-point boundary value problem (BVP).

In differentially flat systems, calculus of variations leads to a directly usable condition [SRA04]. More general control systems may require more advanced theoretical results, such as Pontryagin's maximum principle [Pie69], but flat systems are simpler, as the differential equations can be substituted using the parameterization of the output, what has been done with (6.15) and (6.16).

The condition (6.27) is a necessary but not sufficient condition. To be sufficient, the optimization problem must be convex, this is the case as  $\frac{\partial^2}{\partial (\frac{d}{dt} \Psi_{rd})^2} P_{Loss} > 0$ , as shown in [dWR99]. So, to find the unique and optimal solution, solving (6.27) regarding the constraints is sufficient.

The condition of optimality for the flux is a second-order nonlinear time-varying boundary value problem (BVP). Solving a BVP is different than direct optimization of a cost function, for instance, as in the previous chapter, by parameterization of the trajectories and subsequently solving the parametric optimization problem.

Especially for nonlinear continuous-time systems, the BVP can represent a helpful simplification.

BVPs are typically solved numerically. A key technique is the shooting method. The problem is transformed into an initial value problem (IVP), so the initial conditions  $\Psi_{rd}(0)$  and  $\frac{d}{dt}\Psi_{rd}(0)$  are given. Then, the differential equation can be solved, for instance by numerical integration. In a loop, the unknown value for  $\frac{d}{dt}\Psi_{rd}(0)$  is adapted iteratively until the final value  $\Psi_{rd}(T)$  corresponds to the desired final value. This technique is, however, not suitable for this application, foremost because of the high computational burden coming from successive numerical integration. Instead, an approximate analytical solution is searched.

### 6.2.3 Approximate solution using prototype fitting

In a numerical study, whose results are displayed later (in Fig. 6.2 resp. in appendix F.4), it turns out that the numerical, accurate, solution to the BVP has a strong similarity to an exponential function. If  $\Psi_{rd}(t)$  is an exponential function, then the second derivative  $\frac{d^2}{dt^2}\Psi_{rd}(t)$  is also exponential and the Euler-Lagrange equation (6.27) could become satisfied. Therefore, the idea is close to define an analytic prototype of the flux trajectory as

$$\Psi_{rd}(t) = f_0 + (f_1 - f_0)(1 - e^{-\mu t}), \quad (6.28)$$

where the initial and final conditions are met by imposing  $f_0 = \Psi_{rd}(0)$  and  $f_1 = \Psi_{rd}(T) = \Psi_{rd}^{Opt}$ , and the eigenvalue  $\mu$  is a free parameter to be determined such that the condition of optimality (6.27) is approximately satisfied.

The Euler-Lagrange equation (6.24) should – for an optimal flux – be satisfied at every instant, however, as the prototype (6.28) is not perfectly suited, there will be remaining terms. The remainder of (6.27), defined by

$$e(\mu, t) = k_1\Psi_{rd}^4 - k_3\Psi_{rd}^3\left(\frac{d^2}{dt^2}\Psi_{rd}\right) - k_4\tau_M^2, \quad (6.29)$$

is explicitly calculated in appendix F.4.

Determination of the optimal flux eigenvalue  $\mu$  is done in a very simple way. A time instant  $t_s$  is determined at which  $\mu$  is fixed such that the remainder of (6.27) is zero at this time, thus  $e(\mu, t_s) = 0$ . So, first a suitable  $t_s$  is searched. Based on the developments in appendix F.4, where it is known that for a constant torque the optimal flux eigenvalue is in the area  $\sqrt{\frac{k_1}{k_3}} < \mu < 2\sqrt{\frac{k_1}{k_3}}$ . This is far too rough as final solution as the equation for a nonconstant  $\tau_M$  is much different, however, it is

suitable to determine the instant  $t_s$ . Therefore, define

$$t_s = \sqrt{\frac{k_3}{k_1}}. \quad (6.30)$$

Then  $e(\mu, t_s)$  is nothing but a univariate function of the undetermined flux time constant  $\mu$ . In a numerical search, then, determine  $\mu$  such that the remainder of (6.27) becomes

$$e(\mu, t_s) = 0. \quad (6.31)$$

This gives a good approximation to the optimal flux trajectory. To solve  $e(\mu, t_s) = 0$ , Newton-Raphson search [Pie69,PTVF92] is applied, starting from the open-loop rotor eigenvalue  $\mu^{k=0} = \eta$ , the iterative law

$$\mu^{k+1} = \mu^k - \frac{e(\mu^k, t_s)}{\frac{\partial}{\partial \mu} e(\mu^k, t_s)} \quad (6.32)$$

leads to a good approximation already after few iteration steps (i.e.  $k = 10$ ). To improve reliability, the search area is limited to

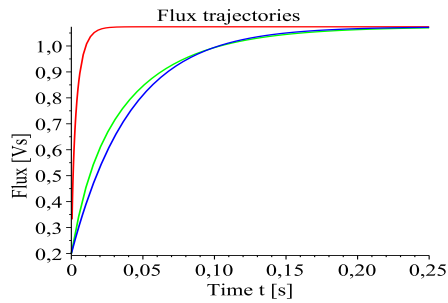
$$\eta < \mu < \lambda, \quad (6.33)$$

which is reasonable; the optimal flux transient will not be faster than the torque transient and neither slower than the open-loop rotor time constant. Furthermore, to prevent nonconvergent cycles, the step size  $|\mu^{k+1} - \mu^k|$  is limited to  $\frac{\lambda}{10}$ . In a numerical study, it turned out that this simple search method is both accurate and reliable.

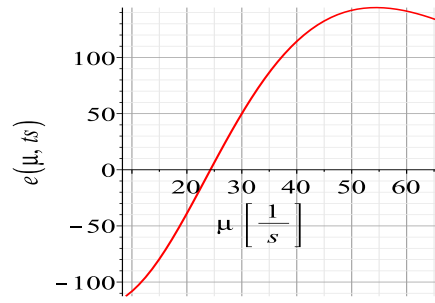
In Fig. 6.2 this procedure is evaluated for three case scenarios. They are all performed at rated speed, the time constant of the torque function is 5 ms (or  $\lambda = 200\frac{1}{s}$ ).

Subfigure (a) presents the resulting flux trajectories of a torque step from 0 to rated torque. The flux starts at the lower limit  $\Psi_{rd}(0) = 0.2$  Vs and rises to its steady-state optimum. The red trajectory is that calculated with the steady-state optimality condition (6.12), it is very fast and therefore requires a strong field-generating current  $i_{sd}$ , generating high losses. The green trajectory is the optimal flux trajectory resulting from the dynamic optimization problem, it is the exact solution of the BVP (6.24). It is much slower. It is the optimal-efficiency transient, it considers all losses generated by all currents, and the stator eddy current losses. The blue trajectory is the approximate solution to the BVP, also applied for the experiments. It is close to the green trajectory, however, at the beginning it is too slow

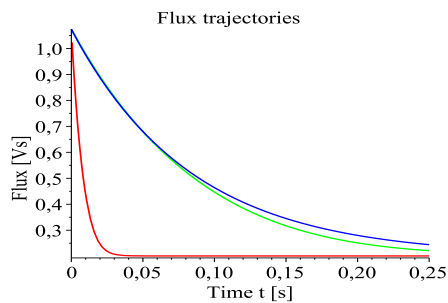




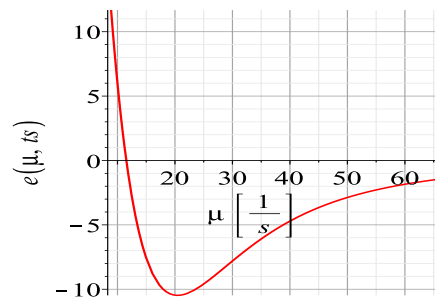
(a) Zero to max torque at nominal speed: Flux trajectories.



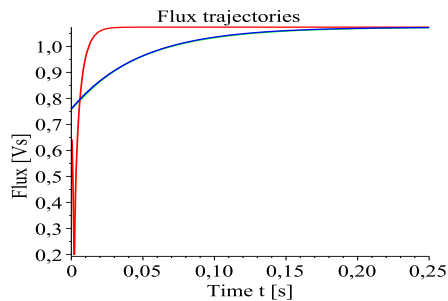
(b) Zero to max torque at nominal speed: Error function  $e(\mu, t_s)$ .



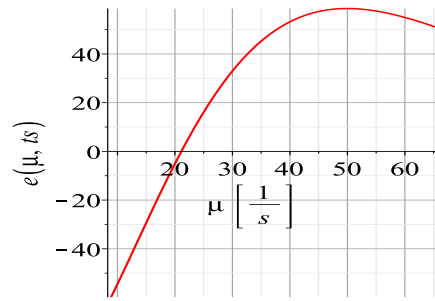
(c) Max to zero torque at nominal speed: Flux trajectories.



(d) Max to zero torque at nominal speed: Error function  $e(\mu, t_s)$ .



(e) Negative half to max torque at nominal speed: Flux trajectories.



(f) Negative half to max torque at nominal speed: Error function  $e(\mu, t_s)$ .

Figure 6.2: Evaluation of the optimization problem. Left: calculated flux trajectories, red: steady-state calculation, green: optimal dynamic calculation, blue: approximated dynamic calculation. Right: corresponding error function  $e(\mu, t_s)$  that has to be solved numerically.

and at the end too fast, a better prototype would therefore be  $e^{-\mu t^\alpha}$  with  $0 < \alpha < 1$ , however, then, a more complicated numerical search must be performed. It is seen

that the time constant is not the solution of the autonomous differential equation  $\sqrt{\frac{k_3}{k_1}} = 0.103$  s. Also at the 'strategic' instant  $t_s = 0.052$  s the flux is not exact, only the differential equation is zero there. Subfigure (b) shows the nonlinear function  $e(\mu, t_s)$  that is used to find the eigenvalue of the flux trajectory  $\mu$ . This function is well suited for a Newton-Raphson search, as it is very smooth, monotonic and somewhat linear at the beginning.

Subfigure (c) shows the resulting flux trajectories of a torque step from rated to almost zero torque. Again, the steady-state optimality condition gives an extremely fast trajectory that requires a strong negative field-generating current to reduce the flux. The optimal trajectory is much slower, the time constant is not too different to the open-loop rotor time constant  $\frac{L_r}{R_r} = 0.11$  s, so here almost no field-generating current is applied. The approximated solution again fits the optimal value quite well. Subfigure (d) shows the nonlinear function  $e(\mu, t_s)$ , which is again no problem for a numerical search.

Subfigure (e) shows the resulting flux trajectories of a torque step from negative half to positive rated torque. The steady-state optimality condition tries to reduce the flux to the minimum and then reinduce it, however, this is not very sensible. The optimal trajectory is much slower and monotonic. The approximated trajectory is almost identical, so the approximation works well. Subfigure (f) again demonstrates that the numerical search is no problem.

There is one limitation, however, caused by the simple choice of prototype function. If the situation from subfigure (e) is repeated with a slower torque transient, say a time constant of 20 ms, the optimal flux is first lower and increased later than the approximation calculates, as shown in Fig. 6.3. So, at very slow sign changes of the torque, the simple prototype is suboptimal, nevertheless, it is still better than the result of the steady-state optimality condition.

#### 6.2.4 Control structure: Trajectory generation and trajectory tracking

Putting optimal control, as from optimization theory [Pie69], into practice is not an easy task. The fact that open-loop trajectories are calculated and that the closed-loop trajectories may be different is a key problem. Flatness-based control, however, offers a simple solution with nonlinear trajectory tracking control [vNM98, HRD03].

The trajectory generation is an algebraic scheme with only few numerical calculations that are performed online. Thus it is possible to perform fine-tuning during commissioning or to include parameter adaptation schemes, for instance of the rotor time constant [QD08].

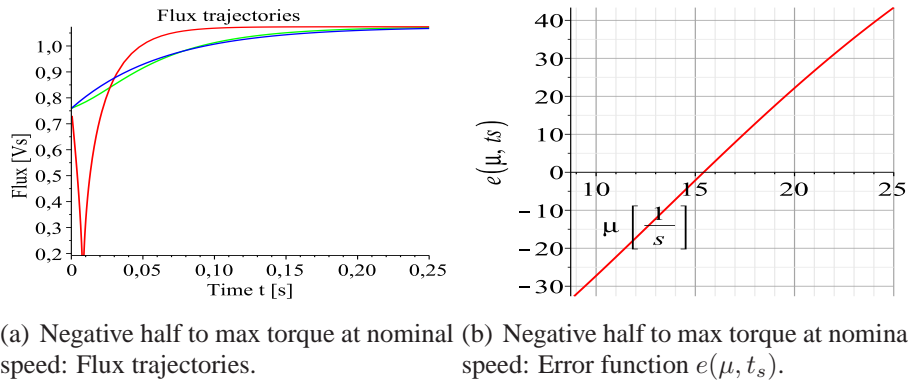


Figure 6.3: Evaluation of a case where the prototype is not suited, i.e. a very slow torque transient  $\lambda = 50\frac{1}{s}$ . Left: calculated flux trajectories, red: steady-state calculation, green: optimal dynamic calculation, blue: approximated dynamic calculation. Right: corresponding error function  $e(\mu, t_s)$  that has to be solved numerically.

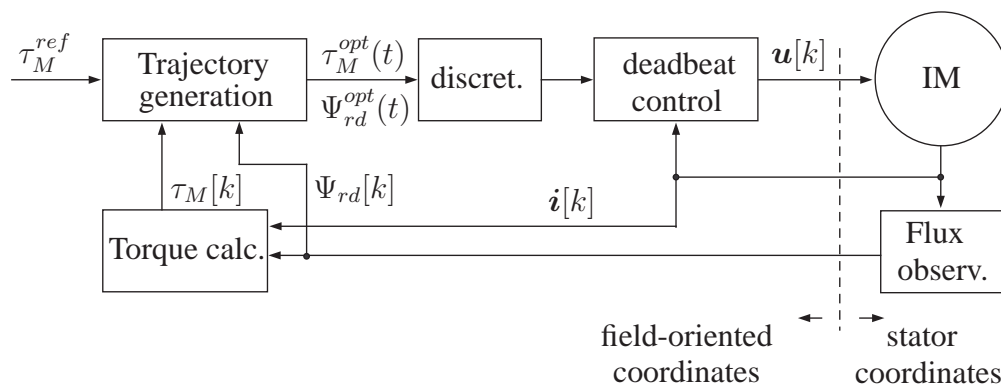


Figure 6.4: Flatness-based optimal torque control: Real-time trajectory generation and trajectory tracking control.

The applied control structure is shown in Fig. 6.4. In the field-oriented frame, the actual values of the flux magnitude and the torque can be easily calculated based on the observed flux and measured currents. These serve as initial conditions to define  $m_0$  and  $f_0$  in the trajectory generation algorithm. The desired flux transient and the corresponding (near-)optimal flux trajectory are calculated, and discretized with knowledge of the sampling time  $T_s$ . The future values of the torque and flux are then fed, as references, to a tracking controller. The deadbeat controller proposed in chapter 3 is chosen as it provides very precise tracking (except for a fixed

delay between reference and output) and good suppression of disturbance effects. However, any fast flux or current controller may be used, as long as it can adapt to the problems arising from time-varying rotor flux and the inherited parametric uncertainties.

The scheme somewhat reminds the MPC controller in Fig. 5.8, however, this is not model predictive control as constraints are not explicitly included in the controller design and only a part of the model is used in the optimization. Still, it is a predictive controller. Furthermore, it is an optimal controller, as the prediction horizon is quasi infinite. The open-loop trajectories, calculated by the trajectory generator, and the closed-loop trajectories, which result from recalculating the trajectories at every sampling step, are quasi identical. Only differences are the simplifications of the design model and the real machine model, meaning the iron resistance, saturation, harmonic effects, etc.

Stator current limitations were not included in the trajectory generation. Technically, this is possible, for instance by a coordinate transformation as proposed in [GP09]. However, the inclusion of current constraints, no matter with which technique, makes the nonlinear optimization problem even more complicated. Furthermore, in the case of an active current constraint, one has to decide for either minimum-time behavior (or best torque tracking), or to slow down the desired torque trajectory to minimize losses. Here, current limitation is implemented by saturating the references for the currents. For simplicity, a quadratic saturation is chosen of the type  $-I_{max} \leq \mathbf{i}_{s,dq} \leq I_{max}$ .

About 10% of the computational resources are used at 8 kHz sampling rate, therefore a 200 MHz DSP should provide more than sufficient computational power to implement the scheme.

### 6.3 Experimental results

For the results, the proposed dynamic optimization scheme is compared to the regular operation with rated flux  $\Psi_{rd} = 0.9$  Vs and the steady-state optimal control where the flux is directly depending on the torque [dWR99].

It is applied to a low-efficiency induction machine whose parameters are given in appendix B.1, already applied in section 4.2.3. This machine features a strong magnetic saturation, which is deteriorating the results as the model assumes linear flux.

The first analysis is in standstill in order to use a torque transducer to compare the calculated and the real torque  $\tau_M$ . A second analysis covers a servo drive application scenario.

