

Electrical
Engineering

Arne Linder

Modellbasierte Prädiktivregelung in der Antriebstechnik

模型预测控制与电力驱动技术

Model-Based Predictive Control of Electric Drives

Edited by Ralph Kennel and Marcelo Lobo Heldwein,
translated by Qing Chen, Rahul Kanchan and Peter Stolze

Modellbasierte Prädiktivregelung in der Antriebstechnik

Arne Linder

模型预测控制与电力驱动技术

陈庆 译

Model-Based Predictive Control of Electric Drives

Translation by Rahul Kanchan and Peter Stolze

Herausgeber/Editors/作品法人:

Prof. i. R. Dr.-Ing. Dr. h. c. Ralph Kennel, Fellow IET, Senior Member IEEE

[德] 拉尔夫·肯内尔 教授

Prof. Dr.sc. ETH Zürich Marcelo Lobo Heldwein, Senior Member IEEE

Chair of High-Power Converter Systems, TUM School of Engineering and Design
Technical University of Munich

[巴西] 马塞洛·罗伯·黑尔德魏因 教授

Autoren/Authors/作者:

Dr.-Ing. Arne Linder

Product Manager Drives Industrial Automation
Kollmorgen Europe GmbH, Ratingen, Germany

[德] 阿尔内·林德 博士

M.Sc. Qing Chen 陈庆

Chair of High-Power Converter Systems, TUM School of Engineering and Design
Technical University of Munich

Dr. Rahul Kanchan

Principal Scientist

ABB Corporate Research Center, Västerås, Sweden

[印度] 拉胡尔·坎禅 博士

Dr.-Ing. Peter Stolze

MAN Energy Solutions SE, Augsburg, Germany

[德] 彼得·施托尔策 博士

The German National Library has registered this publication in the German National Bibliography. Detailed bibliographic data are available on the Internet at <https://portal.dnb.de>.

Imprint

1. Edition

Copyright © 2022 TUM.University Press

Copyright © 2022 Arne Linder, Qing Chen, Rahul Kanchan, Peter Stolze

All rights reserved

Layout design and typesetting: Qing Chen

Cover design: Caroline Ennemoser

Cover illustration: Arne Linder

New and translated edition of the dissertation of the same title, which was published by Logos Verlag Berlin in 2006. The English edition was first published by Cuvillier Verlag Göttingen in 2010.

TUM.University Press

Technical University of Munich

Arcisstrasse 21

80333 Munich

Germany

DOI: 10.14459/2023md1703047

ISBN printed edition: 978-3-95884-080-5

www.tum.de

Unserem hochverehrten Lehrer
Dedicated to our esteemed teacher
献给我们尊敬的老师



Herrn Prof. i. R. Dr.-Ing. Ralph Kennel
anlässlich seiner Emeritierung im Jahr 2022 gewidmet
on the occasion of his retirement in 2022
值其于2022年退休之际

Vorwort zur dreisprachigen Ausgabe/三语版 书籍序言/Preface to the trilingual edition

- Das vorliegende dreisprachige wissenschaftliche Buch wurde durch die dreisprachige Ausgabe eines botanischen Buches inspiriert, die vom Taschen Verlag in Deutschland unter dem Titel „Ein Garten Eden. Meisterwerke der botanischen Illustration“ herausgegeben wurde.
- Die deutsche Ausgabe des vorliegenden Buches wurde 2006 vom Logos Verlag, Berlin, veröffentlicht, deren Inhalt auf einer von Dr. Arne Linder verfassten Dissertation basiert. Die englische Ausgabe wurde 2010 vom Cuvillier Verlag, Göttingen, veröffentlicht. Die Übersetzer sind Dr. Rahul Kanchan und Dr. Peter Stolze.
- Das vorliegende Buch wurde von Qing Chen, einem Doktoranden der Technischen Universität München, auf Grundlage von Vorschlägen von Prof. Kennel, Prof. Heldwein, Dr. Linder, Dr. Kanchan und Dr. Stolze gestaltet. Darüber hinaus wurden die chinesischen Inhalte von Qing Chen aus dem Deutschen übersetzt.
- Die vorliegende dreisprachige Ausgabe ist 456 Seiten lang, während die deutsche Ausgabe 259 Seiten und die englische Ausgabe 269 Seiten umfasst.
- Da es sich bei dem Inhalt dieses Buches um ein technisches Thema handelt, enthält es viele Abbildungen, Tabellen und mathematische Formeln. Um diese nicht unnötig oft zu wiederholen, erscheinen nicht alle Abbildungen, Tabellen und mathematische Formeln im englischen und chinesischen Text.
- Aus ästhetischen und typografischen Gründen erscheint der englische Text, der auf die Seiten mit geraden Seitenzahlen gehört, gelegentlich auf Seiten mit ungeraden Seitenzahlen. In diesen Fällen wurde eine horizontale Linie zur Abgrenzung des deutschen und englischen Textes verwendet. Um die Kontinuität des Inhalts des Glossars zu gewährleisten, erscheinen dort in Anhang B zuerst alle Begriffe in deutscher Sprache, und dann in englischer und chinesischer Übersetzung.
- Vielleicht liegt es an der Unkenntnis der Autoren, dass sie noch nicht auf ähnliche dreisprachige Ausgabe von wissenschaftlichen Büchern gestoßen sind. Dies ist ein neues Experiment: ein Versuch, wissenschaftliche Inhalte wie künstlerische, in drei unterschiedlichen Sprachen verfassten Bücher zu gestalten.

-
- 此三语版科学性书籍受德国塔森出版社发行的三语版植物学书籍的启发，其德文名称为《Ein Garten Eden. Meisterwerke der botanischen Illustration》。
 - 本书德文版于 2006 年由德国柏林逻辑出版社发行，其内容基于 Arne Linder 博士撰写的博士论文。本书英文版于 2010 年由德国哥廷根库维利耶出版社发行，英文译者为 Rahul Kanchan 博士和 Peter Stolze 博士。
 - 此三语版本书籍的设计工作由慕尼黑工业大学博士研究生陈庆完成，并根据 Ralph Kennel 教授, Marcelo Heldwein 教授, Linder 博士, Kanchan 博士和 Stolze 博士的建议进行适当修改。此外，中文版内容由陈庆基于德文版翻译。
 - 此三语版本科学性书籍的篇幅为 456 页，而前面所提及的德文版篇幅为 259 页；英文版篇幅为 269 页。
 - 鉴于本书讨论的是技术性主题，所以不可避免地包含诸多图表与数学公式。为了避免图表和数学公式不必要的重复出现，英文和中文部分并未全部给出。
 - 基于审美及排版的需要，应属于偶数页的英文部分会偶尔出现在奇数页，此时在德文和英文部分之间使用长横线作为分割。为了保证附录部分的连续性，首先给出附录 B 的德语术语表，然后依次是英文和中文翻译部分。
 - 也许是作者孤陋寡闻，还未碰到过类似的三语版科学性书籍。这是一个新的尝试：类似于以三语版方式呈现的艺术性书籍，本书以三语版呈现科学性的内容。

-
- This trilingual scientific book was inspired by the trilingual edition of a book on botany published by Taschen Verlag in Germany under the title "Ein Garten Eden. meisterwerke der botanischen Illustration".
 - The German edition of this book was published in 2006 by Logos Verlag, Berlin, whose content is based on the dissertation written by Dr. Arne Linder. The English edition of this book was published in 2010 by Cuvillier Verlag, Göttingen. The translators are Dr. Rahul Kanchan and Dr. Peter Stolze.
 - The design of this trilingual book was done by Chen Qing, a PhD student at the Technical University of Munich, based on the suggestions of Prof. Kennel, Prof. Heldwein, Dr. Linder, Dr. Kanchan and Dr. Stolze. In addition, the Chinese content was translated from German by Qing Chen.
 - This trilingual book is 456 pages long, while the German version is 259 pages long; the English version is 269 pages long.

-
- Since the content of this book is about technical subject, it contains many figures, tables and mathematical formulas. In order not to repeat them unnecessarily, therefore, not all figures, tables and mathematical formulas will be appear in the English and Chinese text.
 - For aesthetic and typographical reasons, some English texts, which belong to pages with even page numbers, occasionally appear on pages with odd page numbers. In these cases, a horizontal line has been used to delineate the German and English text. To ensure continuity of content in the glossary, all terms appear there in Appendix B first in German, and then in English and Chinese translations.
 - Perhaps it is the authors ignorance that they have not yet encountered a similar trilingual edition of scientific book. This is a new experiment: an attempt to make scientific content look like artistic books written in three different languages.

München, im Herbst 2022/2022 年秋, 于慕尼黑/Munich, autumn 2022
Ralph Kennel (拉尔夫·肯内尔)
Marcelo Lobo Heldwein (马塞洛·罗伯·黑尔德魏因)

Preface to the english edition

This book is a translation of the PhD thesis submitted by Arne Linder to the Department of Electrical, Information and Printing Technology at Wuppertal University in 2005. Dr. Arne Linder showed in his outstanding work for the first time that in general model-based predictive control of electrical drives, even with prediction horizons greater than one sampling cycle is possible, despite the high sampling rates which are necessary in drive control.

The presented work shall give a general overview about the predictive control methods used so far in drive control as well as a classification of them. Furthermore, a family tree of predictive control methods in drive technology is presented which also shows relations between them, derivations and further development of certain methods. Generalized Predictive Control (GPC) is also introduced, first the two PI current controllers in a field-oriented control scheme of an induction machine are replaced by two single GPC controllers and their results are then compared to the ones of classical PI controllers. In a next step, the speed controller is replaced by a GPC controller. Different types of models, methods for handling nonlinearities and model discretization are also discussed in this work. The ability of GPC to provide multivariable control is examined, using one single controller for both flux- and torque-producing currents. Moreover, possibilities for a direct inverter control method which makes the modulator needless are presented as well: The chances of this approach, problems but also possible remedies are discussed in detail. As a conclusion, further control methods which are partly considered to be predictive are discussed. The sections contain not only simulation, but also experimental results which clearly prove the applicability of the presented control methods for electrical drives. I would like to thank Dr. Rahul Kanchan, Dipl.-Ing. Peter Stolze and Dipl.-Ing. (syr.) Nael Al-Sheakh Ameen for the translation of this work. M.Sc. Juan Carlos Ramirez Martinez, M.Sc. Bahaa Galal and M.A. Johanna Schmidt helped to finalize this work by proofreading the English translation.

Munich, June 2010
Ralph Kennel

Vorwort zur ersten deutschen Auflage

In der Antriebstechnik werden bisher fast ausschließlich lineare PI-Regler eingesetzt. Diverse prädiktive Regelverfahren werden zwar seit längerem diskutiert, die Prädiktion erstreckt sich aber bis auf wenige Ausnahmen nur auf den nächsten Abtastschritt. Ziel dieser Arbeit ist es, die aus der klassischen Regelungstechnik bekannten und in der Verfahrenstechnik mit großem Erfolg angewandten sogenannten *Modellbasierten Prädiktivregler* mit ihrem hohen Prädiktionshorizont für ihre Eignung in der Antriebstechnik zu untersuchen. Es werden verschiedene Strategien vorgestellt, erläutert und bewertet, wobei die vorliegende Arbeit dem interessierten Leser gleichzeitig eine Hilfe bei der Implementierung dieser Verfahren sein soll. Da modellbasierte Prädiktivverfahren in der Antriebstechnik bisher kaum bekannt sind, umfasst diese Dissertation auch ausführliche Herleitungen der einzelnen Regelalgorithmen.

Selbstverständlich wäre diese Arbeit ohne die tatkräftige Hilfe anderer nicht oder zumindest nicht in diesem Umfang möglich gewesen. Daher möchte ich mich an dieser Stelle bei all denen bedanken, die mir während meiner Tätigkeit in Wuppertal mit Rat und Tat zur Seite gestanden haben.

Zuerst möchte ich dabei meine Frau und meine Kinder nennen, die auf meine Arbeit stets Rücksicht genommen und mich nach Kräften unterstützt haben.

Ich danke auch Herrn Prof. Dr.-Ing. Ralph Kennel, dem Leiter des Lehrstuhls für elektrische Maschinen und Antriebe; er hat mich mit der Thematik der prädiktiven Regelung vertraut gemacht, was schließlich in das Thema dieser Arbeit mündete. Auch danke ich ihm für die Betreuung der Arbeit und die Durchsicht des Manuskripts. Darüberhinaus hat Herr Kennel es mir ermöglicht, meine Forschungsergebnisse auf internationalen Konferenzen vorzustellen und auf diese Weise weitere interessante Gedankenanstöße zu erhalten.

Herrn Prof. Dr.-Ing. Bernd Tibken vom Lehrstuhl für Automatisierungstechnik und Regelungstechnik danke ich für die Übernahme des Korreferats und die fachliche Unterstützung in regelungstechnischen Fragestellungen.

Bei Herrn Prof. Dr.rer.nat. Manfred Mendel vom Fachbereich Mathematik der Universität Wuppertal bedanke ich mich für die Hilfestellung in mathematischen Problemen. Als Spezialist auf dem Gebiet des Operations Research konnte er mir wertvolle Anregungen und Informationen besonders zu dem schwierigen Gebiet der ganzzahligen Optimierungen geben.

Mein ganz besonderer Dank gilt dem Team um Herrn Prof. Dr. Manfred Morari vom Institut für Automatik der ETH Zürich, besonders den Entwicklern der „Multi-

Parametric Toolbox“, Herrn Dipl.-Ing. Michal Kvasnica, Herrn Dipl.-Ing. Pascal Grieder, Herrn M. Sc. Mato Baotić und Herrn Dipl.-Ing. Tobias Geyer für ihre Hilfsbereitschaft und Kooperation bei der Beantwortung von Fragen bezüglich der Toolbox. Herrn Dr.-Ing. Georgios Papafotiou, der mich zu einem Besuch des Instituts einlud, möchte ich dabei nicht unerwähnt lassen. Während meiner Zeit in Zürich standen mir die oben genannten Mitarbeiter geduldig zu vielen Fragen zu Model Predictive Control und der Toolbox Rede und Antwort.

Nicht zuletzt möchte ich mich bei allen anderen Kollegen vom Lehrstuhl für elektrische Maschinen und Antriebe bedanken. Sie haben nicht nur am wissenschaftlichen Teil der Arbeit durch fruchtbringende Fachdiskussionen mitgewirkt; auch der praktische Versuchsaufbau, an dem die experimentellen Ergebnisse ermitteln wurden, wäre ohne ihre praktische Mitarbeit undenkbar.

Wuppertal, Juni 2006

Arne Linder

Inhaltsverzeichnis/Contents/目录

Abbildungsverzeichnis/List of Figures/图清单	xxi
Tabellenverzeichnis/List of Tables/表清单	xxv
1 Einleitung/Introduction/引言	1
2 Feldorientierte Regelung/Field-oriented Control/磁场定向控制	9
3 Kaskadenregelung mit PI-Regler/Cascade Control with PI-controllers/ 基于 PI 控制器的串级控制	17
3.1 Stromregelung/Current control/电流控制	21
3.2 Drehzahlregelung/Speed control/转速控制	23
3.3 Flussregelung/Flux control/磁通控制	23
3.4 Experimentelle Ergebnisse/Experimental results/实验结果	23
4 Prädiktive Regelungen/Predictive Control/预测控制	25
4.1 Einteilung nach Funktionsprinzip/Classification based on the operational principle/基于工作原理分类	31
4.2 Einteilung nach Prädiktionshorizont und Steuerungsprinzip/Classification according to prediction horizon and control principle/ 基于预测步长和控制原理分类	45
5 Modellbasierte Prädiktivregelung/Model Predictive Control/模型预测控制	49
5.1 Funktionsprinzip/Functional principle/工作原理	51
5.2 Modelle/Models/模型	59
5.2.1 Zustandsraummodell/State space model/状态空间模型	61
5.2.2 Übertragungsfunktionsbasierte, lineare Modelle/Linear transfer function- based models/基于传递函数的线性模型	63
5.2.3 Nichtlineare Modelle/Nonlinear models/非线性模型	69
6 Allgemeine Prädiktivregelung/ Generalized Predictive Control/广义预测控制	75
6.1 „Klassisches GPC“/“Classical GPC”/经典广义预测控制	75
6.1.1 Mathematische Herleitung/Mathematical derivation/数学推导	75
6.1.2 Experimentelle Ergebnisse/Experimental results/实验结果	93

6.2	GPC mit Filter/GPC with filter/带滤波器的 GPC	93
6.2.1	Mathematische Herleitung/Mathematical derivation/数学推导	99
6.2.2	Simulationen/Simulations/仿真	109
6.3	Kaskadenregelung mit GPC-Reglern/Cascade control with GPC controllers/ 基于 GPC 控制器的串级控制	113
6.3.1	Stromregelung/Current control/电流控制	115
6.3.2	Drehzahlregelung/Speed control/转速控制	117
6.3.3	Experimentelle Ergebnisse/Experimental results/实验结果	119
6.3.4	Rechenzeiten/Computation times/计算时间	133
7	Zeitdiskretes Maschinenmodell für Stromregelung/Time-discrete Machine Model for Current Control/用于电流控制的时间离散电机模型	137
7.1	Herleitung/Derivation/推导	137
7.1.1	MATLAB	141
7.1.2	Differenzenquotient/Difference quotient/微商	141
7.1.3	Laplace-Transformation/Laplace transformation/拉普拉斯变换	143
7.1.4	Potenzreihe/Power series/幂级数	147
7.2	Experimentelle Ergebnisse/Experimental results/实验结果	151
7.3	Modifiziertes Maschinenmodell für GPC/Modified machine model for GPC/ 用于 GPC 的修正电机模型	157
8	Multivariable GPC-Regelung/Control/多变量广义预测控制	165
8.1	„Klassisches“ MIMO-GPC/“Classical” MIMO-GPC/经典 MIMO-GPC	165
8.1.1	Bestimmung der Übertragungsfunktion/Determination of the transfer function/ 传递函数的确定	165
8.1.2	Berechnung der Systemmatrizen/Calculation of the system matrices/ 系统矩阵的计算	169
8.1.3	Mathematische Herleitung/Mathematical derivation/数学推导	171
8.1.4	Berücksichtigung des Regelhorizonts/Consideration of the control horizon/ 考虑控制步长后的简化	185
8.2	Störgrößenbehandlung mit GPC/Consideration of disturbance inputs with GPC/广义预测控制的扰动变量处理	187
8.2.1	Bestimmung der Übertragungsfunktion/Determination of the transfer function/ 传递函数的确定	189
8.2.2	Berechnung der Systemmatrizen/Calculation of the system matrices/ 系统矩阵的计算	189
8.2.3	Mathematische Herleitung/Mathematical derivation/数学推导	191
8.2.4	Berücksichtigung des Regelhorizonts/Consideration of the control horizon/ 考虑控制步长后的简化	203
8.3	MIMO-GPC mit Filter/MIMO-GPC with filter/带滤波器的 MIMO-GPC	205

8.3.1	Bestimmung der Übertragungsfunktion/Determination of the transfer function/传递函数的确定	207
8.3.2	Berechnung der Systemmatrizen/Calculation of the system matrices/系统矩阵的计算	207
8.3.3	Mathematische Herleitung/Mathematical derivation/数学推导	207
8.3.4	Berücksichtigung des Regelhorizonts/Consideration of the control horizon/考虑控制步长后的简化	223
8.4	Experimentelle Ergebnisse/Experimental results/实验结果	225
8.4.1	Stromregelung/Current control/电流控制	225
8.4.2	Rechenzeiten/Computation times/计算时间	227
8.5	Vergleichende Zusammenfassung/Comparative summary/对比性总结	231
9	Direkte Modellbasierte Prädiktivregelung /Direct Model-based Predictive Control /直接模型预测控制	233
9.1	Publizierte Verfahren/Published techniques/已发表的方法	237
9.2	Umrichterbetrieb mit DMPC/Inverter operation with DMPC/基于 DMPC 的逆变器运行	239
9.2.1	Berücksichtigung der Bootstrap-Schaltung/Consideration of the Bootstrap circuit/考虑自举电路	243
9.3	Modellbildung/Model formulation/模型建立	245
9.3.1	Einfaches Maschinenmodell/Simple machine model/简化电机模型	251
9.4	Implizite Lösung/Implicit solution/隐式求解	253
9.4.1	Lösungsverfahren/Solving algorithms/求解方法	257
9.4.2	Mathematische Herleitung/Mathematical derivation/数学推导	269
9.4.3	Experimentelle Ergebnisse/Experimental results/实验结果	269
9.4.4	Rechenzeiten/Computation times/计算时间	277
9.5	Explizite Lösung/Explicit solution/显式求解	281
9.5.1	Standard-Algorithmus/Standard algorithm/标准算法	299
9.5.2	Minimum-Time-Controller/Minimum Time Controller/最小时间控制器	299
9.5.3	Binärer Suchbaum/Binary search tree/二叉搜索树	305
9.5.4	Optimale Komplexitäts-Reduzierung/Optimal complexity reduction/最优的复杂度降低	315
9.5.5	Experimentelle Ergebnisse/Experimental results/实验结果	327
10	Verwandte Regelstrukturen/Related Control Structures/相关的控制结构	343
10.1	Internal Model Control/Internal Model Control/内部模型控制	343
10.2	Lineare Quadratische Regelung/ Linear Quadratic Control/线性二次控制	349
10.2.1	Funktionsweise von LQR/Functional principle of LQR/ LQR 工作原理	349
10.2.2	GPC und LQR/GPC and LQR/GPC 和 LQR	353

11 Zusammenfassung und Ausblick/ Summary and Outlook/总结和展望	359
Literaturverzeichnis/Bibliography/参考文献	371
A Glossar Polynommatrizen/Glossary Polynomial Matrices/词汇表 多项式矩阵	383
B Nomenklatur/Nomenclature/术语表	397
C Normierungsgrößen/Normalization Values/标么量	431
D Physikalische Maschinengrößen/Physical Machine Constants/电机参数	433
E Polynome und Matrizen bei GPC/ Polynomials and Matrices for GPC/ 用于 GPC 的多项式和矩阵	435
E.1 SISO-System/SISO-System/单输入单输出系统	435
E.2 MIMO-System/MIMO-System/多输入多输出系统	437
E.2.1 Definitionen/Definitions/定义	437
E.2.2 Dimensionen/Dimensions/维度	445
F Verfahren zur Matrixinversion/Methods for Matrix Inversion/矩阵求逆方法	447
F.1 Gauß-Verfahren/Gauss algorithm/高斯法	447
F.2 Gauß-Jordan-Verfahren/Gauss Jordan algorithm/高斯-若尔当法	449
F.3 Austauschverfahren/Exchange algorithm/互换法	449
F.4 LR-Zerlegung/LR decomposition/LR 分解	449
F.5 Verfahren von Cholesky/Algorithm of Cholesky/科列斯基法	449
F.6 Rechenzeitaufwand/Computation times/计算时间	451
G Alternativverfahren Matrizenzerlegung/ Alternative Method for Matrix De- composition/矩阵分解的替代性方法	453

Abbildungsverzeichnis/List of Figures/图清单

1.1	Typische Struktur einer Kaskadenregelung	5
1.2	Typical structure of a cascade controller	8
1.3	串级控制典型结构	8
2.1	Dreiphasiges System und zweiphasiges Ersatzsystem; Three-phase system and alternative two-phase system; 三相系统和两相等效系统	11
2.2	Complexes Maschinenmodell/Complex machine model/复数电机模型	13
2.3	Skalares Maschinenmodell (Feldkoordinaten); Scalar machine model (Field coordinates); 标量电机模型 (磁场坐标系)	15
2.4	Rotormodell/Rotor model/转子模型	16
3.1	Feldorientierte Antriebsregelung mit PI-Reglern	19
3.2	Field-oriented drive control with PI-controllers	20
3.3	基于 PI 控制器的磁场定向驱动控制	20
3.4	Vereinfachtes Modell für die Stromregelung (Feldkoordinaten); Simplified model for current control (field coordinates); 用于电流控制的简化模型 (磁场坐标系)	21
3.5	Stromregelkreis mit PI-Regler; Current control loop with PI-controller; 带有 PI 控制器的电流控制闭环	21
3.6	Drehzahlregelkreis mit PI-Regler; Speed control loop with PI-controller; 带有 PI 控制器的电流控制闭环	23
4.1	Typische Struktur eines Prädiktivreglers	29
4.2	Typical structure of a predictive controller	30
4.3	预测控制器典型结构	30
4.4	Hysteresebasierter Prädiktivregler nach Holtz/Stadtfeld [55]; Hysteresis-based predictive controller according to Holtz/Stadtfeldt [55]; 学者 Holtz 和 Stadtfeld 提出的基于滞环的预测控制器 [55]	33
4.5	Trajektorienbasierter Prädiktivregler nach Mutschler [90]; Trajectory-based predictive controller according to Mutschler [90]; 基于学者 Mutschler 提出的基于轨迹的预测控制器 [90]	35
4.6	Stammbaum prädiktiver Regelverfahren	41

4.7	Family tree of predictive control schemes	43
5.1	Typische Struktur eines MPC-Reglers	51
5.2	Typische Struktur eines MPC-Reglers	55
5.3	一种 MPC 控制器的典型结构	55
5.4	Regel- und Prädiktionshorizont beim MPC-Regler	57
5.5	Definition of the control and prediction horizon	58
5.6	MPC 控制器的控制步长和预测步长	58
5.7	Struktur eines Hammerstein-Modells/Structure of a Hammerstein model/哈默斯坦模型结构	71
5.8	Struktur eines Wiener Modells	73
5.9	Structure of a Wiener model	73
5.10	维纳模型结构	73
6.1	Zeitdiskrete Übertragungsfunktion/Discrete time transfer function/时间离散传递函数	77
6.2	Struktur eines GPC-Reglers ohne Filter	95
6.3	Structure of a GPC controller without filter	95
6.4	一种无滤波的 GPC 控制器结构	96
6.5	Struktur eines GPC-Reglers mit Filter	97
6.6	Structure of a GPC controller with filter	98
6.7	一种带滤波的 GPC 控制器结构	98
6.8	Simulation Führungsverhalten ohne Störung des Messsignals	109
6.9	Simulation: Reference action without noise disturbances	110
6.10	Simulation Führungsverhalten mit Störung des Messsignals: (a) ohne Filterung, (b) mit Filterung; Simulation: Reference action with noise disturbances: (a) without Filter, (b) mit Filter; 测量信号含有扰动时的系统响应仿真: (a) 无滤波, (b) 含滤波	111
6.11	测量信号无扰动时的系统响应仿真	112
6.12	Feldorientierte Antriebsregelung mit GPC-Reglern	113
6.13	Field-oriented drive control with GPC controllers	113
6.14	基于 GPC 控制器的磁场定向驱动控制	114
6.15	Stromregelkreis mit GPC-Regler/Current control loop with GPC controller/带有 GPC 控制器的电流控制闭环	115
6.16	Drehzahlregelkreis mit GPC-Regler/Speed control loop with GPC controller/带有 GPC 控制器的转速控制闭环	117
6.17	Stromregelung Großsignalverhalten: (a) PI-Regler, (b) GPC-Regler; Current control: Large-signal behavior: (a) PI controller, (b) GPC controller; 电流控制的大信号响应: (a) PI 控制器, (b) GPC 控制器	121

6.18	Stromregelung Kleinsignalverhalten: (a) PI-Regler, (b) GPC-Regler; Current control: Small-signal behavior: (a) PI-Regler, (b) GPC-Regler; 电流控制的小信号响应: (a) PI 控制器, (b) GPC 控制器	123
6.19	Stromregelung: Großsignalverhalten bei $\omega = 0,4$ /Current control: Large-signal behavior at $\omega = 0.4$ /电流控制: 转速 $\omega = 0.4$ 下的大信号响应	125
6.20	Drehzahlregelung Kleinsignalverhalten: (a) PI-Regler; (b) GPC-Regler; Speed Control: Small-signal behavior: (a) PI-Regler; (b) GPC-Regler; 转速控制的小信号响应: (a) PI 控制器, (b) GPC 控制器	129
6.21	Drehzahlregelung Kleinsignalverhalten, zeitlich gedehnt; Speed Control: Small-signal behavior, zoomed time scale; 局部放大的转速控制的小信号响应	131
7.1	Prädiktion bei Sollwertsprung in i_{sq} : (a) Vorausberechnung von i_{sq} , (b) Vorausberechnung von i_{sd} ; Prediction with a step change in the reference variable i_{sq} : (a) Advance calculation of i_{sq} , (b) Advance calculation of i_{sd} ; 参比变量 i_{sq} 阶跃变化时的预测: (a) 提前预测 i_{sq} , (b) 提前预测 i_{sq}	153
7.2	Prädiktion bei Sollwertsprung in i_{sd} : (a) Vorausberechnung von i_{sq} , (b) Vorausberechnung von i_{sd} ; Prediction with a step change in the reference variable i_{sd} : (a) Advance calculation of i_{sq} , (b) Advance calculation of i_{sd} ; 参比变量 i_{sd} 阶跃变化时的预测: (a) 提前预测 i_{sq} , (b) 提前预测 i_{sq}	155
8.1	Mehrgrößen-Stromregelung: (a) ohne Störgrößenberücksichtigung, (b) mit Störgrößenberücksichtigung; Multidimensional current control: (a) Without consideration of disturbances, (b) With consideration of disturbances; 多变量电流控制: (a) 不考虑扰动, (b) 考虑扰动	229
9.1	Schaltbild Zweipunktumrichter mit Spannungszwischenkreis; Two-level inverter circuit with DC link/Two-level inverter circuit with DC link; 带中间直流环节的两电平逆变器电路	235
9.2	Prinzip einer Maschinenregelung: (a) konventionell, (b) direkt; Principle of machine control: (a) conventional, (b) direct; 电机控制基本原理: (a) 传统的, (b) 直接的	237
9.3	Mögliche Schaltzustände eines Zweistufenumrichters; Possible switching states of a two-level inverter; 一个两电平逆变器可能的开关状态	239
9.4	Skalares Maschinenmodell (Statorkoordinaten)/Scalar machine model (stator coordinates)/标量电机模型 (定子坐标系)	247
9.5	Vereinfachtes Modell für die Stromregelung (Statorkoordinaten); Simplified model for current control (stator coordinates); 用于电流控制的简化模型 (定子坐标系)	251

9.6 Vereinfachtes Modell für die Stromregelung (Brückenzustände); Simplified model for current control (bridge switching states); 用于电流控制的简化模型 (半桥状态)	251
9.7 Stromregelung mit DMPC-Regler, stationärer Betrieb: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$; Current control with DMPC controller, stationary operation: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$; 基于 DMPC 控制器的电流控制, 稳态运行: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$	273
9.8 Stromregelung mit DMPC-Regler, dynamischer Betrieb: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$; Current control with DMPC controller, dynamisch operation: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$; 基于 DMPC 控制器的电流控制, 动态运行: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$	275
9.9 Explizite Lösung für drei Halbbrücken: (a) Phase a, (b) Phase b, (c) Phase c; Explicit solution for three half bridges: (a) Phase a, (b) Phase b, (c) Phase c; 三个半桥的显式解: (a) a 相, (b) b 相, (c) c 相	289
9.10 Trajektorien der expliziten Lösung/Trajectories of the explicit solution/显式解的轨迹	289
9.11 Funktionsprinzip Suchbaum/Functional principle of the search tree/搜索树工作原理	309
9.12 Stromregelung mit explizitem DMPC- bzw. PI-Regler/Current control with explicit DMPC- and PI-Controller/基于显式 DMPC 和 PI 控制器的电流控制	339
10.1 IMC-Struktur/IMC structure/IMC 结构	345
10.2 IMC-Struktur, umgeformt auf klassische Reglerstruktur/IMC structure transformed to the classical controller struture/IMC 结构, 转化为典型的控制器结构	347
10.3 Struktur einer LQ Regelung ($\mathbf{w} = 0$, siehe Text)/Structure of LQ control ($\mathbf{w} = 0$, see text)/LQ 控制的结构 ($\mathbf{w} = 0$, 见正文)	353

Tabellenverzeichnis/List of Tables/表清单

4.1	Klassifizierung von Prädiktivreglern	45
4.2	Classification of predictive control/预测控制器分类	48
6.1	Einstellungen für GPC-Stromregler/Parameter settings for the GPC current controller/GPC 电流控制器的参数设置	119
6.2	Einstellungen für GPC-Drehzahlregler	127
6.3	Parameter settings for a GPC speed controller	128
6.4	GPC 转速控制器的参数设置	130
6.5	Rechenzeitvergleich PI-Regler und GPC-Regler	133
6.6	Comparison of computation times for PI and GPC controller	134
6.7	PI 和 GPC 控制器的计算时间对比	136
8.1	Vergleich der verschiedenen MIMO-GPC-Versionen; /Comparison of different MIMO-GPC variations; /MIMO-GPC 的不同变体形式的对比	231
9.1	Mögliche Schaltzustände eines Zweistufenumrichters; Possible switching states of a two-level inverter; 一个两电平逆变器可能的开关状态	241
9.2	Rechenzeiten DMPC-Regler	277
9.3	Computation times for a DMPC controller	278
9.4	DMPC 控制器的计算时间	278
9.5	Rechenzeit für MIMO-Stromregler	327
9.6	Calculation times for MIMO current controllers	328
9.7	基于 MIMO 电流控制器的计算时间	328
F.1	Rechenzeiten für Matrixinversionen in μs	451
F.2	Calculation times for matrix inversion in μs	452
F.3	矩阵求逆所需的计算时间, 单位 μs	452

1 Einleitung/Introduction/引言

In der Industrie werden geregelte Antriebe für vielfältige Zwecke eingesetzt; ihre Leistungsbandbreite reicht von wenigen Watt für einen kleinen Stellmotor bis hin zu mehreren hundert Kilowatt bei Traktionsanwendungen. Über Jahrzehnte war der Gleichstrom-Reihenschlussmotor wegen seiner leichten Regelbarkeit der am meisten eingesetzte Motortyp, bei kleineren Leistungsanforderungen kamen auch permanenterregte Gleichstrommaschinen zum Einsatz. Nachteil dieser Maschinen ist jedoch der verschleißbehaftete Kommutator, der zu einem erhöhten Wartungsaufwand und zu einer erhöhten Verschmutzung der Maschine durch den Bürstenabrieb führt. Gleichzeitig ist konstruktionsbedingt eine vollständige Kapselung der elektrisch leitenden Bauteile relativ aufwändig. Drehfeldmaschinen sind zwar ebenfalls seit der Anfangszeit der elektrischen Maschinen bekannt, wurden aber selten eingesetzt, da die Drehzahl der Maschine direkt mit der Drehfrequenz des speisenden Systems zusammenhängt. Folglich wurden Maschinen dieser Art nur da verwendet, wo dieser Effekt gewünscht ist, z. B. als Generatoren in Kraftwerken, oder z. B. bei Lüftern, wo eine Drehzahlverstellung unnötig ist.

Zusammen mit den Fortschritten der Halbleitertechnik wuchsen auch die Möglichkeiten, Drehfeldmaschinen auf einfache Art und Weise drehzahlverstellbar zu gestalten. Dazu wird im einfachsten Fall die Netzspannung gleichgerichtet und aus diesem sogenannten Zwischenkreis mit Hilfe eines Umrichters ein Drehspannungssystem beliebiger Frequenz erzeugt. Hiermit war zwar das Problem gelöst, die Maschine mit beliebigen Drehzahlen zu betreiben, eine dynamische Regelung wie bei einer Gleichstrommaschine war aber trotzdem weiterhin nicht möglich. Problematisch blieb die nicht vorhandene Trennung von Felderzeugung und Drehmomentbildung. Dieses Problem wurde erst mit der Einführung der sogenannten feldorientierten Regelung (s. Kap. 2) vor etwa 30 Jahren gelöst.

Stand der Technik ist seitdem sowohl für Synchron- als auch für Asynchronmaschinen die feldorientierte Regelung; als Regler kommen dabei PI-Regler in Kaskadenstruktur zum Einsatz (s. Kap. 3). Unerwünschte Nebeneffekte und Nichtlinearitäten der Maschine werden mit Vorsteuerungen oder Störgrößenaufschaltungen dergestalt bekämpft, dass die Qualität der erzielten Regelung allen in der Antriebstechnik bisher vorkommenden Anwendungen gerecht wird.

In industry, variable speed electrical drives are applied to various applications; their power bandwidth differs from few watts for a small servo-motor up to several hundreds of kilowatts for traction applications. For decades, the DC series motor was the mostly used drive motor because of its simple speed controllability; for smaller power demands,

permanent-magnet DC machines were also used. Nevertheless, the disadvantage associated with these machines is the wear-associated commutator which leads to a rise in maintenance expenses and also to a raised soiling of the machine by brush abrasion. At the same time, a complete encapsulation of all electrical components is relatively complex due to the construction. Rotating field AC machines are known since the initial times of electrical machines, however, they were rarely used as the speed of these machines changes with the frequency of the supplying grid. Consequently, machines of this kind were only used where this feature is required, e. g. as generators in power stations, or, e. g., for fans where speed adjustment is unnecessary.

Along with the advances of semiconductor technology, the possibilities to make AC machines in a simple way speed variable increased. Therefore, the supply voltage is rectified into a so-called DC-link, from where a three-phase voltage with variable frequency can be generated with the help of an inverter. Herewith, the task to operate the machine with any speed was solved, however, a dynamic control similar to DC machines was not possible. This was due to the fact that the separate control of field flux and torque was not available. This problem was solved with the introduction of the so-called field oriented control (see Chapter 2) about 30 years ago.

Since this time, field-oriented control for synchronous as well as for asynchronous machines is state-of-the-art; thereby, PI-controllers are used in a cascaded structure (see Chapter 3). Undesirable side effects and nonlinearities of the machine are tackled with precontrols or feed forward compensation techniques in such a way that the quality of the achieved control meets the requirements of all drive applications existing so far.

在工业领域，可控驱动可用于多种用途，其功率范围涵盖了从几瓦的小型伺服电机至数百千瓦的大功率牵引应用场合。直流电机因其可控性简单而曾经成为最为广泛应用的电机类型，在某些较小功率应用场合有时也采用永磁体励磁式直流电机。然而，直流电机的缺点在于其易磨损的换向器，它会导致高的维护成本，以及由于其配套的电刷磨损而使得电机容易变脏。与此同时，从设计角度来看，导电部件的完全封装也相对复杂。虽说交流电机在电机出现初期已为人所知，但却很少投入使用，因其转速直接取决于供电系统的频率，从而很难实现转速调节。所以，交流电机过去通常仅在无需进行速度调节的场合得到应用，如发电厂发电机组或风机等。

随着半导体技术的进步，使得以更简单的方式实现交流电机转速调节成为可能。最为简单的方式是首先将电网电压整为直流，然后借助于逆变器将上述所谓的中间直流电压转化为任意频率的交流电压。至此，虽解决了无法任意调节转速的问题，但要获得类似于直流电机的动态控制效果依然是不可能的，其原因在于励磁磁场的产生和转矩的形成这两部分无法独立实现。直至 1970 年左右，通过引入了所谓的磁场定向控制才解决了上述问题（见第 2 章）。

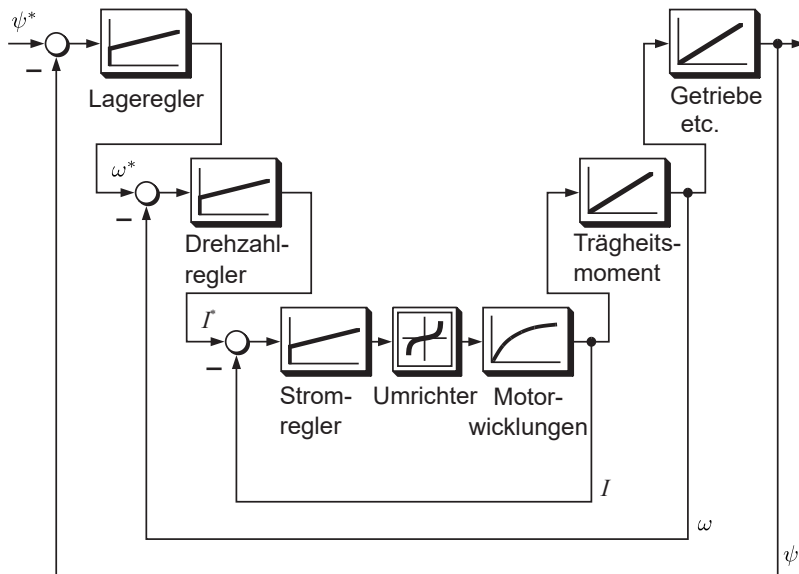
自那时起，磁场定向控制一直作为同步电机和异步电机控制的最前沿技术，其控制器类型通常是在串级结构中使用的 PI 控制器（见第 3 章）。一般来说，可采取前馈或接入扰动电路的措施来抑制或消除不希望的外界干扰及电机的非线性因素，从而使得控制效果能够满足驱动技术领域的几乎所有应用场合。

Es erhebt sich nun die Frage, wieso neue Regelverfahren untersucht werden sollen, wenn das bisher verwendete Regelverfahren allen derzeit bekannten Anforderungen genügt. Dies soll anhand der klassischen Struktur eines geregelten Antriebs betrachtet werden. Wie Abb. 1.1 zeigt, besteht eine Lageregelung aus drei ineinander verschachtelten, also kaskadierten, Regelkreisen. Die äußeren Regelschleifen bestehen dabei aus einem Integrator, der das Verhalten des Trägheitsmoments bzw. des Getriebes beschreibt, und dem jeweils nächsten, unterlagerten Regelkreis. Bei der Auslegung einer solchen Struktur geht man üblicherweise so vor, dass man die unterlagerte Regelung durch ein PT_1 -Glied approximiert und die Reglerparameter entsprechend dem sogenannten *Symmetrischen Optimum* einstellt. Gute Regeleigenschaften können aber nur dann erzielt werden, wenn die Zeitkonstanten der inneren und der nächstäußeren Regelschleife sich um einen Faktor von mindestens 7–10 unterscheiden [38]. Bei einer Lageregelung muss die Stromregelung folglich etwa 50–100 mal so schnell sein, wie der Lageregelkreis. Bei hochdynamischen Antrieben ist abzusehen, dass über kurz oder lang die gewünschte Dynamik mit Hilfe einer Kaskadenstruktur nicht mehr zu erzielen ist, da ein entsprechend schneller Stromregler nicht realisierbar ist. Da der Zwang nach einem ultraschnellen Stromregelkreis einzig und allein aus der Kaskadenstruktur selbst erwächst, könnte die gewünschte Dynamik im Lage- oder Drehzahlregelkreis mit einer anderen Reglerstruktur auch dann erzielt werden, wenn die Rechenzeit des Regelungsprozessors deutlich länger ist als im Fall der Kaskadenregelung. Es ist daher abzusehen, dass mittelfristig lineare Regelkreise in Kaskadenstruktur den wachsenden Anforderungen an die Dynamik der elektrischen Antriebe nicht mehr gerecht werden können. Als Alternative bietet sich die prädiktive oder vorausberechnende Regelung an, die keine Regelkaskade benötigt. Kap. 4 erläutert das grundlegende Prinzip einer solchen Regelstrategie.

Alle im Bereich der Antriebstechnik bisher veröffentlichten Regelverfahren berechnen das Regelverhalten lediglich für den nächsten Abtastschritt voraus. Entsprechend kann auch die Optimierung der Stellgröße nur über den einen, einzigen, vorausberechneten Abtastzyklus erfolgen. Aus der klassischen Regelungstechnik sind allerdings Verfahren bekannt, die einen höheren Prädiktionshorizont aufweisen und die daher als *Long-range Predictive Control* bezeichnet werden; wegen des zur Vorausberechnung verwendeten Streckenmodells werden sie auch *Model Predictive Control (MPC)* oder *Modellbasierte Prädiktivregler* genannt. Diese Verfahren werden in Kap. 5 näher diskutiert. Bedingt durch den großen Prädiktionshorizont sind diese Strategien im Vergleich zur herkömmlichen linearen Regelung relativ rechenaufwändig, weswegen ihr Einsatz in der Antriebstechnik bisher allein aus diesem Grund nicht möglich war. Die verfügbare Rechenleistung nimmt jedoch kontinuierlich zu, während gleichzeitig der Preis für leistungsfähige Prozessoren sukzessive abnimmt, so dass das Argument des zu hohen mathematischen Aufwands an Gewicht verloren hat und in Zukunft noch mehr verlieren wird. In den Kapiteln 6–9 werden unterschiedliche MPC-Verfahren vorgestellt und ihre Vor- und Nachteile für den Einsatz in der Antriebstechnik erläutert.

Now the question rises why new control techniques should be examined if the control techniques available today can fulfill all known requirements until now. This shall be explained on the basis of the classical structure of a controlled electrical drive. As shown in Figure 1.2, a position control scheme consists of a threefold cascaded control structure. The external control loops consist of an integrator which describes the behavior of the system's inertia and the gearbox, respectively, and the next inner control loop. When designing such a cascaded structure, normally at first the inner loop is approximated by a PT_1 -block and then, the controller parameters are determined according to the so-called *symmetrical optimum*. However, good control properties can only be achieved if the time constants of the inner control loop and the subsequent external control loop differ by a factor of at least 7–10 [38]. Consequently, for a position control scheme, the current controller must be about 50–100 times faster than the position control loop. It can be foreseen that, sooner or later, the dynamics which can be achieved with a cascade structure are no longer suitable for highly dynamic drive applications, as a sufficiently fast current controller is not feasible. As the need for an ultra fast current control loop arises solely from the cascaded structure itself, the desired fast dynamics in the position or speed control loop could be achieved with other control structures, even if the computation time of the control process is significantly higher than with cascaded control. Hence, it can be seen that, in the medium term, linear controllers in cascaded structure can no longer meet the growing requirements for the dynamics of electrical drives. Predictive or precalculating controllers, which need no cascaded control, offer an alternative. Chapter 4 explains the basic principle of such a control strategy.

All control strategies published so far in the domain of drive technology precalculate the control behavior merely for the next sampling cycle. Correspondingly, the optimization of the actuating variable can also only be done for this single precalculated sampling cycle. However, from the classic control theory, strategies are known which allow a higher prediction and are therefore called *Long Range Predictive Control*. Since they use a model of the controlled process for the precalculation, they are also called *Model Predictive Controllers* (MPC) or *Model-Based Predictive Controllers*. These strategies are discussed in detail in Chapter 5. Due to the longer prediction horizon, these strategies are relatively complex in their calculations in comparison to ordinary linear controllers; hence, their implementation in drive technology was not possible yet. Nevertheless, the availability of faster computing devices is increasing continuously and at the same time their price is continuously decreasing. So the argument of too high computational expenses has already lost much of its relevance and will lose even more in the future. In the Chapters 6–9 different MPC strategies are introduced and their advantages and disadvantages are explained with respect to their implementation in drive technology.

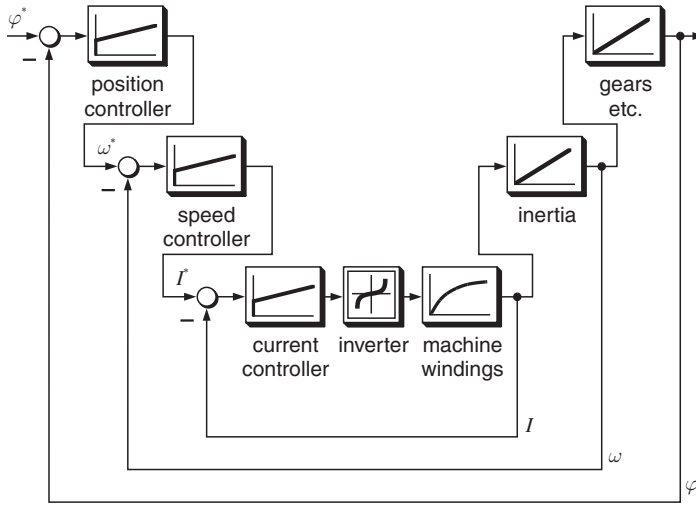


1.1: Typische Struktur einer Kaskadenregelung

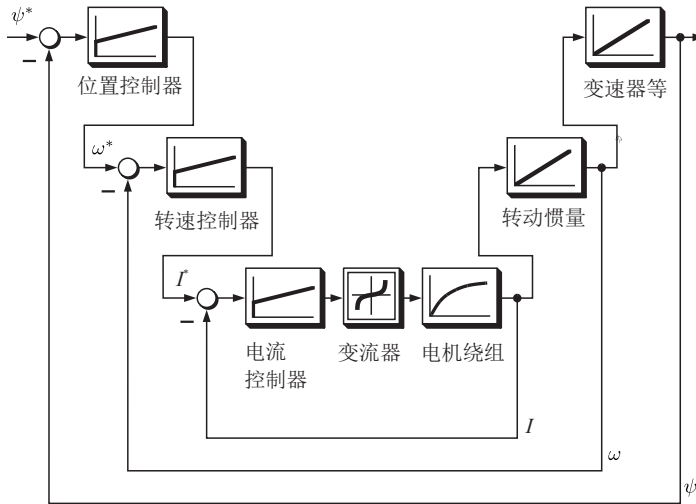
现在要提出的问题是：为何还要研究新的控制方法，既然已有的控制方法已经能够满足当前几乎所有已知的应用场合？为此，我们首先考察可控驱动领域常用的串级控制典型结构。如图 1.3 所示，位置控制是由三个前后相互连接的控制回路组成，所以称之为串级结构。最外层的控制回路包含一个积分器，它所代表的是转动惯量及变速器的特性，当然还包含内层的全部控制回路。对串级类型结构进行设计时，通常可将内层控制回路近似为一个 PT_1 -环节，然后即可应用所谓的对称最优方法进行控制器参数设计。然而，只有当外层回路的时间常数至少为内层回路时间常数的 7–10 倍时，才有可能获得较好的控制效果 [38]。那么对于位置控制来说，电流控制回路的响应时间应约为位置控制回路的 50–100 倍。基于此可以预见的是，在高动态驱动应用场合下，借助于串级结构将很难满足所期望的动态特性，因为与此相对应的是超快的电流控制，这是难以实现的。由于对电流控制回路快速性的要求源于串级结构本身的特点，那么即使当控制处理器的计算时间明显慢于串级控制所需的情形，也是有可能通过使用其他类型的控制结构来实现位置或速度控制回路的高动态特性。可以得出结论：长远来看，串级结构下的线性控制回路将逐渐难以满足电力驱动领域更加严苛的要求。无需串级结构的预测控制提供了一种可替代方案，第 4 章将阐述预测控制基本原理。

到目前为止，驱动控制技术领域所发表的控制方法仅提前预测出下一个采样周期的控制特性。与此相对应，控制量的优化也仅能基于提前预测的单个采样周期来实现。然而，在经典控制理论领域，实现更多步长的方法也早已为人所知，通常称之为多步长预测控制。由于采用的是被控系统模型来完成预测，所以也称为模型预测控制。我们将在第 5 章对模型预测控制方法进行详细探讨。顾名思义，由于多步长预测控制采用了更大的预测步长，相比于传统的线性控制，它需要更强大的计算能力。正是由于这一限制，使得到目前为止在驱动控制领域还无法投入使用上述方法。随着市面上所能提供处理器的计算性能不断增加，同时其价格也在持续下降，可得出结论：有关控制方法受到计算能力限制的讨论将逐渐退出历史舞台，并在未来可能无足轻重。第 6–9 章将介绍一些不同的模型预测控制方法，并分别阐述将其应用至驱动技术领域而带来的优缺点。

Notizen/Notes/笔记



1.2: Typical structure of a cascade controller



1.3: 串级控制典型结构

2 Feldorientierte Regelung/Field-oriented Control/磁场定向控制

Eine Gleichstrommaschine besitzt für den Antriebstechniker den großen Vorteil, dass sie getrennte Feld- und Ankerwicklungen besitzt und sich die beiden wichtigsten Zustandsgrößen, nämlich der Fluss und das Drehmoment, daher getrennt voneinander über den Erreger- bzw. Ankerstrom einstellen lassen. Der Kommutator als mechanischer Wechselrichter sorgt dafür, dass der Strombelag des Rotors immer die richtige Lage hat, so dass der Ankerstrom und der Erregerfluss immer senkrecht aufeinander stehen. Eine Asynchronmaschine hingegen, die wegen ihrer Wartungsfreiheit für den praktischen Einsatz große Vorteile bietet, besitzt im Gegensatz dazu nur eine einzige aktive Wicklung, die Ständerwicklung; die Rotorwicklung ist entweder kurzgeschlossen (Kurzschlussläufer) oder mit verstellbaren Widerständen verbunden (Schleifringläufer). Folglich müssen sowohl das Feld als auch das Drehmoment in der Maschine über die Ständerwicklung aufgebaut werden. Eine einfache Aufteilung der Regelstruktur in Feld- und Drehmomentregelung ist so ohne weiteres nicht möglich; für eine hohe Antriebsperformance müssen beide Größen aber unbedingt unabhängig voneinander geregelt werden [57].

Abhilfe schafft hier die sogenannte *feldorientierte Regelung* [79], die mit Hilfe der *Raumzeigerdarstellung* [70] beschrieben werden kann. Grundidee dieser Darstellungsform ist der Gedanke, dass sich ein dreiphasiges Stromsystem wie in Abb. 2.1(a) zu sehen durch ein dreiachsiges Koordinatensystem darstellen lässt. Die drei Achsen a , b und c sind jedoch nicht voneinander linear unabhängig, was eine mathematische Beschreibung der Vorgänge in einem dreiphasigen System sehr erschwert. Daher wird ein zweiphasiges Ersatzsystem mit zwei voneinander linear unabhängigen Achsen konstruiert. Abb. 2.1(b) zeigt dieses äquivalente System. Um zu einer einfachen Darstellungsform der zweiphasigen Größen zu kommen, wählt man üblicherweise ein komplexes Koordinatensystem. Mit Hilfe der in den Gln. (2.1) bis (2.4) angegebenen Transformationsformeln können nun die entsprechenden Größen, hier am Beispiel der Statorspannungen dargestellt, von einem Koordinatensystem in das andere übertragen werden.

$$\mathbf{u}_s = \frac{2}{3} (u_{sa} + \mathbf{a} \cdot u_{sb} + \mathbf{a}^2 \cdot u_{sc}) \quad \text{wobei} \quad \mathbf{a} = e^{j\frac{2\pi}{3}} \quad (2.1)$$

$$u_0 = \frac{1}{3} (u_{sa} + u_{sb} + u_{sc}) \quad (2.2)$$

$$u_{s\alpha} = \Re\{\mathbf{u}_s\} = \frac{2}{3} (u_{sa} - \frac{1}{2} u_{sb} - \frac{1}{2} u_{sc}) \quad (2.3)$$

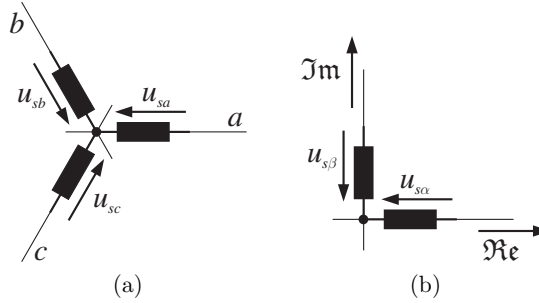
$$u_{s\beta} = \Im\{\mathbf{u}_s\} = \frac{1}{\sqrt{3}} (u_{sb} - u_{sc}) \quad (2.4)$$

In drive applications, a DC machine has the great advantage that it has separate field and armature windings. Thus, both of its most important states, namely flux and torque, can be controlled independently of each other by the field winding current and by the armature current, respectively. The commutator acting as a mechanical converter makes sure that the current distribution of the rotor is always positioned properly, so that the rotor current and the main flux are always in quadrature to each other. Asynchronous machines, on the other hand, which offer great advantages because of their maintenance free operation, have only one active winding, the stator winding. The rotor winding is either short circuited (squirrel-cage motor) or connected with variable resistors (slip ring rotor). Consequently, both flux and torque in the machine have to be raised via the stator windings. Splitting the control structure in a simply way into flux and torque control is not possible. However, for high drive performance both states must be controlled independently of each other [57].

A remedy for solving this problem is the use of the so-called *field-oriented control* [79], which can be described with the help of *space vector representation* [70]. The basic idea behind this representation is that the three-phase current system can be represented by a three-axis coordinate system as shown in Figure 2.1(a). Unfortunately, the three axes a , b and c are not linearly independent of each other, a fact that complicates a mathematical description of the actions in a three-phase system. Hence, an alternative two-phase system with two axes linearly independent of each other is constructed. Figure 2.1(b) shows this equivalent system. To obtain an easy representation of the two phase quantities, usually a complex coordinate system is selected. Thus, the corresponding quantities from one coordinate system, e. g. the stator voltage quantities, can be transformed into another coordinate system with the help of the transformation equations (2.1) to (2.4).

对于驱动技术工程师来说，直流电机的一大优势是其所具有的相互独立的励磁绕组和电枢绕组。由于上述特点，可分别通过励磁电流和电枢电流来实现对磁链和转矩的调节。值得一提的是，磁链和转矩同时也是电机最为重要的两个状态变量。换向器作为机械式变流器的功能在于它可使得转子电流方向始终处于正确的位置，从而保证电枢电流与励磁磁链总能够相互正交。相反，异步电机的一大优势则是较低的维护成本，它仅包含一组有效绕组，也即定子绕组；转子绕组则有两种形式，分别为短接式或外接可调电阻式。正是如此，异步电机的磁链和转矩均需通过定子绕组来构建，而将控制结构简单分为独立的磁链控制和转矩控制看似是不可能的。但是，对于高性能驱动控制来说，则必须实现对磁链和转矩的独立控制 [57]。

可行的解决方案是一种借助于空间矢量原理 [70] 的磁场定向控制 [79]。空间矢量原理的基本思想：如图 2.1(a) 所示，一个三相电流系统可使用一个三相坐标系进行描述，但由于 a , b 和 c 三相绕组并非相互独立，这使得三相系统的各相过程的数学描述较为困难。于是设想使用一个相互独立的两相坐标系来构造一个两相等效系统，图 2.1(b) 给出了这样一个两相等效系统。更进一步，为了能够得到一种简单的两相变量描述形式，可选取数学上最为常见的复数坐标系。借助于方程组 (2.1) - (2.4)，可将三相坐标系中的变量转换至两相坐标系，此处以定子电压为例。



2.1: Dreiphasiges System und zweiphasiges Ersatzsystem;
 Three-phase system and alternative two-phase system;
 三相系统和两相等效系统

Die Idee der feldorientierten Regelung beruht nun darauf, dass zuerst die drei Ständerströme i_{sa} , i_{sb} und i_{sc} in den drei Wicklungssträngen in einen komplexen Stromraumzeiger \mathbf{i}_s transformiert werden. Dazu wird eine (2.1) analoge Formel für die Statorströme verwendet. In einem beliebigen mit der Drehgeschwindigkeit ω_k rotierenden Koordinatensystem lässt sich eine elektrische Asynchronmaschine dann durch folgende Gleichungen beschreiben [70, 71]:

$$\mathbf{u}_s = r_s \cdot \mathbf{i}_s + \frac{d\psi_s}{d\tau} + j\omega_k \psi_s \quad (2.5)$$

$$0 = r_r \cdot \mathbf{i}_r + \frac{d\psi_r}{d\tau} + j(\omega_k - \omega) \psi_r \quad (2.6)$$

$$\psi_s = l_s \cdot \mathbf{i}_s + l_h \cdot \mathbf{i}_r \quad (2.7)$$

$$\psi_r = l_r \cdot \mathbf{i}_r + l_h \cdot \mathbf{i}_s \quad (2.8)$$

Für eine Stromregelung, wie sie in der inneren Schleife einer Kaskadenregelung implementiert wird, ist die Wahl von \mathbf{i}_s und ψ_r als Zustandsvariablen die geeignete Lösung. Nach Umformungen gemäß [56, 57] erhält man folgende Differenzialgleichungen:

$$\mathbf{i}_s + \tau_\sigma' \frac{d\mathbf{i}_s}{d\tau} = -j\omega_k \tau_\sigma' \mathbf{i}_s + \frac{k_r}{r_\sigma} \left(\frac{1}{\tau_r} - j\omega \right) \psi_r + \frac{1}{r_\sigma} \mathbf{u}_s \quad (2.9)$$

$$\psi_r + \tau_r \frac{d\psi_r}{d\tau} = -j(\omega_k - \omega) \tau_r \psi_r + l_h \cdot \mathbf{i}_s \quad (2.10)$$

wobei $\tau_s = \frac{l_s}{r_s}$, $\tau_r = \frac{l_r}{r_r}$, $\sigma = 1 - \frac{l_h^2}{l_s l_r}$, $l_s' = \sigma l_s$, $l_r' = \sigma l_r$, $k_r = \frac{l_h}{l_r}$, $k_s = \frac{l_h}{l_s}$, $\tau_s' = \frac{\sigma l_s}{r_s}$, $\tau_r' = \frac{\sigma l_r}{r_r}$, $r_\sigma = r_s + r_r \cdot k_r^2$ und $\tau_\sigma' = \frac{\sigma l_s}{r_\sigma}$. Abb. 2.2 zeigt den dazugehörigen Signalfussplan. Die Doppelstriche symbolisieren dabei komplexe Größen.

The idea of field-oriented control is based on the fact that the three stator currents i_{sa} , i_{sb} and i_{sc} in the three stator windings are first transformed into a complex state space vector \mathbf{i}_s . Therefore, a transformation analog to (2.1) is used for the stator currents. Then, an induction machine can be described in an arbitrary reference coordinate system, rotating with ω_k , with the equations (2.5) to (2.8) [70, 71].

For current control, as it is implemented in the internal loop of a cascade control structure, it is an adequate solution to choose \mathbf{i}_s and $\boldsymbol{\psi}_r$ as state variables. After reformulating according to [56, 57], the differential equations (2.9) and (2.10) can be obtained:

$$\mathbf{i}_s + \tau_\sigma' \frac{d\mathbf{i}_s}{d\tau} = -j\omega_k \tau_\sigma' \mathbf{i}_s + \frac{k_r}{r_\sigma} \left(\frac{1}{\tau_r} - j\omega \right) \boldsymbol{\psi}_r + \frac{1}{r_\sigma} \mathbf{u}_s \quad (2.9)$$

$$\boldsymbol{\psi}_r + \tau_r \frac{d\boldsymbol{\psi}_r}{d\tau} = -j(\omega_k - \omega) \tau_r \boldsymbol{\psi}_r + l_h \cdot \mathbf{i}_s \quad (2.10)$$

in which $\tau_s = \frac{l_s}{r_s}$, $\tau_r = \frac{l_r}{r_r}$, $\sigma = 1 - \frac{l_m^2}{l_s l_r}$, $l_s' = \sigma l_s$, $l_r' = \sigma l_r$, $k_r = \frac{l_h}{l_r}$, $k_s = \frac{l_h}{l_s}$, $\tau_s' = \frac{\sigma l_s}{r_s}$, $\tau_r' = \frac{\sigma l_r}{r_r}$, $r_\sigma = r_s + r_r \cdot k_r^2$ and $\tau_\sigma' = \frac{\sigma l_s}{r_\sigma}$. Figure 2.2 shows the corresponding signal flow graph. In this work, double lines represent complex values.

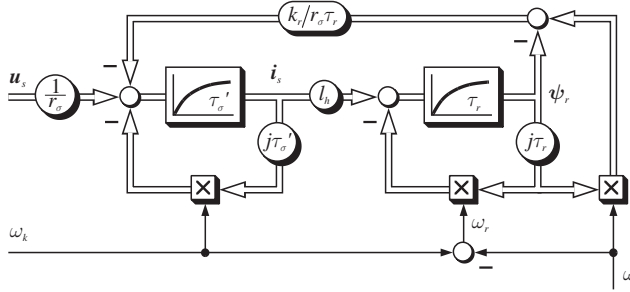
磁场定向控制的基本思想：首先将三相绕组中的三相电流 i_{sa} , i_{sb} 和 i_{sc} 变换为一个复数空间矢量 \mathbf{i}_s ，所使用的是类似于方程 (2.1) 的变换形式。在一个以任意速度 ω_k 旋转的坐标系中，异步电机可通过方程组 (2.5) - (2.8) 进行描述 [70, 71]。

对于电流控制来说，正如前面介绍串级控制中的内层回路所实现的那样，选择 \mathbf{i}_s 和 $\boldsymbol{\psi}_r$ 作为状态变量是一个可行的方案。基于 [56, 57]，经适当变换后可得到微分方程 (2.9) 和 (2.10)：

$$\mathbf{i}_s + \tau_\sigma' \frac{d\mathbf{i}_s}{d\tau} = -j\omega_k \tau_\sigma' \mathbf{i}_s + \frac{k_r}{r_\sigma} \left(\frac{1}{\tau_r} - j\omega \right) \boldsymbol{\psi}_r + \frac{1}{r_\sigma} \mathbf{u}_s \quad (2.9)$$

$$\boldsymbol{\psi}_r + \tau_r \frac{d\boldsymbol{\psi}_r}{d\tau} = -j(\omega_k - \omega) \tau_r \boldsymbol{\psi}_r + l_h \cdot \mathbf{i}_s \quad (2.10)$$

其中， $\tau_s = \frac{l_s}{r_s}$, $\tau_r = \frac{l_r}{r_r}$, $\sigma = 1 - \frac{l_m^2}{l_s l_r}$, $l_s' = \sigma l_s$, $l_r' = \sigma l_r$, $k_r = \frac{l_h}{l_r}$, $k_s = \frac{l_h}{l_s}$, $\tau_s' = \frac{\sigma l_s}{r_s}$, $\tau_r' = \frac{\sigma l_r}{r_r}$, $r_\sigma = r_s + r_r \cdot k_r^2$ 和 $\tau_\sigma' = \frac{\sigma l_s}{r_\sigma}$ 。图 2.2 给出了对应的复数电机模型，其中，带箭头的双线代表复数，单线代表标量。



2.2: Komplexes Maschinenmodell/Complex machine model/复数电机模型

Als Koordinatensystem wählt man nun ein mit dem Drehfeld umlaufendes System, dessen reelle Achse in Richtung des Feldes liegt. Den Ständerstrom i_s zerlegt man mit Hilfe dieses Koordinatensystems in einen drehmomentbildenden Anteil i_{sq} und einen feldbildenden Anteil i_{sd} . Die beiden Stromkomponenten lassen sich unabhängig voneinander regeln, wodurch eine der Gleichstrom-Nebenschlussmaschine gleichwertige Dynamik erzielt werden kann. Zerlegt man die beiden komplexen Gln. (2.9) und (2.10) in Real- und Imaginärteil, so erhält man vier skalare Formeln, die das Verhalten der Asynchronmaschine beschreiben. Dabei wird wegen der Feldorientierung $\omega_k = \omega_s$ gesetzt; aus dem gleichen Grund muss $\psi_{rq} = 0$ sein. Der entsprechende skalare Signalfussplan kann Abb. 2.3 entnommen werden.

$$i_{sd} + \tau_\sigma' \frac{di_{sd}}{d\tau} = \omega_s \tau_\sigma' i_{sq} + \frac{k_r}{r_\sigma \tau_r} \psi_{rd} + \frac{1}{r_\sigma} u_{sd} \quad (2.11)$$

$$i_{sq} + \tau_\sigma' \frac{di_{sq}}{d\tau} = -\omega_s \tau_\sigma' i_{sd} - \frac{k_r}{r_\sigma} \omega \psi_{rd} + \frac{1}{r_\sigma} u_{sq} \quad (2.12)$$

$$\psi_{rd} + \tau_r \frac{d\psi_{rd}}{d\tau} = l_h i_{sd} \quad (2.13)$$

$$0 = -(\omega_s - \omega) \tau_r \psi_{rd} + l_h i_{sq} \quad (2.14)$$

Gl. (2.14) ist für die Regelung ohne Bedeutung und wird daher im Weiteren nicht mehr berücksichtigt. Sie beschreibt aber die Bedingung für Feldorientierung, d. h. mit ihrer Hilfe kann der Lagewinkel des Feldkoordinatensystems gegenüber dem ortsfesten Statorkoordinatensystem bestimmt werden. Dazu löst man (2.14) nach der Schlupffrequenz oder Rotorfrequenz¹ $\omega_r = \omega_s - \omega$ auf und erhält

$$\omega_r = (\omega_s - \omega) = \frac{l_h i_{sq}}{\tau_r \psi_{rd}} \quad (2.15)$$

¹ Unter Rotorfrequenz ist *nicht* die mechanische Drehfrequenz des Rotors, sondern die Drehfrequenz der Ströme im Rotor zu verstehen.

Now a coordinate system rotating with the speed of the rotating field is selected as the base coordinate system with the real axis aligned in the direction of the field. With the help of this new coordinate system, the stator current space vector \mathbf{i}_s can be divided into a torque-producing component i_{sq} and into a flux-producing component i_{sd} . As both of these current components can be controlled independently of each other, dynamic control similar to a shunt-wound DC machine can be achieved. If the complex equations (2.9) and (2.10) are divided into their real and imaginary parts, four scalar equations (2.11) to (2.14) which describe the dynamic behavior of the induction machine can be obtained. Because of field orientation, $\omega_k = \omega_s$ is set; for the same reason, $\psi_{rq} = 0$ is set. The appropriate signal flow graph can be seen in Figure 2.3.

Equation (2.14) is of no significance for the control and hence, it is not considered for further analysis. However, it describes the condition for field orientation. i. e. the position angle of the field coordinate system with respect to the fixed stator coordinate frame can be determined with it. Therefore, equation (2.14) is resolved for the slip or rotor frequency¹ $\omega_r = \omega_s - \omega$ and after doing this, the following equation can be obtained:

$$\omega_r = (\omega_s - \omega) = \frac{l_h i_{sq}}{\tau_r \psi_{rd}} \quad (2.15)$$

现选择一个随旋转磁场转动的两相复数坐标系作为参考坐标系，实轴方向与磁场方向保持一致。那么，借助于此复数坐标系可将定子电流 \mathbf{i}_s 分解为转矩分量 i_{sq} 和励磁分量 i_{sd} 。通过对上述两个分量进行独立控制，即可获得类似于直流电机的动态性能。如果进一步将复数方程 (2.9) 和 (2.10) 分解为实部和虚部形式，即可得到常见的用于描述异步电机特性的四个标量方程 (2.11) – (2.14)。考虑到磁场定向，则有 $\omega_k = \omega_s$ ，而且 $\psi_{rq} = 0$ 。对应的标量电机模型信号流图如图 2.3 所示。

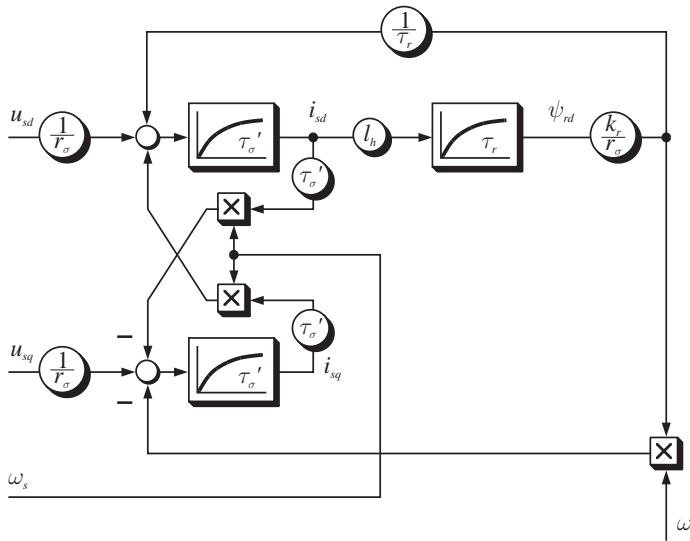
方程 (2.14) 对于实际控制来说并无用处，接下来将不再讨论。但是，方程 (2.14) 描述的是用于磁场定向的条件。换言之，基于此方程才可确定磁场坐标系相对于固定定子坐标系的位置角。为此，需将上述方程向转差频率或转子频率¹ $\omega_r = \omega_s - \omega$ 侧变换，从而有：

$$\omega_r = (\omega_s - \omega) = \frac{l_h i_{sq}}{\tau_r \psi_{rd}} \quad (2.15)$$

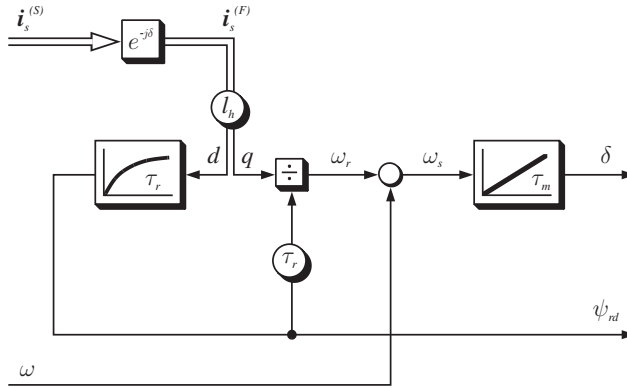
¹ The rotor frequency is *not* the mechanical rotating frequency but the frequency of the currents flowing in the rotor.

¹ 转子频率所指的并非是转子的机械旋转频率，而是转子电流的旋转频率。

Den für die Berechnung obiger Beziehung notwendigen Wert von ψ_{rd} erhält man aus der Differentialgleichung (2.13). Addiert man nun die so ermittelte Drehgeschwindigkeit ω_r zur bekannten mechanischen Drehgeschwindigkeit ω , so ergibt sich die Statorfrequenz ω_s . Die Integration von ω_s liefert den Feldwinkel δ . Der entsprechende Signalfflussplan kann Abb. 2.4 entnommen werden.



2.3: Skalares Maschinenmodell (Feldkoordinaten);
 Scalar machine model (Field coordinates);
 标量电机模型 (磁场坐标系)



2.4: Rotormodell/Rotor model/转子模型

The value of ψ_{rd} required for the calculation of the above relationship can be obtained from the differential equation (2.13). By adding the calculated slip speed ω_r to the mechanical rotating speed ω of the rotor, the stator speed ω_s results. The integration of ω_s provides the field angle δ . The overall signal flow graph of the induction machine can be seen in [Figure 2.4](#).

计算上述表达式所必需的 ψ_{rd} 可由微分方程 (2.13) 求得。若将此时得到的转速 ω_r 加上已知的机械转速 ω ，即可得到定子频率 ω_s 。定子频率的积分即为磁链角 δ 。相应的转子模型信号流图如图 2.4 所示。

3 Kaskadenregelung mit PI-Regler/Cascade Control with PI-controllers/ 基于 PI 控制器的串级控制

Zu Beginn der Untersuchungen ist es sinnvoll, zunächst eine Vergleichsmöglichkeit zu schaffen. Dazu wird eine feldorientierte Regelung einer Asynchronmaschine mit konventionellen PI-Reglern für Strom und Drehzahl in Kaskade realisiert.

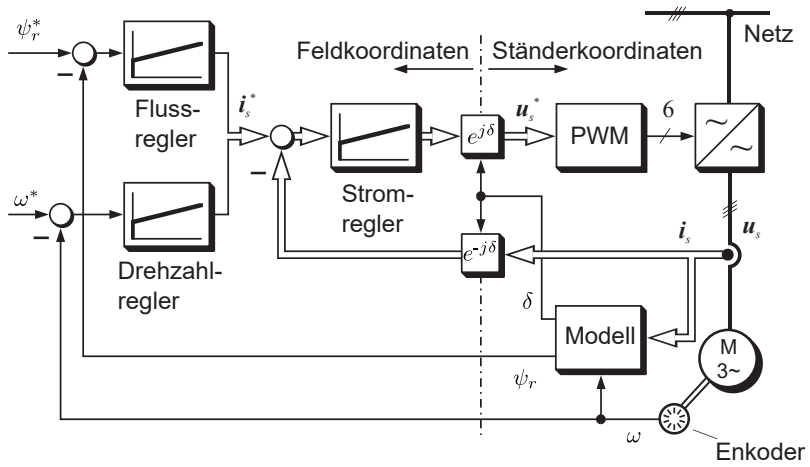
Konventionelle PI-Regler können immer nur eine einzige Größe mit Hilfe einer einzigen Stellgröße regeln; man bezeichnet diese Reglertypen als SISO-Regler (Single Input, Single Output). Im Gegensatz dazu stehen Reglerstrukturen, die mehrere Regel- und Stellgrößen gleichzeitig verarbeiten; diese werden als MIMO-Regler (Multiple Input, Multiple Output) oder mehrdimensionale Regler bezeichnet. Sollen aber nun bei einer PI-Antriebsregelung sowohl die Ströme als auch die Drehzahl geregelt werden, so muss die Regelung als Kaskadenregelung ausgeführt werden. Abb. 3.1 zeigt die typische Struktur einer feldorientierten Antriebsregelung mittels Kaskade. Die innere Regelschleife wird durch die beiden Stromregler für feld- und drehmomentbildenden Statorstrom gebildet. Um die Übersichtlichkeit des Signalflussplans zu verbessern, sind die beiden Stromregler als ein einziger, komplexwertiger Stromregler eingezeichnet. Übergeordnet befinden sich der Drehzahl- und der Flussregelkreis.

Before investigating in predictive control of induction machines, it is meaningful to create a basic control scheme for comparison first. Therefore, field-oriented control of an induction machine with conventional PI-controllers for current and speed is realized in a cascaded structure.

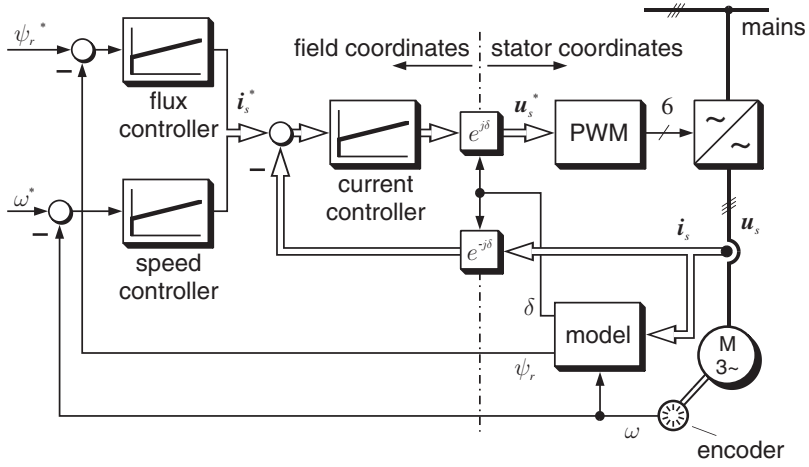
Conventional PI-controllers can only control one actuating variable with the help of one reference variable. Controllers of this type are called SISO (single input, single output) controllers. In contrast to this, there are controller structures which can control several actuating variables at the same time; these are called MIMO (multiple input, multiple output) controllers or multidimensional controllers. Thus, if in an electrical drive the current as well as the speed has to be controlled with PI-controllers, the control structure must be implemented as a cascaded control. Figure 3.2 shows the typical structure of field oriented control with cascaded PI-controllers. The internal control loop is formed by both current controllers for the field- and for the torque-producing stator current components. In order to improve the clarity of the signal flow graph, both current controllers are represented by one single complex variable current controller. The speed and flux control loops are superordinated.

在任何研究工作的开始，首先确定一个可行的对比方案是有必要且有意义的。为此，本书选择使用串级控制方式下实现的异步电机转速和电流控制作为对比方案，也即结合传统 PI 控制器实现的磁场定向控制。

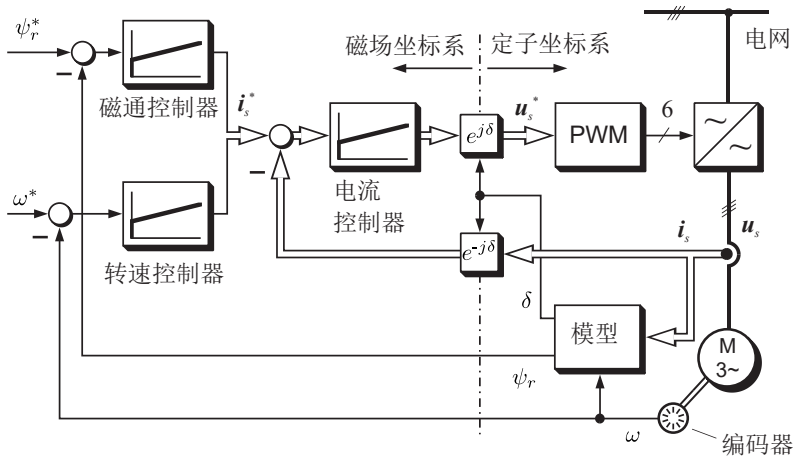
传统 PI 控制器借助单一控制变量仅能控制单个变量，人们称这一类型的控制器为单输入单输出控制器 (或 SISO-控制器)。此外，也存在可以同时处理多个控制变量和被控变量的控制器结构，通常称之为多输入多输出控制器 (或 MIMO-控制器)，或多维控制器。但是，如果要求基于 PI 的驱动控制既可控制电流，也可控制转速，那么则必须采用串级控制方式。图 3.3 给出了串级控制结构下基于 PI 控制器的磁场定向驱动控制的信号流图，内层控制回路是由两个分别用于控制励磁电流分量和转矩电流分量的电流控制器所构建。为了提高信号流图的可读性，图中使用了一个复数形式的电流控制器来代替上述两个控制器。此外，转速和磁链控制回路位于外层。



3.1: Feldorientierte Antriebsregelung mit PI-Reglern



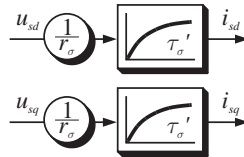
3.2: Field-oriented drive control with PI-controllers



3.3: 基于 PI 控制器的磁场定向驱动控制

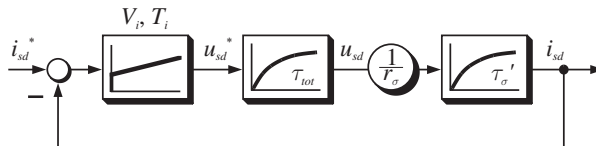
3.1 Stromregelung/Current control/电流控制

Zur Realisierung der Stromregelung geht man für den Reglerentwurf von den skalaren Gln. (2.11) und (2.12) aus. Da die Regelung mit einfachen PI-Reglern ausgeführt werden soll, bietet es sich an, die Rückwirkung des Flusses ψ_{rd} auf die Ströme sowie die Kreuzkopplung der Ströme untereinander in Gl. (2.9) zu vernachlässigen. Abb. 3.4 zeigt das entsprechende Ergebnis. Die physikalischen Größen der bei den Untersuchungen, die dieser Dissertation zugrunde liegen, verwendeten Maschine sind im Anhang D angegeben.



3.4: Vereinfachtes Modell für die Stromregelung (Feldkoordinaten);
Simplified model for current control (field coordinates);
用于电流控制的简化模型 (磁场坐标系)

Neben den Zeitkonstanten der Statorwicklungen muss zur Auslegung des PI-Reglers noch die Totzeit des Umrichters berücksichtigt werden; sie wird durch ein weiteres PT₁-Glied mit $\tau_{tot} = 1,5 \tau_0$ approximiert. Der komplette Regelkreis für den feldbildenden Strom kann Abb. 3.5 entnommen werden. Da die beiden Regelstrecken für feld- und drehmomentbildenden Strom identisch sind, ist das Signalflussbild der Stromregelung für i_{sq} dasselbe. Die beiden wirksamen Zeitkonstanten τ_{tot} und τ'_σ unterscheiden sich um einen Faktor von etwa 50, so dass zur ersten Reglerauslegung das symmetrische Optimum anzuwenden ist [38].



3.5: Stromregelkreis mit PI-Regler;
Current control loop with PI-controller;
带有 PI 控制器的电流控制闭环

3.1 Current control

For the realization of the current control, the stator equations (2.11) and (2.12) are regarded at first. Since current control has to be implemented with simple PI-controllers, it makes sense to neglect the effect of the flux ψ_{rd} on the currents as well as to neglect the cross coupling between the two current components in equation (2.9). Figure 3.4 shows the resulting structure. The actual parameters of the induction machine used for the investigations in this book are given in appendix D.

Apart from the time constants of the stators windings, the dead time of the inverter must be taken into consideration while designing the PI-controller; it is approximated with another PT₁-block with the time constant $\tau_{tot} = 1.5 \tau_0$. The complete control loop for the flux producing current component can be seen in Figure 3.5. Since both current control loops for the field- and for the torque-producing current components are identical, the signal flow graph for the control of i_{sq} is the same as the one for i_{sd} . The two effective time constants τ_{tot} and τ_{σ}' differ from each other by a factor of about 50, thus, the controller has to be designed according to the symmetric optimum [38].

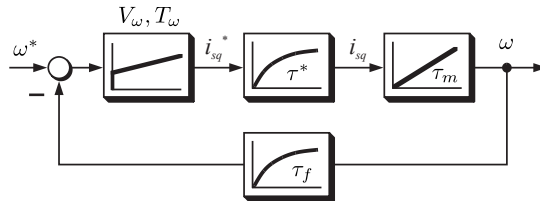
3.1 电流控制

为了实现电流控制，在进行控制器设计时需考虑标量方程 (2.11) 和 (2.12)。由于使用简单的 PI 控制器来实现上述控制，则有必要忽略磁链分量 ψ_{rd} 对电流的反馈作用，以及方程 (2.9) 中的电流交叉耦合影响。图 3.4 给出了忽略上述因素后用于电流控制的简化信号流图。本书研究过程所使用的电机参数见附录 D。

除定子绕组时间常数之外，在 PI 控制器设计时还需考虑变流器的死区时间，它可近似为一个 $\tau_{tot} = 1.5 \tau_0$ 的额外 PT₁-环节。带有 PI 控制器的励磁电流控制闭环如图 3.5 所示。由于用于励磁电流分量和转矩电流分量的被控对象是一致的，所以适用于 i_{sq} 的转矩电流信号流图与 i_{sd} 情况完全一致。由于两个有效的时间常数 τ_{tot} 和 τ_{σ}' 之间相差大约 50 倍，所以采用对称最优方式 [38] 进行控制器设计。

3.2 Drehzahlregelung/Speed control/转速控制

Die Auslegung der äußeren Drehzahlregelschleife erfordert zuerst die Annäherung des Verhaltens des inneren Stromregelkreises durch ein PT_1 -Glied. Die Zeitkonstante τ^* beschreibt dabei die Dynamik der gesamten, geschlossenen unterlagerten Regelschleife. Die mechanische Zeitkonstante des Versuchsaufbaus kann experimentell ermittelt werden. Da das Drehzahlsignal wegen der Verwendung eines niedrigauflösenden Inkrementalgebers durch den relativ hohen Quantisierungsfehler stark springt, muss es mit einem Tiefpassfilter im Rückführungszeitweig geglättet werden; hierfür wird ein einfaches PT_1 -Glied verwendet. Abb. 3.6 zeigt den kompletten Drehzahlregelkreis. Da es sich um eine Regelstrecke aus einem PT_1 -Glied und einem Integrierer handelt, muss auch hier das symmetrische Optimum angewandt werden.



3.6: Drehzahlregelkreis mit PI-Regler;
Speed control loop with PI-controller;
带有 PI 控制器的电流控制闭环

3.3 Flussregelung/Flux control/磁通控制

Auf eine explizite Flussregelung wurde verzichtet; statt dessen wurde ein fester Sollwert für den feldbildenden Statorstromanteil i_{sd} vorgegeben. Durch Wahl von $i_{sd}^* = 0,33$ kann in erster Näherung davon ausgegangen werden, dass die Maschine konstant mit Nennfluss erregt wird.

3.4 Experimentelle Ergebnisse/Experimental results/实验结果

Experimentelle Ergebnisse der PI-Kaskadenregelung sind im Vergleich mit der GPC-Regelung in Kap. 6.3.3 auf Seite 119 ff. für den Strom- bzw. auf Seite 127 ff. für den Drehzahlregelung zu finden. Weitere Messwerte zeigt Kap. 9.5.5 auf Seite 327 ff., wo die PI-Regelung mit einer direkten modellbasierten Prädiktivregelung mit expliziter Lösung des Optimierungsproblems verglichen wird.

3.2 Speed Control

The design of the outer speed control loop requires the approximation of the behavior of the internal current control loop with a first order transfer function. Thereby, the time constant τ^* describes the dynamics of the complete inner current control loop. The mechanical time constant of the experiment setup can be determined experimentally. Since the speed feedback signal bounces in discrete steps if a low resolution incremental encoder with relative high quantization error is used, it must be smoothed with a low-pass filter in the feedback path; this is achieved with the use of a simple first order transfer function element. [Figure 3.6](#) shows the complete speed control loop. Since it is a control system with a first order transfer function and an integrator, the symmetrical optimum has to be applied here, too.

3.3 Flux Control

Instead of an explicit flux control loop, a steady reference value for the flux producing component of the stator current i_{sd} used. By setting $i_{sd}^* = 0.33$, it can be assumed that, as a first approximation, the machine is constantly excited with nominal flux.

3.4 Experimental results:

Experimental results of the cascaded PI-controllers in comparison with GPC control are presented in [Chapter 6.3.3](#) on page [119](#) et seqq. for current control and on page [127](#) et seqq. for speed control, respectively. Further results are shown in [Chapter 9.5.5](#) on page [327](#) et seqq., where the PI-controller is compared to direct model-based predictive control with explicit solution of the optimization problem.

3.2 转速控制

外层转速控制回路的设计通常以内层电流控制回路的近似为前提条件: 内层回路可近似为一个 PT_1 -环节。时间常数 τ^* 描述了整个内层控制回路的动态特性; 被控对象的机械时间常数可通过实验方式获取。由于采用了包含相对较高质量误差的低精度增量式编码器, 从而会导致转速信号跳变程度剧烈。为此, 需在反馈通道引入一个低通滤波器以平滑速度信号, 此时可使用一个简单的 PT_1 -环节。[图 3.6](#) 给出了带有 PI 控制器的电流控制闭环。由图可知, 被控对象是由一个 PT_1 -环节和一个积分器组成, 所以此处也必须使用对称优化方式。

3.3 磁通控制

本书研究工作不再实现所谓的显式磁通控制, 取而代之的是给定一个恒定的励磁电流分量 i_{sd} 。通过设置 $i_{sd}^* = 0.33$, 可近似认为电机是在额定磁通下实现的励磁。

3.4 实验结果

PI 串级控制实验结果将在[章节 6.3.3](#) 给出, 并与 GPC 控制方法作对比: [第 119](#) 页给出了电流控制对比结果, [第 127](#) 页给出了转速控制对比结果。其它实验结果将在[章节 9.5.5](#) 给出, [第 327](#) 页对比了 PI 控制和基于优化问题显式解的直接模型预测控制。

4 Prädiktive Regelungen/Predictive Control/预测控制

Die in der elektrischen Antriebstechnik zu Anfang eingesetzten linearen Regelglieder, damals meist aus Operationsverstärkern aufgebaute PID-Regler, nutzen die Information über eine eventuell vorhandene Regelabweichung, um daraus ein Stellsignal zu generieren. Eine Kenntnis über die Strecke selbst besitzt ein Regler dieses Typs nicht, sie wird lediglich zur Auslegung der Reglerparameter benötigt. Mit dem Aufkommen von preiswerten Mikrorechnern und dem damit verbundenen Einzug der digitalen Regelung in die Antriebstechnik kam bald der Gedanke auf, mit Hilfe eines mathematischen Modells der zu regelnden Strecke deren Verhalten vorzuberechnen und aus diesen vorausgerechneten Werten optimale Stellgrößen herzuleiten. Die *prädiktive* oder *vorausberechnende Regelung* war geboren.

Erste Ideen prädiktiver Regelverfahren wurden bereits in den 1960er Jahren von Emeljanov [35] veröffentlicht. Nach einer eher ruhigen Zeit im nächsten Jahrzehnt entstanden in den 80er Jahren viele für die Antriebstechnik fundamentale prädiktive Verfahren wie Direct Torque Control (DTC) [1, 113] oder Predictive Current Control [55]. Weitere Veröffentlichungen betrafen die Ankerstromregelung von Gleichstromantrieben mit Hilfe netzgeführter Stromrichter [54, 65]; dieser Forschungsbereich hat jedoch mehr und mehr an Bedeutung verloren. Nach 1990 erschienen weitere Publikationen über prädiktive Antriebsregelung; einige davon, wie [100], stellen Erweiterungen und Verbesserungen bekannter Verfahren dar, während andere Autoren, z. B. [37, 118], komplett neue Strategien vorschlugen, aber auch diese entpuppen sich in den meisten Fällen bei näherer Betrachtung als Weiterentwicklungen oder Kombinationen aus bereits veröffentlichten, fundamentalen prädiktiven Regelungsstrategien. Es ist daher sinnvoll, zunächst einmal klarzustellen, was eine prädiktive Regelung auszeichnet und welche verschiedenen grundlegenden Funktionsprinzipien es für vorausberechnende Regler gibt.

The linear PID-controllers used in electric drive technology in the beginning were mostly built with analog operational amplifiers and used the control deviation in order to generate an actuating signal. A controller of this type does not possess any knowledge about the plant itself, knowledge about the plant is only required for the controller design. With the availability of inexpensive microcomputers and digital control techniques in drive technology which were developed therefore, the idea to precalculate the plant's behavior via a mathematical model and to determine optimum values for the actuating variables from these precalculated values was born. It was the birth of *predictive or precalculating control*.

The first ideas for predictive control methods have been published in the 1960s by Emeljanov [35]. After a rather quiet period in the next decade, many predictive control algorithms, which are fundamental for drive technology, like Direct Torque Control (DTC) [1, 113] or Predictive Current Control [55] were developed in the 1980s. Further publications concerned the control of the armature current of DC drives with the help of line-commutated converters [54, 65]; however, this field of research has lost more and more of its significance. After 1990, further publications about predictive drive control appeared; some of them, like [100], show extensions and improvements of known control methods, while other authors, e. g. [37, 118], published completely new control strategies. But at the end, in most cases, these methods turned out to be only further enhancements or combinations of already published fundamental predictive control strategies. Hence, it makes sense to point out the basic functional and fundamental principles of predictive control strategies first.

在电力驱动技术发展初期，所投入使用的线性控制环节大多是由运算放大器组成的 PID 控制器，它基于实际的控制偏差来生成相应的控制信号。这一类型的控制器并不包含被控对象本身的相关信息，只需设计控制器参数。随着廉价处理器的出现，以及数字控制方法逐渐渗透至驱动技术领域，从而不断涌现新的想法：例如，借助于被控对象的数学模型来预测其行为，并根据预测结果来生成最优控制变量。于是，预测控制应运而生。

早在 20 世纪 60 年代，首个基于预测控制思想的方法已由学者 Emeljanov [35] 所发表。直至 20 世纪 80 年代，驱动控制技术领域才出现一些基础的预测方法，比如直接转矩控制 [1, 113] 或预测电流控制 [55] 等。还有一些论文探讨的是基于网侧变流器实现直流电机电枢电流控制相关的主题 [54, 65]，这一领域目前已经失去研究意义。自 1990 年始，陆续出现了一些关于预测驱动控制相关的研究成果，例如，[100] 对已有的方法进行了扩展和改进。与此同时，其他一些学者 [37, 118] 则提出了全新的控制策略。遗憾的是，经仔细考察后发现，其中的大多数均可看作是对已经发表的一些基础预测控制策略的扩展或组合。正是如此，有必要首先理清预测控制的优势，以及预测控制器所蕴含的基本原理。

Wie bereits in Kap. 1 erläutert, besitzen die im Bereich der Antriebstechnik bislang eingesetzten linearen PI-Regler in Kaskadenstruktur einige fundamentale Nachteile. Prädiktive Regler erlauben es, auf die Kaskade zu verzichten, da bei einem Prädiktivregler alle Regelgrößen des zu regelnden Systems in einer einzigen Regelschleife verarbeitet werden können.

Abb. 4.1 zeigt die typische Struktur eines Prädiktivreglers am Beispiel einer Lageregelung einer elektrischen Maschine. Die gemessenen Zustandsgrößen, nämlich der Maschinenstrom I , die Drehgeschwindigkeit ω und die Lage φ werden gleichzeitig in einem Modell des Antriebs verarbeitet. Mit Hilfe dieses Modells erhält man eine exakte Information über den aktuellen Systemzustand, die dann an den Block „Prädiktion und Berechnung“ weitergeleitet wird. Dieser Funktionsblock kann als das Herz eines Prädiktivreglers angesehen werden; er ermittelt durch Vergleich des aktuellen Systemzustands mit dem gewünschten Verhalten, also dem Sollwert, eine optimale Stellgröße. Indem diese Stellgröße über das Stellglied an die Regelstrecke ausgegeben wird, schließt sich der Regelkreis. Die Berechnung der optimalen Stellgröße im Block „Prädiktion und Berechnung“ erfolgt nach der gewünschten Optimalitätsbedingung. Hier sind selbstverständlich viele verschiedene Kriterien wie minimaler Stromfehler, minimale Stromverzerrungen o. ä. denkbar. Auch eine Bewertung des Stellaufwands kann vorgenommen werden.

Das oben geschilderte Grundprinzip ist allen Prädiktivreglern gemeinsam; Unterschiede bestehen lediglich in der Funktionsweise des Blocks „Prädiktion und Berechnung“. Aufgrund dieser Unterschiede in Prädiktion und Optimierung lassen sich die Prädiktivverfahren nach verschiedenen Kriterien einteilen, nämlich

- nach dem grundlegenden Funktionsprinzip
- nach der Prädiktionstiefe
- nach der Umrichtersteuerung

Teilt man die bisher veröffentlichten Prädiktivregler entsprechend der obigen Kriterien ein, so ergeben sich verschiedene Familien, die jeweils typische Eigenschaften aufweisen.

As already explained in Chapter 1, linear PI-controllers in cascaded structure, which are used in drive control applications, have some basic disadvantages. With predictive control, it is possible to abandon the cascaded loop since in a predictive controller all control variables can be controlled simultaneously in one single controller.

Figure 4.2 shows the typical structure of a predictive controller; as example, a position control of an electric drive is chosen. The measured state variables, namely the machine current I , the rotating speed ω and the position angle φ are processed simultaneously in a model of the drive. With the help of this model, exact information of the actual system state can be obtained, which is then transferred into a block called “prediction and

calculation”。This functional block can be regarded as the heart of a predictive control system; it determines an optimum value for the actuating variable by comparing the current machine state with the desired behavior, i. e. the reference value. By applying this actuating value to the plant via the actuator, the control loop is closed. The calculation of the optimum value for the actuating variable in the block “prediction and calculation” is done depending on the desired optimum condition, such as minimum current error, minimum current distortion or similar things. An evaluation of the efforts for changing the values of the actuating variables can be done, too.

The basic principle described above is common to all predictive control strategies; the only differences are in the functionality of the block “prediction and calculation”. Because of these differences in prediction and optimization, predictive algorithms can be classified according to different criteria, namely:

- the basic functional principle
- the prediction horizon
- the converter control

If the predictive controllers published so far are classified according to the above criteria, it can be seen that they can further be classified into different families with typical characteristics.