### 6.3.1 Torque response

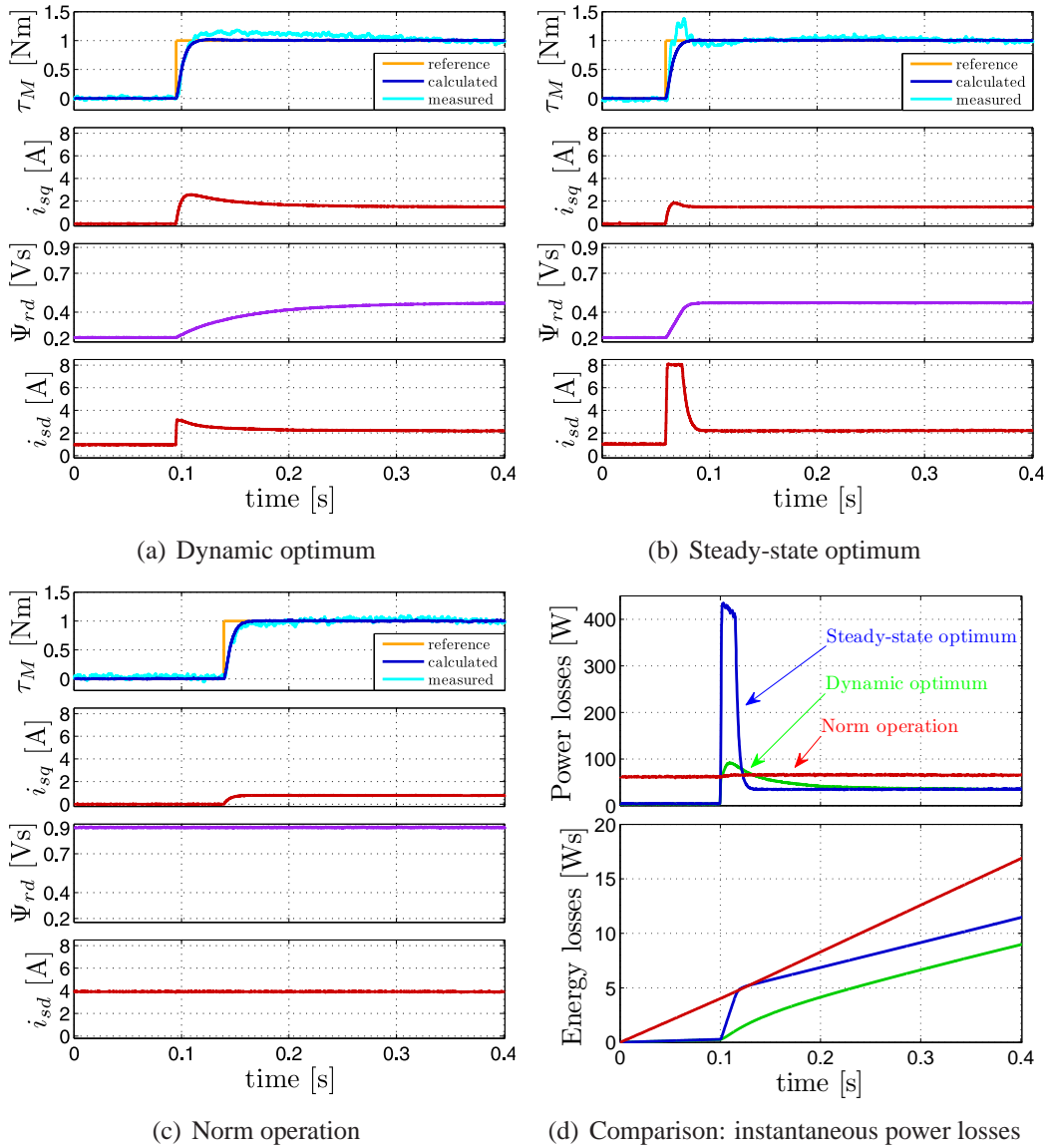


Figure 6.5: Optimal efficiency control: Torque step response at zero speed, small steps  $\tau_M = 0.1$  Nm. Comparison between dynamic, steady-state and no optimization.

In Fig. 6.5, the small-signal response from zero to 1 Nm (13.5% of the rated torque) is analyzed. Subfigure (a) shows the results of the proposed method. While

the calculated torque (dark blue) exactly corresponds to the desired asymptotic trajectory with a time constant of 5 ms, the measured torque (cyan) is slightly larger. This error is caused by a misestimation from the rotor flux observer because of the magnetic saturation, however, in steady-state the flux is estimated correctly. As discussed in section 2.6, the inductance value  $L_m$  is designed such that it is the sequential inductance at rated flux, therefore the parameter applied for the observer and the torque calculation will be smaller than the real parameter. The flux is controlled slowly from the lower saturation limit  $\Psi_{rd} = 0.2$  Vs to its optimal value for  $\omega_M = 0$  rpm and  $\tau_M = 1$  Nm, which is  $\Psi_{rd} = 0.47$  Vs. Only a small field-generating current is necessary to perform the flux increase.

Subfigure (b) shows the response if the optimization is performed according to the steady-state criterion. The flux is supposed to be adjusted proportionally to the torque, thus with a time constant of 5 ms [dWR99]. Because of the current limitation, however, it is slower, and the field-generating current  $i_{sd}$  is at the saturation limit. The following saturation law is active: if the field-generating current is at the maximum, the torque-generating current is increased such that the torque is still tracked. Again, the measured torque is not exactly the desired transient, foremost because of the high  $i_{sd}$  and the magnetic saturation, sensitivity is higher for high field-generating currents.

Subfigure (c) shows the response with norm operation at the rated flux level  $\Psi_{rd} = 0.9$  Vs. There is no problem of current saturation during the transient, and as the flux is constant and well-estimated, the calculated torque corresponds well to the measured torque.

Subfigure (d) compares the three measurements in terms of instantaneous power loss and energy loss. In norm operation (red), the losses are high in steady-state but quasi unchanged during the transient. The power loss is constant and overall, much energy is lost (17 J). In operation with the steady-state criterion (blue), the losses are much lower during constant torque. However, the losses become extremely high during the transient, caused by the high field-generating current in the stator and the rotor. While energy is saved in steady-state, a lot is lost during the transient, resulting in a loss of 12 J. In operation with the dynamic criterion (green), losses are low during both the steady-state and the transient. As the flux transient is performed slower, the losses have a lower peak value but are of longer duration. Overall the scheme is the best and results in only 9 J energy loss (obviously depending on the considered period).

In Fig. 6.6, the step response from zero to 3.7 Nm (50% of the rated torque) is analyzed. The optimal flux at this desired torque is at the rated level  $\Psi_{rd,Opt} = 0.9$  Vs as the experiments are performed at zero speed, therefore, a flux change from the lower to the upper saturation limit is required.

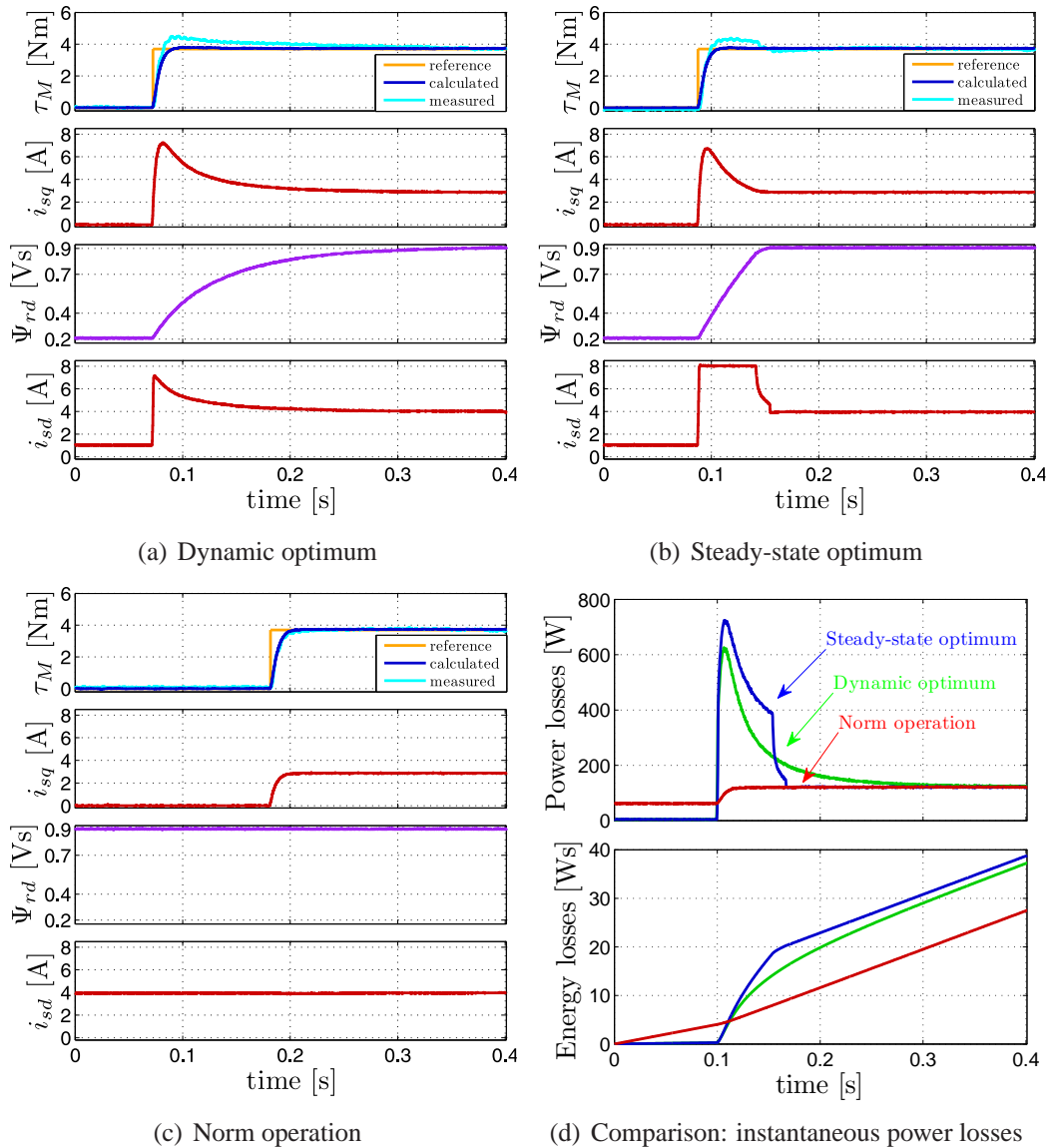


Figure 6.6: Optimal efficiency control: Torque step response at zero speed, large steps  $\tau_M = 0.50\%$ . Comparison between dynamic, steady-state and no optimization.

In subfigures (a) and (b), again, the measured (cyan) and calculated (dark blue) torque differ during flux variation because of magnetic saturation. This effect does not appear at norm operation in (c). The required current is very high, and in (b), the field-generating current is in saturation for a long time. Nevertheless, the de-

sired asymptotic torque trajectory is well tracked. In subfigure (d) the instantaneous power and energy losses are shown. This time the steady-state criterion is closer to the dynamic criterion, as the current saturation slows down the flux transient. The norm operation (red) is better than both optimizing schemes, clearly, the flux change inherits a considerable energy loss. The boundary condition of steady-state optimality in these two schemes enforces the flux variation. Still, the dynamic optimization criterion is better than the steady-state criterion, it will always be better. The norm operation results in an energy loss of 27.5 J, the steady-state criterion in 39 J and the dynamic criterion in 37 J. The savings potential is much better at high speeds, however.

### 6.3.2 Efficiency evaluation: A servo application

Whether efficiency can be improved by flux adaptation, and how much the dynamic optimization is an improvement compared to the stationary optimization is a matter of the application. In [LY92a], a fixed-distance fixed-time point-to-point positioning process is proposed to evaluate the efficiency of a dynamic optimization procedure. It gives a good trade-off between the improvements of field-weakening in steady-state and the losses inherited by dynamically adapting the flux. It is a typical servo application. The setup is shown in Fig. 6.7, it is an induction machine with a torque transducer and a flywheel.



Figure 6.7: Setup for closed-cycle efficiency evaluation: Induction machine with torque transducer and flywheel.

It is a closed-cycle process, as shown in Fig. 6.8, kinetic energy will be charged during acceleration and electrical energy recovered during deceleration. From the electric power  $P_{In}$ , in the induction machine, the controllable and uncontrollable losses  $P_{Loss}$  and  $P_{Loss}^{unctr}$  are detracted and the rotor absorbs some kinetic energy via  $P_{Kin}^{Rotor}$ , the remaining power  $P_{Out}$  is the output power measured with the torque transducer. This output power is partially lost due to friction  $P_{loss}^{wheel}$ , the rest  $P_{Wheel}^{kin}$  is transformed into kinetic energy of the flywheel.



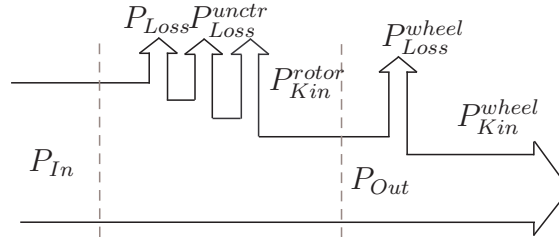
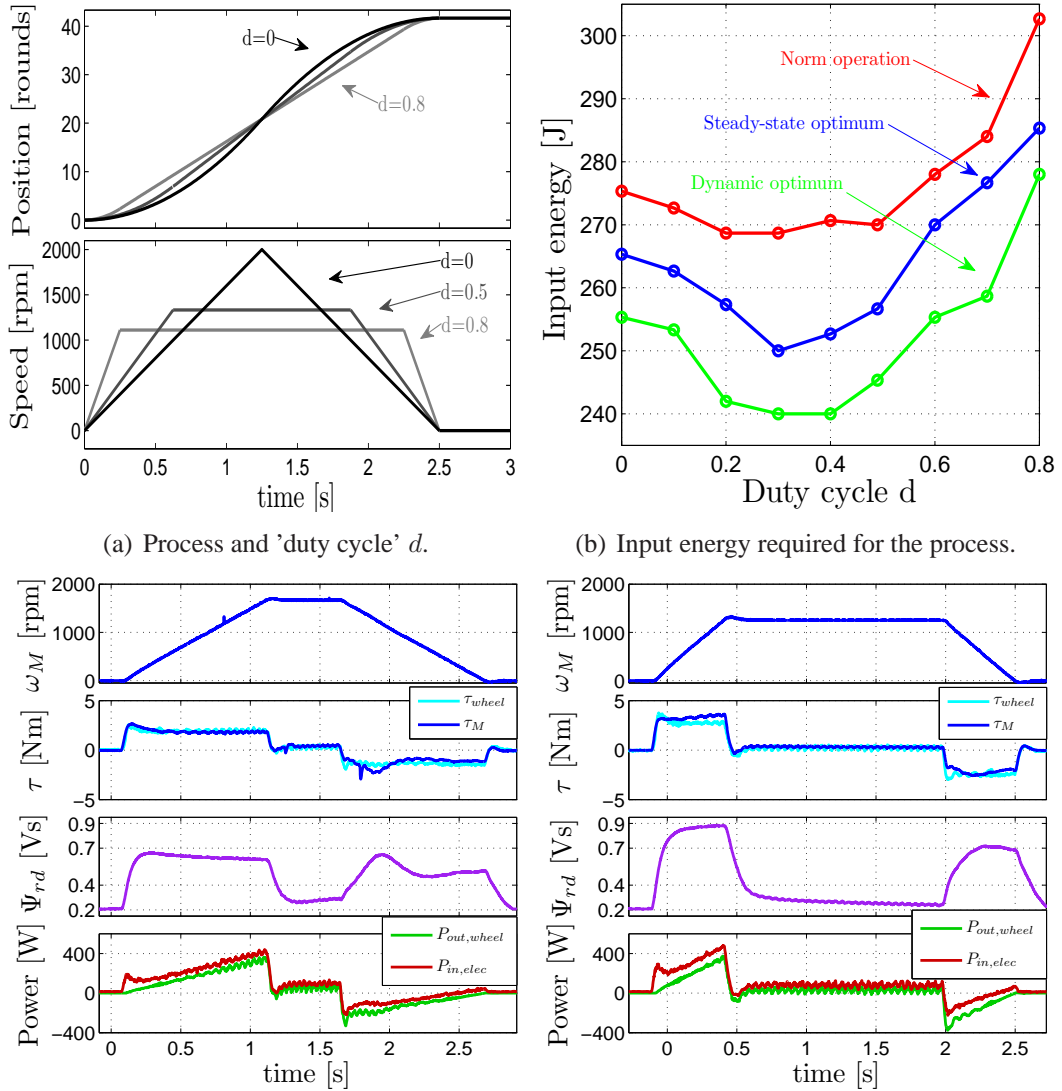


Figure 6.8: Power flow: Measured electrical input power  $P_{in}$ , measured output power  $P_{out}$ .

The process, shown in Fig. 6.9 (a), is an s-curve from zero to 41.6 rounds that is performed within 2.5 s. The speed trajectories, as degree of freedom, are sized with the 'duty cycle'  $d$ . A high  $d$  inherits a high torque at the beginning and at the end, but almost zero in the middle, whereas a low duty cycle comes with a continuous but low torque. The settling time is chosen such that for  $d = 0.8$ , rated torque is required to track the speed reference. The odd traveling distance of 41.6 rounds stems from the chosen settling time, which again depends on the flywheel, and the fact that 2000 rpm is the maximum speed.

The input power is calculated from the voltage commands and the current measurements. The results, shown in subfigure (b), show that it makes sense to reduce the flux magnitude in such a process. Only the input energy is shown, the average output energy is about 120 J which is the friction of the flywheel bearing, it is almost the same for any  $d$ . The input energy is spent to a good portion to the uncontrollable losses, mainly the friction of the induction motor bearings and the air cooling fan. The total energy absorbed by and recovered from the flywheel is 280 J at  $d = 1$  and 70 J at  $d = 0.8$ , as the inertia is  $\Theta = 0.044 \text{ kgm}^2$ . Considering this, the controllable losses shrink considerably while the drive is doing the same job. Steady-state flux adaptation is an improvement versus norm operation as most of the time, the drive is operating below half rated torque, full torque may only be required during the speed variation. Dynamic optimization is a further improvement as the four torque changes and the time-varying speed, inheriting flux variation, are also considered.

Figs 6.9 (c) and (d) show the time response for the duty cycles  $d = 0.2$  and  $d = 0.6$  with the proposed dynamic optimization scheme. The electrical input power (red) peaks during the torque transient. Energy is required to accelerate the rotor and the flywheel, the mechanical power required for the latter is shown in green. In the middle, only a low torque is required to keep speed constant, and the flux can be reduced. There is a certain torque ripple due to a small imbalance of the



(c) Dynamic optimization: response for  $d = 0.2$ . (d) Dynamic optimization: response for  $d = 0.6$ .

Figure 6.9: Fixed-distance fixed-time point-to-point positioning process: 41.6 rounds in 2.5 s.

flywheel. During deceleration, then, the energy is recovered from the flywheel, it can be fed back to the inverter. Losses are high during the flux variation phases and if the flux is kept at a high level.

## 6.4 Conclusions

This chapter discusses the controllable losses appearing in induction motors. A model-based method for loss minimization was proposed that considers time-varying torque. For an asymptotic torque response with an arbitrary defined time constant, the losses are minimized. So far, only a numerical scheme exists that optimizes time-varying torque and flux trajectories offline, the speed and load trajectories must already be known in the design phase. Here, an online trajectory generation scheme is proposed that generates optimal flux and current trajectories to meet demands appearing on-line.

The optimization problem is a hard-to-tackle nonlinear optimization problem. However, using the simplifications from flatness-based optimal control, and an efficient approximation with a function prototype, a simple solution is found, only few iterations of a Newton-Raphson search procedure and some algebraic equations must be evaluated online. Even though the optimal solution has been heuristically approximated, the solution is apparently fine. The optimization procedure is embedded in a flatness-based trajectory generation and tracking control scheme.

The merits of the dynamic optimization were pointed out by experimental results. Losses are higher during torque transients if the flux is adapted, and in contrast to the existing schemes optimizing only the steady-state values, these losses are also reduced by the proposed method. However, the improvement is limited by the current saturation, such that it is only visible for small torque steps. A point-to-point positioning process was chosen to demonstrate the improvements in a servo drive application. The process covers a typical servo application as it includes torque steps, speed variation and both motor and generator mode. Here, the advantages of flux adaptation versus rated flux operation, and the improvements of considering the dynamics, have become clear.

The method is suitable for energy- and cost-efficient servo drives with induction machines.



## CHAPTER 7

---

### Conclusions and outlook

---

Differential flatness has been used to design advanced control systems for electrical drives. Based on existing basic algorithms, such as space vector theory, field-oriented control, flux observers, etc., that are extensively covered in the literature, some new advanced algorithms have been proposed.

A drive operates to some satisfaction using the named basic algorithms, together with standard control systems such as PI control. However, by applying advanced algorithms, the performance and efficiency of a drive can be significantly improved. Using high-performance predictive control, the tracking and disturbance rejection capability is improved, the resulting current and torque is quickly adjusted to an operator's reference and current ripples are suppressed. Furthermore, by using optimal control, efficiency and dynamic response may be further improved. A drive system can be exploited to its physical limits, while still operating safely and robustly.

Chapters 3 and 4 studied deadbeat control, one of the most widespread predictive current control methods. The sensitivity of this scheme to noise and parametric uncertainties is well-known, especially the sensitivity to the inductance value. Furthermore such a scheme is supposed to compensate undesired secondary effects, such as torque ripples caused by non-ideal sinusoidal windings, without explicitly identifying them. Using results from flatness-based control, the concept of feedforward linearization considerably reduces parametric sensitivity and decouples the disturbance estimator from the controller. A mix between the standard method and feedforward linearization is proposed to trade off robustness increase and perfor-

mance reduction. From an implementation point of view, the method is just a minor modification to a standard deadbeat controller, but with a big impact on the performance. As a result, the robustness to resistance and especially to inductance uncertainties is improved to a satisfactory level, even highly saturating drives can be controlled. The problem of inductance uncertainties appears on SynRMs and IMs and is so strong that it prevents the application of standard deadbeat without adjusting the parameters. Furthermore, the disturbance estimator is made faster such that torque ripples can be well suppressed, this problem appears on IMs and PMSMs. Without the decoupling inherited from feedforward linearization, this is not possible because of noise sensitivity and the destabilizing interaction between controller and estimator. Because of the good robustness and performance properties which have been pointed out both analytically and experimentally, the proposed deadbeat controller is suitable for an industrial application.

A future work is the application of online-identification of the machine parameters. Some works on this issue exist, based on either extended hardware or heuristic methods. However, more general methods, for instance based on time-scale decoupling, would be of interest.

Another possible future work is a comparative evaluation of high-performance field-oriented control to direct torque control (DTC) and predictive torque control (PTC). These control systems come up with a similar performance compared to field-oriented deadbeat control, however, the parametric sensitivity problem is less strong, or at least, less discussed. The proposed method is applicable to these schemes as well, some preliminary experimental results are given in appendix E. An analytical investigation is still missing, however, the method of 'equivalent control' has been proposed for variable structure systems as powerful analysis tool for switching control systems, which could serve as an analytical tool to study these schemes.

Chapter 5 studied model predictive control (MPC) for a PMSM. This optimal torque controller design aims at improving the power efficiency and the dynamics. It is a widely known control method in some industrial application areas, however, rather new for electrical drives. This method comes with a quite involved implementation effort, foremost because an online optimization algorithm is required. There are two problems of this scheme, the first is whether it is possible to implement because of the computational requirements. Second, the results must show merits in order to justify the increased implementation effort. Existing schemes either did not respect the voltage constraints (GPC), were of a short horizon (1 step, deadbeat or PTC) or applied offline optimization (explicit MPC with a look-up-table). The proposed implementation is the first running constrained long-range scheme with online optimization. The computational burden has been reduced to a certain

level by using flatness and some approximations in the optimization problem, leading to a simple to solve linear programming problem. The results show a fast but at the same time smooth torque response, steady-state efficiency optimization by field-weakening, and in high-speed operation, improved torque dynamics because of dynamical field-weakening to bypass the voltage saturation. Such results can only be obtained by including constraints and a high horizon. Especially the dynamic use of field-weakening, known from minimum-time current control, is newly seen on predictive controllers.

However, the computational demands are still quite high, such that the contribution should be seen as academic benchmark rather than industry-ready scheme. While it should be clear now that even more merits of MPC are seen with constrained long-horizon optimization, a future work is the further reduction of computational burden, for instance by implementing heuristic methods.

Chapter 6 proposed a trajectory optimization scheme for IMs. The torque controller aims at best energy efficiency for torque transients with a predefined asymptotic shape. This problem is rarely discussed, servo induction drives are operated in rated operation, therefore they do not profit from the potential efficiency improvement in steady state. Efficiency-improved drives, on the other hand, do not consider efficiency during transients, thus, transients are either slow and somehow efficient, or fast and inefficient. The proposed scheme is efficient in steady-state but also enforces energy efficiency during torque transients, as far as this is physically possible. An implementation of the method requires the solution of a nonlinear optimization problem. An eye was kept on the most simple possible implementation, therefore no fancy nonlinear optimization schemes were chosen, only basic and widely known methods. This is important for a method to be reproducible. It is known and shown that the efficiency improvement based on field-weakening is only sensible in partial load operation, and that a flux adaptation may result in more losses than norm operation, however, the proposed method is always better than the widespread steady-state optimization. On an experimental flywheel test, it is shown that the dynamic optimization scheme can lead to a good efficiency improvement in a servo application.

Again, future improvements are possible. Current saturation was disregarded in the optimization, however, it could improve results, foremost for reference steps from low to high torque. It could further extend the improvement versus steady-state and rated operation.

Throughout these advanced designs, differential flatness has proved as a very helpful and enabling tool. Flatness is natural to an electrical drive, the outputs used in field-oriented control are flat outputs, application of this theory is straightforward. This theory aims at very complex problems, the drive control design task is, from

a theoretical standpoint, less difficult. Still, using the ideas from flatness-based control, considerable improvements in the design and in the results are obtained.

The design is simplified, the problem is simplified, this leads to simpler solutions, a crucial point as the key issues are the complex implementation and the high computational demands. Without the simplifications in the theoretical elaboration of the optimization problems, it would not be possible to obtain a feasible solution, in the sense of calculation effort. The implementation of MPC and of induction machine optimization with the online problem solvers would not have been possible.

Flatness-based control has coped major challenges of model-based control design. Foremost the results of robustness are helpful for predictive controllers. The deadbeat controller has not yet made the way to a broad industrial application, it is only found in some narrow fields, the major obstacle has been the robustness problem.

The flatness-based approach leads to significant improvements in the design and performance of advanced drive control algorithms. This theory is really helpful in applications.



---

# APPENDIX A

---

## List of symbols and abbreviations

---

### A.1 List of symbols

The following list only contains relevant symbols.

The applied convention is as follows:

Scalars are italic letters:	$x$
Vectors are bold lower case letters	$\boldsymbol{x}$
Matrices are bold upper case letters	$\boldsymbol{X}$
References are with a star superscript	$x^*$
Observed values are with a tilde	$\tilde{x}$
Estimated values are with a hat	$\hat{x}$

The reference frame (coordinate system) is denoted with a subscript after a comma for vectors:

Stator fixed frame	$_{,\alpha\beta}$
Rotor fixed frame	$_{,ab}$
Synchronous (rotor field oriented) frame	$_{,dq}$

For more concise notation, no comma is used in the subscript for scalars.

**General symbols:**

$\mathbf{x}$	State vector
$\mathbf{u}$	Input vector
$\mathbf{v}$	Brunovsky input (linearizing input)
$\mathbf{y}$	Output vector
$\mathbf{y}_f$	Flat output vector
$\mathbf{e}$	Disturbance vector
$\mathbf{f}$	System function (vector)
$\mathbf{A}$	System matrix
$\mathbf{B}$	Input vector
$\mathbf{K}$	Feedback gain vector
$G$	Transfer function
$\Delta$	Variation
$\Gamma_x$	Differential parameterization of the states
$\Gamma_u$	Differential parameterization of the inputs
$t$	Time (continuous)
$k$	Time (discrete)
$\Theta$	Moment of inertia

**General electrical variables:**

$i_u, i_v, i_w$	Phase currents [A]
$\mathbf{i}_u, \mathbf{i}_v, \mathbf{i}_w$	Phase current vectors (in $(\alpha, \beta)$ plane) [A]
$\mathbf{i}_s$	Stator current space vector [A]
$i_{sd}$	Direct (field-generating) stator current component [A]
$i_{sq}$	Quadrature (torque-generating) stator current component [A]
$\mathbf{i}_r$	Rotor current space vector (stator normed) [A]
$\mathbf{u}_s$	Stator voltage space vector [V]
$\Psi_s$	Stator flux space vector [Vs]
$\Psi_r$	Rotor flux space vector (stator normed) [Vs]
$\mathbf{A}_{\alpha\beta}^{uvw}$	Clarke transformation matrix from 3-phase to stator frame [-]
$\mathbf{A}_{dq}^{\alpha\beta}$	Park transformation matrix from stator frame to synchronous frame [-]
$n_p$	Number of pole pairs [-]
$\omega_M$	Mechanical speed [ $\frac{\text{rad}}{\text{s}}$ ]
$T_s$	Sampling time [s]
$T_{LP}$	Time constant of lowpass filter [s]
$\tau_M$	Generated torque [Nm]
$T$	Prediction resp. optimization horizon [s]

**Synchronous machine parameters:**

$L_d$	Mutual stator inductance in direct axis [H]
$L_q$	Mutual stator inductance in quadrature axis [H]
$\Psi_{PM}$	Flux magnitude of rotor permanent magnet [Vs]
$R_s$	Stator resistance [ $\Omega$ ]
$k_{Fe}$	Hysteresis loss coefficient [ $\frac{A}{\sqrt{S}}$ ]

**Induction machine parameters:**

$L_{s\sigma}$	Stator leakage inductance [H]
$L_{r\sigma}$	Rotor leakage inductance (stator normed) [H]
$L_m$	Mutual inductance (stator normed) [H]
$L_s = L_m + L_{s\sigma}$	Stator inductance [H]
$L_r = L_m + L_{r\sigma}$	Rotor inductance (stator normed) [H]
$R_r$	Rotor resistance (stator normed) [ $\Omega$ ]
$R_s$	Stator resistance [ $\Omega$ ]
$R_m$	Iron loss resistance (Mutual resistance) [ $\Omega$ ]
$\eta = \frac{R_r}{L_r}$	Inverse rotor time constant [ $\frac{1}{S}$ ]
$\sigma = 1 - \frac{L_m^2}{L_s L_r}$	Leakage coefficient [-]
$\beta = \frac{L_m}{\sigma L_s L_r}$	Coupling factor [ $\frac{A}{\sqrt{S}}$ ]
$\gamma = \frac{1}{\sigma L_s} (R_s + \frac{L_m^2}{L_r^2} R_r)$	Inverse stator time constant [ $\frac{1}{S}$ ]

## A.2 Abbreviations

2DoF	Two-degree-of-freedom (control)
A/D	Analog-to-digital (conversion)
AC	Alternating current
back-EMF	back electromotive force
BVP	Boundary value problem
CCF	Controller canonical form
CPU	Central processing unit
DC	Direct current
DSP	Digital signal processor
DTC	Direct torque control
FGM	Fast gradient method
FPGA	Field programmable gate array
GPC	Generalized predictive control
IGBT	Insulated gate bipolar transistor
IM	Induction machine
IVP	Initial value problem
LMI	Linear matrix inequality
LP	Linear programming (solver)
LQ	Linear-quadratic (problem)
MIMO	Multi-input multi-output (system)
MPC	Model predictive control
MTPA	Maximum torque per Ampere
PI	Proportional-integral (controller)
PM	Permanent-magnet
PMSM	Permanent-magnet synchronous machine
PTC	Predictive torque control
RL	Resistive-inductive
RMS	Real mean square
QP	Quadratic programming (solver)
SISO	Single-input single-output (system)
SynRM	Synchronous reluctance machine

## APPENDIX B

---

### Experimental setup

---

The test bench is shown in Fig. B.1. The real-time system is a PC104-based industrial computer with a 1.4 GHz Pentium M CPU and 1 GB Ram memory. The system is running Linux with an RTAI (real-time application interface) extension. Programming is done in the language *C*.

The rest of this system is as described in [ANK10]. Via a 16-bit ISA bus, some devices specific for drive system control are connected. An FPGA is responsible for space-vector modulation and interrupt generation, it also sends the switching signals to the inverter. Two synchronous A/D converters are applied to digitalize the analog current measurement signals from a current transducer. An encoder interface card is available to read the position of the shaft. There are four D/A channels and a digital scope to output and record internal controller signals in real-time. The system was originally designed at the University of Wuppertal, the only modification are an upgraded CPU and software, as well as an improvement of the real-time application system.

The system is nice and powerful, however, this version suffers from a major limitation. As described in [ANK10], the interrupt latency (the delay in the response) is about  $6 \mu\text{s}$ , caused by both a slow A/D converter and the interrupt handling software. The system must be in the zero-voltage-vector mode during quite some time in order to obtain current samples of acceptable quality. Furthermore only a sinus-triangle modulation is implemented. Because of this, the switching times and – as consequence – the voltage commands are limited. The maximum voltage vector

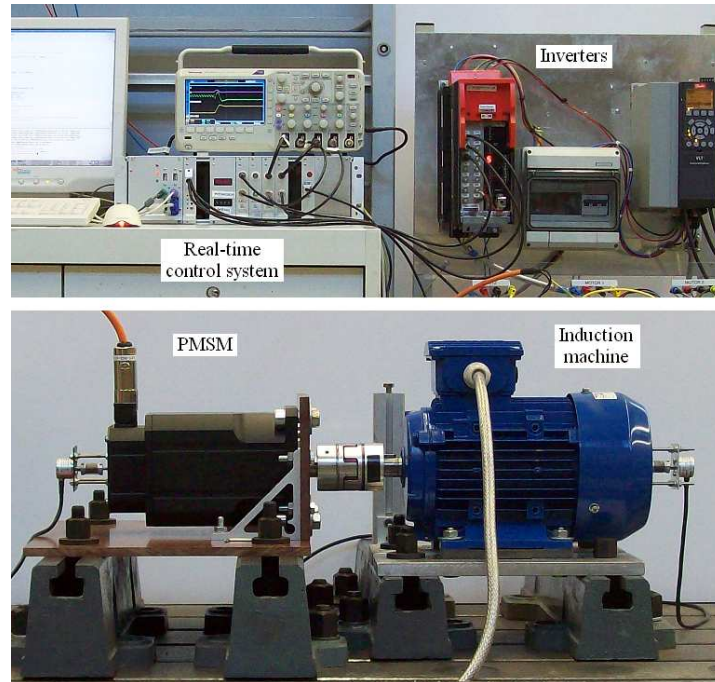


Figure B.1: Test bench: Real-time computer, power inverters, coupled synchronous and induction machine.

magnitude is limited to 220 V. If the system is redesigned such that the current measurement is improved and A/D conversion is faster, with a sinus-triangle modulation the maximum linear voltage would have been 280 V, and by implementing space vector modulation, the maximum linear voltage would have been 325 V [QD08]. The voltage limitation enforces a reduction of the maximum speed.

The test bench further consists of two industrial servo inverters, first a 12.5 A SEW-Eurodrive inverter where the IGBT switching signals can be directly transmitted, and second an unmodified Danfoss VLD FC-302 inverter. The SEW inverter is used for all experimental tests and the Danfoss inverter for the load drive to impose either a load torque or speed. The DC link voltage is not read by the control system. The Danfoss inverter is also used for identification of the induction machine parameters.