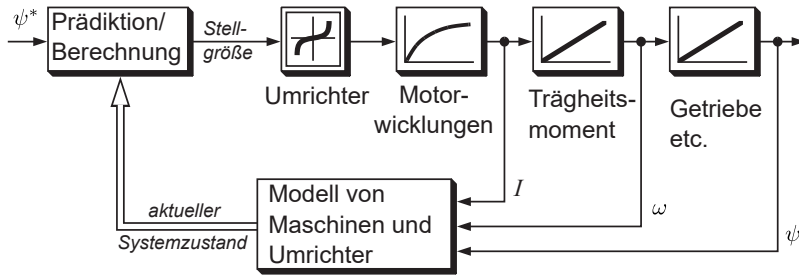
如第 1 章所述, 到目前为止, 驱动控制技术领域所使用的基于串级结构的线性 PI 控制器存在一定的缺点。预测控制器可以不使用串级结构, 因为单个预测控制器可以在一个控制回路同时处理被控对象的所有被控变量。

图 4.3 给出了一种实现电机位置控制的预测控制器典型结构。所测量的所有状态变量, 也即电机电流 I 、转速 ω 和位置角 φ , 同时输入至被控系统模型, 并基于此模型实时获取系统状态的准确信息, 然后进一步传递至预测和计算模块。此功能模块是预测控制的核心, 其功能是通过对比当前的系统状态与所期望的系统特性来生成最优的控制变量。所生成的控制变量经由执行机构作用至被控对象, 从而形成控制闭环。预测和计算模块中最优控制变量的计算是基于期望的最优条件来完成的。显然, 最优条件可以是不同的评价标准, 例如, 最小电流误差、最小电流畸变等。当然, 也可以使用控制变量本身作为评判依据。

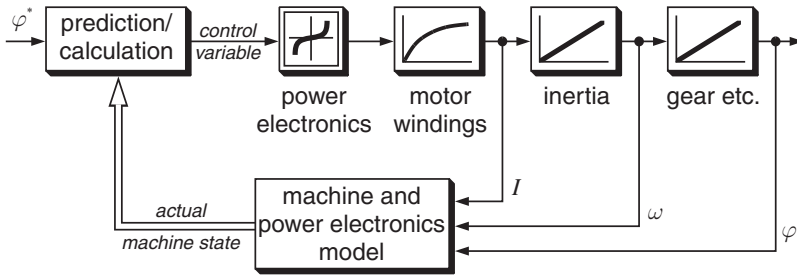
前面讲述的基本原理适用于所有的预测控制器。不同预测控制器的区别也仅仅在于预测和计算模块的工作方式不同。基于上述差别可以将预测控制方法划分为如下三种类型:

- 基于基本工作原理
- 基于预测步长
- 基于调制原理

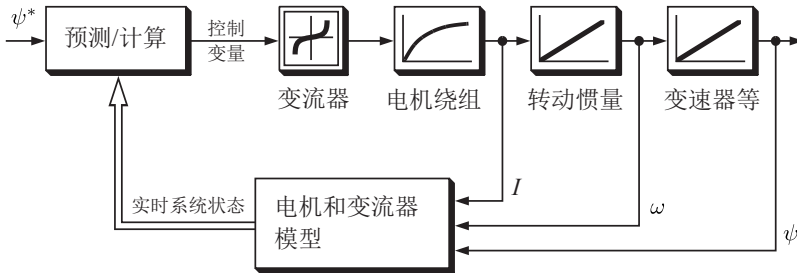
如果将已经发表的预测控制方法按照上述划分标准进行归类, 可以发现, 是可以根据不同方法所体现的典型特点对其进一步细分的, 如下所述。



4.1: Typische Struktur eines Prädiktivreglers



4.2: Typical structure of a predictive controller



4.3: 预测控制器典型结构

4.1 Einteilung nach Funktionsprinzip/Classification based on the operational principle/基于工作原理分类

Betrachtet man die den verschiedenen prädiktiven Regelalgorithmen zu Grunde liegenden Funktionsprinzipien, so lassen sie sich in drei Hauptstrategien gruppieren. Man unterscheidet zwischen *hysteresebasierten*, *trajektorienbasierten* und *modellbasierten* Verfahren, wobei diese Familien jedoch nicht scharf voneinander abgegrenzt sind, sondern der Übergang eher fließend ist.

Hysteresebasierte Prädiktivregler

Grundprinzip hysteresebasierter Regelverfahren ist es, die zu regelnde Größe innerhalb eines Toleranzbands oder einer Toleranzfläche, der sogenannten Hysterese, zu halten. Die einfachste Form eines solchen Reglers ist der bekannte Hysterese- oder Zweipunktregler. Obwohl Zweipunktregler in der Literatur im Allgemeinen nicht als Prädiktivregler bezeichnet werden, zeigen sie eindeutig deren typisches Verhalten.

Eine verbesserte Form eines Mehrgrößen-Zweipunktreglers ist die von Holtz und Stadtfeld veröffentlichte prädiktive Stromregelung [55]. Bei dieser Regelung werden die Schaltzeitpunkte durch die Grenzen des Toleranzbereichs bestimmt.

Abb. 4.4 zeigt das Funktionsprinzip; in diesem Fall wurde eine kreisförmige Hysteresefläche gewählt, deren Lage im Statorkoordinatensystem durch den Referenzraumzeiger des Statorstroms i_s^* vorgegeben wird. Erreicht der Istwert des Statorstroms i_s die Grenze der Hysteresefläche, so wird der nächste Schaltzustand des Umrichters durch Prädiktion und Optimierung ermittelt: Zuerst werden für alle möglichen Schaltzustände die zukünftigen Trajektorien des Stromraumzeigers berechnet und der Zeitraum bestimmt, wann der Iststrom den Toleranzbereich verlassen wird. Basis hierfür sind die bekannten mathematischen Differenzialgleichungen zur Beschreibung von elektrischen Maschinen. Dabei ist zu berücksichtigen, dass sich auch die Toleranzfläche selbst, durch den Referenzwert des Statorstromraumzeigers vorgegeben, innerhalb der komplexen Ebene fortbewegt. Diese Bewegung ist in Abb. 4.4 durch einen gestrichelten Kreis angedeutet.

4.1 Classification based on the operational principle

Considering the functional principles of the different predictive control algorithms, it can be seen that these can be classified into three main groups. A decision can be made between *hysteresis-based*, *trajectory-based* and *model-based* strategies. Indeed, these families are not clearly separated from each other and sometimes the transition between them is rather floating.

Hysteresis-based predictive controllers

The basic principle of hysteresis-based control strategies is to keep the value of the controlled variable within a tolerance band or a tolerance area, the so-called hysteresis. The most simple form of such a controller is the well known hysteresis or bang-bang controller. Although in literature bang-bang controllers are not considered to be predictive controllers, they do, however, clearly show their typical behavior.

An improved form of a multidimensional bang-bang controller is the predictive current control scheme proposed by Holtz and Stadtfeld [55]. Using this controller, the switching instants are determined by the limits of the tolerance band.

Figure 4.4 shows the functional principle; in this case, a circular hysteresis boundary is chosen, whose position in the stator coordinate system is given by the stator current reference vector \mathbf{i}_s^* . If the actual value of the stator current \mathbf{i}_s reaches the border of the hysteresis area, the next switching state of the inverter is selected via prediction and optimization: At first, the future trajectories of the stator current vector are precalculated for all possible switching states and the time instant at which the actual current will leave the tolerance band is determined. The basis for these calculations are the well-known mathematical differential equations of electric machines. Besides, it shall be noted that the circular tolerance region itself moves along with the stator current space vector within the complex plane. This movement is indicated with the dotted circle in Figure 4.4.

4.1 基于工作原理分类

经仔细考察不同预测控制算法的基本工作原理，可将其划分为三大类型：基于滞环的、基于轨迹的和基于模型的方法。然而，上述类型之间并不存在非常严格的界限，相反，不同类型之间的交叉也时有发生。

基于滞环的预测控制器

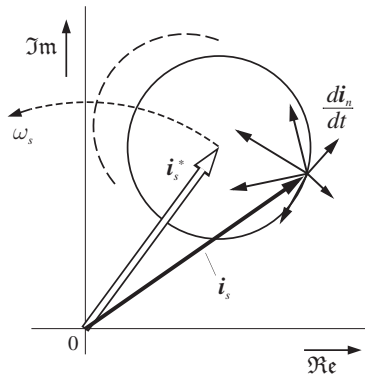
基于滞环的控制方法所依据的基本原理是保证被控变量保持在规定的允差带或允差面，也即所谓的滞环。上述类型最简单的形式是已知的滞环控制器或砰砰控制器。尽管砰砰控制器在文献中并未被称作预测控制器，但它确实体现了预测控制器的典型特征。

多变量砰砰控制器的改进形式是由学者 Holtz 和 Stadtfeld 所发表的预测电流控制 [55]，这一控制方法的开关时刻是由允差范围的边界所确定。

图 4.4 描述了其工作原理，此时选取的是圆形允差面，其在定子坐标系中的位置由定子电流空间矢量参考值 \mathbf{i}_s^* 所给定。当实际定子电流 \mathbf{i}_s 达到允差面的边界时，则通过预测和优化来确定变频器的下一个开关状态。首先，计算出所有不同开关状态所对应的定子电流未来轨迹，并进一步确定实际电流超出允差面所需的时间。完成上述计算的基础是已知的用于描述电机的数学模型。需要注意的是，定子电流参考值所给定的允差面本身在复平面内也是在不断变化的，图 4.4 中的虚线部分体现了这一变化。

Nun wird der Schaltzustand, der die längste voraussichtliche Verweildauer innerhalb der Hysterese aufweist, ausgewählt. Mit diesem Optimierungskriterium wird die Schaltfrequenz des Umrichters minimiert; selbstverständlich sind auch andere Kriterien möglich, z. B. die geringste zeitliche Abweichung.

Hysteresebasierte Verfahren haben den Vorteil, dass eine genaue Kenntnis über das zu regelnde System nicht vorhanden sein muss. Selbst bei eventuellen Modellabweichungen wird der Regelfehler durch die Hysterese innerhalb der vorgegebenen Grenzwerte gehalten. Da hierfür gewährleistet sein muss, dass der Regler beim Überschreiten des Hysteresebereichs möglichst schnell reagiert, ist es problematisch, einen hysteresebasierten Prädiktivregler als Digitalregler zu realisieren, da sich in einem solchen Fall störend bemerkbar macht, dass ein digitaler Regler das Überschreiten der Hysteresebegrenze erst beim nächsten Abtastvorgang detektiert und folglich erst mit Verzögerung eingreifen kann. Hysteresebasierte Prädiktivregelungen eignen sich daher mehr zur Realisierung durch aus Operationsverstärkern aufgebauten Komparatorschaltungen als durch Mikroprozessoren.



4.4: Hysteresebasierter Prädiktivregler nach Holtz/Stadtfield [55];
 Hysteresis-based predictive controller according to Holtz/Stadtfield [55];
 学者 Holtz 和 Stadtfield 提出的基于滞环的预测控制器 [55]

Now the switching state vector which possesses the longest stay-time within the hysteresis is selected. With this optimization criterion, the switching frequency of the inverter is minimized; of course other optimization criteria are also possible, e. g. minimum current distortion or minimum torque ripple.

Hysteresis-based strategies have the advantage that precise knowledge about the system to be controlled is not required. Even with possible model divergences the control error can be kept within the specified limit band by the hysteresis controller. To achieve this, it must always be ensured that the hysteresis controller reacts very quickly if the actual value has gone outside of the hysteresis band. This is a major problem if the hysteresis-based predictive controller is implemented in a digital processor, as the detection of the reference signal crossing the hysteresis band will be done only during the next sampling instant. It may happen that the error has, at this time, already grown to a large value. Hence, hysteresis-based predictive control is more suitable when the realization is done using analog operational amplifiers rather than microcontrollers.

根据计算结果，选择能够在允差面内停留时间最长的开关状态作为新的控制变量，这一选择的优化标准是最小化变流器的开关频率。当然，使用其他类型的评判标准也是可行的，如最小时间偏差等。

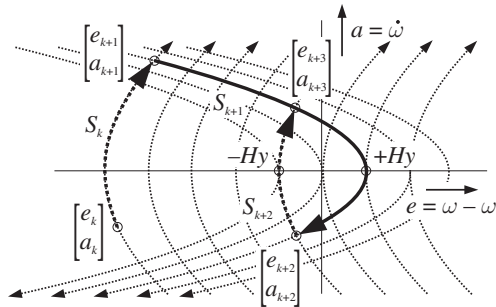
基于滞环的方法所具有的优势是无需使用精确性非常高的被控对象模型，即使在出现模型偏差的情况下，控制误差也可以通过滞环作用而保持在给定的边界范围之内。为此，必须要保证的是，控制器可以在被控变量超出滞环边界范围时尽可能快地做出反应。若以数字控制器实现基于滞环的预测控制器，则很难满足上述要求，因为超出滞环边界的情形需要等到下一个采样周期才能被检测到，从而控制变量会存在滞后或延迟。综上，基于运算放大器组成的比较电路要比基于微处理器的数字控制器更加适合于实现基于滞环的预测控制。

Trajektorienbasierte Prädiktivregler

Trajektorienbasierte Regelverfahren verfolgen den Grundgedanken, die Regelgrößen auf vorausberechneten, dem System eigenen Trajektorien laufen zu lassen. Nachdem sich das System einmal auf einer dieser Trajektorien befindet, verbleibt es aufgrund seiner eigenen Eigenschaften darauf, bis eine von außen erzwungene Änderung eintritt.

Ein erstes Verfahren dieser Art wurde bereits 1984 von Kennel publiziert [65], damals allerdings als Regelstrategie für einen netzgeführten Thyristorsteller. Einige Zeit später entstanden bekannte Regelverfahren wie die direkte Selbstregelung (Direct Self Control, DSC) von Depenbrock [33] oder Direct Mean Torque Control (DMTC) von Flach [37] zur Regelung von Asynchronantrieben. Weitere, wie Sliding Mode Control [35] oder die direkte Drehmomentregelung (Direct Torque Control, DTC) [1, 113], bestehen aus einer Mischung von hysteres- und trajektorienbasierten Grundsätzen, während die direkte Geschwindigkeitsregelung (Direct Speed Control, DSPC) von Mutschler [90] als rein trajektorienbasiert bezeichnet werden kann, auch wenn hier einige hysteresebasierte Aspekte mit hereinspielen. DSPC wird im Folgenden als Beispiel für ein trajektorienbasiertes Prädiktivregelverfahren erläutert.

Wie bei den Verfahren von Depenbrock [33] und Takahashi/Noguchi [1, 113] werden die Schaltzustände des Umrichters in die Gruppen „drehmomentaufbauend“, „drehmomentabbauend langsam“ und „drehmomentabbauend schnell“ unterteilt. Für kurze Zeitspannen können das Trägheitsmoment des Systems sowie die zeitlichen Ableitungen von Maschinen- und Lastdrehmoment als konstant angenommen werden. Das Systemverhalten lässt sich dann durch eine Schar von Parabeln in der Geschwindigkeitsfehler und Beschleunigungs-Ebene darstellen (Abb. 4.5). Diese Parabeln sind gleichzeitig die natürlichen Trajektorien des Systems.



4.5: Trajektorienbasierter Prädiktivregler nach Mutschler [90];
 Trajectory-based predictive controller according to Mutschler [90];
 基于学者 Mutschler 提出的基于轨迹的预测控制器 [90]

Trajectory-based predictive controllers

Trajectory-based control methods are based on the principle to force the system onto precalculated system trajectories. Once the system has been pushed onto one of these trajectories, it remains there because of its own properties until a change is enforced from outside.

The first trajectory-based predictive control scheme has already been published in 1984 by Kennel [65], at that time, however, being a control strategy for a line-commutated thyristor converter. Some time later, well-known control schemes like Direct Self Control (DSC) by Depenbrock [33] or Direct Mean Torque Control (DMTC) by Flach [37] for the control of induction machines were published. Some more schemes, like Sliding Mode Control [35] or Direct Torque Control (DTC) [1, 113] are a combination of hysteresis-based and trajectory-based schemes, whereas Direct Speed Control (DSPC) by Mutschler [90] can be regarded as a pure trajectory-based control scheme even if some hysteresis based aspects are included in it. In the following, DSPC will be explained as an example of trajectory-based predictive control.

Similar to the schemes of Depenbrock [33] and Takahashi/Noguchi [1, 113] the switching states of the inverter are classified into the groups “torque-increasing”, “slowly torque-decreasing” and “quickly torque-decreasing”. For short time intervals, the inertia of the system as well as the derivatives of the load torque and the machine torque can assumed to be constant values. Then the system behavior can be represented by a set of parabolas in the speed error/acceleration plane (Figure 4.5). These parabolas are also natural trajectories of the system.

基于轨迹的预测控制器

基于轨迹的预测控制方法所依据的基本思想是确保被控变量沿着预先确定的系统轨迹变化。一旦系统进入某一固定轨迹，那么系统将保持在这一由其本身特性所决定的轨迹，直至出现外界的作用强迫使其进入新的轨迹。

基于上述思想的方法首次由学者 Kennel 于 1984 年提出 [65]。显然，可以推测当年可用的控制策略还是基于传统类型的晶闸管型变流器。不久，便出现了实现异步电机控制的经典控制方法：学者 Depenbrock 提出的直接自控制 [33]，以及学者 Flach 提出的直接平均转矩控制 [37]。此外，滑模控制 [35] 或直接转矩控制方法 [1, 113] 是基于滞环和基于轨迹这两种基本思想的结合形式。然而，由学者 Mutschler 提出的直接转速控制 [90] 则可以看作是纯粹基于轨迹的方法，即使其中包含了一些基于滞环的考量。接下来，将以直接转速控制为例来阐述基于轨迹的预测控制方法。

类似于学者 Depenbrock [33] 和学者 Takahashi/Noguchi [1, 113] 所提出的方法，变流器的开关状态可以分为三种类型：可增加转矩的、可缓慢减小转矩的和可迅速减小转矩的。对于较短的时间间隔情形，可假定系统转动惯量，电机转矩以及负载转矩的时间常数保持恒定。从而，系统特性可由速度误差-加速度坐标平面内的一簇双曲线来描述，如图 4.5 所示。上述双曲线同时也是系统的自然轨迹。

Beim DSPC-Verfahren nach Mutschler [90] befinde sich das System zu Beginn in einem beliebigen Punkt $[e_k \ a_k]^T$. Gewünschter Zielpunkt ist stets der Ursprung des Koordinatensystems, d. h. es liegt keine Regelabweichung mehr vor ($e = 0$) und die Beschleunigung ist Null ($a = 0$). In diesem Zustand lässt sich das System nicht dauerhaft halten, da hierfür eine unendlich hohe Schaltfrequenz notwendig wäre. Daher wird eine Art Hysterese zwischen $-Hy$ und $+Hy$ definiert; dies ist der oben erwähnte hysteresebasierte Aspekt in DSPC. Auf diese Art und Weise kann die maximale Schaltfrequenz auf einen akzeptablen Wert reduziert werden. Ansonsten ist dieses Verfahren ein rein trajektorienbasierter Ansatz. Um in den zuvor definierten Hysteresebereich zu gelangen, wird zuerst ein drehmomentaufbauender Schaltzustand S_k gewählt. Der Systemzustand bewegt sich nun entlang der gepunkteten Parabel (Trajektorie), bis $[e_{k+1} \ a_{k+1}]^T$ erreicht ist. In diesem Punkt kreuzt die Trajektorie des Schaltzustands S_k die Parabel eines drehmomentabbauenden Schaltzustands S_{k+1} , die durch den Punkt $+Hy$ verläuft. Die Kreuzung zwischen S_k und S_{k+1} in $[e_{k+1} \ a_{k+1}]^T$ ist als optimaler Umschaltzeitpunkt bereits im Voraus berechnet worden. Daher kann zum exakten Zeitpunkt verzögerungsfrei umgeschaltet und damit der gewünschte Zustand $+Hy$ schnellstmöglich erreicht werden. Genau zum vorausberechneten Zeitpunkt wird der Umrichter in den Zustand S_{k+1} geschaltet. Der Systemzustand bewegt sich nun entlang der neuen Parabel, bis der Punkt $[e_{k+2} \ a_{k+2}]^T$ erreicht ist. Hier wird nun ein drehmomentaufbauender Schaltzustand S_{k+2} gewählt; der Systemzustand bewegt sich auf der entsprechenden Trajektorie durch $-Hy$, bis er im Punkt $[e_{k+3} \ a_{k+3}]^T$ wieder auf die Parabel des Schaltzustands S_{k+1} trifft. Im stationären Betrieb wird der Systemzustand sich entlang des Weges $+Hy - [e_{k+2} \ a_{k+2}]^T - -Hy - [e_{k+3} \ a_{k+3}]^T - +Hy$ bewegen; der Geschwindigkeitsfehler e bleibt innerhalb des vordefinierten Toleranzbands von $-Hy$ bis $+Hy$.

Trajektorienbasierte Prädiktivregelung beruht auf einer möglichst genauen Prädiktion des zukünftigen Streckenverhaltens. Regler dieses Typs benötigen daher im Gegensatz zu den hysteresebasierten Verfahren ein exaktes Modell des zu regelnden Systems. Wegen der relativ aufwändigen Vorausberechnung der Systemtrajektorien eignen sie sich besser für die Implementation in Form digitaler Regelungen durch Mikrorechner o. ä.

As proposed in the DSPC scheme according to Mutschler [90], the initial system state is assumed to be at the point $[e_k \ a_k]^T$. The desired operating point is always the origin of the coordinate system, i. e. there is no control error ($e = 0$) and the acceleration is zero ($a = 0$). The system cannot be held in this condition for long time since this would require an infinite switching frequency. Hence, a kind of hysteresis band between $-Hy$ and $+Hy$ is defined; this is described above as the hysteresis-based aspect inside DSPC. In this way, the maximum switching frequency can be reduced to an acceptable value. Apart from that, this strategy is purely a trajectory-based scheme. To reach the predefined hysteresis area, at first, a torque increasing switching state S_k is chosen. Now the system state moves along the dotted parabola (trajectory) until $[e_{k+1} \ a_{k+1}]^T$ is

reached. In this point, the trajectory of the switching state S_k crosses another parabola for a torque-decreasing switching state S_{k+1} , which will pass through the point $+Hy$. The intersection point between S_k and S_{k+1} in $[e_{k+1} \ a_{k+1}]^T$ has already been determined as the optimum switching instant in advance. Hence, at the correct time instant, switching can take place without any delay and therefore the desired state point $+Hy$ can be reached as fast as possible. Exactly in the precalculated time instant, the inverter is commutated into switching state S_{k+1} . The system state now moves along the new parabola until the point $[e_{k+2} \ a_{k+2}]^T$ is reached. At this point, a torque-increasing switching state S_{k+2} is chosen; the system state now moves along the corresponding trajectory through $-Hy$ until it reaches the parabola of the switching state S_{k+1} , again at the point $[e_{k+3} \ a_{k+3}]^T$. In steady state operation, the system state keeps moving along the path $+Hy - [e_{k+2} \ a_{k+2}]^T - -Hy - [e_{k+3} \ a_{k+3}]^T - +Hy$ and the speed error e is kept within the predefined tolerance band $-Hy$ to $+Hy$.

Trajectory-based predictive control is based on a very precise prediction of the future control system behavior. Hence, in contrast to hysteresis controllers, controllers of this type require an exact model of the system to be controlled. Because of the quite complex precalculation of the system trajectories, these methods are better suited for implementations in the form of digital controllers on microprocessors.

基于学者 Mutschler 提出的直接速度控制方法所述 [90], 假定系统初始状态处于任意一点 $[e_k \ a_k]^T$, 显然所期望的目标点始终是坐标系原点, 因为此时意味着控制偏差为零 ($e = 0$), 且加速度为零 ($a = 0$). 但是系统无法持续停留在上述理想状态, 因为它要求无限高的开关频率。正因如此, 才定义了一个在 $-Hy$ 和 $+Hy$ 之间的滞环, 这对应了前面所述的直接转速控制中的滞环因素。基于这一滞环条件可以将最大开关频率降低至可接受的范围。否则, 此方法便成了纯粹基于轨迹的方法。如图所示, 为了能够到达前面定义的滞环范围, 首先选择一个可增加转矩的开关状态 S_k , 系统状态将沿着图中虚线箭头的方向变化, 直至到达 $[e_{k+1} \ a_{k+1}]^T$ 。开关状态 S_k 的轨迹与一个可降低转矩的开关状态 S_{k+1} 对应的双曲线在该点处交叉, 新的轨迹经过点 $+Hy$ 。由于 S_k 和 S_{k+1} 的交叉点 $[e_{k+1} \ a_{k+1}]^T$ 作为最优的开关点已提前确定, 从而可以在准确的时间点无滞后地切换开关状态, 以使得系统尽可能快地到达期望的状态 $+Hy$, 也即在上述提前确定的时间点处将变流器切换至开关状态 S_{k+1} 。接下来, 系统状态将沿着新的轨迹变化, 直至到达 $[e_{k+2} \ a_{k+2}]^T$ 。此时将选择一个可增加转矩的开关状态 S_{k+2} , 然后系统会沿着相应的轨迹经过 $-Hy$, 并到达 $[e_{k+3} \ a_{k+3}]^T$, 此时再次与开关状态 S_{k+1} 对应的双曲线交叉。稳态运行时, 系统状态将沿着路径 $+Hy - [e_{k+2} \ a_{k+2}]^T - -Hy - [e_{k+3} \ a_{k+3}]^T - +Hy$ 呈周期性变化, 从而速度误差 e 可始终处于预先确定的允差范围 $[-Hy, +Hy]$ 。

基于轨迹的预测控制要求能够尽可能准确地预测对被控对象的未来状态。不同于基于滞环的方法, 该类型的控制器要求非常精确的被控对象模型。由于对系统轨迹的预测需要相对强大的计算能力, 所以更适合借助于微处理器以数字控制器的方式来实现基于轨迹的预测控制方法。

Modellbasierte Prädiktivregler

Sowohl hysteresee- als auch trajektorienorientierte Strategien verwenden den aktuellen Systemzustand, um die Stellgröße für den nächsten Abtastschritt vorauszuberechnen. Die Vergangenheit wird nicht explizit berücksichtigt; sie steckt ausschließlich im aktuellen Systemzustand. Besteht zwischen hysteresee- und trajektorienbasierten Regelverfahren in dieser Hinsicht noch eine Art Verwandtschaft, so beruhen die modellbasierten Verfahren (Model Predictive Control, MPC) auf gänzlich unterschiedlichen Ideen. Modellbasierte Verfahren können die Vergangenheit berücksichtigen und optimieren die zukünftigen Stellgrößen nicht nur für den nächsten Zyklus, sondern bis zu einem festgelegten, zukünftigen Kosten- oder Regelhorizont. Betrachtet man die Struktur von modellbasierten Reglern, so sind sie eher mit Zustandsreglern oder Kalman-Filtern zu vergleichen als mit den oben beschriebenen Prädiktivreglern.

Eine genauere Erläuterung modellbasierter Prädiktivregler ist in Kap. 5 zu finden; daher kann an dieser Stelle darauf verzichtet werden.

Stammbaum

Ordnet man die bisher veröffentlichten Prädiktivverfahren aufgrund ihrer Funktionsweisen einer der drei oben genannten Familien zu, so kann ein Stammbaum der prädiktiven Regelverfahren entworfen werden. Wie Abb. 4.6 zeigt, sind die hysteresee- und trajektorienbasierten Verfahren eng miteinander verflochten. Modellbasierte Prädiktivregelung hingegen beruht auf vollständig unterschiedlichen Grundideen und bildet eine von anderen Strategien weitgehend unabhängige Familie. Da sich die MPC-Verfahren untereinander nur relativ wenig unterscheiden, sind sie nicht alle in den Stammbaum aufgenommen worden.

Model-based predictive controllers

Both hysteresis as well as trajectory-based predictive controllers use the current system state to precalculate the value of the controlled variable for the next sampling cycle. The past is not explicitly taken into consideration as it is hidden in the actual system state. Although there is a relationship between hysteresis and trajectory-based predictive control algorithms, model-based strategies (Model Predictive Control, MPC) are based on completely different ideas. Model-based predictive control methods are able to consider the past and to optimize future values of the actuating variables, not only for the next sampling cycle, but up to a specified future cost or control horizon. Comparing the structure of model-based predictive controllers, it can be seen that they are more like state controllers or Kalman filters rather than the predictive controllers described above.

A more detailed description of model-based predictive controllers is given in Chapter 5; thus, it can be omitted at this point.

Family tree

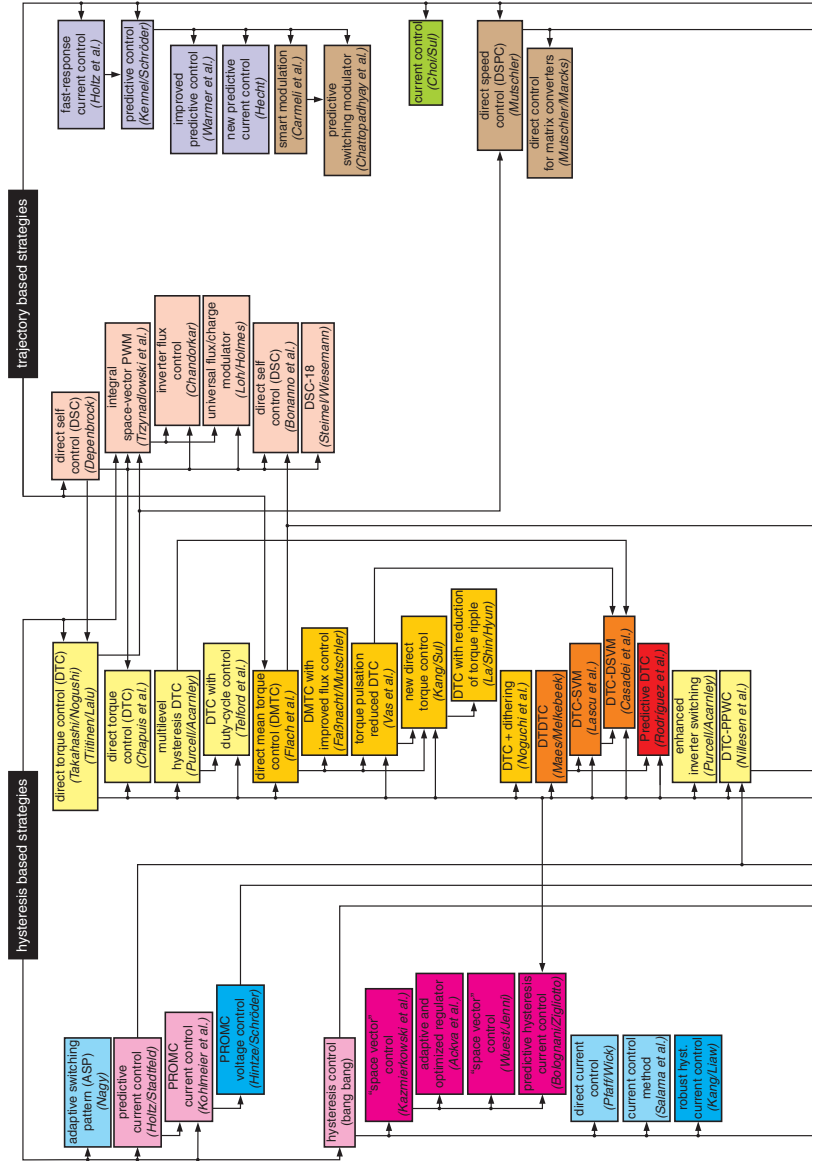
If the predictive control schemes published so far are classified into the tree families mentioned above according to their functional characteristics, a family tree of predictive control strategies can be designed. As shown in Figure 4.7 the hysteresis and trajectory-based control schemes are closely related with each other. However, model-based predictive control is based on totally different ideas and thus, it forms a class that is independent from the other ones. Since MPC strategies do not differ very much from each other, not all of them are included in the family tree.

基于模型的预测控制器

尽管基于滞环和基于轨迹的控制策略均基于实时系统状态来预测下一采样周期的控制变量，但是过去的状态却未被显式地考虑在内，二者局限于当前的系统状态。从这一角度来看，基于滞环和基于轨迹的方法具有共性，而基于模型的控制方法则采用了完全不同的思想。基于模型的预测方法同时考虑系统过去的状态来预测未来的控制变量，所谓的控制变量也并非局限于下一个采样周期，而是可扩展至一个预先设定的控制步长。经考察模型预测控制器的结构可知，相较于前面介绍的基于滞环和基于轨迹的预测控制器，它更类似于状态控制器或卡尔曼滤波器。关于基于模型的预测控制器更为详尽的描述将在第 5 章讲述。

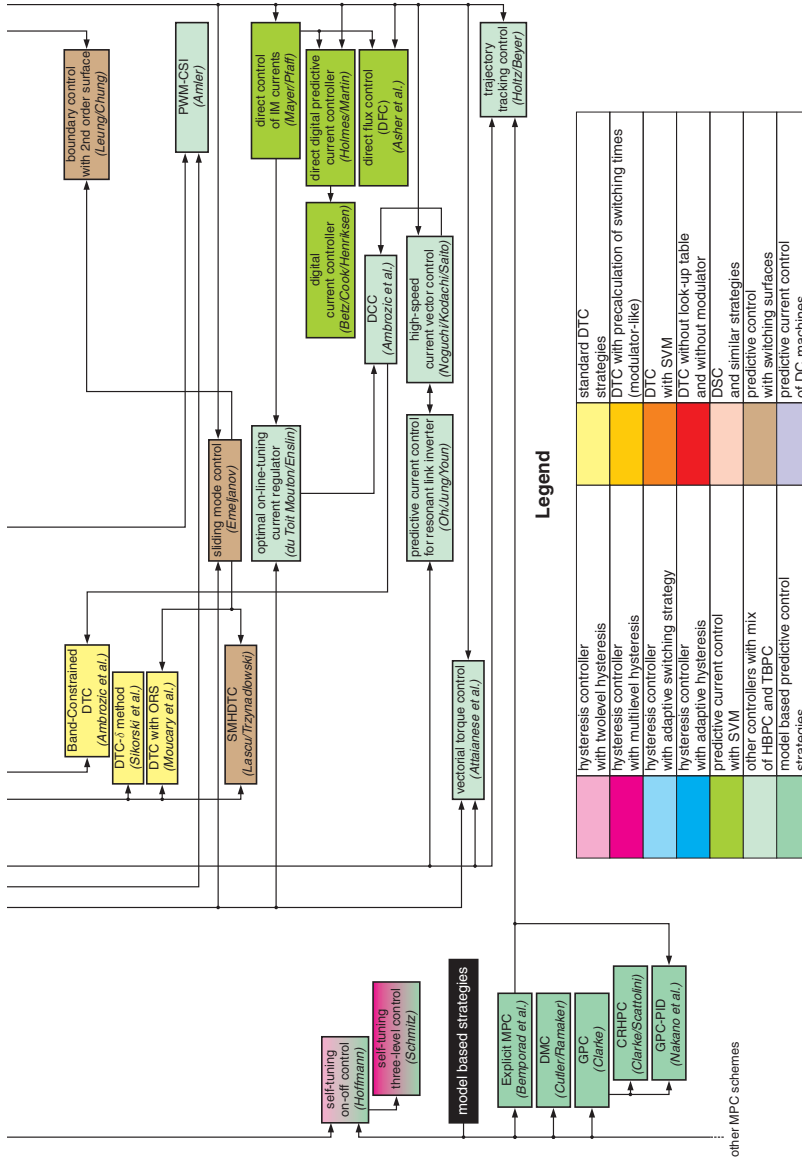
谱系图

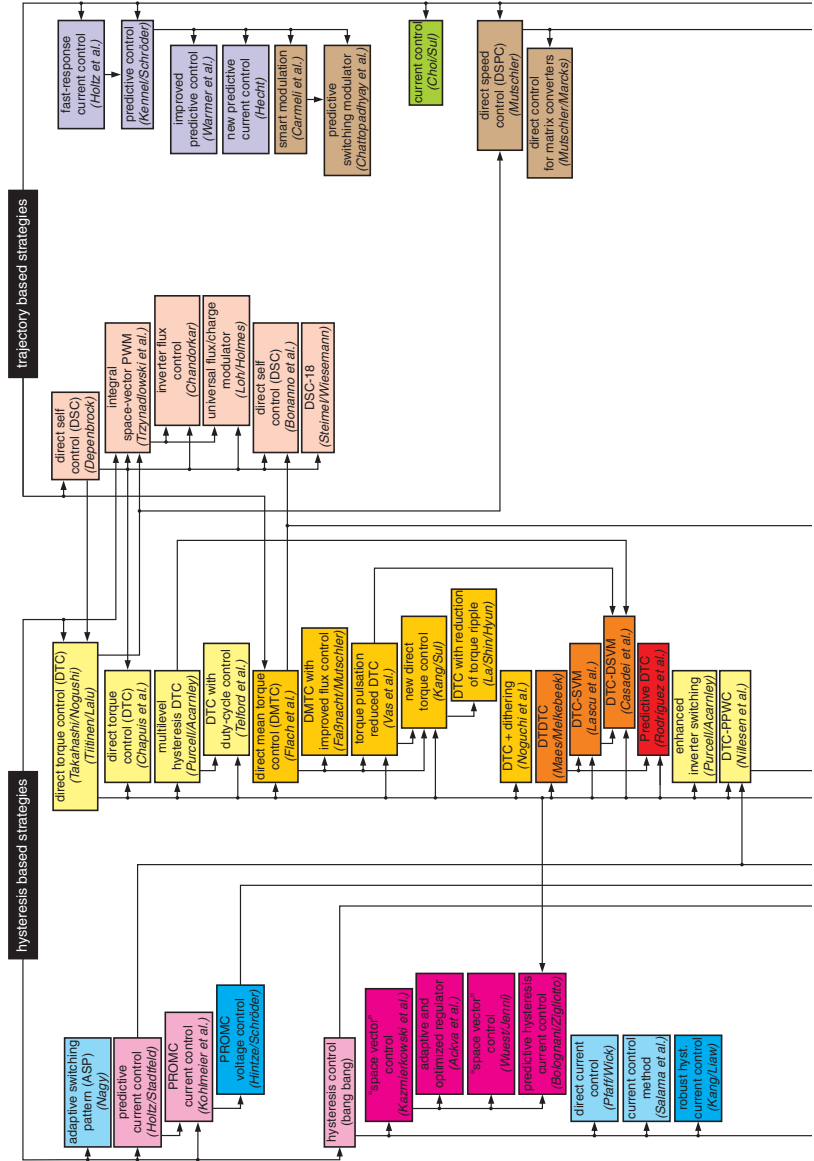
若对已发表的预测控制方法按照其工作原理的不同将其归类至前面所述的三种类型，则可设计出一幅详细的预测控制方法谱系图，如图 4.7 所示。由图可知，基于滞环和基于轨迹的方法存在一定程度上的交织，而基于模型的预测控制方法则单独形成了一个较为独立的类型，原因在于它使用的是完全不同的思想。考虑到诸多基于模型的预测控制方法之间的差别甚微，所以仅从中选取一小部分纳入谱系图。



4.6: Stammbaum prädiktiver Regelverfahren

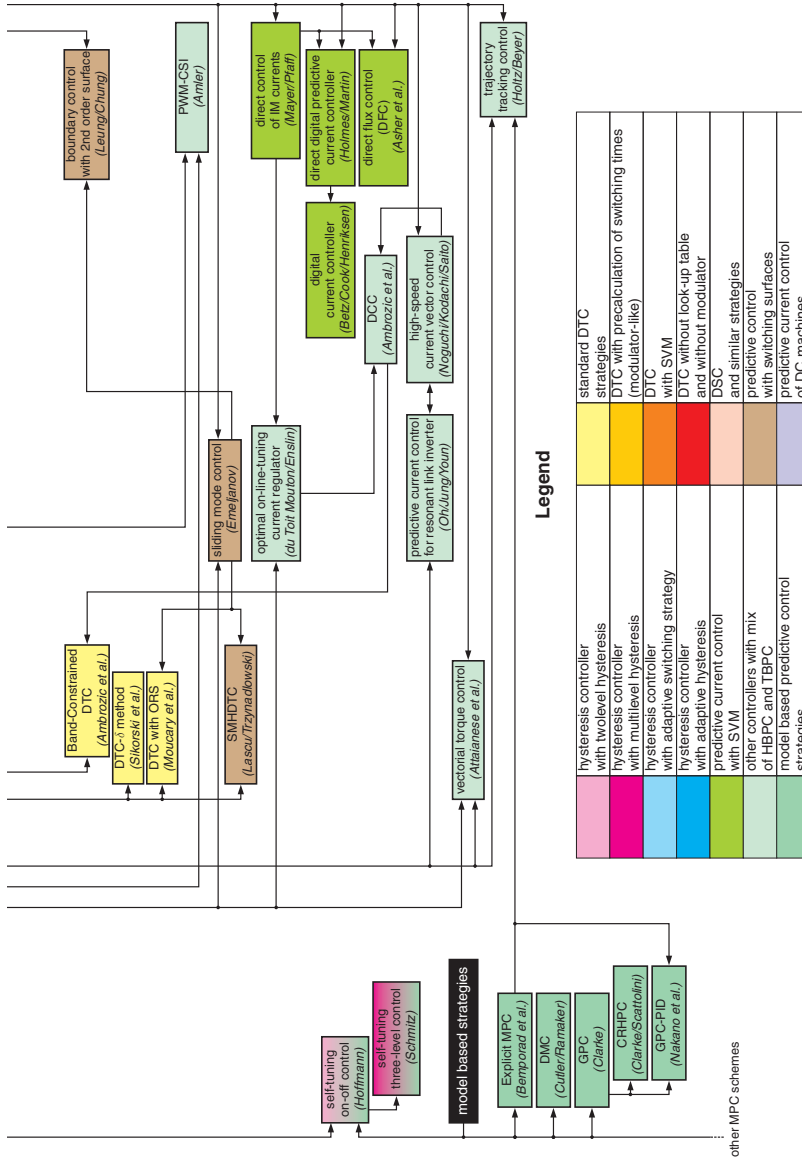
4.1 Classification based on the operational principle/基于工作原理分类





4.7: Family tree of predictive control schemes

4.1 Classification based on the operational principle/基于工作原理分类



4.2 Einteilung nach Prädiktionshorizont und Steuerungsprinzip/Classification according to prediction horizon and control principle/ 基于预测步长和控制原理分类

Eine weitere Klassifizierungsmöglichkeit besteht darin, die verschiedenen prädiktiven Regelverfahren gemäß zweier anderer Kriterien aufzuteilen. Als erstes Unterscheidungsmerkmal bietet sich die Tiefe der Vorausberechnung, der sogenannte *Prädiktionshorizont* an; eine weitere Unterteilung erfolgt nach der Art der Umrichtersteuerung, dem *Steuerungsprinzip*: Während einige Prädiktivregler sofort optimale Umrichterschaltzustände berechnen, den Umrichter also direkt ansteuern, bestimmen andere Verfahren eine wertkontinuierliche Stellgröße, die dann über einen Modulator synthetisiert werden muss, bevor sie an den Umrichter weitergeleitet wird. In Tabelle 4.1 sind beispielhaft einige Prädiktivregelverfahren und ihre Zuordnung zu den verschiedenen Familien entsprechend der obigen Unterscheidungsmerkmale angegeben.

		Steuerungsprinzip	
		mit Modulator	direkt
Prädiktionshorizont	1	Direct Control of IM Currents Direct Flux Control	Direct Torque Control Direct Self Control Direct Speed Control
	> 1	Generalized Predictive Control	Direct Model Predictive Control

4.1: Klassifizierung von Prädiktivreglern

Die weitaus meisten in der Antriebstechnik bisher untersuchten Verfahren besitzen lediglich eine Prädiktionstiefe von einem Abtastschritt. Bekannte Beispiele mit Modulator sind z. B. Direct Control of Induction Motor Currents von Mayer/Pfaff [87] oder die direkte Flussregelung nach Asher et al. [7]. Der größte Teil der einschrittigen Prädiktivregler gehört jedoch zur Gruppe der Prädiktionsverfahren mit direkter Umrichtersteuerung, darunter so prominente Vertreter wie die direkte Drehmomentregelung [1, 113] und ihre Derivate sowie die direkte Selbstregelung [33] und die direkte Geschwindigkeitregelung [90].

4.2 Classification according to prediction horizon and control principle

Another classification method for predictive control algorithms is based on two other criteria. The first distinctive feature is the depth of the precalculation, which is referred to as *prediction horizon*; another partition can be made according to the type of inverter control, the *control principle*: While some predictive controllers immediately calculate optimum inverter switching states, i. e. they control the inverter directly. Other strategies determine a value-continuous control signal which must be synthesized by a modulator before it is passed on to the inverter. Some predictive control methods and their classification in the different families according to the above differentiation criteria are given in [Table 4.2](#).

Most of the control schemes, which have been investigated in drive technology so far, have a prediction depth of only one single sampling cycle. Well known examples with modulator are e. g. Direct Current Control of Induction Motor Currents by Mayer/Pfaff [87] or Direct Flux Control proposed by Asher et al. [7]. Nevertheless, the biggest part of the one-step predictive controllers belongs to the group of the prediction schemes with direct inverter control, among them such prominent ones like Direct Torque Control [1, 113] and its derivatives as well as Direct Self Control [33] and Direct Speed Control [90].

4.2 基于预测步长和调制原理分类

另一种分类方法是基于剩余的两个标准来对预测控制方法进行归类：一是基于预测的程度，也即预测步长；二是基于变流器的调节方式，也即调制原理。一些预测控制器所计算出的最优变流器开关状态可直接用于调节变流器；然而另外一些方法确定的则是连续的控制变量，它必须借助于调制器进行处理才能用于变流器。[表 4.2](#) 给出了一些基于预测步长和调制原理分类的相关预测控制方法。

由表可知，在驱动控制技术领域，得到最广泛研究的方法也仅考虑了下一个采样周期的单步预测。基于调制器的已知方法包括学者 Mayer/Pfaff 提出的异步电机直接电流控制 [87]，学者 Asher 等提出的直接磁通控制 [7]。大多数单步预测控制器还是归类于采用变流器直接调节的预测控制方法，其中较为知名的如直接转矩控制 [1, 113] 以及变型形式，如直接自控制 [33] 和直接速度控制 [90]。

Vorausrechnende Regelungen mit einer Prädiktionstiefe von mehr als einem Abtastschritt sind ausschließlich die Verfahren der modellbasierten Prädiktivregelung. Modellbasierte Prädiktivregler nennt man daher auch *Long-range Predictive Control*, abgekürzt *LRPC*. Als einziges in der Antriebstechnik bisher eingesetztes Verfahren dieser Art ist Generalized Predictive Control [27, 28] zu nennen; zur Eignung dieser Strategie für die Antriebstechnik sind von Kennel, Linder und Linke in 2001 einige Untersuchungen durchgeführt und veröffentlicht worden [66].

Prädiktive Regelverfahren mit einem Prädiktionshorizont von mehr als einem Abtastschritt und direkter Umrichteransteuerung sind bisher zur Regelung elektrischer Antriebe nicht eingesetzt worden, was sicher auch darin begründet ist, dass das gesamte dazugehörige Thema der hybriden Systeme erst seit kurzem Gegenstand tiefergehender Forschung ist. Kap. 9 dieser Arbeit beschäftigt sich mit der Anwendung einer solchen direkten modellbasierten Prädiktivregelung auf einen elektrischen Antrieb.

4.2 Classification a. prediction horizon a. control principle/基于预测步长和控制原理分类

		control principle/控制原理	
		with modulator/带调制器	direkt/直接
Prediction horizon	1	Direct Control of IM Currents/ 异步电机直接电流控制 Direct Flux Control/ 直接磁通控制	Direct Torque Control/直接转矩控制 Direct Self Control/直接自控制 Direct Speed Control/直接速度控制
	> 1	Generalized Predictive Control/ 广义预测控制	Direct Model Predictive Control/ 直接模型预测控制

4.2: Classification of predictive control/预测控制器分类

Predictive control strategies with a prediction depth of more than one single sampling cycle are exclusively model-based predictive controllers. Thus, they are also referred to as *Long-Range Predictive Control*, abbreviated as *LRPC*. The only scheme of this kind used for drive control so far is Generalized Predictive Control [27, 28]. Its suitability for drive applications has been investigated by Kennel, Linder and Linke in 2001; the results have been published in [66].

Predictive control schemes with a prediction horizon of more than one sampling cycle and with direct inverter control have not been used for drive control so far. This can, to some extent, be justified by the fact that the whole topic of hybrid systems which is closely related to these control strategies, is a quite new field of research. Chapter 9 deals with the application of such a direct model-based predictive control to an electric drive.

预测步长大于一个采样周期的预测控制只能使用基于模型的预测控制，从而，人们也称基于模型的预测控制为多步长预测控制。到目前为止，多步长预测控制方法应用于驱动控制的也只有 GPC [27, 28]。此外，上述类型应用到驱动控制技术领域的可行性考察由学者 Kennel, Linder 和 Linke 于 2001 年完成，并以论文形式发表 [66]。

基于直接变流器调节的预测步长大于一个周期的预测控制方法仍未应用到电力驱动控制领域，其根本原因在于与上述策略密切相关的混合系统的研究主题也是近期才有了一些初期的研究。第 9 章将讨论直接模型预测控制在电力驱动控制领域的应用。

5 Modellbasierte Prädiktivregelung/Model Predictive Control/模型预测控制

Das Prinzip der *modellbasierten Prädiktivregelung* oder *Model Predictive Control*, kurz *MPC*, wurde in den 70er Jahren des letzten Jahrhunderts von industriellen Regelungstechnikern entwickelt, nachdem erste Ideen dieser Art bereits in den 60er Jahren veröffentlicht worden waren. Danach hat sich MPC hauptsächlich in der chemischen Industrie stark verbreitet und erlangte erst später auch im akademischen Bereich eine größere Aufmerksamkeit. MPC bezeichnet keinen konkreten Regelalgorithmus, sondern vielmehr eine ganze Familie von Reglertypen. Gemeinsames Merkmal aller dieser Regler ist das Prinzip, eine optimale Stellgröße durch den expliziten Gebrauch eines Modells des zu regelnden Systems und über das Minimieren eines Kostenfunktional zu berechnen.

Zur Einführung in das Thema MPC bieten sich verschiedene Aufsätze aus verschiedenen Fachzeitschriften an, die eine Einführung oder auch eine Übersicht über die modellbasierten Regelverfahren geben wollen. Empfehlenswert ist hier der Beitrag von Morari/Lee [88], in dem ein Überblick über Vergangenheit, Gegenwart und zukünftige Verbesserungsmöglichkeiten von MPC gegeben wird. Auf Formeln wird dabei zugunsten einer allgemeinverständlichen Erklärung weitgehend verzichtet. Für die an mathematischen Herleitungen interessierten Regelungstechniker bietet sich eher das ausführliche Tutorial von Rawlings [102] an.

Zu MPC gehören u. a. Verfahren wie Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Extended Horizon Adaptive Control (EHAC) und Extended Predictive Self-Adaptive Control (EPSAC) [30]. Im Gegensatz zu den gerade genannten Regelstrategien gehört Internal Model Control (IMC, s. Kap. 10.1) trotz seines Namens *nicht* zu den MPC-Reglern, auch wenn einige Autoren dies so sehen, z. B. García, Prett und Morari [40]. Seborg hingegen ordnet es richtigerweise nicht den prädiktiven, sondern „alternativen Verfahren“ zu [109]. Ein ausführlicher Vergleich verschiedener MPC-Verfahren ist bei de Keyser et al. [68] und García et al. [40] zu finden.

The principle of *model-based predictive control* or *Model Predictive Control*—abbreviated *MPC*—was introduced for industrial control applications in the 1970s after first ideas of this strategy have already been published in the 1960s. Subsequently, MPC gained importance mainly in the field of chemical industry and later on it also gained more regard in the academic area. MPC does not denote a special control algorithm, but rather a whole family of controller types. Common characteristic of all these controllers is the principle to determine an optimum value for the actuating variable by using an explicit model of the system to be controlled and by minimizing a cost function.

For a first introduction into the subject of MPC, there exist several papers from technical journals which give an introduction or also a survey about the different model-based control schemes. The article from Morari/Lee [88] is recommended here in which an overview about past, present and future improvement possibilities of MPC are presented. To present the ideas of MPC in an easily understandable manner, the article mainly avoids mathematical equations. For control engineers interested in the mathematical background of MPC, the detailed tutorial by Rawlings [102] is a good reference.

To the family of model-based predictive control belong, among other methods, schemes like Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Extended Horizon Adaptive Control (EHAC) and Extended Predictive Self-Adaptive Control (EP-SAC) [30]. In contrast to these control strategies, Internal Model Control (IMC, see Chapter 10.1) does *not* belong to the MPC class, even though its name suggests this and even if some authors see this in such a way, e.g. García, Prett and Morari [40]. Seborg, on the other hand, classifies it correctly, not as a predictive, but as an “alternative scheme” [109]. A detailed comparison of different MPC strategies is presented by de Keyser et al. [68] and García et al. [40].

基于模型的预测控制或模型预测控制，简称 MPC，其基本原理的首次提出可追溯至上世纪 60 年代。在随后的 70 年代，经由工业控制工程师继承和发扬。起初，MPC 主要在化工领域得到广泛应用，随后也在学术界逐渐获得了关注。MPC 并非意味着某一具体的控制算法，而是代表一大类别的控制器类型，该类型控制器的共同特征：基于被控系统模型预测系统未来行为，并通过最小化成本函数的方式来计算最优的控制量。

有关 MPC 主题的介绍可参考专业期刊上发表的诸多学术论文，它们旨在给出模型预测控制的发展现状。强烈推荐的是学者 Morari/Lee 发表的综述论文 [88]，该论文对 MPC 的历史发展、当前现状及未来改进的可能性均做了详细论述。值得一提的是，作者放弃了难以理解的借助于数学公式的表达方式，而使用了一种更为通俗的解释。如果是对数学推导过程感兴趣的读者或工程师，可参考学者 Rawlings 所发表的论文 [102]。

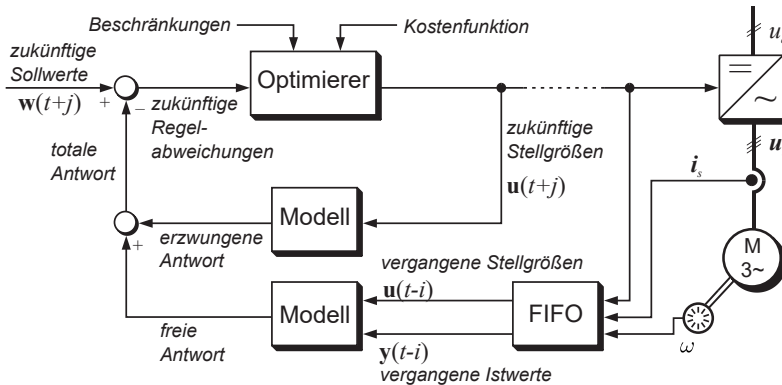
MPC 方法主要包括：动态矩阵控制、模型算法控制、扩展地平线自适应控制和扩展预测自适应控制 [30]。不同于上述控制策略，内部模型控制并不属于 MPC 控制方法，尽管其名称包含模型控制。不过一些学者如 García, Prett 和 Morari [40] 则有不同的见解，学者 Seborg 则将其恰当地归入替代性方法，而非预测性方法 [109]。不同 MPC 方法的详细对比可参考学者 de Keyser 等 [68] 和 García 等 [40] 发表的论文。

Wie bereits in Kap. 4 erwähnt, sind Regelstrukturen, die zur Familie der modellbasierten Prädiktivregler gehören, anders aufgebaut als die in der Antriebstechnik üblicherweise verwendeten Prädiktivregler. Zwar besitzen sie eine ähnliche Struktur, denn sie verwenden ebenfalls ein explizites und separat identifizierbares Modell der Regelstrecke zur Vorausberechnung des Systemverhaltens und damit auch zur Auswahl optimaler Stellgrößen; im Gegensatz zu herkömmlichen, aus dem Bereich der Antriebstechnik bekannten Prädiktivreglern, die im Allgemeinen nur einen einzigen Prädiktions Schritt vorausberechnen, wird bei MPC-Reglern das Streckenverhalten jedoch auf weitere Sicht hin berücksichtigt [67]. Hauptvorteile dieser Regler sind:

- Multivariable Strukturen sind problemlos abbildbar.
- Systembeschränkungen („constraints“) können in systematischer Weise behandelt und im Modell berücksichtigt werden.
- Phasenverschiebungsfreie Sollwertfilter können ohne weiteren Aufwand integriert werden.
- Selbsttätige Identifikation von Modellparametern ist möglich.

5.1 Funktionsprinzip/Functional principle/工作原理

Das Funktionsprinzip eines MPC-Reglers kann anhand der in Abb. 5.1 gezeigten Struktur verdeutlicht werden. Zentraler Punkt ist das Modell, welches für die Prädiktion des zukünftigen Verhaltens des zu regelnden Systems verwendet wird.



5.1: Typische Struktur eines MPC-Reglers

As already mentioned in Chapter 4, control structures belonging to the family of model-based predictive control have totally different structures than the predictive controllers commonly used in drive technology. Of course, these controllers possess a similar structure, as they also use an explicit and separately identifiable model of the controlled system for the precalculation of the system behavior and therefore also for the selection of optimum values for the actuating variables. But in contrast to conventional predictive controllers used in drive control applications which precalculate the plant's behavior only for the next sampling cycle, MPC controllers consider the future system behavior for more than one sampling cycle [67]. Main advantages of these controllers are:

- Multivariable structures are easily representable.
- System constraints can be handled systematically and can be considered in the model.
- Filtering of measured variables without phase displacement can be integrated without further effort.
- Automatic identification of model parameters is possible.

5.1 Functional principle

The functional principle of an MPC controller can be explained with the help of the structure shown in Figure 5.2. Its central part is the model which is used to predict the future behavior of the system to be controlled.

如第 4 章所述, 属于模型预测控制方法的控制器结构与驱动技术领域通常使用的基于滞环或基于轨迹的预测控制器结构有着很大的不同。假使它们拥有类似的结构, 而且均使用被控对象的显式模型来预测系统的未来行为, 并基于此选择最优控制量。但在驱动技术领域, 已知的传统方式预测控制器通常仅能够完成单步预测, 而 MPC 控制器可实现多步预测 [67]。MPC 控制器的优点包括:

- 可轻易地实现多变量结构, 换言之, 单一控制器可同时控制多个变量;
- 系统约束可以在系统层面进行处理, 并在模型中加以考虑;
- 无需额外的代价即可集成无相位偏差的参考值滤波器;
- 模型参数的自动辨识也是可能的。

5.1 工作原理

MPC 控制器的工作原理可借助于图 5.3 所示的 MPC 控制器典型结构进行阐述, 图中最为核心的部分是模型, 其功能是预测被控系统的未来行为。

Die Prädiktion besteht aus zwei Teilen:

Die freie Antwort stellt das zu erwartende Verhalten des Ausgangs $\mathbf{y}(t+j)$ des Systems unter der Annahme dar, dass keine Stellgröße auf die Strecke einwirkt.

Die erzwungene Antwort bildet die zusätzliche Komponente der Systemantwort, beruhend auf den vorausberechneten Stellgrößen $\mathbf{u}(t+j)$.

Bei linearen Systemen kann das vollständige, zukünftige Systemverhalten, die *totale Antwort*, mit Hilfe des Superpositionsprinzips als Addition der freien und der erzwungenen Antwort berechnet werden. Diese Summe wird bis zum sogenannten *Prädiktionshorizont* N_p vorausberechnet. Entsprechend dem Prädiktionshorizont gibt es auch eine Folge von zukünftigen Sollwerten, die der Systemausgang annehmen soll. Die Differenz zwischen zukünftigen Sollwerten und vorausberechneten Istwerten ergibt die zukünftigen Regelabweichungen. Ein Optimieralgorithmus berechnet aus den zu erwartenden Regeldifferenzen unter Berücksichtigung der Systembeschränkungen und der Kostenfunktion eine Folge optimaler Stellgrößen $\mathbf{u}(t+j)$. Eine einfache Steuerung würde nun die vorausberechneten Stellgrößen einfach an die Regelstrecke weiterleiten. Unter Verwendung der vergangenen Stell- und Regelgrößen bis zum *zurückliegenden Horizont* verwandelt sich das Verfahren jedoch in einen vollständigen Regelkreis. Nur das erste Element des berechneten Stellgrößenvektors – $\mathbf{u}(t)$ – wird an die Strecke angelegt, danach beginnt der ganze Ablauf aus Prädiktion, Optimierung und Regelung bei jedem Abtastschritt von neuem. Der Prädiktionshorizont wird dabei quasi weitergeschoben; dieses Prinzip nennt man *Receding Horizon Control* oder *RHC*.

Die Funktionsweise einer modellbasierten Prädiktivregelung basierend auf Receding Horizon Control entspricht einer Art „natürlicher“ Prädiktionsregelung, denn sie ist der menschlichen Verhaltensweise sehr nahe. Beim Autofahren beispielsweise schaut der Fahrer nicht direkt vor seinen Wagen, sondern blickt weit voraus und verändert die Stellgröße, also die Stellung von Gas- und Bremspedal, schon weit vor z. B. einer roten Ampel oder einer Kurve. Dabei berechnet er das Verhalten des Fahrzeugs über die vor ihm liegende Strecke bis zu einem gewissen Horizont unter Berücksichtigung zukünftiger Stellgrößen voraus, optimiert die Menge des Gasgebens oder Bremsens entsprechend seiner Optimierkriterien über diese Strecke und reagiert entsprechend. Wie bei einer echten MPC-Regelung sind auch hier unterschiedliche Optimierkriterien denkbar, die zu verschiedenen Ergebnissen führen. Wünscht der Fahrer eine möglichst kurze Fahrzeit, wird er sicher schärfer beschleunigen und bremsen, als wenn der Kraftstoffverbrauch verringert werden soll.

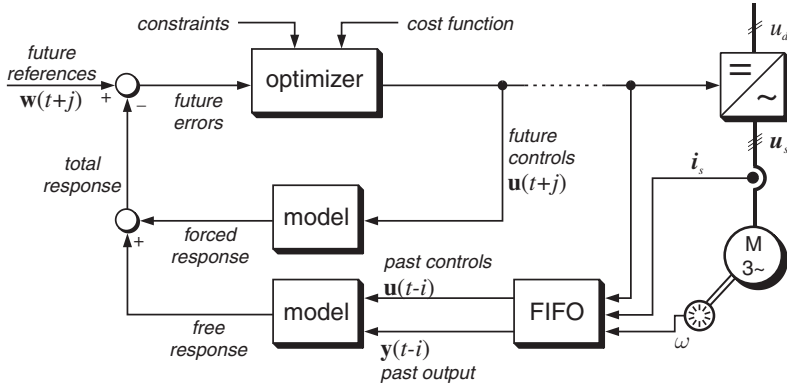
The prediction consists of two components:

The free response shows the expected behavior of the system output $\mathbf{y}(t+j)$ assuming future values of the actuating variables being equal to zero.

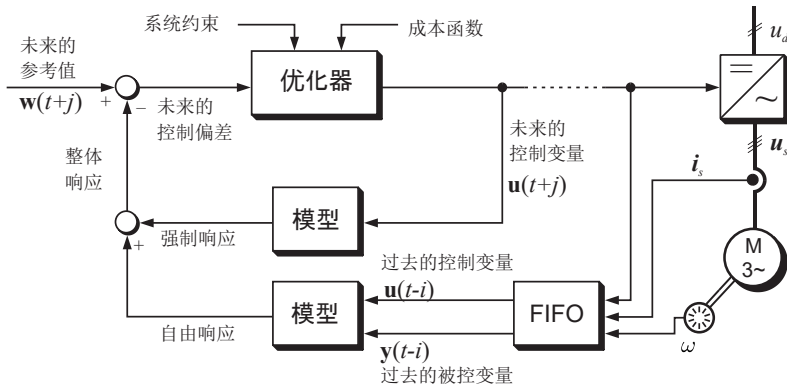
The forced response forms the additional component of the system response based on the precalculated set of future actuating values $\mathbf{u}(t+j)$.

For linear systems, the entire future system behavior, the *total response*, can be determined as the sum of the free and forced response using the superposition principle. This sum is precalculated up to the *prediction horizon* N_p . According to the prediction horizon, a set of future reference values the system output should be equal to, does also exist. The difference between future reference and precalculated actual values delivers the future control error. An optimization algorithm determines a set of optimum future actuating values $\mathbf{u}(t+j)$ from the expected error, taking system restrictions and the cost function into account. A simple open-loop control scheme would only apply this precalculated future sequence of values for the actuating variables to the system. By using past values of the output and of the actuating variables up to the *past horizon*, this method changes into closed-loop control. Only the first element of the calculated state vector – $\mathbf{u}(t)$ – is applied to the system and afterwards the whole procedure of prediction, optimization and controlling is again repeated for each sampling cycle. Hence, the prediction horizon is shifted forward; this principle is called *Receding Horizon Control* or *RHC*.

The functional principle of model-based predictive control based on Receding Horizon Control represents a kind of “natural” predictive control, as it is very close to human behavior. For example when driving a car, the driver does not look immediately in front of his car, but he looks far ahead and changes the actuating variables, e. g. the position of the steering wheel, gas pedal and brake before he approaches for instance a red traffic light or a curve. Besides, he precalculates the behavior of the car for a certain distance in front of him up to a finite horizon taking future values of the actuating variables into account, he optimizes the amount of acceleration or braking according to his optimization criteria for this distance and acts accordingly. Like in real MPC, different optimization criteria are possible, leading to different results. If the driver desires the shortest possible duration of his trip, he will accelerate and brake more rapidly than if a reduction of fuel consumption is an optimization criterion.



5.2: Typische Struktur eines MPC-Reglers



5.3: 一种 MPC 控制器的典型结构

预测包含两个部分：

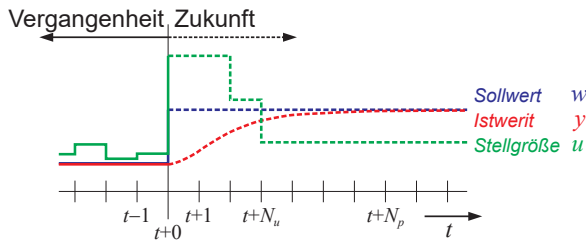
自由响应 表示的是没有控制量施加到被控对象时的期望系统输出 $\mathbf{y}(t+j)$ 。

强制响应 表示的是由控制量 $\mathbf{u}(t+j)$ 施加到被控对象而引起的额外系统响应。

对于线性系统来说，完整的未来系统响应，也称整体响应，可由自由响应和强制响应相加得到，它依据所谓的叠加原理。整体响应将逐步计算至所谓的预测步长 N_p 为止。与预测步长相对应，不可缺少的是系统未来输出的一系列参考值。未来的参考值与预测值之间的差即构成未来的控制偏差。优化算法就是基于期望的系统偏差，并结合系统的约束条件和成本函数来计算一系列最优的控制量 $\mathbf{u}(t+j)$ 。最简单的方式是可以直接将计算得到的控制量施加至被控对象，这种将过去的控制变量和被控变量考虑到过往的步数类似于一种闭环控制方法。一系列最优控制量的第一个元素 $\mathbf{u}(t)$ 施加给被控对象，接下来，则需在新的采样周期进行重新预测、优化和控制。预测步长也相应地向前推移，这一原理也称之为滚动优化控制，简称为 RHC。

基于滚动优化控制的模型预测控制工作原理非常类似于现实世界各种生物的行为机制，如人类的行为方式。例如，驾驶员在行驶过程中并非仅仅盯着其前方的车辆，而会考虑得更加全面和长远。比如，在前方出现红灯或需要转弯的情况下，驾驶员会提前做出决策，判断当前是否需要适当改变控制量，也即所谓的油门或刹车。换言之，驾驶员在行驶过程中会根据最新的路况不断地改变其驾驶行为，并尽可能考虑未来可能出现的情况，按照一定的标准来做出当前的决策，这一特性类似于预测控制方法。由于驾驶员的习惯不同，实际存在诸多不同的决策标准，从而会出现不同的效果。假使某些驾驶员希望保证尽可能短的行驶时间，而非像另一些驾驶员要保证尽可能少的燃料消耗，那么车辆将会出现更为剧烈的加速或制动过程。

Wegen der Vorausberechnung des Systemverhaltens bis zum Prädiktionshorizont entsteht bei MPC-Reglern zwangsläufig ein hoher Rechenbedarf. Durch Einführung eines sogenannten *Regelhorizonts* N_u kann der Berechnungsaufwand deutlich reduziert werden. Man geht dabei von der Voraussetzung aus, dass nach N_u Schritten ein stationärer Zustand erreicht ist und die Stellgröße daher nicht mehr geändert wird. Abb. 5.4 verdeutlicht diesen Sachverhalt. Trotz dieser Maßnahme ist zur modellbasierten Regelung eines Systems im Vergleich zu anderen Regelverfahren eine deutlich höhere Rechenleistung notwendig. Hauptgebiete dieser Algorithmen sind daher u. a. die chemische und verfahrenstechnische Industrie [19, 40], da die dort zu regelnden Prozesse für eine MPC-Regelung optimal geeignet sind. Die Zeitkonstanten liegen durchweg im Minutenbereich, wenn nicht sogar noch höher. Rechenzeitprobleme sind daher nicht zu befürchten.



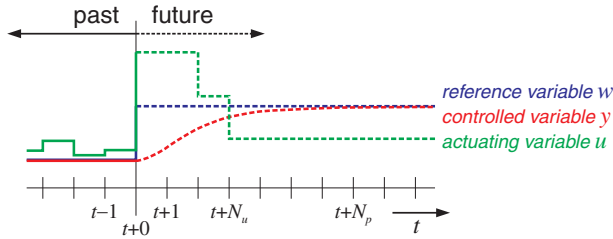
5.4: Regel- und Prädiktionshorizont beim MPC-Regler

Im Bereich der Antriebstechnik werden jedoch wesentlich höhere Abtastraten gefordert. Zur Lösung dieses Problems existieren verschiedene Ansätze [11, 18], praktische antriebstechnische Anwendungen von MPC-Verfahren sind aber außer den bereits erwähnten Untersuchungen von Kennel/Linder/Linke [66] nicht bekannt.

Due to the precalculation of the system behavior up to the prediction horizon, MPC inevitably leads to a high calculation demand. The calculation complexity can be significantly reduced with the introduction of a so-called *control horizon* N_u . After N_u steps, it is assumed that the steady state is reached and so the controller output remains constant. Figure 5.5 illustrates this situation. In spite of this modification in the control scheme, still a higher mathematical effort is necessary for model-based control of a system compared to other control algorithms. Hence, main areas of applications for MPC algorithms are, among others, the chemical and process industry [19, 40], since the processes to be controlled there are very well suited for MPC. The time constants of the whole system are rather large (in the range of minutes or even still higher). Hence, calculation time is not a problem in this case.

In the area of electrical drives, however, much higher sampling rates are needed. Several proposals to overcome this problem have been made [11, 18]; nevertheless, publi-

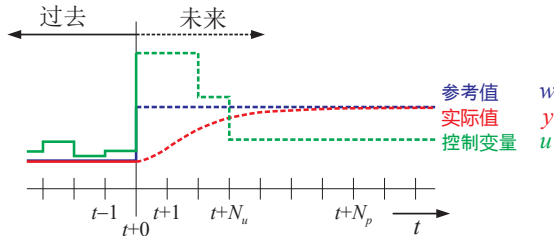
cations on practical applications of MPC strategies for drive control are not noted except for the already mentioned investigations by Kennel/Linder/Linke [66].



5.5: Definition of the control and prediction horizon

由于需要提前预测的是预测步长范围内被控系统的未来行为，MPC 控制器不可避免地要拥有强大的计算能力。通过引入所谓的控制步长 N_u ，即可明显地降低计算量。换言之，假设在经过步数 N_u 之后，系统即可达到稳定状态，接下来则无需继续改变控制量。如图 5.6 所示的 MPC 控制器的基本工作原理示意图，在 $t + N_u$ 步数之后的时间范围控制量将保持不变。然而，即使采用了上述策略，相对于其他控制方法来说，MPC 方法还是需要明显更强的计算能力。正因如此，此类型方法的主要应用领域限制在化学和过程控制行业 [19, 40]，因为这些行业中使用的被控系统的时间常数通常为分钟级，甚至是小时级，从而无需担心计算能力问题。

但在驱动技术领域则需要较高的采样频率。为解决上述问题，已有学者提出了不同的方案，如 [11, 18] 等。然而，除了前面提到的学者 Kennel/Linder/Linke [66] 的考察之外，并无将 MPC 方法实际应用驱动技术领域的具体案例。



5.6: MPC 控制器的控制步长和预测步长

5.2 Modelle/Models/模型

Neben der Kostenfunktion ist das Modell, mit dessen Hilfe das Streckenverhalten vorausgerechnet wird, einer der Kernpunkte eines MPC-Verfahrens. Die meisten zu regelnden Systeme sind zeitkontinuierlicher Art, so dass sich eigentlich eine Beschreibung des Systems im Zeit- oder Laplacebereich anbieten würde. Im Gegensatz zu den Regelstrecken sind jedoch die Regler selbst vielfach zeitdiskret, da sie im Allgemeinen mit Hilfe von Digitalrechnern realisiert werden. In den meisten Fällen ist es ausreichend, zur Reglerauslegung auch diesen durch ein zeitkontinuierliches Modell zu ersetzen, zu parametrisieren und dann als zeitdiskreten Digitalregler zu realisieren. Vorteilhafter ist es jedoch, den diskreten Charakter des Reglers bereits im Modellansatz zu berücksichtigen und folglich eine zeitdiskrete Systembeschreibung zu verwenden. Dies impliziert, dass auch die Regelstrecke selbst in zeitdiskreter Form beschrieben wird. Eine laplacetransformierte Übertragungsfunktion scheidet damit aus, da in diesem Fall die zeitliche Diskretisierung verloren geht. Eine Beschreibung des Systems im Zeitbereich ist selbstverständlich stets möglich, es ist jedoch darauf zu achten, dass die Zeit t nur Werte von $t = kT_0$ annehmen kann, wobei $k \in \mathbb{N}$ anzunehmen ist und T_0 die Abtastperiode repräsentiert.

Zu der gewünschten Beschreibungsform gelangt man, wenn man eine mit der Abtastperiode T_0 zeitdiskret abgetastete Funktion $f(kT_0)$ laplacetransformiert und den in der Laplacetransformierten auftretenden Term e^{T_0s} durch z substituiert. Dadurch erhält man die *Z-Transformierte* $F(z)$ der zeitdiskreten Funktion $f(kT_0)$. Mathematisch lässt sich die *Z-Transformation* durch folgende Gleichung ausdrücken:

$$F(z) = \mathfrak{Z}\{f(kT_0)\} = \sum_{k=0}^{\infty} f(kT_0)z^{-k} \quad \text{wobei } z = e^{T_0s} \quad (5.1)$$

Als Ergebnis erhält man für $F(z)$ ein Polynom in z , wobei der Faktor z^{-k} eine Verschiebung um k Abtastzyklen bedeutet. Man nennt z^{-1} auch den *Verschiebungoperator* oder *backward shift operator*.

Besides the cost function, the model which precalculates the future behavior is one of the crucial points of an MPC scheme. Most systems to be controlled are continuous-time systems so that the description of the system should be made in the time or Laplace domain. In contrast to the controlled systems, the controllers, generally realized with the help of digital computers, are often discrete-time controllers. For the controller design, it is in most cases sufficient to replace it by a continuous-time model, parameterize it and afterwards realize it as a discrete-time digital controller. Nevertheless, it is more advantageous to consider the discrete character of the controller already in the first approach of the controller design and hence to use a discrete-time system description. This also implies that the controlled system is described in the discrete-time domain. A Laplace transformed transfer function cannot be used now, otherwise the discretization

is lost with it. Of course, a representation of the system in the time domain is always possible, however, it has to be made sure that the time t can only assume values $t = kT_0$ with $k \in \mathbb{N}$ and T_0 representing the sampling rate.

The desired representation can be obtained if the time-discrete function $f(kT_0)$ sampled with T_0 is Laplace transformed and if then the term e^{T_0s} in the Laplace transformed function is replaced with z . In this way, the *Z-transformed* representation $F(z)$ of the time-discrete function $f(kT_0)$ can be obtained. Mathematically the *Z-transformation* can be expressed with the following equation

$$F(z) = \mathfrak{Z}\{f(kT_0)\} = \sum_{k=0}^{\infty} f(kT_0)z^{-k} \quad \text{with } z = e^{T_0s} \quad (5.1)$$

As a result, for $F(z)$ a polynomial in z can be obtained in which the factor z^{-k} means a shift of k sampling cycles. The factor z^{-1} is also called *shift operator* or *backward shift operator*.

除了成本函数之外，用于预测被控对象未来行为的模型是预测控制方法的关键。大部分被控系统均具有时间连续特性，因此较为理想的方式是在时域或频域内描述被控系统。不同于被控对象，借助于微处理器实现的数字控制器则具有时间离散特征。大多情况下，在控制器设计时，通常首先使用时间连续模型，并基于此调整参数，然后使用具有离散时间特征的数字控制器实现，该方案已可满足要求。更佳的方案是在模型中考虑控制器的离散特征，并使用时间离散的系统描述方式，这意味着要以时间离散的方式来描述被控对象，从而将基于拉普拉斯变换的传递函数方式排除在外，因为在上述方式下便失去了时间离散化的可能性。在时域内完成的系统描述明显是始终可行的，但要注意的是，时间 t 仅可取 $t = kT_0$ 时刻的值，其中， $k \in \mathbb{N}$ 表示整数， T_0 为采样间隔。

为了获得期望的描述类型，首先对以采样间隔 T_0 离散化的函数 $f(kT_0)$ 进行拉普拉斯变换，然后使用 z 来代替经拉普拉斯变换后出现的 e^{T_0s} 项，从而可得到时间离散函数 $f(kT_0)$ 的 Z 变换形式 $F(z)$ 。 Z 变换的数学描述方式如下：

$$F(z) = \mathfrak{Z}\{f(kT_0)\} = \sum_{k=0}^{\infty} f(kT_0)z^{-k} \quad \text{其中 } z = e^{T_0s} \quad (5.1)$$

可以看出， $F(z)$ 是一个 z 的多项式。其中，系数 z^{-k} 意味着向后推移了 k 个采样周期。通常称系数 z^{-1} 为平移算子或后向平移算子。

Wie aus Gl. (5.1) hervorgeht, ergibt eine Z-Transformation *immer* eine unendliche Reihe. Für viele Funktionen können jedoch geschlossene Ausdrücke für die Z-Transformierte angegeben werden, indem die Eigenschaften von Potenzreihen genutzt werden.

Zur Herleitung eines Streckenmodells im Z-Bereich wird man im Allgemeinen nicht auf die Definition der Z-Transformation gemäß (5.1) zurückgreifen, da dies zu mathematisch sehr aufwändigen Ausdrücken führt. Da in den einschlägigen regelungstechnischen Hand- und Lehrbüchern, z. B. [58] Transformationstabellen für die meistverwendeten Strukturen zu finden sind, gelangt man unter Verwendung dieser Tabellen und der Rechenregeln für Z-Transformierte üblicherweise schneller zum Ziel.

5.2.1 Zustandsraummodell/State space model/状态空间模型

Die einfachste Möglichkeit, zu einer zeitdiskreten Beschreibung einer Strecke zu gelangen, ist das bekannte *Zustandsraummodell*. Ein lineares System lässt sich anschaulich in der sogenannten Zustandsraumbeschreibung darstellen:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (5.2)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (5.3)$$

Dabei bezeichnet man \mathbf{A} als *Systemmatrix*, \mathbf{B} als *Eingangsmatrix*, \mathbf{C} als *Ausgangsmatrix* und \mathbf{D} als *Durchgangsmatrix* [108], sowie die Vektoren $\mathbf{x}(t)$, $\mathbf{u}(t)$ und $\mathbf{y}(t)$ als *Zustands-, Eingangs- und Ausgangsvektor*. Gl. (5.2) wird als *Zustandsgleichung*, (5.3) als *Ausgabegleichung* bezeichnet [84].

Da die modellbasierten Regelverfahren auf einem zeitdiskreten Modell der Strecke basieren, müssen die obigen Gleichungen noch in eine zeitdiskrete Darstellung¹

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \quad (5.4)$$

$$\mathbf{y}(k) = \mathbf{C}_d\mathbf{x}(k) + \mathbf{D}_d\mathbf{u}(k) \quad (5.5)$$

umgewandelt werden. Verschiedene Methoden, wie man die Systemgleichungen von der zeitkontinuierlichen in die zeitdiskrete Form umwandelt, werden in Kap. 7 diskutiert.

From equation (5.1) follows that a Z-transformation *always* results in an infinite series. However, for many functions, closed expressions for their Z-transformed can be given using the characteristics of power series.

¹ Zur Unterscheidung der Matrizen der zeitkontinuierlichen und zeitdiskreten Darstellung werden die Matrizen der zeitdiskreten Darstellung mit dem Index d gekennzeichnet.

For the derivation of a system model in Z-domain, it is not useful to apply the definition of the Z-transformation according to (5.1), as this will lead to very complex mathematical expressions. Since there are transformation tables for the typical structures given in the relevant control system handbooks and textbooks, e. g. [58], the result can normally be obtained much faster by using these tables and by applying the mathematical rules for Z-transformed functions.

5.2.1 State space model

The most simple way to get to a discrete-time representation of a system is the well-known *state space model*. A linear system can easily be represented in a so-called state space description, see equation (5.2) and (5.3).

In the above equations \mathbf{A} is called the *state matrix*, \mathbf{B} the *input matrix*, \mathbf{C} the *output matrix* and \mathbf{D} the *feedforward matrix* [108], whereas the vectors $\mathbf{x}(t)$, $\mathbf{u}(t)$ and $\mathbf{y}(t)$ are representing the *state*, *input* and *output vector*. The equations (5.2) and (5.3) are called *state equation* and *output equation*, respectively [84].

As the model-based control schemes are based on a discrete-time system model, the above equations have to be transformed into a discrete-time representation¹, see equation (5.4) and equation (5.5). Several strategies to transform the system equations from continuous-time into discrete-time representation will be discussed in Chapter 7.

由方程 (5.1) 可知, Z 变换得到的始终是一个无限序列。当然, 对于许多函数类型来说, 通过使用幂级数的特性也可以得到一个有限的 Z 变换表达式。

为了获得 Z 域内的被控对象模型, 通常并不使用方程 (5.1) 所示的 Z 变换定义形式, 因为该方法会形成一个非常复杂的数学表达式。在诸多优秀的控制技术相关的工具书或教科书中, 通常会给出典型结构的变换表, 如 [58]。基于 Z 变换表和 Z 变换的运算法则通常能够更为迅速地完成 Z 变换。

5.2.1 状态空间模型

状态空间模型是实现被控系统时间离散描述最为简单的一种方法。一个线性系统可以表述为如下的状态空间描述形式, 见方程 (5.2) 和 (5.3)。

式中, \mathbf{A} 表示系统矩阵, \mathbf{B} 表示输入矩阵, \mathbf{C} 表示输出矩阵, \mathbf{D} 表示直接输入输出矩阵 [108]; $\mathbf{x}(t)$, $\mathbf{u}(t)$ 和 $\mathbf{y}(t)$ 分别表示状态向量、输入向量和输出向量。方程 (5.2) 被称为状态方程, (5.3) 被称为输出方程 [84]。

由于基于模型的控制方法采用被控系统的时间离散模型, 所以必须将上述方程转化为时间离散描述形式¹, 见方程 (5.4) 和 (5.5)。第 7 章将详细讨论实现上述转化过程的不同方法。

¹ For an easier differentiation between matrices in continuous-time representation and matrices in discrete-time form the matrices in discrete-time representation are marked with the index d .

¹ 为了区分时间连续描述和时间离散描述中的矩阵, 使用下标为 d 的矩阵来表示时间离散描述中的情况。

5.2.2 Übertragungsfunktionsbasierte, lineare Modelle/Linear transfer function-based models/基于传递函数的线性模型

Bei zeitlich veränderlichen oder unbekanntem Streckenparametern ist ein adaptives Streckenmodell notwendig, dessen Modellparameter relativ einfach online geschätzt und der Realität nachgeführt werden können. Zustandsraumbasierte Streckenmodelle sind für diese Zwecke weniger gut geeignet; es ist daher die Verwendung eines Modells empfehlenswert, das auf der Übertragungsfunktion der zu regelnden Strecke basiert. Für diesen Verwendungszweck haben sich die nachfolgend erwähnten Modellstrukturen als besonders vorteilhaft erwiesen. Die Modelle in diesem Kapitel werden der Einfachheit halber alle als SISO-Modelle vorgestellt. Eine Erweiterung auf mehrdimensionale MIMO-Systeme ist jedoch problemlos möglich; für das CARIMA-Modell wird die entsprechende Vorgehensweise in Kap. 8 erläutert.

Die im Folgenden gegebene Übersicht über verschiedene lineare, zeitdiskrete Modelle, mit deren Hilfe die Entwicklung eines adaptiven MPC-Reglers möglich ist, wurde dem Buch von Kanjilal [62, Kap. 2.4] entnommen.

AR

Das *AutoRegressive model*, kurz *AR*, kann folgendermaßen ausgedrückt werden:

$$y(t) + a_1y(t-1) + a_2y(t-2) + \dots + a_ny(t-n) = \xi(t) \quad (5.6)$$

oder kurz

$$A(z^{-1})y(t) = \xi(t) \quad (5.7)$$

wobei

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n} \quad (5.8)$$

Die Funktion $\xi(t)$ stellt Störeinflüsse auf die Strecke dar. Ist der Störterm nicht deterministisch auszudrücken, wird für $\xi(t)$ oft weißes oder farbiges Rauschen eingesetzt.

For time variable or unknown system parameters an adaptive system model is necessary, whose model parameters can be estimated relatively simple online and which can be updated in real time. State space based system models are less suitable for such purposes; therefore, it is recommendable to use a model which is based on the transfer function of the system to be controlled. For this intended objective, the following model structures have turned out to be especially advantageous. For simplicity, all the models in this chapter are presented as SISO models. Nevertheless, an extension to multidimensional MIMO systems is easily possible; a suitable approach is explained in Chapter 8 for the CARIMA model.

A survey about the different linear time-discrete models with which the development of an adaptive MPC controller is possible is given in the following. This overview is taken from the book by Kanjilal [62, Kapitel 2.4].

AR

The *AutoRegressive model*, abbreviated *AR*, can be expressed as equation (5.6), or briefly equation (5.7), in which equation (5.8).

The function $\xi(t)$ represents the noise or disturbance which is affecting the system. If the noise term cannot be expressed deterministically, white or colored noise is often used for $\xi(t)$.

在时变或未知被控对象参数的情况下，有必要使用一个可实时在线估计模型参数的自适应被控对象模型。基于状态空间的被控对象模型显然不太适用于上述情形。因此建议使用基于传递函数的被控对象模型。接下来介绍的模型结构被证明非常适用于上述情形。为简化起见，本部分仅介绍单输入单输出模型。当然，将其进一步扩展至多输入多输出系统也是可行的：第 8 章中将进一步阐述 CARIMA-模型的工作原理。

接下来给出的几种可用于实现自适应 MPC 控制器的线性时间离散模型均摘自学者 Kanjilal 所著的专业书籍 [62, Kapitel 2.4]。

自回归模型

自回归模型简称为 AR，可表示为方程 (5.6)，简化形式为方程 (5.7)，其中方程 (5.8)。

函数 $\xi(t)$ 表示施加于被控对象的扰动影响。假使扰动项无法使用确定的数学表达式描述，那么则通常将 $\xi(t)$ 看作白噪声或有色噪声。

IAR

Das *Integrated AutoRegressive model* oder *IAR* besitzt als Erweiterung des AR-Modells einen integrierten Rauschterm:

$$A(z^{-1})y(t) = \frac{\xi(t)}{\Delta} \quad (5.9)$$

mit

$$\Delta = 1 - z^{-1} \quad (5.10)$$

das bedeutet

$$\Delta y(t) + a_1 \Delta y(t-1) + a_2 \Delta y(t-2) + \dots + a_{na} \Delta y(t-na) = \xi(t) \quad (5.11)$$

ARMA

Beim *ARMA* oder *AutoRegressive Moving Average model* wird das Rauschen durch einen erweiteren Ausdruck repräsentiert:

$$A(z^{-1})y(t) = C(z^{-1})\xi(t) \quad (5.12)$$

wobei

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{nc} z^{-nc} \quad (5.13)$$

In ausführlicher Schreibweise lautet dies:

$$\begin{aligned} y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{na} y(t-na) \\ = \xi(t) + c_1 \xi(t-1) + c_2 \xi(t-2) + \dots + c_{nc} \xi(t-nc) \end{aligned} \quad (5.14)$$

ARIMA

Die Abkürzung *ARIMA* steht für *AutoRegressive Integrated Moving Average model*. Ähnlich dem Unterschied zwischen AR- und IAR-Modellen wird auch hier nun das Rauschen durch eine integrierte Struktur berücksichtigt:

$$A(z^{-1})y(t) = C(z^{-1}) \frac{\xi(t)}{\Delta} \quad (5.15)$$

oder

$$\begin{aligned} \Delta y(t) + a_1 \Delta y(t-1) + a_2 \Delta y(t-2) + \dots + a_{na} \Delta y(t-na) \\ = \xi(t) + c_1 \xi(t-1) + c_2 \xi(t-2) + \dots + c_{nc} \xi(t-nc) \end{aligned} \quad (5.16)$$

IAR

The *Integrated AutoRegressive model* or *IAR* is an extension of the AR model with an integrated noise term, see [equation \(5.9\)](#), in which [equation \(5.10\)](#), therefore [equation \(5.11\)](#).

ARMA

With the *ARMA* or *AutoRegressive Moving Average model*, the noise is represented with an extended term, see [equation \(5.12\)](#), in which [equation \(5.13\)](#). In detail, this can be written as [equation \(5.14\)](#).

ARIMA

The abbreviation *ARIMA* stands for *AutoRegressive Integrated Moving Average model*. Similar to the difference between AR and IAR models the noise is, in this case, considered with an integrated structure again, see [equation \(5.15\)](#) or [equation \(5.16\)](#).

整合自回归模型

作为自回归模型扩展形式的整合自回归模型简称为 IAR，它包含一个整合的噪声项，见 [方程 \(5.9\)](#)，其中 [方程 \(5.10\)](#)，因此 [方程 \(5.11\)](#)。

自回归滑动平均模型

自回归滑动平均模型中的噪声项使用了一个扩展的形式，该模型可表示为 [方程 \(5.12\)](#)，其中 [方程 \(5.13\)](#)，完全展开后的形式为 [方程 \(5.14\)](#)。

自回归整合滑动平均模型

自回归整合滑动平均模型的缩写形式为 ARIMA。类似于 AR 与 IAR 模型之间的区别，此模型中的噪声也具有一个整合的结构形式，见 [方程 \(5.15\)](#)，或 [方程 \(5.16\)](#)。

ARMAX / CARMA

Das *ARMAX*- oder *AutoRegressive Moving Average model with eXogenous inputs* ist dem ARMA-Modell sehr verwandt, besitzt jedoch zusätzliche Eingangsvariablen, welche die von außen auf die Strecke einwirkenden Stellgrößen symbolisieren:

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + C(z^{-1})\xi(t) \quad (5.17)$$

mit

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb} \quad (5.18)$$

Der Wert d repräsentiert die Verzögerung oder Totzeit zwischen dem Eingang u und dem Ausgang y . In den für die Antriebstechnik relevanten Fällen wird üblicherweise $d = 1$ gesetzt. Zusätzliche Verzögerungen können alternativ auch in das Polynom $B(z^{-1})$ absorbiert werden.

ARMAX-Modelle sind auch unter der Bezeichnung *CARMA* oder *Controlled AutoRegressive Moving Average model* bekannt.

ARIMAX / CARIMA

Wie die Erweiterung von ARMA zu ARMAX können auch ARIMA-Modelle durch Hinzufügen von zusätzlichen Eingangsvariablen zu *ARIMAX*-Modellen, d. h. *AutoRegressive Integrated Moving Average model with eXogenous inputs* erweitert werden. Die Beschreibung lautet dann:

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + C(z^{-1})\frac{\xi(t)}{\Delta} \quad (5.19)$$

Die gleiche Struktur haben Modelle vom Typ *CARIMA* oder *Controlled AutoRegressive Integrated Moving Average model*. Sie werden z. B. für GPC-Regler eingesetzt.

ARMAX / CARMA

The *ARMAX* or *AutoRegressive Moving Average model with eXogenous inputs* is very similar to the ARMA model, but possesses additional input variables which represent the actuating variables from outside having an effect on the system, see equation (5.17), in which equation (5.18).

The value d represents the time delay or dead time between the input u and the output y of the system. In systems relevant for drive applications, usually $d = 1$ can be set. Alternatively, additional delays can be absorbed into the polynomial $B(z^{-1})$.

ARMAX models are also known as *CARMA* or *Controlled AutoRegressive Moving Average models*.

ARIMAX / CARIMA

Like the extension of ARMA to ARMAX, ARIMA models can also be extended to *ARIMAX* models, i.e. *AutoRegressive Integrated Moving Average models with eXogenous inputs* by adding additional input variables. The result can be described with the equation (5.19).

Models of the *CARIMA* or *Controlled AutoRegressive Integrated Moving average model* type have an identical structure. They are used e.g. for GPC controllers.

ARMAX / CARMA

带 **eXogenous** 输入 的自回归滑动平均模型类似于 ARMA，不同之处是额外包含了与控制量有关的表达式，见方程 (5.17)，其中方程 (5.18)。

数值 d 表示的是输入 u 和输出 y 之间的滞后或死区时间。对于驱动控制领域相关的情形来说，通常设置 $d = 1$ 。其他的滞后也可以纳入多项式 $B(z^{-1})$ 。

ARMAX 模型也称为 CARMA，即受控自回归滑动模型。

ARIMAX / CARIMA

类似于 ARMA 向 ARMAX 的扩展，ARIMA 模型也可通过添加额外的输入变量而扩展为 ARIMAX 模型，也即带 **eXogenous** 输入的自回归整合滑动平均模型，可表示为方程 (5.19)。

ARIMAX 模型也称为 CARIMA。此模型可用于 GPC 控制器。

5.2.3 Nichtlineare Modelle/Nonlinear models/非线性模型

Obwohl der Zweig der nichtlinearen Regelung noch relativ jung und unerforscht ist, gibt es eine Vielzahl von Modellen, die das Verhalten einer nichtlinearen Regelstrecke abbilden. Einige dieser nichtlinearen Modelle entpuppen sich bei näherem Hinsehen jedoch als lineare Modelle, die ein linearisiertes Abbild des zu regelnden Systems darstellen. Da es sich bei dieser Gruppe nicht um nichtlineare Modelle im eigentlichen Sinn handelt, werden sie hier nicht betrachtet. Unter den verbleibenden Strategien sind das *Hammerstein-Modell* und das *Wiener Modell* die bekanntesten. Neben diesen gibt es mit dem sogenannten *NARMAX-Modell* einen weiteren bemerkenswerten Ansatz zur Modellbildung für nichtlineare Strecken.

NARMAX

Das *Non-linear AutoRegressive Moving Average model with eXogenous inputs* wird von Leontaritis/Billings ausführlich beschrieben [80, 81]. Es handelt sich dabei um eine dem linearen ARMAX-Modell entsprechende, jedoch auf nichtlineare Regelstrecken erweiterte Beschreibung für multivariable, zeitdiskrete Systeme, und zwar sowohl für deterministische als auch für stochastische. Grundlage eines NARMAX-Modells ist die folgende Systembeschreibung für ein allgemeines multivariables, zeitdiskretes und zeitinvariantes System:

$$\mathbf{x}(t+1) = \mathbf{g}[\mathbf{x}(t), \mathbf{u}(t)] \quad (5.20)$$

$$\mathbf{y}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}(t)] \quad (5.21)$$

Vereinfachend wird nun zunächst ein lineares System betrachtet. Für ein solches zeitdiskretes, zeitinvariantes und lineares MIMO-System lässt sich definieren:

$$\mathbf{x}(t+1) = \mathbf{g}[\mathbf{x}(t), \mathbf{u}(t)] = \mathbf{A}_d \mathbf{x}(t) + \mathbf{B}_d \mathbf{u}(t) \quad (5.22)$$

$$\mathbf{y}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}(t)] = \mathbf{C}_d \mathbf{x}(t) + \mathbf{D}_d \mathbf{u}(t) \quad (5.23)$$

Diese Systembeschreibung im Zustandsraum, bestehend aus den Matrizen \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d und \mathbf{D}_d , kann mit Hilfe eines numerischen Lösungsverfahrens in ein multistrukturelles, lineares, übertragungsfunktionsbasiertes Modell umgeformt werden. Die erhaltene Beschreibung ähnelt den Modellen, die aus der linearen Regelungstechnik bekannt sind. Allerdings können die klassischen Darstellungsformen nicht für nichtlineare Systeme verallgemeinert werden, weshalb der neue Ansatz eines multistrukturellen Modells notwendig ist.

Although the area of nonlinear control is still very new and unexplored, a large number of models which represent the behavior of a nonlinear control system exists. However, when a closer look is taken at some of these nonlinear models, they turn out to be linear models using a linearized image of the system to be controlled. As these models are not really nonlinear models in a narrower sense, they are not further discussed here. Among the remaining strategies, the *Hammerstein model* and the *Wiener model* are the most prominent ones. Besides these, there is another interesting proposal for modeling nonlinear systems with the so-called *NARMAX model*.

NARMAX

The *Nonlinear AutoRegressive Moving Average model with eXogenous inputs* is described in detail by Leontaritis/Billings [80, 81]. Its main principle is a description of the controlled system similar to the linear ARMAX model, but extended to be suitable for nonlinear, multivariable, discrete-time control systems which can be deterministic as well as stochastic. Basis of a NARMAX model is the following system description for a general multivariable, discrete-time and invariant-time system, see equations (5.20) and (5.21). Simplistically, a linear system is considered at first. For such a discrete-time, invariant-time and linear MIMO system, the following can be defined, see equations (5.22) and (5.23).

This system description in state space, consisting of the matrices \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d and \mathbf{D}_d , can be transformed with the help of a numerical solution technique into a multistructural linear transfer function model. The obtained description is similar to models which are known from linear control engineering. Indeed, the classical representation forms cannot be generalized for nonlinear systems, which is the reason why the new multistructural model is necessary.