## B.1 Parameters of the test machines

Table B.1: Nominal Parameters of the Synchronous Machine

Manufacturer & Model	Merkes MT5 1050
Rated Power $P_N$	2760 W
Rated Torque $\tau_{MN}$	10.5 Nm
Rated Current (peak)	8 A
Rated Speed $\omega_{MN}$	3000 rpm
Rated Voltage $U_N$ (peak)	560 V
Pole Pairs $n_p$	3
Stator Inductance $L_d, L_q$	4.8, 7.2 mH
Stator Resistance $R$	0.92 $\Omega$
Motor Constant $\Psi_{PM}$ (peak)	0.334 Vs
Hysteresis Loss Constant $k_{Fe}$	1.27 $\frac{A}{Vs}$

Table B.2: Nominal Parameters of the Induction Machine

Manufacturer	MSF Vathauer GmbH & Co KG
Rated Power $P_N$	2200 W
Rated Torque $\tau_{MN}$	7.4 Nm
Rated Current (peak)	7 A
Rated Speed $\omega_{MN}$	2840 rpm
Power Factor $\cos \varphi$	0.85
Rated Voltage $U_N$ (peak)	560 V
Pole Pairs $n_p$	1
Mutual Inductance $L_m$	245 mH
Stator Inductance $L_s$	255 mH
Rotor Inductance $L_r$	255 mH
Stator Resistance $R_s$	2.66 $\Omega$
Rotor Resistance $R_r$	2.27 $\Omega$
Iron Loss Resistance $R_{Fe}$	1400 $\Omega$





---

## APPENDIX C

---

### Detailed machine model deductions

---

#### C.1 Synchronous machine

For the electrical subsystem, the first step is the voltage equation of the stator windings

$$\mathbf{u}_{s,\alpha\beta} = R_s \mathbf{i}_{s,\alpha\beta} + \frac{d}{dt} \Psi_{s,\alpha\beta}. \quad (\text{C.1})$$

Second, the flux linkage equation is applied. The total flux is the sum of the mutual flux generated by the stator windings and (possibly) the permanent-magnet rotor flux. For this model, leakage inductances and magnetic cross-couplings are ignored. The stator flux is given for simplicity in rotor reference frame, here, the dependency of the inductance values to the rotor position is simpler to represent. The equation is given by

$$\Psi_{s,dq} = \begin{pmatrix} L_d & 0 \\ 0 & L_q \end{pmatrix} \mathbf{i}_{s,dq} + \begin{pmatrix} \Psi_{PM} \\ 0 \end{pmatrix}. \quad (\text{C.2})$$

Transformation from stator to rotor reference frame is performed with the rotation matrix  $\mathbf{A}_{dq}^{\alpha\beta}$ , where the rotation angle  $\varphi$  is from  $\alpha$  to  $d$  axis in positive direction, as

defined in Fig. 2.2, and defined by

$$\mathbf{A}_{dq}^{\alpha\beta} = \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix}, \quad (\text{C.3})$$

The electrical rotor angle  $\varphi$  relates to the (measured) mechanical rotor angle by  $\varphi = n_p \varphi_M$  calibrated such that for  $\varphi = 0$  the  $\alpha$  and  $d$  axis are aligned. The notation is such that two current (or voltage or flux) vectors relate by the equation

$$\mathbf{i}_{s,dq} = \mathbf{A}_{dq}^{\alpha\beta} \mathbf{i}_{s,\alpha\beta}. \quad (\text{C.4})$$

The stator voltage equation transformed to rotor reference frame is therefore given by

$$\mathbf{A}_{dq}^{\alpha\beta} \mathbf{u}_{s,\alpha\beta} = R_s \mathbf{A}_{dq}^{\alpha\beta} \mathbf{i}_{s,\alpha\beta} + \mathbf{A}_{dq}^{\alpha\beta} \frac{d}{dt} \Psi_{s,\alpha\beta}, \quad (\text{C.5})$$

where the last term is developed as

$$\mathbf{A}_{dq}^{\alpha\beta} \frac{d}{dt} \Psi_{s,\alpha\beta} = \mathbf{A}_{dq}^{\alpha\beta} \frac{d}{dt} (\mathbf{A}_{\alpha\beta}^{dq} \Psi_{s,dq}) \quad (\text{C.6})$$

$$= \mathbf{A}_{dq}^{\alpha\beta} \left( \frac{d}{dt} \mathbf{A}_{\alpha\beta}^{dq} \right) \Psi_{s,dq} + \mathbf{A}_{dq}^{\alpha\beta} \mathbf{A}_{\alpha\beta}^{dq} \left( \frac{d}{dt} \Psi_{s,dq} \right) \quad (\text{C.7})$$

$$= \omega_e \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \Psi_{s,dq} + \frac{d}{dt} \Psi_{s,dq} \quad (\text{C.8})$$

$$= \omega_e \begin{pmatrix} -\Psi_{sq} \\ \Psi_{sd} \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} L_d i_{sd} \\ L_q i_{sq} \end{pmatrix}, \quad (\text{C.9})$$

where the electrical speed relates to the (measured) mechanical speed by  $\omega_e = n_p \omega_M$ . The state-space model of the electrical subsystem in rotor reference frame follows as

$$\frac{d}{dt} i_{sd} = -\frac{R_s}{L_d} i_{sd} + n_p \omega_M \frac{L_q}{L_d} i_{sq} + \frac{1}{L_d} u_{sd}, \quad (\text{C.10})$$

$$\frac{d}{dt} i_{sq} = -\frac{R_s}{L_q} i_{sq} - n_p \omega_M \frac{L_d}{L_q} i_{sd} - n_p \omega_M \Psi_{PM} + \frac{1}{L_q} u_{sq}. \quad (\text{C.11})$$

The torque is calculated by the cross product of stator flux and current (vectors are extended to  $\mathbb{R}^3$ )

$$\tau_M = \frac{3}{2}n_p(\Psi_{s,dq} \times \mathbf{i}_{s,dq}) \quad (\text{C.12})$$

$$= \frac{3}{2}n_p((\Psi_{sd}, \Psi_{sq}, 0)^T \times (i_{sd}, i_{sq}, 0)^T) \quad (\text{C.13})$$

$$= \frac{3}{2}n_p\Psi_{PM}i_{sq} + \frac{3}{2}n_p(L_d - L_q)i_{sd}i_{sq}. \quad (\text{C.14})$$

The first component  $\frac{3}{2}n_p\Psi_{PM}i_{sq}$  is the electromagnetic torque, based on the interaction between rotor flux and stator current. The second term  $\frac{3}{2}n_p(L_d - L_q)i_{sd}i_{sq}$  is the so-called reluctance torque, it is based on the minimum energy principle, the system will tend to the state with minimum potential energy, in this case, the rotor position where minimal magnetical energy is stored. The reluctance torque only appears if the inductances have different values. Interestingly, both effects are covered by Lorentz's law.

The state-space model is completed by the mechanical equation

$$\frac{d}{dt}\omega_M = \frac{1}{\Theta}(\tau_M - \tau_L), \quad (\text{C.15})$$

where  $\Theta$  is the moment of inertia and  $\tau_L$  is the load torque.

## C.2 Induction machine

For the electrical subsystem, the voltage equations of the stator and rotor windings read as

$$\mathbf{u}_{s,\alpha\beta} = R_s \mathbf{i}_{s,\alpha\beta} + \frac{d}{dt} \Psi_{s,\alpha\beta}, \quad (\text{C.16})$$

$$\mathbf{u}_{r,ab} = R_r \mathbf{i}_{r,ab} + \frac{d}{dt} \Psi_{r,ab}, \quad (\text{C.17})$$

where  $\mathbf{u}_{r,ab} = \mathbf{0}$  as only the squirrel-cage induction motor is regarded. Alternative induction machines are for instance the doubly-fed induction generator (DFIG) [Gen08], used in wind turbine systems, where rotor windings are fed by an inverter and therefore a nonzero input voltage. An outdated alternative induction machine is a wound-rotor slip-ring machine where an external 3-phase resistance is attached to the rotor during speedup to reduce the peak current, such that  $\mathbf{u}_{r,ab} = R_{ext} \dot{\mathbf{i}}_{r,ab}$ .

The flux linkage is the sum of the mutual flux, which is the flux passing through both stator and rotor windings, and the leakage flux which is only passing through one respective winding. The equations in both the stator and rotor are

$$\Psi_{s,\alpha\beta} = \Psi_{s\sigma,\alpha\beta} + \Psi_{sm,\alpha\beta} = L_{s\sigma}\mathbf{i}_{s,\alpha\beta} + L_m(\mathbf{i}_{s,\alpha\beta} + \mathbf{i}_{r,\alpha\beta}), \quad (\text{C.18})$$

$$\Psi_{r,ab} = \Psi_{r\sigma,ab} + \Psi_{rm,ab} = L_{r\sigma}\mathbf{i}_{r,ab} + L_m(\mathbf{i}_{r,ab} + \mathbf{i}_{s,ab}). \quad (\text{C.19})$$

As state variables, the stator current vector  $\mathbf{i}_{s,\alpha\beta}$  and the rotor flux vector  $\Psi_{r,\alpha\beta}$  are chosen. These are used in direct field-oriented control, where the measured stator current vector is decomposed in one component aligned to and orthogonal to the rotor flux. Therefore one needs to eliminate the two remaining vectors of the stator flux  $\Psi_{s,\alpha\beta}$  and the rotor current  $\mathbf{i}_{r,\alpha\beta}$ .

The transformation from rotor  $(a, b)$  to stator  $(\alpha, \beta)$  is performed by rotating by the electrical rotor angle. To find the rotor current vector, the rotor flux linkage is rewritten in stator fixed frame by multiplication with  $\mathbf{A}_{\alpha\beta}^{ab}$ ,

$$\Psi_{r,\alpha\beta} = (L_{r\sigma} + L_m)\mathbf{i}_{r,\alpha\beta} + L_m\mathbf{i}_{s,\alpha\beta}, \quad (\text{C.20})$$

from where the rotor current is found as

$$\mathbf{i}_{r,\alpha\beta} = \frac{1}{L_r}\Psi_{r,\alpha\beta} - \frac{L_m}{L_r}\mathbf{i}_{s,\alpha\beta}. \quad (\text{C.21})$$

This result is inserted into the flux linkage equation of the stator flux to find

$$\Psi_{s,\alpha\beta} = \left( L_s - \frac{L_m^2}{L_r} \right) \mathbf{i}_{s,\alpha\beta} + \frac{L_m}{L_r} \Psi_{r,\alpha\beta}. \quad (\text{C.22})$$

Therefore both vectors – stator flux and rotor current – can be substituted by expressions of the two chosen state vectors, the stator current and the rotor flux, without solving any differential equation and without considering any time dependency.

The voltage equation of the rotor is transformed to stator frame by multiplication with  $\mathbf{A}_{\alpha\beta}^{ab}$ ,

$$R_r \mathbf{A}_{\alpha\beta}^{ab} \mathbf{i}_{r,ab} + \mathbf{A}_{\alpha\beta}^{ab} \frac{d}{dt} \Psi_{r,ab} = 0, \quad (\text{C.23})$$

and by further substitution of  $\Psi_{r,ab}$ ,

$$R_r \mathbf{A}_{\alpha\beta}^{ab} \mathbf{i}_{r,ab} + \mathbf{A}_{\alpha\beta}^{ab} \frac{d}{dt} (\mathbf{A}_{ab}^{\alpha\beta} \Psi_{r,\alpha\beta}) = 0. \quad (\text{C.24})$$

The last term is developed as

$$\mathbf{A}_{\alpha\beta}^{ab} \frac{d}{dt} \left( \mathbf{A}_{ab}^{\alpha\beta} \Psi_{r,\alpha\beta} \right) = \mathbf{A}_{\alpha\beta}^{ab} \frac{d}{dt} \left( \mathbf{A}_{ab}^{\alpha\beta} \right) \Psi_{r,\alpha\beta} + \mathbf{A}_{\alpha\beta}^{ab} \mathbf{A}_{ab}^{\alpha\beta} \left( \frac{d}{dt} \Psi_{r,\alpha\beta} \right) \quad (\text{C.25})$$

$$= \omega_e \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \Psi_{r,\alpha\beta} + \frac{d}{dt} \Psi_{r,\alpha\beta}. \quad (\text{C.26})$$

The first differential equation – the state space description of the rotor flux – therefore follows as

$$\frac{d}{dt} \Psi_{r,\alpha\beta} = -n_p \omega_M \begin{pmatrix} -\Psi_{r\beta} \\ \Psi_{r\alpha} \end{pmatrix} - \frac{R_r}{L_r} \Psi_{r,\alpha\beta} + \frac{R_r L_m}{L_r} \mathbf{i}_{s,\alpha\beta}. \quad (\text{C.27})$$

The voltage equation in the stator with substituted stator flux expression is

$$\mathbf{u}_{s,\alpha\beta} = R_s \mathbf{i}_{s,\alpha\beta} + \frac{d}{dt} \left( \left( L_s - \frac{L_m^2}{L_r} \right) \mathbf{i}_{s,\alpha\beta} + \frac{L_m}{L_r} \Psi_{r,\alpha\beta} \right) \quad (\text{C.28})$$

$$= R_s \mathbf{i}_{s,\alpha\beta} + \left( L_s - \frac{L_m^2}{L_r} \right) \frac{d}{dt} \mathbf{i}_{s,\alpha\beta} + \frac{L_m}{L_r} \frac{d}{dt} \Psi_{r,\alpha\beta}, \quad (\text{C.29})$$

where  $\frac{d}{dt} \Psi_{r,\alpha\beta}$  is replaced by the expression above, rearrangement leads to

$$\begin{aligned} \frac{d}{dt} \mathbf{i}_{s,\alpha\beta} = \frac{1}{L_s - \frac{L_m^2}{L_r}} & \left( -R_s \mathbf{i}_{s,\alpha\beta} + \mathbf{u}_{s,\alpha\beta} - \frac{L_m}{L_r} n_p \omega_M \begin{pmatrix} -\Psi_{r\beta} \\ \Psi_{r\alpha} \end{pmatrix} \right. \\ & \left. + \frac{R_r L_m}{L_r^2} \Psi_{r,\alpha\beta} - \frac{R_r L_m^2}{L_r^2} \mathbf{i}_{s,\alpha\beta} \right). \end{aligned} \quad (\text{C.30})$$

This complicated expression is now simplified with some new parameter definitions, typical in the literature [KKM03, QD08]. These are:

Inverse rotor time constant	$\eta = \frac{R_r}{L_r}$
Leakage (dispersion) coefficient	$\sigma = 1 - \frac{L_m^2}{L_s L_r}$
Coupling factor	$\beta = \frac{L_m}{\sigma L_s L_r}$
Inverse stator time constant	$\gamma = \frac{1}{\sigma L_s} \left( R_s + \frac{L_m^2}{L_r^2} R_r \right)$

With these definitions and the two vector differential equations, the state-space

model in stator-fixed frame is found as

$$\frac{d}{dt}i_{s\alpha} = -\gamma i_{s\alpha} + \beta\eta\Psi_{r\alpha} + \beta n_p\omega_M\Psi_{r\beta} + \frac{1}{\sigma L_s}u_{s\alpha}, \quad (\text{C.31})$$

$$\frac{d}{dt}i_{s\beta} = -\gamma i_{s\beta} + \beta\eta\Psi_{r\beta} - \beta n_p\omega_M\Psi_{r\alpha} + \frac{1}{\sigma L_s}u_{s\beta}, \quad (\text{C.32})$$

$$\frac{d}{dt}\Psi_{r\alpha} = -\eta\Psi_{r\alpha} - n_p\omega_M\Psi_{r\beta} + \eta L_m i_{s\alpha}, \quad (\text{C.33})$$

$$\frac{d}{dt}\Psi_{r\beta} = -\eta\Psi_{r\beta} + n_p\omega_M\Psi_{r\alpha} + \eta L_m i_{s\beta}. \quad (\text{C.34})$$

The first two equations are denoted as *stator model* whereas the last two equations are denoted as *rotor model*.

This model is used for the design of the rotor flux observer. The stator current vector  $\mathbf{i}_{s\alpha\beta}$  is known as it is measured, same as for the speed  $\omega_M$ . The voltage  $\mathbf{u}_{s\alpha\beta}$  are also known from the voltage commands if the inverter nonlinearities are compensated. Only the rotor flux  $\Psi_{r\alpha\beta}$  is unknown. It is clear that due to the many disturbances and parametric uncertainties, a closed-loop full-order observer is required [MR00], an open-loop observer (for instance, an integration of the rotor model) will be inaccurate and considerably reduce performance of the control system.

The torque is the cross product of stator flux and stator current,

$$\tau_M = \frac{3}{2}n_p(\Psi_{s,dq} \times \mathbf{i}_{s,dq}) \quad (\text{C.35})$$

$$= \frac{3}{2}n_p \left( \left( \left( L_s - \frac{L_m^2}{L_r} \right) \mathbf{i}_{s,\alpha\beta} + \frac{L_m}{L_r} \Psi_{r,\alpha\beta} \right) \times \mathbf{i}_{s,dq} \right) \quad (\text{C.36})$$

$$= \frac{3}{2}n_p \frac{L_m}{L_r} (\Psi_{r,\alpha\beta} \times \mathbf{i}_{s,dq}), \quad (\text{C.37})$$

explicitly given by

$$\tau_M = \frac{3}{2}n_p \frac{L_m}{L_r} (\Psi_{r\alpha} i_{s\beta} - \Psi_{r\beta} i_{s\alpha}). \quad (\text{C.38})$$

The mechanical equation is

$$\frac{d}{dt}\omega_M = \frac{1}{\Theta}(\tau_M - \tau_L), \quad (\text{C.39})$$

where  $\Theta$  is the moment of inertia and  $\tau_L$  is the load torque.

In the second step, the model is transformed into the field-oriented frame. The goal is to decouple control of torque and of flux magnitude. The transformation

into the rotor field oriented frame is therefore a physically motivated coordinate transformation.