尽管非线性控制这一领域仍然处于起始阶段，并未得到广泛且深入的研究。但已有大量的模型可用于构建非线性被控对象的过程。经深入考察发现，其中一些非线性模型实际描述的是被控对象的线性化模型。由于上述类别讨论的并非是非真正的非线性模型，此处则不再详述。最为知名的非线性模型有哈默斯坦模型和维纳模型。此外，还有一种可对非线性被控对象建模的 NARMAX 模型。

NARMAX

带 **eXogenous** 输入的非线性自回归滑动平均模型在学者 Leontaritis/Billings 论文 [80, 81] 中有详细描述。上述模型为线性模型的扩展形式，可用于描述具有多变量、时间离散特性的非线性被控对象，既可以是确定性系统，也可以是随机系统。NARMAX 模型的基础是描述一般的多变量时间离散性时不变系统的如下形式，见方程 (5.20) 和 (5.21)。为简化起见，首先考虑一个线性系统。一个时间离散时不变系统可定义为方程 (5.22) 和 (5.23)。

这一由矩阵 \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d 和 \mathbf{D}_d 组成的状态空间描述形式可借助于数值求解方法将其转换至一个具有多重结构的基于传递函数的模型。经转换得到的模型类似于线性控制技术中已知的模型。显然，经典的描述形式无法用于一般的非线性系统，从而需要使用一个具有多重结构模型的方法。

Mit Hilfe einiger Gesetze der modernen Analysis und der differentiellen Geometrie wird die obige Vorgehensweise auf die nichtlineare Systembeschreibung übertragen und angewendet. Durch diese Verallgemeinerung erhält man ein rekursives, nichtlineares, multistrukturelles I/O-Modell, welches der linearen Beschreibung vollständig äquivalent ist. Diese rekursiven, nichtlinearen Modelle sind allerdings nur in einem beschränkten Arbeitsbereich um den Gleichgewichtspunkt gültig.

Durch Einbeziehen der stochastischen Störungen in das Modell erhält man eine erweiterte Systembeschreibung. Da dies nun exakt ein nichtlineares Pendant zum ARMAX-Modell darstellt, wird es auch als NARMAX-Modell bezeichnet.

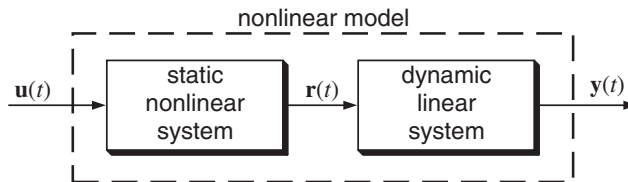
Hammerstein-Modell

Im Gegensatz zum größten Teil der bisher erläuterten linearen und nichtlinearen Modelle ist das *Hammerstein-Modell* kein klassisches MPC-Modell, sondern es wird auch für andere nichtlineare Regelungen angewandt. Die Idee beruht darauf, ein nichtlineares System durch ein Modell, bestehend aus einem statischen, nichtlinearen System, gefolgt von einem dynamischen, linearen System, abzubilden (Abb. 5.7). Die Nichtlinearität wird dabei z. B. durch eine Polynomstruktur ausgedrückt:

$$\mathbf{r}(t) = f_N(\mathbf{u}(t)) = \Gamma_1 \mathbf{u}(t) + \Gamma_2 \mathbf{u}^2(t) + \cdots + \Gamma_{n\alpha} \mathbf{u}^{n\Gamma}(t) \quad (5.24)$$

Für den linearen Teil sind verschiedene Modelle denkbar, in der Regel wird dort ein ARMAX-Modell (s. Kap. 5.2.2) verwendet, wobei der Störterm auch weggelassen werden kann.

Die Parameter des linearen Teilmodells werden üblicherweise mit den bekannten Parameteridentifikationsverfahren online nachgeführt. Der nichtlineare Systemteil ist statisch; hierfür findet folglich keine online-Parameterschätzung statt.



5.7: Struktur eines Hammerstein-Modells/Structure of a Hammerstein model/哈默斯坦模型结构

With the help of some rules of modern analysis and differential geometry, the above approach is transferred and applied to the nonlinear system description. By performing this generalization a recursive, nonlinear and multistructural I/O model is obtained, which is completely equivalent to the linear description. However, these recursive nonlinear models are only valid in a limited operation range around the equilibrium point.

An enlarged system description is obtained by including stochastic disturbances into the model. As this precisely represents a nonlinear counterpart to the ARMAX model, it is also called NARMAX model.

Hammerstein model

In contrast to most of the linear and nonlinear models explained above, the *Hammerstein model* is not a classical MPC model, but it is also used for other nonlinear controls. Its basic idea is to compose a nonlinear model from a static, nonlinear part followed by a dynamic linear system as shown in Figure 5.7. The nonlinearity is expressed e. g. by a polynomial like equation (5.24).

For the linear part, different models are possible. In most cases, an ARMAX model (see Chapter 5.2.2) is used, in which the disturbance term can be neglected.

The parameters of the linear part of the model are usually determined with well-known online parameter identification techniques. The nonlinear system part is static; consequently, no online parameter estimation is performed.

借助于现代分析和微分几何的相关知识，同样可以将上述处理方式应用至非线性系统描述。经过上述一般化处理，可以得到一个完全等效于线性系统描述的递归形式的非线性多结构 I/O 模型。显然，上述递归形式的非线性模型仅适用于平衡点附近的有限运行范围。

通过在模型中引入随机噪声，即可得到一个扩展的系统描述。由于本部分讲述的是完全对应 ARMAX 模型的非线性模型，所以也称为 NARMAX 模型。

哈默斯坦模型

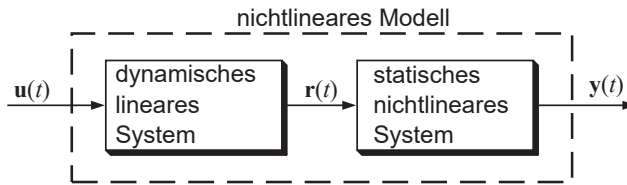
不同于到目前为止已介绍的非线性 and 线性控制，哈默斯坦模型并非是典型的 MPC 模型，但它同样可用于其他的非线性控制。其基本思想是使用一个静态非线性系统和一个后接的动态线性系统共同组成一个非线性系统模型，如图 5.7 所示。系统的非线性可通过下述多项式进行描述，见方程 (5.24)。

至于线性部分，有多种不同的模型可供选择，通常可使用章节 5.2.2 介绍的 ARMAX 模型，其中的扰动项也是可以省略掉的。

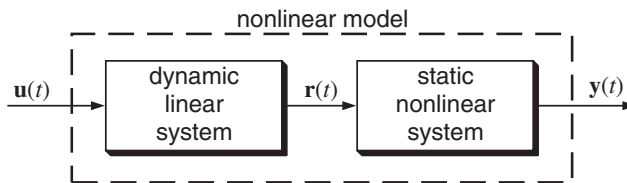
动态线性系统的参数通常使用已知的参数辨识方法进行在线辨识；静态非线性系统部分是静态的，无需进行在线参数辨识。

Wiener Modell

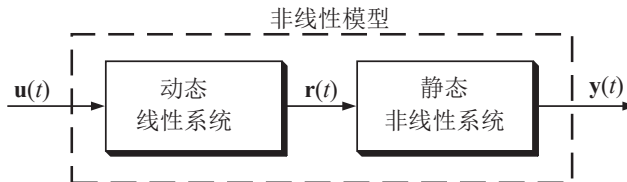
Das *Wiener Modell* hat mit der Österreichischen Hauptstadt nichts gemeinsam, sondern ist nach dem Mathematiker und Physiker *Norbert Wiener* (1894–1964) benannt. Es unterscheidet sich von dem Hammerstein-Modell dadurch, dass das nichtlineare System hier durch ein dynamisches, lineares System, gefolgt von einem statischen, nichtlinearen System modelliert wird (Abb. 5.8). Wie beim Hammerstein-Modell werden die Parameter des linearen Modellteils online nachgeführt, während der nichtlineare Teil statisch ist.



5.8: Struktur eines Wiener Modells



5.9: Structure of a Wiener model



5.10: 维纳模型结构

Nachdem in diesem Kapitel die generelle Funktionsweise eines modellbasierten Prädiktivreglers sowie verschiedene, für diese Reglertypen geeignete Streckenmodelle vorgestellt wurden, soll nun ein MPC-Regler in der Praxis realisiert werden. In den folgenden Kapiteln werden dazu verschiedene modellbasierte Regelstrategien vorgestellt und experimentell untersucht.

Wiener model

The Wiener model has no relation to the Austrian capital, it is named after the mathematician and physicist Norbert Wiener (1894–1964). It differs from the Hammerstein model by the fact that here the nonlinear system is modeled with a dynamic linear system followed by a static nonlinear system (see Figure 5.9). Like the Hammerstein model, the parameters of the linear part are estimated online, whereas the nonlinear part is static.

After in this chapter the general functional principle of model-based predictive controllers as well as different system models suitable for these controllers were introduced, now an MPC controller should be realized in practice. In the following chapters different model-based control strategies are presented and in addition they are investigated experimentally.

维纳模型

维纳模型与奥地利首都维也纳并无关联，它是以数学家和物理学家诺伯特·维纳 (1894–1964) 命名的。维纳模型与哈默斯坦模型之间的差别在于静态非线性系统和动态线性系统的前后顺序不同，如图 5.10 所示。类似于哈默斯坦模型，此时动态线性部分模型参数需进行在线辨识，而静态非线性部分则无必要。

在介绍过基于模型的预测控制器一般工作原理以及不同的适用于此控制器类型的被控对象模型之后，那么进一步则要在实际实现 MPC 控制器。接下来的章节将介绍几种不同的模型预测控制策略，并以实验方式进行详细考察。

6 Allgemeine Prädiktivregelung/ Generalized Predictive Control/广义预测 控制

Das Regelverfahren *Generalized Predictive Control* – abgekürzt *GPC* – gehört ebenfalls zur Gruppe der modellbasierten Prädiktivregler. Es wurde von Clarke von der University of Oxford im Jahr 1987 vorgestellt [27, 28] und verwendet ein übertragungs-basiertes CARIMA-Modell. Das Optimierungsproblem wird analytisch durch Nullsetzen der Ableitung der Kostenfunktion gelöst; die Verwendung mathematisch aufwändiger Lösungsverfahren wie quadratischer oder linearer Programmierung ist folglich unnötig.

6.1 „Klassisches GPC“/“Classical GPC”/经典广义预测控制

In seiner einfachsten Form stellt der GPC-Regler einen eindimensionalen SISO-Regler dar, d. h. das zu regelnde System besitzt nur eine Eingangs- und eine Ausgangsgröße. Eine Berücksichtigung von Störgrößen oder dergleichen findet nicht statt.

6.1.1 Mathematische Herleitung/Mathematical derivation/数学推导

Das CARIMA-Modell

Grundlage eines MPC-Verfahrens ist immer ein Modell des zu regelnden Systems, wobei bei einem GPC-Regler ein auf der Übertragungsfunktion des Systems basierendes CARIMA-Modell zum Einsatz kommt. Diese Übertragungsfunktion läßt sich bei zeitdiskreten Strukturen im Allgemeinen als Bruch zweier Polynome beschreiben:

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{nb} z^{-nb}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na}} \quad (6.1)$$

Die obige Struktur läßt sich besser veranschaulichen, wenn man eine Darstellung verwendet, wie sie in der Nachrichtentechnik für rekursive Filter (IIR-Filter) verwendet wird. Abb. 6.1 zeigt das äquivalente Blockschaltbild zu Formel (6.1).

The control strategy *Generalized Predictive Control*—abbreviated *GPC*—also belongs to the group of model based predictive controllers. It was introduced by Clarke at the University of Oxford in 1987 [27,28] and makes use of a transfer function based CARIMA model. The optimization problem is solved analytically by setting the derivative of the cost function equal to zero; thus, the use of mathematically complex solution algorithms like quadratic (QP) or linear programming (LP) is not required.

6.1 “Classical GPC”

In its most simple form, a GPC controller represents a linear SISO controller, i.e. the system to be controlled has only one input and one output variable. Consideration of disturbances or other disturbing signals is not taken into account.

6.1.1 Mathematical derivation

The CARIMA model

Basis of an MPC scheme is always a model of the system or plant which is to be controlled. In a GPC controller a so called CARIMA model is used which is based on the transfer function of the plant. Considering time-discrete structures, this transfer function can commonly be described as a fraction of two polynomials, see equation (6.1).

The above structure can be explained easier if a representation similar to one used in communications engineering for recursive filters (IIR filters) is used. Figure 6.1 shows the equivalent block diagram for equation (6.1).

广义预测控制隶属于模型预测控制的范畴，简称 GPC。GPC 由牛津大学学者 Clarke 于 1987 年首先提出 [27,28]，它使用的是基于传递函数的 CARIMA 模型。其中，优化问题的解是以解析方式将成本函数的导数置零求得，使用数学上更为复杂的求解方法则是不必要的，如二次规划或线性规划等。

6.1 经典广义预测控制

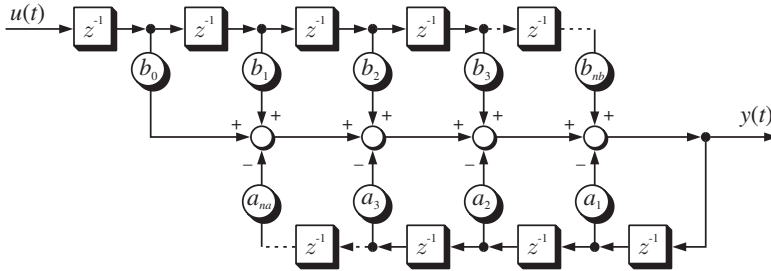
最简单形式的 GPC 控制器是一维的单输入单输出控制器。换言之，被控系统仅包含一个输入变量和一个输出变量，且不考虑扰动变量及其他变量的影响。

6.1.1 数学推导

CARIMA 模型

模型预测控制方法的基础始终是被控系统模型，GPC 控制器使用的是基于系统传递函数的 CARIMA 模型。具有时间离散结构的传递函数通常可表示为两个多项式的商，见方程 (6.1)。

上述结构更为直观的表达方式可借用通信技术中描述无限冲击响应滤波器的方法。图 6.1 给出了方程 (6.1) 的等效框图。



6.1: Zeitdiskrete Übertragungsfunktion/Discrete time transfer function/时间离散传递函数

Hat das System eine Totzeit, so sind die ersten Elemente von $B(z^{-1})$ Null. Aus (6.1) lässt sich die Gleichung des CARIMA-Modells herleiten (s. auch Kap. 5.2.2).

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + C(z^{-1})\frac{\xi(t)}{\Delta} \quad (6.2)$$

Der letzte Term in (6.2) repräsentiert dabei den Einfluss von Störgrößen. Wenn $\xi(t)$ weißes Rauschen darstellt, so kann das Polynom $C(z^{-1}) = 1$ gesetzt werden. Dadurch vereinfacht sich die Gleichung zu:

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + \frac{\xi(t)}{\Delta} \quad (6.3)$$

Der j-Schritt-voraus-Prädiktor

Um einen Prädiktor zu erhalten, wird folgende Diophantische Gleichung¹ angenommen:

$$1 = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \quad (6.4)$$

Mit Hilfe der Definition $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$ ergibt sich für die Diophantische Gleichung:

$$1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \quad (6.5)$$

Hintergedanke der Gl. (6.5) ist, dass hier quasi $1/\tilde{A}(z^{-1})$ berechnet wird, bis der Rest als $z^{-j}F_j(z^{-1})$ faktorisiert werden kann. Der Quotient der Division ist das Polynom $E_j(z^{-j})$. Folglich ist $E_j(z^{-1})$ vom Grad $j-1$, so dass sich $E_j(z^{-1})$ und $F_j(z^{-1})$ definieren lassen:

$$E_j(z^{-1}) = e_{j,0} + e_{j,1}z^{-1} + e_{j,2}z^{-2} + \dots + e_{j,j-1}z^{-(j-1)} \quad (6.6)$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1}z^{-1} + f_{j,2}z^{-2} + \dots + f_{j,na}z^{-na} \quad (6.7)$$

¹ Zur Erläuterung des Begriffs der Diophantischen Gleichung siehe Anhang A.

If the system is affected with a dead time, the first elements of the polynomial $B(z^{-1})$ are equal to zero. The equation of the CARIMA model can be derived from (6.1) (see also Chapter 5.2.2), see equation (6.2).

Thereby, the last term in (6.2) represents the effect of disturbances. If $\xi(t)$ is white noise, the polynomial can be set to $C(z^{-1}) = 1$. Thus, the equation can be simplified to equation (6.3).

The j-step ahead predictor

In order to obtain a predictor, the following Diophantine equation¹ is considered, see equation (6.4).

With the help of the definition $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$, the following Diophantine equation can be obtained, see equation (6.5).

The idea behind equation (6.5) is that the reciprocal $1/\tilde{A}(z^{-1})$ is calculated until the remainder can be factorized as $z^{-j}F_j(z^{-1})$. The quotient of the division is the polynomial $E_j(z^{-j})$. Thus, $E_j(z^{-1})$ is of degree $j - 1$ which leads to the fact that $E_j(z^{-1})$ and $F_j(z^{-1})$ can be defined as equations (6.6) and (6.7).

如果系统包含死区时间，那么，多项式 $B(z^{-1})$ 的首项则为零。根据方程 (6.1) 可推导出 CARIMA 模型的表达式，参见章节 5.2.2，见方程 (6.2)。

方程 (6.2) 最后一项描述的是扰动变量的影响。如果 $\xi(t)$ 表示白噪声，则可设置多项式 $C(z^{-1}) = 1$ ，进而可简化为方程 (6.3)。

J 步前向预测器

为了得到一个预测器，需考虑下述丢番图方程¹，见方程 (6.4)。

定义 $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$ ，代入丢番图方程后可得方程 (6.5)。

方程 (6.5) 背后的思想在于计算出 $1/\tilde{A}(z^{-1})$ ，直至剩余部分可转化为系数 $z^{-j}F_j(z^{-1})$ 。首先将 $z^{-j}F_j(z^{-1})$ 移至方程左侧，然后两侧同时除以 $\tilde{A}(z^{-1})$ ，即可得到多项式 $E_j(z^{-j})$ 。由于多项式 $E_j(z^{-j})$ 的阶数为 $j - 1$ ，可定义 $E_j(z^{-1})$ 和 $F_j(z^{-1})$ 为方程 (6.6) 和 (6.7)。

¹ Refer to appendix A for a description of Diophantine equations.

¹ 关于丢番图方程的详细描述，可参考附录 A。

Nun wird (6.3) mit $\Delta E_j(z^{-1})z^j$ multipliziert:

$$\tilde{A}(z^{-1})E_j(z^{-1})y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) + E_j(z^{-1})\xi(t+j) \quad (6.8)$$

Die Auflösung von (6.5) nach $E_j(z^{-1})\tilde{A}(z^{-1})$ und Einsetzen in obige Gleichung ergibt:

$$y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})y(t) + E_j(z^{-1})\xi(t+j) \quad (6.9)$$

Da $E_j(z^{-1})$ wie bereits erwähnt vom Grad $j-1$ ist, liegen die Rauschkomponenten alle in der Zukunft. Eine bestmögliche Prädiktion von y ist daher:

$$\hat{y}(t+j) = G_j(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})y(t) \quad (6.10)$$

wobei

$$G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$$

Teilt man die Prädiktion in einzelne Schritte auf, so erhält man für die verschiedenen Prädiktionsschritte folgende Gleichungen:

$$\hat{y}(t+1) = G_1(z^{-1})\Delta u(t) + F_1(z^{-1})y(t) \quad (6.11)$$

$$\hat{y}(t+2) = G_2(z^{-1})\Delta u(t+1) + F_2(z^{-1})y(t) \quad (6.12)$$

$$\hat{y}(t+3) = G_3(z^{-1})\Delta u(t+2) + F_3(z^{-1})y(t) \quad (6.13)$$

⋮

$$\hat{y}(t+N_p) = G_{N_p}(z^{-1})\Delta u(t+N_p-1) + F_{N_p}(z^{-1})y(t) \quad (6.14)$$

Berechnung der Steuergrößen

Der GPC-Algorithmus wendet auf die mit (6.10) vorausberechneten Werte eine quadratische Kostenfunktion an, um die zu erwartenden Regelabweichungen $\hat{y}(t+j) - w(t+j)$ zu minimieren. Gleichzeitig wird auch der Stellgrößenaufwand $\Delta u(t+j-1)$ einer Bewertung unterzogen.

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \mu_j (\hat{y}(t+j) - w(t+j))^2 + \sum_{j=1}^{N_u} \lambda_j (\Delta u(t+j-1))^2 \quad (6.15)$$

Einige MPC-Verfahren vernachlässigen den zweiten Term, der die Stellgrößenänderungen berücksichtigt; andere wiederum bewerten die Stellgrößen selbst anstelle ihrer Änderungen. Bei GPC wird jedoch der gesamte Ausdruck ausgewertet. Dabei spielen die einzelnen Parameter folgende Rolle:

Now (6.3) is multiplied with $\Delta E_j(z^{-1})z^j$, see equation (6.8).

Solving (6.5) for $E_j(z^{-1})\tilde{A}(z^{-1})$ and using it in the above equation, the following result can be obtained, see equation (6.9).

Since, as already mentioned, the degree of the polynomial $E_j(z^{-1})$ is $j - 1$, all noise components are in the future. Hence, the best possible prediction for y is equation (6.10).

If the prediction is split into single steps, the following equations for the different prediction steps can be obtained, see equations (6.11) to (6.14).

Calculation of the actuating variables

The GPC algorithm uses a quadratic cost function in order to minimize values of the expected control deviations $\hat{y}(t+j) - w(t+j)$ which were predicted with (6.10). At the same time the costs for changing the values of the actuating variables $\Delta u(t+j-1)$ are evaluated, too, see equation (6.15).

Some MPC methods do not consider the second term which takes the change of the values of the actuating variables into account; other methods evaluate the values of the actuating variables themselves instead of their changes. However, in GPC the entire expression is evaluated. Thereby, the following parameters play a role:

现将方程 (6.3) 两端乘以 $\Delta E_j(z^{-1})z^j$, 可得 方程 (6.8)。

根据方程 (6.5) 求解 $E_j(z^{-1})\tilde{A}(z^{-1})$, 然后代入至上述方程可得 方程 (6.9)。

由于多项式 $E_j(z^{-1})$ 的阶数为 $j - 1$, 所以噪声分量全部位于未来时刻。对于 y 最为合适的预测表达式为 方程 (6.10)。

如果进行逐步预测, 即可得到每一预测步数时的 方程组 (6.11) to (6.14)。

控制量计算

GPC 算法使用一个二次成本函数, 它包含基于方程 (6.10) 得到的预测值, 目的是最小化期望的控制偏差。此外, 控制增量的惩罚项也可同时纳入评判标准, 见 方程 (6.15)。

一些 MPC 方法忽略了上述成本函数中用于惩罚控制增量的右侧第二项, 也有一些方法惩罚的是控制变量, 而非控制增量。GPC 则对上述目标函数的全部项进行评估, 其中, 相关参数含义如下:

-
- Die Horizonte N_1 und N_2 werden als *unterer* und *oberer Kostenhorizont* bezeichnet. Sie geben den Bereich an, über den Prädiktion und Optimierung erfolgen sollen. Regelabweichungen im Zeitraum von $t \dots (t + (N_1 - 1))$ gehen folglich nicht in die Bewertung ein. Dies ist z. B. dann sinnvoll, wenn das zu regelnde System eine Totzeit beinhaltet, während der die Stellgröße keinen Einfluss auf die Regelgröße hat. In diesem Fall ist die Bewertung der in der Totzeitphase liegenden Regelabweichungen sinnlos, da sie in keinem Fall ausgeregelt werden können. Zur Einsparung von Rechenzeit kann daher N_1 auf den Wert der Totzeit gesetzt werden.
 - Der Parameter N_u kennzeichnet den sogenannten *Regelhorizont*. Dieser muss nicht unbedingt mit dem Prädiktionshorizont übereinstimmen; in diesem Fall geht der Optimierungsalgorithmus dann davon aus, dass sich nach N_u Schritten die Stellgröße nicht mehr ändert, d. h. $\Delta u(t + j - 1) = 0$ für $j > N_u$.
 - Die Koeffizienten μ_j bewerten die vorausberechneten, zukünftigen, quadrierten Regelabweichungen $\hat{y}(t + j) - w(t + j)$. Durch eine entsprechende Folge von μ_j besteht die Möglichkeit, zeitlich spätere Regelfehler stärker zu gewichten als nahe bei t liegende. Auch kann durch Ersetzen des Quadrats des Fehlers durch den Absolutwert und die Wahl von $\mu_j \sim j$ eine Optimalfunktion gemäß dem ITAE-Kriterium vorgegeben werden.
 - Die Koeffizienten λ_j stellen eine Gewichtung des quadrierten Werts der Stellgrößenveränderungen $\Delta u(t + j - 1)$ dar. Somit ist es genau wie bei der Bewertung der Regelabweichungen möglich, zeitlich spätere Stellsignale stärker oder weniger stark zu bewerten als nahe bei t liegende.

-
- The parameters N_1 and N_2 are called *lower* and *upper cost horizon*. They indicate the horizon for prediction and optimization. Hence, control deviations in the time interval $t \dots (t + (N_1 - 1))$ are not considered in the evaluation. This does e. g. make sense if the plant is affected with a dead time during which the actuating variable has no effect on the controlled variable. In this case, the evaluation of control deviations being in the dead time phase does not make sense because these cannot be corrected in any way. Thus, in order to save computation time, N_1 can be set equal to the value of the dead time.
 - The parameter N_u denotes the so called *control horizon*. It does not necessarily need to be equal to the prediction horizon; then, in this case, the optimization algorithm assumes that the value of the actuating variable does not change anymore after N_u steps, i. e. $\Delta u(t + j - 1) = 0$ for $j > N_u$.

- The coefficients μ_j weigh the predicted future quadratic control deviations $\hat{y}(t+j) - w(t+j)$. By using an appropriate sequence of μ_j , it is possible to valuate later control deviations higher than control deviations being closer to t . Furthermore, by replacing the quadratic error with the absolute value and by selecting $\mu_j \sim j$, an optimum function according to the ITAE criterion can be given.
- The coefficients λ_j weigh the quadratic value of the change of the values of the actuating variables $\Delta u(t+j-1)$. Consequently, in the same way as for weighing of the control deviations, it is possible to valuate values of the actuating variables being not so close in the future, more or less than the ones being close to t .

-
- 参数 N_1 和 N_2 分别表示下成本步长和上成本步长，它们确定了实现预测和优化的范围。在 $t \dots (t + (N_1 - 1))$ 时间间隔内的控制偏差并未作为成本函数的惩罚项。只有当被控系统包含死区时间时，换言之，控制变量对被控变量不产生影响，上述考虑才可体现其价值。惩罚死区时间范围内的控制偏差是毫无意义的，因为无论如何该时间范围内的偏差是不可控的。为了节省计算时间，可将 N_1 设定为死区时间对应的数值。
 - 参数 N_u 表示所谓的控制步长，控制步长并非一定要与预测步长保持一致。基于优化算法，在经过 N_u 步数之后，控制变量将不再发生变化，也即当 $j > N_u$ 时， $\Delta u(t+j-1) = 0$ 。
 - 系数 μ_j 惩罚二次的未来预测控制偏差 $\hat{y}(t+j) - w(t+j)$ 。可通过设置不同的 μ_j 序列，以使得时间靠后的控制偏差权重大于靠近 t 时刻的控制偏差权重。此外，也可使用绝对值代替平方项，并选择 $\mu_j \sim j$ ，然后即可基于 ITAE 准则确定最优函数。
 - 系数 λ_j 表达的是控制增量 $\Delta u(t+j-1)$ 平方项的权重。类似地，也可同样设置 λ_j 序列，以使得时间靠后的控制偏差权重大于靠近 t 时刻的控制偏差权重。

Alle diese Koeffizienten können als Einstellparameter betrachtet werden, mit deren Hilfe das Verhalten des MPC-Reglers in weiten Grenzen verändert werden kann, um den Regler an das gewünschte Streckenverhalten optimal anzupassen. Auf diese Art und Weise kann einem MPC-Regler das Verhalten fast jedes anderen prädiktiven und nicht-prädiktiven Reglertyps aufgezwungen werden. Für einen Standard-GPC-Regler setzen Clarke/Mohtadi/Tuffs die Werte $\mu_j = 1$ und $\lambda_j = \lambda$ für alle j ein [27]. Der Einfachheit halber setzt man meist auch $N_1 = 1$ und $N_2 = N_p$.

Für die Optimierung der Stellgrößen muss nun zwischen zukünftigen und in der Vergangenheit liegenden Werten von $u(t)$ unterschieden werden, denn in kausalen Systemen kann auf bereits zurückliegende Stellgrößen kein Einfluss mehr ausgeübt werden. Der Term $G_j(z^{-1})\Delta u(t + j - 1)$ in Gl. (6.10) wird daher in zukünftige und vergangene Teilterme zerlegt. Die Summe aus den zurückliegenden Istwerten $F_j(z^{-1})y(t)$ und der Systemreaktion auf die zurückliegenden Stellgrößen bildet die *freie Antwort* \mathbf{f} . Der Rest, bestehend aus der Systemantwort auf die zukünftigen Stellgrößen, ist die *erzwungene Antwort*.

$$\begin{aligned} \mathbf{y} &= \mathbf{G}\mathbf{u} + \underbrace{\mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t-1)}_{=\mathbf{f}} \\ \mathbf{y} &= \underbrace{\mathbf{G}\mathbf{u}}_{\substack{\text{erzwungene} \\ \text{Antwort}}} + \underbrace{\mathbf{f}}_{\substack{\text{freie} \\ \text{Antwort}}} \end{aligned} \quad (6.16)$$

Wobei $\mathbf{u} = \Delta\mathbf{u}$. Mit Hilfe von (6.16) lässt sich die Kostenfunktion (6.15) unter der Voraussetzung, dass $\mu_j = 1$ und $\lambda_j = \lambda$ für alle j , umformen zu

$$J = (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w})^T (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u} \quad (6.17)$$

Um diese Gleichung zu minimieren, wird sie zuerst ausmultipliziert, wodurch sich

$$J = \mathbf{u}^T (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}) \mathbf{u} + \mathbf{u}^T \mathbf{G}^T (\mathbf{f} - \mathbf{w}) + (\mathbf{f} - \mathbf{w})^T \mathbf{G} \mathbf{u} + (\mathbf{f} - \mathbf{w})^T (\mathbf{f} - \mathbf{w}) \quad (6.18)$$

ergibt. Wegen der Identität

$$\mathbf{u}^T \mathbf{G}^T (\mathbf{f} - \mathbf{w}) = (\mathbf{f} - \mathbf{w})^T \mathbf{G} \mathbf{u} \quad (6.19)$$

läßt sich J weiter vereinfachen zu

$$J = \mathbf{u}^T (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}) \mathbf{u} + 2\mathbf{u}^T \mathbf{G}^T (\mathbf{f} - \mathbf{w}) + (\mathbf{f} - \mathbf{w})^T (\mathbf{f} - \mathbf{w}) \quad (6.20)$$

All these coefficients can be considered as setting parameters with whose help the behavior of the MPC controller can be changed a lot in order to adjust the controller optimally for achieving the desired system behavior. In this way, the behavior of nearly every other predictive and non-predictive controller type can be imposed upon an MPC controller. For a standard GPC controller, Clarke/Mohtadi/Tuffs use the values $\mu_j = 1$ and $\lambda_j = \lambda$ for all j [27]. For reasons of simplicity, also the horizons are often set to $N_1 = 1$ and $N_2 = N_p$.

For the optimization of the values of the actuating variables, a differentiation between values of $u(t)$ being in the future and values being in the past has to be made, since in causal systems values of the actuating variables that are in the past cannot be changed anymore. Thus, the term $G_j(z^{-1})\Delta u(t+j-1)$ in equation (6.10) is splitted into sub-terms concerning the future and the past. The sum of the past output values $F_j(z^{-1})y(t)$ and of the system response to the past values of the actuating variables forms the *free response* \mathbf{f} . The rest, consisting of the system response to future values of the actuating variables, is the *forced response*, see equation (6.16), in which $\mathbf{u} = \Delta \mathbf{u}$.

With help of (6.16) and assuming that $\mu_j = 1$ and $\lambda_j = \lambda$ for all j , the cost function (6.15) can be simplified to equation (6.17).

In order to minimize this equation, it is first expanded into separate terms, whereby equation (6.18) results. Because of the identity (6.19), J can be further simplified to equation (6.20).

所有上述介绍的参数均可看作是可调参数，借助上述参数即可改变 MPC 控制器的特性，从而可调整得到适合于被控系统特性的最优控制器。基于上述方式，MPC 控制器几乎可以取代任何一个预测的或非预测的控制器类型。学者 Clarke/Mohtadi/Tuffs 对于所有的 j 均设置 $\mu_j = 1$ 和 $\lambda_j = \lambda$ [27]，并以此作为标准的 GPC 控制器。为简化起见，通常设置 $N_1 = 1$ 和 $N_2 = N_p$ 。

对于控制变量的优化来说，必须区分控制变量 $u(t)$ 的未来值和过去值，因为在一个因果性的系统中，过去时刻的控制变量无法继续产生影响。正因如此，可将方程 (6.10) 中的 $G_j(z^{-1})\Delta u(t+j-1)$ 项分解为代表未来的和过去的两个子项。过去的 $F_j(z^{-1})y(t)$ 与过去控制变量产生的系统响应之和构成了自由响应；剩余的由未来控制变量形成的系统响应则代表强制响应，见方程 (6.16)，其中， $\mathbf{u} = \Delta \mathbf{u}$ 。

结合方程 (6.16)，并且对于所有的 j 来说，满足 $\mu_j = 1$ 和 $\lambda_j = \lambda$ 这一前提条件，成本函数可变形为方程 (6.17)。

为了求得上述方程的最小值，将表达式展开，可得方程 (6.18)。根据恒等式 (6.19)， J 可进一步简化为方程 (6.20)。

Das Minimum einer Gleichung wird ermittelt, indem die Nullstellen der Ableitung gesucht werden:

$$\frac{\partial J}{\partial \mathbf{u}} = 2(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}) \mathbf{u} + 2\mathbf{G}^T(\mathbf{f} - \mathbf{w}) \stackrel{!}{=} 0 \quad (6.21)$$

Auflösen nach \mathbf{u} ergibt die Gleichung für optimale, zukünftige Stellgrößen

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}^T(\mathbf{w} - \mathbf{f}) \quad (6.22)$$

Da nur das erste Element von \mathbf{u} benötigt wird, und dieses $\Delta u(t)$ ist, so folgt aus Gl. (6.22) die Vorschrift für die im nächsten Abtastschritt an die Strecke zu legende Stellgröße:

$$u(t) = u(t-1) + \mathbf{g}^T(\mathbf{w} - \mathbf{f}) \quad (6.23)$$

Dabei ist \mathbf{g}^T die erste Zeile der Matrix $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}^T$.

Rekursion der Diophantischen Gleichung

Um $E_j(z^{-1})$ und $F_j(z^{-1})$ zu berechnen, muss die Diophantische Gleichung (6.5) gelöst werden. Zwar existieren alternative GPC-Verfahren *ohne* Berechnung von Diophantischen Gleichungen, z. B. die von Albertos und Ortega [3]; im Folgenden wird jedoch nach dem „klassischen“ Lösungsschema vorgegangen. Dazu werden zuerst die Diophantischen Gleichungen für j und $j+1$ aufgestellt und voneinander abgezogen:

$$\begin{array}{l} - \left| 1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \right. \\ + \left| 1 = E_{j+1}(z^{-1})\tilde{A}(z^{-1}) + z^{-(j+1)}F_{j+1}(z^{-1}) \right. \\ \hline \left| 0 = (E_{j+1}(z^{-1}) - E_j(z^{-1}))\tilde{A}(z^{-1}) + z^{-j}(z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1})) \right. \end{array} \quad (6.24)$$

Der Term $E_{j+1}(z^{-1}) - E_j(z^{-1})$ ist vom Grad j , da $E_j(z^{-1})$ – wie bereits weiter oben ausgeführt – vom Grad $j-1$ ist. Damit wird angesetzt:

$$E_{j+1}(z^{-1}) - E_j(z^{-1}) = \tilde{R}(z^{-1}) + r_j z^{-j} \quad (6.25)$$

wobei $\tilde{R}(z^{-1})$ ein Polynom vom Grad $\leq j-1$ und r_j eine reelle Zahl sind. Damit ergibt sich für die Diophantische Gleichung

$$0 = \tilde{R}(z^{-1})\tilde{A}(z^{-1}) + z^{-j}(z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1}) + r_j\tilde{A}(z^{-1})) \quad (6.26)$$

Da $\tilde{a}_0 = 1$ ist, muss wegen der Verschiebung der übrigen Terme um z^{-j} das Polynom $\tilde{R}(z^{-1}) = 0$ sein. Die Formel vereinfacht sich folglich zu

$$z^{-1}F_{j+1}(z^{-1}) = F_j(z^{-1}) - r_j\tilde{A}(z^{-1}) \quad (6.27)$$

The minimum of this equation is determined by setting the derivative of J to zero, see equation (6.21).

Solving the equation for \mathbf{u} , the equation (6.22) for optimum future values of the actuating variables can be obtained.

Since only the first element of \mathbf{u} , which is actually $\Delta u(t)$, is required, the law for the value of the actuating variable that has to be applied to the plant can be derived from (6.22), see equation (6.23). Thereby, \mathbf{g}^T is the first row of matrix $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}^T$.

Recursion of the Diophantine equation

The Diophantine equation (6.5) has to be solved in order to calculate $E_j(z^{-1})$ and $F_j(z^{-1})$. Although other GPC algorithms which do *not* require the calculation of Diophantine equations are available, e. g. the ones proposed by Albertos and Ortega [3], in the following the “classical” solution will be presented. For this purpose, the Diophantine equations for the steps j and $j + 1$ are arranged first and then, they are subtracted from each other, see equation (6.24).

The term $E_{j+1}(z^{-1}) - E_j(z^{-1})$ is of degree j , since $E_j(z^{-1})$ —as already explained above—is of degree $j - 1$. Thus, one can write equation (6.25), in which $\tilde{R}(z^{-1})$ is a polynomial of degree $\leq j - 1$ and r_j is a real number. This leads to the Diophantine equation (6.26).

From $\tilde{a}_0 = 1$ follows that $\tilde{R}(z^{-1}) = 0$ because of the shifting of the remaining terms with z^{-j} . As a result, the equation is simplified to equation (6.27).

为了求得方程的最小值，可将其导数置零，见方程 (6.21)。求解上述方程中的 \mathbf{u} ，即可得到最优的、未来的控制变量表达式 (6.22)。由于仅需要 \mathbf{u} 的第一个元素，该元素实际上是 $\Delta u(t)$ ，从而根据方程 (6.22) 可得到在下一采样周期作用于被控系统的控制变量的控制律，见方程 (6.23)，式中， \mathbf{g}^T 是矩阵 $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}^T$ 的第一行元素。

丢番图方程的递归

为了计算 $E_j(z^{-1})$ 和 $F_j(z^{-1})$ ，必须求解丢番图方程 (6.5)。虽说也存在无需求解丢番图方程的其他 GPC 方法，如学者 Albertos 和 Ortega [3] 的研究成果；但是，接下来将讨论经典的求解方法。为此，首先列出 j 和 $j + 1$ 时刻的丢番图方程，然后对其进行求差运算，具体过程见方程 (6.24)。

$E_{j+1}(z^{-1}) - E_j(z^{-1})$ 项的阶数为 j ，如前所述， $E_j(z^{-1})$ 的阶数为 $j - 1$ 。从而可定义方程 (6.25)，其中，多项式 $\tilde{R}(z^{-1})$ 的阶数 $\leq j - 1$ ，且 r_j 全部为实数。从而可得下述形式的丢番图方程 (6.26)。

由于 \tilde{a}_0 的值为 1，那么多项式 $\tilde{R}(z^{-1})$ 的值必须为零，因余项进行了 z^{-j} 的偏移。从而，方程可进一步简化为方程 (6.27)。

Wegen $\tilde{a}_0 = 1$ folgen weiterhin aus dem Vergleich der Koeffizienten obiger Polynomgleichung die Formeln

$$f_{j,0} = r_j \quad (6.28)$$

$$f_{j+1,i} = f_{j,i+1} - r_j \tilde{a}_{i+1} \quad (6.29)$$

Aus (6.25), (6.28) und der oben erwähnten Tatsache, dass $\tilde{R}(z^{-1}) = 0$ ist, erhält man die Rekursionsvorschrift für die Polynome $E_j(z^{-1})$.

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + f_{j,0} z^{-j} \quad (6.30)$$

Die Endbedingungen der Rekursionsvorschrift ergeben sich aus (6.5) für $j = 1$ und unter der Berücksichtigung von $\tilde{a}_0 = 1$:

$$E_1(z^{-1}) = e_0 = 1 \quad (6.31)$$

$$F_1(z^{-1}) = z(1 - \tilde{A}(z^{-1})) \quad (6.32)$$

Aus (6.30) ist unschwer zu erkennen, dass sämtliche Koeffizienten des Polynoms $E_{j+1}(z^{-1})$ von 0 bis $-(j-1)$ mit denen des Polynoms $E_j(z^{-1})$ identisch sind, d. h. die beiden Polynome unterscheiden sich nur durch den bei $E_{j+1}(z^{-1})$ neu hinzugefügten Term $f_{j,0} z^{-j}$. Folglich kann man die einzelnen $E_j(z^{-1})$ -Polynome schreiben:

$$E_1(z^{-1}) = e_0 \quad (6.33)$$

$$E_2(z^{-1}) = e_0 + e_1 z^{-1} \quad (6.34)$$

$$E_3(z^{-1}) = e_0 + e_1 z^{-1} + e_2 z^{-2} \quad (6.35)$$

⋮

$$E_j(z^{-1}) = e_0 + e_1 z^{-1} + e_2 z^{-2} + \dots + e_{j-1} z^{-(j-1)} \quad (6.36)$$

$$\begin{aligned} E_{j+1}(z^{-1}) &= e_0 + e_1 z^{-1} + e_2 z^{-2} + \dots + e_{j-1} z^{-(j-1)} + e_j z^{-j} \\ &= E_j(z^{-1}) + e_j z^{-j} \end{aligned} \quad (6.37)$$

wobei

$$e_j = f_{j,0}$$

Berechnung der freien und der erzwungenen Antwort

Die Prädiktionsgleichung (6.10) wird in die erzwungene und die freie Antwort aufgeteilt (Gl. (6.16)). Eine Berechnung der $G_j(z^{-1})$ -Polynome erübrigt sich, wenn man die Matrix \mathbf{G} und den Vektor \mathbf{f} einfacher einzeln berechnen kann.

Because of $\tilde{a}_0 = 1$, the following expressions result by comparing the coefficients of the above polynomial equation, see [equations \(6.28\) and \(6.29\)](#).

From [\(6.25\)](#), [\(6.28\)](#) and the abovementioned fact that $\tilde{R}(z^{-1}) = 0$, the recursion equation for the polynomials $E_j(z^{-1})$ can be obtained, see [equation \(6.30\)](#).

The final terms of the recursion equations can be obtained from [\(6.5\)](#) for $j = 1$ and under consideration of $\tilde{a}_0 = 1$, see [equations \(6.31\) and \(6.32\)](#).

From [\(6.30\)](#) it can easily be seen that all coefficients of the polynomial $E_{j+1}(z^{-1})$ from 0 to $-(j-1)$ are identical to the ones of the polynomial $E_j(z^{-1})$, i.e. both polynomials differ only in the term $f_{j,0}z^{-j}$, which is newly added to the polynomial $E_{j+1}(z^{-1})$. Hence, the single polynomials $E_j(z^{-1})$ can be written as [equations \(6.33\) to \(6.37\)](#).

Calculation of the free and forced response

The prediction equation [\(6.10\)](#) is split into the forced and into the free response [\(6.16\)](#). A calculation of the polynomials $G_j(z^{-1})$ is unnecessary if it is easier to calculate the matrix \mathbf{G} and the vector \mathbf{f} individually.

由于 $\tilde{a}_0 = 1$ ，通过对比上述多项式方程的系数，可得到[关系式 \(6.28\) 和 \(6.29\)](#)。

根据方程 [\(6.25\)](#)，[\(6.28\)](#) 以及前面所提及 $\tilde{R}(z^{-1})$ 的值为零这一事实，即可得到适用于多项式 $E_j(z^{-1})$ 的[递归方程 \(6.30\)](#)。

上述递归方程的终止条件可由 $j = 1$ 时的方程 [\(6.5\)](#) 结合 $\tilde{a}_0 = 1$ 这一条件获得，见方程 [\(6.31\)](#) 和 [\(6.32\)](#)。

由方程 [\(6.30\)](#) 不难看出，多项式 $E_{j+1}(z^{-1})$ 从 0 至 $-(j-1)$ 的系数与多项式 $E_j(z^{-1})$ 的系数一致。换言之，上述两个多项式之间的区别仅在于 $E_{j+1}(z^{-1})$ 中新增加的项 $f_{j,0}z^{-j}$ 。进而，可以逐一列出 j 由 1 至 $j+1$ 变化时的多项式 $E_j(z^{-1})$ 表达式，见方程 [\(6.33\)](#) 至 [\(6.37\)](#)。

自由响应与强制响应的计算

预测方程 [\(6.10\)](#) 可以分解为自由响应和强制响应，见方程 [\(6.16\)](#)。如果可以轻易计算出矩阵 \mathbf{G} 和矢量 \mathbf{f} ，那么，则可省去多项式 $G_j(z^{-1})$ 的计算。

Zuerst wird \mathbf{G} bestimmt. Die einzelnen Zeilen dieser Matrix berechnet man aus der Vorschrift $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$:

$$\begin{aligned} G_j(z^{-1}) &= E_j(z^{-1})B(z^{-1}) \\ &= \left(e_0 + e_1z^{-1} + e_2z^{-2} + \dots + e_{j-1}z^{-(j-1)} \right) \\ &\quad \cdot \left(b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb} \right) \end{aligned} \quad (6.38)$$

Die Koeffizienten des gemäß obiger Vorschrift hergeleiteten Polynoms $G_j(z^{-1})$ trägt man dann in die j te Zeile der Matrix \mathbf{G} ein. Da diese Matrix für die Berechnung der erzwungenen Antwort verwendet werden soll, dürfen in die Formel jedoch keine Terme $u(t+j)$ eingehen, für die $j < 0$ gilt, da diese Werte in der Vergangenheit liegen würden und demnach nicht für die erzwungene, sondern für die freie Antwort relevant wären. Daher darf das Polynom $G_j(z^{-1})$ nur bis zum Grad $j-1$ berücksichtigt werden. Wie aus der obigen Berechnungsvorschrift zu erkennen ist, sind in diesem Fall die Matrixelemente $g_{j,0} \dots g_{j,j-1}$ und $g_{j+1,0} \dots g_{j+1,j-1}$ alle identisch, d. h. die Matrixzeilen unterscheiden sich nur durch das hinzugefügte Element $g_{j+1,j}$. Demnach gilt:

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & 0 & \dots & 0 \\ g_1 & g_0 & 0 & \dots & 0 \\ g_2 & g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_p-1} & g_{N_p-2} & g_{N_p-3} & \dots & g_0 \end{bmatrix} \quad (6.39)$$

mit

$$g_i = e_0b_i + b_1b_{i-1} + \dots + e_ib_0$$

Ist dabei $i > nb$, so entfallen alle Summanden mit $b_k, k > nb$.

Zur Berechnung der freien Antwort wird nun folgender Weg eingeschlagen: Die alten \mathbf{y} - und \mathbf{u} -Werte werden zu einen neuen Vektor $\mathbf{y}\mathbf{u}$ zusammengefasst. Entsprechend wird eine Matrix $\mathbf{F}\mathbf{G}'$ aus den Koeffizienten der Polynommatrizen $\mathbf{F}(z^{-1})$ und $\mathbf{G}'(z^{-1})$ zusammengesetzt, so dass gilt:

First, the matrix \mathbf{G} is determined. The individual rows of this matrix can be calculated from the expression $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$, see [equation \(6.38\)](#).

The coefficients of the polynomial $G_j(z^{-1})$, which was derived according to the above expression, are then inserted into the j th row of the matrix \mathbf{G} . Since this matrix should be used for the calculation of the forced response, no terms $u(t+j)$ with $j < 0$ shall enter the equation, because these values would be in the past and thus, they would not be relevant for the forced, but for the free response. Therefore, the polynomial $G_j(z^{-1})$ may only be taken into consideration up to the degree $j - 1$. As it can be seen in the upper expression, in this case, the matrix elements $g_{j,0} \dots g_{j,j-1}$ and $g_{j+1,0} \dots g_{j+1,j-1}$ are all identical, i. e. the matrix rows differ only in the newly added element $g_{j+1,j}$. Therefore, see [equation \(6.39\)](#).

If $i > nb$, all summations with $b_k, k > nb$ are void.

Now the following method is adopted for the calculation of the free response: The old \mathbf{y} - and \mathbf{u} -values are combined to a new vector \mathbf{yu} . Accordingly, a matrix \mathbf{FG}' is composed from the coefficients of the polynomial matrices $\mathbf{F}(z^{-1})$ and $\mathbf{G}'(z^{-1})$ which results in:

首先确定矩阵 \mathbf{G} 。矩阵 \mathbf{G} 的每一行元素可根据关系式 $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$ 求得，见方程 [\(6.38\)](#)。

基于上述关系式推导出的多项式 $G_j(z^{-1})$ 的系数代表矩阵 \mathbf{G} 的第 j 行。因矩阵 \mathbf{G} 用于计算强制响应部分，所以方程中不允许出现 $j < 0$ 时的项 $u(t+j)$ ，原因是它代表过去的控制变量，从而与强制响应部分无关，仅与自由响应部分相关。所以，仅考虑阶数小于 $j - 1$ 的多项式 $G_j(z^{-1})$ 即可。由上述计算法则可以看出，矩阵元素 $g_{j,0} \dots g_{j,j-1}$ 和 $g_{j+1,0} \dots g_{j+1,j-1}$ 是一致的，这也意味着矩阵行之间的差别也仅在于所增加的元素 $g_{j+1,j}$ 。从而，矩阵 \mathbf{G} 可表示为 [\(6.39\)](#)。

如果 $i > nb$ ，那么则消去所有带有 $b_k, k > nb$ 的项。

建议采用下述方式计算自由响应：将上一时刻的 \mathbf{y} 和 \mathbf{u} 的值合并为一个新的矢量 \mathbf{yu} 。与之对应，矩阵 \mathbf{FG}' 可由多项式矩阵 $\mathbf{F}(z^{-1})$ 和 $\mathbf{G}'(z^{-1})$ 的系数合并得到，如下所示：

$$\mathbf{f} = \mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t-1) = \mathbf{F}\mathbf{G}' \cdot \mathbf{y}\mathbf{u} \quad (6.40)$$

wobei

$$\mathbf{F}(z^{-1}) = \begin{bmatrix} F_1(z^{-1}) \\ F_2(z^{-1}) \\ \vdots \\ F_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} f_{1,0} + f_{1,1}z^{-1} + \cdots + f_{1,na}z^{-na} \\ f_{2,0} + f_{2,1}z^{-1} + \cdots + f_{2,na}z^{-na} \\ \vdots \\ f_{N_p,0} + f_{N_p,1}z^{-1} + \cdots + f_{N_p,na}z^{-na} \end{bmatrix}$$

$$\mathbf{G}'(z^{-1}) = \begin{bmatrix} g'_{1,0} + g'_{1,1}z^{-1} + \cdots + g'_{1,nb-1}z^{-(nb-1)} \\ g'_{2,0} + g'_{2,1}z^{-1} + \cdots + g'_{2,nb-1}z^{-(nb-1)} \\ \vdots \\ g'_{N_p,0} + g'_{N_p,1}z^{-1} + \cdots + g'_{N_p,nb-1}z^{-(nb-1)} \end{bmatrix}$$

$$\mathbf{F}\mathbf{G}' = \begin{bmatrix} f_{1,0} & f_{1,1} & \cdots & f_{1,na} & g'_{1,0} & g'_{1,1} & \cdots & g'_{1,nb-1} \\ f_{2,0} & f_{2,1} & \cdots & f_{2,na} & g'_{2,0} & g'_{2,1} & \cdots & g'_{2,nb-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{N_p,0} & f_{N_p,1} & \cdots & f_{N_p,na} & g'_{N_p,0} & g'_{N_p,1} & \cdots & g'_{N_p,nb-1} \end{bmatrix}$$

$$\mathbf{y}\mathbf{u} = \begin{bmatrix} y(t) \\ y(t-1) \\ \cdots \\ y(t-na) \\ \Delta u(t-1) \\ \Delta u(t-2) \\ \cdots \\ \Delta u(t-nb) \end{bmatrix}$$

Die einzelnen Elemente $g'_{j,i}$ berechnen sich wie die Elemente von \mathbf{G} mit Hilfe der Multiplikation von $E_j(z^{-1})$ und $B(z^{-1})$, jedoch sind in diesem Fall nur die Werte interessant, für die die $u(t+j)$ -Werte in der Vergangenheit liegen. Die Berechnungsvorschrift lautet daher:

$$g'_{j,i} = \sum_{k=0}^{j-1} e_k b_{j+i-k} \quad j = 1 \dots N_p; \quad i = 0 \dots (nb-1) \quad (6.41)$$

Da auch hier Summanden mit $b_k, k > nb$ nicht existieren können, müssen alle Terme mit $j+i-k > nb$ entfallen. Wegen des Grades von $E_j(z^{-1})$ läuft die Summe nur über $k = 0 \dots (j-1)$, da für höhere k die e_k nicht mehr existent sind.

Die berechnete freie Antwort \mathbf{f} kann nun in Gl. (6.23) eingesetzt und die Regelgröße $u(t)$ für den nächsten Abtastschritt ermittelt werden.

See equation (6.40).

The individual elements $g'_{j,i}$ are calculated in the same way as the elements of \mathbf{G} with the multiplication of $E_j(z^{-1})$ and $B(z^{-1})$. However, in this case, only the values of $u(t+j)$ that are in the past are of interest. Hence, the expression for the calculation is

$$g'_{j,i} = \sum_{k=0}^{j-1} e_k b_{j+i-k} \quad j = 1 \dots N_p; \quad i = 0 \dots (nb - 1) \quad (6.41)$$

As summands with $b_k, k > nb$ can also not exist in this case, all terms with $j+i-k > nb$ must be omitted. Because of the degree of $E_j(z^{-1})$, the summation is only valid for $k = 0 \dots (j-1)$, since for higher values of k the terms e_k do not exist anymore.

Now the calculated free response \mathbf{f} can be used in equation (6.23) and the value of the actuating variable $u(t)$ for the next sampling cycle can be determined.

见方程 (6.40)。

类似于矩阵 \mathbf{G} 中元素的计算, 单个元素 $g'_{j,i}$ 的计算借助于 $E_j(z^{-1})$ 和 $B(z^{-1})$ 的乘积。但此时我们更感兴趣的是位于过去时刻的 $u(t+j)$, 计算法则如下:

$$g'_{j,i} = \sum_{k=0}^{j-1} e_k b_{j+i-k} \quad j = 1 \dots N_p; \quad i = 0 \dots (nb - 1) \quad (6.41)$$

因为带有 $b_k, k > nb$ 的项实际上不存在, 所以必须消去所有 $j+i-k > nb$ 的项。由于 $E_j(z^{-1})$ 的阶数为 $j-1$, 求和的范围为 $k = 0 \dots (j-1)$, 那么对于更高的 k 来说, e_k 也是不存在的。

将计算出的自由响应 \mathbf{f} 代入方程 (6.23), 即可得到用于下一采样周期的控制变量 $u(t)$ 。

6.1.2 Experimentelle Ergebnisse/Experimental results/实验结果

Da die praktischen Versuchen zeigten, dass sich ein GPC-Regler ohne Stabilisierung durch ein Design-Polynom $T(z^{-1})$ wegen zu hoher Empfindlichkeit gegenüber Modellabweichungen oder Störeinflüssen nicht sinnvoll betreiben lässt, gibt es an dieser Stelle keine Messergebnisse.

6.2 GPC mit Filter/GPC with filter/带滤波器的 GPC

Ist die gemessene Regelgröße überschwingungsbehaftet, und ein überschwingungsfreies Abtasten kann nicht gewährleistet werden, so führen die Überschwingungen auf dem Messsignal zu einer unruhigen Stellgröße. Besonders bei sehr schnell reagierenden Reglern wie den MPC-Verfahren führt dies zu einem starken Schwingen in der Stellgröße, da der Regler versucht, der scheinbaren Regelabweichung entgegenzuwirken. Ein ähnliches Problem tritt auf, wenn Modellfehler existieren. In diesem Fall gibt es zwei Abhilfemaßnahmen. Entweder, man macht den Regler träger, so dass er nicht mehr auf höherfrequente Anteile in der Regelabweichung reagiert, oder in den Rückführungsweig wird ein Tiefpassfilter eingefügt. Im zweiten Fall ist jedoch zu bedenken, dass das Tiefpassfilter, wenn es nicht im Regler berücksichtigt wird, zu einem instabilen Verhalten des gesamten Systems führen kann, da die vom Regler registrierte Größe bei dynamischen Vorgängen nicht mehr mit dem tatsächlichen Istwert übereinstimmt. Folglich muss auch bei Glättung des rückgeführten Istwerts mittels Filter der Regler entsprechend angepasst werden, was in der Realität ebenfalls bedeutet, ihn langsamer zu machen. Bei extrem großen Filterzeitkonstanten führt dies dazu, dass nicht mehr die eigentliche Strecke, sondern das Tiefpassfilter geregelt wird, wodurch die Dynamik des Gesamtsystems leidet.

GPC-Regler bieten die Möglichkeit, das Filter in den Regler zu integrieren. Das Tiefpassverhalten wird quasi im reglerinternen Modell berücksichtigt, so dass sich eine verringerte Empfindlichkeit des Reglers für hochfrequente Überschwingungen *ohne eine Beeinflussung der Gesamtdynamik* ergibt. Tatsächlich verhält sich das GPC-Modell in diesem Fall wie ein Kalman-Filter [72], das ja bekanntlich auch ein verzögerungsfreies Filter darstellt.

Um die Idee des GPC-internen Filters leichter zu verstehen, wird die MPC-Struktur aus Abb. 5.1 gemäß Abb. 6.2 umgestellt. Wie bereits erläutert, wird aus der Stellgröße $u(t)$ und dem Istwert $y(t)$ die totale Antwort des Systems vorausberechnet und dieser Wert im Regelgesetz unter Berücksichtigung des Sollwerts $w(t)$ in optimale, zukünftige Stellgrößen umgesetzt. Die Abbildung zeigt außerdem die Möglichkeit, ein parameteradaptives Streckenmodell zu verwenden, worauf hier jedoch nicht näher eingegangen wird.

6.1.2 Experimental results

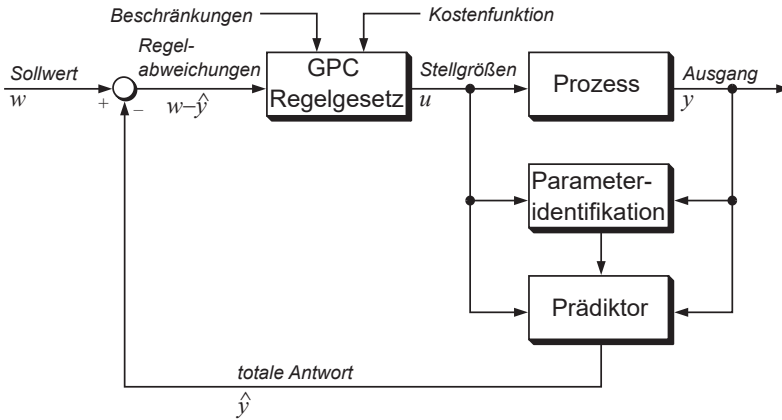
Since practical experiments showed that a GPC controller without stabilization through a design polynomial $T(z^{-1})$ cannot be practically used because of the highly increased sensitivity related to model deviations or noise influences, there are no measurement results at this point.

6.2 GPC with filter

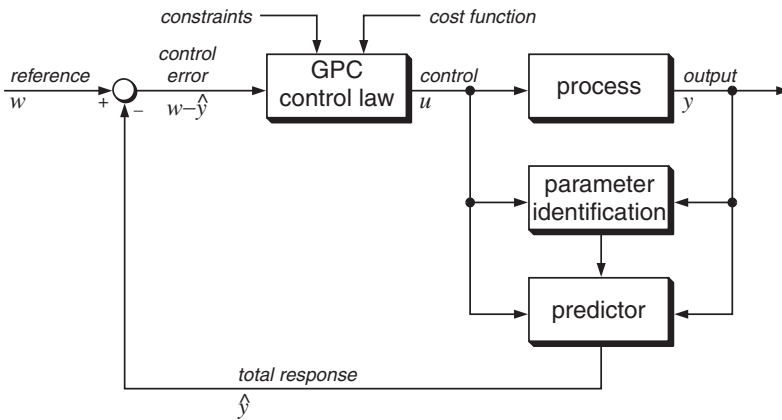
If the measured controlled variable contains harmonics and harmonic-free sampling cannot be guaranteed, then the harmonics on the measuring signal lead to a disturbed actuating variable. Especially in very fast acting controllers like MPC methods, this leads to strong oscillations in the actuating variable, since the controller tries to counteract against the apparent control deviation. A similar problem appears if there are errors in the model. In this case, there are two workarounds. Either the controller can be made more sluggish so that it does not react to higher frequencies in the control deviation anymore, or a low-pass filter is inserted into the feedback path. For the latter case, however, it shall be kept in mind that the low-pass filter, if not considered in the controller design, can lead to an unstable behavior of the entire system, since in dynamic operation the feedback value of the controlled variable is no more equal to the real value of the controlled variable. Hence, if the fed back values of the controlled variable are smoothed via a filter, the controller has to be adapted correspondingly which in reality means to make it slower. Considering extremely large filter time constants, this leads to the fact that not the actual plant itself, but the low-pass filter is controlled, whereby the dynamics of the overall system suffer to a great extent.

GPC controllers offer the possibility to integrate a filter into the controller itself. The low-pass filter behavior is considered in the controller's internal model, so that a reduced sensitivity of the controller for high frequency harmonics results *without influencing the overall dynamics*. In this case, the GPC model actually behaves like a Kalman filter [72], which is also well known for its delay-free filtering properties.

For a better understanding of GPC with internal filter, the MPC structure from Figure 5.2 is modified according to Figure 6.2. As already explained, the complete system response is precalculated from the actuating variable $u(t)$, the controlled variable $y(t)$ and this response is used in the control law for calculating optimum future values for the actuating variables under consideration of the reference value $w(t)$. In addition, Figure 6.3 shows the possibility to use a parameter adaptive plant model, but this shall not be discussed further.



6.2: Struktur eines GPC-Reglers ohne Filter



6.3: Structure of a GPC controller without filter

6.1.2 实验结果

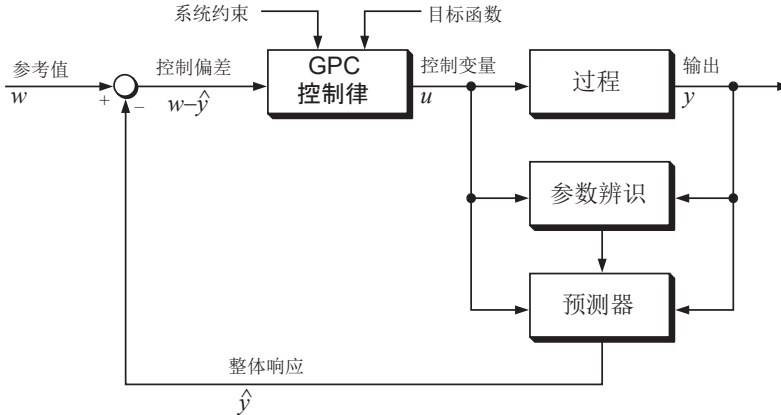
实验表明，未通过设计多项式 $T(z^{-1})$ 进行稳定化处理的 GPC 控制器对于模型偏差或扰动具有很高的灵敏度，因此无法正常运行，从而此处未能提供相应的实验测量结果。

6.2 带滤波器的 GPC

如果测量得到的被控变量包含谐波，并且无法保证在单个周期内消除谐波，那么测量信号所包含的谐波会引起控制变量的剧烈变化，尤其是在具有快速响应特性的控制器情形，如 MPC 方法。上述谐波会引起控制变量的强烈振荡，因为控制器尝试迅速消除所出现的控制偏差。另外，当模型失配时也会出现类似的问题。有两种解决方案可供选择：一是使控制器变得迟缓，使其无法对控制偏差的高频分量作出及时响应；二是在反馈通道引入低通滤波器。应用第二种方案时需额外注意的是，如果低通滤波器并未考虑至控制器，那么有可能引起整个系统的不稳定，因为在动态过程中反馈至控制器的变量与实际的被控变量将不再保持一致。为此，在借助于滤波器平滑待反馈的控制变量时，控制器也应进行相应调整，这实际上也意味着使控制器变得迟缓。在非常大的滤波时间常数情况下，会导致所控制的是滤波器，而非真实的被控系统，从而使得整个系统的动态特性受到影响。

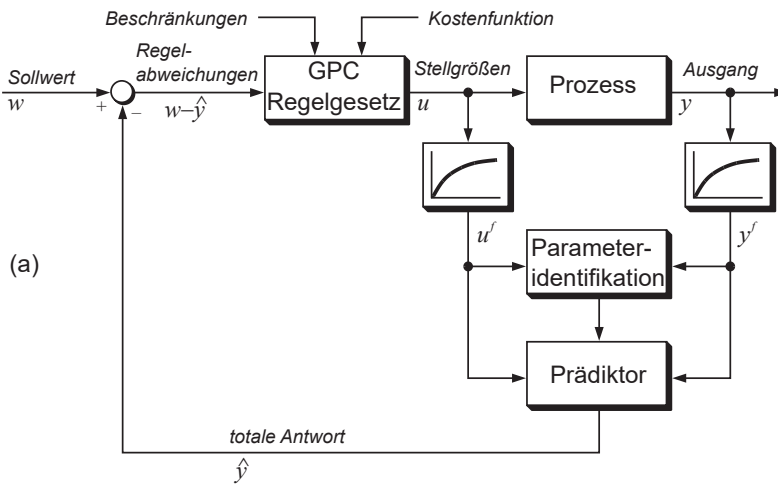
GPC 控制器提供了可将滤波器整合至控制器的可能性。通过在控制器内部模型考虑低通特性，可使控制器对高频谐波的灵敏度降低，且不影响整体的动态性能。此时的 GPC 控制器实际上类似于卡尔曼滤波器 [72]，它表示的是一个无延迟的滤波器。

为了使 GPC 内部滤波器的概念更为容易理解，可将图 5.3 所示的 MPC 控制器按照图 6.4 的方式重新布局。如前所述，基于控制变量 $u(t)$ 和被控变量 $y(t)$ 预测得到系统的整体响应，控制器的控制律基于偏差变量得到最优的、未来时刻的控制变量。图 6.4 同时表明，使用一个参数自适应的被控系统模型也是可行的，此处不进行详细探讨。



6.4: 一种无滤波的 GPC 控制器结构

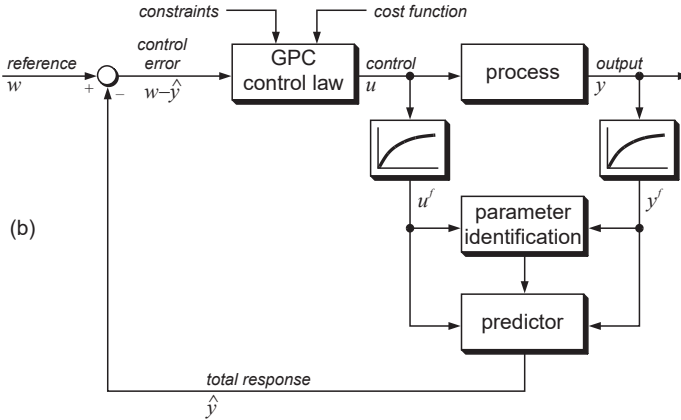
Sollen nun die überschwingungsbehafteten Messwerte zur Unterdrückung der unerwünschten, hochfrequenten Störanteile gefiltert werden, so wird die Regelungsstruktur gemäß Abb. 6.5 durch Einfügen von zwei Tiefpassfiltern mit identischen Zeitkonstanten abgeändert. Sowohl der gemessene Istwert $y(t)$ als auch die Stellgröße $u(t)$ werden in gleicher Weise gefiltert. Die gefilterten Werte $u^f(t)$ und $y^f(t)$ werden dann zur Parameterschätzung und Vorausberechnung der totalen Antwort verwendet. Durch marginale Änderungen in der Struktur des GPC-Regelgesetzes, der Parameteridentifikation und des Prädiktors wird im GPC-Regler die Existenz der Filterung berücksichtigt, wodurch – bezogen auf das Gesamtsystem – ein verzögerungsfreies Herausfiltern der unerwünschten Überschwingungen möglich ist.



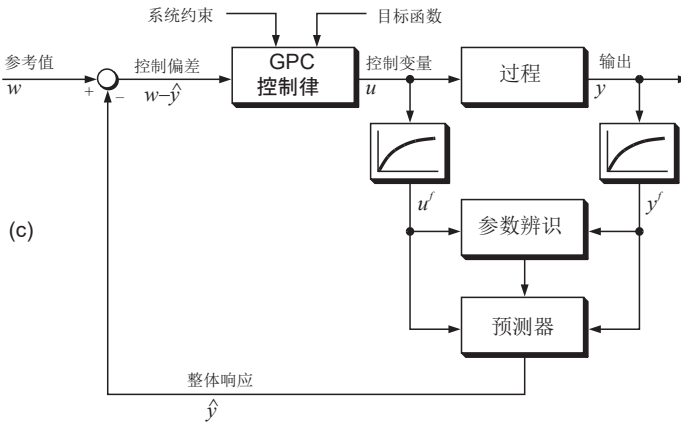
6.5: Struktur eines GPC-Reglers mit Filter

If the measuring signals containing high-frequency noise should be filtered for the suppression of the unwanted, high-frequency disturbances, the controller structure has to be modified by inserting two low-pass filters with identical time constants as shown in Figure 6.6. The measured values of the actual controlled variable $y(t)$ as well as the values of the actuating variable $u(t)$ are filtered in the same way. Then, the filtered values $u^f(t)$ and $y^f(t)$ are used for parameter estimation and prediction of the complete response. With marginal changes in the structure of the GPC control law, in the parameter identification and in the predictor in the GPC controller, filtering can be taken into account, which allows a—related to the entire system—delay-free filtering of unwanted harmonics.

如果要对包含谐波的测量值进行滤波，以抑制不希望出现的高频扰动部分，需将图 6.4 修正为图 6.7 所示的形式，也即引入两个具有相同时间常数的低通滤波器。由图可知，测量的被控变量 $y(t)$ 和控制变量 $u(t)$ 均以相同的方式进行滤波，经滤波得到的 $u^f(t)$ 和 $y^f(t)$ 将用于参数估计及整体响应的预测。通过改变 GPC 控制律、参数辨识以及预测器的结构，可使 GPC 控制器有能力考虑到滤波的因素。从而对于整个系统来说，无延迟地消除不希望出现的谐波也是有可能的。



6.6: Structure of a GPC controller with filter



6.7: 一种带滤波的 GPC 控制器结构

6.2.1 Mathematische Herleitung/Mathematical derivation/数学推导

Das CARIMA-Modell

Das eigentliche CARIMA-Modell aus Kap. 6.1.1 ändert sich durch Einfügen eines Tiefpassfilters vom Prinzip her nicht. Allerdings kann nun in Formel (6.2) nicht mehr davon ausgegangen werden, dass $\xi(t)$ weißes Rauschen darstellt. Analog darf das Polynom $C(z^{-1})$ nicht gleich 1 gesetzt werden, da die Störungen ja nun explizit berücksichtigt werden sollen. Da sie jedoch nur in den seltensten Fällen exakt angegeben werden können, behilft man sich, indem man $C(z^{-1})$ durch ein *Design-Polynom* $T(z^{-1})$ ersetzt. Dieses wird so gewählt, dass es als Tiefpassfilter wirkt. Demnach ergibt sich Gl. (6.3) nun zu

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + T(z^{-1})\frac{\xi(t)}{\Delta} \quad (6.42)$$

mit

$$T(z^{-1}) = t_0 + t_1 z^{-1} + t_2 z^{-2} + \dots + t_n z^{-n}$$

wobei bei einem Filter normalerweise $t_0 = 1$ angesetzt wird.

Der j-Schritt-voraus-Prädiktor

Unter Berücksichtigung des Design-Polynoms ändert sich die bereits aus Gl. (6.5) bekannte Diophantische Gleichung zu

$$T(z^{-1}) = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \quad (6.43)$$

Mit Hilfe derselben Umformungen, wie sie in Kap. 6.1.1 auf Seite 78 f. beschrieben werden, ergibt sich aus obiger Gleichung und (6.42):

$$\begin{aligned} T(z^{-1})y(t+j) &= E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) \\ &\quad + F_j(z^{-1})y(t) \\ &\quad + T(z^{-1})E_j(z^{-1})\xi(t+j) \end{aligned} \quad (6.44)$$

Da auch hier $E_j(z^{-1})$ wieder vom Grad $j-1$ ist, liegen die Rauschkomponenten alle in der Zukunft und können nicht in die Vorausberechnung eingehen.

6.2.1 Mathematical derivation

The CARIMA model

The actual CARIMA model described in Chapter 6.1.1 does in principle not change if a low-pass filter is inserted. However, now it cannot be assumed that $\xi(t)$ in equation (6.2) is white noise. Analog to this, the polynomial $C(z^{-1})$ cannot be set equal to 1, as now the disturbances explicitly should be taken into consideration. However, as the disturbances are normally not exactly known, $C(z^{-1})$ is replaced by a *design polynomial* $T(z^{-1})$. This is selected in such a way that it acts like a low-pass filter. Therefore, equation (6.3) can be modified to equation (6.42), in which for a filter normally $t_0 = 1$ is set.

The j-step ahead predictor

Taking the design polynomial into consideration, the Diophantine equation, which is already known from equation (6.5), changes to equation (6.43).

With the help of the same transformations as already described in Chapter 6.1.1 on page 78 et seq., the following equation can be obtained from the above equation and (6.42), see equation (6.44).

Since here $E_j(z^{-1})$ is of degree $j - 1$, too, all noise components are in the future and thus, they cannot appear in the precalculation.

6.2.1 数学推导

CARIMA 模型

从原理上讲, 章节 6.1.1 介绍的 CARIMA 模型在引入低通滤波器后并未发生变化。显然, 此时不能继续假设方程 (6.2) 中的 $\xi(t)$ 表示白噪声。同理, 多项式 $C(z^{-1})$ 也不可设置为 1, 因为此时要显式地考虑扰动因素。然而, 由于仅在极少数的情况下才能够精确地给出扰动, 所以通常使用一个设计多项式 $T(z^{-1})$ 来代替 $C(z^{-1})$ 。需合理地选择 $T(z^{-1})$, 以使其可以作为低通滤波器。此时, 方程 (6.3) 变化为 方程 (6.42), 并通常设置式中的 $t_0 = 1$ 。

J 步前向预测器

在考虑设计多项式的情况下, 方程 (6.5) 表示的丢番图方程则变化为 方程 (6.43)。

使用与第 78 页章节 6.1.1 所描述的不同变换方法, 根据上述方程和方程 (6.42), 可得方程 (6.44)。

由于此处 $E_j(z^{-1})$ 的阶数也为 $j - 1$, 这也意味着噪声分量全部位于未来时刻, 所以它无法用于预测。

Eine bestmögliche Prädiktion von y kann daher mit

$$T(z^{-1})\hat{y}(t+j) = G_j(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})y(t) \quad (6.45)$$

angegeben werden, wobei wieder

$$G_j(z^{-1}) = E_j(z^{-1})B(z^{-1}) \quad (6.46)$$

gilt. Camacho/Bordons [20] leiten nun mit Hilfe eines direkten Ansatzes über eine Diophantische Gleichung eine optimale Prädiktionsgleichung zur Berechnung von $\hat{y}(t+j)$ direkt aus den Δu - und y -Werten her. Clarke et al. [28] hingegen verwenden statt $y(t)$ und $u(t)$ die mit $1/T(z^{-1})$ gefilterten Werte $y^f(t)$ und $u^f(t)$. Bei dieser Vorgehensweise ergibt sich die Prädiktionsgleichung für $\hat{y}(t+j)$ zu

$$\hat{y}(t+j) = G_j(z^{-1})\Delta u^f(t+j-1) + F_j(z^{-1})y^f(t) \quad (6.47)$$

Da zukünftige Stellgrößen sinnvollerweise *nicht* gefiltert werden, muss das Polynom $T(z^{-1})$ aus den zukünftigen Werten wieder herausgerechnet werden. Dazu wird $G_j(z^{-1})$ zu

$$G_j(z^{-1}) = G_j'(z^{-1})T(z^{-1}) + z^{-j}\Gamma_j(z^{-1}) \quad (6.48)$$

zerlegt. Es ist zu erkennen, dass für den Fall, dass $T(z^{-1}) = 1$ (keine Filterung), $G_j(z^{-1}) = G_j'(z^{-1})$ gilt. Die Prädiktion ist nun durch (6.48) in die freie und die erzwungene Antwort aufgeteilt. Damit ergibt sich die Prädiktionsgleichung zu

$$\hat{y}(t+j) = \underbrace{G_j'(z^{-1})\Delta u(t+j-1)}_{\text{erzwungene Antwort}} + \underbrace{F_j(z^{-1})y^f(t) + \Gamma_j(z^{-1})\Delta u^f(t-1)}_{\text{freie Antwort}} \quad (6.49)$$

Berechnung der Steuergrößen

Die Optimierung selbst läuft genauso wie bei Standard-GPC ohne Filtereigenschaften ab (Kap. 6.1.1 auf Seite 80 ff.). Gl. (6.49) wird zu

$$\mathbf{y} = \mathbf{G}'\mathbf{u} + \mathbf{f}' \quad (6.50)$$

zusammengefasst. Mit Hilfe der Kostenfunktion (6.15) ergibt sich schließlich die Formel zur Vorausberechnung optimaler, zukünftiger Stellgrößen:

$$\mathbf{u} = (\mathbf{G}'^T\mathbf{G}' + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}'^T(\mathbf{w} - \mathbf{f}') \quad (6.51)$$

Wie bei Standard-GPC ist $\mathbf{G}' \neq [G'_1 \ G'_2 \ G'_3 \ \dots]^T$, da \mathbf{G}' nur die zukünftigen Anteile von \mathbf{u} für die erzwungene Antwort berücksichtigt. Auch hier wird für die praktische Regelung nur das erste Element von \mathbf{u} , nämlich $\Delta u(t)$ benötigt. Aus Gl. (6.51) ergibt sich daher für die Stellgröße des nächsten Abtastschritts:

$$u(t) = u(t-1) + \mathbf{g}'^T(\mathbf{w} - \mathbf{f}') \quad (6.52)$$

wobei \mathbf{g}'^T die erste Zeile der Matrix $(\mathbf{G}'^T\mathbf{G}' + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}'^T$ ist.

Hence, the best possible prediction of y is given by equation (6.45), in which again equation (6.46). Now, Camacho/Bordons [20] propose the derivation of an optimum prediction equation for the calculation of $\hat{y}(t+j)$ directly from the Δu - and y -values via a direct approach with a Diophantine equation. However, Clarke et al. [28] use the values $y^f(t)$ and $u^f(t)$, which are filtered with $1/T(z^{-1})$, instead of using $y(t)$ and $u(t)$. Considering this approach, the prediction equation for $\hat{y}(t+j)$ results to equation (6.47).

Since future values of the actuating variables are logically *not* filtered with $T(z^{-1})$, $G_j(z^{-1})$ can be separated in the following way, see equation (6.48).

For the case that $T(z^{-1}) = 1$ (no filtering), the equation $G_j(z^{-1}) = G_j'(z^{-1})$ is valid. Now the prediction is separated into the free and the forced response by equation (6.48). For this reason, the prediction equation results into equation (6.49).

Calculation of the actuating variables: The optimization itself is done in the same way as for standard GPC without filter (Chapter 6.1.1 on page 80 et seq.). Equation (6.49) can be summarized to equation (6.50). With the help of the cost function (6.15), the equation for the precalculation of optimum future values for the actuating variables can be obtained, see equation (6.51).

Analog to standard GPC, it can be stated that $\mathbf{G}' \neq [G'_1 \ G'_2 \ G'_3 \ \dots]^T$ because \mathbf{G}' only considers the future parts of \mathbf{u} for the forced response. Also in this case, only the first element of \mathbf{u} , namely $\Delta u(t)$, is required for the real controller. Hence, the value of the actuating variable for the next sampling cycle results from equation (6.51), see equation (6.52), in which \mathbf{g}^T is the first row of the matrix $(\mathbf{G}'^T \mathbf{G}' + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}'^T$.

对于 y 最为合适的预测表达式为 (6.45), 满足关系式 (6.46)。学者 Camacho/Bordons [20] 借助于直接使用丢番图方程的方法推导出最优的预测方程, 也就是说根据 Δu 和 y 的值直接计算出 $\hat{y}(t+j)$ 。学者 Clarke 等 [28] 则使用经多项式 $1/T(z^{-1})$ 滤波后的 $y^f(t)$ 和 $u^f(t)$ 来代替 $y(t)$ 和 $u(t)$, 基于此方式得到的 $\hat{y}(t+j)$ 的预测方程为 (6.47)。

理论上, 未来时刻的控制变量不应被滤波, 所以需根据未来时刻的值重新计算多项式 $T(z^{-1})$ 。为此, 可将 $G_j(z^{-1})$ 分解为 方程 (6.48)。

可以看出, 对于上述情形, 当 $T(z^{-1}) = 1$ 时 (无滤波), 满足 $G_j(z^{-1}) = G_j'(z^{-1})$ 。根据 方程 (6.48) 可将预测分解为自由响应和强制响应, 从而得到预测方程表达式 (6.49)。

控制变量计算: 优化过程本身与不包含滤波特性的标准 GPC 完全一致, 见第 80 页章节 6.1.1。方程 (6.49) 可综合为 方程 (6.50)。最后, 可根据成本函数 (6.15) 得到用于计算最优的、未来时刻的控制变量表达式, 见 方程 (6.51)。

类似于标准 GPC 情形, $\mathbf{G}' \neq [G'_1 \ G'_2 \ G'_3 \ \dots]^T$, 因为 \mathbf{G}' 仅仅考虑了用于强制响应的 \mathbf{u} 的未来时刻部分。在实际控制中, 仅需要 \mathbf{u} 的首个元素, 也即 $\Delta u(t)$ 。根据 方程 (6.51), 可得到下一采样周期的控制变量的控制律 (6.52), 其中, \mathbf{g}^T 表示矩阵 $(\mathbf{G}'^T \mathbf{G}' + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}'^T$ 的第一行元素。

Rekursion der Diophantischen Gleichung

Gegenüber dem Verfahren ohne Filterung hat sich die die Diophantische Gleichung durch die Berücksichtigung des Design-Polynoms geändert (vergleiche Formel (6.5) mit (6.43)). Dennoch wendet man zur Berechnung von $E_j(z^{-1})$ und $F_j(z^{-1})$ dieselbe Strategie an. Stellt man wie in Kap. 6.1.1 auf Seite 86 f. beschrieben die Diophantischen Gleichungen für j und $j + 1$ auf und subtrahiert sie voneinander, so entfällt $T(z^{-1})$. Die Rekursionsformeln (6.28) bis (6.30) gelten folglich weiterhin.

Die Endbedingungen der Rekursion werden aus Gl. (6.43) für $j = 1$ bestimmt. Es ergeben sich:

$$E_1(z^{-1}) = e_0 = \frac{t_0}{\tilde{a}_0} = 1 \quad \text{da } \tilde{a}_0 = 1 \text{ und } t_0 = 1 \quad (6.53)$$

$$F_1(z^{-1}) = z(T(z^{-1}) - \tilde{A}(z^{-1})) \quad (6.54)$$

Berechnung der freien und der erzwungenen Antwort

Um die Vorausberechnung der freien Antwort zu vereinfachen, werden auch bei Einsatz eines Filters Werte zusammengefasst. In diesem Fall bilden die gefilterten Werte $y^f(t)$ und $\Delta u^f(t)$ aus der Vergangenheit die Vektoren \mathbf{y}^f bzw. \mathbf{u}^f , die dann wiederum zu einem neuen Vektor $\mathbf{y}\mathbf{u}^f$ zusammengefasst werden. An Stelle der Polynommatrizen $\mathbf{F}(z^{-1})$ und $\mathbf{G}'(z^{-1})$ treten nun $\mathbf{F}(z^{-1})$ und $\Gamma(z^{-1})$. Demnach gilt:

$$\mathbf{f}' = \mathbf{F}(z^{-1})\mathbf{y}^f(t) + \Gamma(z^{-1})\Delta u^f(t-1) = \mathbf{F}\Gamma \cdot \mathbf{y}\mathbf{u}^f \quad (6.55)$$

wobei

$$\mathbf{F}(z^{-1}) = \begin{bmatrix} F_1(z^{-1}) \\ F_2(z^{-1}) \\ \vdots \\ F_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} f_{1,0} + f_{1,1}z^{-1} + \cdots + f_{1,na}z^{-na} \\ f_{2,0} + f_{2,1}z^{-1} + \cdots + f_{2,na}z^{-na} \\ \vdots \\ f_{N_p,0} + f_{N_p,1}z^{-1} + \cdots + f_{N_p,na}z^{-na} \end{bmatrix} \quad \text{unverändert}$$

$$\Gamma(z^{-1}) = \begin{bmatrix} \Gamma_1(z^{-1}) \\ \Gamma_2(z^{-1}) \\ \vdots \\ \Gamma_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \gamma_{1,0} + \gamma_{1,1}z^{-1} + \cdots + \gamma_{1,nb-1}z^{-(nb-1)} \\ \gamma_{2,0} + \gamma_{2,1}z^{-1} + \cdots + \gamma_{2,nb-1}z^{-(nb-1)} \\ \vdots \\ \gamma_{N_p,0} + \gamma_{N_p,1}z^{-1} + \cdots + \gamma_{N_p,nb-1}z^{-(nb-1)} \end{bmatrix}$$

Recursion of the Diophantine equation

Compared to GPC without filter, the Diophantine equation has to be changed because of the consideration the design polynomial (compare equation (6.5) with (6.43)). Nevertheless, the same strategy for the calculation of $E_j(z^{-1})$ and $F_j(z^{-1})$ can be used. If the Diophantine equations for j and $j + 1$ are formed and then subtracted from each other as described in Chapter 6.1.1 on page 86 et seq., $T(z^{-1})$ can be omitted. Therefore, the recursion equations (6.28) to (6.30) are also valid in this case.

The final terms of the recursion are obtained from equation (6.43) for $j = 1$ which results in equation (6.53) and (6.54).

Calculation of the free and forced response

In order to simplify the precalculation of the free response if a filter is used, vectors and matrices are combined, too. In this case, the filtered values $y^f(t)$ and $\Delta u^f(t)$ from the past form the vectors \mathbf{y}^f and \mathbf{u}^f which are then again combined to a new vector $\mathbf{y}\mathbf{u}^f$. Instead of the polynomial matrices $\mathbf{F}(z^{-1})$ and $\mathbf{G}'(z^{-1})$ now $\mathbf{F}(z^{-1})$ and $\Gamma(z^{-1})$ are used. Therefore, see equation (6.55).

丢番图方程的递归

相较于无滤波的方法，考虑设计多项式后的丢番图方程也相应发生变化，可对比方程 (6.5) 和 (6.43)。然而，仍然可使用相同的策略计算 $E_j(z^{-1})$ 和 $F_j(z^{-1})$ 。类似于章节 6.1.1 第 86 页所描述的方法，首先列出 j 和 $j + 1$ 时刻的丢番图方程，然后将两个方程相减，从而可消去 $T(z^{-1})$ 。递归方程 (6.28) 至 (6.30) 继续适用。

递归方程的终止条件可由 $j = 1$ 时刻的方程 (6.43) 求得，见方程 (6.53) 和 (6.54)。

自由响应与强制响应的计算

为了简化自由响应的预测，在使用滤波器的情况下，也应同样对相应的数值进行合并。此时，滤波后的过去时刻的 $y^f(t)$ 和 $\Delta u^f(t)$ 构成矢量 \mathbf{y}^f 和 \mathbf{u}^f ，然后再次将其合并为一个新矢量 $\mathbf{y}\mathbf{u}^f$ 。不同于第 91 页中出现的多项式 $\mathbf{F}(z^{-1})$ 和 $\mathbf{G}'(z^{-1})$ ，此处使用了 $\mathbf{F}(z^{-1})$ 和 $\Gamma(z^{-1})$ ，从而有方程 (6.55)。

$$\mathbf{F}\Gamma = \begin{bmatrix} f_{1,0} & f_{1,1} & \cdots & f_{1,na} & \gamma_{1,0} & \gamma_{1,1} & \cdots & \gamma_{1,nb-1} \\ f_{2,0} & f_{2,1} & \cdots & f_{2,na} & \gamma_{2,0} & \gamma_{2,1} & \cdots & \gamma_{2,nb-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{N_p,0} & f_{N_p,1} & \cdots & f_{N_p,na} & \gamma_{N_p,0} & \gamma_{N_p,1} & \cdots & \gamma_{N_p,nb-1} \end{bmatrix}$$

$$\mathbf{y}\mathbf{u}^f = \begin{bmatrix} y^f(t) \\ y^f(t-1) \\ \cdots \\ y^f(t-na) \\ \Delta u^f(t-1) \\ \Delta u^f(t-2) \\ \cdots \\ \Delta u^f(t-nb) \end{bmatrix}$$

Wie Gl. (6.49) zeigt, besteht die Prädiktion aus einem Teil für die erzwungene und einem für die freie Antwort. Zuerst wird wie bei der „filterlosen“ Version (s. Kap. 6.1.1 auf Seite 88 ff.) angesetzt

$$\begin{aligned} G_j(z^{-1}) &= E_j(z^{-1})B(z^{-1}) \\ &= \left(e_0 + e_1z^{-1} + e_2z^{-2} + \cdots + e_{j-1}z^{-(j-1)} \right) \\ &\quad \cdot \left(b_0 + b_1z^{-1} + b_2z^{-2} + \cdots + b_{nb}z^{-nb} \right) \end{aligned} \quad (6.56)$$

Unter Berücksichtigung von Gl. (6.48) erhält man

$$E_j(z^{-1})B(z^{-1}) = G_j'(z^{-1})T(z^{-1}) + z^{-j}\Gamma_j(z^{-1}) \quad (6.57)$$

Da \mathbf{G}' für die Prädiktion der erzwungenen Antwort dienen soll, dürfen in $G_j'(z^{-1})$ keine Terme $u(t+j)$, $j < 0$ eingehen. Es gilt daher genau wie bei \mathbf{G} , dass

1. die Matrixelemente $g'_{j,0} \cdots g'_{j,j-1}$ und $g'_{j+1,0} \cdots g'_{j+1,j-1}$ alle identisch sind und
2. die Matrixzeilen sich nur durch das hinzugefügte Element $g'_{j+1,j}$ unterscheiden.

Daher läßt sich \mathbf{G}' definieren:

$$\mathbf{G}' = \begin{bmatrix} g'_0 & 0 & 0 & \cdots & 0 \\ g'_1 & g'_0 & 0 & \cdots & 0 \\ g'_2 & g'_1 & g'_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g'_{N_p-1} & g'_{N_p-2} & g'_{N_p-3} & \cdots & g'_0 \end{bmatrix}$$

As equation (6.49) shows, the prediction consists of a part for the forced and a part for the free response. First, in the same way as for the “filterless” case (see Chapter 6.1.1 on page 88 et seqq.), the following approach is made, see equation (6.56).

Under consideration of equation (6.48), equation (6.57) can be obtained. Since \mathbf{G}' should be used for the prediction of the forced response, terms $u(t+j)$ with $j < 0$ must not appear in $G'_j(z^{-1})$. Therefore, in the same way as for \mathbf{G} , it can be stated that

1. the matrix elements $g'_{j,0} \cdots g'_{j,j-1}$ and $g'_{j+1,0} \cdots g'_{j+1,j-1}$ are all identical and
2. the matrix rows only differ in the added element $g'_{j+1,j}$.

Hence, \mathbf{G}' can be defined as :

$$\mathbf{G}' = \begin{bmatrix} g'_0 & 0 & 0 & \cdots & 0 \\ g'_1 & g'_0 & 0 & \cdots & 0 \\ g'_2 & g'_1 & g'_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g'_{N_p-1} & g'_{N_p-2} & g'_{N_p-3} & \cdots & g'_0 \end{bmatrix}$$

正如方程 (6.49) 表明, 预测包含自由响应和强制响应两个部分。类似于章节 6.1.1 第 88 页所述无滤波的情形, 首先假定关系式 (6.56)。

结合方程 (6.48), 可得到方程 (6.57)。由于 \mathbf{G}' 用于强制响应的预测, 所以在 $G'_j(z^{-1})$ 中不允许出现 $j < 0$ 时的项 $u(t+j)$ 。这与计算 \mathbf{G} 时的情形完全一致, 且满足:

1. 矩阵元素 $g'_{j,0} \cdots g'_{j,j-1}$ 和 $g'_{j+1,0} \cdots g'_{j+1,j-1}$ 是一致的;
2. 矩阵行与行之间的差别仅在于增加的元素 $g'_{j+1,j}$ 。

从而, \mathbf{G}' 可定义为:

$$\mathbf{G}' = \begin{bmatrix} g'_0 & 0 & 0 & \cdots & 0 \\ g'_1 & g'_0 & 0 & \cdots & 0 \\ g'_2 & g'_1 & g'_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g'_{N_p-1} & g'_{N_p-2} & g'_{N_p-3} & \cdots & g'_0 \end{bmatrix}$$

Zur Bestimmung der Elemente von $\mathbf{F}\Gamma$ empfiehlt sich der von Clarke et al. beschriebene Weg [28]. Dabei stellt man – ähnlich der Vorgehensweise bei der Berechnung der Diophantischen Gleichung für $E_j(z^{-1})$ und $F_j(z^{-1})$ – die Formel (6.48) für $G_{j+1}(z^{-1})$ und $G_j(z^{-1})$ auf und subtrahiert sie voneinander:

$$\begin{array}{l|l} + & G_{j+1}(z^{-1}) = G_{j+1}'(z^{-1})T(z^{-1}) + z^{-(j+1)}\Gamma_{j+1}(z^{-1}) \\ - & G_j(z^{-1}) = G_j'(z^{-1})T(z^{-1}) + z^{-j}\Gamma_j(z^{-1}) \\ \hline & e_j z^{-j} B(z^{-1}) = g_j' z^{-j} T(z^{-1}) + z^{-j} (z^{-1} \Gamma_{j+1}(z^{-1}) - \Gamma_j(z^{-1})) \end{array} \quad (6.58)$$

Aus dem Koeffizientenvergleich obiger Polynomgleichung erhält man die Berechnungsvorschriften:

$$g_j' = \frac{e_j b_0 + \gamma_{j,0}}{t_0} \quad (6.59)$$

$$\gamma_{j+1,i-1} = \gamma_{j,i} + e_j b_i - g_j' t_i \quad (6.60)$$

Da für das letzte Element $\gamma_{j+1,i}$ natürlich der Summand $\gamma_{j,i+1}$ nicht existiert, gilt in diesem Fall

$$\gamma_{j+1,i} = e_j b_{i+1} - g_j' t_{i+1} \quad (6.61)$$

Wie aus obiger Definition leicht einzusehen ist, läuft die Zählvariable i von 0 bis $\max(nb-1, nt-1)$. Da $T(z^{-1})$ in den meisten Fällen ein einfaches Tiefpassfilter darstellt, existieren nur die Elemente $t_0 = 1$ und t_1 . Üblicherweise kann daher davon ausgegangen werden, dass $nb \geq nt$ ist und folglich $i = 0 \dots (nb-1)$ beträgt. Die jeweils nicht existierenden Terme mit $b_k, k > nb$ oder $t_k, k > nt$ entfallen selbstverständlich.

Die Endbedingungen der Rekursion erhält man aus Formel (6.57) für $j = 1$, ebenfalls durch Koeffizientenvergleich:

$$g_0' = \frac{e_0 b_0}{t_0} \quad (6.62)$$

$$\gamma_{1,i} = e_0 b_{i+1} - g_0' t_{i+1} \quad (6.63)$$

Damit sind die Matrizen \mathbf{G}' und $\Gamma(z^{-1})$ berechenbar, wobei für die Berechnung der freien Antwort \mathbf{f} eigentlich nur $\Gamma(z^{-1})$ benötigt wird. Auch für die Berechnung der optimalen Steuergrößen gemäß Formel (6.51) ist \mathbf{G}' nicht komplett notwendig, da nur das erste Element von \mathbf{u} verwendet wird (s. Seite 101). Allerdings sind die g_j' -Werte für die Ermittlung der Elemente der $\Gamma(z^{-1})$ -Matrix erforderlich, so dass \mathbf{G}' trotzdem vollständig berechnet werden muss.

Mit Hilfe der oben dargestellten Rekursionen und Berechnungsformeln kann nun die freie Antwort \mathbf{f} bestimmt, in Gl. (6.52) eingesetzt und so die nächste an die Strecke anzulegende Stellgröße $u(t)$ berechnet werden.

A procedure described by Clarke et al. [28] for the determination of the elements of \mathbf{F} is suggested here. Similar to the approach for the calculation of the Diophantine equation for $E_j(z^{-1})$ and $F_j(z^{-1})$, only the equation (6.48) for $G_{j+1}(z^{-1})$ and $G_j(z^{-1})$ is formed and then $G_{j+1}(z^{-1})$ and $G_j(z^{-1})$ are subtracted from each other, see equation (6.58). The following expressions (6.59) and (6.60) can be obtained by comparing the coefficients of the above polynomial equation. As for the last element $\gamma_{j+1,i}$, the summand $\gamma_{j,i+1}$ does of course not exist, in this case, the following applies, see expression (6.61). As it can be easily seen from the above definition, the counter variable i varies from 0 to $\max(nb - 1, nt - 1)$. Since $T(z^{-1})$ in most cases represents a simple low-pass filter, only the elements $t_0 = 1$ and t_1 exist. Normally it can be assumed that $nb \geq nt$ and therefore $i = 0 \dots (nb - 1)$. As a matter of course, the non-existing terms with $b_k, k > nb$ or $t_k, k > nt$, respectively, are omitted.

The final terms of the recursion can also be obtained by coefficient comparison from equation (6.57) for $j = 1$, see expressions (6.62) and (6.63).

Thus, the matrices \mathbf{G}' and $\Gamma(z^{-1})$ can be calculated, whereas only $\Gamma(z^{-1})$ is required for the calculation of the free response \mathbf{f}' . For the calculation of optimum values for the actuating variables according to equation (6.51), \mathbf{G}' is also not completely necessary, since only the first element of \mathbf{u} is used (see page 101). However, the values g'_j are required for determining the elements of the matrix $\Gamma(z^{-1})$ so that in any case \mathbf{G}' must be completely calculated.

With the help of the recursions and equations shown above, the free response \mathbf{f}' can be determined, be substituted into equation (6.52) and then the next value of the actuating variable $u(t)$ that should be applied to the plant can be calculated.