Technically, it can be seen that it is a transformation of the rotor flux vector from cartesian to polar representation [DLO01]. The rotor is represented as rotor flux magnitude

$$\Psi_{rd} = \sqrt{\Psi_{r\alpha}^2 + \Psi_{r\beta}^2}, \quad (\text{C.40})$$

and as rotor flux angle

$$\rho = \arctan \frac{\Psi_{r\beta}}{\Psi_{r\alpha}} \quad (\text{C.41})$$

for  $\Psi_{r\alpha} \geq 0$  respectively  $\rho = \arctan \frac{\Psi_{r\beta}}{\Psi_{r\alpha}} + \pi$  for  $\Psi_{r\alpha} < 0$ .

However the typical interpretation is to rotate the coordinate frame to the synchronous frame, where the direct axis is aligned to the rotor flux vector. Therefore the rotor flux in the transformed coordinates is defined as

$$\Psi_{r,dq} = \begin{pmatrix} \Psi_{rd} \\ 0 \end{pmatrix}, \quad (\text{C.42})$$

where  $\Psi_{rd}$  is the rotor flux magnitude.

The transformation from stator to rotor flux (or synchronous) frame is performed with the rotation matrix  $\mathbf{A}_{dq}^{\alpha\beta}$ ,

$$\mathbf{A}_{dq}^{\alpha\beta} = \begin{pmatrix} \cos \rho & \sin \rho \\ -\sin \rho & \cos \rho \end{pmatrix}, \quad (\text{C.43})$$

where the notation is such that

$$\mathbf{i}_{s,dq} = \mathbf{A}_{dq}^{\alpha\beta} \mathbf{i}_{s,\alpha\beta}. \quad (\text{C.44})$$

The differential equation of the stator current is multiplied by the transformation matrix  $\mathbf{A}_{dq}^{\alpha\beta}$  where additional terms appear from transforming the current derivatives

$$\mathbf{A}_{dq}^{\alpha\beta} \frac{d}{dt} \mathbf{i}_{s,\alpha\beta} = \mathbf{A}_{dq}^{\alpha\beta} \frac{d}{dt} \left( \mathbf{A}_{\alpha\beta}^{dq} \mathbf{i}_{s,dq} \right) \quad (\text{C.45})$$

$$= \left( \frac{d}{dt} \rho \right) \begin{pmatrix} -i_{sq} \\ i_{sd} \end{pmatrix} + \frac{d}{dt} \mathbf{i}_{s,dq}. \quad (\text{C.46})$$

The derivative of the rotor flux angle is found as

$$\frac{d}{dt}\rho = n_p\omega_M + \eta L_m \frac{\dot{i}_{sq}}{\Psi_{rd}}. \quad (\text{C.47})$$

The complete model in field-oriented coordinates is given by five differential equations [KKM03, QD08]. The first three differential equations describe the dynamics of the currents and the rotor flux magnitude,

$$\frac{d}{dt}i_{sd} = -\gamma i_{sd} + \beta\eta\Psi_{rd} + n_p\omega_M i_{sq} + \eta L_m \frac{\dot{i}_{sq}^2}{\Psi_{rd}} + \frac{1}{\sigma L_s} u_{sd}, \quad (\text{C.48})$$

$$\frac{d}{dt}i_{sq} = -\gamma i_{sq} - \beta n_p\omega_M \Psi_{rd} - n_p\omega_M i_{sd} - \eta L_m \frac{i_{sd}\dot{i}_{sq}}{\Psi_{rd}} + \frac{1}{\sigma L_s} u_{sq}, \quad (\text{C.49})$$

$$\frac{d}{dt}\Psi_{rd} = -\eta\Psi_{rd} + \eta L_m i_{sd}, \quad (\text{C.50})$$

and the fourth differential equation is giving the rotor flux angle derivative, or the electrical excitation frequency,

$$\frac{d}{dt}\rho = n_p\omega_M + \eta L_m \frac{\dot{i}_{sq}}{\Psi_{rd}}. \quad (\text{C.51})$$

The output equation for the torque simplifies to

$$\tau_M = \frac{3}{2}n_p \frac{L_m}{L_r} \Psi_{rd} \dot{i}_{sq}. \quad (\text{C.52})$$

The fifth equation describes the dynamics of the mechanical subsystem,

$$\frac{d}{dt}\omega_M = \frac{1}{\Theta}(\tau_M - \tau_L). \quad (\text{C.53})$$



---

## APPENDIX D

---

### Robustness calculations for deadbeat control

---

The robustness regarding disturbances and parameter uncertainties is analyzed. As simplification, a linear single-input single-output (SISO) analysis is performed for a single RL-load. The model is given as

$$i[k + 1] = (1 - a)i[k] + bu[k] - be[k], \quad (\text{D.1})$$

where the parameters are

$$a = T_s \frac{R}{L}, \quad b = T_s \frac{1}{L}, \quad (\text{D.2})$$

and  $e[k]$  is a possibly time-varying disturbance signal.

#### D.1 Transfer function of deadbeat controller

First, the transfer function of a deadbeat controller with delay compensation but without disturbance estimator is calculated.

The calculation of the feedback value

$$i_{FB}[k] = qi[k|k-1] + (1 - q)i^*[k-1] \quad (\text{D.3})$$

and the delay compensation

$$i[k|k-1] = (1 - \hat{a})i[k-1] + \hat{b}u[k-1] \quad (\text{D.4})$$

are inserted in the deadbeat control law

$$u[k] = \frac{1}{\hat{b}}(i^*[k] - (1 - \hat{a})i_{FB}[k]). \quad (\text{D.5})$$

Inserting the plant equation (D.1) in (D.5), the resulting closed-loop dynamics in the discrete-time frequency domain appear as

$$G_{db}(z) = \frac{I(z)}{I^*(z)} = \frac{b_0z + b_1}{a_0z^2 + a_1z + a_2} \quad (\text{D.6})$$

where

$$a_0 = \frac{1}{\hat{b}}, \quad (\text{D.7})$$

$$a_1 = -\frac{1}{\hat{b}}(1 - a) + q\frac{1}{\hat{b}}(1 - \hat{a}), \quad (\text{D.8})$$

$$a_2 = q\frac{1}{\hat{b}}(1 - \hat{a})^2 - q\frac{1}{\hat{b}}(1 - a)(1 - \hat{a}), \quad (\text{D.9})$$

and

$$b_0 = \frac{1}{\hat{b}}, \quad (\text{D.10})$$

$$b_1 = -(1 - q)\frac{1}{\hat{b}}(1 - \hat{a}). \quad (\text{D.11})$$

Under correct parameters,  $a_1 = a_2 = b_1 = 0$ , and with  $q = 1$ , the transfer function is the deadbeat condition  $i[k + 1] = i^*[k]$ . If the parameters do not match, there are additional dynamics, as it is a second-order plant consisting of delay plus current dynamics. Two poles and one transfer zero determine the dynamics.

## D.2 Transfer function including a disturbance estimator

Now the complete transfer function including the disturbance estimator is calculated. It is necessary to analyze the behavior under a fast disturbance estimator (i.e. small  $T_{LP}$ ) to see the interaction under parameter faults. The closed-loop system has one reference input  $i^*[k]$ , one disturbance input  $e[k]$  and one output  $i[k]$ .

Two transfer functions of the closed-loop system appear, the first one is the response from the reference to the output and the second one is from the disturbance to the output,

$$I(z) = G_{i^*i}I^*(z) + G_{ei}E(z). \quad (\text{D.12})$$

**Reference response**

The calculations of the feedback value

$$i_{FB}[k] = qi[k|k-1] + (1-q)i^*[k-1] \quad (\text{D.13})$$

and the delay compensation

$$i[k|k-1] = (1-\hat{a})i[k-1] + \hat{b}u_R[k-1] \quad (\text{D.14})$$

are inserted into the deadbeat control law

$$u_R[k] = \frac{1}{\hat{b}}(i^*[k] - (1-\hat{a})i_{FB}[k]). \quad (\text{D.15})$$

Additionally, the disturbance estimator

$$\hat{e}[k] = \alpha(u_R[k-2] - \frac{1}{\hat{b}}i[k-1] + \frac{1}{\hat{b}}(1-\hat{a})i[k-2]) + \hat{e}[k-1] \quad (\text{D.16})$$

and the plant dynamics

$$i[k+1] = (1-a)i[k] + bu[k] - be[k], \quad (\text{D.17})$$

where  $u[k] = u_R[k] + \hat{e}[k]$ , are considered. The disturbance input  $e[k]$  is assumed to be 0.

The resulting closed-loop transfer function in the discrete-time frequency domain reads as

$$G_{i^*i}(z) = \frac{I(z)}{I^*(z)} = \frac{b_1z^2 + b_2z + b_3}{a_0z^3 + a_1z^2 + a_2z + a_3}, \quad (\text{D.18})$$

where

$$a_0 = \hat{b}, \quad (\text{D.19})$$

$$a_1 = -\hat{b} - \hat{b}(1-a) + \hat{b}q(1-\hat{a}), \quad (\text{D.20})$$

$$a_2 = b(1-\hat{a})^2q - \hat{b}(1-a)q(1-\hat{a}) + \hat{b}(1-a) + b\alpha - \hat{b}q(1-\hat{a}), \quad (\text{D.21})$$

$$a_3 = \hat{b}(1-a)q(1-\hat{a}) - b(1-\hat{a})^2q + b\alpha q(1-\hat{a}) - b\alpha(1-\hat{a}), \quad (\text{D.22})$$

and

$$b_1 = b \quad (\text{D.23})$$

$$b_2 = b(-1 - (1-q)(1-\hat{a})), \quad (\text{D.24})$$

$$b_3 = b(\alpha + (1-q)(1-\hat{a})). \quad (\text{D.25})$$

### Disturbance response

The calculations of the feedback value

$$i_{FB}[k] = qi[k|k-1] + (1-q)i^*[k-1] \quad (\text{D.26})$$

and the delay compensation

$$i[k|k-1] = (1-\hat{a})i[k-1] + \hat{b}u_R[k-1] \quad (\text{D.27})$$

are inserted into the deadbeat control law

$$u_R[k] = \frac{1}{\hat{b}}(i^*[k] - (1-\hat{a})i_{FB}[k]) \quad (\text{D.28})$$

where the reference input  $i^*[k]$  is assumed to be 0. Additionally, the disturbance estimator

$$\hat{e}[k] = \alpha(u_R[k-2] - \frac{1}{\hat{b}}i[k-1] + \frac{1}{\hat{b}}(1-\hat{a})i[k-2]) + \hat{e}[k-1] \quad (\text{D.29})$$

and the plant dynamics

$$i[k+1] = (1-a)i[k] + bu[k] - be[k], \quad (\text{D.30})$$

where  $u[k] = u_R[k] + \hat{e}[k]$ , are considered.

The resulting closed-loop transfer function in discrete-time frequency domain appears as

$$G_{ei}(z) = \frac{I(z)}{E(z)} = \frac{b_1z^2 + b_2z + b_3}{a_0z^3 + a_1z^2 + a_2z + a_3} \quad (\text{D.31})$$

where

$$a_0 = -\hat{b} \quad (\text{D.32})$$

$$a_1 = -\hat{b}(1-\hat{a})q + \hat{b} + (1-a)\hat{b} \quad (\text{D.33})$$

$$a_2 = \hat{b}(1-\hat{a})q + (1-a)\hat{b}(1-\hat{a})q - (1-a)\hat{b} - bq(1-\hat{a})^2 - b\alpha \quad (\text{D.34})$$

$$a_3 = -b\alpha(1-\hat{a})q + b\alpha(1-\hat{a}) - (1-a)\hat{b}(1-\hat{a})q + bq(1-\hat{a})^2 \quad (\text{D.35})$$

and

$$b_1 = -\hat{b}b \quad (\text{D.36})$$

$$b_2 = \hat{b}\hat{b}(1-q(1-\hat{a})) \quad (\text{D.37})$$

$$b_3 = \hat{b}\hat{b}(1-\hat{a})q. \quad (\text{D.38})$$

### D.3 Interaction of disturbance estimator and deadbeat controller

Under ideal conditions, the deadbeat controller and the disturbance estimator are decoupled. However, a lowpass filter is still required for stable operation. Assuming  $(1 - \hat{a}) = 1$  and  $(1 - a) = 1$ , which is satisfied if the open-loop time constant  $\frac{L}{R}$  is considerably larger than the sampling interval  $T_s$ , as  $a = T_s \frac{R}{L}$ . The denominator of both the reference response  $G_{i^*i}(z)$  and  $G_{ei}(z)$  become

$$(1 - z)z^2 + (1 - z)qz - (1 - z)z + (1 - z)\alpha - \alpha q = 0. \quad (\text{D.39})$$

For the special case  $q = 1$  (conventional deadbeat control) this simplifies to

$$z(-z^2 + z - \alpha) = 0, \quad (\text{D.40})$$

and for  $q = 0$  (feedforward linearization) this simplifies to

$$(z - 1)(-z^2 + z - \alpha) = 0. \quad (\text{D.41})$$

Therefore the two closed-loop poles depending on  $\alpha = \frac{T_s}{T_{LP} + T_s}$  and therefore on the lowpass filter time constant  $T_{LP}$  are

$$z_{1/2} = \frac{1 \pm \sqrt{1 - 4\alpha}}{2}. \quad (\text{D.42})$$

It is seen that these become complex for  $\alpha \geq \frac{1}{4}$ , for any value below, the poles are real and always of magnitude lower than 1. As  $\alpha = \frac{T_s}{T_{LP} + T_s}$  it follows that the condition  $\alpha \geq \frac{1}{4}$  means

$$T_{LP} \geq 3T_s. \quad (\text{D.43})$$

Furthermore it is seen that without lowpass filter, where  $T_{LP} = 0$  and  $\alpha = 1$ , the poles are  $z_{1/2} = \frac{1}{2} \pm j\frac{\sqrt{3}}{2}$  which both have magnitude 1, therefore, the closed-loop system is unstable without lowpass filter. If the system is subject to strong uncertainties, the filter must be set even slower by decreasing  $\alpha$ , and continuously by decreasing  $\alpha$  closed-loop performance gets lost.

### D.4 Steady-state accuracy

Steady-state accuracy of the deadbeat controllers with delay compensation and disturbance estimator is analyzed. In steady-state, the reference, disturbance and the output are constant, thus  $u[k] = \bar{u}$ ,  $i[k] = \bar{i}$  and  $\hat{e}[k] = \bar{e} \forall k$ .

Then, the controller equation turns into

$$\hat{b}\bar{u}_R = (1 - (1 - \hat{a})(1 - q))\bar{i}^* - (1 - \hat{a})q\bar{i}[k|k - 1] \quad (\text{D.44})$$

where  $\bar{i}[k|k - 1]$  denotes the predicted current from the delay compensation algorithm

$$\bar{i}[k|k - 1] = (1 - \hat{a})\bar{i} + \hat{b}\bar{u}_R. \quad (\text{D.45})$$

Furthermore, the disturbance estimator ensures

$$\hat{b}\bar{u}_R = \hat{a}\bar{i}. \quad (\text{D.46})$$

Solving this system of equations by successive substitution, the steady-state accuracy follows as

$$\bar{i}(\hat{a} + q) = \bar{i}^*(\hat{a} + q - \hat{a}q), \quad (\text{D.47})$$

therefore, the steady-state offset is

$$\frac{\bar{i}^*}{\bar{i}} = \frac{\hat{a} + q}{\hat{a} + q - \hat{a}q} = \frac{1}{1 - \frac{\hat{a}q}{\hat{a} + q}} \approx 1 + \frac{\hat{a}q}{\hat{a} + q}. \quad (\text{D.48})$$

Noting that  $\hat{a} \ll 1$ , only a small prediction error remains. For fast sampling systems with large open-loop time constants, this error is negligible.

## APPENDIX E

---

### Outlook: Feedforward linearization for predictive torque control (PTC)

---

This appendix applies the method presented in chapters 3 and 4 to predictive torque control (PTC). It is, however, not intended to be a comparison between field-oriented and direct control methods. The systematic differences, already mentioned in section 2.4, are considerable. Field-oriented control has a constant switching frequency and ignores the ripple of PWM frequency, it is de facto standard in commercial small-power servo drives. Direct torque control (DTC), as an alternative scheme, comes with a constant ripple which is defined by tolerance bands in the controller tuning, but the switching frequency is variable, which is an issue as it may cause excessive noise, especially in the low speed area. Motivated by this, and knowing that DTC still has advantages, PTC has been developed, here, the current ripple can be included, while it is also possible to keep the switching frequency constant [PW05]. In any case a fair comparison requires that the switching frequency of both methods is equal. Such a comparison can be found in [RKE<sup>+</sup>12].

The PTC method implemented here is very basic and does not adjust the switching frequency. For that reason, this appendix has the mere purpose of demonstrating that feedforward linearization, a flatness-based method to increase robustness of model-based control [HD03], is applicable to PTC and has its merits. The result is on a pure experimental basis and is not supported by an analytical study.

The comparison of the methods in terms of robustness is interesting, however. In

order to have a comparable current step response in field-oriented control and DTC / PTC, it is necessary to extend the field-oriented controller with a fast current controller, as it was done with deadbeat control in chapters 3 and 4. Then, as discussed, it is well-known that the sensitivity to parameters becomes high. An inductance error leads to overshoots, badly damped oscillations or even instability. For instance, it was shown in chapter 4 that the conventional deadbeat controller cannot be applied as-is to an induction machine, a robust extension or an adaptation of the parameters is required. For DTC and PTC, such a high sensitivity is not known, even though the torque and current response is also extremely fast. These schemes do not become unstable on inductance uncertainties. However, some negative side-effects appear, the switching frequency increases, thereby acoustic noise appears. Furthermore, a higher current ripple appears if the inductance is overestimated.