为了确定矩阵 \mathbf{F} 中的元素, 建议使用学者 Clarke 等所描述的方法 [28]。类似于计算丢番图方程中 $E_j(z^{-1})$ 和 $F_j(z^{-1})$ 的处理方式, 首先列出 j 和 $j + 1$ 时刻的丢番图方程, 然后对其进行求差运算, 具体过程见方程 (6.58)。

基于上述多项式方程的系数对比可以得到具体计算规则, 见关系式 (6.59) 和 (6.60)。对于最后一个元素 $\gamma_{j+1,i}$ 来说, 显然 $\gamma_{j,i+1}$ 是不存在的, 从而有表达式 (6.61)。

由上述定义不难看出, i 的变化范围为 0 至 $\max(nb - 1, nt - 1)$ 。由于在大多数情况下, $T(z^{-1})$ 表示一个简单的低通滤波器, 所以仅存在元素 $t_0 = 1$ 和 t_1 。因此, 通常可以认为 $nb \geq nt$, 从而有 $i = 0 \dots (nb - 1)$ 。显然, 含有 $b_k, k > nb$ 或 $t_k, k > nt$ 的相关项是不存在的, 可直接消去。

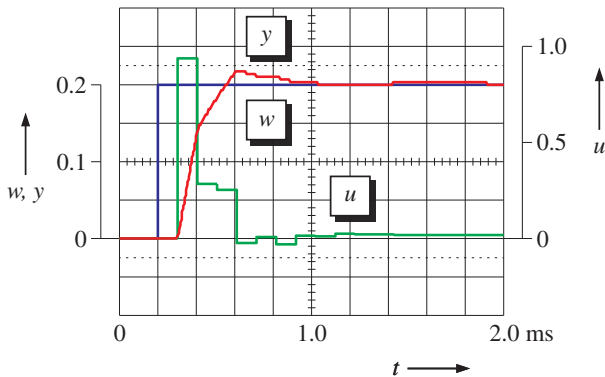
递归的终止条件可由 $j = 1$ 时的方程 (6.57) 求得, 经过系数对比后可得 (6.62) 和 (6.63)。因此, 矩阵 \mathbf{G}' 和 $\Gamma(z^{-1})$ 是可计算的, 其中也仅需要 $\Gamma(z^{-1})$ 以用于计算自由响应 \mathbf{f}' 。此外, 基于方程 (6.51) 可知, 最优控制变量的计算并不需要完整的矩阵 \mathbf{G}' , 而只是需要 \mathbf{u} 的第一个元素, 见第 101 页。显然, 需要使用 g'_j 的值以计算矩阵 $\Gamma(z^{-1})$ 中的元素, 从而必须计算出完整的矩阵 \mathbf{G}' 。

根据上面描述的递归和计算方程即可确定自由响应 \mathbf{f}' , 然后将其代入至方程 (6.52), 即可计算出下一周期作用于被控系统的控制变量 $u(t)$ 。

6.2.2 Simulationen/Simulations/仿真

Das typische Verhalten eines GPC-Reglers mit Filterpolynom $T(z^{-1})$ wird durch die folgenden Simulationsergebnisse verdeutlicht. Es kann gezeigt werden, dass die Tiefpassfilterung der abgetasteten Regelgröße ohne einen Einfluss auf die Dynamik des Gesamtsystems erfolgt.

In Abb. 6.8 ist die Filterung durch Setzen von $T(z^{-1}) = 1$ abgeschaltet. Da auf den gemessenen Istwert keine Störung einwirkt, zeigt sich die Regelung mit einem optimalen Verhalten. Der Sollwert wird in drei Schritten erreicht.



6.8: Simulation Führungsverhalten ohne Störung des Messsignals

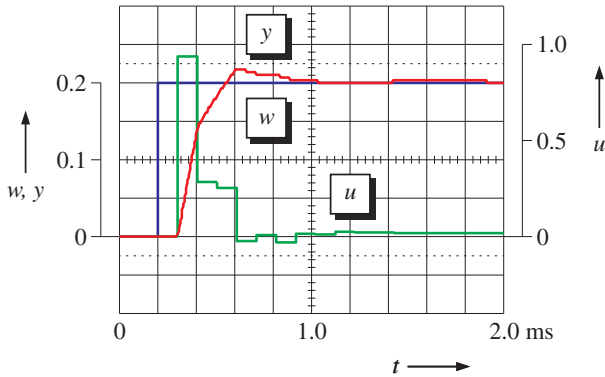
Wird der Messwert mit einem rechteckförmigen Störsignal mit der halben Frequenz der Abtastrate verfälscht, so ergeben sich die in Abb. 6.10(a) dargestellten Signalverläufe. Obwohl die Amplitude des Störsignals nur 0,005 beträgt, schwingt das Ausgangssignal des Reglers stark; die Amplitude von $u(t)$ liegt fast bei 0,2. Der Istwert $y(t)$ der geregelten Strecke wird zwar durch die Streckenzeitkonstanten bedämpft; trotzdem sorgen die gewaltigen, periodischen Sprünge der Stellgröße zwischen +0,2 und -0,2 auch dort für ein unruhiges Verhalten.

Setzt man nun die internen Filterfähigkeiten eines GPC-Reglers ein, hier durch Wahl des Filterpolynoms $T(z^{-1}) = 1 - 0,95z^{-1}$, so verbessert sich das Verhalten der Regelung deutlich (Abb. 6.10(b)). Nach wie vor wird auf den gemessenen Istwert das gleiche rechteckförmige Störsignal addiert. Bedingt durch das Filterverhalten beträgt die Schwingungsamplitude der Stellgröße jetzt nur noch etwa 0,02, was einer Verbesserung um den Faktor 10 entspricht. Entsprechend zeigt der Istwert annähernd keine überlagerten Schwingungen mehr. Trotzdem hat sich die Sprungantwort des Gesamtsystems nur unwesentlich verändert; der Sollwert wird nach wie vor in drei Schritten erreicht.

6.2.2 Simulations

The typical behavior of a GPC controller with filter polynomial $T(z^{-1})$ is verified by the following simulation results. It can be shown that low-pass filtering of the sampled actuating variable can be done without influence on the dynamics of the overall system.

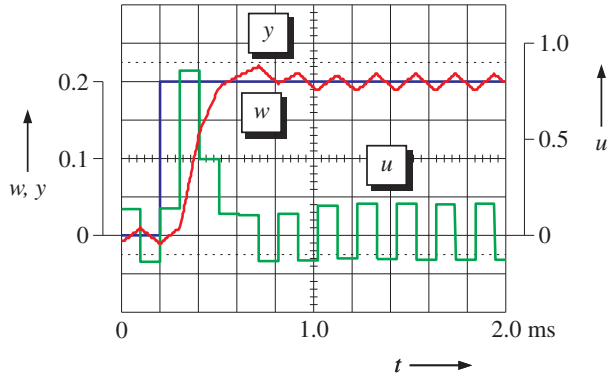
In Figure 6.9, the filtering is turned off by setting $T(z^{-1}) = 1$. Since the measured values of the controlled variables are not effected by any disturbances, the control shows ideal behavior. The reference value is reached in three sampling steps.



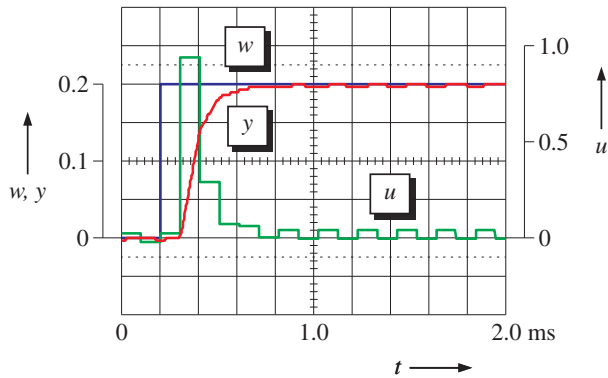
6.9: Simulation: Reference action without noise disturbances

If the measuring signal is disturbed with a rectangle-shaped noise signal with half the sampling frequency, the signals shown in Figure 6.10(a) can be obtained. Although the amplitude of the noise signal is only 0.005, the controller output signal shows strong oscillating behavior; the amplitude of $u(t)$ is approximately 0.2. The actual value of the controlled variable $y(t)$ of the controlled plant is indeed damped by the plant time constants; despite this fact, the huge periodical steps of the actuating variable between +0.2 and -0.2 are causal for the jerky behavior of the system.

If the internal filter features of a GPC controller are enabled now by choosing the filter polynomial $T(z^{-1}) = 1 - 0.95z^{-1}$, the behavior of the controller can be significantly improved (Figure 6.10(b)). Still the same rectangular-shaped disturbance signal is added to the measured values of the controlled variable. Because of the enabled filter, the amplitude of the oscillations of the actuating variable is now only about 0.02, which corresponds to an improvement of the factor 10. Accordingly, the controlled variable shows nearly no more superimposed oscillations. Despite this, the step response of the entire system has not significantly changed; the reference value is still reached in three sampling steps.



(a)



(b)

6.10: Simulation Führungsverhalten mit Störung des Messsignals: (a) ohne Filterung, (b) mit Filterung;

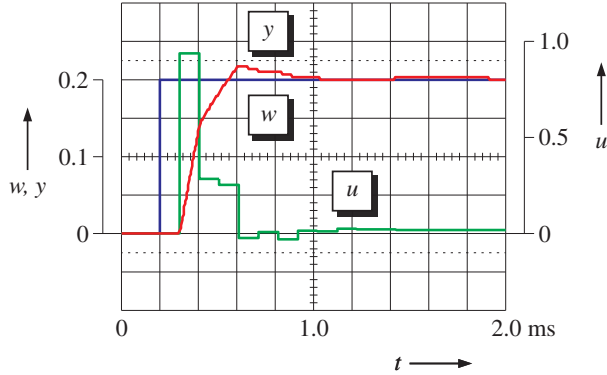
Simulation: Reference action with noise disturbances: (a) without Filter, (b) mit Filter;

测量信号含有扰动时的系统响应仿真: (a) 无滤波, (b) 含滤波

6.2.2 仿真

带有滤波多项式 $T(z^{-1})$ 的 GPC 控制器的典型特性可由随后给出的仿真结果所体现。实际表明，离散化的被控变量低通滤波对于整个系统的动态特性并不产生影响。

图 6.11 给出的是通过设置 $T(z^{-1}) = 1$ 的仿真结果，此时也意味着不存在滤波。由于测得的被控变量不受扰动的影响，从而可以体现出最优的控制特性，也即在三个周期内可达到参比变量。



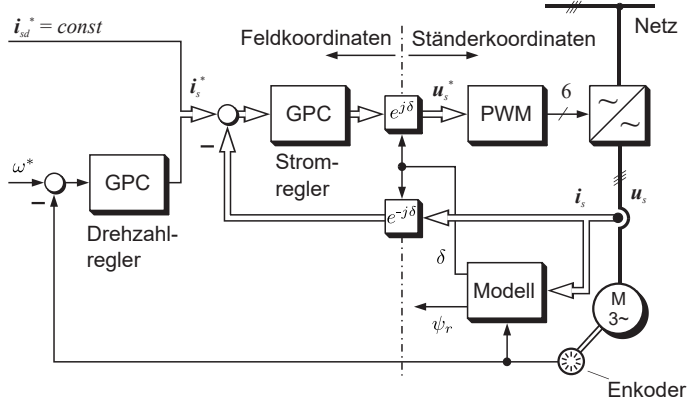
6.11: 测量信号无扰动时的系统响应仿真

如果当测量值受到一个具有二分之一采样频率的矩形波的扰动而失真，图 6.10(a) 中给出了不包含滤波的系统响应结果。可以看出，即使扰动信号的幅值仅为 0.005，控制器的输出信号也会发生强烈震荡，控制变量 $u(t)$ 的幅值已接近 0.2。虽说，被控系统的被控变量 $y(t)$ 的实际值会由于系统时间常数而减弱，然而，在 +0.2 和 -0.2 之间强烈的、周期性跳变的控制变量会引起系统不稳定。

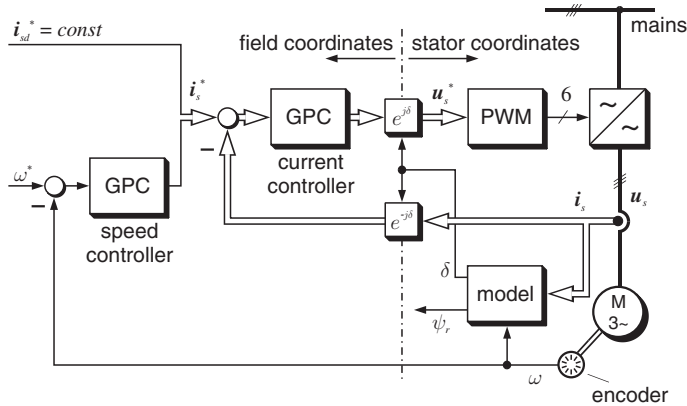
如果此时使用包含内部滤波特性的 GPC 控制器，且选择滤波设计多项式的值为 $T(z^{-1}) = 1 - 0.95z^{-1}$ ，控制效果可得到明显改善，如图 6.10(b) 所示。作为对比，测量值包含一个与前述相同的矩形扰动信号，可以看出，带滤波后的控制变量的振荡幅值约为 0.02，相当于效果改善了 10 倍，同时被控变量也可近似看作无振荡现象。尽管如此，系统动态阶跃响应并未发生明显变化，它仍然需要大约三个周期以达到参比变量。

6.3 Kaskadenregelung mit GPC-Reglern/Cascade control with GPC controllers/基于 GPC 控制器的串级控制

Als erster Ansatz werden die PI-Regler einer herkömmlichen feldorientierten Regelung in Kaskadenstruktur durch ebenfalls in Kaskade geschaltete GPC-Regler ersetzt. Der Signalflussplan der Regelung ist in Abb. 6.12 wiedergegeben. Wie zu erkennen ist, ist auch hier kein echter Flussregler implementiert, sondern der Fluss wird durch Vorgabe eines gleichbleibenden feldbildenden Statorstromanteils i_{sd}^* annähernd konstant gehalten.



6.12: Feldorientierte Antriebsregelung mit GPC-Reglern



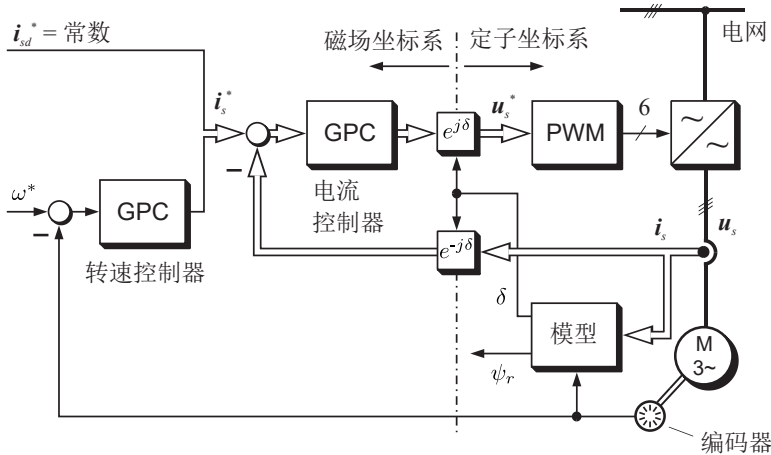
6.13: Field-oriented drive control with GPC controllers

6.3 Cascade control with GPC controllers

As a first approach the PI-controllers of a conventional field-oriented control in cascaded structure are replaced by GPC controllers that are also cascaded. The signal flow graph of the control is shown in Figure 6.13. As it can be seen here, no real flux controller is implemented, but the flux is kept approximately constant because of a constant reference value for the field-producing stator current component i_{sd} .

6.3 基于 GPC 控制器的串级控制

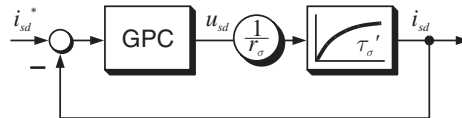
对于传统的磁场定向控制来说，串级结构中的 PI 控制器可直接由 GPC 控制器所替代，如此即形成一种新的基于 GPC 控制器的串级控制，其信号流程图如图 6.14 所示。可以看出，此时也并未实现一个真正的磁通控制器，磁通是由一个给定的励磁电流分量 i_{sd} 参考值所确定，如此磁通也可近似保持恒定。



6.14: 基于 GPC 控制器的磁场定向驱动控制

6.3.1 Stromregelung/Current control/电流控制

Der Stromregelung mit GPC-Regler liegt vom Prinzip her das gleiche Streckenmodell zugrunde wie der konventionellen Regelung (Abb. 3.5 auf Seite 21). Allerdings bietet der zeitdiskrete Ansatz von GPC die Möglichkeit, Totzeiten mit einem Wert von $n \cdot T_0$ direkt im Modell durch Verschiebungen der Koeffizienten des Polynoms $B(z^{-1})$ um z^{-n} zu berücksichtigen. Verwendet man ein CARIMA-Modell gemäß Formel (6.2) oder (6.3), so ist eine Totzeit von $1 \cdot T_0$ bereits durch den Ansatz mit $u(t-1)$ anstelle von $u(t)$ im Modell implementiert. Damit geht die Umrichtertotzeit in die Bestimmung der Parameter des CARIMA-Modells nicht mehr ein, da sie bereits im Modellansatz abgebildet worden ist. Der resultierende, vereinfachte Signalflussplan des Stromregelkreises ist in Abb. 6.15 abgebildet.



6.15: Stromregelkreis mit GPC-Regler/Current control loop with GPC controller/帶有 GPC 控制器的电流控制闭环

Unter Anwendung der üblichen Umrechnungstabellen, z. B. aus [58], erhält man die Z-Übertragungsfunktion der Regelstrecke zu

$$G(z^{-1}) = \frac{1}{r_\sigma} \left(1 - e^{-\frac{\tau_0}{\tau_\sigma T}} \right) z^{-1} = \frac{B(z^{-1})}{A(z^{-1})} \quad (6.64)$$

Da für die Prädiktion eigentlich nie mit $A(z^{-1})$, sondern stets mit $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$ gerechnet wird, berechnet man gleich die einzelnen Koeffizienten von $\tilde{A}(z^{-1})$. Für $\tilde{A}(z^{-1})$ und $B(z^{-1})$ gilt:

$$\begin{aligned} \tilde{a}_0 &= 1 & b_0 &= 0 \\ \tilde{a}_1 &= - \left(1 + e^{-\frac{\tau_0}{\tau_\sigma T}} \right) & b_1 &= \frac{1}{r_\sigma} \left(1 - e^{-\frac{\tau_0}{\tau_\sigma T}} \right) \\ \tilde{a}_2 &= e^{-\frac{\tau_0}{\tau_\sigma T}} \end{aligned} \quad (6.65)$$

6.3.1 Current control

Current control with a GPC controller is, in principle, based on the same plant model as the conventional control (Figure 3.5 on page 21). Indeed, the time-discrete approach of GPC offers the possibility of considering dead times with a value $n \cdot T_0$ directly in the model by multiplying the coefficients of the polynomial $B(z^{-1})$ with z^{-n} . If a CARIMA model according to equation (6.2) or (6.3) is used, a dead time of $1 \cdot T_0$ is already implemented in the model because of the approach with $u(t-1)$ instead of $u(t)$. Therefore, the inverter dead time does not have to be considered for the determination of the CARIMA model parameters, since it has already been defined in the model itself. The resulting simplified signal flow graph of the current control loop is shown in Figure 6.15.

Using the standard conversion tables, e.g. from [58], the Z-transfer function of the plant can be obtained:

$$G(z^{-1}) = \frac{\frac{1}{r_\sigma} \left(1 - e^{-\frac{\tau_0}{\tau\sigma'}}\right) z^{-1}}{1 - e^{-\frac{\tau_0}{\tau\sigma'}} z^{-1}} = \frac{B(z^{-1})}{A(z^{-1})} \quad (6.64)$$

As the prediction is normally not done with $A(z^{-1})$, but with $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$, the individual coefficients of $\tilde{A}(z^{-1})$ are calculated. Thus, for $\tilde{A}(z^{-1})$ and $B(z^{-1})$ the following coefficients can be obtained, see equations (6.65).

6.3.1 电流控制

基于 GPC 控制器的电流控制与基于 PI 控制器的电流控制所控制的实际上是相同的被控系统，可对比第 21 页图 3.5。显然，时间离散的 GPC 方法可通过将多项式 $B(z^{-1})$ 的系数延迟 z^{-n} ，从而直接在模型中考虑到一个数值为 $n \cdot T_0$ 的死区时间。如果使用基于方程 (6.2) 或 (6.3) 的 CARIMA 模型，那么数值为 $1 \cdot T_0$ 的死区时间则可通过使用 $u(t-1)$ 代替 $u(t)$ 的方式在模型中实现。从而，变流器的死区时间不再影响 CARIMA 模型参数的确定，因为这一死区时间已包含在模型中。图 6.15 给出了最终的、简化的带有 GPC 控制器的电流控制闭环信号流图。

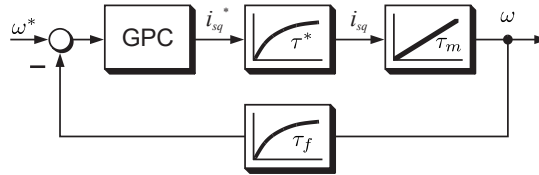
使用常见的 Z 变换转换表，例如文献 [58] 中的附录部分，即可得到被控系统的 Z 传递函数：

$$G(z^{-1}) = \frac{\frac{1}{r_\sigma} \left(1 - e^{-\frac{\tau_0}{\tau\sigma'}}\right) z^{-1}}{1 - e^{-\frac{\tau_0}{\tau\sigma'}} z^{-1}} = \frac{B(z^{-1})}{A(z^{-1})} \quad (6.64)$$

由于完成预测实际无需计算 $A(z^{-1})$ ，而始终需要计算的是 $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$ ，所以 $\tilde{A}(z^{-1})$ 的所有系数也是需要计算出来的。对于 $\tilde{A}(z^{-1})$ 和 $B(z^{-1})$ 来说，其系数满足关系式 (6.65)。

6.3.2 Drehzahlregelung/Speed control/转速控制

Die Vorgehensweise bei der Festlegung des GPC-Reglers für den Drehzahlregelkreis ist vom Prinzip genauso, wie bei der konventionellen PI-Kaskadenregelung in Kap. 3.2 auf Seite 23 beschrieben. Der innere Drehzahlregelkreis wird durch ein PT_1 -Glied mit der Zeitkonstanten τ^* ersetzt. Abb. 6.16 zeigt den Signalflussplan der GPC-Drehzahlregelung. Da die inneren Filtereigenschaften von GPC ausgenutzt werden, kann das zusätzliche Tiefpassfilter im Rückführungszweig deutlich kleiner ausfallen als bei der PI-Regelung.



6.16: Drehzahlregelkreis mit GPC-Regler/Speed control loop with GPC controller/带有 GPC 控制器的转速控制闭环

Die Zeitkonstanten der beiden PT_1 -Glieder sind auch in der Summe noch sehr viel kleiner als die mechanische Zeitkonstante τ_m , daher werden sie für die Auslegung des GPC-Reglers nicht berücksichtigt. Durch simulative Untersuchungen kann gezeigt werden, dass die Integration auch der beiden PT_1 -Glieder in das Streckenmodell lediglich die Ordnung des Systems und damit die Berechnungsmatrizen des GPC-Reglers vergrößert, aber keinen Performancegewinn erzielt. Daher kann die Modellbildung auf den reinen Integrator beschränkt werden; die zeitdiskrete Übertragungsfunktion $G(z^{-1})$ wird zu:

$$G(z^{-1}) = \frac{\tau_0}{\tau_m} z^{-1} \frac{1}{1 - z^{-1}} \quad (6.66)$$

Die Koeffizienten der Polynome $\tilde{A}(z^{-1})$ und $B(z^{-1})$ lauten in diesem Fall:

$$\begin{aligned} \tilde{a}_0 &= 1 & b_0 &= 0 \\ \tilde{a}_1 &= -2 & b_1 &= \frac{\tau_0}{\tau_m} \\ \tilde{a}_2 &= 1 & & \end{aligned} \quad (6.67)$$

6.3.2 Speed control

The approach for designing the GPC controller for the speed control loop is in principle the same as described for conventional PI cascade control in Chapter 3.2 on page 23. The inner current control loop is replaced with a first order transfer function (PT₁-block) with the time constant τ^* . Figure 6.16 shows the signal flow graph of the GPC speed control loop. As the internal filter characteristics of GPC are used, the additional low-pass filter in the feedback path can be significantly smaller than a corresponding one for PI control.

Even the sum of the time constants of both PT₁-blocks is much smaller than the mechanical time constant τ_m , hence, they are not considered in the design of the GPC controller. It can be shown by simulation that including both first order transfer function blocks into the plant model merely increases the order of the system and because of this also the calculation matrices of the GPC controller, but the control performance cannot be improved. Therefore, the model design can be limited to a pure integrator; the time-discrete transfer function then becomes to

$$G(z^{-1}) = \frac{\frac{\tau_0}{\tau_m} z^{-1}}{1 - z^{-1}} \quad (6.66)$$

In this case, the coefficients of the polynomials $\tilde{A}(z^{-1})$ and $B(z^{-1})$ are equations (6.67).

6.3.2 转速控制

从原理上讲, 基于 GPC 控制器的转速控制处理方式与传统的基于 PI 控制器的串级控制完全一致, 可见前述第 23 页章节 3.2。内层的电流控制闭环可等效为时间常数为 τ^* 的一阶滞后元件, 图 6.16 给出了带有 GPC 控制器的转速控制闭环信号流图。由于 GPC 控制器的内部滤波特性得到了充分利用, 从而反馈通道上的低通滤波器的时间常数要明显小于基于 PI 控制器时的情形。

即使是图中所示的两个一阶滞后元件的时间常数之和, 它也远小于机械时间常数 τ_m , 从而在设计 GPC 控制器时可以忽略不计。仿真分析表明, 被控系统模型的两个一阶滞后元件的积分部分增加了系统的阶数, 从而 GPC 控制器的计算矩阵也相应变得复杂, 但却不能够获得更优的控制效果。因此, 可以仅使用纯积分器来构造模型, 时间离散的传递函数 $G(z^{-1})$ 则变为:

$$G(z^{-1}) = \frac{\frac{\tau_0}{\tau_m} z^{-1}}{1 - z^{-1}} \quad (6.66)$$

多项式 $\tilde{A}(z^{-1})$ 和 $B(z^{-1})$ 的系数满足关系式 (6.67)。

6.3.3 Experimentelle Ergebnisse/Experimental results/实验结果

Stromregelung

Zur Bewertung des Verhaltens des GPC-Reglers ist der Vergleich mit einem PI-Regler sinnvoll. Dazu wurde eine Stromregelung einer Asynchronmaschine als konventioneller PI- und als modellbasierter Prädiktivregler implementiert. Der PI-Regler ist wie in Kap. 3.1 auf Seite 21 ff. beschrieben; zunächst nach dem symmetrischen Optimum ausgelegt und dann von Hand weiter optimiert worden. Als bestmögliche Werte haben sich $T_i = 0,33$ und $V_i = 2,3$ herausgestellt. Die Parameter des GPC-Reglers können Tabelle 6.1 entnommen werden.

N_p	N_u	na	nb	nt	\tilde{a}_0	\tilde{a}_1	\tilde{a}_2	b_0	b_1	t_0	t_1	λ
4	2	2	1	1	1,0	-1,9947	0,9947	0,0	0,1650	1,0	-0,95	0,003

6.1: Einstellungen für GPC-Stromregler/Parameter settings for the GPC current controller/GPC 电流控制器的参数设置

Abb. 6.17 zeigt das Großsignalverhalten des geschlossenen Regelkreises; dabei stellt Diagramm 6.17(a) das Verhalten eines PI-geregelten Antriebs dar, während die Messergebnisse in Diagramm 6.17(b) unter Verwendung eines GPC-Reglers ermittelt wurden. In beiden Fällen wurde der drehmomentbildende Stromanteil i_{sq} mit einem Sollwertsprung von $i_{sq}^* = 0$ auf $i_{sq}^* = 0,4$ beaufschlagt. Die Ergebnisse zeigen, dass der GPC-Regler einen geringeren Überschwinger als der PI-Regler aufweist. Die Anstiegszeit ist in beiden Fällen annähernd identisch, da sie im Wesentlichen durch die zur Verfügung stehende Stelleistung begrenzt wird. Es zeigt sich, dass die Verwendung eines modellbasierten Prädiktivreglers in diesem Fall keinen signifikanten Vorteil bietet.

Das Kleinsignalverhalten der beiden Reglertypen kann Abb. 6.18 entnommen werden. Wie der Vergleich des in Abb. 6.18(a) dargestellten Istwertverlaufs des Stromregelkreises mit PI-Regler mit dem in Abb. 6.18(b) angegebenen Verhalten des GPC-Reglers offenlegt, nutzt der PI-Regler wegen des kleinen Sollwertsprungs von nur $\Delta i_{sq}^* = 0,1$ die zur Verfügung stehende Stellreserve bei weitem nicht aus, während der modellbasierte Prädiktivregler bedingt durch seine vollkommen andere Charakteristik die zur Verfügung stehende maximale Stellgröße bedeutend besser ausnutzt. Daher zeigt sich der GPC-Regler im Kleinsignalverhalten dem PI-Regler überlegen.

6.3.3 Experimental results

Current control

To evaluate the behavior of the GPC controller, it is meaningful to compare it with a PI-controller. Because of this, current control of an asynchronous machine was implemented with a conventional PI- and with a model-based predictive controller. The PI-controller

was designed as described in Chapter 3.1 on page 21 et seqq. according to the symmetrical optimum method and then it was further optimized empirically. The best possible values that could be obtained are $T_i = 0.33$ and $V_i = 2.3$. The parameters of the GPC controller can be taken from Table 6.1.

Figure 6.17 shows the large-signal behavior of the closed control loop; Figure 6.17(a) shows the behavior of a PI controlled drive, while the results in Figure 6.17(b) were obtained with a GPC controller. In both cases, the torque producing current component i_{sq} was step-changed from $i_{sq}^* = 0$ to is $i_{sq}^* = 0.4$. The results show that the GPC controller generates a smaller overshoot than the PI-controller. In both cases, the rise time is approximately identical, because it is essentially limited by the available power of the actuator. It can be seen that in this case no significant advantage is offered by the use of a model-based predictive controller.

The small-signal behavior of both controllers can be seen in Figure 6.18. Comparing the actual values of the output (controlled) variable of the current control loop with the PI-controller shown in Figure 6.18(a) to the behavior of the GPC controller shown in Figure 6.18(b), it can be seen that the PI-controller does not make use of the full available actuator power because of the small step-change of the reference value of only $\Delta i_{sq}^* = 0.1$, while the model-based predictive controller uses the available actuator power in an optimal way due to its totally different characteristics. Hence, the GPC controller shows up superior performance in the small-signal behavior compared to the PI-controller.

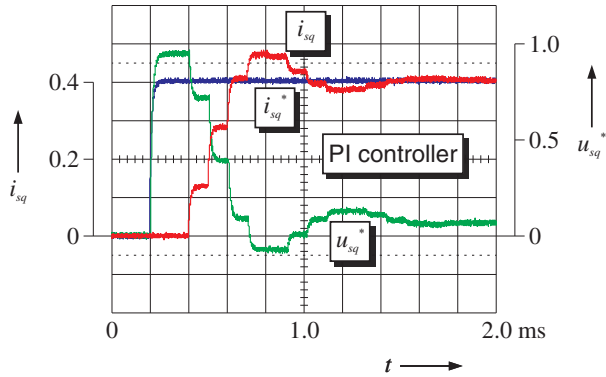
6.3.3 实验结果

电流控制

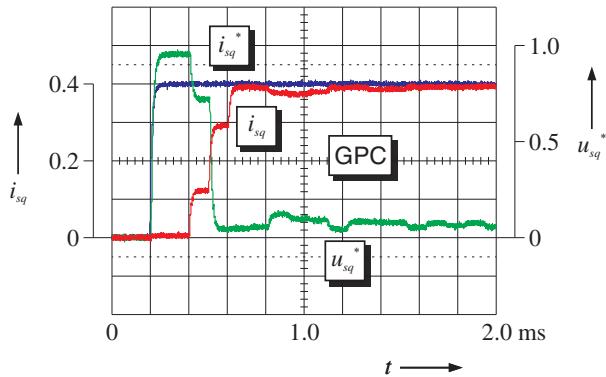
通过与 PI 控制器进行对比来评价 GPC 控制器的性能是有意义的。为此，本部分使用传统的 PI 控制器和基于模型的预测控制器来分别实现异步电机的电流控制。首先，根据对称最优方式设计 PI 控制器，如章节 3.1 第 21 页所述；然后基于手动计算对其进行进一步优化，所得到的最优值为 $T_i = 0.33$ 和 $V_i = 2.3$ 。GPC 控制器的参数设置见表 6.1。

图 6.17 给出了电流闭环控制的大信号响应特性：图 6.17(a) 为基于 PI 控制器的情形；图 6.17(b) 为基于 GPC 控制器所得的结果。上述两种情况下，转矩电流分量 i_{sq} 的参考值均由 $i_{sq}^* = 0$ 至 $i_{sq}^* = 0.4$ 发生阶跃变化。结果表明：相对于 PI 控制器，基于 GPC 控制器的情形表现出更小的超调量；与此同时，二者的上升时间几乎一致，原因是二者均使用了系统所能提供的最大控制变量。上述分析表明，在大信号阶跃情况下，使用基于模型的预测控制器无法体现其优势。

图 6.18 给出了电流闭环控制的小信号响应特性。对比图 6.18(a) 所示的使用 PI 控制器的电流闭环控制所得的被控变量 i_{sq} 变化曲线与图 6.18(b) 所示的使用 GPC 控制器的 i_{sq} 变化曲线可知：PI 控制器在转矩电流分量 i_{sq} 的参考值较小的阶跃变化时，如 $\Delta i_{sq}^* = 0.1$ ，此时它无法充分利用变频器所提供的电压值；而基于模型的控制器的则可以充分利用变频器所能提供的最大控制变量。基于上述分析，GPC 控制器的小信号响应特性要优于 PI 控制器。



(a)



(b)

6.17: Stromregelung Großsignalverhalten: (a) PI-Regler, (b) GPC-Regler;

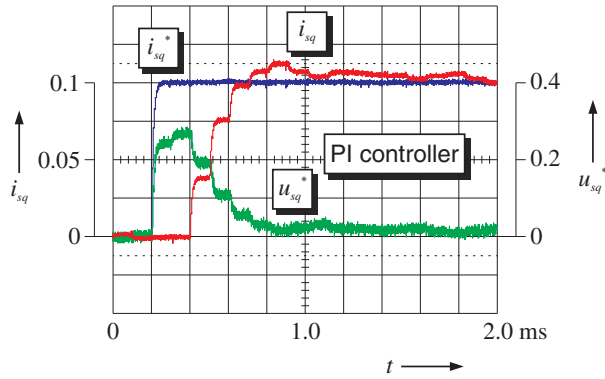
Current control: Large-signal behavior: (a) PI controller, (b) GPC controller;

电流控制的大信号响应: (a) PI 控制器, (b) GPC 控制器

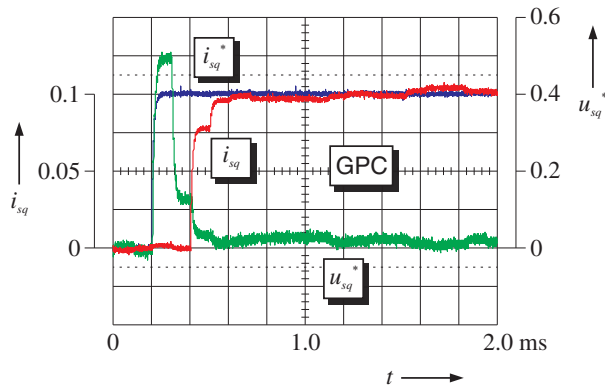
Ein MPC-Regler mit übertragungsfunktionsbasiertem Streckenmodell berücksichtigt im Gegensatz zum PI-Regler nicht nur die momentanen Soll- und Istwerte, sondern zieht auch die Vergangenheit zur Berechnung optimaler Stellgrößen in Betracht. Treten nun in einem Regelkreis konstante Störgrößen auf, so „kennt“ ein modellbasierter Prädiktorregler diese quasi, da die Vergangenheit ebenfalls in die Vorausberechnungen einfließt. Bei einer Antriebsregelung tritt dieser Fall typischerweise in Form der von der Geschwindigkeit abhängigen Gegenspannung des Motors auf, die sich im Stromregelkreis als Störgröße bemerkbar macht. Wegen der im Drehzahlregelkreis weitaus größeren Zeitkonstanten kann der Wert dieser Störung für die Stromregelung als konstant betrachtet werden.

In contrast to a PI-controller, an MPC controller with a transfer function based plant model considers not only present values of the actuating and of the controlled variables, but also considers the past for the calculation of optimum values for the actuating variables. If constant disturbances appear in a control loop, a model-based predictive controller knows them since the past is also considered in the precalculations. In drive control, this case appears typically in the form of the back EMF, which is proportional to the drive speed that acts like a disturbance in the current control loop. Because of the much larger time constants of the speed control loop, the value of this disturbance can be assumed to be constant for the current control.

不同于 PI 控制器，基于被控系统传递函数的 MPC 控制器不仅可以考虑到系统当前的被控变量和参比变量，它还能够考虑到系统过去的状态，并用于计算最优的控制变量。如果在控制闭环中含有恒定的扰动变量，那么，基于模型的预测控制器似乎知晓这一扰动变量，因为在预测时也同时包含了过去的信息。在驱动控制领域，与转速相关的电机反向电动势作为电流控制闭环的扰动变量是无法忽略的。此外，由于转速控制闭环具有相对较大的时间常数，所以对于电流控制来说，可认为上述扰动变量恒定不变。



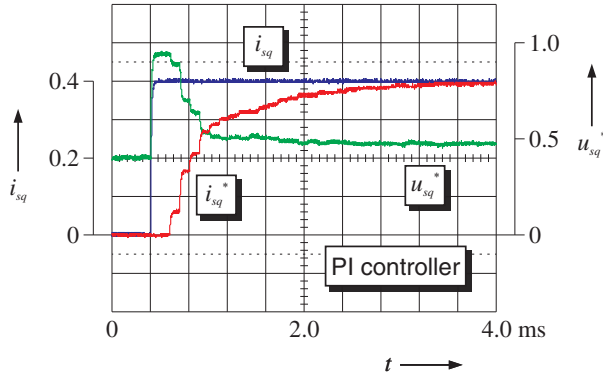
(a)



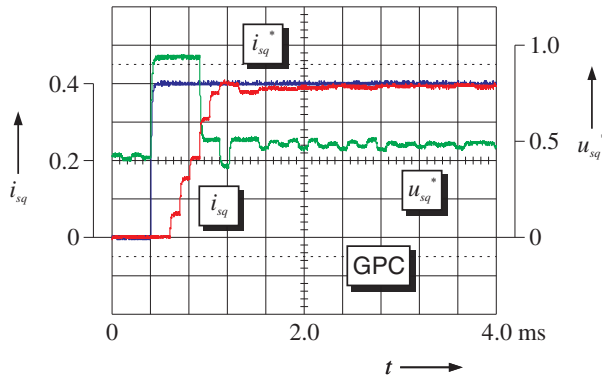
(b)

6.18: Stromregelung Kleinsignalverhalten: (a) PI-Regler, (b) GPC-Regler;
 Current control: Small-signal behavior: (a) PI-Regler, (b) GPC-Regler;
 电流控制的小信号响应: (a) PI 控制器, (b) GPC 控制器

Notizen/Notes/笔记



(a)



(b)

6.19: Stromregelung: Großsignalverhalten bei $\omega = 0,4$ /Current control: Large-signal behavior at $\omega = 0.4$ /电流控制: 转速 $\omega = 0.4$ 下的大信号响应

Die in Abb. 6.19 dargestellten Messungen wurden bei einer normierten Drehgeschwindigkeit der Maschine von $\omega = 0,4$ aufgenommen. Wie Abb. 6.19(a) zeigt, verlängert sich bei einem einfachen PI-Stromregler die Anstiegszeit wegen der durch die Rotation der Maschine hervorgerufene Gegenspannung enorm. Der GPC-Regler hingegen (Abb. 6.19(b)) zeigt deutlich besseres Verhalten, da ihm der störende Einfluss der Gegenspannung aus der Vergangenheit bekannt ist. Da eine Maschine aber selten im Stillstand betrieben wird und die Gegenspannung folglich fast immer vorhanden ist, zeigt der MPC-Regler auch im Großsignalverhalten bessere Ergebnisse als der PI-Regler.

The measurements shown in Figure 6.19 were taken with a normalized rotational speed $\omega = 0.4$ of the machine. As shown in Figure 6.19(a), using a simple PI current controller, the rise time is significantly longer because of the back EMF generated by the rotation of the machine. However, the GPC controller (Figure 6.19(b)) shows much better behavior, since the disturbing influence of the back EMF is known to the GPC controller from the past. Hence, as a machine is rarely operated in standstill mode and thus, the back EMF does nearly always exist, the MPC controller shows better results than the PI-controller in large-signal behavior, too.

图 6.19 所示的测量结果是在电机转速标么值为 $\omega = 0.4$ 的条件下得到的。如图 6.19(a) 所示，在 PI 控制器的情况下，由于电机旋转所引起的反向电动势的值太大，从而导致上升时间增加；而在 GPC 控制的情况下，由于控制器已知晓过去的电机反向电动势所带来的扰动影响，从而其控制效果得到了改善，如图 6.19(b) 所示。实际上，电机很少在静止状态下运行，所以反向电动势几乎是始终存在的。从上述角度来看，MPC 控制器的大信号响应特性也优于 PI 控制器。

Drehzahlregelung

Wie bei der Stromregelung geschehen, soll auch die GPC-Drehzahlregelung mit einer mit PI-Regler ausgestatteten Drehzahlregelung verglichen werden. Für die konventionelle Ausführung wurden zu diesem Zweck die gemäß Kap. 3.2 auf Seite 23 hergeleiteten Anfangseinstellungen experimentell überprüft und optimiert. Werte von $T_\omega = 64,0$ und $V_\omega = 17,2$ sind schließlich als bestmögliche Reglerparameter übernommen worden.

Die Dynamik im Drehzahlregelkreis ist bekanntlich bedeutend niedriger als diejenige des Stromregelkreises. Aus programmtechnischen Gründen müssen jedoch sowohl Strom- als auch Drehzahlregler mit derselben Abtastfrequenz betrieben werden. Nun sollte der Prädiktionshorizont eines MPC-Reglers in etwa den Zeitkonstanten des zu regelnden Systems entsprechen. Folglich ist für einen GPC-Drehzahlregler eine relativ hohe Prädiktionstiefe von $N_p = 200$ erforderlich. Da jedoch der Regelhorizont lediglich $N_u = 1$ beträgt, hält sich der mathematische Gesamtaufwand in Grenzen. Ein höherer Wert für N_u brachte keine Verbesserung des Regelverhaltens, so dass ein größerer Regelhorizont offensichtlich keinen Sinn macht. Die übrigen optimalen Parameter des GPC-Reglers sind in Tabelle 6.2 angegeben.

N_p	N_u	na	nb	nt	\tilde{a}_0	\tilde{a}_1	\tilde{a}_2	b_0	b_1	t_0	t_1	λ
200	1	2	1	1	1,0	-2,0	1,0	0,0	0,00013	1,0	-0,999	0,1

6.2: Einstellungen für GPC-Drehzahlregler

Das Großsignalverhalten eines Drehzahlregelkreises gibt keinen oder nur sehr geringen Aufschluss über die Qualität des Reglers, da die Dynamik in diesem Fall fast ausschließlich durch die zur Verfügung stehende Umrichterleistung begrenzt wird. Daher wird für den Vergleich zwischen PI- und GPC-Drehzahlregler nur das Kleinsignalverhalten betrachtet. Abb. 6.20 zeigt die entsprechenden Ergebnisse.

In der Antriebstechnik besteht oft das Problem, dass das gemessene Drehzahlsignal mit einem Tiefpassfilter gefiltert werden muss, da es ansonsten für eine Regelung nicht brauchbar ist. Diese Schwierigkeit tritt besonders bei der Verwendung von Inkrementalgebern mit niedrigen Strichzahlen auf, da das aus der Zählung der Striche pro Abtastzyklus gewonnene Drehzahlsignal springt. Bei Drehzahlgebern auf Basis eines Tachogenerators entstehen die oben beschriebenen Probleme selbstverständlich nicht; Tachogeneratoren sind aber teurer als Inkrementalgeber, müssen regelmäßig gewartet werden und sind für niedrige Drehzahlen wegen der Proportionalität zwischen der Amplitude des Drehzahlsignals und der Drehzahl selbst nicht brauchbar, weswegen sie in der modernen Antriebstechnik im Allgemeinen nicht mehr eingesetzt werden. Bei der Verwendung von Tiefpassfiltern im Rückführungszweig muss dieses nun bei der Auslegung des Reglers selbst berücksichtigt werden, was die Dynamik des geregelten Gesamtsystems nachteilig beeinflusst.

Speed control

As already done for current control, GPC speed control should also be compared to speed control with a PI-controller. For conventional control, the initial settings obtained according to Chapter 3.2 on page 23 were verified experimentally and then optimized. Finally, the values $T_\omega = 64.0$ and $V_\omega = 17.2$ have been selected as the best possible controller parameters.

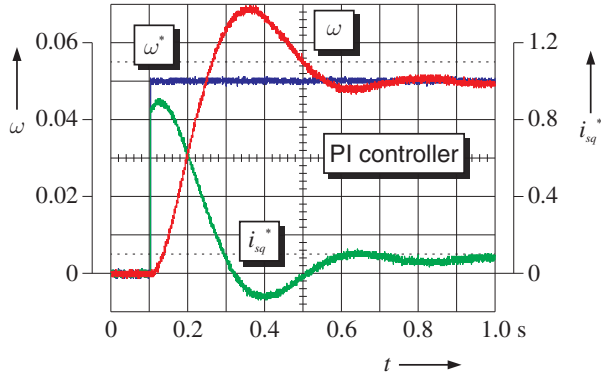
It is well known that the dynamics of the speed control loop are significantly lower than those of the current control loop. However, for technical programming reasons, both current and speed controllers have to be operated with the same sampling frequency. Now the prediction horizon of an MPC controller should approximately match the time constants of the plant to be controlled. Thus, a relatively high prediction length of $N_p = 200$ is necessary for a GPC speed controller. Nevertheless, since the control horizon is only $N_u = 1$, the mathematical complexity is limited to a certain extent. A higher value for N_u did not improve the controller behavior so that a larger control horizon obviously makes no sense. The remaining optimal parameters of the GPC controller are given in Table 6.3.

N_p	N_u	na	nb	nt	\tilde{a}_0	\tilde{a}_1	\tilde{a}_2	b_0	b_1	t_0	t_1	λ
200	1	2	1	1	1,0	-2,0	1,0	0,0	0,00013	1,0	-0,999	0,1

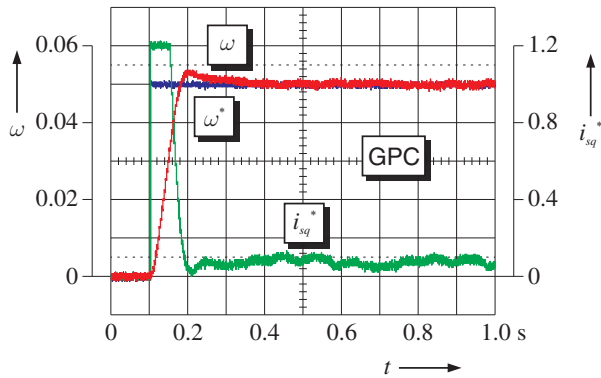
6.3: Parameter settings for a GPC speed controller

The large-signal behavior of a speed control loop gives no or only few information about the quality of the controller because in this case the dynamics are solely limited by the available inverter power. Hence, only the small-signal behavior is examined for the comparison between PI- and GPC controller for speed control. The results are presented in Figure 6.20.

In drive technology, it has quite often to be dealt with the problem that the measured speed signal must be filtered with a low-pass filter, as otherwise it is not useable for control. This problem appears especially when incremental encoders with low resolution are used, because the speed signal received by counting the increments per sampling cycle changes with distinct jumps. Of course, the problems described above do not appear when tacho generators are used as speed encoders; however, tacho generators are more expensive than incremental encoders, they also have to be maintained regularly and they are not useful at low speeds because of the proportionality between the amplitude of the output signal and the actual speed. This is why they are normally not used in modern drive technology. When a low-pass filter is inserted into the feedback path, this has to be considered in the controller design, leading to the fact that the dynamics of the overall control system are deteriorated.



(a)



(b)

6.20: Drehzahlregelung Kleinsignalverhalten: (a) PI-Regler; (b) GPC-Regler;
 Speed Control: Small-signal behavior: (a) PI-Regler; (b) GPC-Regler;
 转速控制的小信号响应: (a) PI 控制器, (b) GPC 控制器

转速控制

正如电流控制的情形，同样也应当比较基于 GPC 控制器和基于 PI 控制器的转速控制。对于传统的 PI 控制器来说，应依据章节 3.2 第 23 页中介绍的转速控制进行实验验证和优化。最终选择 $T_w = 64.0$ 和 $V_w = 17.2$ 作为最优的控制器参数。

转速控制闭环的动态特性要明显小于电流控制闭环。然而，由于软件编程的因素限制，电流控制器和速度控制器必须在相同的采样频率下运行。MPC 控制器的预测步长所对应的时间应当大致等于被控系统的时间常数，从而，GPC 转速控制器所需的预测步长大约为 $N_p = 200$ ；由于控制步长的值为 $N_u = 1$ ，所以整体的数学计算量可保持在一定的范围。此外，由于更大的控制步长 N_u 无法明显提高控制效果，所以选择更大的 N_u 是毫无意义的。表 6.4 给出了 GPC 转速控制器的其他参数设置。

N_p	N_u	na	nb	nt	\tilde{a}_0	\tilde{a}_1	\tilde{a}_2	b_0	b_1	t_0	t_1	λ
200	1	2	1	1	1,0	-2,0	1,0	0,0	0,00013	1,0	-0,999	0,1

6.4: GPC 转速控制器的参数设置

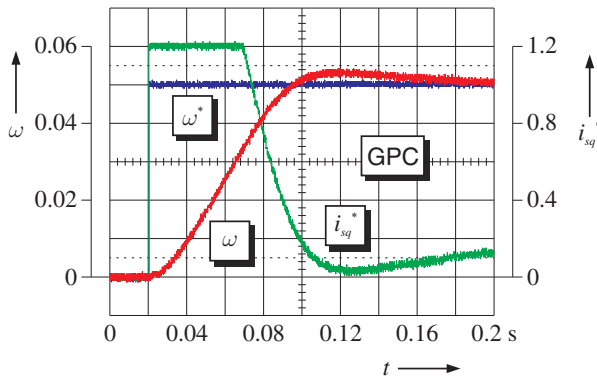
根据转速控制的大信号响应特性很难判断 GPC 控制器与 PI 控制器的孰优孰劣，因为转速的动态特性会受变频器功率的限制。基于上述考虑，GPC 转速控制器和 PI 转速控制器之间的对比仅在小信号响应条件下实施。图 6.20 给出了相应的实验结果。

在驱动技术中存在的一大问题是：所测量的转速信号必须使用低通滤波器进行滤波处理，否则是无法使用的。上述问题尤其会在所使用的增量式编码器具有较少光栅线数的情况下出现，因为线数越少意味着转速信号的跳变程度越为剧烈。然而，基于测速发电机的方式实现转速测量则不会出现上述问题，但要清楚的是，测速发电机的价格要比增量式编码器昂贵，而且需要定期维护。此外，由于转速信号的幅值与转速本身呈一定比例关系，所以在低速情况下，测速发电机是无法使用的。正因如此，在现代驱动技术中不再使用这一方式。在控制器设计时，应充分考虑到反馈通道中所使用的低通滤波器，因为它会对被控系统的动态特性造成不利影响。

Bei Verwendung eines modellbasierten Reglers hingegen kann die Filterung der gemessenen Regelgröße in den Regler selbst verlegt werden, so dass sich kein negativer Effekt bezüglich der Gesamtdynamik ergibt. Dies wurde bereits in Kap. 6.2 dargelegt.

Wie unschwer zu erkennen ist, zeigt der PI-Regler in Abb. 6.20(a) das typische Verhalten eines nach dem symmetrischen Optimum ausgelegten Regelkreises. Die Anstiegs- und Anregelzeiten sind relativ hoch, nicht zuletzt dadurch, dass die Maschine nur mit maximal 90% des Nennstroms beaufschlagt wird. Zwar ließe sich durch Verändern der Reglerparameter V_ω und T_ω der Überschwinger reduzieren; dies würde jedoch zu einer noch langsameren Dynamik des Gesamtsystems führen. Die Wahl von Parameterwerten, die ein weniger träges Verhalten ergeben, verbietet sich, da die Regelung wegen des Tiefpassfilters im Rückführzweig dann instabil wird.

Ein modellbasierter Prädiktivregler hingegen, dessen Messergebnisse in Abb. 6.20(b) dargestellt sind, kann im Drehzahlregelkreis den Vorteil der internen Messwertfilterung ausspielen. Erkennbar sind die deutlich verringerte Anstiegs- und Anregelzeit. Dabei ist die Dynamik hier noch nicht einmal durch den Regler selbst, sondern nur durch die Begrenzung des drehmomentbildenden Anteils des Statorstroms i_{sq} auf $i_{sq,max} = 1,2$ begrenzt. Durch die wesentlich verkürzte Anstiegszeit und den quasi nicht mehr vorhandenen Überschwinger zeigt sich die GPC-Regelung im Drehzahlregelkreis der konventionellen PI-Regelung deutlich überlegen. Abb. 6.21 zeigt nochmals eine zeitlich gedehnte Darstellung des Verhaltens eines GPC-Drehzahlreglers.



6.21: Drehzahlregelung Kleinsignalverhalten, zeitlich gedehnt;
Speed Control: Small-signal behavior, zoomed time scale;
局部放大的转速控制的小信号响应

However, in the case of using a model-based predictive controller, the filtering of the measured controlled variable can be shifted into the controller itself, so that no negative effect with respect to the overall system dynamics results. This has already been explained in Chapter 6.2.

As it can easily be seen, the PI-controller in Figure 6.20(a) shows the typical behavior of a controller designed according to the symmetrical optimum. The rise time is quite high, also due to the fact that the machine is supplied only with a maximum of 90 percent of the nominal current. Although the overshoot could be reduced by changing the controller parameters V_ω and T_ω , this would lead to even slower dynamics of the overall system. The selection of parameter values resulting into a less sluggish behavior must not be done, since the control will then become unstable because of the low pass filter in the feedback path.

However, a model-based predictive controller, whose response is shown in Figure 6.20(b), can use the advantage of the internal filtering of the measured signal in the speed control loop. The reduced rise and settling time can easily be seen. Here, the dynamics are not limited by the controller itself, only by the maximum value of the torque producing component of the stator current i_{sq} to $i_{sq,max} = 1.2$. The GPC controller in the speed control loop is clearly superior to conventional PI control because of the significantly shorter rise time and the almost not existing overshoot. Figure 6.21 once more shows the behavior of a GPC speed controller with a zoomed time scale.

然而，当使用基于模型的控制器的时，所测量的被控变量的滤波已转移至控制器模型中，所以不会妨碍被控系统的动态特性，如章节 6.2 所述。

由图 6.20(a) 不难看出，基于 PI 控制器所得到的是使用了对称最优方式下的控制闭环的典型响应特性。上升时间和再调时间相对较大，主要是因为电机额定电流仅仅达到其额定电流的 90% 左右，即使通过改变控制器参数 V_ω 和 T_ω 可以一定程度地减小超调量，但付出的代价是系统的动态特性将变得迟缓。此外，当调整控制器参数以使得系统具有更为迅速的动态特性时，需要额外注意的是，反馈通道中的低通滤波器有可能引起系统的不稳定。

图 6.20(b) 给出了基于模型的预测控制器得到的结果，它体现了在转速控制闭环中使用内部测量值滤波所带来的优势。可以看出，上升时间和再调时间均明显减小，而且此时的系统动态特性的限制并非由于控制器本身的因素，而是受到定子电流转矩分量最大值 $i_{sq,max} = 1.2$ 的制约。对于转速闭环控制来说，单是明显缩短了上升时间和更轻微的超调这两点，即可表明使用 GPC 控制器要明显优于使用 PI 控制器。图 6.21 给出了一个基于 GPC 控制器的小信号响应特性的局部放大图。

6.3.4 Rechenzeiten/Computation times/计算时间

Bei üblichen Regelverfahren der Antriebstechnik ist es extrem wichtig, die Rechenzeit des Regelprozesses kurz zu halten, da sonst die für eine gute Regelqualität notwendige hohe Abtastrate nicht gehalten werden kann. Wie die in Tabelle 6.5 aufgeführten Messergebnisse zeigen, ist dies jedoch auch mit einer modellbasierten Prädiktivregelung nach dem Prinzip GPC zu erreichen. So benötigt ein GPC-Regler im Stromregelkreis nicht mehr Rechenzeit als ein PI-Regler. Beim Einsatz als Drehzahlregler macht sich beim modellbasierten Prädiktivregler der hier notwendige hohe Prädiktionshorizont von $N_p = 200$ bemerkbar. In Folge steigt die Berechnungsdauer auf etwa den vierfachen Wert gegenüber einem konventionellen Reglertyp an. Trotzdem liegt die Gesamtdauer noch in einem Bereich, der die Anwendung von GPC auch für schnell schaltende Umrichter erlaubt. Betrachtet man darüberhinaus den in Kap. 6.3.3 auf Seite 127 ff. dargestellten Performancegewinn, so wiegt dieser den notwendigen rechentechnischen Mehraufwand in vielen Fällen auf.

Typ	PI	GPC
Stromregler	$2 \times 2,3 \mu\text{s}$	$2 \times 2,3 \mu\text{s}$
Drehzahlregler	$2,3 \mu\text{s}$	$8,8 \mu\text{s}$

6.5: Rechenzeitvergleich PI-Regler und GPC-Regler

Als Prozessrechner kam für diese und auch für alle folgenden Messungen ein mit einem AMD Duron[®] Prozessor (900 MHz Taktfrequenz) bestückter handelsüblicher PC unter dem Echtzeitbetriebssystem Linux/RTAI zum Einsatz. Die erzielten Ergebnisse sind daher von den absoluten Werten her nicht unbedingt auf die in der Antriebstechnik üblichen Mikrorechner übertragbar; das Verhältnis der Rechenzeiten von PI-Regler und GPC-Regler dürfte aber durchaus vergleichbar sein.

Die Berechnung der Systemmatrizen für den GPC-Regler erfordert eine Matrixinversion. Bei den in dieser Arbeit vorgestellten antriebstechnischen Beispielen wurde auf eine Parameteradaption des Streckenmodells verzichtet; die Matrixinversion kann daher offline durchgeführt werden und gehört nicht zu den zeitkritischen Programmteilen. Soll jedoch eine Nachführung der Modellparameter implementiert werden, so hat die Inversion der Systemmatrizen während des laufenden Regelprozesses zu erfolgen. Auch wenn dies nicht innerhalb eines Abtastschritts durchgeführt werden muss, so sollte doch auf eine zeitsparende Programmierung dieser Aufgabe geachtet werden. Ein Vergleich unterschiedlicher Verfahren zur Matrixinversion und der dafür benötigten Rechenzeiten ist in Anhang F zu finden.

6.3.4 Computation times

For the conventional control methods in drive technology, it is extremely important to keep the computation time of the control algorithm as short as possible, because, otherwise, the high sampling rate necessary for good control quality cannot be obtained. However, as the measurement results given in [Table 6.6](#) show, this can also be achieved with model-based predictive control based on the GPC principle. A GPC controller in the current control loop does not need more computation time than a PI-controller. However, when a model-based predictive controller is used as a speed controller, the high prediction horizon with $N_p = 200$ required here becomes apparent. As a result, the computation time is increased to a value being 4 times higher compared to the value of a conventional PI-controller. Nevertheless, the total time is still within a range which permits the use of GPC even for fast switching inverters. In addition, if the performance gains represented in [Chapter 6.3.3](#) on page 127 et seqq. are considered the additional computational expenses are counterbalanced in many cases.

Type	PI	GPC
Current controller	$2 \times 2.3 \mu\text{s}$	$2 \times 2.3 \mu\text{s}$
Speed controller	$2.3 \mu\text{s}$	$8.8 \mu\text{s}$

6.6: Comparison of computation times for PI and GPC controller

As process computer for these and all other measurements a commercial PC with an AMD Duron[®] processor (900 MHz clock frequency) was used together with the real time operating system Linux/RTAI. Thus, the results presented here are in terms of absolute values not necessarily comparable to results that can be achieved with standard microcomputers that are normally used in drive technology; however, the ratio of the computation times for a PI- and a GPC controller should be comparable.

The calculation of the system matrices for the GPC controller involves a matrix inversion. For the drive-technological examples presented in this work parameter adaptation of the system model was omitted; hence, the matrix inversion can be carried out offline and thus, it does not belong to the time critical parts of the program. However, if an adaption of the model parameters should be implemented, then the inversion of the system matrices has to be done during the run time of the control process. Although this does not have to be done within one sampling cycle, a time-saving programming of this task should be taken into consideration. A comparison of different procedures for matrix inversion and the computation times required for these tasks can be found in [appendix F](#).

Notizen/Notes/笔记

6.3.4 计算时间

对于驱动技术领域常见的控制方法来说，保证在较短的计算时间实现控制过程是极其重要的，否则将无法达到为实现良好控制效果所必需的采样速率。如表 6.7 中的测量结果所示，基于 GPC 原理所实现的模型预测控制具有较短的计算时间。对于电流闭环控制来说，GPC 控制器所需的计算时间并未超过 PI 控制器所需的计算时间；对于转速控制器来说，由于基于模型的预测控制器需要非常大的预测步长 $N_p = 200$ ，这一情况下的计算时间大约为 PI 控制器所需计算时间的 4 倍。尽管如此，整体所需的计算时间仍能够满足具有较高频率的变流器所允许的范围。此外，根据章节 6.3.3 第 127 页所述的性能对比可知，在大多情况下，控制性能的改善是以增加的计算量为代价的。

类型	PI	GPC
电流控制器	$2 \times 2.3 \mu\text{s}$	$2 \times 2.3 \mu\text{s}$
速度控制器	$2.3 \mu\text{s}$	$8.8 \mu\text{s}$

6.7: PI 和 GPC 控制器的计算时间对比

本章以及后续所有的测量结果所使用的均是基于配备 AMD Duron[®] 处理器（主频为 900 MHz）的工业级计算机系统，其操作系统为 Linux/RTAI。因此，无法保证本书中的所有结果与选用驱动技术领域常用的微处理器所得结果的一致性。当然，实现 PI 控制器和 GPC 控制器所需计算时间的比例关系不会有较大出入。

要指出的是，GPC 控制器系统矩阵的计算需要完成对矩阵的求逆运算。最后，本研究工作所介绍的相关案例均放弃使用具有参数自适应特性的被控模型，因此矩阵求逆运算可以离线完成，从而不占用在线计算资源。然而，若选择使用具有参数自适应的被控模型，那么系统矩阵的求逆运算则需在控制过程的运行时间内完成，此时无法离线完成。尽管并非要求上述运算一定在单个采样周期内实现，但也应当考虑如何节省计算量。不同的矩阵求逆方法及其所需计算时间见附录 F。

7 Zeitdiskretes Maschinenmodell für Stromregelung/Time-discrete Machine Model for Current Control/用于电流控制的时间离散电机模型

Da die Reglerkaskade einige prinzipbedingte Nachteile mit sich bringt [67] und eine modellbasierte prädiktive Regelung die Möglichkeit einer Mehrgrößenregelung bietet, sollte dieser Vorteil auch ausgenutzt werden. Als erstes Beispiel soll ein MIMO-Stromregler für die feldorientierte Regelung einer Asynchronmaschine, d. h. ein Stromregler, der gleichzeitig sowohl den feldbildenden als auch den drehmomentbildenden Anteil des Statorstroms i_s regelt, hergeleitet werden. Für solch eine mehrdimensionale Regelung ist selbstverständlich auch ein mehrdimensionales Streckenmodell notwendig. Da ein modellbasierter Prädiktivregler zum Einsatz kommen soll, muss als MIMO-Streckenmodell ein zeitdiskretes Maschinenmodell verwendet werden.

7.1 Herleitung/Derivation/推导

Ein lineares System lässt sich anschaulich in der sogenannten Zustandsraumbeschreibung darstellen (s. Gln. (5.2) und (5.3) auf Seite 61). Für ein Maschinenmodell, welches dieser Beschreibung entspricht, wählt man sinnvollerweise folgende Größen für die Vektoren \mathbf{x} , \mathbf{u} und \mathbf{y} :

$$\mathbf{x} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}$$

Es ergibt sich dann aus den bekannten Maschinengleichungen (2.11)–(2.13) folgende Zustandsraumbeschreibung:

$$\frac{d}{d\tau} \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_{\sigma'}} & \omega_s & \frac{k_r}{r_{\sigma}\tau_r\tau_{\sigma'}} \\ -\omega_s & -\frac{1}{\tau_{\sigma'}} & -\frac{k_r\omega}{r_{\sigma}\tau_{\sigma'}} \\ \frac{l_h}{\tau_r} & 0 & -\frac{1}{\tau_r} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} + \begin{bmatrix} \frac{1}{r_{\sigma}\tau_{\sigma'}} & 0 \\ 0 & \frac{1}{r_{\sigma}\tau_{\sigma'}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \quad (7.1)$$

Since a cascaded controller structure has some drawbacks due to its functional principle [67] and since model-based predictive control offers the possibility of MIMO control, this advantage should be utilized. As a first example, a MIMO current controller for field-oriented control of an induction machine should be derived, i. e. a current controller that simultaneously controls the flux producing as well as the torque producing component of the stator current \mathbf{i}_s . For such a multidimensional control, a multidimensional plant model is necessary, too. As a model-based predictive controller should be used, the MIMO plant model of the machine has to be discrete-time.

7.1 Derivation

A linear system can be clearly described in the so-called state space representation (see equations (5.2) and (5.3) on page 61). For a machine model according to this representation, it makes sense to use the following variables for the vectors \mathbf{x} , \mathbf{u} and \mathbf{y} :

$$\mathbf{x} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}$$

The following state space representation results from the known machine equations (2.11)–(2.13), see equations (7.1) and (7.2).

由于串级控制本身固有的缺点 [67]，以及基于模型的预测控制可以实现多变量控制这一优点，所以应充分利用上述优点并尽可能地避免缺点。选取用于异步电机磁场定向控制的 MIMO 电流控制器作为首例，这一电流控制器可以同时控制定子电流 \mathbf{i}_s 的励磁分量和转矩分量。对于上述多维控制来说，一个多维的被控系统模型显然是不可或缺的。由于要实现的是一个基于模型的预测控制器，所以必须使用时间离散的电机模型作为 MIMO 被控系统模型。

7.1 推导

一个线性系统可以使用所谓的状态空间描述方式清晰表达(见第 61 页方程 (5.2) 和 (5.3))。对于电机模型来说，通常使用下述向量 \mathbf{x} ， \mathbf{u} 和 \mathbf{y} 用于上述状态空间描述形式：

$$\mathbf{x} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}$$

根据已知的电机方程 (2.11) – (2.13)，可得到如下所示的状态空间描述方程：见方程 (7.1) 和 (7.2)。

$$\begin{bmatrix} \dot{i}_{sd} \\ \dot{i}_{sq} \\ \dot{\psi}_{rd} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \quad (7.2)$$

Bei einem linearen, zeitinvarianten System sind die Werte der Elemente der Matrizen **A**, **B**, **C** und **D** konstant. In unserem Fall gilt dies jedoch für die Drehgeschwindigkeiten ω_s und ω nicht, da sich sie im Betrieb ändern. Die mathematisch korrekte Methode, ω_s und ω in den Eingangsvektor **u** aufzunehmen, ist leider nicht anwendbar, da die Drehgeschwindigkeiten mit verschiedenen Zustandsgrößen multiplikativ verknüpft sind, was eine Nichtlinearität ergibt. Die Zustandsraumbeschreibung für lineare Systeme kann dies nicht darstellen. Da der Drehzahlregelkreis aber eine vielfach höhere Zeitkonstante als der Stromregelkreis besitzt, können ω_s und ω für die Laufzeit des Stromreglers als konstant angenommen werden. Daher werden ω_s und ω quasi wie Parameter behandelt. Die übrigen Werte sind konstant. Für den im Experiment verwendeten Motor nach Anhang D ergeben sich die Matrizen zu:

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{\tau_{\sigma'}} & \omega_s & \frac{k_r}{r_{\sigma}\tau_r\tau_{\sigma'}} \\ -\omega_s & -\frac{1}{\tau_{\sigma'}} & -\frac{k_r\omega}{r_{\sigma}\tau_{\sigma'}} \\ \frac{l_h}{\tau_r} & 0 & -\frac{1}{\tau_r} \end{bmatrix} = \begin{bmatrix} -0,3964 & \omega_s & 0,07380 \\ -\omega_s & -0,3964 & -4,450\omega \\ 0,04245 & 0 & -0,01658 \end{bmatrix} \quad (7.3)$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{r_{\sigma}\tau_{\sigma'}} & 0 \\ 0 & \frac{1}{r_{\sigma}\tau_{\sigma'}} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 4,641 & 0 \\ 0 & 4,641 \\ 0 & 0 \end{bmatrix} \quad (7.4)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.5)$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (7.6)$$

Diese Gleichungen müssen zur Verwendung in einem modellbasierten Regelverfahren nun in die zeitdiskrete Form gemäß (5.4) und (5.5) umgeformt werden. Dazu existieren mehrere Verfahren.

Considering a linear, time-invariant system, the values of the elements of the matrices **A**, **B**, **C** and **D** are constant. However, in our case, this is not true for the rotating speeds ω_s and ω , as they change their values during machine operation. Unfortunately, the mathematically correct method, i. e. including ω_s and ω in the input vector **u** is not applicable since the rotating speeds are multiplicatively linked with different state variables resulting in a nonlinearity. The state space representation for linear systems cannot represent this fact. As the time constant of the speed control loop has a much higher value than the one of the current control loop, ω_s and ω can be treated as constants for the runtime of the current controller. Hence, ω_s and ω are treated as parameters similar to other machine parameters. The values of the remaining parameters are constant. For the motor used for the experiments, the matrices result (according to appendix D) to equations (7.3)–(7.6).

To use these equations in a model-based control method, must be transformed into a time-discrete form according to (5.4) and (5.5). For this, several procedures exist.

对于线性时不变系统来说，矩阵 **A**、**B**、**C** 和 **D** 中元素的值是常数。然而，本例中的转速 ω_s 和 ω 却无法满足上述条件，因为它们在电机运行过程中是实时变化的。纯数学上可行的方案是将 ω_s 和 ω 纳入输入向量 **u**，但现实中是行不通的，因为转速与其他不同的状态变量之间同时存在非线性关系。所以，基于上述方案无法完成线性系统的状态空间描述。但是，考虑到转速闭环具有比电流闭环更大的时间常数，从而可以假设转速 ω_s 和 ω 在电流控制器的运行时间内保持恒定，如此即可将转速 ω_s 和 ω 看作常数进行处理。同时，其他元素的值均为常数。根据附录 D 中所给出的电机参数，可以得到矩阵表达式 (7.3) – (7.6)。

上述状态方程和输出方程必须转化为方程 (5.4) 和 (5.5) 所表示的时间离散表述形式，才能被用于基于模型的控制方法。实际存在多种方式可实现上述转化，接下来将介绍常见的四种方式。

7.1.1 MATLAB

Das mathematische Computeralgebrasystem *MATLAB*[®] der Firma *The MathWorks Inc.* ist für die Berechnung von Matrizen gut geeignet. Zur einfachen Behandlung von regelungstechnischen Problemen ist ein Ergänzungspaket *Control System Toolbox* erhältlich. Mit den in diesem Paket verfügbaren Funktionen können komplette Systemmodelle von zeitkontinuierlicher in zeitdiskrete Form und von der Zustandsraumbeschreibung in die Übertragungsfunktionen des Systems umgeformt werden. Leider beherrscht *MATLAB*[®] ohne weitere Ergänzungen keine symbolische Mathematik, so dass sich stets nur eine numerische Lösung für die verschiedenen Parameter bzw. Elemente angeben lässt. Da dies in unserem Fall wegen der Änderung von ω_s und ω nicht brauchbar ist, kann *MATLAB*[®] so nicht verwendet werden. Für die weiteren Versuche kommt daher die auch für symbolische Mathematik einsetzbare Software *Maple*[®] der Firma *Waterloo Maple Inc.* zum Einsatz.

7.1.2 Differenzenquotient/Difference quotient/微商

Die einfachste Möglichkeit, eine zeitkontinuierliche Beschreibung eines Vorgangs in eine zeitdiskrete zu verwandeln, besteht darin, den Ableitungsoperator durch eine Differenz anzunähern.

$$\frac{dx}{dt} \approx \frac{x(k+1) - x(k)}{\Delta t} \quad (7.7)$$

Wendet man dies auf die Gln. (5.2) und (5.3) an und setzt als Zeitdifferenz Δt die bei einer digitalen Regelung konstante Abtastzeit T_0 ein, so erhält man

$$\frac{\mathbf{x}(k+1) - \mathbf{x}(k)}{T_0} = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (7.8)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (7.9)$$

Gl. (7.8) lässt sich weiter umformen:

$$\begin{aligned} \mathbf{x}(k+1) - \mathbf{x}(k) &= T_0\mathbf{A}\mathbf{x}(k) + T_0\mathbf{B}\mathbf{u}(k) \\ \mathbf{x}(k+1) &= (T_0\mathbf{A} + \mathbf{I})\mathbf{x}(k) + T_0\mathbf{B}\mathbf{u}(k) \end{aligned} \quad (7.10)$$

Ein Vergleich von (7.10) mit (5.4) sowie (7.9) mit (5.5) ergibt für die Matrizen der diskreten Darstellung folgende Identitäten:

$$\mathbf{A}_d = T_0\mathbf{A} + \mathbf{I} \quad (7.11)$$

$$\mathbf{B}_d = T_0\mathbf{B} \quad (7.12)$$

$$\mathbf{C}_d = \mathbf{C} \quad (7.13)$$

$$\mathbf{D}_d = \mathbf{D} \quad (7.14)$$

7.1.1 MATLAB

The mathematical computer algebra system *MATLAB*[®] from *The MathWorks Inc.* is a powerful tool for the calculation of matrices. For an easy treatment of control systems the additional *Control System Toolbox* is available. With the functions available within this toolbox, complete system models can be transformed from a time-continuous into a time-discrete form and from state space representation into a transfer function based system model. Unfortunately, *MATLAB*[®] cannot handle symbolic mathematics without further extensions, thus, only numerical solutions can be given for the different parameters and elements. Since this is not useable in our case because of the varying parameters ω_s and ω , *MATLAB*[®] cannot be used in this way. Hence, the software *Maple*[®] from *Waterloo Maple Inc.* which is able to handle symbolic mathematics is used for the further work, too.

7.1.2 Difference quotient

The most simple way to transform a time-continuous representation of a system into a time-discrete representation consists of approximating the derivative operator with a difference, see equation (7.7). If this is applied to the equations (5.2) and (5.3) and if the constant sampling time T_0 is used for the time difference Δt , the following equations (7.8) and (7.9) can be obtained. Equation (7.8) can be further transformed into equation (7.10). A comparison of (7.10) with (5.4) and of (7.9) with (5.5), respectively, delivers the following expressions for the matrices of the time-discrete representation:

see equations (7.11)–(7.14).

7.1.1 MATLAB

The MathWorks Inc. 公司的数学软件 *MATLAB*[®] 非常适用于矩阵计算。为了更为简便地处理控制技术的相关任务, *MATLAB*[®] 还提供了特定的附加工具箱, 如控制系统工具箱。基于上述工具箱中的相关函数可以轻易地将复杂的系统模型由时间连续形式转化为时间离散形式, 也可将状态空间描述形式转化为系统的传递函数形式。遗憾的是, *MATLAB*[®] 目前无法实现符号计算, 因此对于含有不同参数及元素的情形也仅能够提供数值解。由于本例中含有变化的转速 ω_s 和 ω , 使得无法借助于 *MATLAB*[®] 进行求解。*Waterloo Maple Inc.* 公司提供的 *Maple*[®] 数学软件可以实现符号运算和数学计算, 可以用于求解本例。

7.1.2 微商

时间连续描述形式转化为时间离散描述形式最为简单的方法是借助于微分算子进行近似处理, 见方程 (7.7)。若将上述方程应用至方程 (5.2) 和 (5.3), 并使用数字控制方式下的恒定采样时间 T_0 来代替时间增量 Δt , 可得方程 (7.8) 和 (7.9)。进一步适当变换方程 (7.8), 可得方程 (7.10)。通过对比方程 (7.10) 和 (5.4), 以及方程 (7.9) 和 (5.5), 可以得到离散描述方式下的矩阵等式 (7.11) – (7.14)。

Führt man diese Transformation für (7.3)-(7.6) durch, so erhält man als Ergebnis:

$$\mathbf{A}_d = \begin{bmatrix} 1 - \frac{T_0}{\tau_{\sigma'}} & T_0 \omega_s & \frac{T_0 k_r}{r_{\sigma} \tau_r \tau_{\sigma'}} \\ -T_0 \omega_s & 1 - \frac{T_0}{\tau_{\sigma'}} & -\frac{T_0 k_r \omega}{r_{\sigma} \tau_{\sigma'}} \\ \frac{T_0 l_h}{\tau_r} & 0 & 1 - \frac{T_0}{\tau_r} \end{bmatrix} \quad (7.15)$$

$$= \begin{bmatrix} 0,9872 & 0,03217 \omega_s & 0,002374 \\ -0,03217 \omega_s & 0,9872 & -0,1432 \omega \\ 0,001366 & 0 & 0,9995 \end{bmatrix}$$

$$\mathbf{B}_d = \begin{bmatrix} \frac{T_0}{r_{\sigma} \tau_{\sigma'}} & 0 \\ 0 & \frac{T_0}{r_{\sigma} \tau_{\sigma'}} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0,1493 & 0 \\ 0 & 0,1493 \\ 0 & 0 \end{bmatrix} \quad (7.16)$$

$$\mathbf{C}_d = \mathbf{C} \quad \text{unverändert} \quad (7.17)$$

$$\mathbf{D}_d = \mathbf{D} \quad \text{unverändert} \quad (7.18)$$

7.1.3 Laplace-Transformation/Laplace transformation/拉普拉斯变换

Ein mathematisch exaktes Berechnen der Matrizen der zeitdiskreten Zustandsraumbeschreibung ist mit der oben erläuterten einfachen Methode natürlich nicht möglich. Dazu muss statt dessen die Lösung der Matrixdifferentialgleichung (5.2) bestimmt werden. Wie im skalaren Fall wählt man als Ansatz:

$$\mathbf{x}(t) = e^{\mathbf{A}t} \mathbf{k} \quad (7.19)$$

Die Matrix $e^{\mathbf{A}t}$ wird als Matrixexponentialfunktion bezeichnet und ist durch folgende Reihe definiert [84]:

$$e^{\mathbf{A}t} = \sum_{i=0}^{\infty} \frac{\mathbf{A}^i t^i}{i!} = \mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2}{2!} t^2 + \frac{\mathbf{A}^3}{3!} t^3 + \frac{\mathbf{A}^4}{4!} t^4 + \dots \quad (7.20)$$

Aus (5.2) erhält man mit Hilfe des Ansatzes (7.19) die Bewegungsgleichung

$$\mathbf{x}(t) = \Phi(t) \mathbf{x}_0 + \int_0^t \Phi(t-\tau) \mathbf{B} \mathbf{u}(\tau) d\tau \quad (7.21)$$

mit

$$\Phi(t) = e^{\mathbf{A}t} \quad (7.22)$$

If the above transformations are carried out on (7.3)–(7.6), the following results can be obtained, see equations (7.15)–(7.18).

7.1.3 Laplace transformation

A mathematically exact calculation of the matrices of the time-discrete state space representation is of course not possible with the simple method described above. For this purpose, the solution of the matrix differential equation (5.2) has to be determined. The following expression is selected analog to the scalar case, see equation (7.19).

The matrix $e^{\mathbf{A}t}$ is called *matrix exponential function*. It is defined by the following series [84]:

$$e^{\mathbf{A}t} = \sum_{i=0}^{\infty} \frac{\mathbf{A}^i t^i}{i!} = \mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2}{2!} t^2 + \frac{\mathbf{A}^3}{3!} t^3 + \frac{\mathbf{A}^4}{4!} t^4 + \dots \quad (7.20)$$

From (5.2) and with the expression (7.19) the equation of motion can be obtained

$$\mathbf{x}(t) = \Phi(t)\mathbf{x}_0 + \int_0^t \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau \quad (7.21)$$

with

$$\Phi(t) = e^{\mathbf{A}t} \quad (7.22)$$

对方程 (7.3) – (7.6) 实施上述变换, 可得矩阵表达式 (7.15) – (7.18)。

7.1.3 拉普拉斯变换

对于时间离散状态空间描述中的矩阵的精确数学计算来说, 单纯借助于上面所述的简单方法显然是行不通的。可行的方法是确定矩阵微分方程 (5.2) 的解。类似于标量时的情形, 使用下述定义 (7.19)。

矩阵 $e^{\mathbf{A}t}$ 也被称为矩阵指数函数, 它可通过级数方式进行定义 [84]:

$$e^{\mathbf{A}t} = \sum_{i=0}^{\infty} \frac{\mathbf{A}^i t^i}{i!} = \mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2}{2!} t^2 + \frac{\mathbf{A}^3}{3!} t^3 + \frac{\mathbf{A}^4}{4!} t^4 + \dots \quad (7.20)$$

根据方程 (5.2) 并结合表达式 (7.19), 可得到如下运动方程:

$$\mathbf{x}(t) = \Phi(t)\mathbf{x}_0 + \int_0^t \Phi(t-\tau)\mathbf{B}\mathbf{u}(\tau)d\tau \quad (7.21)$$

其中,

$$\Phi(t) = e^{\mathbf{A}t} \quad (7.22)$$

wobei Φ als *Übergangsmatrix*, *Transitionsmatrix* oder *Fundamentalmatrix* bezeichnet wird. Wählt man nun die Zeit $t = (k + 1)T_0$, so erhält man aus (7.21) mit Hilfe der Definition (7.22):

$$\mathbf{x}((k + 1)T_0) = e^{\mathbf{A}(k+1)T_0} \mathbf{x}_0 + \int_0^{(k+1)T_0} e^{\mathbf{A}((k+1)T_0 - \tau)} \mathbf{B}\mathbf{u}(\tau) d\tau \quad (7.23)$$

Bei einer digitalen Regelung ändert sich die Stellgröße nur zu den Abtastzeitpunkten. Der Wert von $\mathbf{u}(t)$ ist daher konstant für $kT_0 \leq t \leq (k + 1)T_0$. Somit folgt aus (7.23) nach einigen Zwischenschritten und der Substitution $\alpha = (k + 1)T_0 - \tau$ [85]:

$$\mathbf{x}((k + 1)T_0) = e^{\mathbf{A}T_0} \mathbf{x}(kT_0) + \int_0^{T_0} e^{\mathbf{A}\alpha} d\alpha \mathbf{B}\mathbf{u}(kT_0) \quad (7.24)$$

Durch Koeffizientenvergleich mit (5.4) lassen sich \mathbf{A}_d und \mathbf{B}_d direkt ablesen:

$$\mathbf{A}_d = e^{\mathbf{A}T_0} \quad (7.25)$$

$$\mathbf{B}_d = \int_0^{T_0} e^{\mathbf{A}\alpha} d\alpha \mathbf{B} \quad (7.26)$$

Unter der Voraussetzung eines kontinuierlichen Systems gilt für die Determinante von \mathbf{A} , dass $\det \mathbf{A} \neq 0$; die Eingangsmatrix des diskreten Systems ergibt sich dadurch zu:

$$\mathbf{B}_d = \mathbf{A}^{-1} \left(e^{\mathbf{A}T_0} - \mathbf{I} \right) \mathbf{B} \quad (7.27)$$

Mit Hilfe der Gln. (7.25) und (7.27) lassen sich die Systemmatrix \mathbf{A}_d und die Eingangsmatrix \mathbf{B}_d des zeitdiskreten Systems berechnen. Da die Ausgangsgleichung (5.3) keine Ableitungsoperatoren enthält, kann sie einfach durch die Substitution $t = kT_0$ in die diskrete Form überführt werden. Es zeigt sich im Vergleich mit (5.5), dass $\mathbf{C}_d = \mathbf{C}$ und $\mathbf{D}_d = \mathbf{D}$ immer gilt.

Zur Transformation der zeitkontinuierlichen Zustandsraumbeschreibung in die zeitdiskrete ist es nun noch erforderlich, die Matrixexponentialfunktion $e^{\mathbf{A}t}$ zu berechnen. Neben der in Gl. (7.20) angegebenen Reihenentwicklung zeigt Schwarz [108], dass $e^{\mathbf{A}t}$ auch mit Hilfe der Laplace-Transformation bestimmt werden kann:

$$e^{\mathbf{A}t} = \mathfrak{L}^{-1} \{ (s\mathbf{I} - \mathbf{A})^{-1} \} \quad (7.28)$$

Φ is called *transition matrix*. If the time is set to $t = (k + 1)T_0$, the following equation results from (7.21) with the help of definition (7.22), see equation (7.23).

In a digital control the values of the actuating variables are only changed at the sampling instances. Hence, the value of $\mathbf{u}(t)$ is constant for $kT_0 \leq t \leq (k + 1)T_0$. Thus, the following equation results from (7.23) after some intermediate steps and the substitution $\alpha = (k + 1)T_0 - \tau$ [85], see equation (7.24).

By comparing the coefficients of the above equation with (5.4), \mathbf{A}_d and \mathbf{B}_d can be determined directly, equation (7.25) and (7.26).

Assuming a continuous system, it can be stated that $\det \mathbf{A} \neq 0$; the input matrix of the time-discrete system thus results to equation (7.27).

With the help of the equations (7.25) and (7.27), the state matrix \mathbf{A}_d and the input matrix \mathbf{B}_d of the time-discrete system can be calculated. Since the output equation (5.3) does not contain derivative operators, it can easily be transformed into the discrete form with the substitution $t = kT_0$. Comparing it to (5.5), it can be seen that $\mathbf{C}_d = \mathbf{C}$ and $\mathbf{D}_d = \mathbf{D}$ does always apply.

For the transformation of the time-continuous state space representation into the time-discrete one, it is necessary to calculate the matrix exponential function $e^{\mathbf{A}t}$. Besides the series expansion given in equation (7.20), Schwarz [108] shows that $e^{\mathbf{A}t}$ can also be determined with the help of the Laplace transformation, see equation (7.28).

Φ 被称为过渡矩阵、转移矩阵或基础矩阵。如果选择时间 $t = (k + 1)T_0$ ，那么根据方程 (7.21) 并结合定义 (7.22)，可得到方程 (7.23)。

在数字控制中，控制器输出变量仅在采样时刻发生变化。因此， $\mathbf{u}(t)$ 的值在 $kT_0 \leq t \leq (k + 1)T_0$ 时间范围内保持不变。从而，方程 (7.23) 经过一系列中间步骤并由 $\alpha = (k + 1)T_0 - \tau$ 所替换 [85]，可得方程 (7.24)。

对比上述方程与方程 (5.4) 中的系数，可确定矩阵 \mathbf{A}_d 和 \mathbf{B}_d ，见表达式 (7.25) 和 (7.26)。假设在连续系统的前提下，满足系统矩阵 \mathbf{A} 的行列式不等于零这一条件，也即 $\det \mathbf{A} \neq 0$ ，从而可以得到离散系统的输入矩阵，见方程 (7.27)。

基于方程 (7.25) 和 (7.27) 可以计算出时间离散系统的系统矩阵 \mathbf{A}_d 和输入矩阵 \mathbf{B}_d 。由于输出方程 (5.3) 不包含微分算子，可令 $t = kT_0$ ，即可轻易地将其转化为离散形式。进一步，将其与方程 (5.5) 进行对比，可知 $\mathbf{C}_d = \mathbf{C}$ 和 $\mathbf{D}_d = \mathbf{D}$ 依然有效。

为了实现时间连续状态空间描述方式向时间离散状态空间描述方式的转换，矩阵指数函数 $e^{\mathbf{A}t}$ 的计算是不可或缺的。除了方程 (7.20) 所示的级数展开方式以外，学者 Schwarz 指出 [108]， $e^{\mathbf{A}t}$ 也可以借助于拉普拉斯逆变换的方式求得，相应的表达式为 (7.28)。

Da Maple® Funktionen zur Durchführung von Laplace-Operationen beinhaltet, ist obige Transformation einfach zu implementieren. Eine analytische Berechnung zeigt aber, dass in diesem Fall die System- und Eingangsmatrix der zeitdiskreten Darstellung mehr als 30 Seiten DIN A 4 umfassen. Auch wenn sicher noch einige Zusammenfassungen und Vereinfachungen möglich sein werden, so ist das Ergebnis wegen des enormen Rechenzeitbedarfs sicher nicht für eine Echtzeitregelung implementierbar. Damit scheidet die analytisch exakte Berechnung der zeitdiskreten Darstellung mittels Laplace-Transformation aus.

7.1.4 Potenzreihe/Power series/幂级数

Als dritte Möglichkeit zur Berechnung der Matrixexponentialfunktion bietet sich an, die Reihenentwicklung aus Gl. (7.20) nach dem Erreichen der gewünschten Genauigkeit abzubrechen. Beendet man die Reihenentwicklung nach dem ersten Glied, so erhält man:

$$e^{AT_0} \approx T_0 \mathbf{A} + \mathbf{I} \quad (7.29)$$

Dies entspricht mathematisch dem Ersatz des Ableitungsoperators durch eine Differenz, wie ein Einsetzen von (7.29) in die Gln. (7.25) und (7.27) und Vergleich mit den in Kap. 7.1.2 hergeleiteten Umrechnungsformeln (7.11) und (7.12) zeigt.

Bei einem Abbruch nach dem zweiten Glied werden die Berechnungsformeln der einzelnen Matrixelemente selbstverständlich aufwändiger. System- und Eingangsmatrix lauten nun:

Since Maple[®] contains functions for the execution of Laplace operations the above transformation can be implemented easily. However, an analytic calculation shows that in this case the state and input matrix of the time-discrete representation require more than 30 DIN A 4 pages. Even though some truncations and simplifications are certainly possible, the result can surely not be implemented in a real time control due to its extremely high computation time. Thus, the analytically exact calculation of the time-discrete representation by means of Laplace transformation cannot be considered to be feasible anymore.

7.1.4 Power series

The third possibility for the calculation of the matrix exponential function is by truncating the series expansion from equation (7.20) after the desired accuracy is obtained. If the series expansion is truncated after the first element, the following expression can be obtained:

$$e^{AT_0} \approx T_0\mathbf{A} + \mathbf{I} \quad (7.29)$$

Mathematically, this corresponds to a substitution of the derivative operator with a difference, as shown by a substitution of (7.29) in the equations (7.25) and (7.27) and by a comparison with the transformation equations (7.11) and (7.12), which were derived in Chapter 7.1.2.

In the case of a truncation after the second element, the calculation equations for the individual matrix elements do of course become more complex. Then, the state and input matrix are:

由于 Maple[®] 中包含实现拉普拉斯运算的相关函数，从而可以轻易地实现上述变换。但是，计算结果表明，本例中的时间离散状态空间描述的系统矩阵和输入矩阵需要占据超过 30 页 DIN A 4 纸的篇幅。即使做出一些简化，仍然需要足够长的计算时间，显然，这对于实时控制来说是无法接受的。由此得出结论，可以排除借助于拉普拉斯逆变换的方式来求得时间离散状态空间描述的矩阵精确解析解的方法。

7.1.4 幂级数

矩阵指数函数计算的第三种方法是在满足所期望的精度条件下对方程 (7.20) 所示的级数展开式进行截断处理。如果仅取所述级数展开式的前两项，可得：

$$e^{AT_0} \approx T_0\mathbf{A} + \mathbf{I} \quad (7.29)$$

数学上来看，上述近似等效于章节 7.1.2 所介绍的借助于微分算子取近似的方法，只需将方程 (7.29) 代入至方程 (7.25) 和 (7.27)，即可得到与章节 7.1.2 所给出的方程 (7.11) 和 (7.12) 完全一致的形式。

当取级数展开式的前三项时，矩阵元素计算的过程则明显变得复杂，此时得到的系统矩阵和输入矩阵形式如下页所示：

$$\mathbf{A}_d = \left[\begin{array}{c} \frac{T_0^2(r_\sigma \tau_r^2 - \omega_s^2 \tau_\sigma'^2 r_\sigma \tau_r^2 + k_r l_h \tau_\sigma') - 2T_0 \tau_\sigma' r_\sigma \tau_r^2 + 2\tau_\sigma'^2 r_\sigma \tau_r^2}{2\tau_\sigma'^2 r_\sigma \tau_r^2} \\ \frac{T_0^2(2\omega_s r_\sigma \tau_r - k_r \omega l_h) - 2T_0 \omega_s r_\sigma \tau_r \tau_\sigma'}{2r_\sigma \tau_r \tau_\sigma'} \\ \frac{T_0 l_h (2\tau_\sigma' \tau_r - T_0(\tau_r + \tau_\sigma'))}{2\tau_\sigma' \tau_r^2} \\ \frac{T_0 \omega_s (\tau_\sigma' - T_0)}{T_0^2 - 2T_0 \tau_\sigma' + \tau_\sigma'^2 (2 - T_0^2 \omega_s^2)} \\ \frac{2\tau_\sigma'^2}{T_0^2 l_h \omega_s} \\ \frac{2\tau_r}{T_0 k_r (2\tau_\sigma' \tau_r - T_0(\tau_r + \omega_s \omega \tau_\sigma' \tau_r^2 + \tau_\sigma'))} \\ \frac{2\tau_\sigma'^2 r_\sigma \tau_r^2}{T_0 k_r (T_0 \omega (\tau_r + \tau_\sigma') - 2\omega \tau_\sigma' \tau_r - T_0 \omega_s \tau_\sigma')} \\ \frac{2\tau_\sigma'^2 r_\sigma \tau_r}{T_0^2 (k_r l_h + r_\sigma \tau_\sigma') - 2T_0 r_\sigma \tau_r \tau_\sigma' + 2r_\sigma \tau_r^2 \tau_\sigma'} \\ \frac{2\tau_\sigma' r_\sigma \tau_r^2}{2\tau_\sigma' r_\sigma \tau_r^2} \end{array} \right] \quad (7.30)$$

$$= \left[\begin{array}{c} 0,9873 - 0,0005175 \omega_s^2 \\ -0,03176 \omega_s - 9,776 \cdot 10^{-5} \omega \\ 0,001357 \\ 0,03176 \omega_s \\ 0,9873 - 0,0005175 \omega_s^2 \\ 2,197 \cdot 10^{-5} \omega_s \\ 0,002358 - 0,002303 \omega_s \omega \\ -0,1422 \omega - 3,819 \cdot 10^{-5} \omega_s \\ 0,9995 \end{array} \right]$$

$$\mathbf{B}_d = \left[\begin{array}{cc} \frac{(2\tau_\sigma' - T_0)T_0}{2\tau_\sigma'^2 r_\sigma} & \frac{\omega_s T_0^2}{2\tau_\sigma' r_\sigma} \\ -\frac{\omega_s T_0^2}{2\tau_\sigma' r_\sigma} & \frac{(2\tau_\sigma' - T_0)T_0}{2\tau_\sigma'^2 r_\sigma} \\ \frac{T_0^2 l_h}{2r_\sigma \tau_\sigma' \tau_r} & 0 \end{array} \right] = \left[\begin{array}{cc} 0,1484 & 0,002402 \omega_s \\ -0,002402 \omega_s & 0,1484 \\ 1,020 \cdot 10^{-4} & 0 \end{array} \right] \quad (7.31)$$

Notizen/Notes/笔记

7.2 Experimentelle Ergebnisse/Experimental results/实验结果

Um die Genauigkeit der verschiedenen Prädiktionsalgorithmen zu evaluieren, wurden sie testweise in eine konventionelle, feldorientierte Regelung mit PI-Reglern implementiert und die vorausgerechneten Ergebnisse mit den tatsächlichen im System vorherrschenden Werten verglichen. Die Abb. 7.1 zeigt das Verhalten bei einem Sollwertsprung des drehmomentbildenden Stroms i_{sq} , während die Abb. 7.2 die Reaktionen des Systems auf eine Sollwertänderung des feldbildenden Stromanteils i_{sd} darstellen.

Wie erwartet zeigt in Abb. 7.1(a) die Vorausberechnung von i_{sq} den Wert, den der Strom im *nächsten* Takt erreichen wird. Die Abweichungen zwischen Vorausberechnung und Realität sind relativ gering und beruhen auf parasitären Effekten, die im Modell nicht berücksichtigt worden sind.

In Abb. 7.1(b), wo die Vorausberechnung von i_{sd} dargestellt ist, kommt die Änderung in der Prädiktion von i_{sd} *gleichzeitig* mit dem tatsächlichen Stromanstieg. Auf den ersten Blick ist dies erstaunlich, denn eigentlich sollte die Vorausberechnung ja gerade den Einfluss von u_{sq} auf i_{sd} berücksichtigen. Formel (7.16) zeigt jedoch, dass es bei der Prädiktion mittels einfachem Differenzenquotient *keinen direkten* Durchgriff von u_{sq} auf i_{sd} gibt. Das Potenzreihenverfahren berücksichtigt diesen Einfluss zwar, allerdings mit einem äußerst kleinen Wert (s. Formel (7.31)). Offensichtlich ist zur exakten Modellierung der Kreuzkopplung eine Strecke höherer Ordnung notwendig; eine Strecke zweiter Ordnung ergibt exakt die gleichen Ergebnisse wie die einfache Differenzenbildung. Die Übereinstimmung zwischen Prädiktion und Realität ist jedoch in beiden Fällen ähnlich gut, wie für die Prädiktion von i_{sq} in Abb. 7.1(a).

Abb. 7.2(a) zeigt nun das Verhalten von i_{sq} bei einem Sprung im feldbildenden Strom von $i_{sd} = 0,1$ auf $i_{sd} = 0,35$. Der drehmomentbildende Strom wurde auf $i_{sq} = 0$ geregelt. Da die Veränderung in i_{sd} zu keiner messbaren Beeinflussung von i_{sq} geführt hat, betragen sowohl die tatsächlichen Messwerte als auch die beiden vorausgerechneten Werte konstant 0.

Wie Abb. 7.2(b) zeigt, verhält sich die Vorausberechnung des feldbildenden Stromanteils i_{sd} bei Anlegen der Spannung u_{sd} genauso, wie die Prädiktion von i_{sq} bei Anlegen von u_{sq} (Abb. 7.1(a)). Es werden jeweils die im *nächsten* Taktzyklus gemessenen Werte vorausgerechnet; die Übereinstimmung zwischen Realität und Berechnung mittels Modell ist daher ausreichend gut.

7.2 Experimental results

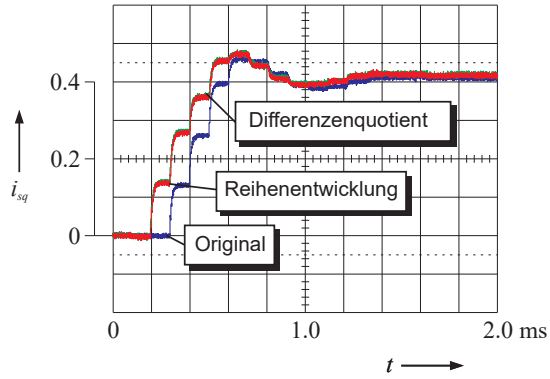
In order to evaluate the accuracy of the different prediction algorithms, they were implemented in a conventional field-oriented control with PI-controllers and the predicted results were compared to the values obtained from the actual system. [Figure 7.1](#) shows the response when a step in the reference value for the torque producing component of the stator current i_{sq} was applied, while [Figure 7.2](#) shows the system response for a reference value change in the flux producing component of the stator current component i_{sd} .

In [Figure 7.1\(a\)](#) the value of the advance calculation for i_{sq} is shown, i. e. the value that will be reached by the current in the *next* sampling cycle. The deviations between the advance calculation and the actual values are relatively small. They are caused by parasitic effects which have not been taken into account when the system model was derived.

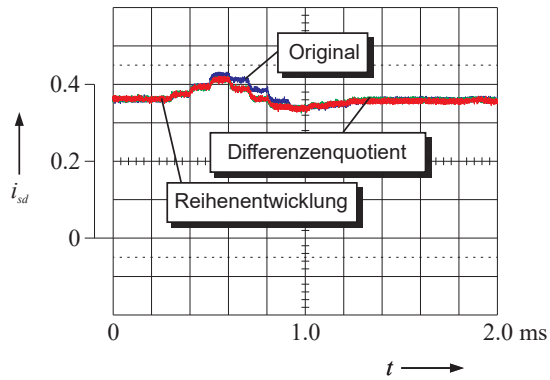
The prediction of i_{sd} is shown in [Figure 7.1\(b\)](#), the change in the predicted values appears at the *same time* as the actual current rise. At first sight this is astonishing since the prediction should consider the direct influence of u_{sq} on i_{sd} . Nevertheless, equation (7.16) shows that there is *no direct* influence of u_{sq} on i_{sd} when the prediction is done with a simple difference quotient. However, the power series method considers this influence, but with an extremely small value (see equation (7.31)). Obviously a system of higher order is necessary for the exact modeling of the cross coupling; a second order system delivers exactly the same results as the simple difference equation. Nevertheless, the correlation between prediction and reality is in both cases as good as for the prediction of i_{sq} in [Figure 7.1\(a\)](#).

[Figure 7.2\(a\)](#) shows the response of i_{sq} when a step in the flux producing component of the stator current from $i_{sd} = 0.1$ to $i_{sd} = 0.35$ was applied. The torque producing component of the stator current was controlled to $i_{sq} = 0$. As the step change in i_{sd} does not lead to a measurable influence on i_{sq} , the actual values as well as both predicted values are constantly 0.

As shown in [Figure 7.2\(b\)](#), the prediction of the flux producing component of the stator current i_{sd} when a voltage u_{sd} is applied behaves in the same way as the prediction of the torque producing component of the stator current i_{sq} when u_{sq} is applied ([Figure 7.1\(a\)](#)). The values that will be measured in the *next* sampling cycle are precalculated; hence, the correlation between the values of the actual system and the values calculated with the model is sufficiently good.



(a)



(b)

7.1: Prädiktion bei Sollwertsprung in i_{sq} : (a) Vorausberechnung von i_{sq} , (b) Vorausberechnung von i_{sd} ;

Prediction with a step change in the reference variable i_{sq} : (a) Advance calculation of i_{sq} , (b) Advance calculation of i_{sd} ;

参比变量 i_{sq} 阶跃变化时的预测: (a) 提前预测 i_{sq} , (b) 提前预测 i_{sd}

7.2 实验结果

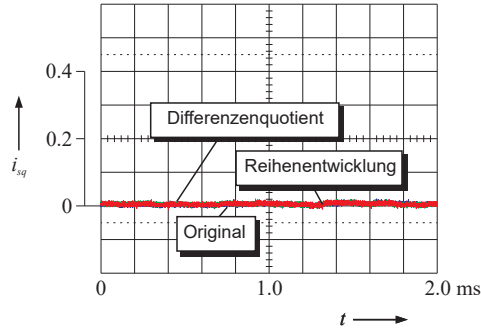
为了评价不同预测算法的精度,需在含有 PI 控制器的传统磁场定向控制方式下实现上述算法,并将系统的实际值与根据模型计算的预测值进行对比分析。图 7.1 给出了定子电流转矩分量 i_{sq} 阶跃变化时的系统响应;图 7.2 给出了定子电流励磁分量 i_{sd} 阶跃变化时的系统响应。

正如所预期的那样,图 7.1(a) 给出了 i_{sq} 的预测值,而实际电流将在下一采样周期到达此预测值。可以看出,预测值与实际值之间的偏差相对较小,而且此偏差取决于并未在模型中考虑的寄生效应。

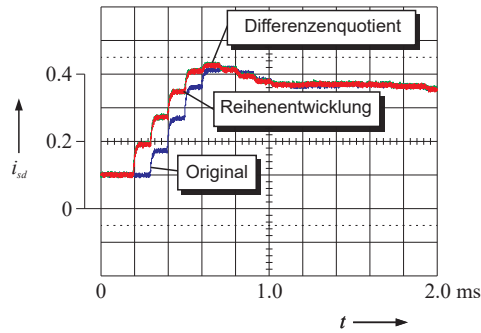
图 7.1(b) 给出了 i_{sd} 的预测值。可以看出, i_{sd} 的预测值与实际电流值在上升阶段同时发生变化。乍一看似乎出乎意料,因为这意味着预测时考虑到了 u_{sq} 对 i_{sd} 的影响。然而,方程 (7.16) 表明,当使用基于简单的微商方法进行预测时, u_{sq} 对 i_{sd} 并无直接影响。虽说基于幂级数的方法可以考虑到上述影响,但其程度却是很微小的,这从方程 (7.31) 中的元素 $0.002402 \omega_s$ 可以看出。显然,为了实现耦合因素的精确建模,则需要更高阶的被控系统,因为二阶的被系统与简单的微分方式所得到的是完全相同的结果。在级数展开式与微商这两种情形下,预测值与实际值之间的关系几乎一致,正如图 7.1(a) 所示。

图 7.2(a) 给出当定子电流励磁分量由 $i_{sd} = 0.1$ 阶跃变化至 $i_{sd} = 0.35$ 时,定子电流转矩分量 i_{sq} 的变化情况。可以看出,定子电流转矩分量被稳定控制在 $i_{sq} = 0$ 。由此可知, i_{sd} 的变化无法对 i_{sq} 产生影响,所以实际的测量值以及基于两种预测方法的预测值均为零。

如图 7.2(b) 所示,定子电流励磁分量 i_{sd} 的预测值在电压 u_{sd} 的作用下所引起的变化趋势与图 7.1(a) 中所示的定子电流转矩分量 i_{sq} 的预测值在电压 u_{sq} 作用下所产生的变化趋势一致。由于两者均提前预测出了在下一周期的测量值,所以实际值与基于模型所计算的预测值的对应关系已经足够精确。



(a)



(b)

7.2: Prädiktion bei Sollwertsprung in i_{sd} : (a) Vorausberechnung von i_{sq} , (b) Vorausberechnung von i_{sd} ;

Prediction with a step change in the reference variable i_{sd} : (a) Advance calculation of i_{sq} , (b) Advance calculation of i_{sd} ;

参比变量 i_{sd} 阶跃变化时的预测: (a) 提前预测 i_{sq} , (b) 提前预测 i_{sq}

Alle Messungen bestätigen, dass die Berechnung des diskreten Maschinenmodells mittels Reihenentwicklung der Matrixexponentialfunktion gegenüber der einfachen Annäherung mittels Differenzenquotient keine Verbesserung des Prädiktionsverhaltens ergibt. Ein Vergleich der sich aus der Reihenentwicklung ergebenden Berechnungsformeln (7.30) und (7.31) mit den entsprechenden Gln. (7.15) und (7.16) des Differenzenquotienten zeigt, dass der Berechnungsaufwand bei einer Reihenentwicklung deutlich höher ist. Daher ist dem in Kap. 7.1.2 erläuterten Verfahren der Modellbildung mit Hilfe des Differenzenquotienten der Vorzug zu geben.

All measurements confirm that the calculation of the time-discrete machine model using the series expansion of the matrix exponential function shows no improvement in the prediction compared to the simple approximation via the difference quotient. A comparison of the equations (7.30) and (7.31), resulting from the series expansion with the corresponding equations (7.15) and (7.16) for the difference quotient, shows that the calculation complexity is clearly much higher if series expansion is used. Hence, the method explained in Chapter 7.1.2, i. e. modeling via the difference quotient should be preferred.

实际测量结果表明，相较于基于微商的简单近似方法，基于矩阵指数函数的级数展开式方法所得的预测效果并无改善。对比上述两种方法所得到的系统矩阵和输入矩阵：级数展开式方法下的方程 (7.30) 和 (7.31)，以及微商方法下的方程 (7.15) 和 (7.16)，可以看出，使用基于级数展开式的方法需要更为强大的计算能力。基于上述分析可知，应优先选择章节 7.1.2 中所述的基于微商的建模方法。

7.3 Modifiziertes Maschinenmodell für GPC/Modified machine model for GPC/ 用于 GPC 的修正电机模型

Im praktischen Einsatz des in Kap. 7.1 aus den Gln. (5.4) und (5.5) hergeleiteten Maschinenmodells zeigt sich, dass die Approximierung der Nichtlinearitäten durch Einführen von ω und ω_s als Parameter nicht durchführbar ist. Die diskrete Systemmatrix \mathbf{A}_d würde sich in jedem Abtastzyklus ändern, weshalb die diversen Polynommatrizen des GPC-Reglers, wie in Kap. 8 näher ausgeführt, jedesmal komplett neu berechnet werden müssten. Da dies u. a. eine Matrixinversion beinhaltet, sprengen die notwendigen Rechenoperationen die zur Verfügung stehende Rechenzeit. Das Verfahren ist daher mit den vorhandenen Ressourcen nicht einsetzbar.

Eine Alternative wäre, die Rückkopplungsabweichung aus dem Modell zu extrahieren und durch zusätzliche Systemeingänge zu ersetzen. Aus den bekannten Werten von ω , ω_s , i_{sd} , i_{sq} und ψ_{rd} würden bei diesem Ansatz drei scheinbare „Eingänge“ $\omega_s i_{sd}$, $\omega_s i_{sq}$ und $\omega \psi_{rd}$ berechnet, die auf das System zurückgeführt werden. Dies funktioniert in der Simulation gut, auch ein passiv mitlaufendes Modell zeigt in der Praxis gute Ergebnisse; soll diese Struktur jedoch für eine Regelung mittels GPC eingesetzt werden, so interpretiert der GPC-Regler die zur Berücksichtigung der Kreuzkopplung und der Gegenspannung eingefügten zusätzlichen pseudo-Eingänge $\omega_s i_{sd}$, $\omega_s i_{sq}$ und $\omega \psi_{rd}$ als echte Streckeneingänge und gibt konsequenterweise auch Stellgrößen für diese aus. Da es sich jedoch nicht um reelle Systemeingänge handelt, können diese Eingänge nicht von außen beeinflusst werden. Das bedeutet, dass ein Modell dieser Art für eine GPC-Regelung *nicht verwendet werden kann*.

Abhilfe schafft der Gedanke, die Rückkopplungen als bekannte Störgrößen zu behandeln und im Modellsatz zu berücksichtigen. Dazu wird die Zustandsraumbeschreibung (5.2) und (5.3) – wie bei Camacho/Bordons erwähnt – um die Störeinflüsse erweitert [20]:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{v} \quad (7.32)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{F}\mathbf{v} \quad (7.33)$$

Hinzugefügt werden die *Eingangsmatrix für Störgrößen* \mathbf{E} und die *Durchgangsmatrix für Störgrößen* \mathbf{F} . Die Vektoren \mathbf{x} , \mathbf{u} , \mathbf{v} und \mathbf{y} werden gemäß dem Maschinenmodell (2.11)–(2.13) definiert, wobei im Gegensatz zu Kap. 7.1 der Ausgangsvektor \mathbf{y} den Fluss ψ_{rd} hier *nicht* enthält. Andernfalls würde der GPC-Regler ψ_{rd} als echten Ausgang der Regelstrecke verstehen und einen Sollwert erwarten.

7.3 Modified machine model for GPC

In practical applications of the machine model derived from the equations (5.4) and (5.5) in Chapter 7.1 it can be seen that the approximation of the nonlinearities by treating ω and ω_s as parameters is not feasible. The discrete state matrix \mathbf{A}_d would change in every sampling cycle and therefore the various polynomial matrices of the GPC controller, as shown in Chapter 8, would also have to be completely calculated again in every sampling cycle. Since this contains, among other things, a matrix inversion, the necessary arithmetic operations exceed the available computation time. Hence, this method is not feasible with the available resources.

An alternative would be to extract the feedback coupling branches from the model and to substitute them with additional system inputs. In this approach from the known values of ω , ω_s , i_{sd} , i_{sq} and ψ_{rd} , three virtual “inputs” $\omega_s i_{sd}$, $\omega_s i_{sq}$ and $\omega \psi_{rd}$ would be calculated, which would then be fed back to the system. In simulations, this works well, in practice, a passively running model also shows good results; however, if this structure should be used for control via GPC, the GPC controller then interprets the additional pseudo inputs $\omega_s i_{sd}$, $\omega_s i_{sq}$ and $\omega \psi_{rd}$, which were added in order to consider the cross coupling and the back EMF, as real plant inputs and consequently the GPC controller delivers values for these actuating variables. However, since these are no real actuating variables, they cannot be influenced by the controller. Thus, a model of this kind *cannot be used* for GPC control.

A solution is to treat the feedback couplings as known disturbance variables and to consider them already in the modeling. Therefore, the state space model (5.2) and (5.3)—as described by Camacho/Bordons—is expanded for the consideration of the disturbance inputs [20]

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{v} \quad (7.32)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{F}\mathbf{v} \quad (7.33)$$

The *input matrix for disturbance variables* \mathbf{E} and the *feedforward matrix for disturbance variables* \mathbf{F} are added. The vectors \mathbf{x} , \mathbf{u} , \mathbf{v} and \mathbf{y} are defined according to the machine model (2.11)–(2.13), in which in contrast to Chapter 7.1 the output vector \mathbf{y} does *not* contain the flux ψ_{rd} . Otherwise the GPC controller would treat ψ_{rd} as a real output of the plant and expect a reference value for this variable.

Da aber der Fluss nicht geregelt werden soll, sondern lediglich ein konstanter flussbildender Strom i_{sd} vorgegeben wird, ist dies überflüssig. Die Definitionen der einzelnen Vektoren lauten daher:

$$\mathbf{x} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \omega_s i_{sd} \\ \omega_s i_{sq} \\ \omega \psi_{rd} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}$$

wodurch sich die Zustandsraumbeschreibung

$$\begin{aligned} \frac{d}{d\tau} \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} &= \begin{bmatrix} -\frac{1}{\tau_{\sigma'}} & 0 & \frac{k_r}{r_{\sigma} \tau_r \tau_{\sigma'}} \\ 0 & -\frac{1}{\tau_{\sigma'}} & 0 \\ \frac{l_h}{\tau_r} & 0 & -\frac{1}{\tau_r} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1}{r_{\sigma} \tau_{\sigma'}} & 0 \\ 0 & \frac{1}{r_{\sigma} \tau_{\sigma'}} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \end{aligned} \quad (7.34)$$

$$\begin{aligned} &+ \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -\frac{k_r}{r_{\sigma} \tau_{\sigma'}} \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \omega_s i_{sd} \\ \omega_s i_{sq} \\ \omega \psi_{rd} \end{bmatrix} \\ \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \omega_s i_{sd} \\ \omega_s i_{sq} \\ \omega \psi_{rd} \end{bmatrix} \end{aligned} \quad (7.35)$$

ergibt. Die Matrizen **A**, **B**, **C** und **D**, sowie **E** und **F** ergeben sich damit zu:

Since the flux should not be controlled and only a constant flux-producing current i_{sd} will be given, this is redundant. The definitions of the individual vectors are as follows:

$$\mathbf{x} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \omega_s i_{sd} \\ \omega_s i_{sq} \\ \omega \psi_{rd} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}$$

whereby the state space representation is given by equations (7.34) and (7.35). So the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} as well as \mathbf{E} and \mathbf{F} result to:

7.3 用于 GPC 的修正电机模型 在实际应用章节 7.1 所推导的形如方程 (5.4) 和 (5.5) 形式的离散电机模型时, 通过引入 ω 和 ω_s 作为参数以实现所谓的非线性近似被证明是行不通的。由于离散的系统矩阵 \mathbf{A}_d 在每一个采样周期均发生变化, 所以 GPC 控制器的不同多项式矩阵也需要在每一个采样周期进行重新计算, 这一点将在第 8 章详细讲述。由于上述计算过程包含复杂的矩阵求逆, 所以必要的计算过程已经超出了系统所能提供的运算时间。在现有计算能力的条件下, 仍然无法实现这一方法。

另外一种可选的方案是从模型中提取出反馈通路, 并使用额外的系统输入进行替代。根据已知的 ω , ω_s , i_{sd} , i_{sq} 和 ψ_{rd} , 可以计算出三个设想的虚拟输入变量 $\omega_s i_{sd}$, $\omega_s i_{sq}$ 和 $\omega \psi_{rd}$, 并将其反馈至系统。上述方案在仿真时可以良好运行, 并且一个被动的正序模型在实际中也可得到良好的结果。假使在基于 GPC 的控制方法中采用上述结构, 那么, GPC 控制器将把考虑交叉耦合和反向电动势因素所引入的伪输入变量 $\omega_s i_{sd}$, $\omega_s i_{sq}$ 和 $\omega \psi_{rd}$ 看作被控系统的实际输入, 从而使得执行机构的输出变量也是上述伪输入变量。由于此时讨论的并非真实的系统输入变量, 所以它们并不受外界因素的影响, 这也意味着上述方案无法应用于 GPC 控制。

可行的解决方案是将反馈看作为已知的扰动变量处理, 并在相应的模型中将其考虑在内。为此, 需将由学者 Camacho/Bordons 所描述的状态空间描述方程 (5.2) 和 (5.3) 进一步扩展为 [20]:

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{v} \quad (7.32)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \mathbf{F}\mathbf{v} \quad (7.33)$$

所增加的是扰动变量输入矩阵 \mathbf{E} 和扰动变量输入输出矩阵 \mathbf{F} 。向量 \mathbf{x} , \mathbf{u} , \mathbf{v} 和 \mathbf{y} 可根据电机模型 (2.11) – (2.13) 进行定义。通过与章节 7.1 相关内容进行对比, 可知此时的输出向量 \mathbf{y} 中不包含磁通 ψ_{rd} 。否则的话, GPC 控制器将会把 ψ_{rd} 看作为被控系统的真实输出, 并期望一个 ψ_{rd} 的参考给定值。

由于磁通实际上是不被控制的, 只需给定一个恒定的励磁电流分量 i_{rd} 即可, 所以, 将 ψ_{rd} 看作被控变量是多此一举的。单一向量可定义为:

$$\mathbf{x} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \omega_s i_{sd} \\ \omega_s i_{sq} \\ \omega \psi_{rd} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}$$

从而可得到状态空间描述方程 (7.34) 和 (7.35)。矩阵 \mathbf{A} , \mathbf{B} , \mathbf{C} 和 \mathbf{D} 以及 \mathbf{E} 和 \mathbf{F} 为:

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{\tau_{\sigma'}} & 0 & \frac{k_r}{r_{\sigma}\tau_r\tau_{\sigma'}} \\ 0 & -\frac{1}{\tau_{\sigma'}} & 0 \\ \frac{l_h}{\tau_r} & 0 & -\frac{1}{\tau_r} \end{bmatrix} = \begin{bmatrix} -0,3964 & 0 & 0,07380 \\ 0 & -0,3964 & 0 \\ 0,04245 & 0 & -0,01658 \end{bmatrix} \quad (7.36)$$

$$\mathbf{B} = \begin{bmatrix} \frac{1}{r_{\sigma}\tau_{\sigma'}} & 0 \\ 0 & 1 \\ 0 & \frac{1}{r_{\sigma}\tau_{\sigma'}} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 4,641 & 0 \\ 0 & 4,641 \\ 0 & 0 \end{bmatrix} \quad (7.37)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (7.38)$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (7.39)$$

$$\mathbf{E} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -\frac{k_r}{r_{\sigma}\tau_{\sigma'}} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -4,450 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.40)$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.41)$$

Da für die Regelung der Strecke mittels eines MPC-Reglers ein zeitdiskretes Modell benötigt wird, müssen die oben hergeleiteten Gleichungen in eine zeitdiskrete Zustandsraumbeschreibung

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) + \mathbf{E}_d\mathbf{v}(k) \quad (7.42)$$

$$\mathbf{y}(k) = \mathbf{C}_d\mathbf{x}(k) + \mathbf{D}_d\mathbf{u}(k) + \mathbf{F}_d\mathbf{v}(k) \quad (7.43)$$

überführt werden. Die Matrizen \mathbf{E}_d und \mathbf{F}_d stellen dabei die Eingangs- bzw. Durchgangsmatrix der zeitdiskreten Störfunktion $\mathbf{v}(k)$ dar.

Da sämtliche Systemmatrizen nun keine symbolischen Parameter, sondern nur noch Zahlenwerte enthalten, ist eine analytische Berechnung von \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d , \mathbf{D}_d , \mathbf{E}_d und \mathbf{F}_d nicht mehr notwendig. Ein mit Hilfe von Maple[®] selbstprogrammiertes Verfahren zur wahlweisen Bestimmung der Matrizen der zeitdiskreten Zustandsraumbeschreibung über Differenzenquotient (Kap. 7.1.2), Laplace-Transformation (Kap. 7.1.3) oder Potenzreihe (Kap. 7.1.4) ist daher überflüssig. Die Berechnung der Matrizen der Gln. (7.42) und (7.43) kann einfach mit der Software MATLAB[®] erfolgen. Es ergeben sich:

See equations (7.36) and (7.41).

Since a time-discrete model is necessary for the control of the plant with an MPC controller, the equations derived above have to be transferred into a time-discrete state space representation:

$$\mathbf{x}(k+1) = \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d \mathbf{u}(k) + \mathbf{E}_d \mathbf{v}(k) \quad (7.42)$$

$$\mathbf{y}(k) = \mathbf{C}_d \mathbf{x}(k) + \mathbf{D}_d \mathbf{u}(k) + \mathbf{F}_d \mathbf{v}(k) \quad (7.43)$$

The matrices \mathbf{E}_d and \mathbf{F}_d thereby represent the input and feedforward matrix for the time-discrete disturbance function $\mathbf{v}(k)$.

As all system matrices now contain no symbolic parameters, but only numerical values, an analytic calculation of \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d , \mathbf{D}_d , \mathbf{E}_d and \mathbf{F}_d is not required anymore. Therefore, a method programmed with the help of Maple[®] in order to determine the matrices of the time-discrete state space representation via difference quotient (Chapter 7.1.2), Laplace transformation (Chapter 7.1.3) or power series expansion (Chapter 7.1.4) is not necessary. The calculation of the matrices of the equations (7.42) and (7.43) can be easily done with MATLAB[®], resulting in the following values:

见方程 (7.36) 和 (7.41)。

由于基于 MPC 控制器实现被控系统的控制需要一个时间离散的模式，所以必须将上述推导出的方程转化为时间离散的状态空间描述形式：

$$\mathbf{x}(k+1) = \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d \mathbf{u}(k) + \mathbf{E}_d \mathbf{v}(k) \quad (7.42)$$

$$\mathbf{y}(k) = \mathbf{C}_d \mathbf{x}(k) + \mathbf{D}_d \mathbf{u}(k) + \mathbf{F}_d \mathbf{v}(k) \quad (7.43)$$

矩阵 \mathbf{E}_d 和 \mathbf{F}_d 分别表示时间离散扰动函数 $\mathbf{v}(k)$ 的输入矩阵和直接输入输出矩阵。

由于系统的所有矩阵元素并不包含符号参数，并且全部为实数值，所以不必使用解析的方式求解矩阵 \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d , \mathbf{D}_d , \mathbf{E}_d 和 \mathbf{F}_d 。借助于数学软件 Maple[®] 编写程序来实现章节 7.1.2 所述的微商，章节 7.1.3 所述的拉普拉斯变换以及章节 7.1.4 所述的幂级数，并由此确定时间离散状态空间描述的矩阵则明显是多余的。状态方程 (7.42) 和输出方程 (7.43) 中相关矩阵的计算可轻易地借助于 MATLAB[®] 实现，计算结果如下：

$$\mathbf{A}_d = \begin{bmatrix} 0,9873 & 0 & 0,002358 \\ 0 & 0,9873 & 0 \\ 0,001357 & 0 & 0,9995 \end{bmatrix}$$

$$\mathbf{B}_d = \begin{bmatrix} 0,1484 & 0 \\ 0 & 0,1484 \\ 0,0001015 & 0 \end{bmatrix}$$

$$\mathbf{C}_d = \mathbf{C} \quad \text{unverändert}$$

$$\mathbf{D}_d = \mathbf{D} \quad \text{unverändert}$$

$$\mathbf{E}_d = \begin{bmatrix} 0 & 0,03197 & 0 \\ -0,03197 & 0 & -0,1423 \\ 0 & 2,187 \cdot 10^{-5} & 0 \end{bmatrix}$$

$$\mathbf{F}_d = \mathbf{F} \quad \text{unverändert}$$

Es fällt auf, dass der Wert von $\mathbf{E}_{d3,2} = 2,187 \cdot 10^{-5}$ beträgt; erwartet wird jedoch $\mathbf{E}_{d3,2} = 0$. Offensichtlich führen interne Rundungsfehler von MATLAB[®] zu dieser Abweichung. Die Differenz ist relativ gering, kann aber bei der Implementation des Modells in eine Regelung selbstverständlich korrigiert werden.

$$\begin{aligned}
\mathbf{A}_d &= \begin{bmatrix} 0.9873 & 0 & 0.002358 \\ 0 & 0.9873 & 0 \\ 0.001357 & 0 & 0.9995 \end{bmatrix} \\
\mathbf{B}_d &= \begin{bmatrix} 0.1484 & 0 \\ 0 & 0.1484 \\ 0.0001015 & 0 \end{bmatrix} \\
\mathbf{C}_d &= \mathbf{C} \quad \text{unchanged} \\
\mathbf{D}_d &= \mathbf{D} \quad \text{unchanged} \\
\mathbf{E}_d &= \begin{bmatrix} 0 & 0.03197 & 0 \\ -0.03197 & 0 & -0.1423 \\ 0 & 2.187 \cdot 10^{-5} & 0 \end{bmatrix} \\
\mathbf{F}_d &= \mathbf{F} \quad \text{unchanged}
\end{aligned}$$

It is remarkable that $\mathbf{E}_{d3,2} = 2.187 \cdot 10^{-5}$, even though it is expected to be equal to zero: $\mathbf{E}_{d3,2} = 0$. Obviously internal rounding errors of MATLAB[®] lead to this deviation. The difference is relatively small, however, it can of course be corrected when the model is integrated into a control structure.

$$\begin{aligned}
\mathbf{A}_d &= \begin{bmatrix} 0.9873 & 0 & 0.002358 \\ 0 & 0.9873 & 0 \\ 0.001357 & 0 & 0.9995 \end{bmatrix} \\
\mathbf{B}_d &= \begin{bmatrix} 0.1484 & 0 \\ 0 & 0.1484 \\ 0.0001015 & 0 \end{bmatrix} \\
\mathbf{C}_d &= \mathbf{C} \quad \text{无变化} \\
\mathbf{D}_d &= \mathbf{D} \quad \text{无变化} \\
\mathbf{E}_d &= \begin{bmatrix} 0 & 0.03197 & 0 \\ -0.03197 & 0 & -0.1423 \\ 0 & 2.187 \cdot 10^{-5} & 0 \end{bmatrix} \\
\mathbf{F}_d &= \mathbf{F} \quad \text{无变化}
\end{aligned}$$

值得注意的是, $\mathbf{E}_{d3,2} = 2.187 \cdot 10^{-5}$, 然而所期望的值为零。显然是由于 MATLAB[®] 内部的舍入误差所引起的上述偏差。可以看出, 上述偏差相对较小, 而且在控制系统实现上述模型时是可以对其进行修正的。

8 Multivariable GPC-Regelung/Control/多 变量广义预测控制

Da MPC-Regelungen vom Prinzip her bereits als Mehrgrößenregler angelegt sind, ist die Erweiterung einer SISO-GPC-Regelung in ein multivariablen Verfahren kein allzu großer Aufwand. Die Herleitung entspricht daher fast exakt der in Kap. 6.1.1 für ein SISO-System vorgestellten, allerdings werden bedingt durch die höhere Dimension Vektoren zu Matrizen und Polynome zu Polynommatrizen. Dabei ist zu beachten, dass für Polynommatrizen und Matrizenpolynome besondere mathematische Gesetzmäßigkeiten gelten. Gute Zusammenstellungen dieser Regeln – wenn auch in englischer Sprache – finden sich bei Goodwin/Sin [44] und auf den Webseiten der Firma PolyX [125]. Anhang A gibt einen Überblick über einige der wichtigsten Rechenregeln. Weiterführende Informationen über Matrixtheorie können den Büchern von Gantmacher [39], Lancaster/Tismenetsky [75] sowie Zurmühl/Falk [124] entnommen werden.

8.1 „Klassisches“ MIMO-GPC/“Classical” MIMO-GPC/经典 MIMO-GPC

Beim „klassischen“ mehrdimensionalen GPC handelt es sich um eine einfache Erweiterung des eindimensionalen Verfahrens auf Mehrgrößenregelung.

8.1.1 Bestimmung der Übertragungsfunktion/Determination of the transfer function/传递函数的确定

Bevor eine auf einem übertragungsfunktionsbasierten Modell fußende MPC-Regelung angesetzt werden kann, muss die Zustandsraumbeschreibung (Gln. (5.4) und (5.5)) in eine Übertragungsfunktion bzw. matrix umgewandelt werden. Bei einer Eingrößenregelung entsteht bei der Umwandlung eine zeitdiskrete Übertragungsfunktion der Art

$$G_d(z^{-1}) = g_{d0} + g_{d1}z^{-1} + g_{d2}z^{-2} + \dots + g_{dn}z^{-n} = \frac{Y(z)}{U(z)} \quad (8.1)$$

Liegt hingegen eine Mehrgrößenregelung vor, so ergibt die Division $\mathbf{Y}(z)/\mathbf{U}(z)$ ein Matrizenpolynom, d. h. ein Polynom, dessen Koeffizienten reelle Matrizen sind.

$$\mathbf{G}_d(z^{-1}) = \mathbf{G}_{d0} + \mathbf{G}_{d1}z^{-1} + \mathbf{G}_{d2}z^{-2} + \dots + \mathbf{G}_{dn}z^{-n} = \frac{\mathbf{Y}(z)}{\mathbf{U}(z)} \quad (8.2)$$

As MPC controllers are in principle already designed as multidimensional controllers, the extension of a SISO-GPC to a multivariable method does not involve too much complexity and costs. Hence, the derivation nearly exactly corresponds to the one presented in Chapter 6.1.1 for a SISO system, however, due to the higher dimensions, vectors become matrices and polynomials become polynomial matrices. Besides, it has to be noted that special mathematical rules apply for polynomial matrices and matrix polynomials. Good summaries of these rules can be found in the publication by Goodwin/Sin [44] and on the website of the company PolyX [125]. Appendix A gives a good overview about some of the most important mathematical rules. Further information about matrix theory can be taken from the books by Gantmacher [39], Lancaster/Tismenetsky [75] and Zurmühl/Falk [124].

8.1 “Classical” MIMO-GPC

The “classical” multidimensional GPC is a simple extension of the uni-dimensional method to a multidimensional one.

8.1.1 Determination of the transfer function

Before an MPC based on a transfer function based model can be designed, the state space representation (equations (5.4) and (5.5)) must be converted into a transfer function and/or matrix. If the transformation is done for a uni-dimensional control, a time-discrete transfer function of the form (8.1) results. However, for a multidimensional control the division $\mathbf{Y}(z)/\mathbf{U}(z)$ results into a matrix polynomial, i. e. a polynomial whose coefficients are real matrices, see equation (8.2).

从原理上来看，模型预测控制通常被设计为多变量控制器，而且将单输入单输出形式的 GPC 控制扩展至多变量形式也并非难事。推导过程几乎与章节 6.1.1 所介绍的 SISO 系统完全一致。显然，此时由于更高的维度，使得矢量变为矩阵，多项式变为多项式矩阵。值得注意的是，对于多项式矩阵和矩阵多项式来说，它们需满足相应的数学运算规则。上述规则的详细总结可参考学者 Goodwin/Sin 的著作 [44]，以及 PolyX 公司的官方网站 [125]。附录 A 对一些重要的运算规则进行了概述，关于矩阵理论更为详细的信息可参考相关学者的著作，如 Gantmacher [39]，Lancaster/Tismenetsky [75] 和 Zurmühl/Falk [124]。

8.1 经典 MIMO-GPC

对于经典多维广义预测控制来说，它表示将简单的单维方法扩展至多变量控制。

8.1.1 传递函数的确定

在应用一个基于传递函数模型的 MPC 控制方法之前，必须将状态空间描述 (方程 (5.4) 和 (5.5)) 转换为传递函数或矩阵的形式。对于单变量控制来说，转换后得到的是一个具有下述形式的时间离散传递函数，见方程 (8.1)。而对于多变量控制来说， $\mathbf{Y}(z)/\mathbf{U}(z)$ 表示的是矩阵多项式，也就是说多项式的系数全部为实数矩阵，见方程 (8.2)。

Da für die einzelnen Elemente von $\mathbf{Y}(z)$ und $\mathbf{U}(z)$ gilt

$$\begin{aligned} \mathbf{y}(t) &: \text{Vektor}(n \times 1) && \text{mit } n = \text{Anzahl der Systemausgänge} \\ \mathbf{u}(t) &: \text{Vektor}(m \times 1) && \text{mit } m = \text{Anzahl der Systemeingänge} \end{aligned}$$

gilt auch für die zeitdiskrete Übertragungsmatrix $\mathbf{G}_d(z^{-1})$

$$\mathbf{G}_d(z^{-1}) : \text{Matrix}(n \times m) \quad (\text{Zeilen} \times \text{Spalten})$$

Matrizenpolynome lassen sich in Polynommatrizen und umgekehrt überführen [39]; Polynommatrizen sind Matrizen, deren Elemente Polynome sind. Betrachten wir dazu die Polynommatrix $\mathbf{P}(z^{-1})$:

$$\mathbf{P}(z^{-1}) = \|\|P_{j,i}(z^{-1})\|\|^{n,m} = \|p_{j,i_0} + p_{j,i_1}z^{-1} + \dots + p_{j,i_k}z^{-k}\|^{n,m} \quad (8.3)$$

Die Polynommatrix $\mathbf{P}(z^{-1})$ können wir als Polynom in z schreiben, wobei die Koeffizienten reelle Matrizen sind:

$$\mathbf{P}(z^{-1}) = \mathbf{P}_0 + \mathbf{P}_1z^{-1} + \dots + \mathbf{P}_kz^{-k} \quad (8.4)$$

mit

$$\mathbf{P}_x = \|p_{j,i_x}\|^{n,m}$$

Somit läßt sich $\mathbf{P}(z^{-1})$ von einer Polynommatrix in ein Matrizenpolynom und umgekehrt überführen.

Zur Berechnung von $\mathbf{G}_d(z^{-1})$ geht man von der Zustandsraumbeschreibung aus Gln. (5.4) und (5.5):

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \quad (5.4)$$

$$\mathbf{y}(k) = \mathbf{C}_d\mathbf{x}(k) + \mathbf{D}_d\mathbf{u}(k) \quad (5.5)$$

Obige Gleichungen werden nun in den Z-Bereich transformiert:

$$z\mathbf{X}(z) = \mathbf{A}_d\mathbf{X}(z) + \mathbf{B}_d\mathbf{U}(z) \quad (8.5)$$

$$\mathbf{Y}(z) = \mathbf{C}_d\mathbf{X}(z) + \mathbf{D}_d\mathbf{U}(z) \quad (8.6)$$

Durch Umformen erhält man aus (8.5):

$$\mathbf{X}(z) = (z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d\mathbf{U}(z) \quad (8.7)$$

Einsetzen in (8.6) ergibt

$$\mathbf{Y}(z) = (\mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d)\mathbf{U}(z) \quad (8.8)$$

Damit läßt sich die Übertragungsfunktion $\mathbf{G}_d(z^{-1})$ zu

$$\mathbf{G}_d(z^{-1}) = \mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{B}_d + \mathbf{D}_d \quad (8.9)$$

bestimmen.

For the individual elements of $\mathbf{Y}(z)$ and $\mathbf{U}(z)$ the following applies:

$$\begin{aligned} \mathbf{y}(t) &: \text{Vector}(n \times 1) && \text{with } n = \text{number of system outputs} \\ \mathbf{u}(t) &: \text{Vector}(m \times 1) && \text{with } m = \text{number of system inputs} \end{aligned}$$

Because of this, for the time-discrete transfer function $\mathbf{G}_d(z^{-1})$, the following applies, too:

$$\mathbf{G}_d(z^{-1}) : \text{Matrix}(n \times m) \quad (\text{rows} \times \text{columns})$$

Matrix polynomials can be transferred into polynomial matrices and vice versa [39]; polynomial matrices are matrices whose elements are polynomials. Considering the polynomial matrix $\mathbf{P}(z^{-1})$, see equation (8.3). The polynomial matrix $\mathbf{P}(z^{-1})$ can be written as a polynomial in z with the coefficients being real matrices, see equation (8.4). Therefore, a polynomial matrix $\mathbf{P}(z^{-1})$ can be transferred into a matrix polynomial and vice versa.

For the calculation of $\mathbf{G}_d(z^{-1})$, the state space representation (equations (5.4) and (5.5)) is considered first, see equations (5.4) and (5.5). These equations are then transformed into the Z-domain, see equation (8.5) and (8.6). By rearranging (8.5), the following equation (8.7) can be obtained. Replacing it into (8.6) results in equation (8.8). Thus, the transfer function $\mathbf{G}_d(z^{-1})$ can be written as equation (8.9).

由于 $\mathbf{Y}(z)$ 和 $\mathbf{U}(z)$ 的单一元素满足:

$$\begin{aligned} \mathbf{y}(t) &: \text{向量}(n \times 1) && \text{其中, } n = \text{系统输出的数量} \\ \mathbf{u}(t) &: \text{向量}(m \times 1) && \text{其中, } m = \text{系统输入的数量} \end{aligned}$$

时间离散传递矩阵 $\mathbf{G}_d(z^{-1})$ 满足:

$$\mathbf{G}_d(z^{-1}) : \text{矩阵}(n \times m) \quad (\text{行} \times \text{列})$$

矩阵多项式与多项式矩阵是可以相互转换的 [39]。多项式矩阵表示的是元素为多项式的矩阵, 考虑下述多项式矩阵 $\mathbf{P}(z^{-1})$, 见方程 (8.3)。多项式矩阵 $\mathbf{P}(z^{-1})$ 也可以写为 z 的多项式, 其系数分别为实数矩阵, 见方程 (8.4)。从而, $\mathbf{P}(z^{-1})$ 从多项式矩阵的形式转化为了矩阵多项式, 反之亦然。

为了计算 $\mathbf{G}_d(z^{-1})$, 需要使用到前述章节中介绍的状态空间描述, 见方程 (5.4) 和 (5.5)。然后, 将上述方程转化至 Z 域, 见方程 (8.5) 和 (8.6)。对方程 (8.5) 进行适当变换, 可得方程 (8.8)。将其代入至方程 (8.6), 可得方程 (8.8)。从而, 可确定传递函数 $\mathbf{G}_d(z^{-1})$ 的表达式 (8.9)。

8.1.2 Berechnung der Systemmatrizen/Calculation of the system matrices/系统矩阵的计算

Bei der Eingrößenregelung lassen sich die Systempolynome $A(z^{-1})$ und $B(z^{-1})$, auf denen das CARIMA-Modell basiert, aus der gebrochenen Übertragungsfunktion $G_d(z^{-1})$ ablesen. Wie in Gl. (6.1) dargestellt, bildet $A(z^{-1})$ den Nenner und $B(z^{-1})$ den Zähler von $G_d(z^{-1})$.

Liegt jedoch ein Mehrgrößensystem vor, handelt es sich bei $\mathbf{A}(z^{-1})$ und $\mathbf{B}(z^{-1})$ nun um *Systempolynommatrizen*. Da $\mathbf{G}_d(z^{-1})$ ebenfalls eine Polynommatrix ist, können $\mathbf{A}(z^{-1})$ und $\mathbf{B}(z^{-1})$ nicht mehr einfach durch Ablesen von Nenner und Zähler von $\mathbf{G}_d(z^{-1})$ bestimmt werden. Ein einfaches Verfahren zur Ermittlung von $\mathbf{A}(z^{-1})$ und $\mathbf{B}(z^{-1})$ wird von Camacho in [20, Kap. 6.2.1] vorgestellt. Desweiteren wird dort ein auch bei Goodwin und Sin [44, Kap. 2.3.5] beschriebenes, mathematisch aufwändigeres Verfahren erläutert, welches links koprim¹ Ergebnisse für $\mathbf{A}(z^{-1})$ und $\mathbf{B}(z^{-1})$ liefert. Dies ist für die Implementierung eines multivariablen GPC-Reglers zwar nicht erforderlich, allerdings ergeben sich unter Verwendung von links koprimen Systemmatrizen effizientere Algorithmen mit kürzerer Rechenzeit. Dieses Verfahren wird im Anhang G näher erläutert; eine sehr gute und ausführliche Beschreibung findet sich auch bei Geering [41], weshalb an dieser Stelle darauf verzichtet werden kann. Der Einfachheit halber wird daher im Folgenden nur das einfachere Verfahren von Camacho [20] behandelt.

Bekanntlich muss gelten

$$\mathbf{G}_d(z^{-1}) = \mathbf{A}(z^{-1})^{-1} \mathbf{B}(z^{-1}) z^{-1} \quad (8.10)$$

Der einfachste Weg besteht nun darin, dass man $\mathbf{A}(z^{-1})$ als Diagonalmatrix wählt, deren Diagonalelemente die kleinsten gemeinsamen Vielfachen der Nenner der jeweiligen Zeilen von $\mathbf{G}_d(z^{-1})$ sind. Die Polynommatrix $\mathbf{B}(z^{-1})$ wird dann einfach mit

$$\mathbf{B}(z^{-1}) = \mathbf{A}(z^{-1}) \mathbf{G}_d(z^{-1}) z \quad (8.11)$$

berechnet.

8.1.2 Calculation of the system matrices

In the case of a uni-dimensional control, the system polynomials $A(z^{-1})$ and $B(z^{-1})$, on which the CARIMA model is based can easily be taken from the transfer function $G_d(z^{-1})$. As shown in equation (6.1), $A(z^{-1})$ is the denominator and $B(z^{-1})$ is the numerator of $G_d(z^{-1})$.

However, if it is a multidimensional system, then $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ are system *polynomial matrices*. Since $\mathbf{G}_d(z^{-1})$ is also a polynomial matrix, $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$

¹ Zur Erläuterung der mathematischen Fachbegriffe für Polynommatrizen siehe Anhang A.

cannot simply be determined by taking the denominator and numerator of $\mathbf{G}_d(z^{-1})$. A simple method for the determination of $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$ is presented by Camacho in [20, Chapter 6.2.1]. Furthermore, a mathematically more complex method which is also described by Goodwin and Sin [44, Chapter 2.3.5] is explained which delivers left coprime¹ results for $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$. Although this is not necessary for the implementation of a multivariable GPC controller, using left coprime system matrices results in more efficient algorithms leading to shorter computation times. This procedure is explained in detail in appendix G; a very good and detailed description can also be found in the publication by Geering [41], this is the reason why further explanations are omitted at this point. For the sake of simplicity, only the simpler method by Camacho [20] is treated in the following.

It is well-known that equation (8.10). Now the most simple way is to assume that $\mathbf{A}(z^{-1})$ is a diagonal matrix whose diagonal elements are the smallest common multiples of the denominators of the corresponding rows of $\mathbf{G}_d(z^{-1})$. Then, the polynomial matrix $\mathbf{B}(z^{-1})$ can simply be calculated with equation (8.11).

8.1.2 系统矩阵的计算

对于单变量控制来说，基于 CARIMA 模型的系统多项式 $A(z^{-1})$ 和 $B(z^{-1})$ 可轻易地通过写为分式形式的传递函数 $G_d(z^{-1})$ 直接得到，如方程 (6.1) 所示， $A(z^{-1})$ 为 $G_d(z^{-1})$ 的分母， $B(z^{-1})$ 为 $G_d(z^{-1})$ 的分子。

但是对于多变量控制来说， $\mathbf{A}(z^{-1})$ 和 $\mathbf{B}(z^{-1})$ 所代表的是系统多项式矩阵。由于 $\mathbf{G}_d(z^{-1})$ 也是一个多项式矩阵，使得 $\mathbf{A}(z^{-1})$ 和 $\mathbf{B}(z^{-1})$ 无法通过直接读取 $\mathbf{G}_d(z^{-1})$ 的分母和分子的形式确定。学者 Camacho 在其著作 [20, 章节 6.2.1] 中介绍了一种确定 $\mathbf{A}(z^{-1})$ 和 $\mathbf{B}(z^{-1})$ 的简单方法。更进一步，学者 Goodwin 和 Sin 在其著作 [44, 章节 2.3.5] 中描述了一种数学上更为复杂的方法，基于此可以得到左互素¹ 的 $\mathbf{A}(z^{-1})$ 和 $\mathbf{B}(z^{-1})$ 。尽管多变量 GPC 控制器的实现并未做出互素的强制要求，但使用左互素系统矩阵可以得到更为有效的算法，从而可以节约计算时间。附录 G 对上述方法做了进一步的阐述，更为详尽且准确的描述可参考学者 Geering 的著作 [41]，此处不再赘述。为简化起见，接下来仅介绍学者 Camacho 所提出的简单方法 [20]。

已知且必须满足的是方程 (8.10)。最简单的方式是选择 $\mathbf{A}(z^{-1})$ 作为对角矩阵，其对角线上的元素是 $\mathbf{G}_d(z^{-1})$ 对应行的分母的最小公倍数。从而，多项式矩阵 $\mathbf{B}(z^{-1})$ 可根据方程 (8.11) 计算得到。

¹ See appendix A for an explanation of the mathematical terms for polynomial matrices.

¹ 有关多项式矩阵的数专业术语的阐述可参见附录 A。

8.1.3 Mathematische Herleitung/Mathematical derivation/数学推导

Das CARIMA-Modell

Analog zum SISO-System wird hier für das CARIMA-Modell angesetzt

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = \mathbf{B}(z^{-1})\mathbf{u}(t-1) + \mathbf{C}(z^{-1})\frac{\boldsymbol{\xi}(t)}{\Delta} \quad (8.12)$$

wobei nun gilt:

$$\begin{aligned} \mathbf{A}(z^{-1}) &= \mathbf{I} + \mathbf{A}_1 z^{-1} + \mathbf{A}_2 z^{-2} + \dots + \mathbf{A}_{na} z^{-na} \\ \mathbf{B}(z^{-1}) &= \mathbf{B}_0 + \mathbf{B}_1 z^{-1} + \mathbf{B}_2 z^{-2} + \dots + \mathbf{B}_{nb} z^{-nb} \\ \mathbf{C}(z^{-1}) &= \mathbf{I} + \mathbf{C}_1 z^{-1} + \mathbf{C}_2 z^{-2} + \dots + \mathbf{C}_{nc} z^{-nc} \\ \Delta &= 1 - z^{-1} \end{aligned}$$

Die Dimensionen der einzelnen Matrizen betragen:

$$\begin{aligned} \mathbf{A}(z^{-1}) &: \text{Matrix}(n \times n) \\ \mathbf{B}(z^{-1}) &: \text{Matrix}(n \times m) \\ \mathbf{C}(z^{-1}) &: \text{Matrix}(n \times n) \\ \boldsymbol{\xi}(t) &: \text{Vektor}(n \times 1) \end{aligned}$$

Wie bekannt kann auch hier, sofern $\boldsymbol{\xi}(t)$ weißes Rauschen darstellt, das Matrixpolynom $\mathbf{C}(z^{-1}) = \mathbf{I}$ gesetzt werden. Gl. (8.12) vereinfacht sich dann zu

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = \mathbf{B}(z^{-1})\mathbf{u}(t-1) + \frac{\boldsymbol{\xi}(t)}{\Delta} \quad (8.13)$$

Der j-Schritt-voraus-Prädiktor

Wie in Kap. 6.1.1, wird auch hier eine Diophantische Gleichung zur Herleitung eines optimalen Prädiktors verwendet:

$$\begin{aligned} \mathbf{I} &= \mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})\Delta + z^{-j}\mathbf{F}_j(z^{-1}) \\ &= \mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1}) + z^{-j}\mathbf{F}_j(z^{-1}) \end{aligned} \quad (8.14)$$

mit

$$\begin{aligned} \mathbf{A}(z^{-1}) &= \Delta\mathbf{A}(z^{-1}) \\ \mathbf{E}_j(z^{-1}) &= \mathbf{E}_{j,0} + \mathbf{E}_{j,1}z^{-1} + \mathbf{E}_{j,2}z^{-2} + \dots + \mathbf{E}_{j,j-1}z^{-(j-1)} \\ \mathbf{F}_j(z^{-1}) &= \mathbf{F}_{j,0} + \mathbf{F}_{j,1}z^{-1} + \mathbf{F}_{j,2}z^{-2} + \dots + \mathbf{F}_{j,na}z^{-na} \end{aligned}$$

8.1.3 Mathematical derivation

The CARIMA model

Analog to SISO systems, here the following approach is taken for the CARIMA model, see equation (8.12).

The dimensions of the individual matrices are:

$$\mathbf{A}(z^{-1}) : \text{Matrix}(n \times n)$$

$$\mathbf{B}(z^{-1}) : \text{Matrix}(n \times m)$$

$$\mathbf{C}(z^{-1}) : \text{Matrix}(n \times n)$$

$$\boldsymbol{\xi}(t) : \text{Vector}(n \times 1)$$

In this case, the matrix polynomial can be set to $\mathbf{C}(z^{-1}) = \mathbf{I}$ if $\boldsymbol{\xi}(t)$ represents white noise. Then equation (8.12) simplifies to equation (8.13).

The j-step ahead predictor

As in Chapter 6.1.1, here a Diophantine equation is used for the derivation of an optimum predictor, too, see equation (8.14).

8.1.3 数学推导

CARIMA 模型

类似于 SISO 系统，CARIMA 模型可表示为方程 (8.12)。

上述各矩阵的维数分别为：

$$\mathbf{A}(z^{-1}) : \text{矩阵}(n \times n)$$

$$\mathbf{B}(z^{-1}) : \text{矩阵}(n \times m)$$

$$\mathbf{C}(z^{-1}) : \text{矩阵}(n \times n)$$

$$\boldsymbol{\xi}(t) : \text{向量}(n \times 1)$$

类似地，如果 $\boldsymbol{\xi}(t)$ 表示白噪声，则设置矩阵多项式 $\mathbf{C}(z^{-1}) = \mathbf{I}$ 。进而，方程 (8.12) 可简化为方程 (8.13)。

J 步前向预测器

正如章节 6.1.1 第 75 页所示，此时也需使用一个丢番图方程用于最优预测器的推导过程，见方程 (8.14)。

in den Dimensionen

$$\mathbf{E}_j(z^{-1}) : \text{Matrix}(n \times n)$$

$$\mathbf{F}_j(z^{-1}) : \text{Matrix}(n \times n)$$

Bezüglich der Grade der Matrizenpolynome gilt das bei Eingrößensystemen gesagt analog.

Nach Auflösen von (8.14) nach $\mathbf{E}_j(z^{-1})\tilde{\mathbf{A}}(z^{-1})$ und Einsetzen des Ergebnisses in die mit $\Delta\mathbf{E}_j(z^{-1})z^j$ multiplizierte Formel (8.13) erhält man die Prädiktionsgleichung

$$\mathbf{y}(t+j) = \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{F}_j(z^{-1})\mathbf{y}(t) + \mathbf{E}_j(z^{-1})\boldsymbol{\xi}(t+j) \quad (8.15)$$

Auch hier liegen die Rauschterme alle in der Zukunft, sind damit unbekannt und können für die Prädiktion folglich vernachlässigt werden. Eine bestmögliche Prädiktion ist daher

$$\hat{\mathbf{y}}(t+j) = \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{F}_j(z^{-1})\mathbf{y}(t) \quad (8.16)$$

Anders als bei SISO-GPC setzt man hier jedoch nun an

$$\mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1}) = \mathbf{G}_j(z^{-1}) + z^{-j}\mathbf{G}_{jp}(z^{-1}) \quad \text{Grad}(\mathbf{G}_j(z^{-1})) < j \quad (8.17)$$

wobei die $\mathbf{G}_j(z^{-1})$ -Matrizenpolynome sich hier im Gegensatz zu den $G_j(z^{-1})$ -Polynomen im eindimensionalen Fall nur auf die Zukunft beziehen. Der die freie Antwort bildende Teil steckt komplett im $z^{-j}\mathbf{G}_{jp}(z^{-1})$ -Term. Damit wird die Prädiktionsgleichung zu

$$\begin{aligned} \hat{\mathbf{y}}(t+j) &= \underbrace{\mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1)}_{\text{erzwungene Antwort}} \\ &\quad + \underbrace{\mathbf{G}_{jp}(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{F}_j(z^{-1})\mathbf{y}(t)}_{\substack{\text{alles Werte in der Vergangenheit} \\ \rightarrow \text{freie Antwort } \mathbf{f}(t+j)}} \end{aligned} \quad (8.18)$$

$$\hat{\mathbf{y}}(t+j) = \mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{f}(t+j) \quad (8.19)$$

mit

$$\mathbf{f}(t+j) = \mathbf{G}_{jp}(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{F}_j(z^{-1})\mathbf{y}(t)$$

der Dimensionen

$$\mathbf{G}_j(z^{-1}), \mathbf{G}_{jp}(z^{-1}) : \text{Matrix}(n \times m)$$

$$\mathbf{f}(t) : \text{Vektor}(n \times 1)$$

with the dimensions

$$\mathbf{E}_j(z^{-1}) : \text{Matrix}(n \times n)$$

$$\mathbf{F}_j(z^{-1}) : \text{Matrix}(n \times n)$$

Concerning the degree of the matrix polynomials, the statements given for uni-dimensional systems apply analogously.

After solving (8.14) for $\mathbf{E}_j(z^{-1})\tilde{\mathbf{A}}(z^{-1})$ and after inserting this result into equation (8.13), which was multiplied with $\Delta\mathbf{E}_j(z^{-1})z^j$ the prediction equation (8.15) can be obtained. All noise terms are in the future, too, which means that they are unknown and so they can be neglected for the prediction. Hence, the best possible prediction is equation (8.16).

Different from the SISO-GPC case, equation (8.17) is set, whereas here the $\mathbf{G}_j(z^{-1})$ matrix polynomials, in contrast to the $G_j(z^{-1})$ polynomials in the uni-dimensional case only refer to the future. The part forming the free response is completely inside the $z^{-j}\mathbf{G}_{jp}(z^{-1})$ term. Thus, the prediction equation becomes to equations (8.18) and (8.19), with the dimensions

$$\mathbf{G}_j(z^{-1}), \mathbf{G}_{jp}(z^{-1}) : \text{Matrix}(n \times m)$$

$$\mathbf{f}(t) : \text{Vector}(n \times 1)$$

维数为

$$\mathbf{E}_j(z^{-1}) : \text{矩阵}(n \times n)$$

$$\mathbf{F}_j(z^{-1}) : \text{矩阵}(n \times n)$$

至于矩阵多项式的阶数，它类似于单变量系统的情形。

方程 (8.13) 两侧首先乘以 $\Delta\mathbf{E}_j(z^{-1})z^j$ ，然后求解方程 (8.14) 中的 $\mathbf{E}_j(z^{-1})\tilde{\mathbf{A}}(z^{-1})$ ，并将其代入乘以 $\Delta\mathbf{E}_j(z^{-1})z^j$ 之后的方程 (8.13)，经适当变换，即可得到预测方程表达式方程 (8.15)。此处的噪声分量也全部位于未来时刻，由于它们是未知的，所以可在预测时进行忽略。从而，最优的预测方程表达式变为方程 (8.16)。

不同于第 6 章所述的 SISO-GPC 情形，此时令方程 (8.17)，其中，矩阵多项式 $\mathbf{G}_j(z^{-1})$ 仅与未来时刻有关，而单维情形下的多项式 $G_j(z^{-1})$ 则同时与过去和未来有关。自由响应部分隐含完整的 $z^{-j}\mathbf{G}_{jp}(z^{-1})$ 项，从而，预测方程可进一步写为方程 (8.18) 和 (8.19)，维数为

$$\mathbf{G}_j(z^{-1}), \mathbf{G}_{jp}(z^{-1}) : \text{矩阵}(n \times m)$$

$$\mathbf{f}(t) : \text{向量}(n \times 1)$$

Die einzelnen Prädiktionsschritte lauten demnach für ein MIMO-System:

$$\hat{\mathbf{y}}(t+1) = \mathbf{G}_1(z^{-1})\Delta\mathbf{u}(t) + \mathbf{f}(t+1) \quad (8.20)$$

$$\hat{\mathbf{y}}(t+2) = \mathbf{G}_2(z^{-1})\Delta\mathbf{u}(t+1) + \mathbf{f}(t+2) \quad (8.21)$$

$$\hat{\mathbf{y}}(t+3) = \mathbf{G}_3(z^{-1})\Delta\mathbf{u}(t+2) + \mathbf{f}(t+3) \quad (8.22)$$

⋮

$$\hat{\mathbf{y}}(t+N_p) = \mathbf{G}_{N_p}(z^{-1})\Delta\mathbf{u}(t+N_p-1) + \mathbf{f}(t+N_p) \quad (8.23)$$

Berechnung der Steuergrößen

Auf die mit Hilfe von (8.19) vorausberechneten Werte wird eine quadratische Kostenfunktion angewendet, um eine optimale Steuerfolge zu berechnen. Die Kostenfunktion selbst ist dabei dieselbe wie bei der Eingrößenregelung.

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \mu_j \|\hat{\mathbf{y}}(t+j) - \mathbf{w}(t+j)\|^2 + \sum_{j=1}^{N_u} \lambda_j \|\Delta\mathbf{u}(t+j-1)\|^2 \quad (8.24)$$

Da durch die Betragsbildung und das Quadrieren des zu erwartenden Regelfehlers bzw. der zukünftigen Stellgrößenänderungen nur ein skalarer Wert übrig bleibt, sind die Gewichtungsfaktoren μ_j und λ_j auch im mehrdimensionalen Fall skalare Werte. Eine unterschiedliche Gewichtung der verschiedenen Ausgänge der Strecke ist somit leider nicht möglich, denn dafür müssten die Regelabweichung oder die $\Delta\mathbf{u}$ -Werte vor der Betragsbildung mit einer Gewichtungsmatrix multipliziert werden.

Wie auch beim eindimensionalen Fall wählt man $\mu_j = 1$ für alle j . Meist wird auch $N_1 = 1$ und $N_2 = N_p$ gesetzt.

Für die Stellgrößenoptimierung wird der Term $\mathbf{G}_{j_p}(z^{-1})\Delta\mathbf{u}(t-1)$ in Gl. (8.19) zu einer Multiplikation von reellen Matrizen $\mathbf{G}\mathbf{U}$ zusammengefasst. Damit kann man wie beim SISO-System schreiben

$$\mathbf{Y} = \mathbf{G}\mathbf{U} + \mathbf{F} \quad (8.25)$$

wobei auch hier $\mathbf{U} = \Delta\mathbf{U}$. Analog zum eindimensionalen Fall lässt sich durch Einsetzen von (8.25) in (8.24) die der eindimensionalen Gl. (6.20) entsprechende zusammengefasste MIMO-Kostenfunktion bilden:

$$J = \mathbf{U}^T(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})\mathbf{U} + 2\mathbf{U}^T\mathbf{G}^T(\mathbf{F} - \mathbf{W}) + (\mathbf{F} - \mathbf{W})^T(\mathbf{F} - \mathbf{W}) \quad (8.26)$$

Nach Ableitung von (8.26) nach $\frac{\partial}{\partial \mathbf{U}}$ werden durch Nullsetzen des sich ergebenden Terms optimale, zukünftige Stellgrößen gemäß

$$\mathbf{U} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}^T(\mathbf{W} - \mathbf{F}) \quad (8.27)$$

bestimmt, wobei auch hier die Vereinfachungen $\mu_j = 1$ und $\lambda_j = \lambda$ für alle j getroffen worden sind.

Therefore, the single prediction steps for a MIMO system are equations (8.20) to (8.23).

Calculation of the actuating variables

A quadratic cost function is applied to the values precalculated with (8.19) in order to calculate an optimum sequence of values for the actuating variables. The cost function is the same as the one for uni-dimensional control, see equation (8.24). Because of calculating the absolute value and because of squaring the expected control error and, respectively, future changes of the actuating variables, only a scalar value remains. Hence, the weighting factors μ_j and λ_j are even in the multidimensional case scalar values, too. Unfortunately, a different weighting of the different system outputs is, because of this, not possible because then the control errors or the $\Delta \mathbf{u}$ -values would have to be multiplied with a weighing matrix before the magnitude is calculated.

According to the uni-dimensional case, $\mu_j = 1$ for all j is selected and mostly also $N_1 = 1$ and $N_2 = N_p$ is set.

To optimize the values of the actuating variables, the term $\mathbf{G}_{jp}(z^{-1})\Delta \mathbf{u}(t-1)$ in equation (8.19) is combined to a multiplication of real matrices $\mathbf{G}\mathbf{U}$. Thus, similar to the SISO system, the following can be written, see equation (8.25), in which also $\mathbf{U} = \Delta \mathbf{U}$. Analog to the uni-dimensional case, a simplified MIMO cost function corresponding to the uni-dimensional equation (6.20) can be set up by substituting (8.25) in (8.24), see equation (8.26).

After the derivation of (8.26) with respect to $\frac{\partial}{\partial \mathbf{U}}$ and by setting the resulting term equal to zero, optimum future values for the actuating variables according to equation (8.27) can be determined, in which the simplifications $\mu_j = 1$ and $\lambda_j = \lambda$ for all j were also applied.

对于 MIMO 系统来说, 单步预测的表达式为 方程 (8.20) 至 (8.23)。

控制变量计算

为了计算最优的控制序列, 可使用一个二次成本函数, 它基于方程 (8.19) 所得的预测值。成本函数本身形式与单变量控制时的情形一致, 见方程 (8.24)。由于对期望的控制偏差和未来的控制增量进行求模并平方, 最终所得为标量值, 所以在多维情况下的权重系数 μ_j 和 λ_j 也为标量值。遗憾的是, 无法对被控对象不同的输出选择不同的权重, 原因是控制偏差和控制增量 $\Delta \mathbf{u}$ 的值在求模之前已经与相应的权重矩阵相乘。

类似于单维情形, 此处也对所有的 j , 选取 $\mu_j = 1$ 。通常也设置 $N_1 = 1$ 和 $N_2 = N_p$ 。

为了实现控制变量的优化, 可将方程 (8.19) 中的 $\mathbf{G}_{jp}(z^{-1})\Delta \mathbf{u}(t-1)$ 项整合为一个实矩阵的乘积形式 $\mathbf{G}\mathbf{U}$ 。如此, 即可得到与单输入单输出系统时类似的表达式, 见方程 (8.25), 其中, $\mathbf{U} = \Delta \mathbf{U}$ 。类似于单维情形, 将方程 (8.25) 代入至方程 (8.24), 并经过适当变换, 即可得到一个与单维时所得方程 (6.20) 相对应的经整合后的 MIMO 成本函数, 见方程 (8.26)。

对方程 (8.26) 向 $\frac{\partial}{\partial \mathbf{U}}$ 求导, 并将其置零, 即可确定最优的、未来的控制变量表达式 (8.27), 其中, 此时也使用了 $\mu_j = 1$ 和对所有的 j 来说 $\lambda_j = \lambda$ 这些简化条件。

Rekursion der Diophantischen Gleichung

Die Lösung der Diophantischen Gleichung (8.14) erfolgt ebenfalls ähnlich wie beim SISO-System, indem sie für j und $j+1$ aufgestellt und die sich ergebenden Formeln voneinander subtrahiert werden:

$$\begin{array}{r|l} - & \mathbf{I} = \mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1}) + z^{-j}\mathbf{F}_j(z^{-1}) \\ + & \mathbf{I} = \mathbf{E}_{j+1}(z^{-1})\mathbf{A}(z^{-1}) + z^{-(j+1)}\mathbf{F}_{j+1}(z^{-1}) \\ \hline & 0 = (\mathbf{E}_{j+1}(z^{-1}) - \mathbf{E}_j(z^{-1}))\mathbf{A}(z^{-1}) + z^{-j}(z^{-1}\mathbf{F}_{j+1}(z^{-1}) - \mathbf{F}_j(z^{-1})) \end{array} \quad (8.28)$$

Analog zu Gl. (6.25) ergibt sich wegen der Grade der Matrizenpolynome der Ansatz

$$\mathbf{E}_{j+1}(z^{-1}) - \mathbf{E}_j(z^{-1}) = \mathbf{R}(z^{-1}) + \mathbf{R}_j z^{-j} \quad (8.29)$$

wobei $\mathbf{R}(z^{-1})$ ein Matrizenpolynom vom Grad $\leq j-1$ und \mathbf{R}_j eine reelle Matrix ist. Wie im eindimensionalen Fall gilt auch hier, dass $\mathbf{R}(z^{-1}) = 0$, weil $\mathbf{A}(z^{-1})$ monisch² ist ($\mathbf{A}_0 = \mathbf{I}$). Eingesetzt in die Diophantische Gleichung ergibt sich:

$$z^{-1}\mathbf{F}_{j+1}(z^{-1}) = \mathbf{F}_j(z^{-1}) - \mathbf{R}_j\mathbf{A}(z^{-1}) \quad (8.30)$$

Da, wie bereits erwähnt, $\mathbf{A}(z^{-1})$ monisch ist, folgt nach einem Koeffizientenvergleich der obigen Matrizenpolynomgleichung:

$$\mathbf{F}_{j,0} = \mathbf{R}_j \quad (8.31)$$

$$\mathbf{F}_{j+1,i} = \mathbf{F}_{j,i+1} - \mathbf{R}_j\mathbf{A}_{i+1} \quad (8.32)$$

wobei i von $0 \dots \text{Grad}(\mathbf{F}_{j+1})$ läuft. Aus den Gln. (8.29), (8.31) und der Identität $\mathbf{R}(z^{-1}) = 0$ folgt ferner die Rekursionsvorschrift für $\mathbf{E}_j(z^{-1})$:

$$\mathbf{E}_{j+1}(z^{-1}) = \mathbf{E}_j(z^{-1}) + \mathbf{F}_{j,0}z^{-j} \quad (8.33)$$

Unter Berücksichtigung der Tatsache, dass $\mathbf{A}(z^{-1})$ monisch ist, erhält man für $j = 1$ aus (8.14) die Endbedingungen der Rekursion zu

$$\mathbf{E}_1(z^{-1}) = \mathbf{E}_0 = \mathbf{I} \quad (8.34)$$

$$\mathbf{F}_1(z^{-1}) = z(\mathbf{I} - \mathbf{A}(z^{-1})) \quad (8.35)$$

Auch im mehrdimensionalen Fall sind, wie aus (8.33) leicht zu ersehen ist, die Koeffizienten von 0 bis $-(j-1)$ des Matrizenpolynoms $\mathbf{E}_{j+1}(z^{-1})$ alle identisch.

² Zur Erläuterung siehe Anhang A.

Recursion of the Diophantine equation

The Diophantine equation (8.14) is also solved similar to the SISO system by setting it up for j and $j + 1$ and then by subtracting the resulting expressions from each other, see equation (8.28).

Analog to equation (6.25), the approach (8.29) is taken because of the degrees of the matrix polynomials. In (8.29) $\mathbf{R}(z^{-1})$ is a matrix polynomial of the degree $\leq j - 1$ and \mathbf{R}_j is a real matrix. As in the uni-dimensional case, here it also applies that $\mathbf{R}(z^{-1}) = 0$, because $\mathbf{A}(z^{-1})$ is monic² ($\mathbf{A}_0 = \mathbf{I}$). After substituting this into the Diophantine equation, the following results, see equation (8.30).

Since, as already mentioned, $\mathbf{A}(z^{-1})$ is monic, the following results after a coefficient comparison of the above matrix polynomial equation, see equations (8.31) and (8.32), in which i varies from $0 \dots \text{degree}(\mathbf{F}_{j+1})$. Moreover, from the equations (8.29), (8.31) and the identity $\mathbf{R}(z^{-1}) = 0$ the recursion law for $\mathbf{E}_j(z^{-1})$ follows, see equation (8.33).

Considering that $\mathbf{A}(z^{-1})$ is monic the final terms of the recursion are obtained from (8.14) for $j = 1$, see equations (8.34) and (8.35). In the multidimensional case, as it can be seen easily from (8.33), the coefficients from 0 to $-(j - 1)$ of the matrix polynomial $\mathbf{E}_{j+1}(z^{-1})$ are all identical, too.

丢番图方程的递归

至于丢番图方程 (8.14) 的求解, 它类似于前面所述 SISO 系统的相应求解方式。为此, 首先列出 j 和 $j + 1$ 时刻的丢番图方程, 然后对其进行求差运算, 具体过程见方程 (8.28)。

类似于方程 (6.25), 考虑到矩阵多项式的阶数, 同样给出假设 (8.29), 其中, $\mathbf{R}(z^{-1})$ 的阶数是 $\leq j - 1$ 的矩阵多项式, \mathbf{R}_j 为实数矩阵。类似于单维情形, 此处也满足 $\mathbf{R}(z^{-1}) = 0$, 因为 $\mathbf{A}(z^{-1})$ 是单调的² ($\mathbf{A}_0 = \mathbf{I}$)。然后, 将其代入至丢番图方程, 可得方程 (8.30)。

如前所述, 由于 $\mathbf{A}(z^{-1})$ 是单调的, 通过对上述矩阵多项式方程进行系数对比, 可得方程 (8.31) 和 (8.32), 其中, i 的范围对应 $0 \dots \mathbf{F}_{j+1}$ 的阶数。更进一步, 根据方程 (8.29) 和 (8.31), 并结合恒等式 $\mathbf{R}(z^{-1}) = 0$, 可得适用于 $\mathbf{E}_j(z^{-1})$ 的递归规则, 见方程 (8.33)。

考虑到 $\mathbf{A}(z^{-1})$ 是单调的这一事实, 根据方程 (8.14), 即可得到 $j = 1$ 时的递归终止条件, 见方程 (8.34) 和 (8.35)。由方程 (8.33) 不难看出, 即使在多维情况下, 矩阵多项式 $\mathbf{E}_{j+1}(z^{-1})$ 从 0 至 $-(j - 1)$ 的系数也完全一致。

² For an explanation see appendix A.

² 具体解释可见附录 A。

Folglich kann man die einzelnen Polynome schreiben:

$$\mathbf{E}_1(z^{-1}) = \mathbf{E}_0 \quad (8.36)$$

$$\mathbf{E}_2(z^{-1}) = \mathbf{E}_0 + \mathbf{E}_1 z^{-1} \quad (8.37)$$

$$\mathbf{E}_3(z^{-1}) = \mathbf{E}_0 + \mathbf{E}_1 z^{-1} + \mathbf{E}_2 z^{-2} \quad (8.38)$$

⋮

$$\mathbf{E}_j(z^{-1}) = \mathbf{E}_0 + \mathbf{E}_1 z^{-1} + \mathbf{E}_2 z^{-2} + \dots + \mathbf{E}_{j-1} z^{-(j-1)} \quad (8.39)$$

$$\begin{aligned} \mathbf{E}_{j+1}(z^{-1}) &= \mathbf{E}_0 + \mathbf{E}_1 z^{-1} + \mathbf{E}_2 z^{-2} + \dots + \mathbf{E}_{j-1} z^{-(j-1)} + \mathbf{E}_j z^{-j} \\ &= \mathbf{E}_j(z^{-1}) + \mathbf{E}_j z^{-j} \end{aligned} \quad (8.40)$$

wobei

$$\mathbf{E}_j = \mathbf{F}_{j,0}$$

Berechnung der freien und der erzwungenen Antwort

Wie auf Seite 173 gezeigt, wird die Prädiktionsgleichung (8.16) in die erzwungene und die freie Antwort getrennt (Gl. (8.19) bzw. (8.25)). Aus (8.18) geht hervor, dass für die erzwungene Antwort nur der Teil $\mathbf{G}_j(z^{-1})$ interessant ist. Demnach setzt man für die Berechnung von $\mathbf{G}_j(z^{-1})$ gemäß (8.17) an

$$\mathbf{G}_j(z^{-1}) = \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1}) \quad (8.41)$$

wobei definitionsgemäß aus dem Produkt $\mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})$ nur die Terme vom Grad $\leq (j-1)$ verwendet werden dürfen. Dabei ist zu beachten, dass für alle $\mathbf{E}_{k,j}$ gilt $\mathbf{E}_{k,j} = \mathbf{E}_j$. Man erhält folglich die Polynommatrix $\mathbf{G}_j(z^{-1})$ aus den Termen vom Grad $0 \dots (j-1)$ der Beziehung

$$\mathbf{G}_j(z^{-1}) = (\mathbf{E}_0 + \mathbf{E}_1 z^{-1} + \dots + \mathbf{E}_{j-1} z^{-(j-1)}) (\mathbf{B}_0 + \mathbf{B}_1 z^{-1} + \dots + \mathbf{B}_{nb} z^{-nb}) \quad (8.42)$$

Für $\mathbf{G}_{j+1}(z^{-1})$ berechnet man analog die Terme $0 \dots j$ aus:

$$\begin{aligned} \mathbf{G}_{j+1}(z^{-1}) &= \mathbf{E}_{j+1}(z^{-1})\mathbf{B}(z^{-1}) \\ &= (\mathbf{E}_j(z^{-1}) + \mathbf{F}_{j,0} z^{-j})\mathbf{B}(z^{-1}) \\ &= (\mathbf{E}_0 + \mathbf{E}_1 z^{-1} + \dots + \mathbf{E}_{j-1} z^{-(j-1)})\mathbf{B}(z^{-1}) + \mathbf{F}_{j,0} z^{-j}\mathbf{B}(z^{-1}) \\ &= \mathbf{G}_j(z^{-1}) \\ &\quad + \underbrace{\mathbf{E}_0 \mathbf{B}_j z^{-j} + \mathbf{E}_1 \mathbf{B}_{j-1} z^{-j} + \dots + \mathbf{E}_{j-1} \mathbf{B}_1 z^{-j}}_{\substack{\text{zusätzlich aus } (\mathbf{E}_0 + \mathbf{E}_1 z^{-1} + \dots + \mathbf{E}_{j-1} z^{-(j-1)})\mathbf{B}(z^{-1}) \\ \text{da für } \mathbf{G}_{j+1}(z^{-1}) \text{ nun Terme bis zum Grad } j \\ \text{in die Berechnung einfließen müssen}}} + \mathbf{F}_{j,0} \mathbf{B}_0 z^{-j} \\ &= \mathbf{G}_j(z^{-1}) + (\mathbf{F}_{j,0} \mathbf{B}_0 + \mathbf{E}_{j-1} \mathbf{B}_1 + \dots + \mathbf{E}_0 \mathbf{B}_j) z^{-j} \end{aligned}$$

Therefore, the individual polynomials can be written as equations (8.36) to (8.40).

Calculation of the free and forced response

As shown on page 173, the prediction equation (8.16) is separated into the forced and into the free response (equations (8.19) and (8.25)). From (8.18) follows that only the part $\mathbf{G}_j(z^{-1})$ is essential for the forced response. Therefore, for the calculation of $\mathbf{G}_j(z^{-1})$ according to (8.17) the approach (8.41) is taken in which, according to definitions, only the terms of the product $\mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})$ with degree $\leq (j-1)$ shall be used. Besides, it has to be noted that for all $\mathbf{E}_{k,j}$, it applies that $\mathbf{E}_{k,j} = \mathbf{E}_j$. Therefore, the polynomial matrix $\mathbf{G}_j(z^{-1})$ can be obtained from the terms of degree $0 \dots (j-1)$ of the following equation, see equation (8.42). For $\mathbf{G}_{j+1}(z^{-1})$ the terms $0 \dots j$ are calculated analogously:

$$\begin{aligned} \mathbf{G}_{j+1}(z^{-1}) &= \mathbf{E}_{j+1}(z^{-1})\mathbf{B}(z^{-1}) \\ &= (\mathbf{E}_j(z^{-1}) + \mathbf{F}_{j,0}z^{-j})\mathbf{B}(z^{-1}) \\ &= (\mathbf{E}_0 + \mathbf{E}_1z^{-1} + \dots + \mathbf{E}_{j-1}z^{-(j-1)})\mathbf{B}(z^{-1}) + \mathbf{F}_{j,0}z^{-j}\mathbf{B}(z^{-1}) \\ &= \mathbf{G}_j(z^{-1}) \\ &\quad + \underbrace{\mathbf{E}_0\mathbf{B}_jz^{-j} + \mathbf{E}_1\mathbf{B}_{j-1}z^{-j} + \dots + \mathbf{E}_{j-1}\mathbf{B}_1z^{-j}}_{\substack{\text{additionally from } (\mathbf{E}_0 + \mathbf{E}_1z^{-1} + \dots + \mathbf{E}_{j-1}z^{-(j-1)})\mathbf{B}(z^{-1}) \\ \text{since for } \mathbf{G}_{j+1}(z^{-1}) \text{ only terms up to degree } j \\ \text{need to be considered in the calculation}}} + \mathbf{F}_{j,0}\mathbf{B}_0z^{-j} \\ &= \mathbf{G}_j(z^{-1}) + (\mathbf{F}_{j,0}\mathbf{B}_0 + \mathbf{E}_{j-1}\mathbf{B}_1 + \dots + \mathbf{E}_0\mathbf{B}_j)z^{-j} \end{aligned}$$

因此，即可逐一列出 $\mathbf{E}_{j+1}(z^{-1})$ 的表达式，见 方程 (8.36) 至 (8.40)。

自由响应与强制响应的计算 如正文第 173 页所述，预测方程 (8.16) 可分解为强制响应和自由响应两部分 (见方程 (8.19) 和 (8.25))。根据方程 (8.18) 可知，对于强制响应部分只有 $\mathbf{G}_j(z^{-1})$ 部分是重要的。从而，为了计算 $\mathbf{G}_j(z^{-1})$ ，需基于方程 (8.17) 假设 (8.41)，其中，根据定义， $\mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})$ 中只有阶数 $\leq (j-1)$ 的项才是可利用的。要注意的是，对于所有的 $\mathbf{E}_{k,j}$ ，均满足 $\mathbf{E}_{k,j} = \mathbf{E}_j$ 。根据上述关系式中阶数范围为 $0 \dots (j-1)$ 的项，即可得到多项式矩阵 $\mathbf{G}_j(z^{-1})$ 的表达式，见 方程 (8.42)。可根据下式计算 $\mathbf{G}_{j+1}(z^{-1})$ ：

$$\begin{aligned} \mathbf{G}_{j+1}(z^{-1}) &= \mathbf{E}_{j+1}(z^{-1})\mathbf{B}(z^{-1}) \\ &= (\mathbf{E}_j(z^{-1}) + \mathbf{F}_{j,0}z^{-j})\mathbf{B}(z^{-1}) \\ &= (\mathbf{E}_0 + \mathbf{E}_1z^{-1} + \dots + \mathbf{E}_{j-1}z^{-(j-1)})\mathbf{B}(z^{-1}) + \mathbf{F}_{j,0}z^{-j}\mathbf{B}(z^{-1}) \\ &= \mathbf{G}_j(z^{-1}) \\ &\quad + \underbrace{\mathbf{E}_0\mathbf{B}_jz^{-j} + \mathbf{E}_1\mathbf{B}_{j-1}z^{-j} + \dots + \mathbf{E}_{j-1}\mathbf{B}_1z^{-j}}_{\substack{\text{额外来自于 } (\mathbf{E}_0 + \mathbf{E}_1z^{-1} + \dots + \mathbf{E}_{j-1}z^{-(j-1)})\mathbf{B}(z^{-1}) \\ \text{因为 } \mathbf{G}_{j+1}(z^{-1}) \text{ 中只有阶数至 } j \text{ 才需要在计算中得以考虑}}} + \mathbf{F}_{j,0}\mathbf{B}_0z^{-j} \\ &= \mathbf{G}_j(z^{-1}) + (\mathbf{F}_{j,0}\mathbf{B}_0 + \mathbf{E}_{j-1}\mathbf{B}_1 + \dots + \mathbf{E}_0\mathbf{B}_j)z^{-j} \end{aligned}$$

Wie bei einer eindimensionalen GPC-Regelung unterscheiden sich auch hier die einzelnen \mathbf{G}_j nur durch das neu hinzugefügte Element. Man kann daher schreiben

$$\mathbf{G}_j(z^{-1}) = \mathbf{G}_0 + \mathbf{G}_1 z^{-1} + \mathbf{G}_2 z^{-2} + \dots + \mathbf{G}_{j-1} z^{-(j-1)} \quad j = 1 \dots N_p \quad (8.43)$$

mit

$$\mathbf{G}_j = \mathbf{F}_{j,0} \mathbf{B}_0 + \sum_{k=1}^j \mathbf{E}_{j-k} \mathbf{B}_k \quad j = 0 \dots (N_p - 1)$$

Da $\mathbf{F}_{j,0} = \mathbf{E}_j$ ergibt sich:

$$\mathbf{G}_j = \sum_{k=0}^j \mathbf{E}_{j-k} \mathbf{B}_k \quad (8.44)$$

Für $k > nb$ entfallen natürlich die entsprechenden Summanden, da dann die \mathbf{B}_k nicht mehr existieren. Folglich unterscheiden sich auch die Zeilen von \mathbf{G} nur um jeweils ein Element. Damit wird (8.25) zu:

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{y}}(t+1) \\ \vdots \\ \hat{\mathbf{y}}(t+j) \\ \vdots \\ \hat{\mathbf{y}}(t+N_p) \end{bmatrix} &= \begin{bmatrix} \mathbf{G}_0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}_{j-1} & \cdots & \mathbf{G}_0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}_{N_p-1} & \cdots & \mathbf{G}_{N_p-j} & \cdots & \mathbf{G}_0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}(t) \\ \vdots \\ \Delta \mathbf{u}(t+j-1) \\ \vdots \\ \Delta \mathbf{u}(t+N_p-1) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{f}(t+1) \\ \vdots \\ \mathbf{f}(t+j) \\ \vdots \\ \mathbf{f}(t+N_p) \end{bmatrix} \end{aligned} \quad (8.45)$$

Zur Berechnung der freien Antwort benötigt man nach Gl. (8.18) die beiden Matrizenpolynome $\mathbf{G}_{jp}(z^{-1})$ und $\mathbf{F}_j(z^{-1})$. Die Berechnung von $\mathbf{F}_j(z^{-1})$ wurde bereits im vorhergehenden Kapitel erläutert. $\mathbf{G}_{jp}(z^{-1})$ hingegen berechnet sich genau wie $\mathbf{G}_j(z^{-1})$ über die Definition (8.17), wobei diesmal jedoch die Terme vom Grad $> j$ interessant sind, da in die freie Antwort nur bereits vergangene Stellgrößen eingehen können. Dies wird über den Verschiebungsfaktor z^{-j} in Gl. (8.17) realisiert.

As in unidimensional GPC, here the individual \mathbf{G}_j terms differ only in the newly added element, too. Hence it can be written as equation (8.43). Since $\mathbf{F}_{j,0} = \mathbf{E}_j$, the following results, see equation (8.44).

For $k > nb$, the corresponding summands do not exist, since then the terms \mathbf{B}_k do not exist anymore. Therefore, the rows of \mathbf{G} also differ only in one element. With this (8.25) results to equation (8.45).

For the calculation of the free response, both matrix polynomials $\mathbf{G}_{jp}(z^{-1})$ and $\mathbf{F}_j(z^{-1})$ are required according to equation (8.18). The calculation of $\mathbf{F}_j(z^{-1})$ has already been explained in the previous chapter. $\mathbf{G}_{jp}(z^{-1})$, however, is calculated exactly like $\mathbf{G}_j(z^{-1})$ via definition (8.17), whereas now, however, the terms with degree $> j$ are of interest, since only past values of the actuating variables are considered in the free response. This is realized via the time shift factor z^{-j} in equation (8.17).

类似于单维 GPC 控制的情形, \mathbf{G}_j 与 \mathbf{G}_{j+1} 之间的差别仅在于新增加的元素。从而, 可以得到 \mathbf{G}_j 的表达式, 见方程 (8.43)。由于 $\mathbf{F}_{j,0} = \mathbf{E}_j$, 可得方程 (8.44)。

当 $k > nb$ 时, 则消去相应的带有 \mathbf{B}_k 的项, 因为此时 \mathbf{B}_k 已不复存在。这也意味着矩阵 \mathbf{G} 的行与行之间的差别也仅在于所增加的元素。从而, 方程 (8.25) 变化为方程 (8.45)。

根据方程 (8.18) 可知, 自由响应的计算需要用到矩阵多项式 $\mathbf{G}_{jp}(z^{-1})$ 和 $\mathbf{F}_j(z^{-1})$: $\mathbf{F}_j(z^{-1})$ 的计算方法已经在前面章节中得到阐述; 而 $\mathbf{G}_{jp}(z^{-1})$ 的计算则类似于基于定义 (8.17) 计算 $\mathbf{G}_j(z^{-1})$ 的方式, 不同之处在于此时仅仅考虑的是阶数 $> j$ 的项, 因为自由响应的计算仅需用到过去的控制变量。借助于方程 (8.17) 中的平移系数 z^{-j} 可以轻易实现上述过程。

Löst man die Gleichung auf, so erhält man für die einzelnen Elemente von $\mathbf{G}_{jp}(z^{-1})$:

$$\mathbf{G}_{jp,i} = \sum_{k=0}^{j-1} \mathbf{E}_k \mathbf{B}_{j+i-k} \quad j = 1 \dots N_p; \quad i = 0 \dots (nb - 1) \quad (8.46)$$

Auch hier muss beachtet werden, dass Summanden mit $(j + i - k) > nb$ entfallen.

Wie schon beim SISO-System eingeführt, werden wieder die Koeffizienten der Polynommatrizen $\mathbf{G}_p(z^{-1})$ und $\mathbf{F}(z^{-1})$ zusammengeführt, so dass gilt:

$$\mathbf{F} = \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{G}_p(z^{-1})\Delta\mathbf{u}(t - 1) = \mathbf{F}\mathbf{G}_p \cdot \mathbf{YU} \quad (8.47)$$

wobei

$$\begin{aligned} \mathbf{F}(z^{-1}) &= \begin{bmatrix} \mathbf{F}_1(z^{-1}) \\ \vdots \\ \mathbf{F}_{N_p}(z^{-1}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F}_{1,0} + \mathbf{F}_{1,1}z^{-1} + \dots + \mathbf{F}_{1,na}z^{-na} \\ \vdots \\ \mathbf{F}_{N_p,0} + \mathbf{F}_{N_p,1}z^{-1} + \dots + \mathbf{F}_{N_p,na}z^{-na} \end{bmatrix} \\ \mathbf{G}_p(z^{-1}) &= \begin{bmatrix} \mathbf{G}_{1p}(z^{-1}) \\ \vdots \\ \mathbf{G}_{N_pp}(z^{-1}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{G}_{1p,0} + \mathbf{G}_{1p,1}z^{-1} + \dots + \mathbf{G}_{1p,nb-1}z^{-(nb-1)} \\ \vdots \\ \mathbf{G}_{N_pp,0} + \mathbf{G}_{N_pp,1}z^{-1} + \dots + \mathbf{G}_{N_pp,nb-1}z^{-(nb-1)} \end{bmatrix} \end{aligned}$$

sowie die Zusammenfassungen

$$\begin{aligned} \mathbf{F}\mathbf{G}_p &= \begin{bmatrix} \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \dots & \mathbf{F}_{1,na} & \mathbf{G}_{1p,0} & \mathbf{G}_{1p,1} & \dots & \mathbf{G}_{1p,nb-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \dots & \mathbf{F}_{N_p,na} & \mathbf{G}_{N_pp,0} & \mathbf{G}_{N_pp,1} & \dots & \mathbf{G}_{N_pp,nb-1} \end{bmatrix} \\ \mathbf{YU} &= \begin{bmatrix} \mathbf{y}(t) \\ \vdots \\ \mathbf{y}(t - na) \\ \Delta\mathbf{u}(t - 1) \\ \vdots \\ \Delta\mathbf{u}(t - nb) \end{bmatrix} \end{aligned}$$

If the equation is solved, the individual elements of $\mathbf{G}_{jp}(z^{-1})$ can be obtained:

$$\mathbf{G}_{jp,i} = \sum_{k=0}^{j-1} \mathbf{E}_k \mathbf{B}_{j+i-k} \quad j = 1 \dots N_p; \quad i = 0 \dots (nb - 1) \quad (8.46)$$

Here it has to be noted, too, that summands with $(j + i - k) > nb$ are not considered.

As already done for the SISO system, the coefficients of the polynomial matrices $\mathbf{G}_p(z^{-1})$ and $\mathbf{F}(z^{-1})$ are combined so that the following applies, see [equation \(8.47\)](#).

求解相应方程，即可得到适用于 $\mathbf{G}_{jp}(z^{-1})$ 单个元素的表达式如下：

$$\mathbf{G}_{jp,i} = \sum_{k=0}^{j-1} \mathbf{E}_k \mathbf{B}_{j+i-k} \quad j = 1 \dots N_p; \quad i = 0 \dots (nb - 1) \quad (8.46)$$

此处也需要注意的是，当 $(j + i - k) > nb$ 时，所有相关的项也不复存在。

类似于 SISO 系统所介绍的情况，此处也将多项式矩阵 $\mathbf{G}_p(z^{-1})$ 和 $\mathbf{F}(z^{-1})$ 的系数进行合并处理，则有[方程 \(8.47\)](#)。

8.1.4 Berücksichtigung des Regelhorizonts/Consideration of the control horizon/考虑控制步长后的简化

Wenn nach N_u Takten die Stellgröße $\mathbf{u}(t)$ als konstant angenommen wird, nennt man N_u den Regelhorizont. Da sich für $N_u < t \leq N_2$ die Stellgröße nun nicht mehr verändert ($\Delta\mathbf{u}(t) = 0$), kann Rechenzeit eingespart werden, da nicht mehr die vollen Prädiktionsmatrizen benötigt werden. So vereinfacht sich Gl. (8.45) unter der Voraussetzung, dass $N_u \leq N_2$ zu:

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{y}}(t + N_1) \\ \vdots \\ \hat{\mathbf{y}}(t + N_2) \end{bmatrix} &= \begin{bmatrix} \mathbf{G}_{N_1-1} & \mathbf{G}_{N_1-2} & \cdots & \mathbf{G}_{N_1-N_u} \\ \vdots & \vdots & & \vdots \\ \mathbf{G}_{N_2-1} & \mathbf{G}_{N_2-2} & \cdots & \mathbf{G}_{N_2-N_u} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}(t) \\ \vdots \\ \Delta\mathbf{u}(t + N_u - 1) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{f}(t + N_1) \\ \vdots \\ \mathbf{f}(t + N_2) \end{bmatrix} \end{aligned} \quad (8.48)$$

oder

$$\mathbf{Y}_{N_{12}} = \mathbf{G}_{N_{12u}} \mathbf{U}_{N_u} + \mathbf{F}_{N_{12}} \quad (8.49)$$

Damit vereinfachen sich die in Kap. 8.1.3 auf Seite 175 genannten Formeln zur Berechnung optimaler Steuergrößen natürlich ebenfalls. Setzt man statt (8.25) Gl. (8.49) in (8.24) ein, so ergibt sich für die Kostenfunktion

$$J = (\mathbf{G}_{N_{12u}} \mathbf{U}_{N_u} + \mathbf{F}_{N_{12}} - \mathbf{W}_{N_{12}})^T (\mathbf{G}_{N_{12u}} \mathbf{U}_{N_u} + \mathbf{F}_{N_{12}} - \mathbf{W}_{N_{12}}) + \lambda \mathbf{U}_{N_u}^T \mathbf{U}_{N_u} \quad (8.50)$$

Sucht man das Minimum obiger Gleichung, dann erhält man die optimalen Steuergrößen unter Berücksichtigung des Regelhorizonts N_u aus der Formel:

$$\mathbf{U}_{N_u} = (\mathbf{G}_{N_{12u}}^T \mathbf{G}_{N_{12u}} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}_{N_{12u}}^T (\mathbf{W}_{N_{12}} - \mathbf{F}_{N_{12}}) \quad (8.51)$$

Auch hier wird selbstverständlich nur $\Delta\mathbf{u}(t)$ benötigt. Daher brauchen nur die ersten m Zeilen von \mathbf{U} berechnet werden.

Schließt man nun rückwärts, welche der ab Seite 179 genannten, zur Berechnung der freien und erzwungenen Antwort notwendigen Formeln sich ebenfalls vereinfachen, so ist klar, dass wegen der Beschränkung von \mathbf{F} auf $\mathbf{F}_{N_{12}}$ auch nicht mehr die gesamte Matrix $\mathbf{F}\mathbf{G}_p$ benötigt wird. Da $\mathbf{Y}\mathbf{U}$ nur die *vergangenen* Werte von $\mathbf{y}(t)$ und $\Delta\mathbf{u}(t)$ beinhaltet, ergibt sich für diese Matrix allerdings keine Vereinfachung. Analog ist die *Spaltenanzahl* von $\mathbf{F}\mathbf{G}_{pN_{12}}$ und $\mathbf{F}\mathbf{G}_p$ identisch, lediglich die Anzahl der *Zeilen* wird reduziert.

8.1.4 Consideration of the control horizon

If the actuating variable $\mathbf{u}(t)$ is assumed to be constant after N_u time steps, N_u is called control horizon. Since the actuating variable does not change anymore ($\Delta\mathbf{u}(t) = 0$) for $N_u < t \leq N_2$, computation time can be saved as full prediction matrices are not required anymore. Thus, equation (8.45), assuming that $N_u \leq N_2$, simplifies to equation (8.48) or (8.49).

Hence, the equations for the calculation of optimum values for the actuating variables given in Chapter 8.1.3 on page 175 can be further simplified. Substituting equation (8.49) instead of (8.25) into (8.24) results in the following cost function, see equation (8.50).

If the minimum of the above equation is determined, optimum values for the actuating variables which take the control horizon N_u into account can be obtained from the following equation, see equation (8.51). Here of course only $\Delta\mathbf{u}(t)$ is required, too. Hence, only the first m rows of \mathbf{U} need to be calculated.

Considering which ones of the equations mentioned from page 179 on, being necessary for the calculation of the free and of the forced response, can be simplified in the same way, it can easily be seen that because of the restriction of \mathbf{F} to $\mathbf{F}_{N_{12}}$ not the complete matrix $\mathbf{F}\mathbf{G}_p$ is needed. Since, however, $\mathbf{Y}\mathbf{U}$ contains only the *past* values of $\mathbf{y}(t)$ and $\Delta\mathbf{u}(t)$, no simplification results for this matrix. Analogously, the number of *columns* of $\mathbf{F}\mathbf{G}_{pN_{12}}$ and $\mathbf{F}\mathbf{G}_p$ is identical, only the number of *rows* is reduced.

8.1.4 考虑控制步长后的简化

假设经过 N_u 步数的采样周期, 控制变量 $\mathbf{u}(t)$ 不再发生变化, 那么则称 N_u 为控制步长。由于 $N_u < t \leq N_2$ 范围内的控制变量不再发生变化, 也意味着 $\Delta\mathbf{u}(t) = 0$, 从而可以节省计算时间, 因为此时无需计算完整的预测矩阵。在满足 $N_u \leq N_2$ 这一假设条件下, 方程 (8.45) 可简化为方程 (8.48) 或 (8.49)。

从而, 章节 8.1.3 第 175 页介绍的用于计算最有控制变量的方程也是可以得到相应地简化。使用方程 (8.49) 代替 (8.25), 并将其代入至方程 (8.24), 从而可以得到一个简化的成本函数表达式, 见方程 (8.50)。

通过寻找上述成本函数的最小值, 即可最终求得在考虑了控制步长 N_u 后的最优控制变量, 见方程 (8.51)。显然, 此时也仅仅需要所谓的 $\Delta\mathbf{u}(t)$, 从而只需计算 \mathbf{U} 的前 m 行即可。

现在回到第 179 页, 用于计算自由响应和强制响应部分所需的方程也可同样得到简化。很明显, 由于矩阵多项式 \mathbf{F} 被限制在 $\mathbf{F}_{N_{12}}$, 那么, 整个矩阵 $\mathbf{F}\mathbf{G}_p$ 也是无需完全计算的。由于 $\mathbf{Y}\mathbf{U}$ 仅包含 $\mathbf{y}(t)$ 和 $\Delta\mathbf{u}(t)$ 过去时刻的值, 所以这一矩阵是无法进行简化的。同样, $\mathbf{F}\mathbf{G}_{pN_{12}}$ 和 $\mathbf{F}\mathbf{G}_p$ 的列数是相同的, 仅仅是行数缩短了。

Es ergibt sich folglich die Gleichung

$$\mathbf{F}_{N_{12}} = \mathbf{F}\mathbf{G}_{\mathbf{p}_{N_{12}}} \cdot \mathbf{Y}\mathbf{U} \quad (8.52)$$

wobei

$$\mathbf{F}\mathbf{G}_{\mathbf{p}_{N_{12}}} = \begin{bmatrix} \mathbf{F}_{N_{1,0}} & \cdots & \mathbf{F}_{N_{1,na}} & \mathbf{G}_{N_{1p,0}} & \cdots & \mathbf{G}_{N_{1p,nb-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_{2,0}} & \cdots & \mathbf{F}_{N_{2,na}} & \mathbf{G}_{N_{2p,0}} & \cdots & \mathbf{G}_{N_{2p,nb-1}} \end{bmatrix} \quad (8.53)$$

8.2 Störgrößenbehandlung mit GPC/Consideration of disturbance inputs with GPC/广义预测控制的扰动变量处理

Verwendet man MPC zur Regelung eines nichtlinearen Systems, z. B. einer Asynchronmaschine, so können die Nichtlinearitäten der zu regelnden Strecke auf zweierlei Weise berücksichtigt werden. Entweder versucht man, sie durch lineare Glieder zu approximieren, oder man verwendet eine nichtlineare Variante von MPC. Allerdings stellen nichtlineare MPC-Verfahren bisher noch ein weitgehend unerforschtes Gebiet dar. Es gibt zwar etliche Veröffentlichungen, die sich mit einer nichtlinearen MPC-Regelung beschäftigen bzw. entsprechende Regelverfahren vorstellen, trotzdem steckt diese Technologie noch ziemlich in den Kinderschuhen. Die Aussage von García, Prett und Morari von 1989 [40], dass in diesem Forschungsbereich kaum die Oberfläche angekratzt sei, ist zwar mittlerweile überholt, aber auch in neueren Veröffentlichungen wird nach wie vor dargelegt, dass es zwar experimentelle Untersuchungen und vereinzelt sogar industrielle Anwendungen von nichtlinearen MPC-Verfahren gibt, eine theoretische Untermauerung aber bisher im Gegensatz zu den linearen Strategien fast vollständig fehlt [88, 102]. Obwohl mittlerweile Fortschritte erzielt wurden, müssen nichtlineare modellbasierte Verfahren bevor sie sich stärker durchsetzen können, erst deutlich zuverlässiger, effizienter und ausfallsicherer werden. Eines der Hauptprobleme jeder nichtlinearen MPC-Regelung ist, dass der Beweis, dass die Regelstrategie stets das globale Minimum findet, nicht so trivial ist wie im linearen Fall und teilweise erst noch erbracht werden muss. Der einfache Schluss von der Betrachtung des offenen auf den geschlossenen Regelkreis funktioniert bei nichtlinearen Regelungen ebenfalls nicht, da das Superpositionsprinzip nur in linearen Systemen gilt. Leider würde es den Rahmen dieser Arbeit sprengen, ein nichtlineares Optimierverfahren inklusive nichtlinearem Modell und Stabilitätsnachweis herzuleiten. Als Fortsetzung des Projekts wäre aber die Entwicklung eines nichtlinearen MPC für elektrische Antriebe durchaus denkbar; Ansätze hierzu finden sich z. B. bei Morari/Lee [88].