Two experimental tests are performed for PTC on a PMSM (Machine: Merkes MT4 0530, rated values: 3000 rpm,  $n_p = 3$ , 5.3 A (peak),  $L_d = 10$  mH,  $L_q = 14$  mH,  $R = 2.2 \Omega$ ). A speedup from 0 to rated speed 3000 rpm is performed. The speed controller is a cascaded PI controller, the torque and current is controlled by PTC (prediction horizon 1 step, cost function  $J = (i_q[k+1] - i_q^*[k])^2 + i_d^2[k+1]$ , sampling frequency = 32.5 kHz). The method is very simple: the continuous current signal used as initial condition for the prediction  $\hat{i}[k|k-1]$  is replaced by a mix between reference and measured signal,  $\hat{i}_{FB}[k] = q\hat{i}[k|k-1] + (1-q)i^*[k-1]$ . Again, a reference governor is implemented that adjusts (slows down)  $i^*[k-1]$  for the case that the reference step is so high that it cannot be reached in the next step due to the voltage limitation. No further modification is necessary. The response is now analyzed for different values of  $q$ , where  $q = 1$  is the unmodified PTC controller and  $q = 0$  would be a feedforward controller, and any  $0 < q < 1$  is a mixed method.

In the first test, shown in Fig. E.1, the current reference is limited to 10 A. In subfigures (a)-(c), for the respective values  $q = 1, 0.5$  and  $0.2$ , not much difference is seen. This is as expected as there is no error in the model. The magnitude spectrum of the torque-generating current  $i_q$  is shown in subfigure (d) for the time where  $i_q \approx 10$  A. The difference between the three methods is small. In Fig. E.3 on subfigures (a)-(c), a zoom of the step response at startup is plotted, it is clearly seen that the tracking capability is untouched.

The second test is shown in Fig. E.2, here, the current reference is limited to 24 A. The time response in subfigures (a)-(c) shows that for the respective values  $q = 1, 0.5$  and  $0.2$ , current tracking is fine. In subfigure (c), a slight drift is seen during acceleration in high speed, probably caused by modeling errors, the controller has lost some performance. Subfigure (d) shows the magnitude spectrum of the torque-generating current  $i_q$  for the duration where  $i_q$  is at the maximum value.



---

Due to the high current, saturation effects appear, as described in section 2.6, the correct inductance value becomes smaller than the nominal value  $L_q = 14$  mH. This overestimation leads to an excessive voltage use of the controller, thereby inheriting a high current ripple. By comparing Figs. E.1 and E.2, it can clearly be seen that the ripple is much higher at  $i_q = 24$  A than at  $i_q = 10$  A. In the magnitude spectrum, a peak appears around 6 – 7 kHz that describes this ripple. For the feedforward linearization approach with  $q = 0.5$ , there is not much change compared to PTC, but at  $q = 0.2$ , this peak disappears. In Fig. E.3 on subfigures (d)-(f), again, a zoom of the step response at startup is plotted, it is clearly seen that the reference response is untouched, the good tracking capability of PTC is maintained.

Therefore, the proposed method of mixing feedforward linearization to the conventional predictive controller is also effective here. The negative effects of a parameter error, foremost of an inductance error, are visibly attenuated, while the control performance only marginally decreases. Flatness-based control is interesting for everybody, not only for field-oriented control.

A full evaluation is left open, for that, analytical arguments must be given, and factors such as frequency, current ripple, suppression of harmonics, etc., must be studied in detail.

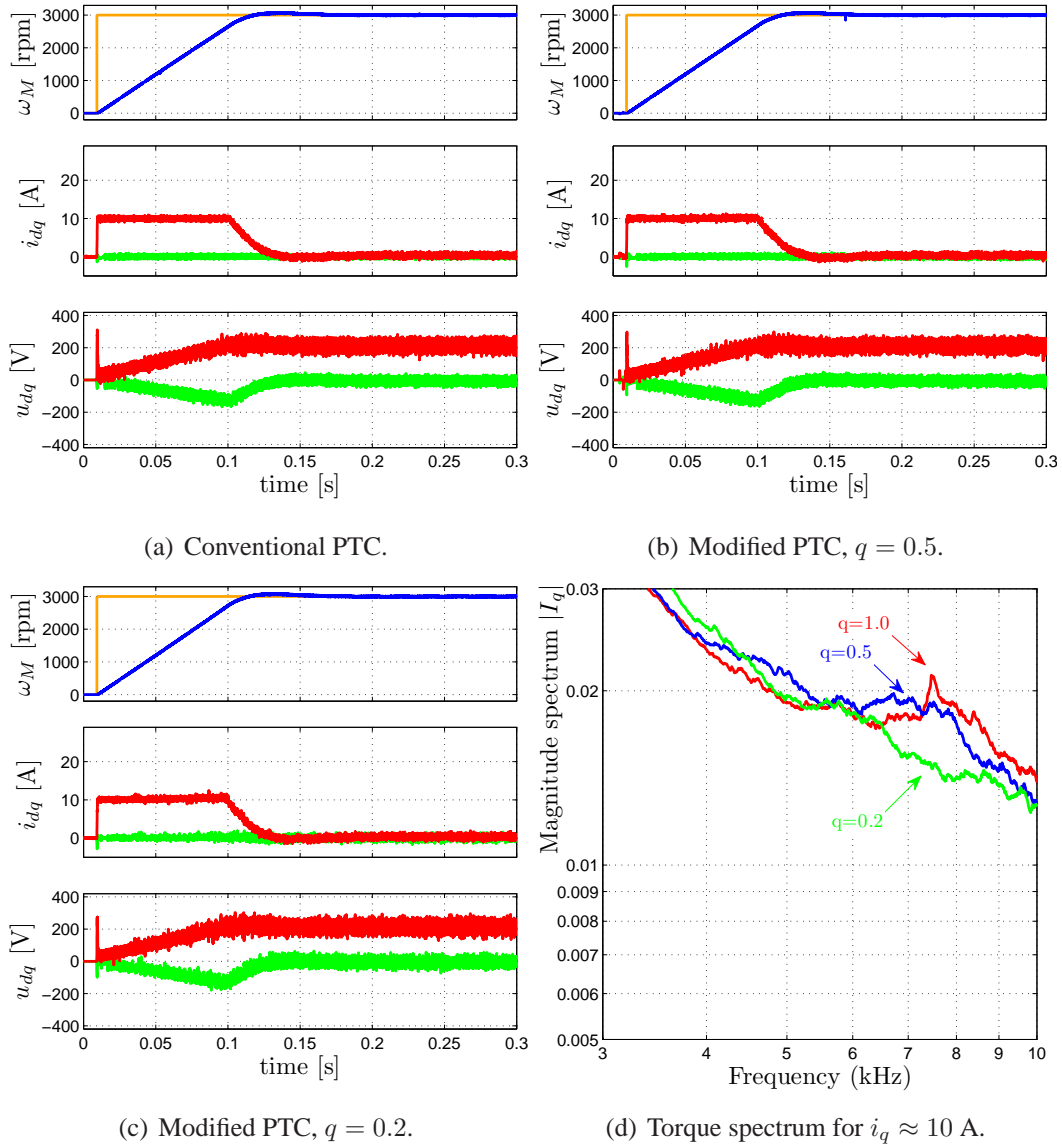


Figure E.1: PTC on a PMSM, speed control, rated current 10 A. From top: measured speed  $\omega_M$ , measured currents in synchronous frame (green  $i_d$ , red:  $i_q$ ), voltage commands in synchronous frame (green  $u_d$ , red:  $u_q$ , signals are low-pass filtered for readability).

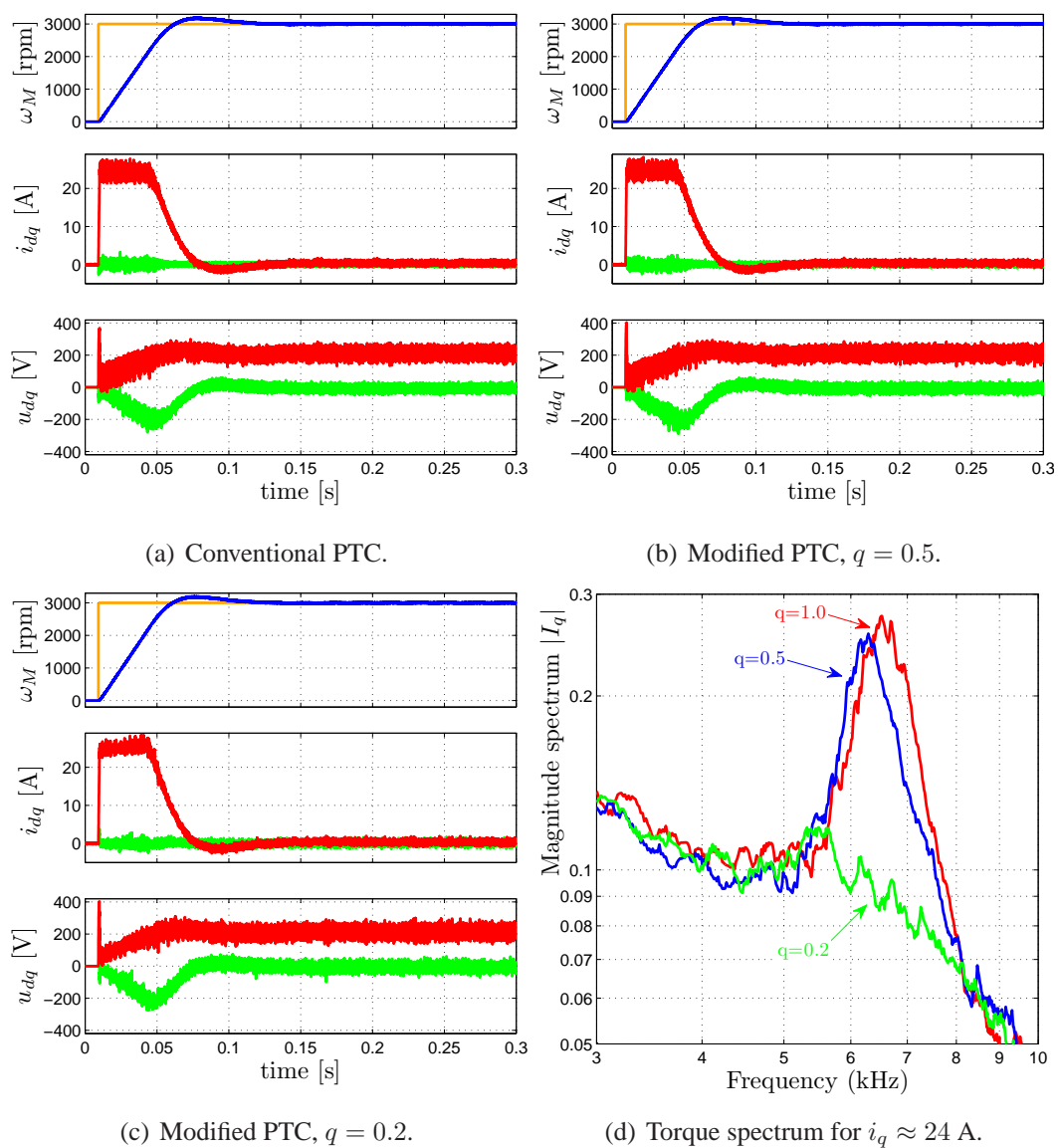


Figure E.2: PTC on a PMSM, speed control, overcurrent 24 A. From top: measured speed  $\omega_M$ , measured currents in synchronous frame (green  $i_d$ , red:  $i_q$ ), voltage commands in synchronous frame (green  $u_d$ , red:  $u_q$ , signals are low-pass filtered for readability).

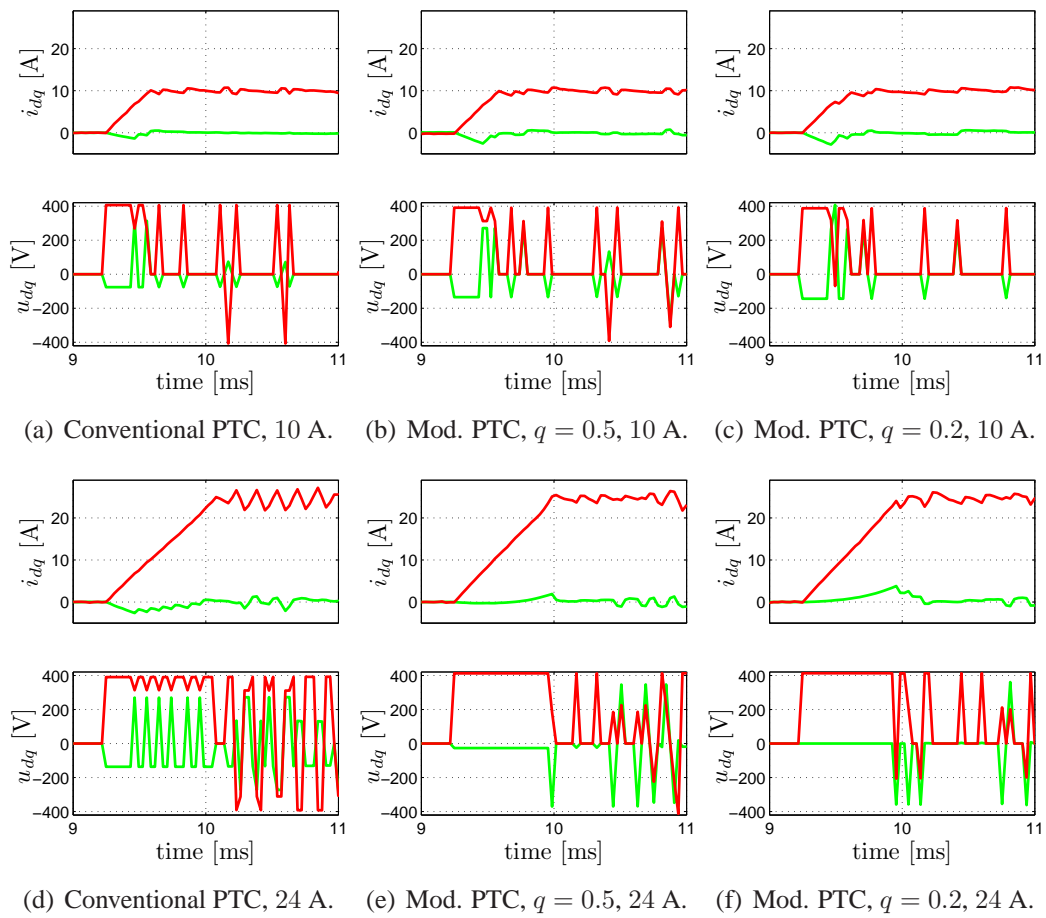


Figure E.3: PTC on a PMSM, zooms of the current step response of the previous figures. From top: measured currents in synchronous frame (green  $i_d$ , red:  $i_q$ ), voltage commands in synchronous frame (green  $u_d$ , red:  $u_q$ , not filtered).

---

## APPENDIX F

---

### Mathematical complements to chapters 5 & 6

---

#### F.1 Analysis of steady state accuracy of the cost function.

In steady-state, as the two current components are constant  $i_d(t) = i_d$  and  $i_q(t) = i_q$ , the cost function

$$J = \int_0^T (P_{Ctrl}(t) + w_L P_{Loss}(t)) dt + T P_{Loss}(T) \quad (F.1)$$

simplifies to

$$J = 2T P_{Ctrl} + T w_L P_{Loss}, \quad (F.2)$$

with

$$P_{Ctrl} = \left( \frac{3}{2} n_p \Psi_{PM} i_q - \tau_M^* \right)^2, \quad (F.3)$$

$$P_{Loss} = \frac{3}{2} n_p \omega_M k_F \left( (L_q i_q)^2 + (L_d i_d + \Psi_{PM})^2 \right) + \frac{3}{2} R_s (i_d^2 + i_q^2). \quad (F.4)$$

The steady-state optimum is found by solving the two equations

$$\frac{\partial J}{\partial i_d} = 0, \quad \frac{\partial J}{\partial i_q} = 0, \quad (F.5)$$

which are fully independent (say  $\frac{\partial i_q}{\partial i_d} = 0$ ) as the reluctance torque, resp. the influence of saliency  $L_d \neq L_q$  was neglected in the torque equation  $\tau_M = \frac{3}{2}n_p\Psi_{PM}i_q$ , compare [MXM00,MKM04]. This is untrue for interior PMSMs but fine for surface-mounted PMSMs.

Solving (F.5) leads to the optimal steady-state value for  $i_d$

$$i_d^{opt} = -\frac{L_d\Psi_{PM}}{L_d^2 + \frac{R_s}{n_p\omega_M k_{Fe}}}, \quad (\text{F.6})$$

which is the value for  $i_d$  that minimizes the sum of copper and hysteresis iron losses, as discussed in section 3.1.2. Solving the second equation in (F.5) leads to the optimal steady-state value for  $i_q$

$$i_q^{opt} = \frac{2\tau_M^* n_p \Psi_{PM}}{3n_p^2 \Psi_{PM}^2 + w_L n_p \omega_M k_{Fe} + w_L R_s}, \quad (\text{F.7})$$

respectively, the generated torque will be

$$\tau_M^{opt} = \frac{\tau_M^*}{1 + \frac{1}{3n_p^2 \Psi_{PM}^2} w_L (n_p \omega_M \Psi_{PM} L_q^2 + R_s)} \approx \tau_M^* \left( 1 - w_L \frac{n_p \omega_M k_{Fe} L_q^2 + R_s}{3n_p^2 \Psi_{PM}^2} \right). \quad (\text{F.8})$$

This means that the reference torque will not be fully reached, because the torque-generating current will be traded off with the losses inherited by  $i_q$ , namely the copper and hysteresis iron losses on the  $q$  axis. However, this term is small as  $(n_p \omega_M k_{Fe} L_q^2 + R_s) \ll 3n_p^2 \Psi_{PM}^2$  and as the tuning parameter  $w_L$  is also small (0.05 in this implementation), so it is disregarded in the implementation.

Furthermore, avoidance of this steady-state error is possible, one possibility is to disregard the losses on the  $q$  axis in the cost function, alternatively, as the offset is only based on known machine parameters and the measured speed, it could be compensated by adapting the torque reference.