Therefore the equation (8.52) results, in which equation (8.53).

8.2 Consideration of disturbance inputs with GPC If MPC is used for the control of a nonlinear system, e. g. an induction machine, the nonlinearities of the controlled system can be considered in two different ways. Either the nonlinear parts of the system are approximated with linear ones or a nonlinear variant of MPC can be used. Indeed, until now nonlinear MPC methods still represent a widely uninvestigated field. Certainly, there are several publications which deal with nonlinear MPC control or which introduce such control methods; nevertheless, this technology is still in its fledgling stages. Although the statement by García, Prett and Morari from 1989 [40] that in this research area barely the basics are understood is outdated now, since experimental investigations and sporadically even industrial applications of nonlinear MPC methods are demonstrated in newer publications. However, a theoretical foundation, in contrast to the linear strategies, is nearly completely absent until today [88, 102]. Although meanwhile progress was made, nonlinear model-based methods must become much more reliable, more efficient and failsafe before they will be widely accepted. One of the major problems of all nonlinear MPC control schemes is the proof that the control method always finds the global minimum which is not as trivial as in the linear case and partly proofs do still have to be deduced. Simply drawing the conclusions if considering the open and then the closed control loop does also not work for nonlinear controls, since the principle of superposition does only apply for linear systems. Unfortunately, it would go beyond the scope of this work to derive a nonlinear optimization method including a nonlinear model and a stability proof. However, as a continuation of this project the development of a nonlinear MPC for electric drives could be conceivable; approaches therefore can be found e. g. in the publication of Morari/Lee [88].

因此, 可得到方程 (8.52), 其中, 有方程 (8.53)。

8.2 广义预测控制的扰动变量处理 假使采用 MPC 控制一个非线性系统, 如异步电机, 那么可以从两个角度来考虑被控对象的非线性特性: 一方面可以使用线性环节作近似化处理; 另一方面可以使用 MPC 的非线性变型形式。到目前为止, 非线性的 MPC 方法仍然是一个几乎未被探索的领域。虽说也存在一定数量的学术论文涉及非线性 MPC 控制这一主题, 并提出了一些基本的控制方法, 但它仍然处于所谓的童年阶段。早在 1989 年, 学者 García, Prett 和 Morari 的观点是 [40]: 在这一研究领域, 几乎可以说还未能触及到最浅显的表面。即使现在有了一定的进展, 但最新出现的学术论文所涉及的也仅是实践层面的考察, 也有个别的将非线性 MPC 方法应用至工业领域的案例, 至于理论层面的突破几乎是完全没有的 [88, 102]。即使存在了一定的进展, 但在真正应用非线性 MPC 方法之前, 还需要持续不断地研究以使其更加可靠、有效和安全。任一非线性 MPC 控制的一大主要问题是: 控制策略是否始终处于全局最优的证明并不像线性控制时那样简单, 而且部分证明还需首先进行严格推导。在非线性的情况下, 适用于开环及闭环控制的一些简单结论将不再有效, 因为叠加原理仅适用于线性系统。遗憾的是, 在本书的研究框架范围内也未能涉及非线性优化方法的推导, 例如非线性模型和稳定性证明等。当然, 将后续的研究主题扩展至电力驱动领域的非线性 MPC 方法探究是值得期待的, 例如, 学者 Morari/Lee 的论文 [88] 也对可能的研究方向做出了展望。

Bei der Regelung eines Antriebs mit einer Asynchronmaschine bietet es sich anstelle einer nichtlinearen Regelung an, die multiplikativ verknüpften Kreuzkopplungen, die als Nichtlinearität im linearen CARIMA-Modell nicht eingeschlossen werden können, als Störeinflüsse zu berücksichtigen.

8.2.1 Bestimmung der Übertragungsfunktion/Determination of the transfer function/传递函数的确定

Ausgehend von der Zustandsraumbeschreibung in zeitdiskreter Darstellung unter Berücksichtigung der Störeinflüsse gemäß Gln. (7.42) und (7.43) wird nun analog zur Übertragungsmatrix $\mathbf{G}_d(z^{-1})$ in Gl. (8.2) eine Störübertragungsmatrix $\mathbf{H}_d(z^{-1})$ wie folgt definiert:

$$\mathbf{H}_d(z^{-1}) = \mathbf{H}_{d0} + \mathbf{H}_{d1}z^{-1} + \mathbf{H}_{d2}z^{-2} + \dots + \mathbf{H}_{dn}z^{-n} = \frac{\mathbf{Y}(z)}{\mathbf{V}(z)} \quad (8.54)$$

wobei die einzelnen Elemente folgende Dimensionen besitzen:

$$\begin{aligned} \mathbf{y}(t) &: \text{Vektor}(n \times 1) && \text{mit } n = \text{Anzahl der Systemausgänge} \\ \mathbf{v}(t) &: \text{Vektor}(l \times 1) && \text{mit } l = \text{Anzahl der Eingänge der Störeinflüsse} \\ \mathbf{H}_d(z^{-1}) &: \text{Matrix}(n \times l) \end{aligned}$$

Mit Hilfe der in Kap. 8.1.1 für die Übertragungsmatrix $\mathbf{G}_d(z^{-1})$ beschriebenen Vorgehensweise erhält man genauso die Störübertragungsmatrix $\mathbf{H}_d(z^{-1})$

$$\mathbf{H}_d(z^{-1}) = \mathbf{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1}\mathbf{E}_d + \mathbf{F}_d \quad (8.55)$$

Somit existieren für die Stell- und für die Störgrößen zwei mathematisch getrennte Systeme, deren Ausgänge sich überlagern.

8.2.2 Berechnung der Systemmatrizen/Calculation of the system matrices/系统矩阵的计算

Wendet man die in Kap. 8.1.2 für die Berechnung der Polynommatrizen $\mathbf{A}(z^{-1})$ und $\mathbf{B}(z^{-1})$ beschriebenen Überlegungen auf die für die Berücksichtigung der Störeinflüsse im CARIMA-Modell notwendige Matrix $\mathbf{D}(z^{-1})$ an, so muss gelten:

$$\mathbf{H}_d(z^{-1}) = \mathbf{A}(z^{-1})^{-1}\mathbf{D}(z^{-1})z^{-1} \quad (8.56)$$

Da $\mathbf{A}(z^{-1})$ bereits aus 8.1.2 bekannt ist, kann $\mathbf{D}(z^{-1})$ direkt mit Hilfe von

$$\mathbf{D}(z^{-1}) = \mathbf{A}(z^{-1})\mathbf{H}_d(z^{-1})z \quad (8.57)$$

berechnet werden.

If a drive with an induction machine has to be controlled, it makes sense to consider the multiplicatively linked cross couplings which cannot be included as non-linearities in the linear CARIMA model as disturbances instead of implementing a nonlinear control strategy.

8.2.1 Determination of the transfer function Based on the time-discrete state space representation considering the disturbances according to the equations (7.42) and (7.43) now analog to the transfer function matrix $\mathbf{G}_d(z^{-1})$ in equation (8.2), a disturbance transfer function matrix $\mathbf{H}_d(z^{-1})$ is defined, see equation (8.54), whereas the individual elements have the following dimensions:

$$\begin{aligned} \mathbf{y}(t) &: \text{Vector}(n \times 1) && \text{with } n = \text{number of system outputs} \\ \mathbf{v}(t) &: \text{Vector}(l \times 1) && \text{with } l = \text{number of disturbance inputs} \\ \mathbf{H}_d(z^{-1}) &: \text{Matrix}(n \times l) \end{aligned}$$

With the help of the approach for the transfer function matrix $\mathbf{G}_d(z^{-1})$, described in Chapter 8.1.1, the disturbance transfer function matrix $\mathbf{H}_d(z^{-1})$ can be obtained in the same way, see equation (8.55). Thus, two mathematically separated systems with overlapping outputs exist for the actuating and for the disturbance variables.

8.2.2 Calculation of the system matrices If the statements for the calculation of the polynomial matrices $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$, described in Chapter 8.1.2, are applied to the matrix $\mathbf{D}(z^{-1})$ which is necessary for the consideration of disturbances in the CARIMA model, the following must be effective, see equation (8.56). Since $\mathbf{A}(z^{-1})$ is already known from 8.1.2, $\mathbf{D}(z^{-1})$ can be calculated directly with the help of equation (8.57).

那么在实现异步电机的驱动控制时，由于无法将多重交叉耦合这一非线性因素考虑至线性的 CARIMA 模型，所以需将其看作扰动因素以代替所谓的非线性控制。

8.2.1 传递函数的确定 根据方程 (7.42) 和 (7.43) 所示的考虑扰动因素后的时间离散状态空间描述形式，并类似于方程 (8.2) 所示的传递矩阵 $\mathbf{G}_d(z^{-1})$ 形式，此处定义一个扰动传递矩阵 $\mathbf{H}_d(z^{-1})$ ，见方程 (8.54)，其中，单个元素的维度分别为：

$$\begin{aligned} \mathbf{y}(t) &: \text{向量}(n \times 1) && \text{其中, } n = \text{系统输出的数量} \\ \mathbf{v}(t) &: \text{向量}(l \times 1) && \text{其中, } l = \text{系统输入的数量} \\ \mathbf{H}_d(z^{-1}) &: \text{矩阵}(n \times l) \end{aligned}$$

根据章节 8.1.1 所述的对传递矩阵 $\mathbf{G}_d(z^{-1})$ 的处理方法，同样可以求得扰动传递矩阵 $\mathbf{H}_d(z^{-1})$ ，见方程 (8.55)。从而，存在两个数学上相互独立的可分别用于控制变量和扰动变量的系统，它们的输出则适用于叠加原理。

8.2.2 系统矩阵的计算 如果将章节 8.1.2 所述的用于计算多项式矩阵 $\mathbf{A}(z^{-1})$ 和 $\mathbf{B}(z^{-1})$ 的方法应用到 CARIMA 模型中考虑扰动变量后的矩阵 $\mathbf{D}(z^{-1})$ 的计算，那么则同样满足方程 (8.56)。由于 $\mathbf{A}(z^{-1})$ 已由章节 8.1.2 所求得，所以可直接借助于方程 (8.57) 求得 $\mathbf{D}(z^{-1})$ 。

8.2.3 Mathematische Herleitung/Mathematical derivation/数学推导

Das CARIMA-Modell

Zur Berücksichtigung der Störgrößen wird das multidimensionale CARIMA-Modell gegenüber Gl. (8.12) um einen deterministischen Störterm ergänzt:

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = \mathbf{B}(z^{-1})\mathbf{u}(t-1) + \mathbf{D}(z^{-1})\mathbf{v}(t) + \mathbf{C}(z^{-1})\frac{\boldsymbol{\xi}(t)}{\Delta} \quad (8.58)$$

mit

$$\mathbf{D}(z^{-1}) = \mathbf{D}_0 + \mathbf{D}_1z^{-1} + \mathbf{D}_2z^{-2} + \dots + \mathbf{D}_{nd}z^{-nd} \quad (8.59)$$

Die Dimensionen von $\mathbf{D}(z^{-1})$ und $\mathbf{v}(t)$ betragen dabei:

$$\begin{aligned} \mathbf{D}(z^{-1}) &: \text{Matrix}(n \times l) \\ \mathbf{v}(t) &: \text{Vektor}(l \times 1) \end{aligned}$$

Wie schon in den vorhergehenden Fällen, wird auch hier wieder davon ausgegangen, dass $\boldsymbol{\xi}(t)$ weißes Rauschen darstellt; $\mathbf{C}(z^{-1})$ kann folglich als Einheitsmatrix \mathbf{I} gesetzt werden. Dadurch wird Gl. (8.58) zu

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = \mathbf{B}(z^{-1})\mathbf{u}(t-1) + \mathbf{D}(z^{-1})\mathbf{v}(t) + \frac{\boldsymbol{\xi}(t)}{\Delta} \quad (8.60)$$

Der j-Schritt-voraus-Prädiktor

Berücksichtigt man die Störgrößen, so verändert sich die bekannte Diophantische Gleichung gegenüber der klassischen Variante nicht. Die Vorausberechnung der zukünftigen Systemausgänge kann auch hier über die Formel (8.14) hergeleitet werden. Multipliziert man Gl. (8.60) mit $\Delta\mathbf{E}_j(z^{-1})z^j$ und setzt die nach $\mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})$ aufgelöste Gl. (8.14) ein, so ergibt sich für die Prädiktion

$$\begin{aligned} \mathbf{y}(t+j) &= \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) \\ &\quad + \mathbf{E}_j(z^{-1})\mathbf{D}(z^{-1})\Delta\mathbf{v}(t+j) \\ &\quad + \mathbf{F}_j(z^{-1})\mathbf{y}(t) \\ &\quad + \mathbf{E}_j(z^{-1})\boldsymbol{\xi}(t+j) \end{aligned} \quad (8.61)$$

Wiederum kann man die in der Zukunft liegenden Rauschterme vernachlässigen und erhält daher

$$\begin{aligned} \hat{\mathbf{y}}(t+j) &= \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) \\ &\quad + \mathbf{E}_j(z^{-1})\mathbf{D}(z^{-1})\Delta\mathbf{v}(t+j) \\ &\quad + \mathbf{F}_j(z^{-1})\mathbf{y}(t) \end{aligned} \quad (8.62)$$

8.2.3 Mathematical derivation

The CARIMA model

The multidimensional CARIMA model in equation (8.12) is complemented with a deterministic disturbance term so that disturbance variables can be considered, see equation (8.58), with equation (8.59). The dimensions of $\mathbf{D}(z^{-1})$ and $\mathbf{v}(t)$ are:

$$\begin{aligned}\mathbf{D}(z^{-1}) &: \text{Matrix}(n \times l) \\ \mathbf{v}(t) &: \text{Vector}(l \times 1)\end{aligned}$$

Like in the previous cases, it is also assumed here that $\boldsymbol{\xi}(t)$ represents white noise; therefore $\mathbf{C}(z^{-1})$ is equal to the identity matrix \mathbf{I} . Thus, equation (8.58) becomes to equation (8.60).

The j-step ahead predictor

If the disturbance variables are considered, the well-known Diophantine equation does not change in comparison to the classical variant. The precalculation of future system outputs can here also be derived via equation (8.14). Multiplicating equation (8.60) with $\Delta \mathbf{E}_j(z^{-1})z^j$ and applying equation (8.14) solved for $\mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})$, the following results for the prediction, see equation (8.61). Again, the noise terms being in the future can be neglected and thus, equation (8.62) can be obtained.

8.2.3 数学推导

CARIMA 模型

相较于方程 (8.12), 考虑扰动变量后的多维 CARIMA 模型多了一个确定的干扰项, 见方程 (8.58), 其中, 有方程 (8.59)。 $\mathbf{D}(z^{-1})$ 和 $\mathbf{v}(t)$ 维数分别为:

$$\begin{aligned}\mathbf{D}(z^{-1}) &: \text{矩阵}(n \times l) \\ \mathbf{v}(t) &: \text{向量}(l \times 1)\end{aligned}$$

正如前面所述情形, 此处也同样认为 $\boldsymbol{\xi}(t)$ 表示白噪声, 从而可将矩阵多项式 $\mathbf{C}(z^{-1})$ 设置为单位矩阵 \mathbf{I} 。所以, 方程 (8.58) 变为方程 (8.60)。

J 步前向预测器

相对于传统的方法来说, 即使考虑扰动变量, 已知的丢番图方程并不发生变化。未来的系统输出预测也可根据方程 (8.14) 推导求得。若将方程 (8.60) 两侧同时乘以 $\Delta \mathbf{E}_j(z^{-1})z^j$, 并将其代入至将方程 (8.14) 向 $\mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})$ 求解后的表达式, 即可得到预测表达式, 见方程 (8.61)。可再次忽略未来时刻的噪声项, 从而可得到方程 (8.62)。

Analog zu (8.17) wird nun zusätzlich

$$\mathbf{E}_j(z^{-1})\mathbf{D}(z^{-1}) = \mathbf{H}_j(z^{-1}) + z^{-j}\mathbf{H}_{jp}(z^{-1}) \quad \text{Grad}(\mathbf{H}_j(z^{-1})) < j \quad (8.63)$$

definiert. Da die Matrizenpolynome $\mathbf{H}_j(z^{-1})$ und $\mathbf{H}_{jp}(z^{-1})$ für die Störgrößen exakt $\mathbf{G}_j(z^{-1})$ und $\mathbf{G}_{jp}(z^{-1})$ entsprechen, gilt auch das in Kap. 8.1.3 auf Seite 171ff. bezüglich der Zerlegung in zukünftige und vergangene Werte gesagte. Demnach wird Formel (8.62) zu

$$\begin{aligned} \hat{\mathbf{y}}(t+j) &= \underbrace{\mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1)}_{\text{erzwungene Antwort}} + \underbrace{\mathbf{H}_j(z^{-1})\Delta\mathbf{v}(t+j)}_{\substack{\text{zukünftige, determ.} \\ \text{Störeinflüsse}}} \\ &+ \underbrace{\mathbf{G}_{jp}(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_{jp}(z^{-1})\Delta\mathbf{v}(t) + \mathbf{F}_j(z^{-1})\mathbf{y}(t)}_{\substack{\text{alles Werte in der Vergangenheit} \\ \rightarrow \text{freie Antwort } \mathbf{f}(t+j)}} \end{aligned} \quad (8.64)$$

$$\hat{\mathbf{y}}(t+j) = \mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{H}_j(z^{-1})\Delta\mathbf{v}(t+j) + \mathbf{f}(t+j) \quad (8.65)$$

mit

$$\mathbf{f}(t+j) = \mathbf{G}_{jp}(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_{jp}(z^{-1})\Delta\mathbf{v}(t) + \mathbf{F}_j(z^{-1})\mathbf{y}(t) \quad (8.66)$$

Dimensionen:

$$\begin{aligned} \mathbf{G}_j(z^{-1}), \mathbf{G}_{jp}(z^{-1}) &: \text{Matrix}(n \times m) && \text{(unverändert)} \\ \mathbf{H}_j(z^{-1}), \mathbf{H}_{jp}(z^{-1}) &: \text{Matrix}(n \times l) \\ \mathbf{f}(t) &: \text{Vektor}(n \times 1) \end{aligned}$$

Die einzelnen Prädiktionsschritte lauten

$$\hat{\mathbf{y}}(t+1) = \mathbf{G}_1(z^{-1})\Delta\mathbf{u}(t) + \mathbf{H}_1(z^{-1})\Delta\mathbf{v}(t+1) + \mathbf{f}(t+1) \quad (8.67)$$

$$\hat{\mathbf{y}}(t+2) = \mathbf{G}_2(z^{-1})\Delta\mathbf{u}(t+1) + \mathbf{H}_2(z^{-1})\Delta\mathbf{v}(t+2) + \mathbf{f}(t+2) \quad (8.68)$$

$$\hat{\mathbf{y}}(t+3) = \mathbf{G}_3(z^{-1})\Delta\mathbf{u}(t+2) + \mathbf{H}_3(z^{-1})\Delta\mathbf{v}(t+3) + \mathbf{f}(t+3) \quad (8.69)$$

⋮

$$\hat{\mathbf{y}}(t+N_p) = \mathbf{G}_{N_p}(z^{-1})\Delta\mathbf{u}(t+N_p-1) + \mathbf{H}_{N_p}(z^{-1})\Delta\mathbf{v}(t+N_p) + \mathbf{f}(t+N_p) \quad (8.70)$$

Now, analog to (8.17), equation (8.63) is defined additionally. Since the matrix polynomials $\mathbf{H}_j(z^{-1})$ and $\mathbf{H}_{jp}(z^{-1})$ for the disturbances exactly correspond to $\mathbf{G}_j(z^{-1})$ and $\mathbf{G}_{jp}(z^{-1})$, the statements in Chapter 8.1.3 on page 171 et seqq. concerning the decomposition in future and past values apply here, too. Therefore, equation (8.62) becomes to

$$\begin{aligned} \hat{\mathbf{y}}(t+j) = & \underbrace{\mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1)}_{\text{forced response}} + \underbrace{\mathbf{H}_j(z^{-1})\Delta\mathbf{v}(t+j)}_{\text{future, deterministic disturbances}} \\ & + \underbrace{\mathbf{G}_{jp}(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_{jp}(z^{-1})\Delta\mathbf{v}(t) + \mathbf{F}_j(z^{-1})\mathbf{y}(t)}_{\substack{\text{all values in the past} \\ \rightarrow \text{free response } \mathbf{f}(t+j)}} \end{aligned} \quad (8.64)$$

further, see equation (8.65) with equation (8.66).

Dimensions:

$$\begin{aligned} \mathbf{G}_j(z^{-1}), \mathbf{G}_{jp}(z^{-1}) &: \text{Matrix}(n \times m) && \text{(unchanged)} \\ \mathbf{H}_j(z^{-1}), \mathbf{H}_{jp}(z^{-1}) &: \text{Matrix}(n \times l) \\ \mathbf{f}(t) &: \text{Vector}(n \times 1) \end{aligned}$$

The individual prediction steps are equations (8.67) to (8.70).

类似于方程 (8.17), 现在额外定义方程 (8.63)。由于表示扰动变量的矩阵多项式 $\mathbf{H}_j(z^{-1})$ 和 $\mathbf{H}_{jp}(z^{-1})$ 与 $\mathbf{G}_j(z^{-1})$ 和 $\mathbf{G}_{jp}(z^{-1})$ 严格对应, 那么, 此时也同样满足章节 8.1.3 第 171 页所述将预测值分解为过去的和未来的两部分的处理方式。从而, 方程 (8.62) 变化为:

$$\begin{aligned} \hat{\mathbf{y}}(t+j) = & \underbrace{\mathbf{G}_j(z^{-1})\Delta\mathbf{u}(t+j-1)}_{\text{强制响应}} + \underbrace{\mathbf{H}_j(z^{-1})\Delta\mathbf{v}(t+j)}_{\substack{\text{未来的、确定的} \\ \text{扰动}}} \\ & + \underbrace{\mathbf{G}_{jp}(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_{jp}(z^{-1})\Delta\mathbf{v}(t) + \mathbf{F}_j(z^{-1})\mathbf{y}(t)}_{\substack{\text{所有过去的数值} \\ \rightarrow \text{自由响应 } \mathbf{f}(t+j)}} \end{aligned} \quad (8.64)$$

更进一步, 见方程 (8.65), 其中, 有方程 (8.66)。

维数:

$$\begin{aligned} \mathbf{G}_j(z^{-1}), \mathbf{G}_{jp}(z^{-1}) &: \text{矩阵}(n \times m) && \text{(不变化)} \\ \mathbf{H}_j(z^{-1}), \mathbf{H}_{jp}(z^{-1}) &: \text{矩阵}(n \times l) \\ \mathbf{f}(t) &: \text{向量}(n \times 1) \end{aligned}$$

每一步的预测表达式可见方程 (8.67) 至 (8.70)。

Berechnung der Steuergrößen

Da sich an der mathematischen Grundlage der optimalen Steuergrößen, einer quadratischen Kostenfunktion, nichts geändert hat, bleibt Formel (8.24) unverändert gültig. Nach einer Zusammenfassung von Gl. (8.65) analog zum Vorgehen in Kap. 8.1.3 auf Seite 175 ergibt sich für die Prädiktion

$$\mathbf{Y} = \mathbf{G}\mathbf{U} + \mathbf{H}\mathbf{V} + \mathbf{F} \quad (8.71)$$

mit $\mathbf{U} = \Delta\mathbf{U}$ und $\mathbf{V} = \Delta\mathbf{V}$. Da der Störterm $\mathbf{H}\mathbf{V}$ von den Stellgrößen unabhängig ist, kann er als eine Art freie Antwort aufgefasst werden. Demnach werden $\mathbf{H}\mathbf{V}$ und \mathbf{F} zu einer neuen freien Antwort $\mathbf{F}' = \mathbf{H}\mathbf{V} + \mathbf{F}$ zusammengefasst. Damit existiert eine vereinfachte Prädiktionsgleichung

$$\mathbf{Y} = \mathbf{G}\mathbf{U} + \mathbf{F}' \quad (8.72)$$

Unter diesen Voraussetzungen lassen sich die optimalen, zukünftigen Stellgrößen wie im Fall ohne Berücksichtigung der bekannten Störeinflüsse berechnen, lediglich die dort verwendete freie Antwort \mathbf{F} ist in Formel (8.27) durch die modifizierte Version \mathbf{F}' zu ersetzen. Die optimalen Reglerausgänge lassen sich, sofern die üblichen Vereinfachungen $\mu_j = 1$ und $\lambda_j = \lambda$ für alle j vorgenommen werden, folglich mit

$$\mathbf{U} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}^T(\mathbf{W} - \mathbf{F}') \quad (8.73)$$

berechnen.

Rekursion der Diophantischen Gleichung

An der zur Ermittlung der Polynommatrizen für die Prädiktion notwendigen Diophantischen Gleichung hat sich gegenüber MIMO-GPC ohne Berücksichtigung der Störgrößen nichts geändert. Gl. (8.14) ist unverändert gültig und daher auch die in Kap. 8.1.3 auf Seite 177 f. beschriebene Vorgehensweise.

Berechnung der freien und der erzwungenen Antwort

Vergleicht man Gl. (8.64) und (8.65) mit (8.18) und (8.19), so zeigt sich, dass die Polynommatrizen $\mathbf{G}_j(z^{-1})$ und $\mathbf{G}_{jp}(z^{-1})$ unverändert übernommen werden können. Daher müssen nur $\mathbf{H}_j(z^{-1})$ und $\mathbf{H}_{jp}(z^{-1})$ neu hergeleitet werden. Da die Definitionen der oben genannten Polynommatrizen in Gl. (8.63) bzw. (8.17) sich absolut ähneln, muss $\mathbf{H}_j(z^{-1})$ wegen der rekursiven Eigenschaften von $\mathbf{E}_j(z^{-1})$ ähnlich $\mathbf{G}_j(z^{-1})$ aufgebaut sein.

Calculation of the control variables Since nothing has changed in the mathematical basis for the calculation of optimum values for the actuating variables, which is still

a quadratic cost function, equation (8.24) remains valid without any changes. After a combination of equation (8.65) analog to the procedure in Chapter 8.1.3 on page 175, the prediction equation results in equation (8.71), with $\mathbf{U} = \Delta \mathbf{U}$ and $\mathbf{V} = \Delta \mathbf{V}$. Since the disturbance term $\mathbf{H}\mathbf{V}$ is independent of the actuating variables, it can be considered as a kind of free response term. Therefore, $\mathbf{H}\mathbf{V}$ and \mathbf{F} are combined to a new free response $\mathbf{F}' = \mathbf{H}\mathbf{V} + \mathbf{F}$. Thus, a simplified prediction equation results, see equation (8.72).

Taking these assumptions into account, optimum future values for the actuating variables can be calculated in the same way as without consideration of known disturbances, only the free response \mathbf{F} used there has to be replaced with its modified version \mathbf{F}' . The optimum controller outputs can be calculated with equation (8.73), if the usual simplifications, $\mu_j = 1$ and $\lambda_j = \lambda$ for all j , are applied.

Recursion of the Diophantine equation The Diophantine equation that is necessary for the determination of the polynomial matrices for the prediction has not changed compared to MIMO-GPC without consideration of disturbances. Equation (8.14) is valid without any modifications and hence also the approach described in Chapter 8.1.3 on page 177 et seq.

Calculation of the free and forced response Comparing equation (8.64) and (8.65) to (8.18) and (8.19) it can be seen that the polynomial matrices $\mathbf{G}_j(z^{-1})$ and $\mathbf{G}_{jp}(z^{-1})$ can be taken without any changes. Hence, only $\mathbf{H}_j(z^{-1})$ and $\mathbf{H}_{jp}(z^{-1})$ have to be derived newly. Since the definitions of the polynomial matrices mentioned above in equation (8.63) and (8.17) are very similar to each other, $\mathbf{H}_j(z^{-1})$ has to be similar to $\mathbf{G}_j(z^{-1})$ because of the recursive properties of $\mathbf{E}_j(z^{-1})$.

控制变量的计算 由于确定最优控制变量的基础并未变化, 也即基于一个二次成本函数, 所以方程 (8.24) 依然有效。类似于章节 8.1.3 第 175 页所述的处理方式, 对方程 (8.65) 经过适当处理后, 可得方程 (8.71), 其中 $\mathbf{U} = \Delta \mathbf{U}$ 和 $\mathbf{V} = \Delta \mathbf{V}$ 。由于扰动项 $\mathbf{H}\mathbf{V}$ 不受控制变量的影响, 所以可将其看作是自由响应的一种形式。随后, 将 $\mathbf{H}\mathbf{V}$ 和 \mathbf{F} 整合为一个新的自由响应 $\mathbf{F}' = \mathbf{H}\mathbf{V} + \mathbf{F}$, 从而得到了一个简化的预测方程, 见方程 (8.72)。

在上述前提条件下, 最优的、未来控制变量的计算则类似于未考虑已知干扰因素的情形, 不同的是需将方程 (8.27) 中的自由响应 \mathbf{F} 由一个修正的 \mathbf{F}' 所替代。只要对于所有的 j 来说, 满足一般的简化条件 $\mu_j = 1$ 和 $\lambda_j = \lambda$, 即可求得最优的控制变量表达式, 见方程 (8.73)。

丢番图方程的递归 相较于未考虑干扰的 MIMO-GPC, 考虑干扰后的用于预测所需的丢番图方程并未发生变化, 所以方程 (8.14) 依然保持有效。递归方法类似于章节 8.1.3 第 177 所描述的方式, 此处不再赘述。

自由响应与强制响应的计算 对比方程 (8.64) 和 (8.65) 与方程 (8.18) 和 (8.19) 可知, 多项式矩阵 $\mathbf{G}_j(z^{-1})$ 和 $\mathbf{G}_{jp}(z^{-1})$ 并未发生变化, 所以可以直接使用。因此, 只需重新计算 $\mathbf{H}_j(z^{-1})$ 和 $\mathbf{H}_{jp}(z^{-1})$ 即可。由于方程 (8.63) 和 (8.17) 中多项式矩阵的定义非常相似, 考虑到 $\mathbf{E}_j(z^{-1})$ 的递归特性, 从而可以类似于 $\mathbf{G}_j(z^{-1})$ 的方式来构造 $\mathbf{H}_j(z^{-1})$ 。

Daher muss gelten:

$$\mathbf{H}_j(z^{-1}) = \mathbf{H}_0 + \mathbf{H}_1 z^{-1} + \mathbf{H}_2 z^{-2} + \dots + \mathbf{H}_{j-1} z^{-(j-1)} \quad j = 1 \dots N_p$$

mit

$$\mathbf{H}_j = \sum_{k=0}^j \mathbf{E}_{j-k} \mathbf{D}_k$$

Auch hier entfallen alle Summanden mit $k > nd$. Gl. (8.71) lässt sich demnach ausführlich schreiben:

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{y}}(t+1) \\ \vdots \\ \hat{\mathbf{y}}(t+j) \\ \vdots \\ \hat{\mathbf{y}}(t+N_p) \end{bmatrix} &= \begin{bmatrix} \mathbf{G}_0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}_{j-1} & \cdots & \mathbf{G}_0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}_{N_p-1} & \cdots & \mathbf{G}_{N_p-j} & \cdots & \mathbf{G}_0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}(t) \\ \vdots \\ \Delta \mathbf{u}(t+j-1) \\ \vdots \\ \Delta \mathbf{u}(t+N_p-1) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{H}_0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{H}_{j-1} & \cdots & \mathbf{H}_0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{H}_{N_p-1} & \cdots & \mathbf{H}_{N_p-j} & \cdots & \mathbf{H}_0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{v}(t+1) \\ \vdots \\ \Delta \mathbf{v}(t+j) \\ \vdots \\ \Delta \mathbf{v}(t+N_p) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{f}(t+1) \\ \vdots \\ \mathbf{f}(t+j) \\ \vdots \\ \mathbf{f}(t+N_p) \end{bmatrix} \end{aligned} \quad (8.74)$$

Zur Berechnung der freien Antwort benötigt man nach Gl. (8.64) neben den bereits bekannten Größen $\mathbf{G}_{jp}(z^{-1})$ und $\mathbf{F}_j(z^{-1})$ noch $\mathbf{H}_{jp}(z^{-1})$. Da sich auch dieser Wert gemäß (8.63) analog zu $\mathbf{G}_{jp}(z^{-1})$ berechnet, gilt folglich für die einzelnen Elemente von $\mathbf{H}_{jp}(z^{-1})$:

$$\mathbf{H}_{jp,i} = \sum_{k=0}^{j-1} \mathbf{E}_k \mathbf{D}_{j+i-k} \quad j = 1 \dots N_p; \quad i = 0 \dots (nd-1) \quad (8.75)$$

Auch hier entfallen wieder die Summanden mit $(j+i-k) > nd$.

Therefore, the following must apply:

$$\mathbf{H}_j(z^{-1}) = \mathbf{H}_0 + \mathbf{H}_1 z^{-1} + \mathbf{H}_2 z^{-2} + \cdots + \mathbf{H}_{j-1} z^{-(j-1)} \quad j = 1 \dots N_p$$

with

$$\mathbf{H}_j = \sum_{k=0}^j \mathbf{E}_{j-k} \mathbf{D}_k$$

All summands with $k > nd$ are omitted as well. Therefore, equation (8.71) can be written in detail as equation (8.74).

Besides the already known terms $\mathbf{G}_{jp}(z^{-1})$ and $\mathbf{F}_j(z^{-1})$ after equation (8.64), the term $\mathbf{H}_{jp}(z^{-1})$ is also needed for the calculation of the free response. Since this term is also calculated according to (8.63) analog to the calculation of $\mathbf{G}_{jp}(z^{-1})$, the following applies for the individual elements of $\mathbf{H}_{jp}(z^{-1})$, see equation (8.75). Summands with $(j + i - k) > nd$ are omitted again.

因此, 必须满足

$$\mathbf{H}_j(z^{-1}) = \mathbf{H}_0 + \mathbf{H}_1 z^{-1} + \mathbf{H}_2 z^{-2} + \cdots + \mathbf{H}_{j-1} z^{-(j-1)} \quad j = 1 \dots N_p$$

其中

$$\mathbf{H}_j = \sum_{k=0}^j \mathbf{E}_{j-k} \mathbf{D}_k$$

此处, 当 $k > nd$ 时, 也消去相应的项。从而, 可将方程 (8.71) 具体表示为方程 (8.74)。

根据方程 (8.64) 可知, 为了计算自由响应, 除了已知的 $\mathbf{G}_{jp}(z^{-1})$ 和 $\mathbf{F}_j(z^{-1})$ 之外, 还需 $\mathbf{H}_{jp}(z^{-1})$ 。由于 $\mathbf{H}_{jp}(z^{-1})$ 也可类似于 $\mathbf{G}_{jp}(z^{-1})$ 的方式计算, 从而, $\mathbf{H}_{jp}(z^{-1})$ 中的每一元素均满足下述表达式, 见方程 (8.75)。此处, 当 $(j + i - k) > nd$ 时, 也应消去相应的项。

Führt man nun die Koeffizienten der Polynommatrizen $\mathbf{G}_p(z^{-1})$, $\mathbf{H}_p(z^{-1})$ und $\mathbf{F}(z^{-1})$ zusammen, so ergibt sich

$$\mathbf{F} = \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{G}_p(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_p(z^{-1})\Delta\mathbf{v}(t) = \mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{YUV} \quad (8.76)$$

wobei zusätzlich zu den bisher bereits bekannten Definitionen von $\mathbf{F}(z^{-1})$ und $\mathbf{G}(z^{-1})$ nun noch

$$\mathbf{H}_p(z^{-1}) = \begin{bmatrix} \mathbf{H}_{1p}(z^{-1}) \\ \vdots \\ \mathbf{H}_{N_p p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{1p,0} + \mathbf{H}_{1p,1}z^{-1} + \cdots + \mathbf{H}_{1p,nd-1}z^{-(nd-1)} \\ \vdots \\ \mathbf{H}_{N_p p,0} + \mathbf{H}_{N_p p,1}z^{-1} + \cdots + \mathbf{H}_{N_p p,nd-1}z^{-(nd-1)} \end{bmatrix} \quad (8.77)$$

$$\mathbf{F}\mathbf{G}_p\mathbf{H}_p = \begin{bmatrix} \mathbf{F}_{1,0} & \cdots & \mathbf{F}_{1,na} & \mathbf{G}_{1p,0} & \cdots & \mathbf{G}_{1p,nb-1} & \mathbf{H}_{1p,0} & \cdots & \mathbf{H}_{1p,nd-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \cdots & \mathbf{F}_{N_p,na} & \mathbf{G}_{N_p p,0} & \cdots & \mathbf{G}_{N_p p,nb-1} & \mathbf{H}_{N_p p,0} & \cdots & \mathbf{H}_{N_p p,nd-1} \end{bmatrix} \quad (8.78)$$

$$\mathbf{YUV} = \begin{bmatrix} \mathbf{y}(t) \\ \vdots \\ \mathbf{y}(t-na) \\ \Delta\mathbf{u}(t-1) \\ \vdots \\ \Delta\mathbf{u}(t-nb) \\ \Delta\mathbf{v}(t) \\ \vdots \\ \Delta\mathbf{v}(t-nd+1) \end{bmatrix} \quad (8.79)$$

gilt. Zur weiteren Vereinfachung der Matrizenoperationen kann – wie bereits erläutert – nochmals zusammengefasst werden $\mathbf{F}' = \mathbf{H}\mathbf{V} + \mathbf{F}$. Dies macht allerdings nur dann Sinn, wenn die zukünftigen Werte der Störeinflüsse $\mathbf{v}(t)$ bekannt sind. Sind sie unbekannt, so wählt man sinnvollerweise für alle zukünftigen Werte den aktuellen Wert, d. h. $\mathbf{v}(t+j) = \mathbf{v}(t)$. Dadurch werden die Änderungen $\Delta\mathbf{v}(t+j)$ alle zu Null, wodurch der komplette Term $\mathbf{H}\mathbf{V}$ verschwindet. In diesen Fällen ist die Berechnung einer modifizierten freien Antwort \mathbf{F}' überflüssig; es wird einfach \mathbf{F} statt \mathbf{F}' in Formel (8.73) eingesetzt. Sind die zukünftigen Störeinflüsse jedoch explizit bekannt, so sollte \mathbf{F}' genau bestimmt werden.

If the coefficients of the polynomial matrices $\mathbf{G}_p(z^{-1})$, $\mathbf{H}_p(z^{-1})$ and $\mathbf{F}(z^{-1})$ are combined together into a single matrix, this results in:

$$\mathbf{F} = \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{G}_p(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_p(z^{-1})\Delta\mathbf{v}(t) = \mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{YUV} \quad (8.76)$$

in which, in addition to the already known definitions $\mathbf{F}(z^{-1})$ and $\mathbf{G}(z^{-1})$, the terms are valid, see equation (8.77) to (8.79).

For a further simplification of the matrix operations, $\mathbf{F}' = \mathbf{H}\mathbf{V} + \mathbf{F}$ can be summarized again as already explained. However, this does only make sense if the future values of the disturbances $\mathbf{v}(t)$ are known. If they are unknown, it is more reasonable to choose the actual value for all future values, i. e. $\mathbf{v}(t+j) = \mathbf{v}(t)$. Thereby, the changes $\Delta\mathbf{v}(t+j)$ become zero which leads to the fact that the complete term $\mathbf{H}\mathbf{V}$ disappears. In these cases, the calculation of a modified free response \mathbf{F}' is unnecessary; simply \mathbf{F} is used instead of \mathbf{F}' in equation (8.73). However, if the future disturbances are explicitly known, \mathbf{F}' should be determined exactly.

如果将多项式矩阵 $\mathbf{G}_p(z^{-1})$, $\mathbf{H}_p(z^{-1})$ 和 $\mathbf{F}(z^{-1})$ 的系数进一步整合, 可得:

$$\mathbf{F} = \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{G}_p(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_p(z^{-1})\Delta\mathbf{v}(t) = \mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{YUV} \quad (8.76)$$

其中, 除了已知的 $\mathbf{F}(z^{-1})$ 和 $\mathbf{G}(z^{-1})$ 的定义之外, 还满足方程 (8.77) 至 (8.79)。

如前所述, 为了进一步简化矩阵运算, 经整合后的表达式为 $\mathbf{F}' = \mathbf{H}\mathbf{V} + \mathbf{F}$ 。只有当扰动项 $\mathbf{v}(t)$ 未来时刻的值是已知的, 上述考虑才有意义。假使 $\mathbf{v}(t)$ 的未来时刻的值是未知的, 那么一种有效的处理方式是假设未来时刻的值与当前时刻的值相同, 也即 $\mathbf{v}(t+j) = \mathbf{v}(t)$ 。从而, $\mathbf{v}(t)$ 的变化量 $\Delta\mathbf{v}(t+j)$ 将全部为零, 那么整个的 $\mathbf{H}\mathbf{V}$ 项也不复存在。在上述情况下, 经修正的自由响应 \mathbf{F}' 的计算则是不必要的, 可直接使用 \mathbf{F} 来代替方程 (8.73) 中的 \mathbf{F}' 。假设未来时刻的扰动变量是已知的, 那么则能够准确地计算出 \mathbf{F}' 。

Die modifizierte freie Antwort ergibt sich dann zu

$$\mathbf{F}' = \mathbf{H} \cdot \mathbf{V} + \mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{Y}\mathbf{U}\mathbf{V} = \mathbf{H}\mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{V}\mathbf{Y}\mathbf{U}\mathbf{V} \quad (8.80)$$

wobei nun

$$\mathbf{H}\mathbf{F}\mathbf{G}_p\mathbf{H}_p = \begin{bmatrix} \mathbf{H}_0 & 0 & \cdots & 0 \\ \mathbf{H}_1 & \mathbf{H}_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{N_p-1} & \mathbf{H}_{N_p-2} & \cdots & \mathbf{H}_0 \\ \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\ \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,na} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\ \mathbf{G}_{1p,0} & \mathbf{G}_{1p,1} & \cdots & \mathbf{G}_{1p,nb-1} \\ \mathbf{G}_{2p,0} & \mathbf{G}_{2p,1} & \cdots & \mathbf{G}_{2p,nb-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{N_pp,0} & \mathbf{G}_{N_pp,1} & \cdots & \mathbf{G}_{N_pp,nb-1} \\ \mathbf{H}_{1p,0} & \mathbf{H}_{1p,1} & \cdots & \mathbf{H}_{1p,nd-1} \\ \mathbf{H}_{2p,0} & \mathbf{H}_{2p,1} & \cdots & \mathbf{H}_{2p,nd-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{N_pp,0} & \mathbf{H}_{N_pp,1} & \cdots & \mathbf{H}_{N_pp,nd-1} \end{bmatrix}$$

$$\mathbf{V}\mathbf{Y}\mathbf{U}\mathbf{V} = \begin{bmatrix} \Delta\mathbf{v}(t+1) \\ \vdots \\ \Delta\mathbf{v}(t+N_p) \\ \mathbf{y}(t) \\ \vdots \\ \mathbf{y}(t-na) \\ \Delta\mathbf{u}(t-1) \\ \vdots \\ \Delta\mathbf{u}(t-nb) \\ \Delta\mathbf{v}(t) \\ \vdots \\ \Delta\mathbf{v}(t-nd+1) \end{bmatrix}$$

Then, the modified free response results to

$$\mathbf{F}' = \mathbf{H} \cdot \mathbf{V} + \mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{Y}\mathbf{U}\mathbf{V} = \mathbf{H}\mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{V}\mathbf{Y}\mathbf{U}\mathbf{V} \quad (8.80)$$

in which

$$\mathbf{H}\mathbf{F}\mathbf{G}_p\mathbf{H}_p = \begin{bmatrix} \mathbf{H}_0 & 0 & \cdots & 0 \\ \mathbf{H}_1 & \mathbf{H}_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{N_p-1} & \mathbf{H}_{N_p-2} & \cdots & \mathbf{H}_0 \\ \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\ \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,na} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\ \mathbf{G}_{1p,0} & \mathbf{G}_{1p,1} & \cdots & \mathbf{G}_{1p, nb-1} \\ \mathbf{G}_{2p,0} & \mathbf{G}_{2p,1} & \cdots & \mathbf{G}_{2p, nb-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{N_p p,0} & \mathbf{G}_{N_p p,1} & \cdots & \mathbf{G}_{N_p p, nb-1} \\ \mathbf{H}_{1p,0} & \mathbf{H}_{1p,1} & \cdots & \mathbf{H}_{1p, nd-1} \\ \mathbf{H}_{2p,0} & \mathbf{H}_{2p,1} & \cdots & \mathbf{H}_{2p, nd-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{N_p p,0} & \mathbf{H}_{N_p p,1} & \cdots & \mathbf{H}_{N_p p, nd-1} \end{bmatrix}$$

$$\mathbf{V}\mathbf{Y}\mathbf{U}\mathbf{V} = \begin{bmatrix} \Delta\mathbf{v}(t+1) \\ \vdots \\ \Delta\mathbf{v}(t+N_p) \\ \mathbf{y}(t) \\ \vdots \\ \mathbf{y}(t-na) \\ \Delta\mathbf{u}(t-1) \\ \vdots \\ \Delta\mathbf{u}(t-nb) \\ \Delta\mathbf{v}(t) \\ \vdots \\ \Delta\mathbf{v}(t-nd+1) \end{bmatrix}$$

修正的自由相应表达式可见方程 (8.80)。

8.2.4 Berücksichtigung des Regelhorizonts/Consideration of the control horizon/考虑控制步长后的简化

Analog zur in Kap. 8.1.4 beschriebenen Vorgehensweise sollte man auch bei GPC mit Störgrößenberücksichtigung die zur Berechnung der optimalen, zukünftigen Stellgrößen notwendige Rechenzeit dadurch verkürzen, dass man den Regelhorizont N_u mit $N_u \leq N_2$ in die Überlegungen einbezieht. Die für die Berechnung von $\Delta \mathbf{u}(t+j)$, $j = N_1 \dots N_2$ nicht notwendigen Teile der Matrizen \mathbf{G} , \mathbf{H} , \mathbf{F}' etc. werden vernachlässigt, bzw. die Matrizen auf die unbedingt nötige Größe reduziert. Es ergibt sich eine in den Dimensionen geschrumpfte Prädiktionsgleichung

$$\begin{aligned}
 \begin{bmatrix} \hat{\mathbf{y}}(t+N_1) \\ \vdots \\ \hat{\mathbf{y}}(t+N_2) \end{bmatrix} &= \begin{bmatrix} \mathbf{G}_{N_1-1} & \mathbf{G}_{N_1-2} & \cdots & \mathbf{G}_{N_1-N_u} \\ \vdots & \vdots & & \vdots \\ \mathbf{G}_{N_2-1} & \mathbf{G}_{N_2-2} & \cdots & \mathbf{G}_{N_2-N_u} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}(t) \\ \vdots \\ \Delta \mathbf{u}(t+N_u-1) \end{bmatrix} \\
 &+ \begin{bmatrix} \mathbf{H}_{N_1-1} & \mathbf{H}_{N_1-2} & \cdots & \mathbf{H}_{N_1-N_u} \\ \vdots & \vdots & & \vdots \\ \mathbf{H}_{N_2-1} & \mathbf{H}_{N_2-2} & \cdots & \mathbf{H}_{N_2-N_u} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{v}(t+1) \\ \vdots \\ \Delta \mathbf{v}(t+N_u) \end{bmatrix} \\
 &+ \begin{bmatrix} \mathbf{f}(t+N_1) \\ \vdots \\ \mathbf{f}(t+N_2) \end{bmatrix}
 \end{aligned} \tag{8.81}$$

oder

$$\mathbf{Y}_{N_{12}} = \mathbf{G}_{N_{12u}} \mathbf{U}_{N_u} + \mathbf{H}_{N_{12u}} \mathbf{V} + \mathbf{F}_{N_{12}} \tag{8.82}$$

Genauso vereinfacht sich die Formel (8.73) zur Berechnung der optimalen Stellgrößen zu

$$\mathbf{U}_{N_u} = (\mathbf{G}_{N_{12u}}^T \mathbf{G}_{N_{12u}} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}_{N_{12u}}^T (\mathbf{W}_{N_{12}} - \mathbf{F}'_{N_{12}}) \tag{8.83}$$

Da in den meisten Fällen, besonders in dem hier beschriebenen Anwendungsfall einer Antriebsregelung, die zukünftigen Störeinflüsse nicht bekannt sein werden, erübrigt sich die Berechnung einer modifizierten freien Antwort $\mathbf{F}'_{N_{12}}$ und damit der Übertragungsmatrix $\mathbf{H}_{N_{12u}}$.

8.2.4 Consideration of the control horizon

Analog to the approach described in Chapter 8.1.4, the computation time necessary for the calculation of optimum future values for the actuating variables should also be reduced when GPC with consideration of disturbances is used by taking the control horizon N_u with $N_u \leq N_2$ into account. The parts of the matrices \mathbf{G} , \mathbf{H} , \mathbf{F}' etc. , which are unnecessary for the calculation of $\Delta \mathbf{u}(t+j)$, $j = N_1 \dots N_2$ are neglected and the matrices are reduced in their dimensions to sizes which are as small as possible. A prediction equation with lesser dimensions results, see equation (8.81) or (8.82).

In the same way, equation (8.73) for the calculation of optimum values for the actuating variables can be simplified to

$$\mathbf{U}_{N_u} = (\mathbf{G}_{N_{12u}}^T \mathbf{G}_{N_{12u}} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}_{N_{12u}}^T (\mathbf{W}_{N_{12}} - \mathbf{F}'_{N_{12}}) \quad (8.83)$$

Since in most cases, especially in the application for a drive control that is described here, future disturbances will not be known, the calculation of a modified free response $\mathbf{F}'_{N_{12}}$ and with it of a transfer function matrix $\mathbf{H}_{N_{12u}}$ is unnecessary.

8.2.4 考虑控制步长后的简化

类似于章节 8.1.4 所述的处理方法, 对于考虑了扰动因素后的 GPC, 它也可以通过将控制步长 N_u 保持在 $N_u \leq N_2$ 的方式来降低计算最优的未来控制变量所需的时间。 \mathbf{G} , \mathbf{H} , \mathbf{F}' 等相应矩阵中与 $\Delta \mathbf{u}(t+j)$, $j = N_1 \dots N_2$ 的计算无关的部分可以忽略。换言之, 在满足必须的计算条件下, 矩阵维度可进一步降低。从而, 可得到一个降低维数的预测方程, 见方程 (8.81) 或 (8.82)。

同样, 用于计算最优控制变量的方程 (8.73) 也可简化为:

$$\mathbf{U}_{N_u} = (\mathbf{G}_{N_{12u}}^T \mathbf{G}_{N_{12u}} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}_{N_{12u}}^T (\mathbf{W}_{N_{12}} - \mathbf{F}'_{N_{12}}) \quad (8.83)$$

由于在大多数情况下, 尤其是本书所描述的驱动控制领域的应用场合, 未来时刻的扰动是未知的, 所以修正的自由响应 $\mathbf{F}'_{N_{12}}$ 和传递矩阵 $\mathbf{H}_{N_{12u}}$ 的计算是不必要的。

Folglich wird zur Voraussage der freien Antwort lediglich die Formel

$$\mathbf{F}_{N_{12}} = \mathbf{F}\mathbf{G}_p\mathbf{H}_p\mathbf{N}_{N_{12}} \cdot \mathbf{YUV} \quad (8.84)$$

benötigt, wobei

$$\mathbf{F}\mathbf{G}_p\mathbf{H}_p\mathbf{N}_{N_{12}} = \begin{bmatrix} \mathbf{F}_{N_{1,0}} & \mathbf{F}_{N_{1,1}} & \cdots & \mathbf{F}_{N_{1,na}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_{2,0}} & \mathbf{F}_{N_{2,1}} & \cdots & \mathbf{F}_{N_{2,na}} \\ \mathbf{G}_{N_{1p,0}} & \mathbf{G}_{N_{1p,1}} & \cdots & \mathbf{G}_{N_{1p,nb-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{N_{2p,0}} & \mathbf{G}_{N_{2p,1}} & \cdots & \mathbf{G}_{N_{2p,nb-1}} \\ \mathbf{H}_{N_{1p,0}} & \mathbf{H}_{N_{1p,1}} & \cdots & \mathbf{H}_{N_{1p,nd-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{N_{2p,0}} & \mathbf{H}_{N_{2p,1}} & \cdots & \mathbf{H}_{N_{2p,nd-1}} \end{bmatrix} \quad (8.85)$$

8.3 MIMO-GPC mit Filter/MIMO-GPC with filter/带滤波器的 MIMO-GPC

Wie bereits bei modellbasierten Eingrößenreglern beschrieben (Kap. 6), so ist auch ein MIMO-GPC-Regler beim Auftreten überschwingungsbehafteter Messwerte kaum beherrschbar. Die möglichen Auswege, durch Zurücknehmen der Reglerdynamik den Regelkreis zu stabilisieren, oder ein Tiefpassfilter in die Messstrecke einzufügen, verbieten sich, da sie die Regelung insgesamt träger machen. Abhilfe schafft die bereits in Kap. 6.2 für den eindimensionalen Fall beschriebene Vorgehensweise, ein Filter in den GPC-Regler zu integrieren. Dies soll im Folgenden für eine mehrdimensionale Regelung adaptiert werden.

Selbstverständlich ist der Einsatz des integrierten Filters sowohl mit als auch ohne Berücksichtigung der Störgrößen im Streckenmodell möglich. Der Einfachheit halber erfolgt hier nur die Herleitung des Verfahrens unter Berücksichtigung von bekannten Störeinflüssen. Eine GPC-Regelung mit Filterung, aber ohne Störgrößenberücksichtigung kann der Leser leicht selbst herleiten; es sind in diesem Fall in der folgenden Beschreibung lediglich die $\mathbf{D}(z^{-1})$ - und $\mathbf{v}(t)$ -Terme sowie die davon abgeleiteten Größen wegzulassen.

Zur praktischen Implementierung der Filtereigenschaften beschreiben Camacho und Bordons [20, Kap. 6.1.2] ein bei Goodwin/Sin [44, Kap. 7.4.2] entnommenes Verfahren; es erscheint aber einfacher und sinnvoller, das in 6.2.1 auf Seite 99 ff. besprochene, an Clarke [28] angelehnte SISO-Prinzip auf Mehrgrößenregelung und Berücksichtigung der Störgrößen zu erweitern.

Therefore, for the prediction of the free response, only the equation (8.84) is required, in which equation (8.84).

MIMO-GPC with filter

As already described for unidimensional model-based controllers (Chapter 6), a MIMO-GPC controller is also not really feasible in the presence of harmonic affected measured values. The possible workarounds, i.e. stabilizing the control loop by reducing the controller dynamics or by integrating a low-pass filter into the measuring system, can absolutely not be recommended as they lead to an all in all more sluggish control. A remedial approach for this, as already described in Chapter 6.2 for the uni-dimensional case, is to integrate a filter into the GPC controller itself. In the following, this should be adapted for a multidimensional control.

Of course, the use of the integrated filter in the system model is possible with and also without consideration of the disturbances. For reasons of simplicity, here only the procedure with consideration of known disturbances is derived. The reader himself can easily derive a GPC control with filter, but without considering the disturbances; in this case only the terms $\mathbf{D}(z^{-1})$ and $\mathbf{v}(t)$ as well as the terms derived from these have to be neglected in the following description.

For a practical implementation of the filter characteristics, Camacho/Bordons [20, Chapter 6.1.2] describe a method that is taken from Goodwin/Sin [44, Chapter 7.4.2]; however, it seems to be more easy and more reasonable to extend the principle discussed in Chapter 6.2.1 on page 99 et seqq. to multidimensional control with consideration of disturbances similar to the SISO principle by Clarke [28].

因此, 仅需使用如下表达式计算自由响应, 见方程 (8.84), 其中, 有方程 (8.84)。

带滤波器的 MIMO-GPC

正如第 6 章所述的单维情形下的广义预测控制器, 一个 MIMO-GPC 控制器也同样无法应对所测变量包含高次谐波的情形。可能的解决方案包括: 一是通过降低控制器的动态特性来稳定控制闭环; 或者是在测量环节引入一个低通滤波器。但上述两种方案均会导致系统的整体响应变得迟缓, 所以是不可取的。建议使用章节 6.2 所描述的适用于单维情形时的处理方式, 也就是将滤波器嵌入至 GPC 控制器中。当然, 对于多维的控制情形来说, 需要进行相应的调整, 如下文所述。

很明显, 在被控系统模型中嵌入滤波器的方案同时适用于考虑或者忽略扰动变量时的情形。为简化起见, 可通过考虑一个已知的扰动变量来推导上述方法。基于对本部分内容的理解, 至于包含滤波的 GPC 控制, 且同时不考虑扰动因素的情形, 读者是可以很轻易地独立完成相应推导过程。需要说明的是, 此时 $\mathbf{D}(z^{-1})$ 和 $\mathbf{v}(t)$ 以及由其推导出的相关项均需全部消去。

学者 Camacho/Bordons 的著作 [20, 章节 6.1.2] 描述了一种包含滤波器特性的实际实现方式, 它借鉴了学者 Goodwin/Sin 的著作 [44, 章节 7.4.2] 所用到的方法。但是, 基于章节 6.2.1 第 99 页所述的依据学者 Clarke 的著作 [28] 所述的 SISO 原理, 并直接将其扩展至考虑扰动因素下的多变量控制则显得更为简单且有意义。

8.3.1 Bestimmung der Übertragungsfunktion/Determination of the transfer function/传递函数的确定

Da das integrierte Filter keinen Einfluss auf die Strecke selbst besitzt, ändert sich die Streckenübertragungsfunktion selbstverständlich nicht. Die in den Kapiteln 8.1.1 und 8.2.1 hergeleiteten Formeln gelten uneingeschränkt.

8.3.2 Berechnung der Systemmatrizen/Calculation of the system matrices/系统矩阵的计算

Die Systemmatrizen hängen als Zerlegung der Übertragungsfunktion ebenfalls nicht von der Filterung, sondern nur von der Strecke selbst ab. Daher können die in den Kapiteln 8.1.2 und 8.2.2 genannten Formeln weiter Anwendung finden.

8.3.3 Mathematische Herleitung/Mathematical derivation/数学推导

Das CARIMA-Modell

Da sich die Strecke selbst nicht verändert hat, hat sich auch das CARIMA-Modell (8.58) aus Kap. 8.2.3 nicht verändert; lediglich die Annahme, dass $\xi(t)$ weißes Rauschen darstellt, kann nun nicht mehr getroffen werden. Folglich darf, wie schon in Kap. 6.2.1 auf Seite 99 für den eindimensionalen Fall beschrieben, $\mathbf{C}(z^{-1})$ nicht gleich der Einheitsmatrix \mathbf{I} gesetzt werden, sondern wird durch ein Design-Polynom $\mathbf{T}(z^{-1})$ ersetzt. Damit ergibt sich das CARIMA-Modell zu

$$\mathbf{A}(z^{-1})\mathbf{y}(t) = \mathbf{B}(z^{-1})\mathbf{u}(t-1) + \mathbf{D}(z^{-1})\mathbf{v}(t) + \mathbf{T}(z^{-1})\frac{\xi(t)}{\Delta} \quad (8.86)$$

mit

$$\mathbf{T}(z^{-1}) = \mathbf{T}_0 + \mathbf{T}_1 z^{-1} + \mathbf{T}_2 z^{-2} + \dots + \mathbf{T}_{nt} z^{-nt} \quad (8.87)$$

wobei bei einem Filter normalerweise $\mathbf{T}_0 = 1$ angesetzt wird. Die Dimension von $\mathbf{T}(z^{-1})$ beträgt

$$\mathbf{T}(z^{-1}) : \text{Matrix}(n \times n) \quad (8.88)$$

8.3.1 Determination of the transfer function

Since the integrated filter has no influence on the system itself, the transfer function of the system does of course not change. The equations derived in the chapters 8.1.1 and 8.2.1 apply without any limitations.

8.3.2 Calculation of the system matrices

Being a decomposition of the transfer function, the system matrices do also not depend on the filtering. Hence, the equations specified in the chapters 8.1.2 and 8.2.2 can be applied further on.

8.3.3 Mathematical derivation

The CARIMA model

Since the plant itself has not changed, the CARIMA model (8.58) from Chapter 8.2.3 has not changed, either; only the assumption that $\xi(t)$ represents white noise cannot be accepted anymore. Therefore, as already described for the unidimensional case in Chapter 6.2.1 on page 99, $\mathbf{C}(z^{-1})$ must not be set equal to the identity matrix \mathbf{I} , but is substituted with a design polynomial $\mathbf{T}(z^{-1})$. Thus, the CARIMA model results into equation (8.86), with equation (8.87), in which for a filter, normally $\mathbf{T}_0 = \mathbf{I}$ is set. The dimension of $\mathbf{T}(z^{-1})$ is

$$\mathbf{T}(z^{-1}) : \text{Matrix}(n \times n) \quad (8.88)$$

8.3.1 传递函数的确定

由于所嵌入的滤波器对于被控对象本身并无影响，显然被控对象的传递函数并不发生变化。从而，章节 8.1.1 和 8.2.1 所推导的方程依然适用。

8.3.2 系统矩阵的计算

系统矩阵也不受滤波因素的影响，而仅取决于被控对象本身。从而，章节 8.1.2 和 8.2.2 所推导的方程依然适用。

8.3.3 数学推导

CARIMA 模型

由于被控对象本身并未发生变化，从而，章节 8.2.3 所述的 CARIMA 模型方程 (8.58) 也保持不变。但是， $\xi(t)$ 表示白噪声这一假设则不再满足。更进一步，正如已在章节 6.2.1 第 99 页所述的单维情形， $\mathbf{C}(z^{-1})$ 也不再可设置为单位矩阵 \mathbf{I} ，而是使用一个设计多项式 $\mathbf{T}(z^{-1})$ 进行替代。从而，可以得到 CARIMA 模型，见方程 (8.86)，且有方程 (8.87)，其中，在存在滤波的情况下，通常设置 $\mathbf{T}_0 = \mathbf{I}$ 。 $\mathbf{T}(z^{-1})$ 的维数为：

$$\mathbf{T}(z^{-1}) : \text{矩阵}(n \times n) \quad (8.88)$$

Der j-Schritt-voraus-Prädiktor

Durch den gegenüber der Gl. (8.60) nun vorhandenen zusätzlichen Term mit $\mathbf{T}(z^{-1})$, ändert sich auch die Diophantische Gleichung (8.14) zu

$$\begin{aligned}\mathbf{T}(z^{-1}) &= \mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})\Delta + z^{-j}\mathbf{F}_j(z^{-1}) \\ &= \mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1}) + z^{-j}\mathbf{F}_j(z^{-1})\end{aligned}\quad (8.89)$$

Analog zur in Kap. 8.1.3 auf Seite 171 ff. erläuterten Vorgehensweise erhält man aus der Multiplikation von (8.86) mit $\Delta\mathbf{E}_j(z^{-1})z^j$ und Einsetzen der nach $\mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})$ aufgelösten Formel (8.89) in das Ergebnis die Prädiktionsgleichung:

$$\begin{aligned}\mathbf{T}(z^{-1})\mathbf{y}(t+j) &= \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}(t+j-1) \\ &\quad + \mathbf{E}_j(z^{-1})\mathbf{D}(z^{-1})\Delta\mathbf{v}(t+j) \\ &\quad + \mathbf{F}_j(z^{-1})\mathbf{y}(t) \\ &\quad + \mathbf{T}(z^{-1})\mathbf{E}_j(z^{-1})\boldsymbol{\xi}(t+j)\end{aligned}\quad (8.90)$$

Wiederum liegen, da $\mathbf{E}_j(z^{-1})$ vom Grad $j-1$ ist, die Rauschkomponenten alle in der Zukunft und können daher nicht in die Prädiktion eingehen. Gleichzeitig kann man alle Größen durch $\mathbf{T}(z^{-1})$ filtern, d. h. durch $\mathbf{T}(z^{-1})$ dividieren. Es ergibt sich nun:

$$\begin{aligned}\hat{\mathbf{y}}(t+j) &= \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1})\Delta\mathbf{u}^f(t+j-1) \\ &\quad + \mathbf{E}_j(z^{-1})\mathbf{D}(z^{-1})\Delta\mathbf{v}^f(t+j) \\ &\quad + \mathbf{F}_j(z^{-1})\mathbf{y}^f(t)\end{aligned}\quad (8.91)$$

mit den gefilterten Größen

$$\mathbf{u}^f(t) = \frac{\mathbf{u}(t)}{\mathbf{T}(z^{-1})}, \quad \mathbf{v}^f(t) = \frac{\mathbf{v}(t)}{\mathbf{T}(z^{-1})}, \quad \mathbf{y}^f(t) = \frac{\mathbf{y}(t)}{\mathbf{T}(z^{-1})}$$

Die zukünftigen Stellgrößen sind noch nicht bekannt, daher macht es nicht viel Sinn, sie zu filtern; das Gleiche gilt für die zukünftigen Störeinflüsse. Aus den entsprechenden Termen ist das $\mathbf{T}(z^{-1})$ -Polynom folglich wieder herauszurechnen. Analog zu den Gln. (6.48), (8.17) und (8.63) ergeben sich die Definitionen:

$$\mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1}) = \mathbf{G}_j'(z^{-1})\mathbf{T}(z^{-1}) + z^{-j}\Gamma_j(z^{-1}) \quad (8.92)$$

$$\mathbf{E}_j(z^{-1})\mathbf{D}(z^{-1}) = \mathbf{H}_j'(z^{-1})\mathbf{T}(z^{-1}) + z^{-j}\Theta_j(z^{-1}) \quad (8.93)$$

The j-step ahead predictor

Compared to equation (8.60), an additional term with $\mathbf{T}(z^{-1})$ exists now and so the Diophantine equation (8.14) changes to equation (8.89).

Analog to the approach described in Chapter 8.1.3 on page 171 et seqq. by multiplying (8.86) with $\Delta \mathbf{E}_j(z^{-1})z^j$ and by substituting equation (8.89) solved for $\mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})$ into the result, the following prediction equation can be obtained, see equation (8.90).

Again, as $\mathbf{E}_j(z^{-1})$ is of degree $j - 1$, all noise terms are completely in the future, which means that they do not have to be considered for the prediction. At the same time, all variables can be filtered with $\mathbf{T}(z^{-1})$, i. e. dividing them by $\mathbf{T}(z^{-1})$. Then, the following results, see equation (8.91), with the filtered variables

$$\mathbf{u}^f(t) = \frac{\mathbf{u}(t)}{\mathbf{T}(z^{-1})}, \quad \mathbf{v}^f(t) = \frac{\mathbf{v}(t)}{\mathbf{T}(z^{-1})}, \quad \mathbf{y}^f(t) = \frac{\mathbf{y}(t)}{\mathbf{T}(z^{-1})}$$

Future values of the actuating variables are not known yet. Hence, it does not make sense to filter them; the same applies for future disturbances. Therefore, the polynomial $\mathbf{T}(z^{-1})$ has to be taken out of the corresponding terms. Analog to the equations (6.48), (8.17) and (8.63), the following definitions result, see equation (8.92) and equation (8.93).

J 步前向预测器

相较于方程 (8.60), 现在额外多了一个含有 $\mathbf{T}(z^{-1})$ 的项, 从而, 丢番图方程 (8.14) 变化为方程 (8.89)。

类似于章节 8.1.3 第 171 页所述的处理方式, 将方程 (8.86) 两侧同乘以 $\Delta \mathbf{E}_j(z^{-1})z^j$, 并将其代入至由方程 (8.89) 向 $\mathbf{E}_j(z^{-1})\mathbf{A}(z^{-1})$ 求解后得到的表达式, 即可得到预测方程的表达式, 见方程 (8.90)。

同样, 由于 $\mathbf{E}_j(z^{-1})$ 的阶数为 $j - 1$, 那么所有的噪声分量则位于未来时刻, 这意味着预测时无需考虑上述分量。同时, 所有的变量均可借助于 $\mathbf{T}(z^{-1})$ 进行滤波, 也即除以 $\mathbf{T}(z^{-1})$ 。如此即可得到方程 (8.91), 其中滤波后的变量为:

$$\mathbf{u}^f(t) = \frac{\mathbf{u}(t)}{\mathbf{T}(z^{-1})}, \quad \mathbf{v}^f(t) = \frac{\mathbf{v}(t)}{\mathbf{T}(z^{-1})}, \quad \mathbf{y}^f(t) = \frac{\mathbf{y}(t)}{\mathbf{T}(z^{-1})}$$

由于未来时刻的控制变量是未知的, 从而并无必要对其进行滤波处理; 同理, 未来时刻的扰动变量也是如此。因此, 需要在相对应的项中消去多项式 $\mathbf{T}(z^{-1})$ 。类似于前述方程 (6.48), (8.17) 和 (8.63), 此时定义如下, 见方程 (8.92) 和方程 (8.93)。

Damit wird die Prädiktionsgleichung zu

$$\begin{aligned} \widehat{\mathbf{y}}(t+j) &= \underbrace{\mathbf{G}_j'(z^{-1})\Delta\mathbf{u}(t+j-1)}_{\text{erzwungene Antwort}} + \underbrace{\mathbf{H}_j'(z^{-1})\Delta\mathbf{v}(t+j)}_{\substack{\text{zukünftige, determ.} \\ \text{Störeinflüsse}}} \\ &\quad + \underbrace{\Gamma_j(z^{-1})\Delta\mathbf{u}^f(t-1) + \Theta_j(z^{-1})\Delta\mathbf{v}^f(t) + \mathbf{F}_j(z^{-1})\mathbf{y}^f(t)}_{\substack{\text{alles Werte in der Vergangenheit} \\ \rightarrow \text{freie Antwort } \mathbf{f}^f(t+j)}} \end{aligned} \quad (8.94)$$

$$\widehat{\mathbf{y}}(t+j) = \mathbf{G}_j'(z^{-1})\Delta\mathbf{u}(t+j-1) + \mathbf{H}_j'(z^{-1})\Delta\mathbf{v}(t+j) + \mathbf{f}^f(t+j) \quad (8.95)$$

mit

$$\mathbf{f}^f(t+j) = \Gamma_j(z^{-1})\Delta\mathbf{u}^f(t-1) + \Theta_j(z^{-1})\Delta\mathbf{v}^f(t) + \mathbf{F}_j(z^{-1})\mathbf{y}^f(t) \quad (8.96)$$

Dimensionen:

$$\begin{aligned} \mathbf{G}_j'(z^{-1}), \Gamma_j(z^{-1}) &: \text{Matrix}(n \times m) \\ \mathbf{H}_j'(z^{-1}), \Theta_j(z^{-1}) &: \text{Matrix}(n \times l) \\ \mathbf{f}^f(t) &: \text{Vektor}(n \times 1) \end{aligned}$$

Wie die Gln. (8.86) bis (8.94) zeigen, müssen die Störgrößen $\mathbf{v}(t)$ mit dem gleichen Filter gefiltert werden, wie die Stellgrößen $\mathbf{u}(t)$ und die Regelgrößen $\mathbf{y}(t)$. Die drei oben genannten Vektoren müssen daher die gleiche Dimension haben, d. h. $n = m = l$. Ist dies nicht der Fall, so kann man sich behelfen, indem man die entsprechenden Systemmatrizen des Maschinenmodells aus Kap. 7.3 künstlich durch Hinzufügen von Nullelementen um die notwendige Anzahl Zeilen vergrößert. Auf diese Art und Weise kann sichergestellt werden, dass alle Systemgrößen mit der gleichen Filtermatrix gefiltert werden können.

Abschließend können die einzelnen Prädiktionsschritte mit

$$\widehat{\mathbf{y}}(t+1) = \mathbf{G}_1'(z^{-1})\Delta\mathbf{u}(t) + \mathbf{H}_1'(z^{-1})\Delta\mathbf{v}(t+1) + \mathbf{f}^f(t+1) \quad (8.97)$$

$$\widehat{\mathbf{y}}(t+2) = \mathbf{G}_2'(z^{-1})\Delta\mathbf{u}(t+1) + \mathbf{H}_2'(z^{-1})\Delta\mathbf{v}(t+2) + \mathbf{f}^f(t+2) \quad (8.98)$$

$$\widehat{\mathbf{y}}(t+3) = \mathbf{G}_3'(z^{-1})\Delta\mathbf{u}(t+2) + \mathbf{H}_3'(z^{-1})\Delta\mathbf{v}(t+3) + \mathbf{f}^f(t+3) \quad (8.99)$$

⋮

$$\widehat{\mathbf{y}}(t+N_p) = \mathbf{G}_{N_p}'(z^{-1})\Delta\mathbf{u}(t+N_p-1) + \mathbf{H}_{N_p}'(z^{-1})\Delta\mathbf{v}(t+N_p) + \mathbf{f}^f(t+N_p) \quad (8.100)$$

angegeben werden.

With these, the prediction equation becomes

$$\begin{aligned} \hat{\mathbf{y}}(t+j) = & \underbrace{\mathbf{G}_j'(z^{-1})\Delta\mathbf{u}(t+j-1)}_{\text{forced response}} + \underbrace{\mathbf{H}_j'(z^{-1})\Delta\mathbf{v}(t+j)}_{\text{future deterministic disturbances}} \\ & + \underbrace{\Gamma_j(z^{-1})\Delta\mathbf{u}^f(t-1) + \Theta_j(z^{-1})\Delta\mathbf{v}^f(t) + \mathbf{F}_j(z^{-1})\mathbf{y}^f(t)}_{\substack{\text{all terms in the past} \\ \rightarrow \text{free response } \mathbf{f}^f(t+j)}} \end{aligned} \quad (8.94)$$

further, see [equation \(8.95\)](#) with [equation \(8.96\)](#). Dimensions:

$$\begin{aligned} \mathbf{G}_j'(z^{-1}), \Gamma_j(z^{-1}) &: \text{Matrix}(n \times m) \\ \mathbf{H}_j'(z^{-1}), \Theta_j(z^{-1}) &: \text{Matrix}(n \times l) \\ \mathbf{f}^f(t) &: \text{Vector}(n \times 1) \end{aligned}$$

As the equations [\(8.86\)](#) to [\(8.94\)](#) show, the disturbance variables $\mathbf{v}(t)$ have to be filtered with the same filter as the actuating variables $\mathbf{u}(t)$ and the output variables $\mathbf{y}(t)$. Hence, the three vectors mentioned above must have the same dimensions, i.e. $n = m = l$. If this is not the case, the corresponding system matrices of the machine model from [Chapter 7.3](#) can be extended as a workaround by artificially adding zero elements in order to get the required number of rows. In this way it can be made sure that all system variables can be filtered with the identical filter matrix.

Finally, the single prediction steps can be written as [equation \(8.97\)](#) to [\(8.100\)](#).

由此可得预测方程的表达式,

$$\begin{aligned} \hat{\mathbf{y}}(t+j) = & \underbrace{\mathbf{G}_j'(z^{-1})\Delta\mathbf{u}(t+j-1)}_{\text{强制响应}} + \underbrace{\mathbf{H}_j'(z^{-1})\Delta\mathbf{v}(t+j)}_{\text{未来的、确定性的扰动}} \\ & + \underbrace{\Gamma_j(z^{-1})\Delta\mathbf{u}^f(t-1) + \Theta_j(z^{-1})\Delta\mathbf{v}^f(t) + \mathbf{F}_j(z^{-1})\mathbf{y}^f(t)}_{\substack{\text{所有过去时刻的数值} \\ \rightarrow \text{自由响应 } \mathbf{f}^f(t+j)}} \end{aligned} \quad (8.94)$$

更进一步, 见方程 [\(8.95\)](#), 其中, 有方程 [\(8.96\)](#)。维数为:

$$\begin{aligned} \mathbf{G}_j'(z^{-1}), \Gamma_j(z^{-1}) &: \text{矩阵}(n \times m) \\ \mathbf{H}_j'(z^{-1}), \Theta_j(z^{-1}) &: \text{矩阵}(n \times l) \\ \mathbf{f}^f(t) &: \text{向量}(n \times 1) \end{aligned}$$

正如方程 [\(8.86\)](#) 至 [\(8.94\)](#) 所示, 必须使用与控制变量 $\mathbf{u}(t)$ 和被控变量 $\mathbf{y}(t)$ 滤波处理时所使用的相同滤波器对扰动变量 $\mathbf{v}(t)$ 进行滤波处理。为此, 上述三个向量必须具有相同的维数, 也即 $n = m = l$ 。如果它们维数不同, 那么可以通过类似于[章节 7.3](#)所述的对电机模型中的相应系统矩阵人工添加零元素的方式将其扩展至所需要的行数。只有经过上述方式处理后, 才能保证所有的系统变量可以使用相同的滤波矩阵进行滤波处理。

最后, 即可分别列出每一步的预测表达式, 见方程 [\(8.97\)](#) 至 [\(8.100\)](#)。

Berechnung der Steuergrößen

Zur Ermittlung der optimalen, zukünftigen Stellgrößen geht man genauso vor, wie in Kap. 8.1.3 und 8.2.3 auf den Seiten 175 bzw. 195 für MIMO-GPC ohne internen Filter und ohne bzw. mit Störgrößenberücksichtigung beschrieben. Es ergibt sich aus Formel (8.95):

$$\mathbf{Y} = \mathbf{G}'\mathbf{U} + \mathbf{H}'\mathbf{V} + \mathbf{F}' \quad (8.101)$$

Zusammenfassen der von den Stellgrößen unabhängigen Teilterme $\mathbf{H}'\mathbf{V}$ und \mathbf{F}' ergibt

$$\mathbf{Y} = \mathbf{G}'\mathbf{U} + \mathbf{F}'' \quad \text{mit} \quad \mathbf{F}'' = \mathbf{H}'\mathbf{V} + \mathbf{F}' \quad (8.102)$$

Unter Anwendung der bereits in Kap. 8.1.3 auf Seite 175 genannten Kostenfunktion (8.24) erhält man schließlich die Berechnungsformel für optimale, zukünftige Stellgrößen

$$\mathbf{U} = (\mathbf{G}'^T \mathbf{G}' + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}'^T (\mathbf{W} - \mathbf{F}'') \quad (8.103)$$

Rekursion der Diophantischen Gleichung

Um die Diophantische Gleichung (8.89) für den mehrdimensionalen Fall zu lösen, geht man genau wie beim eindimensionalen Regler in Kap. 6.2.1 auf Seite 103 beschrieben vor. Auch hier entfällt durch die Subtraktion das Filter- oder Design-Polynom $\mathbf{T}(z^{-1})$. Daher gelten weiterhin die in Kap. 8.1.3 auf Seite 177 ermittelten Rekursionsformeln (8.31) bis (8.33). Wie in Kap. 8.2.3 auf Seite 195 erläutert, ändert die Berücksichtigung der Störgrößen die Diophantische Gleichung und ihre Lösung nicht.

Die Endbedingungen der Rekursion erhält man, indem man Gl. (8.89) für $j = 1$ betrachtet:

$$\mathbf{E}_1(z^{-1}) = \mathbf{E}_0 = \frac{\mathbf{T}_0}{\mathbf{A}_0} = \mathbf{I} \quad \text{da} \quad \mathbf{A}_0 = \mathbf{I} \quad \text{und} \quad \mathbf{T}_0 = \mathbf{I} \quad (8.104)$$

$$\mathbf{F}_1(z^{-1}) = z(\mathbf{T}(z^{-1}) - \mathbf{A}(z^{-1})) \quad (8.105)$$

Berechnung der freien und der erzwungenen Antwort

Die einzelnen Elemente zur Berechnung der freien und der erzwungenen Antwort werden auf die gleiche Weise berechnet, wie im eindimensionalen Fall. Ausgehend von Gl. (8.92) erhält man mit den in Kap. 6.2.1 auf Seite 103 ff. beschriebenen Überlegungen wiederum die Erkenntnis, dass auch hier in $\mathbf{G}'_j(z^{-1})$ keine Terme mit $\mathbf{u}(t+j)$, $j < 0$ eingehen dürfen. Damit ist der Grad von $\mathbf{G}'_j(z^{-1})$ auf $\leq (j - 1)$ begrenzt.

Calculation of the actuating variables For the determination of optimum future values for the actuating variables, the procedure is just the same as described in Chapter 8.1.3 and 8.2.3 on the pages 175 and 195 for MIMO-GPC without internal filter and without or with consideration of disturbances. From equation (8.95) results equation (8.101). Combining the partial terms $\mathbf{H}'\mathbf{V}$ and \mathbf{F}' , which are independent of the actuating variables, results in equation (8.102). Applying the cost function (8.24), which has already been described in Chapter 8.1.3 on page 175, finally, the equation for optimum future values of the actuating variables is obtained, see equation (8.103).

Recursion of the Diophantine equation In order to solve the Diophantine equation (8.89) for the multidimensional case, the procedure is exactly the same as the one already described for the uni-dimensional case in Chapter 6.2.1 on page 103. Again, the filter or design polynomial $\mathbf{T}(z^{-1})$ is omitted because of the subtraction. Hence, the recursion equations (8.31) to (8.33) determined in Chapter 8.1.3 on page 177 do also apply here. As explained in Chapter 8.2.3 on page 195, the consideration of disturbances does not change the Diophantine equation and its solution. The final terms of the recursion are obtained by considering equation (8.89) for $j = 1$, see equation (8.104) and (8.105).

Calculation of the free and forced response The individual elements for the calculation of the free and forced response are calculated in the same way as for the unidimensional case. Starting from equation (8.92) with the considerations described in Chapter 6.2.1 on page 103 et seqq., one can see that here again no terms with $\mathbf{u}(t+j), j < 0$ may enter $\mathbf{G}_j'(z^{-1})$. Because of this, the degree of $\mathbf{G}_j'(z^{-1})$ is limited to $\leq (j-1)$.

控制变量的计算 为了求得最优的控制变量, 可以使用类似于章节 8.1.3 和 8.2.3 所述的分别忽略和考虑干扰变量的不含滤波器的 MIMO-GPC 的处理方式。由方程 (8.95) 可得方程 (8.101)。将与控制变量无关的子项 $\mathbf{H}'\mathbf{V}$ 和 \mathbf{F}' 进行整合后可得方程 (8.102)。基于章节 8.1.3 第 175 页给出的成本函数 (8.24), 即可求得最优的、未来的控制变量表达式, 见方程 (8.103)。

丢番图方程的递归 为了求解多维情况下的丢番图方程 (8.89), 可以采用与章节 6.2.1 第 103 页所述的单维情形下的相同的求解方式。此时, 通过求差运算也可消去滤波或设计多项式 $\mathbf{T}(z^{-1})$ 。从而, 章节 8.1.3 第 177 页所求得的递归方程 (8.31) 至 (8.33) 依然适用。正如章节 8.2.3 第 195 页所述, 考虑扰动后的丢番图方程及其求解并不发生变化。

通过考虑 $j = 1$ 时的方程 (8.89), 即可得到递归的终止条件, 见方程 (8.104) 和 (8.105)。

自由响应和强制响应的计算 自由响应与强制响应的计算所需的单个元素的求解方式与单维情形下的求解方式类似。根据方程 (8.92), 并结合章节 6.2.1 第 103 页所描述的相关内容, 同样可知, $\mathbf{G}_j'(z^{-1})$ 中并不出现含有 $\mathbf{u}(t+j), j < 0$ 的项, 从而, $\mathbf{G}_j'(z^{-1})$ 的阶数被限制在 $\leq (j-1)$ 的范围。

Entsprechend folgt für die Elemente von $\mathbf{G}'_j(z^{-1})$:

1. Die Matrixelemente $\mathbf{G}'_{j,0} \dots \mathbf{G}'_{j,j-1}$ und $\mathbf{G}'_{j+1,0} \dots \mathbf{G}'_{j+1,j-1}$ sind alle identisch.
2. Die Matrixzeilen unterscheiden sich nur durch das jeweils neu hinzugefügte Element $\mathbf{G}'_{j+1,j}$.

Für \mathbf{G}' gilt daher die Definition:

$$\mathbf{G}' = \begin{bmatrix} \mathbf{G}'_0 & 0 & 0 & \dots & 0 \\ \mathbf{G}'_1 & \mathbf{G}'_0 & 0 & \dots & 0 \\ \mathbf{G}'_2 & \mathbf{G}'_1 & \mathbf{G}'_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}'_{N_p-1} & \mathbf{G}'_{N_p-2} & \mathbf{G}'_{N_p-3} & \dots & \mathbf{G}'_0 \end{bmatrix} \quad (8.106)$$

Die Bestimmung der Elemente von $\Gamma(z^{-1})$ erfolgt nun nach der auf Seite 103ff. erläuterten Methode: Die Gl. (8.92) wird für $j+1$ und j aufgestellt und voneinander subtrahiert:

$$\begin{array}{l} + \left| \mathbf{E}_{j+1}(z^{-1})\mathbf{B}(z^{-1}) = \mathbf{G}_{j+1}'(z^{-1})\mathbf{T}(z^{-1}) + z^{-(j+1)}\Gamma_{j+1}(z^{-1}) \right. \\ - \left| \mathbf{E}_j(z^{-1})\mathbf{B}(z^{-1}) = \mathbf{G}_j'(z^{-1})\mathbf{T}(z^{-1}) + z^{-j}\Gamma_j(z^{-1}) \right. \\ \hline \left. \mathbf{E}_j z^{-j} \mathbf{B}(z^{-1}) = \mathbf{G}'_j z^{-j} \mathbf{T}(z^{-1}) + z^{-j} (z^{-1} \Gamma_{j+1}(z^{-1}) - \Gamma_j(z^{-1})) \right. \end{array} \quad (8.107)$$

Ein Koeffizientenvergleich ergibt:

$$\mathbf{G}'_j = \frac{\mathbf{E}_j \mathbf{B}_0 + \Gamma_{j,0}}{\mathbf{T}_0}, \quad \Gamma_{j+1,i-1} = \Gamma_{j,i} + \mathbf{E}_j \mathbf{B}_i - \mathbf{G}'_j \mathbf{T}_i \quad (8.108)$$

und für das letzte Element:

$$\Gamma_{j+1,i} = \mathbf{E}_j \mathbf{B}_{i+1} - \mathbf{G}'_j \mathbf{T}_{i+1} \quad (8.109)$$

Die Endbedingungen der Rekursion können aus Gl. (8.92) für $j=1$ wie folgt bestimmt werden:

$$\mathbf{G}'_0 = \frac{\mathbf{E}_0 \mathbf{B}_0}{\mathbf{T}_0}, \quad \Gamma_{1,i} = \mathbf{E}_0 \mathbf{B}_{i+1} - \mathbf{G}'_0 \mathbf{T}_{i+1} \quad (8.110)$$

Die Grenzen bzw. Grade der einzelnen Polynommatrizen können nach den gleichen Kriterien ermittelt werden, wie bei der SISO-Regelung mit Filter. Genauso entfallen wieder alle Terme mit $\mathbf{B}_k, k > nb$ und $\mathbf{T}_k, k > nt$.

Accordingly, for the elements of $\mathbf{G}'_j(z^{-1})$ follows:

1. The matrix elements $\mathbf{G}'_{j,0} \dots \mathbf{G}'_{j,j-1}$ and $\mathbf{G}'_{j+1,0} \dots \mathbf{G}'_{j+1,j-1}$ are all identical.
2. The matrix rows only differ in the newly added element $\mathbf{G}'_{j+1,j}$.

Hence, the following definition applies for \mathbf{G}' , see [equation \(8.106\)](#).

Now, the determination of the elements of $\Gamma(z^{-1})$ is done with the approach described on page 103 et seqq.: Equation (8.92) is set up for $j+1$ and j and then these equations are subtracted from each other, see [equation \(8.107\)](#).

A coefficient comparison results into [equation \(8.108\)](#) and for the last element, see [equation \(8.109\)](#).

The final terms of the recursion can be determined from [equation \(8.92\)](#) for $j=1$ as follows, see [equation \(8.110\)](#).

The limits or degrees of the individual polynomial matrices can be derived according to the same criteria as for SISO control with filter. In the same way, all terms with $\mathbf{B}_k, k > nb$ and $\mathbf{T}_k, k > nt$ are omitted again.

从而, $\mathbf{G}'_j(z^{-1})$ 中的元素满足:

1. 矩阵元素 $\mathbf{G}'_{j,0} \dots \mathbf{G}'_{j,j-1}$ 和 $\mathbf{G}'_{j+1,0} \dots \mathbf{G}'_{j+1,j-1}$ 是全部相同的;
2. 矩阵行与行之间差别仅在于新增加的元素 $\mathbf{G}'_{j+1,j}$ 。

对于 \mathbf{G}' 来说, 其定义可见 [方程 \(8.106\)](#)。

至于 $\Gamma(z^{-1})$ 中元素的确定, 可采用第 103 页所描述的方法: 首先分别列出 $j+1$ 和 j 时刻的 [方程 \(8.92\)](#), 然后对二者进行求差运算, 具体过程可见 [方程 \(8.107\)](#)。

经过系数对比, 可得 [方程 \(8.108\)](#), 最后一个元素满足 [方程 \(8.109\)](#)。

通过考虑 $j=1$ 时的 [方程 \(8.92\)](#), 即可确定递归的终止条件, 见 [方程 \(8.110\)](#)。

单个矩阵多项式的边界或阶数可通过使用与含有滤波的 SISO 控制时相同的标准获得。同样, 所有含有 $\mathbf{B}_k, k > nb$ 和 $\mathbf{T}_k, k > nt$ 的项都是需要忽略的。

Neben Γ ist für die Berechnung der freien Antwort noch Θ notwendig. Da die Gln. (8.92) und (8.93) vom Aufbau her völlig identisch sind, kann zur Berechnung der Elemente von \mathbf{H}' und Θ der gleiche Lösungsweg eingeschlagen werden, wie schon zur Berechnung von \mathbf{G}' und Γ . Es ergeben sich dann:

$$\mathbf{H}' = \begin{bmatrix} \mathbf{H}'_0 & 0 & 0 & \cdots & 0 \\ \mathbf{H}'_1 & \mathbf{H}'_0 & 0 & \cdots & 0 \\ \mathbf{H}'_2 & \mathbf{H}'_1 & \mathbf{H}'_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}'_{N_p-1} & \mathbf{H}'_{N_p-2} & \mathbf{H}'_{N_p-3} & \cdots & \mathbf{H}'_0 \end{bmatrix} \quad (8.111)$$

Sowie die Rekursionsformeln:

$$\mathbf{H}'_j = \frac{\mathbf{E}_j \mathbf{D}_0 + \Theta_{j,0}}{\mathbf{T}_0}, \quad \Theta_{j+1,i-1} = \Theta_{j,i} + \mathbf{E}_j \mathbf{D}_i - \mathbf{H}'_j \mathbf{T}_i \quad (8.112)$$

und für das letzte Element:

$$\Theta_{j+1,i} = \mathbf{E}_j \mathbf{D}_{i+1} - \mathbf{H}'_j \mathbf{T}_{i+1} \quad (8.113)$$

sowie für die Endbedingungen:

$$\mathbf{H}'_0 = \frac{\mathbf{E}_0 \mathbf{D}_0}{\mathbf{T}_0}, \quad \Theta_{1,i} = \mathbf{E}_0 \mathbf{D}_{i+1} - \mathbf{H}'_0 \mathbf{T}_{i+1} \quad (8.114)$$

Elemente mit $\mathbf{D}_k, k > nd$ und $\mathbf{T}_k, k > nt$ entfallen selbstverständlich. Damit kann die Prädiktionsgleichung (8.101) ausführlich geschrieben werden:

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{y}}(t+1) \\ \vdots \\ \hat{\mathbf{y}}(t+j) \\ \vdots \\ \hat{\mathbf{y}}(t+N_p) \end{bmatrix} &= \begin{bmatrix} \mathbf{G}'_0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}'_{j-1} & \cdots & \mathbf{G}'_0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{G}'_{N_p-1} & \cdots & \mathbf{G}'_{N_p-j} & \cdots & \mathbf{G}'_0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}(t) \\ \vdots \\ \Delta \mathbf{u}(t+j-1) \\ \vdots \\ \Delta \mathbf{u}(t+N_p-1) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{H}'_0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{H}'_{j-1} & \cdots & \mathbf{H}'_0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{H}'_{N_p-1} & \cdots & \mathbf{H}'_{N_p-j} & \cdots & \mathbf{H}'_0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{v}(t+1) \\ \vdots \\ \Delta \mathbf{v}(t+j) \\ \vdots \\ \Delta \mathbf{v}(t+N_p) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{f}'(t+1) \\ \vdots \\ \mathbf{f}'(t+j) \\ \vdots \\ \mathbf{f}'(t+N_p) \end{bmatrix} \end{aligned} \quad (8.115)$$

Besides Γ , Θ is also necessary for the calculation of the free response. Since the equations (8.92) and (8.93) have the absolutely identical framework, the same solution approach as for the calculation of \mathbf{G}' and Γ can be used for the calculation of the elements of \mathbf{H}' and Θ . Then, it follows equation (8.111).

The recursion equations are given by equation (8.112) and for the last element, see equation (8.113), as well as for the last terms, see equation (8.114).

Elements with $\mathbf{D}_k, k > nd$ and $\mathbf{T}_k, k > nt$ are, of course, omitted. Hence, the prediction equation (8.101) can be written in detail as equation (8.115).

除了 Γ 之外，还需要 Θ 才能够完成自由响应的计算。由于方程 (8.92) 和 (8.93) 从结构上来看是完全一致的，从而计算 \mathbf{H}' 和 Θ 可使用与计算 \mathbf{G}' 和 Γ 所使用的相同方法。从而有方程 (8.111)。

以及递归方程，见方程 (8.112)，其中最后一个元素满足方程 (8.113)，终止条件可见方程 (8.114)。

显然，含有 $\mathbf{D}_k, k > nd$ 和 $\mathbf{T}_k, k > nt$ 的元素是需要忽略的。从而，预测方程 (8.101) 可详细表示如下，见方程 (8.115)。

Zur einfacheren Berechnung der freien Antwort \mathbf{F}' werden nun nach bereits bekanntem Schema die gefilterten, vergangenen Werte $\mathbf{y}^f(t)$, $\Delta\mathbf{u}^f(t)$ und $\Delta\mathbf{v}^f(t)$ zu Matrizen \mathbf{Y}^f , \mathbf{U}^f und \mathbf{V}^f , dann zu einer großen Matrix \mathbf{YUV}^f zusammengefasst. Ebenso werden die Polynommatrizen $\mathbf{F}(z^{-1})$, $\Gamma(z^{-1})$ und $\Theta(z^{-1})$ zu $\mathbf{F}\Gamma\Theta$ verschmolzen. Es gilt dann:

$$\mathbf{F}' = \mathbf{F}(z^{-1})\mathbf{y}^f(t) + \Gamma(z^{-1})\Delta\mathbf{u}^f(t-1) + \Theta(z^{-1})\Delta\mathbf{v}^f(t) = \mathbf{F}\Gamma\Theta \cdot \mathbf{YUV}^f \quad (8.116)$$

wobei

$$\mathbf{F}(z^{-1}) = \begin{bmatrix} \mathbf{F}_1(z^{-1}) \\ \vdots \\ \mathbf{F}_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{1,0} + \mathbf{F}_{1,1}z^{-1} + \cdots + \mathbf{F}_{1,na}z^{-na} \\ \vdots \\ \mathbf{F}_{N_p,0} + \mathbf{F}_{N_p,1}z^{-1} + \cdots + \mathbf{F}_{N_p,na}z^{-na} \end{bmatrix} \quad (8.117)$$

$$\Gamma(z^{-1}) = \begin{bmatrix} \Gamma_1(z^{-1}) \\ \vdots \\ \Gamma_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \Gamma_{1,0} + \Gamma_{1,1}z^{-1} + \cdots + \Gamma_{1,nb-1}z^{-(nb-1)} \\ \vdots \\ \Gamma_{N_p,0} + \Gamma_{N_p,1}z^{-1} + \cdots + \Gamma_{N_p,nb-1}z^{-(nb-1)} \end{bmatrix} \quad (8.118)$$

$$\Theta(z^{-1}) = \begin{bmatrix} \Theta_1(z^{-1}) \\ \vdots \\ \Theta_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \Theta_{1,0} + \Theta_{1,1}z^{-1} + \cdots + \Theta_{1,nd-1}z^{-(nd-1)} \\ \vdots \\ \Theta_{N_p,0} + \Theta_{N_p,1}z^{-1} + \cdots + \Theta_{N_p,nd-1}z^{-(nd-1)} \end{bmatrix} \quad (8.119)$$

$$\mathbf{F}\Gamma\Theta = \begin{bmatrix} \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\ \Gamma_{1,0} & \Gamma_{1,1} & \cdots & \Gamma_{1,nb-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma_{N_p,0} & \Gamma_{N_p,1} & \cdots & \Gamma_{N_p,nb-1} \\ \Theta_{1,0} & \Theta_{1,1} & \cdots & \Theta_{1,nd-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Theta_{N_p,0} & \Theta_{N_p,1} & \cdots & \Theta_{N_p,nd-1} \end{bmatrix} \quad (8.120)$$

$$\mathbf{YUV}^f = \begin{bmatrix} \mathbf{y}^f(t) \\ \vdots \\ \mathbf{y}^f(t-na) \\ \Delta\mathbf{u}^f(t-1) \\ \vdots \\ \Delta\mathbf{u}^f(t-nb) \\ \Delta\mathbf{v}^f(t) \\ \vdots \\ \Delta\mathbf{v}^f(t-nd+1) \end{bmatrix} \quad (8.121)$$

For a simpler calculation of the free response \mathbf{F}' , the filtered past values $\mathbf{y}^f(t)$, $\Delta\mathbf{u}^f(t)$ and $\Delta\mathbf{v}^f(t)$ are combined to the matrices \mathbf{Y}^f , \mathbf{U}^f and \mathbf{V}^f and then they are altogether combined to a big matrix \mathbf{YUV}^f according to the already well-known procedure. In the same way, the polynomial matrices $\mathbf{F}(z^{-1})$, $\Gamma(z^{-1})$ and $\Theta(z^{-1})$ are combined to $\mathbf{F}\Gamma\Theta$. Then, it follows equation (8.116), in which equations (8.117) to (8.121).

为了简化自由响应 \mathbf{F}' 的计算, 基于已知的处理方式, 将滤波后的过去时刻的 $\mathbf{y}^f(t)$, $\Delta\mathbf{u}^f(t)$ 和 $\Delta\mathbf{v}^f(t)$ 的值首先写为为矩阵 \mathbf{Y}^f , \mathbf{U}^f 和 \mathbf{V}^f , 然后进一步整合为一个更大的矩阵 \mathbf{YUV}^f 。同理, $\mathbf{F}(z^{-1})$, $\Gamma(z^{-1})$ 和 $\Theta(z^{-1})$ 也可以融合为 $\mathbf{F}\Gamma\Theta$ 。从而有方程 (8.116), 其中, 有方程 (8.117) 至 (8.121)。

Um die Berechnung der GPC-Regelung weiter zu vereinfachen, kann nun, wenn die zukünftigen Werte der Störeinflüsse $\mathbf{v}(t)$ bekannt sind, gemäß der auf Seite 213 beschriebenen Vorgehensweise $\mathbf{F}'' = \mathbf{H}'\mathbf{V} + \mathbf{F}'$ zusammengefasst werden. In diesem Fall ergibt sich eine modifizierte freie Antwort:

$$\mathbf{F}'' = \mathbf{H}' \cdot \mathbf{V} + \mathbf{F}\Gamma\Theta \cdot \mathbf{YUV}^f = \mathbf{H}'\mathbf{F}\Gamma\Theta \cdot \mathbf{VYUV}^f \quad (8.122)$$

mit

$$\mathbf{H}'\mathbf{F}\Gamma\Theta = \begin{bmatrix} \mathbf{H}'_0 & 0 & \cdots & 0 \\ \mathbf{H}'_1 & \mathbf{H}'_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}'_{N_p-1} & \mathbf{H}'_{N_p-2} & \cdots & \mathbf{H}'_0 \\ \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\ \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,na} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\ \Gamma_{1,0} & \Gamma_{1,1} & \cdots & \Gamma_{1,nb-1} \\ \Gamma_{2,0} & \Gamma_{2,1} & \cdots & \Gamma_{2,nb-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma_{N_p,0} & \Gamma_{N_p,1} & \cdots & \Gamma_{N_p,nb-1} \\ \Theta_{1,0} & \Theta_{1,1} & \cdots & \Theta_{1,nd-1} \\ \Theta_{2,0} & \Theta_{2,1} & \cdots & \Theta_{2,nd-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Theta_{N_p,0} & \Theta_{N_p,1} & \cdots & \Theta_{N_p,nd-1} \end{bmatrix} \quad (8.123)$$

$$\mathbf{VYUV}^f = \begin{bmatrix} \Delta \mathbf{v}(t+1) \\ \vdots \\ \Delta \mathbf{v}(t+N_p) \\ \mathbf{y}^f(t) \\ \vdots \\ \mathbf{y}^f(t-na) \\ \Delta \mathbf{u}^f(t-1) \\ \vdots \\ \Delta \mathbf{u}^f(t-nb) \\ \Delta \mathbf{v}^f(t-1) \\ \vdots \\ \Delta \mathbf{v}^f(t-nd+1) \end{bmatrix} \quad (8.124)$$

If future values of the disturbances $\mathbf{v}(t)$ are known, the combination $\mathbf{F}'' = \mathbf{H}'\mathbf{V} + \mathbf{F}'$ can be made according to the approach described on page 213 in order to further simplify the calculation of the GPC control. In this case, the modified free response results into equation (8.122), in which equations (8.123) and (8.124).

在未来时刻的扰动变量 $\mathbf{v}(t)$ 已知的情况下，为了进一步简化 GPC 控制的计算，可根据第 213 页所述的处理方式将相应矩阵整合为 $\mathbf{F}'' = \mathbf{H}'\mathbf{V} + \mathbf{F}'$ 的形式。此时，可以得到修正后的自由响应表达式，见方程 (8.122)，其中有方程 (8.123) 和 (8.124)。

Sind die Störgrößen $\mathbf{v}(t)$ zwar mit ihrem aktuellen Wert bekannt, ihr zukünftiger Verlauf jedoch unbekannt, nimmt man sie als konstant an, wodurch der Ausdruck $\Delta\mathbf{v}(t)$ zu Null wird und der Term $\mathbf{H}'\mathbf{V}$ verschwindet. Die Berechnung einer modifizierten freien Antwort \mathbf{F}'' ist dann unnötig und in Formel (8.103) wird \mathbf{F}'' durch \mathbf{F}' ersetzt.

8.3.4 Berücksichtigung des Regelhorizonts/Consideration of the control horizon/考虑控制步长后的简化

Entsprechend Kap. 8.1.4 und 8.2.4 ist auch bei Einsatz der Filtereigenschaften von GPC eine Vereinfachung der Berechnungen durch Berücksichtigung des Regelhorizonts N_u , $N_u \leq N_2$ möglich. Gl. (8.115) wird dann zu:

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{y}}(t + N_1) \\ \vdots \\ \hat{\mathbf{y}}(t + N_2) \end{bmatrix} &= \begin{bmatrix} \mathbf{G}'_{N_1-1} & \mathbf{G}'_{N_1-2} & \cdots & \mathbf{G}'_{N_1-N_u} \\ \vdots & \vdots & & \vdots \\ \mathbf{G}'_{N_2-1} & \mathbf{G}'_{N_2-2} & \cdots & \mathbf{G}'_{N_2-N_u} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}(t) \\ \vdots \\ \Delta\mathbf{u}(t + N_u - 1) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{H}'_{N_1-1} & \mathbf{H}'_{N_1-2} & \cdots & \mathbf{H}'_{N_1-N_u} \\ \vdots & \vdots & & \vdots \\ \mathbf{H}'_{N_2-1} & \mathbf{H}'_{N_2-2} & \cdots & \mathbf{H}'_{N_2-N_u} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{v}(t + 1) \\ \vdots \\ \Delta\mathbf{v}(t + N_u) \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{f}'(t + N_1) \\ \vdots \\ \mathbf{f}'(t + N_2) \end{bmatrix} \end{aligned} \quad (8.125)$$

oder

$$\mathbf{Y}_{N_{12}} = \mathbf{G}'_{N_{12u}} \mathbf{U}_{N_u} + \mathbf{H}'_{N_{12u}} \mathbf{V} + \mathbf{F}'_{N_{12}} \quad (8.126)$$

Die gleichen Vereinfachungen lassen sich für die Formel zur Berechnung der optimalen Stellgrößen (8.103) treffen:

$$\mathbf{U}_{N_u} = (\mathbf{G}'_{N_{12u}}{}^T \mathbf{G}'_{N_{12u}} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}'_{N_{12u}}{}^T (\mathbf{W}_{N_{12}} - \mathbf{F}''_{N_{12}}) \quad (8.127)$$

Davon ausgehend, dass in den meisten Fällen die zukünftigen Störgrößen unbekannt sein werden, kann, wie weiter oben ausgeführt, $\mathbf{F}''_{N_{12}}$ durch $\mathbf{F}'_{N_{12}}$ ersetzt werden. Die Matrix $\mathbf{H}'_{N_{12u}}$ ist dann ebenfalls überflüssig und braucht nicht berechnet werden.

If the current values of the disturbances $\mathbf{v}(t)$ are known, but not their future trend, then they are assumed to be constant. This leads to the fact that the expression $\Delta\mathbf{v}(t)$ becomes zero and that the term $\mathbf{H}'\mathbf{V}$ disappears. Then, the calculation of a modified free response \mathbf{F}'' is unnecessary and in equation (8.103), \mathbf{F}'' is replaced with \mathbf{F}' .

8.3.4 Consideration of the control horizon

According to chapters 8.1.4 and 8.2.4 a simplification of the calculations under consideration of the control horizon N_u , $N_u \leq N_2$ is possible, even if the filter characteristics of GPC are used. Then equation (8.115) becomes equation (8.125) or equation (8.126). The same simplifications can be carried out for equation (8.103) with which the calculation of optimum values for the actuating variables is done, see equation (8.127). Based on the fact that in most cases future values of the disturbances will be unknown, $\mathbf{F}''_{N_{12}}$ can be replaced with $\mathbf{F}'_{N_{12}}$ as already described above. Then, the matrix $\mathbf{H}'_{N_{12u}}$ is redundant, too, and it is not necessary to calculate it.

如果仅知晓当前时刻的扰动变量 $\mathbf{v}(t)$ ，而未来时刻的扰动变量 $\mathbf{v}(t)$ 是未知的，那么则通过假设未来时刻的值与当前时刻的值相同，从而可使得变化量 $\Delta\mathbf{v}(t)$ 为零，而且 $\mathbf{H}'\mathbf{V}$ 项也消失。此时，并无必要计算修正的自由响应 \mathbf{F}'' ，方程 (8.103) 中的 \mathbf{F}'' 可直接使用 \mathbf{F}' 进行替代。

8.3.4 考虑控制步长后的简化

类似于章节 8.1.4 和 8.2.4，结合了滤波特性的 GPC 也同样可以通过将控制步长 N_u 限制在 $N_u \leq N_2$ 的范围以减小计算量。方程 (8.115) 可简化为方程 (8.125) 或方程 (8.126)。用于计算最优控制量的方程 (8.103) 也同样可简化为方程 (8.127)。

由于在大多情况下，未来时刻的扰动变量是未知的，如前所述，可使用 $\mathbf{F}'_{N_{12}}$ 来代替 $\mathbf{F}''_{N_{12}}$ 。矩阵 $\mathbf{H}'_{N_{12u}}$ 也同样是冗余的，并无必要进行计算。

Die freie Antwort $\mathbf{F}'_{N_{12}}$ lautet demnach:

$$\mathbf{F}'_{N_{12}} = \mathbf{F}\Gamma\Theta_{N_{12}} \cdot \mathbf{YUV}^f \quad (8.128)$$

wobei

$$\mathbf{F}\Gamma\Theta_{N_{12}} = \begin{bmatrix} \mathbf{F}_{N_{1,0}} & \mathbf{F}_{N_{1,1}} & \cdots & \mathbf{F}_{N_{1,na}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_{2,0}} & \mathbf{F}_{N_{2,1}} & \cdots & \mathbf{F}_{N_{2,na}} \\ \Gamma_{N_{1,0}} & \Gamma_{N_{1,1}} & \cdots & \Gamma_{N_{1,nb-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma_{N_{2,0}} & \Gamma_{N_{2,1}} & \cdots & \Gamma_{N_{2,nb-1}} \\ \Theta_{N_{1,0}} & \Theta_{N_{1,1}} & \cdots & \Theta_{N_{1,nd-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \Theta_{N_{2,0}} & \Theta_{N_{2,1}} & \cdots & \Theta_{N_{2,nd-1}} \end{bmatrix} \quad (8.129)$$

8.4 Experimentelle Ergebnisse/Experimental results/实验结果

8.4.1 Stromregelung/Current control/电流控制

Die Wirksamkeit der Störgrößenberücksichtigung eines Mehrgrößen-GPC-Reglers wurde experimentell erprobt. Dazu wurden die Statorstromkomponenten i_{sd} und i_{sq} mit verschiedenen Sollwertsprüngen beaufschlagt und das Verhalten des Reglers für den jeweils anderen Statorstromanteil beobachtet. Wie Abbildung 2.3 auf Seite 15 zeigt, ist die Kreuzkopplung zwischen i_{sd} und i_{sq} proportional zur Drehfrequenz ω_s . Folglich fällt bei stehender Maschine ($\omega = 0$) die Rückkopplung des Rotorfeldes ψ_r auf die Statorströme i_s weg. Für die Experimente wurde daher die mechanische Drehgeschwindigkeit ω mit $\omega = 0,5$ (halbe Nenndrehzahl) vorgegeben; die elektrische Drehgeschwindigkeit ω_s wird bedingt durch den Schlupf geringfügig höher liegen. Die verwendeten Prädiktionshorizonte betragen $N_p = 4$ und $N_u = 2$; die Grade der einzelnen Systempolynome haben die Werte $na = 1$, $nb = 1$, $nd = 1$ und $nt = 1$. Die dem MIMO-GPC-Regler zugrunde liegenden System- oder Modellmatrizen sind folgendermaßen besetzt:

$$\mathbf{G}_d(z^{-1}) = \begin{bmatrix} \frac{0,1484 z^{-1}}{1-0,9873 z^{-1}} & 0 & 0 \\ 0 & \frac{0,1484 z^{-1}}{1-0,9873 z^{-1}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (8.130)$$

$$\mathbf{H}_d(z^{-1}) = \begin{bmatrix} 0 & \frac{0,03197 z^{-1}}{1-0,9873 z^{-1}} & 0 \\ \frac{-0,03197 z^{-1}}{1-0,9873 z^{-1}} & 0 & \frac{-0,1423 z^{-1}}{1-0,9873 z^{-1}} \\ 0 & 0 & 0 \end{bmatrix} \quad (8.131)$$

Therefore, the free response $\mathbf{F}'_{N_{12}}$ is given by equation (8.128), in which equation (8.129).

8.4 Experimental results

8.4.1 Current control

The effectiveness of considering disturbances using a multidimensional GPC controller was experimentally tested. For these tests, different reference value steps were applied to the stator current components i_{sd} and i_{sq} and the controller behavior for the other current component was observed. As Figure 2.3 on page 15 shows, the cross coupling between i_{sd} and i_{sq} is proportional to the rotating frequency ω_s . Therefore, considering a machine in standstill operation ($\omega = 0$), a feedback of the rotor flux ψ_r on the stator current i_s is not existence. Because of this, the reference value for the mechanical rotating speed ω was set to $\omega = 0.5$ (half nominal speed) for the experiments; the electrical rotating speed ω_s will be slightly higher because of the slip. The used prediction horizons are $N_p = 4$ and $N_u = 2$; the degrees of the individual system polynomials have the values $na = 1$, $nb = 1$, $nd = 1$ and $nt = 1$. In the following, the system matrices for the MIMO-GPC controller are given, see equations (8.130) and (8.131).

因此，自由响应 $\mathbf{F}'_{N_{12}}$ 表达式为方程 (8.128)，其中，有方程 (8.129)。

8.4 实验结果

8.4.1 电流控制

多变量 GPC 控制器对于考虑扰动因素后的有效性已经得到了验证。为此，通过分别设置定子电流分量 i_{sd} 和 i_{sq} 不同大小的参考值阶跃变化，并观察了控制器对其所对应的电流分量所作出的响应。正如第 15 页图 2.3 所示， i_{sd} 和 i_{sq} 的交叉耦合与交流频率 ω_s 成正比关系。由此可知，对于静止条件 ($\omega = 0$) 下的电机来说，转子磁链 ψ_r 对定子电流 i_s 的耦合作用则相应消失。相关的实验设置如下：给定参考机械转速为 $\omega = 0.5$ （额定转速的二分之一）；由于转差率的缘故，电气旋转角速度 ω_s 相对稍许要高一些。所使用的预测步长和控制步长分别为 $N_p = 4$ 和 $N_u = 2$ ；单个系统多项式的阶数为 $na = 1$ ， $nb = 1$ ， $nd = 1$ 和 $nt = 1$ 。对于 MIMO-GPC 控制器来说，最为基本的系统或模型矩阵具有如下形式，见方程 (8.130) 和 (8.131)。

$$\mathbf{T}(z^{-1}) = \begin{bmatrix} 1 - 0,95 z^{-1} & 0 & 0 \\ 0 & 1 - 0,95 z^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.132)$$

$$\mathbf{A}(z^{-1}) = \begin{bmatrix} 1 - 0,9873 z^{-1} & 0 & 0 \\ 0 & 1 - 0,9873 z^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.133)$$

$$\mathbf{B}(z^{-1}) = \begin{bmatrix} 0,1484 z^{-1} & 0 & 0 \\ 0 & 0,1484 z^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (8.134)$$

$$\mathbf{D}(z^{-1}) = \begin{bmatrix} 0 & 0,03197 z^{-1} & 0 \\ -0,03197 z^{-1} & 0 & -0,1423 z^{-1} \\ 0 & 0 & 0 \end{bmatrix} \quad (8.135)$$

$$\lambda = \begin{bmatrix} 0,003 & 0 & 0 \\ 0 & 0,003 & 0 \\ 0 & 0 & 0,003 \end{bmatrix} \quad (8.136)$$

Die Messergebnisse sind in Abb. 8.1 zu finden. Die Diagramme zeigen die Wirkung eines Sprungs im drehmomentbildenden Strom von $i_{sq} = 0$ auf $i_{sq} = 1$ auf den mit dem gleichen MIMO-GPC-Regler geregelten feldbildenden Strom i_{sd} . Dieser wird auf den konstanten Wert von $i_{sd} = 0,2$ geregelt. Abb. 8.1(a) zeigt, dass bei Verwendung eines GPC-Reglers ohne Berücksichtigung der Störeinflüsse der feldbildende Anteil i_{sd} während der Anstiegsphase von i_{sq} eine erkennbare Abweichung von seinem Sollwert aufweist. Berücksichtigt man den bekannten Einfluss der Kreuzkopplung im Regler, so verschwindet der Regelfehler von i_{sd} während des Anstiegs von i_{sq} vollständig (Abb. 8.1(b)).

8.4.2 Rechenzeiten/Computation times/计算时间

Die für den mehrdimensionalen Stromregler inklusive Störgrößenberücksichtigung und interner Filterung notwendige Rechenzeit liegt mit $4,1 \mu\text{s}$ etwa in der gleichen Größenordnung wie diejenige für zwei eindimensionale GPC-Stromregler (s. Kap. 6.3.4 auf Seite 133). Da die MIMO-Regelung jedoch im Gegensatz zu den beiden eindimensionalen Stromreglern die Kreuzkopplung zwischen i_{sd} und i_{sq} als Störeinfluss berücksichtigt, ist die Mehrgrößenregelung der getrennten Regelung des feld- und drehmomentbildenden Stroms eindeutig vorzuziehen.

See equations (8.132) to (8.136).

The measurement results can be found in Figure 8.1. The plots show the effect of a step in the torque producing current component from $i_{sq} = 0$ to $i_{sq} = 1$ to the field producing current component i_{sd} , which is controlled with the same MIMO-GPC controller. The field producing current component is controlled to the constant value $i_{sd} = 0.2$. Figure 8.1(a) shows that during the increase of the torque producing current component i_{sq} the flux producing current component i_{sd} shows a recognizable deviation from its reference value when a GPC controller without consideration of disturbances is used. If the known influence of the cross coupling is considered in the controller, the control error of i_{sd} disappears nearly completely during the increase of i_{sq} (Figure 8.1(b)).

8.4.2 Computation times

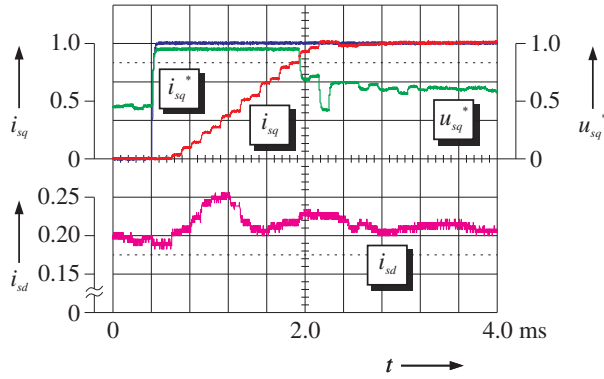
The computation time necessary for the multidimensional current controller with consideration of disturbances is around $4.1 \mu\text{s}$, i. e. approximately in the same range as the calculation time for two uni-dimensional GPC current controllers (see Chapter 6.3.4 on page 133). Since, in contrast to the two uni-dimensional currents controllers, the MIMO control considers the cross coupling between i_{sd} and i_{sq} as a disturbance input, multivariable control for an individual control of the flux and torque producing current components is clearly preferable.

见方程 (8.132) 至 (8.136)。

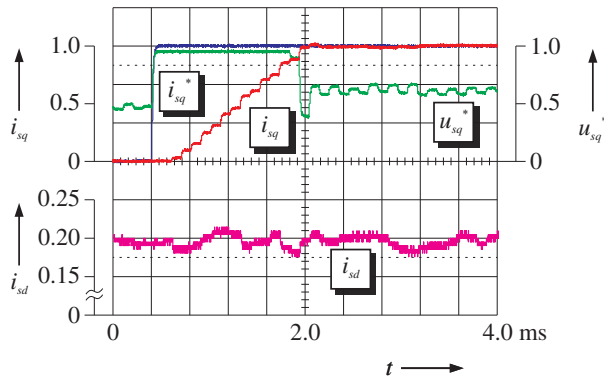
图 8.1 给出了相应的测量结果。可以看出，定子电流转矩分量由 $i_{sq} = 0$ 至 $i_{sq} = 1$ 的阶跃变化对于所使用同一 MIMO-GPC 控制器控制的定子电流励磁分量 i_{sd} 产生的影响。定子电流励磁分量 i_{sd} 的目标是被控制在一个恒定的数值 $i_{sd} = 0.2$ 。然而，图 8.1(a) 表明，在使用一个未考虑扰动因素的 GPC 控制器时，定子电流励磁分量 i_{sd} 在 i_{sq} 的上升阶段出现了一个明显的相对于其给定值的偏差；如果可以在控制器中考虑到已知的交叉耦合因素，那么，定子电流励磁分量 i_{sd} 在 i_{sq} 的上升阶段的控制偏差则会消失，如图 8.1(b) 所示。

8.4.2 计算时间

考虑到扰动因素以及内部滤波的多维电流控制器所需的计算时间大约为 $4.1 \mu\text{s}$ ，这一数值与两个单维的 GPC 电流控制器所需的时间大致在同一个数量级（见章节 6.3.4 第 133 页）。在上述前提条件下，由于 MIMO 控制器额外考虑到了 i_{sd} 和 i_{sq} 之间的交叉耦合因素，所以可实现定子电流励磁分量和转矩分量独立控制的多变量控制方式明显更具优势。



(a)



(b)

8.1: Mehrgrößen-Stromregelung: (a) ohne Störgrößenberücksichtigung, (b) mit Störgrößenberücksichtigung;

Multidimensional current control: (a) Without consideration of disturbances, (b) With consideration of disturbances;

多变量电流控制: (a) 不考虑扰动, (b) 考虑扰动

Notizen/Notes/笔记

8.5 Vergleichende Zusammenfassung/Comparative summary/对比性总结

Eine mathematische Betrachtung der diversen Variationen von MIMO-GPC zeigt die genauen Unterschiede und ist hilfreich zur Interpretation der obigen Ergebnisse. Tabelle 8.1 zeigt eine Übersicht über die Berechnungsformeln zur Ermittlung der freien Antwort und der zukünftigen, optimalen Stellgrößen in den verschiedenen Konstellationen.

	ohne Störgrößen	mit Störgrößen
ohne Filter	$\mathbf{F} = \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{G}_p(z^{-1})\Delta\mathbf{u}(t-1)$ $= \mathbf{F}\mathbf{G}_p \cdot \mathbf{Y}\mathbf{U}$ $\bar{\mathbf{U}} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}^T(\mathbf{W} - \mathbf{F})$	$\mathbf{F} = \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{G}_p(z^{-1})\Delta\mathbf{u}(t-1) + \mathbf{H}_p(z^{-1})\Delta\mathbf{v}(t)$ $= \mathbf{F}\mathbf{G}_p\mathbf{H}_p \cdot \mathbf{Y}\mathbf{U}\mathbf{V}$ $\mathbf{F}' = \mathbf{H} \cdot \mathbf{V} + \mathbf{F}$ $\bar{\mathbf{U}} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}^T(\mathbf{W} - \mathbf{F}')$
mit Filter	$\mathbf{F}' = \mathbf{F}(z^{-1})\mathbf{y}^f(t) + \Gamma(z^{-1})\Delta\mathbf{u}^f(t-1)$ $= \mathbf{F}\Gamma \cdot \mathbf{Y}\mathbf{U}^f$ $\bar{\mathbf{U}} = (\mathbf{G}'^T\mathbf{G}' + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}'^T(\mathbf{W} - \mathbf{F}')$	$\mathbf{F}' = \mathbf{F}(z^{-1})\mathbf{y}^f(t) + \Gamma(z^{-1})\Delta\mathbf{u}^f(t-1) + \Theta(z^{-1})\Delta\mathbf{v}^f(t)$ $= \mathbf{F}\Gamma\Theta \cdot \mathbf{Y}\mathbf{U}\mathbf{V}^f$ $\mathbf{F}'' = \mathbf{H}' \cdot \mathbf{V}' + \mathbf{F}'$ $\bar{\mathbf{U}} = (\mathbf{G}'^T\mathbf{G}' + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}'^T(\mathbf{W} - \mathbf{F}'')$

8.1: Vergleich der verschiedenen MIMO-GPC-Versionen; /Comparison of different MIMO-GPC variations; /MIMO-GPC 的不同变体形式的对比

Wie aus Kap. 8.2.3 auf Seite 195 ff. hervorgeht, sind \mathbf{G} bzw. \mathbf{G}' jeweils völlig identisch, egal ob Störgrößen berücksichtigt werden oder nicht. Folglich ist der einzige Unterschied zwischen den Strukturen der MIMO-GPC-Regler mit und ohne Berücksichtigung von bekannten Störgrößen die Berechnung der freien Antwort \mathbf{F} bzw. \mathbf{F}' . Dabei wird nur der Term $\mathbf{H}_p(z^{-1})\Delta\mathbf{v}(t)$ bzw. $\Theta(z^{-1})\Delta\mathbf{v}^f(t)$ hinzugefügt, da die \mathbf{F} , \mathbf{G}_p , \mathbf{F}' und Γ alle von der Berücksichtigung der Störgrößen unabhängig sind. Untersucht man die Werte in der Polynommatrix $\Theta(z^{-1})$, so stellt man fest, dass der Einfluss der Störgrößen auf die Berechnung der freien Antwort relativ gering ist, was aber das tatsächliche Verhalten der Maschine beschreibt. Folglich ändert sich auch das Verhalten der Regelung nur wenig, wenn die Störeinflüsse im GPC-Regler berücksichtigt werden. Trotzdem ist, wie Abb. 8.1 zeigt, eine deutliche Unterdrückung der Kreuzkopplung feststellbar.

8.5 Comparative summary

A mathematical view of the different variations of MIMO-GPC shows the exact differences and is helpful for the interpretation of the results shown above. Table 8.1 gives an overview of the various equations for the calculation of the free response and of future optimum values for the actuating variables in different configurations.

As already described in Chapter 8.2.3 on page 195 et seq., \mathbf{G} and \mathbf{G}' are absolutely identical, no matter if disturbance inputs are taken into consideration, or not. Thus, the only difference between the structures of MIMO-GPC controllers with and without consideration of known disturbances is the calculation of the free response \mathbf{F} or \mathbf{F}' . Thereby, only the term $\mathbf{H}_p(z^{-1})\Delta\mathbf{v}(t)$ or $\Theta(z^{-1})\Delta\mathbf{v}^f(t)$ is added, since \mathbf{F} , \mathbf{G}_p , \mathbf{F}' and Γ are all independent of the consideration of disturbances. Examining the values in the polynomial matrix $\Theta(z^{-1})$, it can be seen that the influence of the disturbances on the calculation of the free response is quite small which, however, describes the actual behavior of the machine. Thus, the behavior of the controller does not change much if the disturbances are taken into account in the GPC controller. Despite this, as shown in Figure 8.1, a significant suppression of the cross coupling is noticeable.

8.5 对比性总结

通过对 MIMO-GPC 的不同变体形式进行数学层面的考察, 可以看出其明显的区别, 并且也有助于解释前面所得到的结果。表 8.1 给出了在不同设置下的为求得自由响应和未来的、最优控制变量所需的相应计算方程概览。

由章节 8.2.3 第 195 页相关内容可知, 无论是否考虑扰动因素, \mathbf{G} 和 \mathbf{G}' 都是完全一致的。从而, 考虑和未考虑已知扰动的两种 MIMO-GPC 控制器之间的区别仅在于自由响应 \mathbf{F} 或 \mathbf{F}' 的计算。为此, 需额外增添的项也仅包括 $\mathbf{H}_p(z^{-1})\Delta\mathbf{v}(t)$, 以及 $\Theta(z^{-1})\Delta\mathbf{v}^f(t)$, 因为 \mathbf{F} , \mathbf{G}_p , \mathbf{F}' 和 Γ 并不受到扰动因素的影响。通过考察多项式矩阵 $\Theta(z^{-1})$ 的值可知, 扰动因素对于自由响应计算的影响非常有限, 然而, 它确实描述了电机的实际特性。可以得出结论, 即使在 GPC 控制器中考虑了扰动因素, 控制器的特性的改变也非常微小。尽管如此, 如图 8.1 所示, 交叉耦合的抑制作用依然非常明显。

9 Direkte Modellbasierte Prädiktivregelung /Direct Model-based Predictive Control /直接模型预测控制

Die bisher erläuterten Verfahren zur optimalen Regelung eines Antriebs mittels Long-range Predictive Control beruhen alle auf dem Grundsatz, dass ein optimaler Spannungsraumzeiger bestimmt wird, der im nächsten Abtastzyklus an die zu regelnde Maschine angelegt werden soll. Als Spannungsquelle für die Maschine wird im Allgemeinen ein Zweipunktumrichter mit Spannungszwischenkreis verwendet, wie ihn Abb. 9.1 darstellt. Dieser besteht aus drei Halbbrücken, hier mit den Buchstaben a , b und c gekennzeichnet, welche die drei Phasen der angeschlossenen Maschine jeweils mit dem positiven oder dem negativen Anschluss der Zwischenkreisspannung u_d verbinden können. Daraus folgt, dass nur eine endliche Anzahl von möglichen Spannungsraumzeigern \mathbf{u}_s existiert. In der Regelungstechnik bezeichnet man solche Systeme, die aus voneinander abhängigen physikalischen Gesetzen und logischen Regeln bestehen sowie Beschränkungen des Betriebsbereichs unterliegen, d. h. deren Eingangs-, Zustands- und Ausgangsvariablen teilweise kontinuierliche, teilweise aber auch diskrete Werte annehmen können, als *Gemischt-Logische Dynamische Systeme* bzw. *Mixed Logical Dynamical Systems (MLD)* oder auch als *Hybride Systeme*.

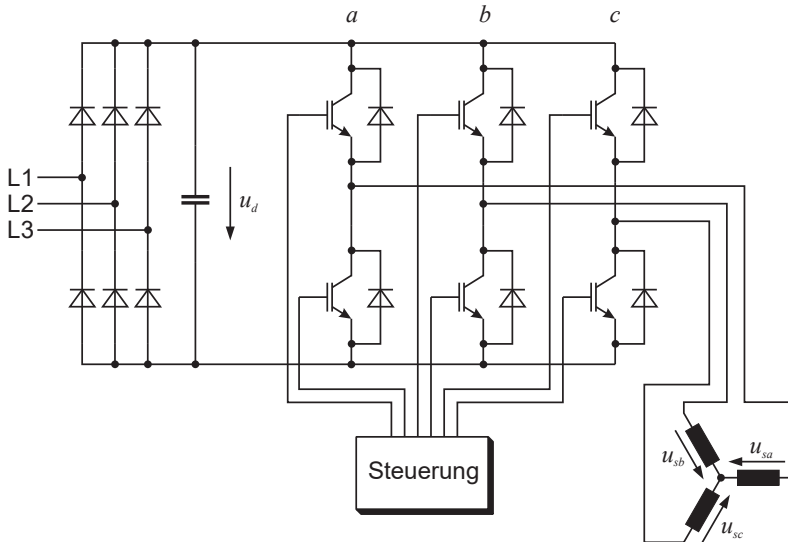
Üblicherweise geht man nun in der Antriebstechnik so vor, dass dem Reglerausgang ein sogenannter Modulator nachgeschaltet wird, der den Sollspannungsraumzeiger diskretisiert (Abb. 9.2(a)). Das bedeutet, dass der Modulator den gewünschten, aber vom Umrichter nicht darstellbaren Raumzeiger aus drei anderen, vom Umrichter erzeugbaren Spannungswerten synthetisiert, so dass sich im zeitlichen Mittel über den Abtastzyklus der gewünschte Wert ergibt. Da konventionelle modellbasierte Verfahren diese Tatsache nicht berücksichtigen, wird die eigentlich optimale Regelung durch den weder im Reglerentwurf noch im Modell berücksichtigten Modulator beeinträchtigt. Eine Optimalregelung, die den Fakt, dass das Stellglied nur eine endliche Anzahl von möglichen Ausgangszuständen annehmen kann, berücksichtigt, muss den bisher diskutierten MPC-Verfahren überlegen sein.

All the techniques for optimum control of a drive by means of Long Range Predictive Control (LRPC), which have been described so far, are based on the principle that an optimum voltage space vector is determined, which will be applied to the machine being controlled in the next switching cycle. Commonly a two-level inverter fed from a DC link voltage is used as voltage source for the machine, as shown in Figure 9.1. This inverter consists of three half bridges, labeled with the letters a , b and c . They can independently connect the three phases of the machine either to the positive or to the negative DC link voltage u_d . Hence, it follows that only a finite number of possible voltage space vectors \mathbf{u}_s can exist. In control engineering, such systems, consisting of interdependent physical laws, logical rules and which are subjected to constraints of the operating range, i. e. their input variables, state variables and output variables can assume partly continuous but also partly discrete values, are named *Mixed Logical Dynamical Systems (MLD)* or *Hybrid Systems*.

In drive control, the normal approach is that the controller output is fed to a so-called modulator which discretizes the reference space voltage vector (Figure 9.2(a)). This means that the modulator synthesizes the desired space voltage vector which can normally not be delivered from the inverter out of three other space voltage vectors that can be delivered. The value of these voltage space vectors averaged over one sampling interval is equal to the desired reference value. As conventional model-based techniques do not consider this fact, the actual optimum control is impaired by the modulator which is neither considered in the controller design nor in the model. An optimum control which regards the fact that the inverter can realise only a finite number of possible output states must be superior to the MPC techniques discussed so far.

到目前为止，所讲述的基于多步长预测控制实现驱动系统最优控制的方法均基于同一原理，也即确定一个在下一采样周期可用于被控电机的最优电压控制矢量。一般来说，通常使用如图 9.1 所示的带有中间直流电压的两电平逆变器作为电机的电压源，它由三个桥臂组成，分别使用字母 a 、 b 和 c 标识，且作为枢纽与三相电机的输出端口相连接，从而可以提供正或负的中间直流电压 u_d 。这也意味着仅存在有限数量的可能的电压空间矢量 \mathbf{u}_s 。在控制技术中，当一个系统是由相互依存的物理规律和逻辑法则所组成，并且受到运行范围的限制，通常被称为混合系统。换言之，系统的输入变量、状态变量和输出变量部分是连续的值，但也有一部分是离散的值，所以称之为混合动态系统。

在驱动技术领域，通常在控制器输出端口连接一个所谓的调制器，用于离散化参考空间电压矢量，如图 9.2(a) 所示。也就是说，调制器通过三个可由逆变器直接产生的电压值来合成所期望的无法由逆变器直接产生的空间矢量，以使其在一个采样周期内的时间平均值等效于期望值。由于传统的基于模型的方法没有考虑到上述事实，实际的最优控制会由于既未在控制器设计，也未在模型中考虑上述调制器而受到影响。一种能够考虑到执行机构有限数量的可能输出变量的最优控制方法一定会优于迄今为止已讨论的 MPC 方法。



9.1: Schaltbild Zweipunktumrichter mit Spannungszwischenkreis;

Two-level inverter circuit with DC link

帶中间直流环节的两电平逆变器电路

Daher soll im Folgenden ein neues MPC-Verfahren vorgestellt werden, welches eine direkte Umrichtersteuerung ermöglicht, d. h. der Regelalgorithmus ermittelt direkt den optimalen Umrichterschaltzustand für den nächsten Abtastzyklus. Dieses Verfahren wird *Direct Model Predictive Control* oder *DMPC* genannt (Abb. 9.2(b)).

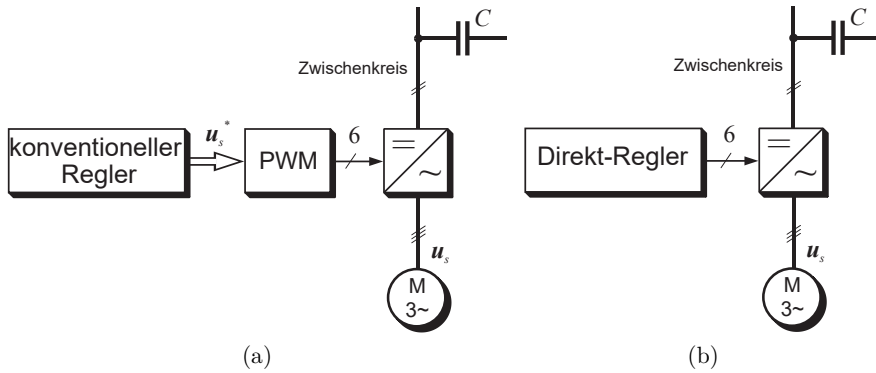
Das grundlegende Funktionsprinzip einer direkten modellbasierten Prädiktivregelung lässt sich am besten am Beispiel eines Schachspielers verdeutlichen. Dieser berechnet bis zu einem „Prädiktionshorizont“ alle diskreten Werte von Stellgrößen, in diesem Fall die Züge seiner Figuren, für den weiteren Spielverlauf inklusive der möglichen gegnerischen Bewegungen, also dem Verhalten der Regelstrecke, voraus. Dabei geht er ähnlich dem Verzweigungsprinzip (s. Kap. 9.4.1) vor, denn Überlegungen zu Zugkombinationen, die erkennbar zu einem Nachteil für ihn führen, werden gar nicht erst weiter vertieft, sondern er wird statt dessen gleich nach Alternativen suchen. Nach Abschluss der Optimierung wird er den für ihn bestmöglichen Zug wählen, ausführen und dann auf die Reaktion des Gegners warten, was wiederum dem Receding-Horizon-Prinzip entspricht. Nach der Systemreaktion in Form des gegnerischen Zuges wird die ganze Vorausberechnung und Optimierung von Neuem gestartet.

Hence, a new MPC scheme is introduced in the following: It allows direct inverter control, i. e. the control algorithm directly determines the optimum inverter switching state for the next sampling cycle. This technique is called *Direct Model Predictive Control* or *DMPC* (Figure 9.2(b)).

The basic functional principle of direct model-based predictive control can be explained best with an example of playing chess. The chess player precalculates all discrete values of the variables, in this case the moves of his chessmen, up to a “prediction horizon” for the further match, including the possible movements of his opponent, i. e. the behavior of the controlled system. Besides, he takes his decisions similar to the Branch and Bound principle (see Chapter 9.4.1) because movement combinations which lead to a recognizable disadvantage for him are not examined further, instead he immediately looks for alternatives. After finishing the optimization, he will choose the best possible move for him, carry it out and then wait for the reaction of the opponent which corresponds again to the principle of Receding Horizon. After the system reaction, i. e. the opponent’s move, the whole precalculation and optimization will be started again.

因此，接下来将介绍一种新型的 MPC 方法，它可以实现逆变器的直接调节。换言之，控制算法可直接确定用于下一采样周期的逆变器开关状态，可称之为直接模型预测控制，如图 9.2(b) 所示。

直接模型预测控制的基本工作原理可以国际象棋为例进行阐述：棋手提前计算出在接下来的预测步长范围内所有的离散输出变量，具体的说，就是考虑在接下来的步数内如何移动棋子，这需要考虑到后续的棋盘局势变化及对手潜在的棋子走向，很明显可以看作是被控系统的特性。棋手的决策一定程度上类似于章节 9.4.1 介绍的分支界限法，因为若某一步棋子移动的组合方式对于棋手来说是不利的，那么则停止继续可能的走向，并另辟新的路径。经深思熟虑，棋手会选择最优的移动方式并完成棋子的移动，接下来就是等待对手做出反应，上述过程的重复也即前面所述的滚动优化原则。在被控系统做出反应后，也即对手完成了棋子移动，棋手需根据最新的当前棋局继续进行预测，并做出最合适的决策。



9.2: Prinzip einer Maschinenregelung: (a) konventionell, (b) direkt;
 Principle of machine control: (a) conventional, (b) direct;
 电机控制基本原理: (a) 传统的, (b) 直接的

9.1 Publierte Verfahren/Published techniques/已发表的方法

Wie schon in Kap. 5 erwähnt, sind modellbasierte Regelungen bereits seit über 30 Jahren in der Regelungstechnik bekannt und werden auch angewendet. Einen kurzen historischen Überblick bieten z. B. die Veröffentlichungen von Clarke [30] und Morari/Lee [88]. Nichtsdestotrotz hat die implizite Berücksichtigung schaltender Stellglieder bisher kaum Eingang in diese Regelverfahren gefunden. Eine Ausnahme stellen hier lediglich die an der RWTH Aachen durchgeführten Arbeiten von Hoffmann [51, 52] und Schmitz [105] aus den 1980er Jahren dar. Erst in jüngster Zeit hat die Forschung bezüglich der Regelung hybrider Systeme wieder an Bedeutung gewonnen und es sind Veröffentlichungen bekannt, die sich mit der Anwendung modellbasierter Prädiktivregelverfahren auf diese Systemstrukturen beschäftigen [10].

Im Jahr 1984 veröffentlichte Hoffmann in seiner Dissertation [51] unter dem Namen *Adaptive Zweipunktregelung* erstmalig eine MPC-Regelung, bei der die besonderen Randbedingungen eines Stellglieds mit einer definiten Anzahl diskreter Schaltzustände bereits im Ansatz berücksichtigt werden. Hoffmann beschränkte sich dabei auf einen Zweipunktregler. Die Beschreibung der Regelstrategie zeigt klar, dass es sich um eine typische MPC-Regelung handelt:

- Der eigentliche Regler beinhaltet ein Streckenmodell. Die Modellparameter können online adaptiert werden.

9.1 Published techniques

As already mentioned in Chapter 5, model-based control schemes are already known for more than 30 years in control engineering and are also applied in practical applications. A short historical overview is given e.g. in the publications of Clarke [30] and Morari/Lee [88]. Nevertheless, the implicit consideration of switching actuators has barely found an entree in these published control techniques. The work carried out at the Technical University of Aachen in the 1980s by Hoffmann [51, 52] and Schmitz [105] can be taken as an exception. Just recently, research in the control of Hybrid Systems has gained in importance and now publications which deal with the application of model-based predictive techniques to these systems are known [10].

In 1984, Hoffmann published in his dissertation [51] under the name *Adaptive Two-step Control* for the first time an MPC control scheme which considered the special constraints of an actuator with a finite number of discrete switching states already in the approach itself. Thereby, Hoffmann only dealt with a two-step controller. The description of the control strategy shows clearly that it is a typical MPC control scheme:

- The controller itself contains a system model. The model parameters can be adapted online.

9.1 已发表的方法

如在第 5 章所提及, 基于模型的控制方法在控制工程领域已有 30 多年历史, 并已得到实际应用。学者 Clarke [30] 和学者 Morari/Lee [88] 在其发表的论文中均做了简要的历史性概述。尽管如此, 到目前为止所出现的控制方法鲜有隐式地考虑到具有开关特性的执行机构。20 世纪 80 年代, 由亚琛工业大学学者 Hoffmann [51, 52] 和学者 Schmitz [105] 所做的研究工作可以看作是一个例外。直到最近, 有关混合系统的研究才重新得到重视, 而且已有一些涉及基于模型的预测控制方法在混合系统中实际应用的研究成果 [10]。

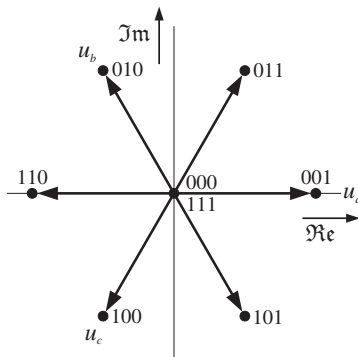
1984 年, 学者 Hoffmann 在其名为自适应砰砰控制的博士论文 [51] 中首次提出了一种模型预测控制方法, 它已经考虑到具有有限离散数量的开关状态的执行机构这一特殊边界条件。学者 Hoffmann 所研究的是砰砰控制器, 其控制策略的描述清晰地表明它是一种典型的模型预测控制方法:

- 控制器本身包含被控系统模型, 而且模型参数可以在线更新;

- Über einen Prädiktor wird das zukünftige Streckenverhalten für alle möglichen Steuergrößenfolgen vorausgerechnet und über eine Kostenfunktion bewertet. Die Folge mit dem besten Ergebnis wird ausgewählt und der *erste* Wert dieser Folge auf die Regelstrecke ausgegeben.
- Die Vorausberechnung erfolgt bis zu einem vorherbestimmten Kosten- und Regelhorizont.

Wie obige Punkte beweisen, handelt es sich bei der Adaptiven Zweipunktregelung eindeutig um einen Sonderfall von Direct Model Predictive Control.

Die Hoffmann'sche Zweipunktregelung wurde von Schmitz weiterentwickelt und als *Adaptive Prädiktive Dreipunktregelung* veröffentlicht [105]. Neben der Erweiterung des Verfahrens auf eine Dreipunktregelung hat Schmitz noch zusätzliche Verbesserungen im Detail vorgenommen, unter anderem im Bereich der Suche nach der optimalen Stellgrößenfolge. Hier wurde besonderes Augenmerk auf die Reduzierung der notwendigen Rechenzeit gelegt, ein Aspekt, der auch heute noch entscheidend für die Realisierbarkeit einer DMPC-Regelung sein kann.



9.3: Mögliche Schaltzustände eines Zweistufenumrichters;
Possible switching states of a two-level inverter;
一个两电平逆变器可能的开关状态

9.2 Umrichterbetrieb mit DMPC/Inverter operation with DMPC/基于 DMPC 的逆变器运行

Als Stellglied für den Stromregelkreis kommt in der Antriebstechnik in den weitaus meisten Fällen ein Zweipunktumrichter mit Spannungszwischenkreis zum Einsatz. Wie allgemein bekannt ist, hat ein solcher Umrichter acht verschiedene Schaltzustände, mit denen sich sieben unterschiedliche Spannungsraumzeiger direkt erzeugen lassen.

- The future behavior of the system is precalculated for all possible sequences of control values using a predictor and it is evaluated via a cost function. The sequence of control values with the best result is selected and the *first* value of this sequence is applied to the controlled system.
- The advance calculation will be done up to a predetermined cost and control horizon.

As the above points prove, Adaptive Two-step Control is definitely a special case of Direct Model Predictive Control.

The two-step control by Hoffmann was further developed by Schmitz and published as *Adaptive Predictive Three-step Control* [105]. Besides the extension of the method to a three-step control, Schmitz has carried out additional improvements in detail, among other things, with regard to the search for an optimum sequence of control values. Special attention was given to the reduction of the necessary computation time, an aspect which can, even today, be a decisive factor for the feasibility of a DMPC control.

9.2 Inverter operation with DMPC

In drive technology, in most cases, a two-level inverter fed from a DC link is used as an actuator for closed-loop current control. As it is commonly known, such an inverter has eight discrete switching states which can directly generate seven different voltage vectors.

- 预测器提前计算出对于所有可能的控制变量序列所引起的被控系统的未来响应, 并使用一个成本函数进行评价, 从而选择出具有最优控制效果的控制变量序列, 该序列的首项作用于被控系统;
- 预测是在事先确定的成本和控制步长范围内完成的。

以上三点表明, 自适应砰砰控制实际上可以看作直接模型预测控制的一种特殊情况。

学者 Hoffmann 的两位式砰砰控制经由学者 Schmitz 进一步改进, 发表出了题为自适应三位式砰砰控制的研究论文 [105]。除了将两位式控制方法扩展到三位式之外, 学者 Schmitz 还在具体细节上做出了改进, 例如关于如何寻找最优控制变量序列这一方面。最引人关注的是如何减少必需的计算时间, 即使是现在, 这一主题仍然对于直接模型预测控制的实现至关重要。

9.2 DMPC 下的逆变器运行

在驱动控制领域, 绝大多数情况下均使用带有中间直流电压的两电平逆变器作为电流控制闭环的执行机构。通常来说, 两电平逆变器包含 8 中不同的开关状态, 它们对应 7 种不同的电压空间矢量。

Die möglichen Spannungswerte lassen sich mit den in Kap. 2 auf Seite 9 gegebenen Formeln (2.1) bis (2.4) bestimmen. Abb. 9.3 zeigt die Ergebnisse in Raumzeigerdarstellung; die dazugehörige Tabelle 9.1 stellt die entsprechenden Werte in tabellarischer Form dar. Selbstverständlich kann das DMPC-Verfahren auf beliebige Umrichter mit einer größeren Anzahl von Stufen und Strom- oder Spannungszwischenkreis angewendet werden, lediglich Abb. 9.3 und Tabelle 9.1 müssen dann entsprechend angepasst werden. Dabei ist jedoch zu beachten, dass bei mehrstufigen Umrichtern die Anzahl der möglichen Spannungsraumzeiger und damit auch die Anzahl der vorauszuberechnenden möglichen Schaltfolgen exponentiell ansteigt.

Vektor- nummer	Schaltzustände			Spannungen	
	Brücke c	Brücke b	Brücke a	$u_{s\alpha}$	$u_{s\beta}$
0	0	0	0	0	0
1	0	0	1	$\frac{2}{\sqrt{3}}$	0
2	0	1	0	$-\frac{1}{\sqrt{3}}$	1
3	0	1	1	$\frac{1}{\sqrt{3}}$	1
4	1	0	0	$-\frac{1}{\sqrt{3}}$	-1
5	1	0	1	$\frac{1}{\sqrt{3}}$	-1
6	1	1	0	$-\frac{2}{\sqrt{3}}$	0
7	1	1	1	0	0

9.1: Mögliche Schaltzustände eines Zweistufenumrichters;

Possible switching states of a two-level inverter;

一个两电平逆变器可能的开关状态

Da sich die Schaltzustände von einem rotierenden Koordinatensystem aus gesehen ständig verändern würden, realisiert man einen Direktregler besser in ortsfesten (Stator) Koordinaten. Dadurch werden sich die Sollwerte zwar zeitlich sinusförmig ändern, bei einem MPC-Regler führt dies jedoch nicht zu einem Schleppfehler, sofern die Sinusform des Sollwerts vorausberechnet und dem Reglereingang mitgeteilt werden kann. Zwar ist dazu ein gewisser, zusätzlicher Rechenaufwand notwendig, die Transformation der jeweils möglichen Schaltzustände des Umrichters in rotierende Koordinaten und ihre mathematische Vorausberechnung wäre jedoch weitaus rechenaufwändiger. Darüberhinaus kann bei knapper Rechenleistung die Prädiktion der Sinusform des Stromsollwerts auf Kosten eines Schleppfehlers einfach entfallen, d. h. es wird über den gesamten Prädiktionshorizont N_p ein konstanter Sollwert angenommen. Würde man den DMPC-Regler in Feldkoordinaten implementieren, wäre eine Prädiktion der zukünftigen Lage der möglichen Umrichterschaltzustände jedoch *unbedingt notwendig*, da sonst keine Regelung möglich wäre.

The possible voltage values can be calculated with the equations (2.1) to (2.4) given in Chapter 2 on page 9. Figure 9.3 shows the results in space vector representation; the corresponding values are depicted in Table 9.1. Of course, the DMPC method can be applied to any kind of inverter with a higher number of switching levels and of voltage or current source type. Then, however, Figure 9.3 and Table 9.1 must be modified accordingly. It has to be pointed out that the number of possible voltage space vectors and thus, the number of possible switching sequences to be precalculated will rise exponentially when multilevel inverters are considered.

Since the inverter switching states seen from a rotating coordinate frame will be constantly changing, it is better to realize a direct controller in a stationary (stator) coordinate system. Thus, the reference values will change sinusoidally with time, however, with an MPC controller this does not lead to a contouring error as long as the sine form of the reference value can be precalculated and then be forwarded to the controller input. Although there is some additional computation time necessary, this is less time consuming than the transformation of all the possible switching states of the inverter into the rotating coordinate frame and their following forecast. Furthermore, predicting the sine form of the reference current value can simply be omitted at the expense of a contouring error, i.e. for the complete prediction horizon N_p , it can be assumed that the reference value is constant. Implementing the DMPC controller in field coordinates, on the other hand, would mean that a prediction of the future location of the possible inverter switching states would be *absolutely necessary*, since otherwise no control would be possible.

可能的电压值可根据第 2 章第 9 页所给出的方程 (2.1) 至 (2.4) 进行计算。图 9.3 给出了所求得结果的空间矢量描述形式，表 9.1 以表格方式给出了上述计算结果。当然，直接模型预测控制方法也可应用于含有更高电平数任意类型逆变器，例如电流型逆变器或电压型逆变器等。与此同时，图 9.3 和表 9.1 中的相关信息需要进行相应调整。要特别注意的是，在多电平逆变器情形下，可能产生的电压空间矢量数量以及预测得到的可能开关序列将以指数形式增加。

由于开关状态在旋转坐标系视角下是不断变化的，因此最好在固定定子坐标系下实现直接控制器。虽说参比变量随着时间以正弦波形发生变化，但在使用 MPC 控制器的情况下，并不会引起跟踪误差，前提是预测得到参比变量的正弦波形并将其作为控制器输入。尽管上述方式会引起额外的计算量，但仍小于将逆变器可能的开关状态变换至旋转坐标系以及随后的预测所需的全部计算量。此外，在计算能力裕量较小时，可以直接省去参考电流正弦波形的预测。换言之，假设参考电流在预测步长 N_p 范围内保持不变，由此所节省的计算量是以跟踪误差为代价。假使在磁场坐标系下实现直接模型预测控制器，那么对逆变器可能开关状态的将来位置预测是绝对必要的，否则无法实现控制。

9.2.1 Berücksichtigung der Bootstrap-Schaltung/Consideration of the Bootstrap circuit/考虑自举电路

Preiswerte Umrichter besitzen für die Energieversorgung der Treiberbausteine des IGBT-Moduls oft eine sogenannte *Bootstrap-Schaltung*. Bei dieser Schaltungsart ist die Versorgung der Treiber der „oberen“ Halbbrückenventile mit speziellen DC/DC-Wandlern nicht erforderlich; leider ergibt sich dabei der gravierende Nachteil, dass die „oberen“ Treiber nur mit Energie versorgt werden, wenn das „untere“ Halbbrückenventil eingeschaltet ist. Man behilft sich damit, dass man die entsprechenden Treiberbaugruppen mit einem Speicherkondensator versieht, der die Energieversorgung des entsprechenden Treibers während der Einschaltzeit des „oberen“ IGBTs sicherstellt. Da dieser Speicherkondensator aber immer nur dann nachgeladen wird, wenn das „untere“ Ventil eingeschaltet ist, ergibt sich, dass das Tastverhältnis nicht beliebig gewählt werden kann. Bei Verwendung eines Modulationsverfahrens, z. B. Raumzeigermodulation, kann man durch Beschränken des Modulationsgrads auf z. B. $m_{max} = 0,95$ sicherstellen, dass das „untere“ Halbleiterventil genügend oft eingeschaltet wird, um die Versorgung des Treibers des „oberen“ Ventils sicherzustellen. Ein direktes Umrichtersteuerverfahren, das prinzipbedingt keinen Modulator benötigt, besitzt diese Möglichkeit nicht. Trotzdem muss auch hier sichergestellt werden, dass nach einer bestimmten Einschaltdauer des „oberen“ Halbbrückenventils für einen Abtastzyklus das „untere“ Ventil geschaltet wird, um den Stützkondensator der Spannungsversorgung des „oberen“ IGBT-Treibers nachzuladen.

Klassische Direktsteuerverfahren, z. B. Direct Torque Control (DTC) [1, 113] oder Direct Self Control (DSC) [33], bieten keine Möglichkeit, dem Regler mitzuteilen, welche Spannungsvektoren er bedingt durch die oben dargelegte Einschränkung durch die Bootstrap Schaltung momentan nicht schalten darf. Im Allgemeinen wird man in solchen Fällen in der betroffenen Halbbrücke einfach zwangsweise das „untere“ Ventil schalten und so den von der Regelung vorausberechneten, optimalen Spannungsvektor zumindest teilweise ignorieren. Dadurch ergibt sich aber eine nicht-optimale Regelung, da durch die *nachträgliche* Korrektur des Optimalvektors nicht sichergestellt werden kann, dass aus den verbleibenden, erlaubten Schaltzuständen der bestmögliche ausgewählt worden ist.

9.2.1 Consideration of the Bootstrap circuit

Low-cost inverters often have a so-called *bootstrap circuit* as power supply for the driver components of the IGBT module. When such a circuit is used, a power supply for the driver circuits of the “upper” half-bridge IGBT switches with special DC/DC converters is not required; unfortunately, this leads to the serious drawback that the “upper” IGBT drivers are only supplied with power if the “lower” half-bridge IGBT switches are turned on. This fact can be handled when the corresponding driver modules are equipped with a

storage capacitor which backs up the power supply while the “upper” IGBTs are turned on. Since this storage capacitor is only recharged when the “lower” devices are switched on, the duty cycles cannot be arbitrarily chosen. By using a modulation technique, e. g. space vector modulation, it can be made sure that the “lower” semiconductor devices are switched on often enough to guarantee power supply for the driver of the “upper” switches by limiting the modulation index to e. g. $m_{max} = 0.95$. A direct inverter control method without the need for a modulator does not have this possibility. Nevertheless, in this case it also has to be made sure that after a certain on time of the “upper” device, the “lower” device will be switched on for at least one sampling cycle in order to recharge the capacitor for the power supply of the “upper” IGBT drivers.

Classical direct control methods, e. g. Direct Torque Control (DTC) [1, 113] or Direct Self Control (DSC) [33], offer no such possibility to pass the information to the controller which voltage vectors should not be switched due to the aforementioned constraints of the bootstrap circuit. In these cases, the easiest method is simply to switch on the “lower” device of the related half bridge compulsorily and thus, to ignore the optimum voltage vector precalculated by the control algorithm at least partially. However, this leads to a non-optimum control since by the *subsequent* correction of the optimum vector it cannot be guaranteed that the best possible switching state has been selected from the remaining allowed ones.

9.2.1 考虑自举电路

低成本的逆变器通常使用所谓的自举电路，它作为 IGBT 模块的驱动级电源，如此便无必要使用特殊的 DC-DC 变换器为半桥上部的 IGBT 驱动级提供能量。遗憾的是，上述方式存在一个严重的缺点，因为它总是需要首先导通半桥下部的 IGBT 才能够满足半桥上部的 IGBT 驱动级所需的能量。通常人们使用带有自举电路的驱动级用以确保半桥上部的 IGBT 导通时间段内所需的能量。由于自举电路只有在半桥下部 IGBT 导通时才能够被充电，从而占空比无法任意选择。当使用调制技术时，例如空间矢量调制，可以通过限制调制比范围以确保半桥下部 IGBT 具有足够长的导通时间，从而进一步确保半桥上部 IGBT 驱动级的能量供应，例如 $m_{max} = 0.95$ 。从原理上看，无需使用调制器的直接逆变器控制方法不能够采取上述措施。尽管如此，仍需保证在一个采样周期内半桥上部 IGBT 导通确定的时间后再导通半桥下部的 IGBT，目的是为半桥上部 IGBT 驱动级的电压源进行充电。

经典的控制方法无法告知控制器当前哪些电压矢量由于自举电路的限制而绝对不允许使用，如直接转矩控制 [1, 113] 或直接自控制 [33] 等。在上述情况下，人们通常简单地直接令半桥下部的 IGBT 强制导通，从而部分地忽略了控制器所预测的最优电压矢量。上述处理方式直接导致了系统的非最优控制，因为随后对最优矢量的修正无法确保可以从剩余的所允许的开关状态中选到最优矢量。

Modellbasierte Regelstrategien (MPC-Regler) sind prädestiniert für Anwendungsfälle, in denen eine Beschränkung durch das Stellglied oder die Regelstrecke bereits im Regler berücksichtigt werden soll. Entsprechend ist es beim DMPC-Verfahren möglich, dem Regler die Beschränkung durch die Bootstrap-Schaltung *vorher* mitzuteilen. Terno [115, S. 28] nennt dies die Definition einer *unzulässigen Familie*, d. h. einer Teilmenge von möglichen Lösungen des Optimierungsproblems, die wegen irgendwelcher Beschränkungen nicht verwendet werden dürfen und daher als Optimallösungen nicht zur Verfügung stehen. Der Regelalgorithmus weiß also, wann er welche Spannungsvektoren nicht schalten darf, und *berücksichtigt dies um den Regelhorizont im Voraus*. Dadurch ist unter allen Bedingungen sichergestellt, dass der Regler unter den verbleibenden Schaltmöglichkeiten stets die optimale Schaltfolge findet. DMPC-Verfahren sind folglich *jedem* anderen Direktsteuerverfahren überlegen.

9.3 Modellbildung/Model formulation/模型建立

Wegen der nahen Verwandtschaft mit herkömmlichen MPC-Reglern kann die Modellbildung nach denselben Grundsätzen erfolgen; auch Hoffmann verwendet in [51] einen Ansatz ähnlich einem CARIMA-Modell. Bei der Herleitung des Maschinenmodells ist jedoch zu beachten, dass ein DMPC-Regler – wie in Kap. 9.2 ausgeführt – besser in einem ortsfesten (Stator-) Koordinatensystem realisiert werden sollte.

Grundlage für alle Modelle zur Regelung einer Asynchronmaschine ist stets das komplexe Maschinenmodell aus Kap. 2, das durch die Gln. (2.9) und (2.10) beschrieben wird (siehe auch Abb. 2.2 auf Seite 13). Wählt man nun ein stationäres Koordinatensystem, d. h. $\omega_k = 0$, und zerlegt die Gln. (2.9) und (2.10) in Real- und Imaginärteil, so erhält man folgendes Gleichungssystem:

$$i_{s\alpha} + \tau_\sigma \frac{di_{s\alpha}}{d\tau} = \frac{k_r}{r_\sigma \tau_r} \psi_{r\alpha} + \frac{k_r}{r_\sigma} \omega \psi_{r\beta} + \frac{1}{r_\sigma} u_{s\alpha} \quad (9.1)$$

$$i_{s\beta} + \tau_\sigma \frac{di_{s\beta}}{d\tau} = \frac{k_r}{r_\sigma \tau_r} \psi_{r\beta} - \frac{k_r}{r_\sigma} \omega \psi_{r\alpha} + \frac{1}{r_\sigma} u_{s\beta} \quad (9.2)$$

$$\psi_{r\alpha} + \tau_r \frac{d\psi_{r\alpha}}{d\tau} = -\omega \tau_r \psi_{r\beta} + l_h i_{s\alpha} \quad (9.3)$$

$$\psi_{r\beta} + \tau_r \frac{d\psi_{r\beta}}{d\tau} = \omega \tau_r \psi_{r\alpha} + l_h i_{s\beta} \quad (9.4)$$

Zur Verdeutlichung der Zusammenhänge zwischen $i_{s\alpha}$, $i_{s\beta}$, $\psi_{r\alpha}$ und $\psi_{r\beta}$ zeigt Abb. 9.4 den Signalfussplan einer Asynchronmaschine in stationären Koordinaten.

Model-based control strategies (MPC controllers) are predestined for applications in which constraints of the actuator or of the plant have to be considered already in the controller design. Accordingly, when using DMPC methods, it is possible to pass the

information about the constraint resulting from the bootstrap circuit to the controller already *beforehand*. Terno [115, p. 28] defines this as a *prohibited family*, i. e. a subset of possible solutions of the optimization problem which may not be used because of any constraints and hence are not available as an optimum solution. The knowledge at which time some voltage vectors may not be switched can be included in the design of the control algorithm and hence, the controller *will consider this for the whole control horizon in advance*. Because of this, it is made sure that the controller will find the optimum switching sequence among the remaining possibilities under all conditions. Therefore, DMPC techniques are superior to *any* other direct control method.

9.3 Model formulation

Because of the close relationship with conventional MPC, modeling can be done following the same principles. In [51] Hoffmann also uses an approach similar to a CARIMA model. However, considering the derivation of the machine model one has to take care that a DMPC controller—as mentioned in Chapter 9.2—is better realized in a stationary (stator) coordinate system.

Basic for all models used to control an induction machine is always the complex machine model as stated in Chapter 2, which can be described by equations (2.9) and (2.10) (see also Figure 2.2 on page 13). If a stationary coordinate frame is selected, i. e. $\omega_k = 0$, and the equations (2.9) and (2.10) are splitted into real and imaginary part, the following system of equations can be obtained, see equations (9.1) to (9.4).

In order to clarify the relations between $i_{s\alpha}$, $i_{s\beta}$, $\psi_{r\alpha}$ and $\psi_{r\beta}$, Figure 9.4 shows the signal flow graph of an induction machine in stationary coordinates.

基于模型的控制策略注定适用于执行机构或被控系统的限制已经被考虑入控制器的应用情形。因此，在直接模型预测控制方法中，可以提前告知控制器由于自举电路所引起的限制。学者 Terno 将其定义为可能性家族 [115, 第 28 页]。换言之，优化问题部分可能的解由于某些限制而无法使用，因此无法提供一个最优解。控制算法已清楚哪些电压矢量在何时不允许被切换，并在控制步长范围内提前考虑到这一点。如此，便可确保控制器在任何条件下始终可以在剩余的开关状态中选择最优的开关序列。可以认为，直接模型预测控制方法要优于其他的直接控制方法。

9.3 模型建立

由于与传统的 MPC 控制器存在紧密关系，从而可以基于完全相同的原则来建立模型。在文献 [51] 中，学者 Hoffmann 使用了一个与 CARIMA 模型类似的方法。然而，推导电机模型时需要特别注意的是，直接模型预测控制器最好能在一个静止的定子坐标系下来实现，如章节 9.2 所述。

用于控制异步电机的所有模型的基础始终离不开第 2 章所述的复数电机模型，此模型由方程 (2.9) 和 (2.10) 所表示。如果选择一个静止坐标系，此时 $\omega_k = 0$ ，然后将方程 (2.9) 和 (2.10) 分解为实部和虚部，从而可以得到如下方程系统，见方程 (9.1) 至 (9.4)。

为了更为清晰地表达 $i_{s\alpha}$ ， $i_{s\beta}$ ， $\psi_{r\alpha}$ 和 $\psi_{r\beta}$ 之间的关系，图 9.4 给出了在静止坐标系下的一种异步电机信号流程图。

In the same way as with continuous signal MPC controllers, the control algorithm can be developed as a single or MIMO controller; consideration of disturbances and filtering of noisy measurement signals can be implemented, too. Since, in these cases, the approach is identical to the ones described in chapters 6 and 8, only one of the many possible combinations is discussed here: A MIMO-DMPC algorithm is introduced in which the cross coupling between $\psi_{r\alpha}$ and $\psi_{r\beta}$ as well as the impact of $i_{s\alpha}$ and $i_{s\beta}$ caused by the rotor flux (Figure 9.4) is neglected due to simplicity.

As the signal flow graph in Figure 9.4 shows, the stator voltages $u_{s\alpha}$ and $u_{s\beta}$ are the input variables of the plant and thus, they are also the actuating variables. A MIMO-DMPC controller for current control of an induction machine with a DC link inverter, using this model with $u_{s\alpha}$ and $u_{s\beta}$ as plant inputs, has to optimize two actuating variables, which can, according to Table 9.1, take only a few discrete values. In addition to that, $u_{s\alpha}$ and $u_{s\beta}$ are not independent from each other because they depend on the switching states of the inverter half bridges. Furthermore, the change of the actuating values $\Delta \mathbf{u}$ does in this case *not* express the real effort for changing the actuating variables, because the effort is not dependent on the change of the voltage level of $u_{s\alpha}$ or $u_{s\beta}$, but only on the switching operations in the inverter itself.

与连续 MPC 控制器一样，控制过程可以设计为单一控制器或 MIMO 控制器，其中，扰动变量的考虑和含噪声的测量信号滤波也全部可以整合到控制器结构中。由于上述情况下的处理方式几乎与第 6 章和第 8 章已讲述的相关部分完全一致，所以接下来仅讨论诸多组合方式中的一种，也即 MIMO 直接模型预测控制。为简化起见， $\psi_{r\alpha}$ 和 $\psi_{r\beta}$ 之间的耦合以及转子磁通对 $i_{s\alpha}$ 和 $i_{s\beta}$ 的影响均忽略不计，如图 9.4 所示。

如图 9.4 中的信号流图所示，定子电压 $u_{s\alpha}$ 和 $u_{s\beta}$ 代表被控系统的输入量，也即所谓的控制变量。当选用 MIMO-直接模型预测控制器结合电压源型两电平逆变器来实现异步电机的电流控制时， $u_{s\alpha}$ 和 $u_{s\beta}$ 作为被控系统输入量，则必须根据表 9.1 所示的不同开关状态来优化上述两个控制变量。另外， $u_{s\alpha}$ 和 $u_{s\beta}$ 并非完全相互独立，它们取决于逆变器半桥的开关状态。此外，上述方式下的控制增量 $\Delta \mathbf{u}$ 也无法反映真实的控制消耗，因为控制消耗与 $u_{s\alpha}$ 或 $u_{s\beta}$ 的变化程度无关，它所表示的仅仅是逆变器本身的切换过程。

Sinnvoll ist daher, statt der komplexen Statorspannungen die Schaltzustände der drei Halbbrücken als Stellgrößen zu verwenden. Selbstverständlich muss in diesem Fall das Streckenmodell um die Umrechnung von den Halbbrückenzuständen auf die Maschinenspannungen $u_{s\alpha}$ und $u_{s\beta}$ ergänzt werden. Die notwendigen Umrechnungsfaktoren können Tabelle 9.1 entnommen werden. Eine solche Vorgehensweise bietet mehrere Vorteile:

- Die Kostenfunktion bewertet als Stellgrößenänderung den tatsächlichen Stellaufwand, d. h. die Schaltvorgänge im Umrichter. Dies bedeutet, dass lediglich durch Wahl des entsprechenden Bewertungsfaktors die mittlere Schaltfrequenz des DMPC-geregelten Umrichters gesenkt oder angehoben werden kann. Der Regler versucht dann, durch Schalten anderer Brücken ein in der Gesamtbewertung, bestehend aus Regelabweichung und Schalthäufigkeit, optimales Ergebnis zu erzielen.
- Die Stellgrößen entsprechen direkt den Zuständen der einzelnen Halbbrücken. Daher werden direkt die Schaltzustände des Umrichters optimiert. Die Ergebnisse können ohne weitere Umrechnung an den Umrichter ausgegeben werden; eine zusätzliche mathematische Umformung ist nicht notwendig.
- Ein Zweipunktumrichter mit Spannungszwischenkreis erlaubt acht verschiedene Schaltzustände, mit denen jedoch nur sieben unterschiedliche Spannungsraumzeiger erzeugt werden können. Die Wahl der Schaltzustände als Stellgröße ermöglicht es, alle acht Schaltmöglichkeiten des Umrichters zu berücksichtigen. Abhängig von der bezüglich der Schalthäufigkeit besseren Alternative wird dann der Nullzeiger entweder durch die Schalterstellung 000 oder 111 realisiert. Wählt man hingegen die Statorspannungen $u_{s\alpha}$ und $u_{s\beta}$ als Stellgrößen, so geht dieser Freiheitsgrad verloren.
- Jede Halbbrücke besitzt bei einem Umrichter mit Spannungszwischenkreis nur zwei mögliche Zustände. Folglich können auch die zu optimierenden drei Stellgrößen nur je zwei verschiedene Zustände annehmen, sind also mathematisch als boolesche Variable zu betrachten. Es handelt sich bei dem Optimierungsproblem daher um eine sogenannte boolesche Optimierung. Für diese Art von Optimierungsaufgaben existieren eine Reihe von Vereinfachungen, die zu effizienten Optimieralgorithmen führen.

Because of this, it is reasonable to use the switching states of the three half bridges as actuating variables instead of the complex stator voltages. As a matter of course, in this case, the conversion from the half bridge states to the machine voltages $u_{s\alpha}$ and $u_{s\beta}$ has to be added to the plant model. The necessary conversion factors can be taken from Table 9.1. Such an approach has several advantages:

- The cost function evaluates the real switching effort, i. e. the switchings that have to be executed in the inverter. This means that only by selecting the appropriate

weighting factor, the average switching frequency of the DMPC controlled inverter can be lowered or raised. The controller then tries to achieve an optimal result for the whole cost function, consisting of control deviation and switching operations, by switching other bridges.

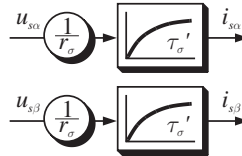
- The actuating variables directly correspond to the states of the single half bridges. Hence, the inverter states are directly optimized. The results can be forwarded to the inverter without any further conversion; an additional mathematical transformation is not necessary.
- A two-level inverter with DC link permits eight different switching states. These can, however, generate only seven different space voltage vectors. Choosing the inverter states as actuating variables, it is possible to consider all eight switching possibilities of the inverter. Dependent on the better alternative concerning the switching frequency, the zero voltage space vector is created either by the switching state 000 or 111. On the contrary, if the stator voltages $u_{s\alpha}$ and $u_{s\beta}$ are selected as actuating variables, this degree of freedom is lost.
- Considering an inverter with DC link, every half bridge has only two possible states. Thus, each of the three actuating variables can take only two different states. Hence, the optimization problem is a so called boolean optimization. For this kind of optimization, there exists a wide range of simplifications which lead to efficient optimization algorithms.

基于上述分析，更为合理的方式是使用三个半桥的开关状态取代定子电压作为控制变量。显然，需要添加半桥开关状态与电机电压 $u_{s\alpha}$ 和 $u_{s\beta}$ 之间的对应关系至被控系统模型，必要的转换系数可根据表 9.1 来获取。上述处理方式具有以下优点：

- 成本函数以逆变器真实的切换过程作为控制增量进行评估，这也意味着仅通过选择相应的权重系数即可降低或提高由 DMPC 所控制的逆变器的平均开关频率。控制器试图通过切换其他半桥以获得最优效果，其评判标准由控制偏差和开关频率两者共同组成。
- 控制变量直接对应半桥的开关状态。从而，直接优化的也是逆变器开关状态，优化所得结果无需额外数学变换即可直接作用于逆变器。
- 电压源型两电平逆变器允许出现 8 种不同的开关状态，然而仅能产生 7 种不同的电压空间矢量。选择开关状态作为控制量是可以考虑到逆变器的所有 8 种开关状态。此外，零矢量 000 或 111 的选择是基于开关频率的考量。反之，如果选择定子电压 $u_{s\alpha}$ 和 $u_{s\beta}$ 作为控制变量，那么上述自由度将会丧失。
- 电压源型逆变器的每一半桥仅有两种可能的状态，因此待优化的三个控制量也分别包含两种可能性，数学上可以看作布尔变量。从而，所讨论的优化问题可转化为所谓的布尔优化问题。存在一系列的简化方法用于布尔类型的优化，从而可以得到最有效的优化算法。

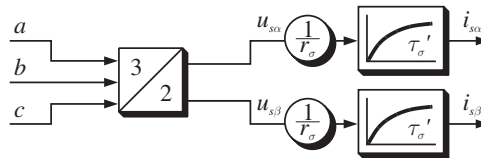
9.3.1 Einfaches Maschinenmodell/Simple machine model/简化电机模型

Ein einfaches Maschinenmodell ohne Berücksichtigung von Kreuzkopplung oder Störgrößen erleichtert den Einstieg in das Verfahren der direkten modellbasierten Prädikativregelung. Zu Beginn wird daher die Maschine gemäß Abb. 9.5 auf zwei einfache, nicht miteinander verknüpfte PT₁-Glieder reduziert.



9.5: Vereinfachtes Modell für die Stromregelung (Statorkoordinaten);
Simplified model for current control (stator coordinates);
用于电流控制的简化模型 (定子坐标系)

Da als Eingangsgrößen, wie oben ausgeführt, nicht die Statorspannungen $u_{s\alpha}$ und $u_{s\beta}$, sondern die Schaltzustände a, b und c der einzelnen Halbbrücken verwendet werden sollen, muss das o. a. Maschinenmodell noch um die entsprechende Umrechnung erweitert werden (Abb. 9.6). Der Umrechnungsblock von a, b, c auf $u_{s\alpha}, u_{s\beta}$ beinhaltet die in Tabelle 9.1 angegebenen Faktoren.



9.6: Vereinfachtes Modell für die Stromregelung (Brückenzustände);
Simplified model for current control (bridge switching states);
用于电流控制的简化模型 (半桥状态)

Setzt man den in Abb. 9.6 gezeigten Signalflussplan in eine Zustandsraumbeschreibung gemäß Formeln (5.2) und (5.3) (s. Seite 61) um, wählt man die Vektoren \mathbf{x} , \mathbf{u} und \mathbf{y} folgendermaßen:

$$\mathbf{x} = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}$$

9.3.1 Simple machine model

A simple machine model without consideration of the cross coupling or of disturbances makes it easier to get familiar with direct model-based predictive control. Thus, the machine model is reduced to two single and independent first order transfer functions as it can be seen in [Figure 9.5](#).

Since, as mentioned above, not the stator voltages $u_{s\alpha}$ and $u_{s\beta}$, but the switching states a , b and c of the individual half bridges should be selected as input variables, the corresponding translation has to be added to the machine model shown above ([Figure 9.6](#)). The translation block from a, b, c to $u_{s\alpha}, u_{s\beta}$ contains the factors given in [Table 9.1](#).

If the signal flow graph shown in [Figure 9.6](#) is translated into state space equations according to equations (5.2) and (5.3) (see page 61), the vectors \mathbf{x} , \mathbf{u} and \mathbf{y} are selected as follows:

$$\mathbf{x} = \begin{bmatrix} \dot{i}_{s\alpha} \\ \dot{i}_{s\beta} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \dot{i}_{s\alpha} \\ \dot{i}_{s\beta} \end{bmatrix}$$

9.3.1 简化电机模型

一个不考虑耦合或扰动的简化电机模型对于理解直接模型预测控制方法大有帮助。首先把电机简化为两个相互独立的一阶滞后元件，如[图 9.5](#)所示。

由于使用了每一半桥的开关状态 a , b 和 c 来代替定子电压 $u_{s\alpha}$ 和 $u_{s\beta}$ 作为控制变量，那么上述简化的电机模型需根据相应的转换关系进一步扩展，如[图 9.6](#)所示。实现转换的模块包含了[表 9.1](#)所给出的系数。

如果将[图 9.6](#)所示的信号流图转化为基于方程 (5.2) 和 (5.3) 的状态空间矢量描述形式 (见第 61 页)，并分别选择矢量 \mathbf{x} , \mathbf{u} 和 \mathbf{y} ：

$$\mathbf{x} = \begin{bmatrix} \dot{i}_{s\alpha} \\ \dot{i}_{s\beta} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \dot{i}_{s\alpha} \\ \dot{i}_{s\beta} \end{bmatrix}$$

Damit lautet die Zustandsraumbeschreibung von Abb. 9.6:

$$\frac{d}{d\tau} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_{\sigma'}} & 0 \\ 0 & -\frac{1}{\tau_{\sigma'}} \end{bmatrix} \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} \frac{2}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & \frac{1}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & -\frac{1}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} \\ 0 & \frac{1}{r_{\sigma}\tau_{\sigma'}} & -\frac{1}{r_{\sigma}\tau_{\sigma'}} \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (9.5)$$

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (9.6)$$

9.4 Implizite Lösung/Implicit solution/隱式求解

Ein einfacher Weg zu einer Direktregelung eines Umrichters ohne zwischengeschalteten Modulator unter Beibehaltung einer Long-range Predictive Control besteht in der entsprechenden Adaption des passenden GPC-Reglers. Mehrdimensionale Regelungen mit und ohne Filterung der Messwerte sowie Berücksichtigung von Störgrößen sind problemlos möglich. Es empfiehlt sich, die gemessenen Statorströme über einen Tiefpass zu filtern, da die Regelung ansonsten wegen der typischen Charakteristik von GPC-Verfahren nur schwer zu beherrschen ist (s. Kap. 6.1.2 auf Seite 93). Daher wird die Zustandsraumbeschreibung aus (9.5) und (9.6) zu:

$$\frac{d}{d\tau} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_{\sigma'}} & 0 \\ 0 & -\frac{1}{\tau_{\sigma'}} \end{bmatrix} \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} \frac{2}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & \frac{1}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & -\frac{1}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} \\ 0 & \frac{1}{r_{\sigma}\tau_{\sigma'}} & -\frac{1}{r_{\sigma}\tau_{\sigma'}} \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (9.7)$$

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ Dummy \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (9.8)$$

Zustands-, Eingangs- und Ausgangsvektor lauten demnach

$$\mathbf{x} = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ Dummy \end{bmatrix} \quad (9.9)$$

Thus, the state space representation of [Figure 9.6](#) is given as:

$$\frac{d}{d\tau} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_{\sigma'}} & 0 \\ 0 & -\frac{1}{\tau_{\sigma'}} \end{bmatrix} \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} \frac{2}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & \frac{1}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & -\frac{1}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} \\ 0 & \frac{1}{r_{\sigma}\tau_{\sigma'}} & -\frac{1}{r_{\sigma}\tau_{\sigma'}} \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (9.5)$$

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (9.6)$$

9.4 Implicit solution

A simple way for designing a direct control method of an inverter without an intermediary modulator under retention of Long Range Predictive Control is to adapt a suitable GPC controller. Multidimensional controllers with and without filtering of the measured values as well as consideration of disturbances are possible without any problems. It is advisable to filter the measured stator currents with a low-pass filter since, otherwise, the control will be very difficult to handle because of the typical characteristics of GPC methods (see [Chapter 6.1.2](#) on page [93](#)). Thus, the state space representation given in the equations [\(9.5\)](#) and [\(9.6\)](#) can be rewritten in the following way, see [equation \(9.7\)](#) and [\(9.8\)](#). According to that, state, input and output vector are [equation \(9.9\)](#).

从而, 可得到[图 9.6](#)所示的状态空间矢量描述形式, 见[方程 \(9.5\)](#) 和 [\(9.6\)](#)。

9.4 隐式求解

一种以简单的方式来实现无需连接调制器且能够同时保证多步长预测控制的逆变器直接控制方法需要对 GPC 控制器进行适当调整。一种可实现包含和不包含测量值滤波以及考虑扰动变量的多维控制也是毫无问题的。但是, 仍然建议使用低通滤波器对所测量的定子电流进行滤波, 否则由于 GPC 方法的典型特征而使得控制器实现起来非常困难, 见[章节 6.1.2](#) 第 [93](#) 页。因此, 根据[方程 \(9.5\)](#) 和 [\(9.6\)](#) 得到的新的状态空间描述形式为[方程 \(9.7\)](#) 和 [\(9.8\)](#)。此外, 状态向量、输入向量和输出向量分别为[方程 \(9.9\)](#)。

Die „Dummy“-Variable im Ausgangsvektor \mathbf{y} ist für die interne Filterung von GPC notwendig. Da eine GPC-Regelung ohne interne Filterung durch ein $\mathbf{T}(z^{-1})$ -Design-Polynom – wie bereits ausgeführt – nicht sinnvoll betreibbar ist, muss eine, wenn auch leichte, Filterung mittels $\mathbf{T}(z^{-1})$ integriert werden. Da sowohl die Stellgrößen $\Delta\mathbf{u}(t)$ als auch die Regelgrößen $\mathbf{y}(t)$ mit dem gleichen Filterpolynom gefiltert werden, müssen sie beide die gleiche Dimension aufweisen. Da $\mathbf{u}(t)$ die Dimension 3 besitzt, muss bei $\mathbf{y}(t)$ notwendigerweise ein „Dummy“-Wert hinzugefügt werden, um die gleiche Dimension zu erhalten und auch $\mathbf{y}(t)$ mit dem gleichen Filter filtern zu können.

Auch bei dem hier vorgestellten Verfahren wird aus der über den j -Schritt-voraus-Prädiktor berechneten freien und erzwungenen Antwort sowie dem zukünftigen Verlauf der Führungsgröße mit Hilfe einer Kostenfunktion eine optimale Stellgrößenfolge ermittelt. Hierin unterscheidet sich die direkte Regelung nicht von der konventionellen MPC-Regelung. Daher kann auch die gleiche quadratische Kostenfunktion wie in Kap. 6.1.1 auf Seite 79 in Formel (6.15) angegeben angewendet werden. Wie Kap. 8.1.3 auf Seite 175 aufzeigt, gilt diese Kostenfunktion sowohl für den Eingrößen- als auch für den Mehrgrößenregler (Formel (8.24)). Da die Direktregelung im vorliegenden Fall als MIMO-Regelung ausgeführt werden soll, wird im Folgenden nur noch der mehrdimensionale Fall behandelt. Die für ein SISO-Verfahren geltenden Formeln und Aussagen können leicht hergeleitet werden.

Ein analytisches Minimieren des Kostenfunctionals J ergibt nun eine Gl. (8.27) für optimale, zukünftige Stellgrößen, wobei die Stellgröße $\mathbf{u}(t)$ jedoch jeden beliebigen Wert annehmen kann. Diese Gleichung bezeichnet man als *Einschritt-Optimierung*. Bei direkter Ansteuerung des Umrichters ohne zwischengeschalteten Modulator hingegen ist eine Einschritt-Optimierung nicht möglich, da die Stellgröße nur einige wenige diskrete Werte annehmen kann. Mathematisch bedeutet dies, dass die Stellgröße $\mathbf{u}(t)$ ein Element der Menge der möglichen Stellgrößen \mathcal{S} sein muss, die ihrerseits eine Untermenge der natürlichen Zahlen \mathbb{N} ist, d. h. $\mathbf{u}(t) \in \mathcal{S} \subseteq \mathbb{N}$.

The “dummy” variable in the output vector \mathbf{y} is necessary for the internal filtering of GPC. Since it is not reasonable to use a GPC controller without internal filtering via a $\mathbf{T}(z^{-1})$ design polynomial—as already mentioned—, at least a smooth filtering via $\mathbf{T}(z^{-1})$ has to be done. Since the actuating variables $\Delta\mathbf{u}(t)$ as well as the controlled variables $\mathbf{y}(t)$ have to be filtered with the same polynomial, both vectors need to have the same dimension. Since $\mathbf{u}(t)$ has the dimension three, a “dummy” value has to be added to $\mathbf{y}(t)$ in order to get the same dimension and to make use of an identical filter for $\mathbf{y}(t)$.

In the method presented here from the free and the forced response calculated though the j -step ahead predictor and from future values of the reference variable, an optimal sequence of values for the actuating variables is determined by using a cost function. In this case, direct control does not differ from conventional MPC. Hence, the same

quadratic cost function as denoted in Chapter 6.1.1 on page 79 in equation (6.15) can be used. As proven in Chapter 8.1.3 on page 175 this cost function is applicable for a unidimensional as well as for a multidimensional controller (equation (8.24)). Since the direct controller, in this case, should be a MIMO controller, in the following, only the multidimensional case will be dealt with. Equations and statements valid for SISO methods can easily be derived.

Analytical minimization of the cost function J leads to equation (8.27) for optimal future values of the actuating variables. However, in this case the actuating variable $\mathbf{u}(t)$ can take any value. This equation is called *single-step optimization*. If the inverter is controlled directly without an intermediary modulator a single-step optimization is not possible, because the actuating variable can take only a few discrete values. Seen from the view of mathematics, this means that the actuating variable $\mathbf{u}(t)$ has to be an element of the set of possible actuating variables \mathcal{S} , which itself is a subset of the natural numbers \mathbb{N} , i. e. $\mathbf{u}(t) \in \mathcal{S} \subseteq \mathbb{N}$.

输出向量 \mathbf{y} 中的“dummy”变量对于 GPC 的内部滤波来说是非常必要的。未经设计多项式 $\mathbf{T}(z^{-1})$ 进行内部滤波的 GPC 控制是无法正常运行的。因此，有必要将实现滤波的设计多项式 $\mathbf{T}(z^{-1})$ 集成到控制器。由于控制变量 $\Delta\mathbf{u}(t)$ 和被控变量 $\mathbf{y}(t)$ 均采用相同的滤波多项式进行滤波，所以二者必须具有相同的维度。控制变量 $\mathbf{u}(t)$ 的维度为 3，所以为了使被控变量 $\mathbf{y}(t)$ 也具有相同的维度，才在 $\mathbf{y}(t)$ 中额外添加了一个“dummy”变量，以便能够使用相同的滤波器滤波。

同样，此处介绍的方法也是通过基于 J 步前向预测器预测出的自由响应，强制响应以及参比变量的未来变化趋势的成本函数来求得最优的控制变量序列。此时的直接控制与传统的模型预测控制并无差别，而且也同样可以使用章节 6.1.1 第 79 页方程所示的二次成本函数 (6.15)。如章节 8.1.3 第 175 页所示，二次成本函数 (8.24) 不仅可用于单变量控制器，同样也可用于多变量控制器。鉴于本章以 MIMO 为例介绍直接控制，所以接下来仅讨论多维的情况。当然，适用于 SISO 方法的方程和相关内容是可以很容易推导出来的。

通过成本函数 J 的解析形式的最小化可得到表示最优的未来时刻的控制变量方程 (8.27)，其中的控制变量 $\mathbf{u}(t)$ 可以任意取值，上述方程也被称为单步优化。在不包含中间调制器的逆变器直接控制情况下，单步优化是行不通的，因为控制变量仅可取一些离散的值。从数学角度来看，这意味着控制变量 $\mathbf{u}(t)$ 必须是所有可能的控制量集合 \mathcal{S} 中的一个元素，集合 \mathcal{S} 也是自然数 \mathbb{N} 的一个子集，也即 $\mathbf{u}(t) \in \mathcal{S} \subseteq \mathbb{N}$ 。

Es ist daher notwendig, das Optimierungsproblem durch eine *Mehrschritt-Optimierung* zu lösen. Leider kann eine Mehrschritt-Optimierung im Gegensatz zur Einschritt-Optimierung wegen der durch die diskreten Stellgrößen hervorgerufenen Nichtlinearität nicht mehr analytisch durchgeführt werden. Es muss daher eine Entscheidungs- und Suchstrategie angewendet werden, die die bestmögliche Stellgrößenfolge ermittelt, d. h. der kleinste Wert der zusammengefassten Bewertungsfunktion (8.26) unter Berücksichtigung der besonderen Bedingungen für \mathbf{U} muss gefunden werden. Da nur der Wert von \mathbf{U} gesucht wird, für den J minimal wird, kann der konstante Anteil $(\mathbf{W} - \mathbf{F})^T(\mathbf{W} - \mathbf{F})$ in Formel (8.26) entfallen; die Optimiervorschrift lautet daher:

$$J = \min_{\mathbf{U}} \{ \mathbf{U}^T (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}) \mathbf{U} + 2 \mathbf{U}^T \mathbf{G}^T (\mathbf{F} - \mathbf{W}) \} \quad (9.10)$$

9.4.1 Lösungsverfahren/Solving algorithms/求解方法

Zur Suche des Minimums von J existieren verschiedene Ansätze. Um zu entscheiden, welches diskrete Optimierverfahren angewendet werden soll, müssen die Anforderungen an die Optimierung definiert werden. Folgende Entscheidungskriterien spielen dabei eine Rolle (s. auch [105, Kap. 5.3.3] und [115, Kap. 2]):

- Ist die Endlichkeit des Algorithmus gewährleistet?
- Welche Rechenzeit benötigt der Algorithmus?
- Wenn der Algorithmus nicht in einer vorgegebenen Zeit zu einem Ergebnis kommt, liefert er dann zumindest Näherungslösungen?
- Wieviel Speicherplatz wird benötigt?
- Besitzt der Algorithmus eine klare Struktur und leichte Lesbarkeit (strukturierte Programmierung)?
- Findet der Algorithmus in jedem Fall das *globale* und nicht nur ein *lokales* Minimum?

It is therefore necessary to solve the optimization problem using a *multi-step optimization*. Unfortunately, a multi-step optimization, in contrary to a single-step optimization, cannot be done analytically anymore because of the nonlinearity caused by the discrete actuating variables. Thus, a decision and search strategy which can find the best possible sequence of values for the actuating variables has to be used, i. e. the smallest value of the cost function (8.26) under consideration of the special constraints for \mathbf{U} has to be found. Since only the value of \mathbf{U} which leads to a minimized J has to be searched

for, the constant term $(\mathbf{W} - \mathbf{F})^T(\mathbf{W} - \mathbf{F})$ in equation (8.26) can be omitted; thus, the optimization rule is equation (9.10).

9.4.1 Solving algorithms

For the search for the minimum of J several methods exist. In order to decide which discrete optimization method should be used, the demands for the optimization have to be defined. The following criteria play a role for this decision (see also [105, Chapter 5.3.3] and [115, Chapter 2]):

- Can finiteness of the algorithm be guaranteed?
- Which computation time is necessary for the algorithm?
- If the algorithm does not deliver a solution in a defined time, does it deliver at least approximate solutions instead?
- How much memory is needed?
- Has the algorithm a clear structure and is it easily readable (structured programming)?
- Does the algorithm find the *global* and not only a *local* minimum in each case?

因此，有必要使用多步优化来求解这一优化问题。遗憾的是，不同于单步优化的情形，由于离散控制变量所引起的非线性使得无法以解析方式进行多步优化。正是如此，必须使用决策和搜索策略来确定最优控制变量序列。换言之，必须找到在考虑特殊条件下的成本函数 (8.26) 的最小值 J 。因为要寻找的是使得 J 最小的 \mathbf{U} 值，所以方程 (8.26) 中的不变项 $(\mathbf{W} - \mathbf{F})^T(\mathbf{W} - \mathbf{F})$ 可以消去，从而有方程 (9.10)。

9.4.1 求解方法

实际存在多种方法用于寻找最小值 J 。为了确定应当使用哪一种离散优化方法，必须首先对优化的标准进行定义。通常来说，应当考虑如下决策标准 (见 [105, 章节 5.3.3] 和 [115, 第 2 章]):

- 算法的有限性是否能够得到保证?
- 算法的执行需要多长的计算时间?
- 假使算法在确定时间内无法得到一个解，那么它是否能够至少提供一个近似解?
- 需要多大的存储空间?
- 算法是否具有清晰的结构，且易读性强，适合结构化编程?
- 算法是否在任一情况下均能够找到全局最小值，而非局部最小值?

In den folgenden Kapiteln werden verschiedene diskrete Optimierverfahren erläutert und unter dem Gesichtspunkt des Einsatzes in einem DMPC-Regler zur Stromregelung einer Asynchronmaschine diskutiert. Einen sehr guten Überblick bietet hierzu auch die Veröffentlichung von Terno [115].

Erschöpfende Suche (Vollständige Enumeration)

Das einfachste Verfahren beruht auf der Idee, alle möglichen Stellgrößenfolgen \mathbf{U} zu bestimmen und nacheinander mit der Kostenfunktion J zu bewerten. Diejenige Folge mit dem geringsten Kostenwert ist die gesuchte Lösung. Da hierbei ohne Rücksicht auf den Berechnungsaufwand sämtliche Möglichkeiten ausprobiert werden, wird das Verfahren in der Fachliteratur *erschöpfende Suche* oder *vollständige Enumeration* genannt. Wie leicht einzusehen ist, findet diese Strategie auf jeden Fall das globale Minimum, da ja alle Möglichkeiten ausprobiert werden. Am Beispiel eines zweistufigen Stellglieds wird von Tsang/Clarke [119] ein Bang-bang GPC-Regler beschrieben; als Lösungsalgorithmus empfehlen die Autoren ebenfalls die simple erschöpfende Suche. Es ist allerdings fraglich, ob diese Empfehlung auch für Systeme mit mehrstufigen Stellgliedern (z. B. dreiphasigen Umrichtern) und höheren Regelhorizonten gilt, oder ob dort nicht andere Optimierverfahren besser geeignet sind, denn der Rechenaufwand für eine erschöpfende Suche ist nicht zu unterschätzen. Für einen Steller mit x diskreten Stellmöglichkeiten gibt es bei einem Regelhorizont N_u genau x^{N_u} mögliche Schaltfolgen des Stellglieds, d. h. der Rechenbedarf steigt exponentiell mit der Prädiktionstiefe. Für einen Zweipunktumrichter mit $x = 8$ verschiedenen Schaltzuständen müssen bei einem Wert von $N_u = 2$ bereits 64, bei $N_u = 3$ schon 512 Schaltfolgen einzeln ausgerechnet werden. Wie die in Kap. 9.4.4 gezeigten Ergebnisse bestätigen, ist die Implementierung einer erschöpfenden Suchstrategie für eine antriebstechnische Regelaufgabe nicht durchführbar.

In the following chapters different discrete optimization methods are described and discussed under the aspect of their use in a DMPC controller for current control of an induction machine. A good overview about the algorithms is also given in the publication by Terno [115].

Exhaustive search (Complete enumeration)

The most simple method is based on the principle that all possible sequences of values of the actuating variables \mathbf{U} are determined first and then, one after each other is evaluated with the cost function J . The sequence which leads to the smallest value of the cost function is the solution that was searched. Since in this method, all possibilities are tested, regardless of the computational effort, this method is called *exhaustive search* or *complete enumeration* in technical literature. As it can easily be seen, this strategy will always find the global minimum because all possible combinations are tested. Us-

ing the example of a two-position actuator, a bang-bang GPC controller is described by Tsang/Clarke [119]; the authors also recommend the simple exhaustive search for solving the optimization problem. However, it is questionable if this can also be recommended for systems with multilevel actuators (e. g. three phase inverters) and for longer control horizons, or if other optimization techniques are better suited in these cases since the computation time for an exhaustive search should not be underestimated. For an actuator with x discrete actuating possibilities and a control horizon N_u exactly x^{N_u} different possible switching sequences exist, i. e. the needed computation time increases exponentially with the number of predicted time steps. For a two-level inverter with $x = 8$ different switching states and $N_u = 2$ already 64, with $N_u = 3$ as many as 512 switching sequences have to be calculated individually. As the results shown in Chapter 9.4.4 prove, an implementation of an exhaustive search strategy is not feasible in drive control.