From a steady-state point of view, the weight  $w_L$  only has a small influence. Nevertheless,  $w_L$  should be sufficiently large to avoid numerical problems, it multiplies the weight for  $i_d$  such that the cost function is convex. Furthermore, its value is important for dynamic behavior, as it trades off efficiency and dynamic response.

## F.2 Proof that the cost function remains convex after continuous parameterization with a polynomial

Given is a cost functional

$$J = \int_0^T \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt, \quad (\text{F.9})$$

with state vector  $\mathbf{x} \in \mathbb{R}^n$  and weight matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$ . Assume  $\mathbf{Q}$  is positive definite. Then,

$$\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) > 0, \quad \forall \mathbf{x}(t) \neq 0, \quad (\text{F.10})$$

and

$$J = \int_0^T \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) dt > 0, \quad \mathbf{x}(t) \neq 0 \forall t \in [0, T]. \quad (\text{F.11})$$

The inverse model replaces  $\mathbf{x}(t)$  by polynomials  $\mathbf{p}(t)$  where the coefficients are affine functions of  $\alpha$

$$J = \int_0^T \mathbf{p}^T(t) \mathbf{Q} \mathbf{p}(t) dt > 0, \quad \mathbf{x}(t) = \mathbf{p}(t) \neq 0 \forall t \in [0, T]. \quad (\text{F.12})$$

It can be shown with the Cauchy products of power series that there exist polynomials  $\mathbf{R}(T)$  with linear coefficients in  $\alpha$ , such that

$$J = \int_0^T \mathbf{p}^T(t) \mathbf{Q} \mathbf{p}(t) dt = \mathbf{R}^T(T) \mathbf{Q} \mathbf{R}(T) > 0, \quad \forall \mathbf{R}(T) \neq 0. \quad (\text{F.13})$$

These polynomials  $\mathbf{R}(T)$  can be rewritten as

$$\mathbf{R}(T) = \mathbf{A} \alpha, \quad (\text{F.14})$$

and we assume  $\text{rank}(\mathbf{A}) = n$  with  $n = \dim(\mathbf{x})$ . It follows that

$$J = \alpha^T \mathbf{A}^T \mathbf{Q} \mathbf{A} \alpha > 0, \quad \forall \mathbf{R}(T) \neq 0. \quad (\text{F.15})$$

The matrix of the parameterized cost function is  $\mathbf{K} = \mathbf{A}^T \mathbf{Q} \mathbf{A}$  and as we assumed  $\mathbf{Q}$  is positive definite we know there is a matrix  $\mathbf{B}$  such that  $\mathbf{Q} = \mathbf{B}^T \mathbf{B}$  (Cholesky decomposition). The weight matrix is then

$$\mathbf{K} = \mathbf{A}^T \mathbf{B}^T \mathbf{B} \mathbf{A} = (\mathbf{B} \mathbf{A})^T (\mathbf{B} \mathbf{A}), \quad (\text{F.16})$$

which is positive definite as any matrix  $\mathbf{K} = \mathbf{C}^T \mathbf{C}$  for some  $\mathbf{C}$  with  $\text{rank}(\mathbf{C}) = n$  is positive semidefinite [C.D. Meyer, Matrix analysis and applied linear algebra, SIAM books, 2000, p. 566]. If the weight matrix is positive definite the cost function is convex. The parameterized cost functional  $J$  is thus a convex function of the parameters  $\alpha$ . ■

### F.3 Parameterization of the constraints with a polynomial basis: Sufficient affine conditions

The polynomial

$$P(s) = \sum_{i=0}^N c_i s^i \leq 0, \quad (\text{F.17})$$

with coefficients  $c_i \in \mathbb{R}$ ,  $s \in \mathbb{R}$ , is analyzed on non-positivity over a segment  $s \in [0, 1]$ . A necessary and sufficient condition is

$$P(0) = c_0 \leq 0, \quad (\text{F.18})$$

which in the following is assumed to be satisfied. Furthermore, the  $N$  conditions

$$P\left(\frac{k}{N}\right) \leq 0, \quad k = 1..N, \quad (\text{F.19})$$

are also assumed to hold for all  $c_i$ .

These conditions can be rewritten in matrix notation

$$\mathbf{c}_0 + \mathbf{Q}\mathbf{c} \leq 0, \quad (\text{F.20})$$

with  $\mathbf{c}_0 = (c_0, \dots, c_0)^T \in \mathbb{R}^N$ ,  $\mathbf{c} = (c_1, \dots, c_N)^T \in \mathbb{R}^N$  and  $\mathbf{Q} \in \mathbb{R}^{N \times N}$  such that

$$\mathbf{Q} = (q_{ij}) = \left( \frac{\partial}{\partial c_j} P(i/N) \right) = \left( \left( \frac{i}{N} \right)^j \right), \quad (\text{F.21})$$

$$i = 1..N, j = 1..N.$$

It can be shown that  $\det(\mathbf{Q}) \neq 0$  for  $N > 0$ , and that  $\mathbf{Q}$  is positive definite. It follows that

$$\mathbf{c} \leq -\mathbf{Q}^{-1}\mathbf{c}_0, \quad (\text{F.22})$$

which can be placed into the polynomial equation

$$P(s) = c_0 + \mathbf{s}^T \mathbf{c} \leq c_0 - \mathbf{s}^T \mathbf{Q}^{-1} \mathbf{c}_0 = (-c_0)\epsilon, \quad (\text{F.23})$$

with  $\mathbf{s} = (s, \dots, s^N)^T \in \mathbb{R}^N$  and  $\epsilon = -1 + \mathbf{s}^T \mathbf{Q}^{-1} (1, \dots, 1)^T$ . As we assumed  $c_0 \leq 0$ , the upper bound of  $P(s)$  under the mentioned conditions is when  $\epsilon$  is at its maximum. It can be shown that the upper bound of  $\epsilon$ ,  $\Delta = \sup\{\epsilon\} \forall s \in [0, 1]$ , is positive



and only dependent on  $N$ , as  $\mathbf{Q}$  is known. Some values, which were computed numerically, are shown in the table below.

$N$	2	3	4	10	20
$\Delta = \sup\{\epsilon\}$	0.125	0.064	0.041	0.012	0.005

Therefore, if the conditions (F.19) hold, we have

$$P(s) \leq -\Delta P(0). \quad (\text{F.24})$$

Shifting the conditions by the constant (and negative) factor  $\Delta P(0)$ , the sufficient conditions for non-positivity of the polynomial  $P(s)$  are found:

$$P(0) \leq 0, \quad (\text{F.25})$$

$$P\left(\frac{k}{N}\right) - \Delta \cdot P(0) \leq 0, \quad k = 1..N. \quad (\text{F.26})$$

■

## F.4 Analysis: Remainder of the Euler-Lagrange equation

With the arbitrary defined torque trajectory (6.23)

$$\tau_M(t) = m_0 + (m_1 - m_0)(1 - e^{-\lambda t}), \quad (\text{F.27})$$

and the prototype flux trajectory (6.28)

$$\Psi_{rd}(t) = f_0 + (f_1 - f_0)(1 - e^{-\mu t}), \quad (\text{F.28})$$

the remainder of the Euler-Lagrange equation (6.29)

$$e(\mu, t) = k_1 \Psi_{rd}^4 - k_3 \Psi_{rd}^3 \left( \frac{d^2}{dt^2} \Psi_{rd} \right) - k_4 \tau_M^2, \quad (\text{F.29})$$

is found to be:

$$\begin{aligned} e(\mu, t) = & e^{-\mu t} f_1^3 (f_0 - f_1) (4k_1 - k_3 \mu^2) + e^{-2\mu t} f_1^2 (f_0 - f_1)^2 (6k_1 - 3k_3 \mu^2) \\ & + e^{-3\mu t} f_1 (f_0 - f_1)^3 (4k_1 - 3k_3 \mu^2) + e^{-4\mu t} (f_0 - f_1)^4 (k_1 - k_3 \mu^2) \\ & + k_1 f_1^4 - k_4 (m_0 + (m_1 - m_0)(1 - e^{-\lambda t}))^2. \end{aligned} \quad (\text{F.30})$$

Assuming that the torque is constant,  $\tau_M(t) = m_0$ , with optimal end value  $f_1 = \sqrt{m_0 \sqrt{\frac{k_4}{k_1}}}$ , it is seen that

$$e(\mu, t) = \sum_{i=1}^4 e^{-i\mu t} (f_1)^{4-i} (f_0 - f_1)^i \alpha_i, \quad (\text{F.31})$$

such that the coefficients of the time-varying functions  $e^{-\mu t}$  are all multiplied with coefficients  $\alpha_1 = 4k_1 - k_3\mu^2$ ,  $\alpha_2 = 6k_1 - 3k_3\mu^2$ ,  $\alpha_3 = 4k_1 - 3k_3\mu^2$  or  $\alpha_4 = k_1 - k_3\mu^2$ .

Therefore the optimal eigenvalue  $\mu$  could be between  $\sqrt{\frac{k_1}{k_3}}$  and  $2\sqrt{\frac{k_1}{k_3}}$ . However it is unclear where this eigenvalue is, the coefficients depend on both  $f_0 - f_1$ , the difference of the actual to the optimal flux, and on  $f_1$ , the desired final value for the flux. This nonlinearity prevents finding an exact solution without a numerical search procedure, even for the simple case of constant torque. Anyhow find that if  $\tau_M(t)$  is constant, the optimal eigenvalue is

$$\sqrt{\frac{k_1}{k_3}} < \mu < 2\sqrt{\frac{k_1}{k_3}}. \quad (\text{F.32})$$

In the simplest case of zero torque  $\tau_M(t) = m_0 = 0 \forall t$ , the Euler-Lagrange equation (6.24) simplifies to the homogenous linear equation of second order

$$\frac{d^2}{dt^2} \Psi_{rd}(t) = \frac{k_1}{k_3} \Psi_{rd}(t) \quad (\text{F.33})$$

with  $\Psi_{rd}(0) \geq 0$  and  $\Psi_{rd}(T) = 0$ . Then it is clear that the solution is  $\Psi_{rd}(t) = \Psi_{rd}(0) e^{-\sqrt{\frac{k_1}{k_3}} t}$ , the optimal eigenvalue is analytically found as

$$\mu = \sqrt{\frac{k_1}{k_3}}. \quad (\text{F.34})$$

These results again show that an analytical solution of the boundary value problem (6.24) can be found if the torque can be assumed as constant [dWR99], however, the general time-varying problem is hard to solve. The underlying results serve as initial guess for a numerical search of an approximate solution.

# APPENDIX G

---

## Related works

---

### G.1 Advised student theses

In chronological order:

- F. Birnkammer, *Minimum-Zeit Stromregelung für Gleichstrom- und Synchronmaschinen*, Bachelor's thesis, 2013.
- R. Dost, *Laufzeitanalyse der Rechenalgorithmen einer modellprädiktiven Regelung*, Studienarbeit, 2012.
- A. Dötlinger, *Dynamic Optimization of an Induction Machine*, Diploma thesis, 2011. Awarded with the 'Studienpreis der SEW-Eurodrive-Stiftung' 2012.
- R. Leibbrandt, *Nonlinear Model Predictive Control for Induction Machines*, Bachelor's thesis, 2011.
- V. Çiftçibasi, *Speed-Sensorless Control of Induction Motors using MRAS Observers*, Bachelor's thesis, 2011.
- H. Sado-Kamgaing, *Optimierung einer Asynchronmaschine*, Studienarbeit, 2010.
- J. Jung and A. Dötlinger, *Implementierung einer MPC für eine Synchronmaschine*, Lab Project, 2010.

- A. Dötlinger, *Predictive Control of a PMSM based on Real-time Dynamic Optimization*, Bachelor's thesis, 2010.
- P. Putzer, G. Walder, M. Campestrini, *Field-oriented Control with a DSP*, Lab Project, 2010.

## G.2 List of publications

The following list includes all peer-review publications, written during the time of the author's studies at the institute.

- J-F. Stumper, A. Doetlinger and R. Kennel, Loss minimization of induction machines in dynamic operation, *IEEE Transactions on Energy Conversion*, accepted for publication, 2013.
- D. Paulus, J-F. Stumper and R. Kennel, Sensorless Control of Synchronous Machines based on Direct Speed and Position Estimation in Polar Stator-Current Coordinates, *IEEE Transactions on Power Electronics*, Vol. 28, No. 5, pp. 2503-2513, 2013.
- J-F. Stumper, E. Fuentes, S. Kuehl and R. Kennel, Predictive Torque Control for AC Drives: Analysis and Improvement of Parametric Robustness, *Proc. Energy Conversion Congress and Exposition (ECCE)*, Denver, 2013.
- J-F. Stumper, V. Hagenmeyer, S. Kuehl and R. Kennel, Flatness-based Dead-beat Control revisited: Robust and High-performance Design for Electrical Drives, *Proc. American Control Conference (ACC)*, Washington DC, 2013.
- J-F. Stumper and R. Kennel, Real-time Dynamic Efficiency Optimization for Induction Machines, *Proc. American Control Conference (ACC)*, Washington DC, 2013.
- J-F. Stumper, A. Doetlinger and R. Kennel, Classical Model Predictive Control of a Permanent Magnet Synchronous Motor, *EPE Journal (European Power Electronics and Drives)*, invited paper, Vol. 22, No. 3, pp. 24-31, 2012.
- J-F. Stumper, D. Paulus and R. Kennel, A Nonlinear Estimator for Dynamical and Robust Sensorless Control of Permanent Magnet Synchronous Machines, *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, pp. 922-927, 2011.

- J-F. Stumper, A. Doetlinger, J. Jung and R. Kennel, Predictive Control of a Permanent Magnet Synchronous Machine based on Real-Time Dynamic Optimization, *Proc. European Conference on Power Electronics and Applications (EPE - ECCE Europe)*, Birmingham, paper 99, 2011.
- J-F. Stumper, D. Paulus and R. Kennel, Encoderless Field-Oriented Control of a Synchronous Reluctance Machine with an Estimator in Polar Stator-Current Coordinates, *Proc. European Conference on Power Electronics and Applications (EPE - ECCE Europe)*, Birmingham, paper 106, 2011.
- J-F. Stumper and R. Kennel, Field-Oriented Control of a Speed-Sensorless Induction Machine for the Complete Speed Range with a Nonlinear Observer, *Proc. IEEE International Symposium on Sensorless Control for Electrical Drives (SLED)*, Birmingham, pp. 107-113, 2011.
- J-F. Stumper and R. Kennel, Computationally Efficient Trajectory Optimization for Linear Control Systems with Input and State Constraints, *Proc. American Control Conference (ACC)*, San Francisco, pp. 1904-1909, 2011.
- J-F. Stumper, S. Kuehl and R. Kennel, Robust Deadbeat Control for Synchronous Machines rejecting Noise and Uncertainties by Predictive Filtering, *Proc. International Conference on Power Electronics (ICPE - ECCE Asia)*, Jeju, pp. 1378-1385, 2011.
- D. Paulus, J-F. Stumper, P. Landsmann and R. Kennel, Encoderless Field-Oriented Control of a Synchronous Reluctance Machine with Position and Inductance Estimators, *Proc. International Conference on Power Electronics (ICPE - ECCE Asia)*, Jeju, pp. 1153-1160, 2011.
- J-F. Stumper and R. Kennel, Inversion of Linear and Nonlinear Observable Systems with Series-defined Output Trajectories, *Proc. IEEE Multi-Conference on Systems and Control (MSC), International Symposium on Computer-Aided Control System Design (CACSD)*, Yokohama, pp. 1993-1998, 2010.
- J-F. Stumper, D. Paulus, P. Landsmann and R. Kennel, Encoderless Field-oriented Control of a Synchronous Reluctance Machine with a Direct Estimator, *Proc. IEEE International Symposium on Sensorless Control for Electrical Drives (SLED)*, Padova, pp. 18-23, 2010.
- D. Paulus, J-F. Stumper, P. Landsmann and R. Kennel, Robust Encoderless Speed Control of a Synchronous Machine by direct Evaluation of the Back-EMF Angle without Observer, *Proc. IEEE International Symposium on Sensorless Control for Electrical Drives (SLED)*, Padova, pp. 8-13, 2010.

- J-F. Stumper, F. Svaricek and R. Kennel, Trajectory Tracking Control with Flat Inputs and a Dynamic Compensator, *Proc. European Control Conference (ECC)*, Budapest, pp. 248-253, 2009.

---

## Bibliography

---

- [Abr00] F. Abrahamsen. *Energy optimal control of induction motor drives*. Ph.D. thesis, Aalborg University, 2000.
- [Ack88] J. Ackermann. *Abtastregelungen*. Berlin: Springer, 1988.
- [AD06] F. Antritter and J. Deutscher. Nonlinear tracking control of a DC motor via a boost-converter using linear dynamic output feedback. In *Proc. IEEE Conf. Decision & Control*, pages 6587–6592, 2006.
- [ANK10] N.A. Ameen, A.A. Naassani, and R. Kennel. Design of a digital system dedicated for electrical drive applications. In *EPE J., Vol. 20, No. 4*, pages 37–44, 2010.
- [BBK92] L. Ben-Brahim and A. Kawamura. Digital control of induction motor current with deadbeat response using predictive state observer. In *IEEE Trans. Power Electron., Vol. 54, No. 2*, pages 551–559, 1992.
- [BBPZ08] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto. Combined speed and current model predictive control with inherent field-weakening features for PMSM drives. In *Proc. of the IEEE Med. Electrotechnical Conf.*, pages 472–478, 2008.
- [BBPZ09] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto. Design and implementation of model predictive control for electrical motor drives. In *IEEE Trans. Ind. Electron.*, pages 1925–1936, 2009.
- [BCN95] M. Bodson, J.N. Chiasson, and R.T. Novak. A systematic approach to selecting flux references for torque maximization in induction motors. In *IEEE Trans. Control Sys. Tech., Vol. 3, No. 4*, pages 388–397, 1995.

- [BKKP11] S. Bolognani, R. Kennel, S. Kuehl, and G. Paccagnella. Speed and current model predictive control of an ipm synchronous motor drive. In *Proc. IEEE Int. Electric Machines and Drives Conf.*, pages 1597–1602, 2011.
- [BLNH05] G.H. Bode, P.C. Loh, M.J. Newman, and D.G. Holmes. Improved robust predictive current regulation algorithm. In *IEEE Trans. Ind. Appl.*, Vol. 41, No. 6, pages 1720–1733, 2005.
- [BS97] R.E. Brown and M.A. Stone. On the use of polynomial series with the Rayleigh-Ritz method. In *Composite Structures*, Vol. 39, No. 3–4, pages 191–196, 1997.
- [CKK<sup>+</sup>08] P. Cortés, M. Kazmierkowski, R. Kennel, D. Quevedo, and J. Rodríguez. Predictive control in power electronics and drives. In *IEEE Trans. Ind. Electron.*, Vol. 55, No. 2, pages 4312–4324, 2008.
- [CKKY98] S.K. Chung, H.S. Kim, C.G. Kim, and M.J. Youn. A new instantaneous torque control of PM synchronous motor for high-performance direct-drive applications. In *IEEE Trans. Power Electron.*, Vol. 13, No. 3, pages 388–400, 1998.
- [CKS95] J.W. Choi, H.W. Kim, and S.K. Sul. New current control concept—minimum time current control in the three-phase PWM converter. In *IEEE Power Electron. Spec. Conf.*, pages 332–338, 1995.
- [CS98] J.W. Choi and S.K. Sul. Generalized solution of minimum time current control in three-phase balanced systems. In *IEEE Trans. Ind. Electron.*, Vol. 45, No. 5, pages 738–744, 1998.
- [CTM<sup>+</sup>05] C. Cavallaro, A.O. Di Tommaso, R. Micelli, A. Raciti, G.R. Galluzzo, and M. Trapanese. Efficiency enhancement of permanent-magnet synchronous motor drives by online loss minimization approaches. In *IEEE Trans. Ind. Electron.*, Vol. 52, No. 4, pages 1153–1160, 2005.
- [DLO01] E. Delaleau, J.P. Louis, and R. Ortega. Modeling and control of induction motors. In *Int. J. Appl. Math. Comput. Sci.*, Vol. 11, No. 1, pages 105–129, 2001.
- [DS04] E. Delaleau and A.M. Stanković. Flatness-based hierarchical control of the PM synchronous motor. In *Proc. of the American Control Conf.*, pages 65–70, 2004.



- [dWR99] C. Canudas de Wit and J. Ramírez. Optimal torque control for current-fed induction motors. In *IEEE Trans. Automat. Control*, Vol. 44, No. 5, pages 1084–1089, 1999.
- [FBB12] H. Ennadifi F. Becker and M. Braun. Straightforward current control – one step controller based on current slope detection. In *EPE J.*, Vol. 22, No. 2, pages 13–21, 2012.
- [FLMR92] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. On differentially flat nonlinear systems. In *Proc. IFAC-Symp. Nonlinear Control Systems*, pages 408–412, 1992.
- [FLMR95] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. In *Int. J. of Control*, Vol. 61, No. 6, pages 1327–1361, 1995.
- [FM00a] M. Fliess and R. Marquez. Continuous-time linear predictive control and flatness: a module-theoretic setting with examples. In *Int. J. of Control*, Vol. 73, No. 7, pages 606–623, 2000.
- [FM00b] M. Fliess and R. Marquez. Towards a module theoretic approach to discrete time linear predictive control. In *Proc. Int. Symp. Math. Theory of Networks and Systems*, 2000.
- [Föl85] D. Föllinger. *Regelungstechnik*. Heidelberg: Hüthig, 1985.
- [Gen08] A. Gensior. *Beiträge zur flachheitsbasierten Folgeregelung leistungselektronischer Systeme*. Dissertation, Dresden: Technische Universität, 2008.
- [GF06] M. Guay and J.F. Forbes. Real-time dynamic optimization of controllable linear systems. In *J. of Guidance, Control and Dynamics*, Vol. 29, No. 4, pages 929–935, 2006.
- [GGPL98] A. Ganji, P. Guillaume, R. Pintelon, and P. Lataire. Induction motor dynamic and static inductance identification using a broadband excitation technique. In *IEEE Trans. Energy Convers.*, Vol. 13, No. 1, pages 15–20, 1998.
- [GLSW94] G.O. Garcia, J.C.M. Luis, R.M. Stephan, and E.H. Watanabe. An efficient controller for an adjustable speed induction motor drive. In *IEEE Trans. Ind. Electron.*, Vol. 41, No. 5, pages 533–539, 1994.

- [GP09] K. Graichen and N. Petit. Incorporating a class of constraints into the dynamics of optimal control problems. In *Optim. Control Appl. Meth.*, Vol. 30, No. 6, pages 537–561, 2009.
- [HD03] V. Hagenmeyer and E. Delaleau. Exact feedforward linearization based on differential flatness. In *Int. J. of Control*, Vol. 76, No. 6, pages 537–556, 2003.
- [HD08] V. Hagenmeyer and E. Delaleau. Continuous-time non-linear flatness-based predictive control: an exact feedforward linearisation setting with an induction drive example. In *Int. J. of Control*, Vol. 81, No. 10, pages 1645–1663, 2008.
- [HL03] D. Henrion and J.B. Lasserre. LMIs for constrained polynomial interpolation with application in trajectory planning. In *Systems & Control Letters*, Vol. 55, No. 6, pages 473–477, 2003.
- [HRD03] V. Hagenmeyer, A. Ranftl, and E. Delaleau. Flatness-based control of the induction drive minimising energy dissipation. In *Nonlinear and Adaptive Control*, A. Zinober and D. Owens (eds), LNCIS Vol. 281, pages 149–160, 2003.
- [HS96] J. Holtz and L. Springob. Identification and compensation of torque ripple in high-precision permanent magnet motor drives. In *IEEE Trans. Ind. Electron.*, Vol. 43, No. 2, pages 309–320, 1996.
- [JP97] J.N. Juang and M. Phan. *Deadbeat predictive controllers*. NASA Technical Memorandum No. 112862, 1997.
- [KG83] A. Kusko and D. Galler. Control means for minimization of losses in AC and DC motor drives. In *IEEE Trans. Ind. Appl.*, Vol. 19, No. 4, pages 561–570, 1983.
- [KHK92] G.S. Kim, I.J. Ha, and M.S. Ko. Control of induction motors for both high dynamic performance and high power efficiency. In *IEEE Trans. Ind. Electron.*, Vol. 39, No. 4, pages 323–333, 1992.
- [KKM03] F. Khorrami, P. Krishnamurthy, and H. Melkote. *Modeling and adaptive nonlinear control of electric motors*. Berlin: Springer Verlag, 2003.

- [KL03] B.H. Kenny and R.D. Lorenz. Stator- and rotor-flux-based deadbeat direct torque control of induction machines. In *IEEE Trans. Ind. Appl.*, Vol. 39, No. 4, pages 1093–1101, 2003.
- [KMY90] T. Kawabata, T. Miyashita, and Y. Yamamoto. Dead beat control of three phase PWM inverter. In *IEEE Trans. Power Electron.*, Vol. 7, No. 3, pages 21–28, 1990.
- [KNL85] D.S. Kirschen, D.W. Novotny, and T.A. Lipo. On-line efficiency optimization of a variable frequency induction motor drive. In *IEEE Trans. Ind. Appl.*, Vol. 21, No. 4, pages 610–616, 1985.
- [KP02] A. Kilthau and M. Pacas. Appropriate models for the control of the synchronous reluctance machine. In *Proc. IEEE Ind. Appl. Conf.*, pages 2289–2295, 2002.
- [Kuk96] O. Kukrer. Discrete-time current control of voltage-fed three-phase PWM inverters. In *IEEE Trans. Power Electron.*, Vol. 11, No. 2, pages 260–269, 1996.
- [KY01] K.-H. Kim and M.J. Youn. A simple and robust digital current control technique of a pm synchronous motor using time delay control approach. In *IEEE Trans. Power Electron.*, Vol. 16, No. 1, pages 72–82, 2001.
- [Lee11] J.H. Lee. Model predictive control: Review of the three decades of development. In *Int. J. Control, Automation and Systems*, Vol. 9, No. 3, pages 415–424, 2011.
- [Leo74] W. Leonhard. *Regelung in der elektrischen Antriebstechnik*. Stuttgart: B.G. Teubner, 1974.
- [Lév09] J. Lévine. *Analysis and control of nonlinear systems: A flatness-based approach*. Mathematical engineering. Berlin: Springer, 2009.
- [LK05] A. Linder and R. Kennel. Model predictive control for electrical drives. In *Proc. Power Electron. Spec. Conf. (PESC)*, pages 1793–1799, 2005.
- [LKKS10] A. Linder, R. Kanchan, R. Kennel, and P. Stolze. *Model-based predictive control of electric drives*. Göttingen: Cuvillier, 2010.

- [LML03] S. Liu, P. Mercorelli, and K. Lehman. Robust control of a permanent magnet linear motor in the presence of large disturbances. In *Proc. IEEE Conf. Control Appl.*, pages 1301–1306, 2003.
- [LN04] S. Lim and K. Nam. Loss-minimising control scheme for induction motors. In *IEE Proc. Electr. Power Appl.*, Vol. 151, No. 4, pages 385–397, 2004.
- [LY92a] R.D. Lorenz and S.M. Yang. AC induction servo sizing for motion control applications via loss minimizing real-time flux control. In *IEEE Trans. Ind. Appl.*, Vol. 28, No. 3, pages 589–593, 1992.
- [LY92b] R.D. Lorenz and S.M. Yang. Efficiency-optimized flux trajectories for closed-cycle operation of field-orientation induction machine drives. In *IEEE Trans. Ind. Appl.*, Vol. 28, No. 3, pages 574–580, 1992.
- [MCP07] R. Morales-Caporal and M. Pacas. A predictive torque control for the synchronous reluctance machine taking into account the magnetic cross saturation. In *IEEE Trans. Ind. Electron.*, Vol. 7, No. 3, pages 1161–1167, 2007.
- [MES07] Y.A.I. Mohamed and E.F. El-Saadany. An improved deadbeat current control scheme with a novel adaptive self-tuning load model for a three-phase PWM voltage-source inverter. In *IEEE Trans. Ind. Electron.*, Vol. 54, No. 2, pages 747–759, 2007.
- [MKM04] C. Mademlis, I. Kioskeridis, and N. Margaris. Optimal efficiency control strategy for interior permanent-magnet synchronous motor drives. In *IEEE Trans. Energy Convers.*, Vol. 19, No. 4, pages 715–723, 2004.
- [MKY03] H.T. Moon, H.S. Kim, and M.J. Youn. A discrete-time predictive current control for PMSM. In *IEEE Trans. Power Electron.*, Vol. 18, No. 1, pages 464–472, 2003.
- [ML92] J.C. Moreira and T.A. Lipo. Modeling of saturated ac machines including air gap flux harmonic components. In *IEEE Trans. Ind. Appl.*, Vol. 28, No. 2, pages 343–349, 1992.
- [ML99] M. Morari and J.H. Lee. Model predictive control: Past, present and future. In *Comp. & Chem. Eng.*, Vol. 23, pages 667–682, 1999.

- [MMB99] L. Malesani, P. Mattavelli, and S. Buso. Robust dead-beat current control for PWM rectifiers and active filters. In *IEEE Trans. Ind. Appl.*, Vol. 35, No. 3, pages 613–620, 1999.
- [MR96a] P. Martin and P. Rouchon. Flatness and sampling control of induction motors. In *Proc. IFAC World Congress*, pages 389–394, 1996.
- [MR96b] P. Martin and P. Rouchon. Two remarks on induction motors. In *Proc. CESA'96 IMACS Multiconference*, pages 76–79, 1996.
- [MR00] P. Martin and P. Rouchon. Two simple flux observers for induction motors. In *Int. J. Adaptive Control and Signal Processing*, Vol. 14, No. 2, pages 171–175, 2000.
- [MXM00] C. Mademlis, J. Xypteras, and N. Margaris. Loss minimization in surface permanent-magnet synchronous motor drives. In *IEEE Trans. Ind. Electron.*, Vol. 47, No. 1, pages 115–122, 2000.
- [MYKT99] K. Matsuse, T. Yoshizumi, S. Katsuta, and S. Taniguchi. High-response flux control of direct-field-oriented induction motor with high efficiency taking core loss into account. In *IEEE Trans. Ind. Appl.*, Vol. 35, No. 1, pages 62–69, 1999.
- [Nih07] M. Nihtilä. Discrete signal processing with flat system models. In *Proc. IEEE Int. Conf. Signal Process. and Comm.*, pages 636–639, 2007.
- [Nus99] U. Nuss. Umrichter mit maximaler Regeldynamik für die Prüfstandstechnik. In *Antriebstechnik*, Vol. 38, No. 2, pages 28–31, 1999.
- [OS02] R. Ottersen and J. Svensson. Vector current controlled voltage source converter - deadbeat control and saturation strategies. In *IEEE Trans. Power Electron.*, Vol. 17, No. 2, pages 279–285, 2002.
- [Pie69] D.A. Pierre. *Optimization theory with applications*. New York: Wiley, 1969.
- [PLR05] G. Pannocchia, N. Laachi, and J.B. Rawlings. A candidate to replace PID control: SISO-constrained LQ control. In *AICHE J.*, Vol. 51, No. 4, pages 1178–1189, 2005.

- [PTVF92] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C: The art of scientific computing, 2nd ed.* Cambridge: Cambridge University Press, 1992.
- [PW05] M. Pacas and J. Weber. Predictive direct torque control for the PM synchronous machine. In *IEEE Trans. Ind. Electron.*, Vol. 52, No. 5, pages 1350–1356, 2005.
- [QD08] N.P. Quang and J.A. Dittrich. *Vector control of three-phase AC Machines: System development in the practice.* Berlin: Springer Verlag, 2008.
- [Qua93] N.P. Quang. *Praxis der feldorientierten Drehstromantriebsregelungen.* Böblingen: expert Verlag, 1993.
- [RCA01] F. Rotella, F.J. Carrillo, and M. Ayadi. Digital flatness-based robust controller applied to a thermal process. In *Proc. IEEE Int. Conf. Control Appl.*, pages 936–941, 2001.
- [RKE<sup>+</sup>12] J. Rodríguez, R. Kennel, J.R. Espinoza, M. Trincado, C.A. Silva, and C.A. Rojas. High-performance control strategies for electrical drives: An experimental assessment. In *IEEE Trans. Ind. Electron.*, Vol. 59, No. 2, pages 812–820, 2012.
- [RMM10] S. Richter, S. Mariéthoz, and M. Morari. High-speed online MPC based on a fast gradient method applied to power converter control. In *Proc. of the American Control Conf.*, pages 4737–4743, 2010.
- [RR00] C.V. Rao and J.B. Rawlings. Linear programming and model predictive control. In *Int. J. Process Control*, Vol. 10, No. 2&3, pages 283–289, 2000.
- [Sch09] D. Schröder. *Elektrische Antriebe: Regelung von Antriebssystemen.* Berlin: Springer, 2009.
- [SDP96] D. Chen S. Devasia and B. Paden. Nonlinear inversion-based output tracking. In *IEEE Trans. Automatic Control*, Vol. 41, No. 7, pages 930–942, 1996.
- [SH98] L. Springob and J. Holtz. High-bandwidth current control for torque-ripple compensation in PM synchronous machines. In *IEEE Trans. Ind. Electron.*, Vol. 45, No. 5, pages 713–721, 1998.

- [SL91] J.J. Slotine and W. Li. *Applied nonlinear control*. New Jersey: Prentice-Hall, 1991.
- [SRA04] H. Sira-Ramírez and S.K. Agrawal. *Differentially flat systems*. Control engineering series. New York: Marcel Dekker, 2004.
- [TF11] S. Thomsen and F. Fuchs. Design and analysis of a flatness-based control approach for speed control of drive systems with elastic couplings and uncertain loads. In *Proc. European Conf. on Power Electron. and Appl.*, 2011.
- [USU03] N. Urasaki, T. Senjyu, and K. Uezato. A novel calculation method for iron loss resistance suitable in modeling permanentmagnet synchronous motors. In *IEEE Trans. Energy Convers.*, Vol. 18, No. 1, pages 41–47, 2003.
- [vL02] J. von Löwis. *Flachheitsbasierte Trajektorienfolgeregelung elektromechnischer Systeme*. Dissertation, Dresden: Technische Universität, 2002.
- [VL03] S.N. Vukosavić and E. Levi. A method for transient torque response improvement in optimum efficiency induction motor drives. In *IEEE Trans. Energy Convers.*, Vol. 18, No. 4, pages 484–493, 2003.
- [vNM98] M. van Nieuwstadt and R.M. Murray. Real-time trajectory generation for differentially flat systems. In *Int. J. of Robust and Nonlinear Control*, Vol. 8, pages 995–1020, 1998.
- [Wal01] O. Wallmark. *Control of a permanent magnet synchronous motor with non-sinusoidal flux density distribution*. Master Thesis, Chalmers University of Technology, Sweden, 2001.
- [WB08] J. Weigold and M. Braun. Predictive current control using identification of current ripple. In *IEEE Trans. Ind. Electron.*, Vol. 55, No. 12, pages 4346–4353, 2008.
- [WB10] Y. Wang and S. Boyd. Fast model predictive control using online optimization. In *IEEE Trans. Control Sys. Tech.*, pages 267–278, 2010.
- [YL02] S.M. Yang and C.H. Lee. A deadbeat current controller for field oriented induction motor drives. In *IEEE Trans. Power Electron.*, Vol. 17, No. 5, pages 772–778, 2002.