接下来将描述几种不同的离散优化方法，并讨论其在直接模型预测控制器中实现异步电机电流控制的可能性。此外，学者 Terno 的著作 [115] 对于不同的算法做了很精辟的综述。

穷举搜索（完全遍历）

最简单易懂的方法基于下述原则：首先确定所有可能的控制变量 \mathbf{U} 的序列，然后使用成本函数 J 对其进行逐一评判。使得成本函数最小的那个控制量序列即为所求的解。由于上述方式尝试了所有可能性，而不考虑计算量问题，在专业文献中通常被称为穷举搜索或完全遍历。不难看出，上述策略在任意情况下找到的均是全局最小值，因为所有的可能性均逐一进行了尝试。以一个两位置执行机构为例，学者 Tsang/Clarke 描述了一个砰砰 GPC 控制器 [119]；作者也同样推荐使用简单的穷举搜索作为求解算法。然而，上述建议是否同样适用于具有多位式的执行机构的系统及多步长的情形呢？这一点是有待商榷的。此外，是否存在其他更为合适的优化方法，因为穷举搜索所需的计算量是不容小觑的。对于一个具有 x 个可能离散控制变量的执行机构来说，当控制步长为 N_u 时，一共存在 x^{N_u} 个可能的开关序列。换言之，计算量将随着预测步长呈指数形式递增。对于一个具有 $x = 8$ 个不同开关状态的两电平逆变器来说：当 $N_u = 2$ 时，有 64 种开关序列；当 $N_u = 3$ 时，已有 512 种开关序列。如章节 9.4.4 中的结果表明，在驱动控制领域很难实现穷举搜索策略。

Anfahrstrategie: Um das Verfahren der vollständigen Enumeration zu verbessern, hat Hoffmann es in seiner Arbeit zur Einsparung von Rechenzeit um eine sogenannte Anfahrstrategie erweitert [51, Kap. 3.3.1]. Dazu wird im dynamischen Fall, also bei großen Sollwertsprüngen, zuerst eine Stellgröße geschaltet, welche die Regelgröße mit maximaler Geschwindigkeit auf den Sollwert zubewegt. Schaltet man in einem Zeitpunkt t_1 auf einen entgegengesetzten Stellgrößenzustand, so kann man bei passender Wahl von t_1 den Sollwert in minimaler Zeit ohne Überschwinger erreichen. Die Hoffmann'sche Anfahrstrategie berechnet den Umschaltzeitpunkt t_1 und die Gesamtdauer des Vorgangs im Voraus und reduziert so im transienten Zustand den Rechenaufwand signifikant.

Vergleicht man die Anfahrstrategie von Hoffmann mit den aus der Antriebstechnik bekannten und in Kap. 4.1 vorgestellten trajektorienbasierten Prädiktivreglern, so sind beide Verfahren auf denselben Prinzipien aufgebaut. In beiden Fällen wird eine systemeigene Trajektorie durch Wahl eines entsprechenden Schaltzustands des wertdiskreten Stellglieds ausgewählt und dann ein optimaler Umschaltzeitpunkt bestimmt. Während bei den üblichen trajektorienbasierten Verfahren zu diesem vorausberechneten Zeitpunkt stets ein anderer, ebenfalls vorausberechneter Schaltzustand an die Regelstrecke angelegt wird, schaltet die Anfahrstrategie nach zwei Schritten auf die direkte modellbasierte Prädiktivregelung mittels erschöpfender Suche des optimalen Schaltzustands zurück. Das Grundprinzip der trajektorienbasierten Regelung ist jedoch bei beiden Strategien gleich.

Zur DMPC-Regelung einer Asynchronmaschine ist die Anfahrsteuerung uninteressant, da bei der Stromregelung nur sehr wenig Abtastschritte zum Erreichen des Sollwerts notwendig sind. Selbst ein Sprung von $i_{sq} = 0$ auf $i_{sq} = 1,0$ bei $\omega = 0,5$ benötigt nur 13 Abtastzyklen. Der Aufwand für die Anfahrstrategie lohnt sich folglich nicht.

Bei der Drehzahlregelung ist die nach einem Sollwertsprung zum Erreichen des neuen Werts notwendige Zeit zwar bedeutend größer, da aber auch im stationären Fall weiterhin ein relativ hoher Prädiktionshorizont notwendig ist (s. Kap. 6.3.3 auf Seite 127 ff.), führt die Implementierung einer Anfahrstrategie auch für diese Anwendung nicht zu einer Verringerung der maximal notwendigen online-Rechenzeit.

Speed-up strategy: To improve the complete enumeration method, Hoffmann has extended it by a so-called speed-up strategy [51, Chapter 3.3.1] in order to save computation time. In the dynamic case, i. e. when large steps of the reference value occur, an actuating variable which moves the controlled variable to the reference value with maximum speed is applied first. If an opposite state of the actuating variable is switched at an adequate time t_1 , the reference value can be reached in a minimal amount of time without any overshoot. In the speed-up strategy by Hoffmann the point of time t_1 for switching and the whole duration of the process are calculated in advance and hence in transient state the necessary computation time can be reduced significantly.

If the speed-up strategy by Hoffmann is compared to the trajectory based predictive

controllers, which are known in drive control and which are presented in Chapter 4.1, one can see that both methods are based on the same principles. In both cases a system trajectory is selected by choosing an appropriate switching state of the value discrete actuator and then an optimum switching time is determined. Taking conventional trajectory based methods into consideration, at this predetermined time, another, also predetermined switching state will be applied to the plant, whereas the speed-up strategy switches back to direct model-based predictive control with exhaustive search for the optimal switching state after two steps. However, both strategies are based on the basic principle of trajectory based control.

For DMPC of an induction machine, the speed-up strategy is not interesting since the current controller only needs few sampling cycles for reaching the reference value. Even a step of $i_{sq} = 0$ to $i_{sq} = 1.0$ at $\omega = 0.5$ takes just 13 sampling cycles. Thus, the effort for the speed-up strategy is not worthwhile.

Taking speed control into consideration, the time necessary for reaching the reference value after a step change is indeed significantly larger, but even in the stationary case still a relatively high prediction horizon is necessary (see Chapter 6.3.3 on page 127 et seqq.). Thus, the implementation of a speed-up strategy for this application does not lead to a decrease of the maximum necessary online computation time, too.

加速策略: 为了改善基于完全遍历的方法, 学者 Hoffmann 在其研究成果中提出了一种所谓的加速策略 [51, 章节 3.3.1], 其首要目的是减少计算量。在动态情况下, 也即当出现较大的参考值阶跃, 首先应用的是可以使被控变量以最大速度靠近参考值的控制变量。如果能够在合适的时刻 t_1 使用一个完全相反的控制变量, 那么可在最短的时间内无超调地到达参考值。学者 Hoffmann 提出的加速策略提前计算出了切换时刻 t_1 以及整个过程的持续时间, 从而在动态情况下有效地减少了计算量。

对比学者 Hoffmann 的加速策略和章节 4.1 介绍的驱动控制领域已为人熟知的基于轨迹的预测控制器可知, 二者基于相同的原理。在两种情况下, 均首先选择具有离散值的执行机构的相应开关状态来确定系统本身的轨迹, 然后确定一个最优的切换时刻。传统的基于轨迹的方法在提前计算出的时刻始终将一个同样提前计算出的开关状态施加到被控系统; 然而, 加速策略在经过两步之后则重新切换至基于穷举搜索的直接模型预测控制。尽管如此, 两种方法均是基于轨迹控制的原理。

对于异步电机的直接模型预测控制来说, 加速策略是没有意义的, 因为在电流控制是只需要很少的采样周期即可达到参考值。即使在 $\omega = 0.5$ 的条件下, 出现一个 $i_{sq} = 0$ 至 $i_{sq} = 1.0$ 的阶跃变化, 也仅需要 13 个采样周期。所以, 并不适合为了采用加速策略而额外付出更多的代价。

至于转速控制的情况, 虽说在给定一个阶跃参考值后需要明显更长的时间才能够到达新给定的参考值, 但是由于在稳态情况下仍然需要相对较高的预测步长 (见章节 6.3.3 第 127 页), 所以即使采用了加速策略也无法降低所需的最大在线计算时间。

Verzweigungsprinzip (Branch and Bound)

Schmitz schlägt in seiner Veröffentlichung [105, Kap. 5.2.2] ein rechenzeitsparendes Optimierverfahren namens *Branch and Bound* vor. Grundgedanke hierbei ist, dass nur für den ersten Pfad der Entscheidungsbaum von der Wurzel bis zum letzten Knoten durchlaufen wird. Der für diese Stellgrößenfolge ermittelte Kostenwert J wird als vorläufiges Minimum abgespeichert. Nun werden andere Pfade durch den Entscheidungsbaum nur dann noch weiter durchlaufen, wenn sie vielversprechend für ein neues Minimum sind. Dies bedeutet, dass in dem Fall, wenn der vorläufige Kostenfunktionswert eines teilweise untersuchten Pfades größer wird als das bisherige, vorläufige Minimum, der gerade untersuchte Knoten samt aller darunterliegenden Knoten nicht das globale Minimum enthalten kann. Der entsprechende Teilbaum braucht daher nicht weiter untersucht werden und wird ausgeschnitten. Das Gleiche geschieht, wenn der noch zu untersuchende Teilbaum zu einer unzulässigen Familie gehört und daher ebenfalls als Optimallösung ausscheidet [115]. Hat der Algorithmus einen möglichen Pfad bis zum Endknoten durchlaufen und ist der Kostenwert kleiner als das bisherige Minimum, so wird das vorläufige Minimum durch den Kostenwert des gerade untersuchten Baumpfades ersetzt. Auch mit dieser Suchstrategie ist sichergestellt, dass in jedem Fall das globale Minimum gefunden wird.

Das Verzweigungsprinzip führt besonders bei hohen Regelhorizonten N_u zu einer deutlichen Reduzierung des Rechenbedarfs, da Teilbäume, die vorhersehbar nicht zu einem Minimum führen, gar nicht erst untersucht werden. Eine weitere Verkürzung der Rechen-dauer stellt sich dadurch ein, dass bei der weiteren Untersuchung von Teilbäumen die vorläufigen Kosten bis zu dem aktuellen Knoten bereits bekannt sind; der Kostenanteil, der durch die zwischen dem aktuellen Knoten und der Baumwurzel liegenden Baumknoten entsteht, braucht folglich nicht erneut berechnet werden. Darüberhinaus kann weitere Rechenzeit eingespart werden, wenn man im Voraus eine Abschätzung über den in einem Teilbaum erreichbaren minimalen Kostenwert vornimmt. Liegt dieser Wert über dem bisherigen Minimum, so ist eine Auswertung des Teilbaums nicht notwendig, da sie vorhersehbar zu keiner Optimallösung führt.

Branch and Bound

In his publication [105, Chapter 5.2.2], Schmitz proposes a computation time saving optimization method called *Branch and Bound*. It is based on the principle that the decision tree is run through from the root to the last node only for the first path. The value J of the cost function valid for this sequence of values of the actuating variables is saved as the temporary minimum. Now, other paths through the decision tree are only examined further if they are promising for a new minimum. This means that, if the temporary value of the cost function of a partly run through path is already larger than the current, temporary minimum, the currently investigated path including all further

nodes cannot contain the global minimum. Thus, the correspondent part of the decision tree does not have to be run through anymore and will be cut out. The same happens if the part of the tree which is still to be investigated belongs to a prohibited family and thus, can also not contain the optimum possible solution [115]. If the algorithm has run through a possible path to the end node and if the value of the cost function for this sequence is smaller than the temporary minimum, the temporary minimum will then be replaced by the value of the cost function for the currently investigated path of the decision tree. This search strategy also guarantees that the global minimum will be found in any case .

Especially with high control horizons N_u the Branch and Bound strategy leads to a significant reduction of the computation time because branches of the decision tree that will recognizably not lead to a minimum are not investigated. A further reduction of computation time is achieved when investigating other branches of the same subtree since the temporary costs up to the current node are already known; thus, the part of the costs created by the nodes lying between the root of the decision tree and the current node does not have to be calculated again. Furthermore, additional computation time can be saved if an estimation about the minimum reachable value of the cost function in a branch is done in advance. Is this value higher than the temporary minimum, an investigation of the branch is not necessary, because it will not lead to an optimum solution.

分支定界法

学者 Schmitz 在其著作 [105, 章节 5.2.2] 中提出了一种可节约计算时间的优化方法, 并命名为分支定界法。基本思想是对搜索树的第一条路径从根节点到最后一个结点进行遍历, 并将基于这一控制变量序列所得到成本函数的值保存为当前的最小值。然后, 对搜索树的其他路径继续进行搜索的条件是该路径有可能得到一个新的最小值。换言之, 如果某条路径中的中间某个结点所得到的成本函数值已经大于当前的最小值, 那么, 此结点及其下方的所有结点不可能包含一个全局最小值。从而, 无需对上述结点对应的子树继续判断, 直接剪去即可。此外, 如果待判断的子树属于不被允许的序列, 此时无法得到一个最优解, 所以也要被剪去 [115]。假如算法遍历了某一可能的路径直至其最后一个结点, 且得到的成本函数值小于当前的最小值, 那么则使用最新路径所得的成本函数值来代替当前的最小值。基于这一搜索策略可以保证在任何情况下均能找到一个全局最小值。

尤其当控制步长 N_u 较大时, 分支定界法能够明显降低计算量, 因为所有无法得到最小值的子树将不再进行搜索。能进一步减小计算量的前提条件是: 当对子树向下搜索时, 活动结点与根结点之间的成本函数值是全部已知的, 从而可以省去重新计算所需的时间。此外, 进一步节约计算量的方式是提前估计出某一子树可能达到的最小的成本函数值, 如果该值大于当前的最小值, 那么也无需对该子树继续进行搜索, 因为可以预见的是它无法得到一个最优解。

Wie Kap. 9.4.4 zeigt, ist eine online berechnete Lösung des Optimierungsproblems mittels Branch and Bound für eine Antriebsregelung ebenfalls zu langsam. Ergänzend ist hinzuzufügen, dass Verzweungsverfahren den Nachteil haben, dass eine tatsächliche Verkürzung der Rechenzeit *nicht garantiert werden kann!* Im ungünstigsten Fall kann es nämlich vorkommen, dass in der Tat *sämtliche* Fälle durchgerechnet werden müssen.

Das Schnittprinzip (cutting planes)

Das Schnittprinzip lässt sich mit wenigen Worten erläutern. Grundidee ist hier, zuerst die Optimallösung unter Vernachlässigung der Ganzzahligkeitsbedingung zu bestimmen. Danach prüft man zuerst, ob die erhaltene Lösung vielleicht bereits Element der gültigen Menge $S \subseteq \mathbb{N}$ ist. Ist dies nicht der Fall, so werden nach und nach einschränkende Bedingungen hinzugefügt, die einen Teil des gesamten Lösungsbereichs wegschneiden; diese Einschränkungen werden als *Schnittebenen* bezeichnet. Danach wird jeweils das neue lineare Optimierungsproblem unter Berücksichtigung der Einschränkungen gelöst. Diese Schritte werden solange wiederholt, bis sich schließlich eine optimale Lösung ergibt, die der Ganzzahligkeitsbedingung entspricht. Eine ausführliche Beschreibung dieses Verfahrens findet sich neben dem Buch von Terno [115, Kap. 2.4] bei Hadley [47, ab Kap. 8.11].

Verfahren, die auf Schnittebenen basieren, haben jedoch den fundamentalen Nachteil, dass sie nur äußerst langsam konvergieren [63, Kap. 1.6.4]. Daher scheiden sie für den Gebrauch in einer Antriebsregelung von vorneherein aus.

As Chapter 9.4.4 shows, an online calculated solution of the optimization problem via Branch and Bound is also too slow for use in drive control. Additionally, it is to mention that the Branch and Bound principle has the drawback that an actual reduction of the computation time *cannot be guaranteed!* In the worst case it can happen that the cost function has to be evaluated for *all possible* sequences.

Cutting Planes

The Cutting Planes strategy can be explained with few words. The basic idea is to determine the optimum solution neglecting the fact, that the solution has to be integer. The first step is to check if the obtained solution is already an element of the valid set $S \subseteq \mathbb{IN}$. If this is not the case, constraints, which cut off parts of the area of possible solutions, are added one by one; these constraints are called *cutting planes*. Then, the new linear optimization problem is solved under consideration of the constraints. These steps are repeated until an optimum solution which fulfills the integer constraint is found. A detailed description of this method can be found in the book of Terno [115, Chapter 2.4] and also with Hadley [47, from Chapter 8.11 on].

However, methods based on cutting planes have the fundamental drawback that they converge extremely slow [63, Chapter 1.6.4]. Thus, they cannot be used in drive control.

如章节 9.4.4 所示，借助于分支定界法来实现驱动控制优化问题的在线求解过程仍然非常缓慢。需要补充的是：分支定界法的缺点是无法保证计算量的降低，因为在最坏的情况下，该方法仍然需要遍历所有可能的控制变量序列，正如穷举搜索时的情形。

切割原理 切割原理可以使用简单的几句话来进行解释。基本思想是首先确定一个忽略整数约束下的最优解，然后检查所得到的解是否已经属于有效集 $S \subseteq \mathbb{IN}$ 中的元素。如果不属于的话，则需逐渐增加约束条件，每一约束条件都会切割全部求解域的一部分，上述约束条件也被称为切割平面。随后，分别求解考虑了新的约束条件后所形成的新的线性优化问题，重复上述步骤，直至最终得到一个满足整数约束的最优解。该方法更为详尽的描述可参考学者 Terno 的著作 [115, 章节 2.4]，以及学者 Hadley 的著作 [47, 自章节 8.11 起]。

基于切割平面方法的根本性缺点是其收敛速度非常缓慢 [63, 章节 1.6.4]，从而在驱动控制中无法使用这一方法。

Das Erweiterungsprinzip

Terno erwähnt neben den bisher angesprochenen Verfahren auch noch das sogenannte *Erweiterungsprinzip* [115, Kap. 2.3]. Dieses Verfahren wurde von Schoch im Jahr 1970 an der Bergakademie Freiberg entwickelt und ist in [106] ausführlich beschrieben. Die Optimierungsaufgabe wird hier durch ein Ersatzproblem ersetzt. Dieses Ersatzproblem ist auf einem Teilbereich $x \in \mathbf{R} \cap \mathbf{S}$ des für das originäre Problem gültigen Bereichs $x \in \mathbf{S}$ definiert. Zusätzlich soll das Ersatzproblem noch die Bedingungen erfüllen, dass $\mathbf{R} \cap \mathbf{S} \neq \emptyset$, dass eine untere Schranke für alle Werte des Ersatzproblems existiert und dass das Minimum des Ersatzproblems über $x \in \mathbf{R} \cap \mathbf{S}$ auch das Minimum der originären Optimierungsaufgabe über $x \in \mathbf{S}$ ist. Es werden nun Teilmengen $\mathbf{U}_k \subset \mathbf{R}$ untersucht, die mittels oberer und unterer Schranke für das Ersatzproblem definiert sind. Die Elemente von \mathbf{U}_k werden auf Zugehörigkeit zu \mathbf{S} geprüft. Das Verfahren wird solange fortgesetzt, bis ein solches Element gefunden und als optimal erkannt worden ist. Bei dieser Strategie wird folglich die Menge der untersuchten Elemente schrittweise erweitert, daher die Bezeichnung *Erweiterungsprinzip*.

Wie Schoch weiter ausführt, lassen sich sowohl das Schnitt- als auch das Verzweigungsprinzip als Sonderfall des Erweiterungsprinzips darstellen [106, Kap. 1.2.2 bzw. 1.3.2].

Diskrete Dynamische Programmierung

Die *Dynamische Programmierung* zerlegt zur Lösung eines separablen Optimierungsproblems selbiges in mehrere Teilprobleme, die nacheinander behandelt werden. Dadurch entsteht ein mehrstufiger Entscheidungsprozess, wobei der Ausgangszustand der einen Stufe den Eingangszustand der nächsten Stufe darstellt. Mit anderen Worten: Eine mehrdimensionale Optimierungsaufgabe wird in mehrere eindimensionale Teilprobleme zerlegt, die sukzessive, am besten in rekursiver Form, gelöst werden. Laut Hoffmann [51, Kap. 3.3.3] führt die Dynamische Programmierung zwar zu einer Rechenzeiterparnis, ist aber wegen des damit verbundenen stark erhöhten Speicherbedarfs wenig geeignet. Aus diesem Grund wird sie hier nicht weiter betrachtet; tiefergehende Ausführungen können z. B. bei Cooper/Cooper [31] nachgelesen werden.

Expansion Strategy

Besides the methods already mentioned, Terno also refers to the so called *Expansion Strategy* [115, Chapter 2.3]. This method was developed by Schoch at the Mining University (Bergakademie) Freiberg in 1970; in [106] it is described in detail. Here the optimization task is solved by using a substitution problem. This substitution problem is defined in a subset $x \in \mathbf{R} \cap \mathbf{S}$ of the set $x \in \mathbf{S}$ which is valid for the original problem. Additionally, the substitution problem should fulfill the constraints that $\mathbf{R} \cap \mathbf{S} \neq \emptyset$, that a lower boundary exists for all values of the substitution problem and that the

minimum of the substitution problem defined for $x \in \mathbf{R} \cap \mathbf{S}$ represents the minimum of the original optimization task for $x \in \mathbf{S}$. Now subsets $\mathbf{U}_k \subset \mathbf{R}$ which are defined for the substitution problem via an upper and a lower boundary are examined. The elements of \mathbf{U}_k are tested if they are members of \mathbf{S} . The procedure is repeated until such an element is found. Hence, in this method the set of the examined elements is expanded step by step which leads to the name *Expansion Strategy*.

As further mentioned by Schoch, Cutting Planes as well as Branch and Bound are special cases of the Expansion Strategy [106, Chapter 1.2.2 and 1.3.2].

Discrete Dynamic Programming

For the solution of a separable optimization problem via *Dynamic Programming*, the optimization problem is separated into several sub-problems which are treated one after each other. This leads to a multi-level decision process with the output state of one level being the input state of the next one. Explained in other words this means that a multidimensional optimization task is splitted into several uni-dimensional sub-problems which will be solved successively; the best to perform this is in a recursive matter. According to Hoffmann [51, Chapter 3.3.3], Dynamic Programming can indeed save computational time, however it is less suitable because of its highly increased memory requirements. For this reason it is not discussed here any further; more detailed descriptions can be found e. g. in Cooper/Cooper [31].

扩展策略

除了前面所述方法之外, 学者 Terno 还提及了所谓的扩展策略[115, 章节 2.3]。这一方法是由弗莱贝格工业大大学学者 Schoch 所提出, 文献 [106] 对其进行了详细介绍。此时, 优化任务由一个等效问题所替代, 这一等效问题定义在原问题有效集合 $x \in \mathbf{S}$ 内的一个子集合 $x \in \mathbf{R} \cap \mathbf{S}$ 。此外, 等效问题也需满足 $\mathbf{R} \cap \mathbf{S} \neq \emptyset$ 这一约束条件, 从而, 对于等效问题的全部数值存在一个下确界, 且等效问题在 $x \in \mathbf{R} \cap \mathbf{S}$ 的最小值同时也是原问题在 $x \in \mathbf{S}$ 的最小值。仍需对由等效问题的下确界和上确界所定义的子集 $\mathbf{U}_k \subset \mathbf{R}$ 检查 \mathbf{U}_k 的元素是否属于集合 \mathbf{S} 。需要不断重复上述过程, 直至找到一个最优元素。鉴于上述策略所采用的元素集合是逐步扩展的, 所以被称为扩展策略。

正如学者 Schoch 所指出, 切割原理和分支定界法均可以看作扩展策略的特殊情况 [106, 章节 1.2.2 和 1.3.2]。

离散动态规划

为了求解一个可分离的优化问题, 动态规划首先将其分解为多个子问题, 然后逐一求解子问题。基于上述策略形成了一个多级决策过程, 其中, 某一级的输出状态作为下一级的输入状态。换言之, 一个多维的优化任务被分解为多个单维的子问题, 然后按顺序求解上述子问题, 最好是以递归方式求解。根据学者 [51, 章节 3.3.3] 所述, 尽管动态规划能够节约计算时间, 但是它却需要更多的存储空间, 从而使得此方法不太适用。本文不作进一步介绍, 读者可参考学者 Cooper 的著作 [31] 及其他文献。

9.4.2 Mathematische Herleitung/Mathematical derivation/数学推导

Die an dieser Stelle vorgestellte DMPC-Regelung mit impliziter Lösung des Optimierungsproblems ist, wie bereits erläutert, eine GPC-Regelung, bei der jedoch die wertdiskrete Struktur des Stellglieds berücksichtigt wird. Da sich dies jedoch ausschließlich auf die Optimierungsmethode auswirkt, bleiben die in Kap. 8 vorgestellten mathematischen Hintergründe unverändert gültig. Die Berechnung der Systemmatrizen des CARIMA-Modells, des j -Schritt-voraus-Prädiktors etc. sind dort beschrieben. Die Optimierungsvorschrift für den direkten GPC-Regler ist in Formel (9.10) auf Seite 257 mit den dazugehörigen Erläuterungen angegeben.

9.4.3 Experimentelle Ergebnisse/Experimental results/实验结果

Wie bereits erläutert, lässt sich eine GPC-Regelung ohne Messwertfilter nicht vernünftig betreiben; daher wird direkt eine Regelstruktur mit integrierten Filtereigenschaften gewählt. Die Prädiktionshorizonte betragen $N_p = 3$ und $N_u = 2$; die Grade der Systempolynome $na = 1$, $nb = 1$ und $nt = 1$. Eine Störgrößenberücksichtigung ist bei dem verwendeten einfachen Modell (Abb. 9.6) nicht möglich und daher auch nicht implementiert. Es ergeben sich folgende System- und Polynommatrizen:

$$\mathbf{G}(z^{-1}) = \begin{bmatrix} \frac{0,1713 z^{-1}}{1-0,9873 z^{-1}} & -\frac{0,08566 z^{-1}}{1-0,9873 z^{-1}} & -\frac{0,08566 z^{-1}}{1-0,9873 z^{-1}} \\ 0 & \frac{0,1484 z^{-1}}{1-0,9873 z^{-1}} & -\frac{0,1484 z^{-1}}{1-0,9873 z^{-1}} \\ 0 & 0 & 0 \end{bmatrix} \quad (9.11)$$

$$\mathbf{T}(z^{-1}) = \begin{bmatrix} 1 - 0,95 z^{-1} & 0 & 0 \\ 0 & 1 - 0,95 z^{-1} & 0 \\ 0 & 0 & 1 - 0,95 z^{-1} \end{bmatrix} \quad (9.12)$$

$$\mathbf{A}(z^{-1}) = \begin{bmatrix} 1 - 0,9873 z^{-1} & 0 & 0 \\ 0 & 1 - 0,9873 z^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9.13)$$

$$\mathbf{B}(z^{-1}) = \begin{bmatrix} 0,1713 z^{-1} & -0,08566 z^{-1} & -0,08566 z^{-1} \\ 0 & 0,1484 z^{-1} & -0,1484 z^{-1} \\ 0 & 0 & 0 \end{bmatrix} \quad (9.14)$$

Zum Vergleich des Einflusses der Stellgrößenbewertung wurden für λ zwei verschiedene Matrizen gewählt und zwar einmal eine geringe Bewertung der Stellgrößenänderung mit $\lambda = 0,001$ und einmal eine deutlich höhere Bewertung des Stellaufwands mit $\lambda = 0,1$.

Wie die aus Formel (8.24) hergeleitete Gl. (9.10) zeigt, geht in die Bewertung nicht die Stellgröße selbst, sondern die *Veränderung der Stellgröße* ein. Eine Stellgrößenänderung in einer der drei Eingänge a , b oder c bedeutet aber gleichzeitig das Schalten einer Halbbrücke. Daher wird mit dem Bewertungsfaktor λ hier in der Tat die Schalthäufigkeit des Umrichters bewertet (siehe Kap. 9.3).

9.4.2 Mathematical derivation The DMPC controller with implicit solution of the optimization task presented here is, as already explained, a GPC strategy in which the discrete-valued structure of the actuator is considered. Since this affects only the method of optimization, the mathematical derivations presented in Chapter 8 are valid without any modifications. The calculation of the system matrices of the CARIMA model, the j -step ahead predictor etc. are described there. The optimization rule is given in equation (9.10) on page 257 together with explanations.

9.4.3 Experimental results As already mentioned, a GPC control without filtering of the measured values cannot be used reasonably; hence, a controller structure with integrated filter characteristics is selected. The prediction horizons are $N_p = 3$ and $N_u = 2$; the degrees of the system polynomials are $na = 1$, $nb = 1$ and $nt = 1$. With this simple model (Figure 9.6) it is not possible to consider disturbances and hence, this is not implemented either. The following system and polynomial matrices result, see equations (9.11) to (9.14).

In order to compare the influence of the weighing of the actuating variables, two different matrices were selected as λ , one for a little weighing of the change of the values of the actuating variables with $\lambda = 0.001$ and one for a significantly higher weighing of the change with $\lambda = 0.1$.

Equation (9.10), derived from equation (8.24), shows that for the weighting not the values of the actuating variables themselves are considered, but the *change of the values of the actuating variables*. A change in one of the three inputs a , b or c means that a half bridge has to be switched. Hence, with the weighting factor λ the number of inverter switching actions is evaluated indeed (see Chapter 9.3).

9.4.2 数学推导 如前面章节所述, 本部分介绍的使用优化问题的隐式求解方法实现的直接模型预测控制是一种 GPC 控制, 它可以考虑到执行机构的数值离散结构特性。由于上述考虑所影响的仅仅是优化方法, 所以第 8 章所介绍的数学推导部分无需作出任何变化而依然适用。CARIMA 模型中系统矩阵的计算、 J 步前向预测器等相关内容也在第 8 章中进行了介绍, 此处不再赘述。此外, 第 257 页所示的方程 (9.10) 及相应的解释给出了直接 GPC 控制的优化规则。

9.4.3 实验结果 如前所述, 不包括测量值滤波器的 GPC 控制器是无法正常运行的, 因此需选择一个已经整合了滤波特性的控制结构。预测步长 $N_p = 3$, 控制步长 $N_u = 2$; 系统多项式的阶数分别为 $na = 1$, $nb = 1$ 和 $nt = 1$ 。基于第 251 页图 9.6 所示的简化电机模型无法考虑扰动, 因此不可实现。系统矩阵和多项式矩阵可见方程 (9.11) 至 (9.14)。

为了考察控制变量的权重系数影响, 现选择两个不同的矩阵作为 λ : 一个是控制增量较小的权重系数 $\lambda = 0.001$; 一个是明显较大的权重系数 $\lambda = 0.1$ 。

如由方程 (8.24) 所推导出的方程 (9.10) 所示, 上述权重系数反映的并非是控制变量本身, 而是其变化情况, 也即控制增量。三个输入量 a , b 或 c 中某一个量的变化意味着对应半桥状态的切换, 所以, 权重系数 λ 反映了逆变器的开关次数 (见章节 9.3)。

Aus Gründen, die in Kap. 9.2 dargelegt wurden, ist ein DMPC-Regler besser in stationären Koordinaten zu realisieren als in einem rotierenden Koordinatensystem. Daher sind die Ergebnisse in $\alpha\beta$ -Koordinaten aussagekräftiger. Abb. 9.7 zeigt das Verhalten eines einfachen MIMO-DMPC-Stromreglers im stationären Zustand. Wie aufgrund der Ausführungen in Kap. 9.2 erwartet, tritt trotz Regelung in Statorkoordinaten kein Schleppfehler auf.

Der Vergleich von Abb. 9.7(a) mit 9.7(b) verdeutlicht den Einfluss des Bewertungsfaktors λ auf das Verhalten des geschlossenen Regelkreises. Ein größerer λ -Wert führt zu einer höheren Gewichtung des Stellaufwands; folglich nimmt der Regler nun eine höhere Regelabweichung in Kauf, um ein häufiges Schalten der Halbleiter zu vermeiden. Gleiche Ergebnisse scheinen auf den ersten Blick auch mit einem einfachen Hystereseregler möglich zu sein – dies ist allerdings falsch, denn es gibt einige gravierende Unterschiede:

- Bei einem Hystereseregler findet der Schaltvorgang immer erst nach Überschreiten einer festen Fehlergrenze statt. Ein DMPC-Regler wählt jedoch in jedem Abtastzyklus eine optimale zukünftige Schaltfolge gemäß der gewählten Optimierungskriterien. Damit wird je nach momentanem Systemzustand bei unterschiedlich großen Fehlern geschaltet.
- Ein Hystereseregler nimmt keine echte Bewertung des Stellaufwands vor. Entsprechend kann ein hysteresebasierter Zweipunktregler z. B. nicht entscheiden, welcher der beiden mit einem Umrichter schaltbaren Nullzeiger vom Stellaufwand her günstiger ist.
- Gleichfalls ist es bei Verwendung eines Hysteresereglers nicht möglich, bestimmte, unerwünschte Umrichterschaltzustände von vornherein von der Optimierung auszuschließen (s. Kap. 9.2.1).

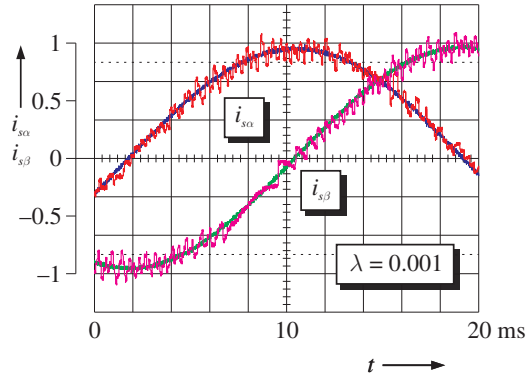
Weiterhin ist noch zu bemerken, dass der Bewertungsfaktor wohl die Schalthäufigkeit und die Regelabweichung beeinflusst, nicht aber die Dynamik des Gesamtsystems. Abb. 9.8 bestätigt dies; die Abbildung zeigt die Sprungantwort des geregelten Systems. Um neben der Systemdynamik noch die gegenseitige Beeinflussung von $i_{s\alpha}$ und $i_{s\beta}$ bewerten zu können, wurde nur der Sollwert von $i_{s\alpha}$ sprunghaft verändert. Wie un schwer zu erkennen ist, ist die Anstiegszeit sowohl in Abb. 9.8(a) bei $\lambda = 0,001$ als auch bei der Einstellung $\lambda = 0,1$ in Abb. 9.8(b) exakt gleich und entspricht dem physikalisch minimal möglichen Wert. Lediglich die Schalthäufigkeit ist bei einer geringeren Gewichtung des Stellaufwands (Abb. 9.8(a)) deutlich höher. Eine gegenseitige Beeinflussung der beiden Statorstromkomponenten ist nicht feststellbar, da die Amplitude der Stromüberschwingungen bedeutend größer ist als der Einfluss der Kreuzkopplung. Folglich macht es keinen Sinn, das hier verwendete einfache Maschinenmodell gemäß Abb. 9.6 um eine Berücksichtigung der Kreuzkopplung zwischen $i_{s\alpha}$ und $i_{s\beta}$ zu ergänzen, da hierdurch lediglich der Aufwand für die Regelung steigen würde, ohne eine Verbesserung der Regelqualität zu erzielen.

For reasons explained in Chapter 9.2, a DMPC controller is better realized in stationary coordinates than in a rotating reference frame. Hence, results shown in $\alpha\beta$ coordinates are more meaningful. Figure 9.7 shows the behavior of a simple MIMO-DMPC current controller in stationary state. Matching the explanations in Chapter 9.2, no contouring error occurs.

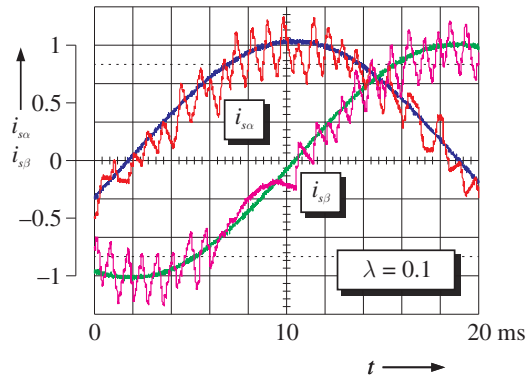
Comparing Figure 9.7(a) with Figure 9.7(b) shows the influence of the weighting factor λ on the behavior of the closed control loop. A higher value for λ leads to a higher weighting of the change of the actuating variables; hence, the controller now accepts a higher control error in order to avoid frequent switching of the semiconductors. It seems that similar results can be obtained when using a simple hysteresis controller—however, this is not true because there are some serious differences:

- When using a hysteresis controller, a switching event only takes place when a constant error limit has been exceeded. However, DMPC selects an optimal future switching sequence in advance according to the selected optimization criteria. This means that dependent on the current system state, switching takes place at different error values.
- A hysteresis controller does not really evaluate the switching effort. Thus, a hysteresis-based two-level controller can e. g. not decide which one of the two zero voltage space vectors, being able to be switched by the inverter, is favorable for less switching actions.
- When using a hysteresis controller, it is not possible to exclude certain undesired switching states of the inverter already in the beginning of the optimization (see Chapter 9.2.1).

Furthermore, it is to be noted that the weighting factor does influence the number of switching events and the control error, but not the dynamics of the entire system. Figure 9.8 showing the step response of the controlled system confirms this. In order to evaluate the mutual influence of $i_{s\alpha}$ and $i_{s\beta}$ besides the system dynamics, only the reference value for $i_{s\alpha}$ was step changed. As it can be seen easily, the rise time in Figure 9.8(a) with $\lambda = 0.001$ as well as with $\lambda = 0.1$ in Figure 9.8(b) is exactly the same and corresponds to the the physically possible minimum value. Only the number of switching events is much higher when less importance is given to the switching effort (Figure 9.8(a)). A mutual influence of both stator current components cannot be detected since the amplitude of the current harmonics is significantly higher than the influence of the cross coupling. Hence, it makes no sense to add a consideration of the cross coupling between $i_{s\alpha}$ and $i_{s\beta}$ to the simple machine model according to Figure 9.6 because it would only result in a higher controller complexity without an improvement of the control quality.



(a)



(b)

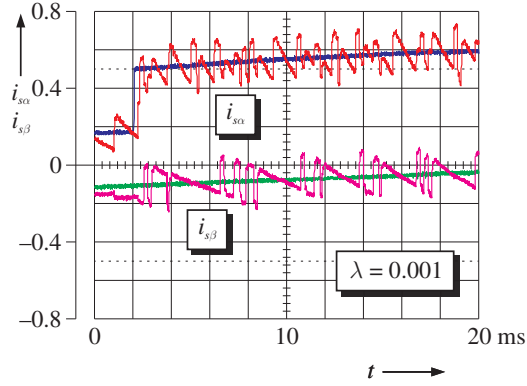
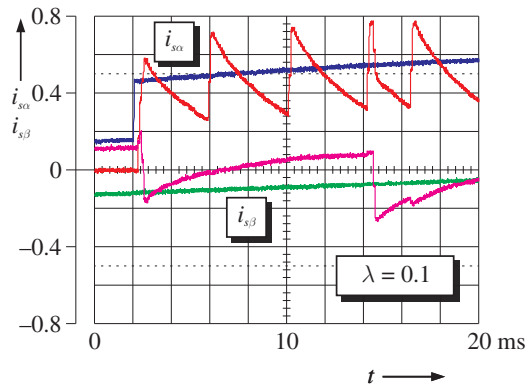
- 9.7: Stromregelung mit DMPC-Regler, stationärer Betrieb: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$;
 Current control with DMPC controller, stationary operation: (a) $\lambda = 0,001$, (b)
 $\lambda = 0,1$;
 基于 DMPC 控制器的电流控制，稳态运行: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$

根据章节 9.2 所给出的理由，最好在静止坐标系下实现 DMPC 控制器，而非旋转坐标系。因此，基于 $\alpha\beta$ 坐标所呈现的结果会更具说服力。图 9.7 给出了一个简单的 MIMO-DMPC 电流控制器的稳态特性。如章节 9.2 所给出的解释，即使在定子坐标系下也不会出现跟踪误差。

对比图 9.7(a) 和图 9.7(b) 可以清晰看出权重系数 λ 对于闭环控制特性的影响。较大的 λ 值会导致对于控制增量更高的惩罚，从而控制器以较大的控制偏差为代价来避免逆变器半桥的频繁状态切换。看似基于简单的滞环控制器也可以得到相同的结果，但这一理解显然是错误的，因为二者之间存在一些本质的区别：

- 对于滞环控制器来说，总是在超过某一固定的误差范围内才进行开关状态的切换。而直接模型预测控制器则是基于确定的优化准则在每一个采样周期选择一个最优开关序列，此时，开关状态的切换可能出现在不同大小的误差值，它取决于当前的系统状态。
- 对于滞环控制器来说，它并未真正对控制增量进行惩罚。因此，一个基于滞环的两位式控制器无法决定逆变器的两个电压矢量中的哪一个对应较少的开关次数。
- 在应用滞环控制器时，无法在优化的初始阶段提前排除一些确定的、不期望的逆变器开关状态（见章节 9.2.1）。

还应指出的是，虽说权重系数会影响开关频率和控制偏差，但却不影响整个系统的动态特性。图 9.8 所示的被控系统的阶跃响应证实了上述论断。为了能够在评判系统动态特性之外同时评判 $i_{s\alpha}$ 和 $i_{s\beta}$ 之间的互相影响，我们仅令 $i_{s\alpha}$ 的参考值发生阶跃变化。不难看出，图 9.8(a) 所示的 $\lambda = 0.001$ 的情况与图 9.8(b) 所示的 $\lambda = 0.1$ 的情况具有几乎相同的上升时间，并且它们对应的是物理意义上的可能的最小值。开关频率则在惩罚控制增量的权重系数较小时明显更高，如图 9.8(a) 所示。定子电流分量 $i_{s\alpha}$ 和 $i_{s\beta}$ 之间的相互影响是难以确定的，因为电流谐波的幅值要明显大于耦合因素的影响。因此，在图 9.6 所示的简化电机模型的基础上添加 $i_{s\alpha}$ 和 $i_{s\beta}$ 之间的耦合关系毫无意义，因为它徒使控制系统的复杂性增加，却无法改善控制效果。

(a) $\lambda = 0,001$ (b) $\lambda = 0,1$

9.8: Stromregelung mit DMPC-Regler, dynamischer Betrieb: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$;

Current control with DMPC controller, dynamic operation: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$;

基于 DMPC 控制器的电流控制，动态运行: (a) $\lambda = 0,001$, (b) $\lambda = 0,1$

Notizen/Notes/笔记

9.4.4 Rechenzeiten/Computation times/计算时间

Zur Untersuchung der Realisierbarkeit einer DMPC-Regelung im Bereich der Antriebstechnik wurden die notwendigen Rechenzeiten bei unterschiedlichen Reglerkonfigurationen und Lösungsverfahren ermittelt. Die entsprechenden Ergebnisse können Tabelle 9.2 entnommen werden.

Verfahren	N_u	Fälle	max. Rechenzeit
vollst. Enumeration	2	64	35 μs
vollst. Enumeration	3	512	> 500 μs
Verzweigungsprinzip	2	≤ 64	27 μs
Verzweigungsprinzip	3	≤ 512	186 μs

9.2: Rechenzeiten DMPC-Regler

Wie sich unschwer erkennen lässt, führen bei Verfahren mit vollständiger Enumeration Regelhorizonte N_u von mehr als 2 Prädiktionsschritten zu inakzeptablen Rechenzeiten. Dabei macht sich nicht nur die gestiegene Anzahl von möglichen Schaltfolgen bemerkbar; durch die Erhöhung der Horizonte vergrößern sich darüberhinaus auch die verschiedenen Berechnungsmatrizen, so dass der für die Berechnung des Bewertungsfaktors J notwendige Rechenaufwand steigt. Da der Kostenwert J bei der vollständigen Enumeration für *jede mögliche Schaltfolge* ermittelt werden muss, sind die beiden Zunahmen des Rechenaufwands multiplikativ verknüpft. Eine DMPC-Stromregelung mit einem Regelhorizont von $N_u = 3$ ist damit nur noch mit Abtastraten oberhalb etwa 650 μs , entsprechend einer Abtastfrequenz von höchstens 1,5 kHz, möglich. Regelhorizonte mit $N_u > 3$ sind nicht mehr realisierbar, da sie eine zu niedrige Abtastrate erfordern würden.

Die Anwendung des Branch-and-Bound-Verfahrens zur Stromregelung einer Asynchronmaschine zeigt, dass sich bei kleinen Regelhorizonten die Verringerung des Berechnungsaufwands wie erwartet nur wenig bemerkbar macht. Bei den gleichen Grundvoraussetzungen wie bei der erschöpfenden Suche ergeben sich nun für $N_u = 2$ Berechnungszeiten von maximal 27 μs . Bei einem Regelhorizont von $N_u = 3$ benötigt die Berechnung zwischen 4,6 μs und 186 μs .

9.4.4 Computation times For investigating the feasibility of DMPC for drive systems, the computation times for different controller configurations and solution methods were determined. The corresponding results can be seen in Table 9.3.

As it can be seen easily, considering methods with full enumeration control horizons N_u of more than two prediction steps leads to unacceptable computation times. Thereby, not just the higher number of possible switching sequences becomes noticeable; enlarging the horizons enlarges the different matrices further, which leads to an increase of the computational effort necessary for the calculation of the cost function J . Since, considering complete enumeration, the cost function J has to be evaluated for *every*

possible sequence of switching states, both increases of the computational effort multiply one with each other. Thus, a DMPC current control with a control horizon of $N_u = 3$ is only feasible with sample cycles above $650 \mu\text{s}$, corresponding to a maximum sampling frequency of 1.5 kHz . Control horizons with $N_u > 3$ cannot be realized due to the low sampling frequency needed.

Method	N_u	Events	Max. computation time
Complete Enumeration	2	64	35 μs
Complete Enumeration	3	512	> 500 μs
Branch and Bound	2	≤ 64	27 μs
Branch and Bound	3	≤ 512	186 μs

9.3: Computation times for a DMPC controller

The application of the Branch and Bound method for current control of an induction machine shows that, with smaller control horizons, the reduction of the computational effort becomes, as expected, only slightly noticeable. Taking the same basic conditions as for exhaustive search, for $N_u = 2$ a maximum computation time of $27 \mu\text{s}$ results. For a control horizon of $N_u = 3$ the computation lasts between $4.6 \mu\text{s}$ and $186 \mu\text{s}$.

9.4.4 计算时间 为了研究 DMPC 控制在驱动技术领域应用的可行性，需要确定的是在不同控制器配置和求解方法的条件下所必须的计算时间。表 9.4 给出了几种不同情况下的最大计算时间。

方法	N_u	事件	最大计算时间
完全遍历	2	64	35 μs
完全遍历	3	512	> 500 μs
分支定界法	2	≤ 64	27 μs
分支定界法	3	≤ 512	186 μs

9.4: DMPC 控制器的计算时间

不难看出，在完全遍历方式下，当控制步长 N_u 大于 2 时会导致无法接受的最大计算时间。要说明的是，计算时间的增加并非仅仅由于可能的开关序列数量的增加所引起，控制步长的增加还会导致不同的计算矩阵的扩展，从而用于计算成本函数 J 的权重系数所必须的时间也增大。由于在完全遍历方式下，需确定所有可能开关序列下的成本函数值 J ，所以上述两种计算量的增加是与乘法相关的。控制步长 $N_u = 3$ 时的 DMPC 电流控制所需的采样时间要大于 $650 \mu\text{s}$ ，其对应的最大采样频率约为 1.5 kHz 。控制步长 $N_u > 3$ 时的情形则无法实现，因为它要求很低的采样频率。

分支定界法用于实现异步电机的电流控制结果表明，在较小的控制步长时，计算量的减小正如所预期的一样不够明显。在与穷举搜索相同的前提条件下： $N_u = 2$ 时，所需的最大计算时间为 $27 \mu\text{s}$ ；而当控制步长 $N_u = 3$ 时，所需时间范围为 $4.6 \mu\text{s}$ 至 $186 \mu\text{s}$ 。

Der große Unterschied zwischen maximaler und minimaler Berechnungsdauer ist typisch für Verfahren nach dem Verzweigungsprinzip, da je nach Betriebspunkt und Sollwerten die optimale Schaltfolge unterschiedlich schnell gefunden wird. Trotzdem zeigt der Versuch, dass auch mit Branch and Bound eine DMPC-Regelung mit Regelhorizonten größer $N_u = 2$ für die Maschinenregelung nicht sinnvoll einsetzbar ist.

Wie der Vergleich der vollständigen Enumeration mit einer auf dem Verzweigungsprinzip basierenden Suchstrategie zeigt, ist durch die Verwendung eines intelligenten Suchalgorithmus zwar eine Rechenzeitverkürzung möglich, allerdings reicht die Beschleunigung für die Bedürfnisse der Antriebstechnik bei weitem nicht aus. Aus der Mathematik sind nun zwar weitere, sehr elegante Lösungsverfahren speziell für boolesche Optimierungsprobleme bekannt, mit denen sich viel Rechenzeit einsparen lässt. Als Beispiel sei das Verfahren von Lasserre für quadratische, boolesche Optimieraufgaben genannt [78]. Lösungsstrategien für lineare, boolesche Programme diskutiert Terno in seinem Buch [115, Kap. 5.10]. Da die mathematische Forschungsarbeit auf diesem Gebiet noch nicht abgeschlossen ist, macht es keinen Sinn, hier eine vollständige Aufzählung boolescher Optimieralgorithmen zu liefern; der mathematisch interessierte Leser wird stattdessen in den entsprechenden Datenbanken für Zeitschriftenaufsätze unter der AMS Klassifikation 90C09 suchen¹. Es ist allerdings zweifelhaft, ob die dort vorgestellten Verfahren zum Erfolg, d. h. zu einer genügend schnellen Abarbeitung der Optimieraufgabe führen. Selbst der Einsatz von sehr speziellen Suchstrategien wird die zur online-Lösung des Optimierungsproblems notwendige Zeit nicht unter einen Wert senken können, der für einen vernünftigen Einsatz des Verfahrens in der Antriebstechnik notwendig ist. Folglich ist eine direkte modellbasierte Prädiktivregelung mit *impliziter Lösung* des Optimierungsproblems in der Antriebstechnik *nicht einsetzbar*.

The big difference between the maximum and minimum computation time is typical for Branch and Bound methods because, dependent on the current system state and reference values, the optimum switching sequence is found in a different amount of time. Nevertheless, the experiments show that even with Branch and Bound, a DMPC control with control horizons higher than $N_u = 2$ is not suitable for drive control.

The comparison between complete enumeration and a search strategy based on Branch and Bound shows that by using an intelligent search algorithm, a reduction of the computation time is possible. This acceleration, however, is absolutely not sufficient for the needs in drive systems. However, from mathematics, other very elegant solution algorithms, which can save much computation time, are known especially for boolean optimization problems. As an example, one can mention the method by Lasserre for quadratic boolean optimization tasks [78]. Solution strategies for linear boolean pro-

¹ Die Zahlengruppe 90C09 ist die Codenummer der *American Mathematical Society* für die Themengruppe „Boolean Programming“.

grams are discussed by Terno [115, Chapter 5.10]. As the mathematical research in this area is not completed, providing an entire list of boolean optimization algorithms here does not make sense; readers interested in mathematics can look for papers with the AMS classification 90C09¹ in adequate data bases. Indeed, it is questionable if the methods presented there will lead to success, i.e. to a sufficiently fast solution of the optimization task. Even the application of very specialized search strategies will not be able to bring the computation time necessary for the online-solution of the optimization problem under a value which is needed for a reasonable use of the method in drive technology. Hence, direct model-based predictive control with an *implicit solution* of the optimization problem is *not feasible* in drive technology.

最大和最小计算时间之间如此之大的差别正是由于分支定界法的典型特点所致，因为最优开关序列的确定速度取决于工作点和参考值。尽管如此，实际数据表明，当控制步长 N_u 大于 2 的情况下，即使采用分支定界法也无法基于 DMPC 实现电机控制。

完全遍历与分支定界法之间的对比结果表明，虽说借助于智能的搜索算法能够减小计算时间，显然这一减少的程度对于驱动技术领域来说还是远远不够的。在数学领域，对于布尔优化问题，还有一些可以节约更多计算时间的更为有效的求解算法，例如学者 Laserre 所提出的用于二次布尔优化任务的方法 [78]。学者 Terno 在其专著 [115, 章节 5.10] 中讨论了用于线性布尔规划的求解策略。由于上述领域的数学研究工作仍在持续进行中，在此提供一个完整的布尔优化算法列表也没有意义。对于数学感兴趣的读者可以查询美国数学学会分类代码 90C09¹ 下的相关资料。然而，上述资料所介绍的方法是否切实可行也是不确定的。换言之，是否能够以足够快的速度求解优化任务。总之，即使采用非常特别的搜索策略也无法能够将优化问题在线求解所需的时间降低至驱动技术领域中所期望的数值以下。从而，借助于优化问题的隐式求解方法实现驱动技术领域中的直接模型预测控制是不大可行的。

¹ The number 90C09 is the *American Mathematical Society* code number for “Boolean Programming”.

¹ 代码 90C09 是美国数学学会对于主题“布尔规划”的分类代码。

9.5 Explizite Lösung/Explicit solution/显式求解

Im Gegensatz zu den meisten Wissenschaftlern, die sich mit modellbasierter Prädiktivregelung beschäftigen und die das entstehende quadratische oder lineare Optimierungsproblem implizit lösen, haben Bemporad und Morari einen anderen Weg gewählt. Sie versuchen, das Optimierungsproblem in Abhängigkeit vom Zustandsvektor \mathbf{x} des Systems explizit, d. h. quasi offline zu lösen [11, 12]. Im Gegensatz zu den bisher erläuterten Verfahren, die auf GPC beruhen und ein Übertragungsfunktionsbasiertes Systemmodell verwenden, benutzt der von Bemporad/Morari angewandte klassische MPC-Ansatz ein Zustandsraummodell der zu regelnden Strecke (vergleiche auch Formeln (5.4) und (5.5) auf Seite 61):

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (9.15)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (9.16)$$

unter Berücksichtigung der Beschränkungen

$$\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n, \quad \mathbf{y} \in \mathbf{Y} \subset \mathbb{R}^p, \quad \mathbf{u} \in \mathbf{U} \subset \mathbb{R}^m \quad (9.17)$$

Des Weiteren wird eine Kostenfunktion wie folgt definiert:

$$\begin{aligned} J(N_1, N_2) = & \mathbf{x}(t + N_2)^T \mathbf{P} \mathbf{x}(t + N_2) \\ & + \sum_{j=N_1}^{N_2} \left(\mathbf{x}(t + j - 1)^T \mathbf{Q} \mathbf{x}(t + j - 1) \right. \\ & \left. + \mathbf{u}(t + j - 1)^T \mathbf{R} \mathbf{u}(t + j - 1) \right) \end{aligned} \quad (9.18)$$

In diesem Fall handelt es sich um eine quadratische Kostenfunktion mit endlichem Zeithorizont, dabei bewerten \mathbf{Q} die Systemzustände und \mathbf{R} die Stellgrößen; \mathbf{P} gewichtet den Endwert des Zustandsvektors. Lineare Kostenfunktionen mit endlichem oder unendlichem Kostenhorizont sind ebenfalls denkbar. Des Weiteren ist die Kostenfunktion (9.18) gegenüber (6.15) und (8.24) dadurch vereinfacht worden, dass der Referenzwert $\mathbf{w}(t+j)$ für alle $j \geq 0$ zu Null angenommen wird. Wie bereits bekannt, wird der Einfachheit halber meist auch $N_1 = 1$ und $N_2 = N_p$ gesetzt. Diese Kostenfunktion ist nun unter den folgenden Bedingungen zu minimieren:

$$\mathbf{x}(t+j) \in \mathbf{X} \quad j = 1 \dots N_p \quad (9.19)$$

$$\mathbf{u}(t+j) \in \mathbf{U} \quad j = 0 \dots N_p \quad (9.20)$$

$$\mathbf{x}(t+j+1) = \mathbf{A}\mathbf{x}(t+j) + \mathbf{B}\mathbf{u}(t+j) \quad k \geq 0 \quad (9.21)$$

$$\mathbf{u}(t+j) = \mathbf{K}\mathbf{x}(t+j) \quad N_1 \leq k \leq N_2 \quad (9.22)$$

$$\mathbf{Q} = \mathbf{Q}^T \succeq 0 \quad \text{„}\succeq\text{“ bedeutet positiv semidefinit} \quad (9.23)$$

9.5 Explicit solution

Contrary to most researchers who deal with model-based predictive control and who solve the emerging quadratic or linear optimization problem implicitly, Bemporad and Morari have chosen another way: They try to solve the optimization problem in dependence of the state vector \mathbf{x} of the system explicitly, i. e. quasi offline [11, 12]. In contrast to the methods described so far, which are based on GPC and which make use of a transfer function based system model, the classical MPC approach used by Bemporad and Morari uses a state space model of the system to be controlled (compare equations (5.4) and (5.5) on page 61), see equation (9.15) and (9.16), under consideration of the constraints (9.17).

Furthermore, a cost function is defined as follows, see equation (9.18). In this case, we have a quadratic cost function with finite time horizon, \mathbf{Q} weights the system states and \mathbf{R} the actuating variables; \mathbf{P} weights the final value of the state vector. Linear cost functions with a finite or infinite cost horizon are also possible. Furthermore, the cost function (9.18) has been simplified compared to (6.15) and (8.24) by assuming the reference value $\mathbf{w}(t+j)$ being zero for all $j \geq 0$. As already known, $N_1 = 1$ and $N_2 = N_p$ are chosen for reasons of simplicity. Now this cost function has to be minimized under the following constraints, see constraints (9.19) to (9.23).

9.5 显式求解

不同于大部分从事于模型预测控制领域的研究学者，学者 Bemporad 和 Morari 则另辟蹊径，对于所形成的线性或二次优化问题采取显式求解的方式。二位学者尝试基于系统的状态向量 \mathbf{x} 来显式求解优化问题，换言之，以近似离线的方式求解 [11, 12]。不同于前面章节所述的借助于传递函数模型所实现的 GPC 方法，学者 Bemporad 和 Morari 使用的是经典 MPC 方法用到的是被控系统状态空间模型（对比第 61 页方程 (5.4) 和 (5.5)），见方程 (9.15) 和 (9.16)，并考虑约束条件 (9.17)。

更进一步，定义如下所示的成本函数，见方程 (9.18)。在这一情况下表达的是有限时间步长内的二次成本函数，其中 \mathbf{Q} 、 \mathbf{R} 和 \mathbf{P} 分别代表系统状态、控制变量和状态向量终值的权重系数。当然，具有有限或无限成本步长的线性成本函数也是可想而知的。对比第 79 页方程 (6.15) 和第 175 页方程 (8.24) 可知，成本函数 (9.18) 是基于对于所有的 $j \geq 0$ 均假设参考值 $\mathbf{w}(t+j)$ 为零这一条件下的简化形式。如前所述，为简化起见，通常设置 $N_1 = 1$ ， $N_2 = N_p$ 。从而，要在满足下述约束条件的情况下求得成本函数的最小值，见约束条件 (9.19) 至 (9.23)。

$$\mathbf{R} = \mathbf{R}^T \succ 0 \quad \text{„}\succ\text{“ bedeutet positiv definit} \quad (9.24)$$

$$\mathbf{P} \succ 0 \quad (9.25)$$

Durch Ersetzen von

$$\mathbf{x}(t+j) = \mathbf{A}^j \mathbf{x}(t) + \sum_{k=0}^{j-1} \mathbf{A}^k \mathbf{B} \mathbf{u}(t+j-1-k) \quad (9.26)$$

erhält man das Optimierungsproblem, die Kostenfunktion (9.18) zu minimieren, in der Form:

$$J'(\mathbf{x}(t)) = \frac{1}{2} \mathbf{x}(t)^T \mathbf{Y} \mathbf{x}(t) + \min_{\mathbf{U}} \left\{ \frac{1}{2} \mathbf{U}^T \mathbf{H} \mathbf{U} + \mathbf{x}(t)^T \mathbf{F} \mathbf{U} \right\} \quad (9.27)$$

unter der Randbedingung

$$\mathbf{G} \mathbf{U} \leq \mathbf{W} + \mathbf{E} \mathbf{x}(t) \quad (9.28)$$

mit

$$\mathbf{U} = \left[\mathbf{u}(t)^T, \dots, \mathbf{u}(t+N_u-1)^T \right]^T \in \mathbb{R}^s \quad s = m \cdot N_u \quad (9.29)$$

$$\mathbf{H} = \mathbf{H}^T \succ 0 \quad (9.30)$$

Die Matrizen \mathbf{H} , \mathbf{F} , \mathbf{Y} , \mathbf{G} , \mathbf{W} und \mathbf{E} sind aus \mathbf{Q} , \mathbf{R} und der ursprünglichen Optimierungsvorschrift, basierend auf (9.18) berechenbar. Da als Ergebnis der Optimierung nur der Stellgrößenvektor \mathbf{U} benötigt wird, kann der Term mit \mathbf{Y} in (9.27) vernachlässigt werden.

Das Optimierungsproblem in (9.27) ist ein sogenanntes *Quadratisches Programm (QP)*. Da J' vom momentanen Systemzustand $\mathbf{x}(t)$ abhängt, muss das QP bei jedem Abtastschritt erneut gelöst werden. Obwohl zeiteffiziente Lösungsalgorithmen für QP-Probleme heute bekannt sind, ist die online-Berechnung von $\mathbf{u}(t)$ für schnelle regelungstechnische Anwendungen kaum möglich; dies gilt insbesondere für die schnellen Prozesse in der Antriebstechnik. Die Anwendung des „klassischen“ MPC-Verfahrens wird daher auf Prozesse mit niedriger Abtastrate und/oder geringer Komplexität beschränkt bleiben.

In [11] schlagen Bemporad, Morari et al. ein neues Verfahren für lineare, zeitinvariante (linear time-invariant, LTI) Systeme vor, welches die Lösung des Optimierungsproblems *offline* verlagert; diese Strategie wird in [12] auf stückweise affine (piecewise affine, PWA) und hybride Systeme erweitert. Grundidee ist, dass der Zustandsvektor $\mathbf{x}(t)$ als Vektor von Parametern behandelt wird. Dadurch wird das Optimierungsproblem (9.27) zu einem *multi-parametrischen Quadratischen Programm*, kurz *mp-QP*. Durch Lösen dieses mp-QP-Problems erhält man eine Lösungsfunktion $\mathbf{U}_{opt} = f(\mathbf{x}(t))$; dadurch ist das MPC-Regelgesetz explizit verfügbar. Im Folgenden wird beschrieben, wie man zu dieser Lösungsfunktion kommt und welche Eigenschaften sie besitzt.

See constraints (9.24) and (9.25).

By replacing equation (9.26) the optimization problem can be obtained, i. e. to minimize the cost function (9.18), in the form, see equation (9.27), with the constraints (9.28), with equation (9.29) and (9.30).

The Matrices \mathbf{H} , \mathbf{F} , \mathbf{Y} , \mathbf{G} , \mathbf{W} and \mathbf{E} can be calculated from \mathbf{Q} , \mathbf{R} and from the original optimization rules, based on (9.18). Since only the vector of the actuating variables \mathbf{U} is required as the result of the optimization, the term with \mathbf{Y} in (9.27) can be neglected.

The optimization problem in (9.27) is a so called *Quadratic Program (QP)*. Since J' is dependent on the current system state $\mathbf{x}(t)$, the QP has to be solved again for every sampling cycle. Although time efficient solution algorithms for QP problems are known today, the online calculation of $\mathbf{u}(t)$ for fast applications in control engineering is hardly feasible, especially for the fast processes in drive technology. Hence, the application of the “classical” MPC method will be limited to processes with low sampling rate and/or low complexity.

In [11] Bemporad, Morari et al. proposed a new method for linear time invariant (LTI) systems which transfers the solution of the optimization problem *offline*; in [12] this strategy is expanded to piecewise affine (PWA) and hybrid systems. The basic idea is to treat the state vector $\mathbf{x}(t)$ as a parameter vector. Thereby, the optimization problem (9.27) becomes a *multi-parametric Quadratic Program*, short *mp-QP*. By solving this mp-QP problem, one obtains a solution function $\mathbf{U}_{opt} = f(\mathbf{x}(t))$; hence, the MPC control law is explicitly available. In the following, it is described how this solution function can be obtained and which characteristics it has.

见约束条件 (9.24) 和 (9.25)。

使用表达式 (9.26) 进行替换, 即可得到使成本函数 (9.18) 最小化的优化问题, 可表示为方程 (9.27), 满足约束条件 (9.28), 以及方程 (9.29) 和 (9.30)。

矩阵 \mathbf{H} , \mathbf{F} , \mathbf{Y} , \mathbf{G} , \mathbf{W} 和 \mathbf{E} 可根据方程 (9.18) 中的 \mathbf{Q} , \mathbf{R} 以及初始的优化准则求得。由于仅需要表示控制变量的向量 \mathbf{U} 作为最终的优化结果, 所以方程 (9.27) 中含有 \mathbf{Y} 的项可以忽略。

方程 (9.27) 所表示的优化问题是所谓的二次规划 (QP)。由于 J' 取决于系统的当前状态 $\mathbf{x}(t)$, 所以需要在每一个采样周期对 QP 进行重新求解。尽管目前已存在多种适合于 QP 问题的快速有效求解算法, 但对于控制技术领域应用场合来说, $\mathbf{u}(t)$ 的在线计算也是难以实现的, 尤其是驱动技术领域需要快速响应的场合下。因此, 经典 MPC 方法的应用被局限在具有低采样频率或低复杂性的场合。

在文献 [11] 中, 学者 Bemporad 和 Morari 等提出了一种适用于线性时不变系统的新型方法, 该方法将优化问题的求解转移至离线方式。这一策略随后被扩展至分段仿射系统和混合系统 [12]。基本思想是将状态向量 $\mathbf{x}(t)$ 看作参数向量, 从而方程 (9.27) 表示优化问题则转变为多参数二次规划, 简称 mp-QP。通过求解 mp-QP 问题, 即可得到表示最优解的函数表达式 $\mathbf{U}_{opt} = f(\mathbf{x}(t))$ 。如此, 即以显式方式得到了 MPC 控制律。随后将讲述如何获得上述最优解函数以及该方法所具有的特点。

Zuerst definiert man eine Hilfsgröße

$$\mathbf{z} = \mathbf{U} + \mathbf{H}^{-1}\mathbf{F}^T\mathbf{x}(t) \quad \mathbf{z} \in \mathbb{R}^s \quad (9.31)$$

Durch quadratische Ergänzung wird (9.27) umgeformt zu

$$J'_z(\mathbf{x}(t)) = \min_{\mathbf{z}} \left\{ \frac{1}{2} \mathbf{z}^T \mathbf{H} \mathbf{z} \right\} \quad (9.32)$$

unter der Randbedingung

$$\mathbf{G} \mathbf{z} \leq \mathbf{W} + \mathbf{S} \mathbf{x}(t) \quad (9.33)$$

mit

$$\mathbf{S} = \mathbf{E} + \mathbf{G} \mathbf{H}^{-1} \mathbf{F}^T \quad (9.34)$$

$$J'_z(\mathbf{x}(t)) = J'(\mathbf{x}(t)) - \frac{1}{2} \mathbf{x}^T (\mathbf{Y} - \mathbf{F} \mathbf{H}^{-1} \mathbf{F}^T) \mathbf{x}(t) \quad (9.35)$$

Dadurch erscheint der als Parameter betrachtete Zustandsvektor $\mathbf{x}(t)$ nur noch in den Beschränkungen.

Um nun mit der Lösung des mp-QP-Problems zu beginnen, wird ein Startvektor \mathbf{x}_0 innerhalb des möglichen Zustandsraums benötigt, für den das obige Optimierungsproblem lösbar ist. Wie man zu so einem Startpunkt gelangt, wird z. B. von Bemporad et al. beschrieben [13]. Mit Hilfe des so ermittelten \mathbf{x}_0 kann nun eine Optimallösung \mathbf{z}_0 von (9.32) ermittelt werden. Da die Matrix \mathbf{H} positiv definit ist, ist die Lösung eindeutig und man erhält einen Satz aktiver Beschränkungen $\mathbf{G} \mathbf{z}_0 = \mathbf{W} + \mathbf{S} \mathbf{x}_0$. Diese Beschränkungen bilden eine Region \mathcal{P}_0 . Der Optimalwert von \mathbf{z} und der dazugehörige Vektor des Lagrange-Multiplikators λ sind nun eindeutig definierte affine Funktionen von \mathbf{x} über \mathcal{P}_0 ; dies lässt sich mit Hilfe der Karush-Kuhn-Tucker Optimalitätsbedingungen beweisen. Genauso ist beweisbar, dass \mathcal{P}_0 einen Polyeder bzw. ein Polytop im Zustandsraum darstellt.

Ist die erste Region \mathcal{P}_0 festgelegt, müssen der übrige Zustandsraum untersucht und weitere Regionen kreierte werden. Eine mögliche Vorgehensweise wird ebenfalls von Bemporad et al. in [13] erläutert. Da für alle Regionen die für \mathcal{P}_0 bewiesenen Sachverhalte genauso gelten, handelt es sich bei diesen folglich gleichfalls um Polytope, in denen jeweils ein affines Gesetz zur Bestimmung eines optimalen \mathbf{z} gilt. Da auch die Verbindung zwischen \mathbf{z} und \mathbf{U} affin ist (s. Gl. (9.31)), so existiert nun eine stückweise affine Beschreibung $\mathbf{U}_{opt} = f(\mathbf{x}(t))$.

First, an auxiliary variable is defined, see equation (9.31). By quadratic complement, (9.27) is transformed to equation (9.32). Under the constraint (9.33) with equation (9.34) and (9.35). Thus, the state vector $\mathbf{x}(t)$, considered as a parameter, only appears in the constraints.

In order to solve the mp-QP problem, an initial vector \mathbf{x}_0 within the possible state space, for which the upper optimization problem can be solved, has to be found now. How such a start point can be found is described e.g. by Bemporad et al. [13]. With the help of \mathbf{x}_0 , obtained as described above, an optimum solution \mathbf{z}_0 of (9.32) can be determined. Since the Matrix \mathbf{H} is positive definite, the solution is unique and a set of active constraints $\mathbf{G}\mathbf{z}_0 = \mathbf{W} + \mathbf{S}\mathbf{x}_0$ is obtained. These constraints form a region \mathcal{P}_0 . The optimum value of \mathbf{z} and the corresponding vector of the lagrange multiplier λ are now uniquely defined affine functions of \mathbf{x} over \mathcal{P}_0 ; this can be proved with the help of the Karush-Kuhn-Tucker conditions. In the same way, it can be proved that \mathcal{P}_0 forms a polyhedron or a polytope in state space.

If the first region \mathcal{P}_0 is defined, the remaining state space has to be analyzed and further regions have to be created. A possible approach is also described by Bemporad et al. in [13]. Hence, as the issues proven for \mathcal{P}_0 are also valid for all regions, these are also polytopes, each with an affine law for determining an optimum \mathbf{z} . As the relationship between \mathbf{z} and \mathbf{U} is also affine (see equation (9.31)), a piecewise affine description $\mathbf{U}_{opt} = f(\mathbf{x}(t))$ exists now.

首先, 定义一个辅助变量, 见方程 (9.31)。使用配方法可将方程 (9.27) 变换为方程 (9.32), 满足约束条件 (9.33), 以及方程 (9.34) 和 (9.35)。从而, 被看作为参数的状态向量 $\mathbf{x}(t)$ 仅出现在约束条件。

为了能够开始求解 mp-QP 问题, 需要首先在可能的状态空间内确定一个初始向量 \mathbf{x}_0 , 并且它所对应的优化问题是有解的。那么, 如何获得这样一个初始点呢? 学者 Bemporad 等在文献 [13] 中进行了描述。借助于所得到的 \mathbf{x}_0 , 即可确定方程 (9.32) 的最优解 \mathbf{z}_0 。由于矩阵 \mathbf{H} 正定, 那么仅存在一个唯一的解, 并且可以得到一组有效的约束条件 $\mathbf{G}\mathbf{z}_0 = \mathbf{W} + \mathbf{S}\mathbf{x}_0$ 。从几何角度来看, 上述约束条件构成一个区域 \mathcal{P}_0 , 且 \mathbf{z} 的最优值以及拉格朗日乘数 λ 对应的向量也是由 \mathbf{x} 在以 \mathcal{P}_0 为定义域中定义的仿射函数唯一确定的, 这一点可以借助于 KKT 优化条件得到证明。同样可以证明, \mathcal{P}_0 在状态空间中构成一个多面体或多胞形。

如果已确定了第一个区域 \mathcal{P}_0 , 那么必须分析剩余的状态空间并确定其他的区域。一种可行的处理方法同样由学者 Bemporad 等在文献 [13] 中进行了阐述。上述针对 \mathcal{P}_0 的证明方式也同样适用于其他所有的区域, 它们也同样是多胞形, 并且同时满足用于确定最优 \mathbf{z} 的仿射规则。由于 \mathbf{z} 和 \mathbf{U} 之间存在仿射关系, 见方程 (9.31), 从而存在一个分段仿射表达式 $\mathbf{U}_{opt} = f(\mathbf{x}(t))$ 。

Für den mathematisch weniger interessierten Anwender ist es jedoch nicht notwendig, sich mit den oben erwähnten und teilweise auch ausgeführten Algorithmen und mathematischen Beweisen im Detail auseinanderzusetzen. An der Eidgenössischen Technischen Hochschule (ETH) Zürich ist am Lehrstuhl von Prof. Morari eine Toolbox für MATLAB[®] entwickelt worden, welche die einfache Berechnung expliziter Lösungen für verschiedene LTI- oder PWA-Systeme mit unterschiedlichen Kostenfunktionen ermöglicht [73]. Abb. 9.9 zeigt die Ergebnisse einer Berechnung der expliziten Lösung für die Stromregelung einer Asynchronmaschine. Hierbei wurde eine 1-norm-Kostenfunktion mit einem Prädiktionshorizont von zwei Schritten gewählt. Zur einfacheren Bewertung der erhaltenen stückweise affinen Lösungsfunktion ist die Lösung für den Ursprung als fester Sollwert ermittelt worden. Wie unschwer zu erkennen ist, werden die drei Phasen a , b , und c stets so geschaltet, dass der Stromraumzeiger zum Ursprung hin bewegt wird. Im Ursprung selbst wird der Vektor [000] geschaltet, was bewirkt, dass der Istwert des Stromraumzeigers auf diesem Wert verbleibt.

Die Trajektorien eines geschlossenen Regelkreises mit einem Regler, der das PWA-Regelgesetz anwendet, zeigt Abb. 9.10. Als Beispiel werden zwei Trajektorien im Folgenden näher betrachtet.

Bei der Bahnkurve A wird zu Beginn, im Punkt A_1 , wie aus dem Vergleich mit Abb. 9.9 hervorgeht, der Schaltzustand [100] geschaltet, d. h. $c = 1$, $b = 0$ und $a = 0$. Dieser Spannungsraumzeiger bewegt den Systemzustand am stärksten in Richtung auf den Ursprung zu, auch wenn dieser nicht direkt erreicht werden kann. Im Punkt A_2 gelangt der Zustandsvektor \mathbf{x} an den Wert $i_{s\beta} = 0$; daher findet gemäß des vorbestimmten Regelgesetzes ein Schaltvorgang in Phase b statt (s. Abb. 9.9(b)). An die Maschine wird nun der Schaltzustand [110] angelegt, was den Zustandsvektor entlang der Achse $i_{s\beta} = 0$ direkt in den Ursprung führt. Dort werden die Halbbrücken b und c wieder auf den „unteren“ Halbleiter geschaltet, so dass ein Nullvektor auf die Maschine gegeben wird. Der Systemzustand bleibt folglich im Ursprung stehen.

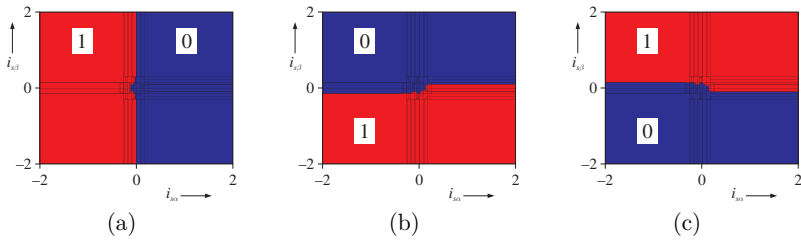
Betrachtet man die Trajektorie B , so wird zunächst im Punkt B_1 ebenfalls der Zustand [100] an die Maschine angelegt. Fünf Abtastschritte später erreicht der Systemzustand \mathbf{x} den Wert $i_{s\alpha} = 0$. Da es keinen vom Umrichter schaltbaren Spannungsvektor gibt, der \mathbf{x} von hier aus direkt entlang der Achse $i_{s\alpha} = 0$ in den Ursprung führt, so pendelt der Systemzustand zwischen den Bereichen für $a = 1$ und $a = 0$ (Abb. 9.9(a)). Entsprechend werden an die Maschine abwechselnd die Zustände [101] und [100] geschaltet. Dadurch wird der Maschinenzustand gezwungen, entlang der Trennungslinie zwischen den Bereichen mit dem Regelgesetz $\mathbf{u} = [101]$ und $\mathbf{u} = [100]$ zu gleiten. Das DMPC-Verfahren erhält hier den Charakter einer Sliding-Mode-Regelung [35], da der Systemzustand entlang synthetisch erzeugter Bahnkurven geführt wird. Auch in diesem Fall gelangt die geregelte Größe \mathbf{x} auf dem schnellstmöglichen Weg in den Ursprung.

Indeed, for users less interested in mathematics, it is not necessary to deal in detail with the algorithms and mathematical proofs mentioned and also partly explained above. At the Swiss Federal Institute of Technology (ETH) Zurich at Prof. Morari's department, a toolbox for MATLAB was developed with which calculation of explicit solutions for different LTI and PWA systems with different cost functions can easily be done [73]. Figure 9.9 shows the results of a calculation of the explicit solution for current control of an induction machine. Here, a 1-norm cost function with a prediction horizon of two steps was selected. For an easier rating of the obtained piecewise affine solution function, the solution was determined for the origin as fixed reference value. As it can easily be seen, the three phases a , b , and c are always switched in such a way that the current space vector is moved to the origin. In the origin itself the vector $[0, 0, 0]$ is selected, which leads to the actual value of the current space vector remaining on this point.

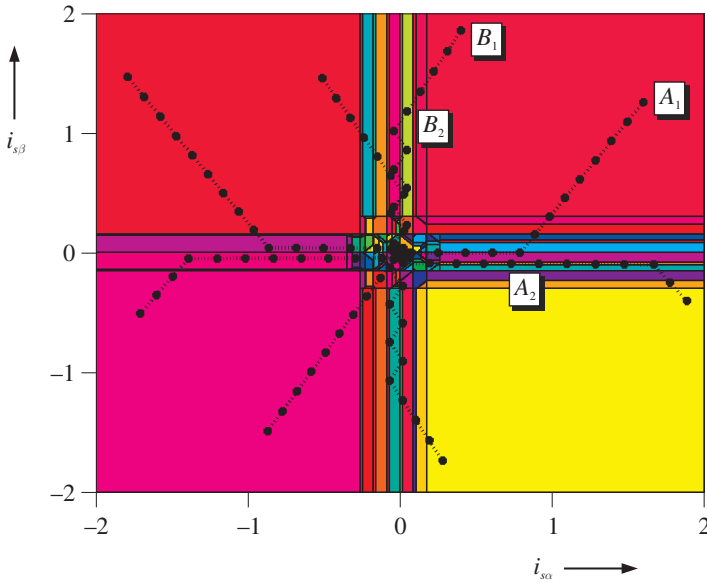
The trajectories of a closed loop control using a controller using the PWA control law is shown in Figure 9.10. As an example, a closer look is taken at two trajectories in the following.

Considering trajectory A at the beginning, at point A_1 , as it can be obtained from a comparison with Figure 9.9, the state $[1, 0, 0]$ is switched, i. e. $c = 1$, $b = 0$ and $a = 0$. This voltage space vector moves the system state in the fastest way towards the origin although the origin cannot be reached directly. At point A_2 the state vector \mathbf{x} reaches the value $i_{s\beta} = 0$; hence, according to the predetermined control law, a switching operation takes place in phase b (see Figure 9.9(b)). Now, the switching state $[1, 1, 0]$ is applied to the machine which moves the state vector directly into the origin along the axis $i_{s\beta} = 0$. Then, the half bridges b and c are again switched back to the "lower" semiconductors, leading to a zero vector applied to the machine. Hence, the system state will remain in the origin.

If trajectory B is considered, first of all, in point B_1 , the state $[1, 0, 0]$ is applied to the machine, too. Five sampling cycles later, the system state reaches the value $i_{s\alpha} = 0$. Since there is no voltage vector being switchable by the inverter that can move \mathbf{x} directly into the origin along the axis $i_{s\alpha} = 0$, the system state is oscillating between the sectors for $a = 1$ and $a = 0$ (Figure 9.9(a)). According to that, the states $[1, 0, 1]$ and $[1, 0, 0]$ are alternately applied to the machine. Thus, the machine state is forced to slide along the boundary between the sectors with the control law $\mathbf{u} = [1, 0, 1]$ and $\mathbf{u} = [1, 0, 0]$. Here the DMPC method gains the character of sliding mode control [35], because the system state is moved along synthetically created trajectories. In this case, the controlled variable \mathbf{x} will be moved into the origin as fast as possible.



9.9: Explizite Lösung für drei Halbbrücken: (a) Phase a, (b) Phase b, (c) Phase c;
 Explicit solution for three half bridges: (a) Phase a, (b) Phase b, (c) Phase c;
 三个半桥的显式解: (a) a 相, (b) b 相, (c) c 相



9.10: Trajektorien der expliziten Lösung/Trajectories of the explicit solution/显式解的
 轨迹

对于数学不感兴趣的读者不必深入研究上面所提及的算法以及相关数学证明。苏黎世理工大学 Morari 教授所负责的研究所开发出了一个特定的 MATLAB 工具箱，它可以简单地实现具有不同成本函数的不同 LTI 系统或 PWA 系统的显式求解 [73]。图 9.9 给出了实现异步电机电流控制的显式解的计算结果，它是基于控制步长为两步时的 1 范数成本函数所得。为了更为简单地评价所得到的分段仿射解函数，我们以原点作为固定的参考点进行求解。不难看出， a 、 b 和 c 三相不断地切换状态，以使得电流空间矢量向原点方向移动。在原点处所选择的矢量为 $[0, 0, 0]$ ，其作用是使得电流空间矢量的实际值维持在原点处。

图 9.10 给出了使用 PWA 控制律的控制器所得到的控制闭环的轨迹。作为示例，接下来详细探讨其中的两条轨迹。

对于轨迹 A 来说，结合图 9.9 可知，在起始点 A_1 处所应用的开关状态为 $[1, 0, 0]$ ，也即 $c = 1$ ， $b = 0$ 和 $a = 0$ 。这一电压空间矢量可使得系统状态以最快的速度向原点移动，即使它无法直接指向原点，其方向可参见第 239 页图 9.3 中的开关状态所对应的电压空间矢量的方向。在 A_2 点处，状态向量 \mathbf{x} 的 β 轴分量 $i_{s\beta} = 0$ ，基于已确定的控制律可知，此时 b 相状态需发生变化，即由 0 变为 1，见图 9.9(b)。从而，施加给电机的开关状态为 $[1, 1, 0]$ ，它可以使得状态向量沿着 $i_{s\beta} = 0$ 直接向原点移动。在原点处，半桥 b 和 c 将导通其下桥臂，此时零矢量 $[0, 0, 0]$ 施加给电机，从而系统状态将维持在原点处。

紧接着，我们探讨轨迹 B，在起始点 B_1 处，施加给电机的开关状态为 $[1, 0, 0]$ ，经过大约 4 个采样周期后，系统状态向量 \mathbf{x} 的 α 轴分量到达 $i_{s\alpha} = 0$ 。由于逆变器并不存在一个可使得 \mathbf{x} 沿着 $i_{s\alpha} = 0$ 方向直接向原点移动的电压矢量，从而系统状态将在 $a = 1$ 和 $a = 0$ 所在的区域内振荡。相应地，施加给电机的状态则在 $[1, 0, 1]$ 和 $[1, 0, 0]$ 之间不断地切换。从而，电机状态将被迫沿着控制律 $\mathbf{u} = [1, 0, 1]$ 和 $\mathbf{u} = [1, 0, 0]$ 所在区域的分界线向原点滑动。基于上述分析可知，此时的 DMPC 方法具有了滑模控制的特点 [35]，因为系统沿着一个合成的轨迹滑动。在这种情况下，被控变量 \mathbf{x} 也是以最快的路径到达原点。

Die ermittelte explizite Lösung besteht aus 217 Polytopen (Regionen), in denen jeweils ein stückweise affines Regelgesetz gilt. Auf den ersten Blick scheint dies verwunderlich, da es bei einem Zweipunktwechselrichter nur acht verschiedene Schaltzustände gibt und entsprechend für den nächsten Schritt nur acht verschiedene Regelgesetze existieren können. Es ist jedoch zu beachten, dass bei der Berechnung der expliziten Lösung der Lösungsvektor \mathbf{U} über den gesamten Prädiktionshorizont ermittelt wird. Folglich sind in dem hier behandelten Fall schon $8^2 = 64$ verschiedene Regelgesetze möglich. Da darüberhinaus auch noch die Bedingung existiert, dass alle Regionen konvexe Polytope sein müssen, kann die Gesamtzahl der ermittelten Regionen deutlich über der theoretisch möglichen Maximalzahl von Regelgesetzen liegen.

Tracking

Für den praktischen Anwendungsfall einer Regelung ist die bisher beschriebene Optimierung auf einen festen Sollwert selten brauchbar, da sich die Sollwerte im praktischen Betrieb dynamisch ändern. Entsprechend muss das Streckenmodell dahingehend geändert werden, dass nicht der Systemzustand \mathbf{x} selbst, sondern die Regelabweichung $\mathbf{x} - \mathbf{w}$ zu Null geregelt wird. Dieser Vorgang, bei dem der Istwert einem „freien“ Sollwert folgen soll, wird *Tracking* genannt. Dazu wird der Zustandsvektor \mathbf{x} um die Stellgröße \mathbf{u} und den Sollwert \mathbf{w} erweitert und als Eingangsgröße die Veränderung der Stellgröße, also $\Delta\mathbf{u}$ verwendet. Die Modellgleichung (9.15) wird dadurch zu:

$$\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{u}(t) \\ \mathbf{w}(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t-1) \\ \mathbf{w}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} \cdot \Delta\mathbf{u}(t) \quad (9.36)$$

Es ergeben sich damit die neuen Modellgleichungen

$$\mathbf{x}_t(t+1) = \mathbf{A}_t \mathbf{x}_t(t) + \mathbf{B}_t \mathbf{u}_t(t) \quad (9.37)$$

$$\mathbf{y}_t(t) = \mathbf{C}_t \mathbf{x}_t(t) + \mathbf{D}_t \mathbf{u}_t(t) \quad (9.38)$$

mit

$$\mathbf{x}_t(t) = \begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{u}(t) \\ \mathbf{w}(t+1) \end{bmatrix}, \quad \mathbf{u}_t(t) = \Delta\mathbf{u}(t), \quad (9.39)$$

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{B}_t = \begin{bmatrix} \mathbf{B} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \quad (9.40)$$

$$\mathbf{C}_t = \mathbf{I}, \quad \mathbf{D}_t = \mathbf{0} \quad (9.41)$$

The determined explicit solution consists of 217 polytopes (regions), each with a piecewise affine control law being valid. At first view, this seems to be remarkable since a two-level inverter has only eight different switching states and hence for the next step only eight different control laws can exist. However, it has to be noted that the solution vector \mathbf{U} is determined over the whole prediction horizon when the explicit solution is calculated. Hence, in the case treated here already $8^2 = 64$ different control laws are possible. Furthermore, the complete number of determined regions can be significantly higher than the maximum number of control laws being theoretically possible because of the constraint that all regions have to be convex polytopes.

Tracking

For the practical application of a control system, the optimization to a fixed reference value described so far can rarely be used because in practical operation, the reference values will change dynamically. Correspondingly, the plant model has to be changed so that not the system state \mathbf{x} itself, but the control error $\mathbf{x} - \mathbf{w}$ has to be controlled to become zero. This procedure in which the actual value has to follow a “free” reference value is called *tracking*. For that, the state vector \mathbf{x} is extended by the actuating variable \mathbf{u} and by the reference value \mathbf{w} , and as input variable the change of the values of the actuating variable, i. e. $\Delta\mathbf{u}$, is used. Consequently, the model equation (9.15) becomes to equation (9.36). Thus, the new model equations can be obtained, see equations (9.37) and (9.38), with equations (9.39) to (9.41).

所求得的显式解由 204 个多胞形（区域）组成，每一区域都有其相对应的分段仿射控制律。初看之下，这一结果着实让人惊讶，因为一个仅包含 8 个不同开关状态的两电平逆变器，应当对应 8 个不同的控制律才看似合理。但要清楚的是，在计算显式解时，解向量 \mathbf{U} 是在整个预测步长内确定的，本部分所讨论的情况则对应 $8^2 = 64$ 个不同的控制律。此外，由于还存在所有的区域必须为凸多胞体这一限制条件，那么实际得到的区域数量则会明显多于根据控制律确定的理论最大数量。

跟踪

对于控制工程实际应用来说，迄今为止所讨论的基于固定参考值条件下所描述的优化问题几乎很少使用，因为实际运行状态下的参考值大多是动态变化的。相应地，被控系统的模型也必须随之改变，此时控制的并非是系统状态 \mathbf{x} 本身，而是使控制偏差 $\mathbf{x} - \mathbf{w}$ 为零。这一实际值应跟随一个任意参考值的过程被称为跟踪。为此，状态向量 \mathbf{x} 需借助于额外的控制变量 \mathbf{u} 和参考变量 \mathbf{w} 进行扩展，并使用控制增量 $\Delta\mathbf{u}$ 作为输入变量。那么，模型方程 (9.15) 则变化为方程 (9.36)。从而，可以得到新的模型方程表达式，见方程 (9.37) 和 (9.38)，其中有方程 (9.39) 至 (9.41)。

Die aus Gleichung (9.18) bekannte Kostenfunktion

$$J = \sum_{j=N_1}^{N_2} \left(\mathbf{x}(t+j-1)^T \mathbf{Q} \mathbf{x}(t+j-1) + \mathbf{u}(t+j-1)^T \mathbf{R} \mathbf{u}(t+j-1) \right) \quad (9.42)$$

wird ebenfalls so modifiziert, dass sie die Gewichtung der Regelabweichung $\mathbf{x}-\mathbf{w}$ umfasst:

$$J = \sum_{j=N_1}^{N_2} \left(\begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \\ \mathbf{w}(t+j) \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{Q} & \mathbf{0} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ -\mathbf{Q} & \mathbf{0} & \mathbf{Q} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \\ \mathbf{w}(t+j) \end{bmatrix} + \Delta \mathbf{u}(t+j-1)^T \cdot \mathbf{R}_\Delta \cdot \Delta \mathbf{u}(t+j-1) \right) \quad (9.43)$$

Der neue Bewertungsfaktor \mathbf{R}_Δ bewertet dabei die Stellgrößenänderungen $\Delta \mathbf{u}$. Es existiert nun passend zu dem neuen Streckenmodell (9.37)-(9.38) eine Bewertungsfunktion

$$J = \sum_{j=N_1}^{N_2} \left(\mathbf{x}_t(t+j-1)^T \mathbf{Q}_t \mathbf{x}_t(t+j-1) + \mathbf{u}_t(t+j-1)^T \mathbf{R}_t \mathbf{u}_t(t+j-1) \right) \quad (9.44)$$

mit

$$\mathbf{Q}_t = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ -\mathbf{Q} & \mathbf{0} & \mathbf{Q} \end{bmatrix}, \quad \mathbf{R}_t = \mathbf{R}_\Delta \quad (9.45)$$

Werden statt des in den Gln. (9.15) und (9.16) angegeben Modells und der Kostenfunktion (9.18) das Modell aus (9.37) und (9.38) sowie die Kostenfunktion (9.44) verwendet und dann zur Ermittlung optimaler Lösungen genauso weiter vorgegangen wie bisher beschrieben, so ergibt sich eine explizite Lösung des MPC-Problems unter Berücksichtigung von Sollwerten. Da die Dimension des Zustandsvektors durch das Einschließen von Sollwerten und vorangegangenen Stellgrößen zunimmt, erhöht sich auch die Dimension des Lösungsraums entsprechend. Die Lösung eines Tracking-Problems ist daher stets aufwändiger als diejenige, die quasi einen Nullvektor als konstanten Sollwert annimmt.

The cost function known from equation (9.18), see equation (9.42), is also modified so that it contains the evaluation of the control error $\mathbf{x} - \mathbf{w}$:

$$J = \sum_{j=N_1}^{N_2} \left(\begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \\ \mathbf{w}(t+j) \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{Q} & \mathbf{0} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ -\mathbf{Q} & \mathbf{0} & \mathbf{Q} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \\ \mathbf{w}(t+j) \end{bmatrix} + \Delta \mathbf{u}(t+j-1)^T \cdot \mathbf{R}_\Delta \cdot \Delta \mathbf{u}(t+j-1) \right) \quad (9.43)$$

The new weighting factor \mathbf{R}_Δ thereby weights the change of the values of the actuating variables $\Delta \mathbf{u}$. Now, adequate to the new plant model (9.37)-(9.38), a cost function, see equation (9.44) with equation (9.45) exists.

If, instead of the model given in the equations (9.15) and (9.16) and the cost function (9.18), the model in (9.37) and (9.38) and the cost function (9.44) is used and one proceeds as described so far in order to determine optimal solutions, an explicit solution of the MPC problem under consideration of reference values is possible. Due to the fact that the dimension of the state vector enlarges when reference values and former values of the actuating variables are considered, the dimension of the solution space will be enlarged accordingly. Hence, the solution of a tracking problem is always more complex than a solution which takes a zero vector as constant reference value.

根据方程 (9.18) 可知的成本函数 (9.42) 需要进行相应地修正, 以使其包含控制偏差 $\mathbf{x} - \mathbf{w}$ 这一惩罚项:

$$J = \sum_{j=N_1}^{N_2} \left(\begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \\ \mathbf{w}(t+j) \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{Q} & \mathbf{0} & -\mathbf{Q} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ -\mathbf{Q} & \mathbf{0} & \mathbf{Q} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \\ \mathbf{w}(t+j) \end{bmatrix} + \Delta \mathbf{u}(t+j-1)^T \cdot \mathbf{R}_\Delta \cdot \Delta \mathbf{u}(t+j-1) \right) \quad (9.43)$$

其中, 新的权重系数 \mathbf{R}_Δ 用于惩罚控制增量 $\Delta \mathbf{u}$ 。存在一个适用于方程 (9.37) - (9.38) 所表示的被控系统模型的成本函数, 其表达式为方程 (9.44), 其中方程 (9.45)。

如果使用方程 (9.37) 和 (9.38) 所表示的模型和成本函数 (9.44) 来代替方程 (9.15) 和 (9.16) 所表示的模型和成本函数 (9.18), 那么即可使用前面所述的方式来求得最优解, 从而可以得到在考虑了参考值情况下的 MPC 问题的显式解。由于状态向量中引入了参考变量和前一时刻的控制变量, 其维度则同时增加, 那么解空间的维度也相应增加。由此可知, 跟踪问题的解总是要比以零向量作为恒定参考值的情况下更为复杂。

Verzögerungen

Digitale Regelungen verursachen üblicherweise eine Verzögerung von einem Abtastschritt zwischen dem Zeitpunkt, zu dem der Regler einen Sollwert ermittelt hat und dem Moment, zu dem dieser Sollwert tatsächlich an die Regelstrecke angelegt wird. MPC-Strategien, die auf einem Übertragungsfunktionsbasierten Modell der zu regelnden Strecke beruhen, lassen sich auf diesen Fall sehr gut anwenden, da die Verzögerung einfach wie in Kap. 6.1.1 auf Seite 75 erwähnt implementiert werden kann, indem das Polynom $B(z^{-1})$ bzw. die Polynommatrix $\mathbf{B}(z^{-1})$ des CARIMA-Modells mit z^{-1} multipliziert wird. Die Verwendung eines zustandsraumbasierten Prädiktivreglers schließt diese einfache Methode leider aus; es ist aber möglich, eine entsprechende Verzögerung durch Erweitern des Zustandsvektors in das Streckenmodell aufzunehmen. Dazu wird die Zustandsgleichung des Systems wie folgt modifiziert:

$$\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{u}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t-1) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \cdot \mathbf{u}(t) \quad (9.46)$$

Damit ergibt sich eine neue Modellbeschreibung:

$$\mathbf{x}_d(t+1) = \mathbf{A}_d \mathbf{x}_d(t) + \mathbf{B}_d \mathbf{u}_d(t) \quad (9.47)$$

$$\mathbf{y}_d(t) = \mathbf{C}_d \mathbf{x}_d(t) + \mathbf{D}_d \mathbf{u}_d(t) \quad (9.48)$$

mit

$$\mathbf{x}_d(t) = \begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{u}(t) \end{bmatrix}, \quad \mathbf{u}_d(t) = \mathbf{u}(t), \quad (9.49)$$

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}, \quad (9.50)$$

$$\mathbf{C}_d = \mathbf{I}, \quad \mathbf{D}_d = \mathbf{0} \quad (9.51)$$

Da im Zustandsgrößenvektor \mathbf{x}_d nun auch die Stellgröße \mathbf{u} enthalten ist, in der Kostenfunktion jedoch nur die Zustandsgröße selbst bewertet werden soll, muss Gl. (9.18) wiederum entsprechend angepasst werden. Es gilt in diesem Fall:

$$J = \sum_{j=N_1}^{N_2} \left(\begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t+j) \\ \mathbf{u}(t+j-1) \end{bmatrix} + \mathbf{u}(t+j-1)^T \cdot \mathbf{R} \cdot \mathbf{u}(t+j-1) \right) \quad (9.52)$$

In diesem Fall wird nicht die Stellgrößenänderung, sondern wieder die Stellgröße *selbst* bewertet. Die neue Kostenfunktion lässt sich demnach schreiben:

$$J = \sum_{j=N_1}^{N_2} \left(\mathbf{x}_d(t+j-1)^T \mathbf{Q}_d \mathbf{x}_d(t+j-1) + \mathbf{u}_d(t+j-1)^T \mathbf{R}_d \mathbf{u}_d(t+j-1) \right) \quad (9.53)$$

Delays

Normally, digital controls cause a delay of one sampling cycle between the point of time at which the controller has determined a reference value and the point of time at which this reference value is actually applied to the plant. MPC strategies based on a transfer function based model of the plant can easily be adopted in this case, because the delay can be implemented easily by multiplying the polynomial $B(z^{-1})$ or the polynomial matrix $\mathbf{B}(z^{-1})$ of the CARIMA model with z^{-1} , as described in Chapter 6.1.1 on page 75. The use of a state space based predictive controller unfortunately excludes this simple method; however, it is possible to add an appropriate delay to the plant model by extending the state vector. Thus, the state equation of the system is modified as follows, see equation (9.46). Consequently, a new representation of the model can be obtained, see equations (9.47) and (9.48), with equations (9.49) to (9.51).

The vector of state variables \mathbf{x}_d does now also contain the actuating variable \mathbf{u} . In the cost function, however, only the state variables themselves should be evaluated. Thus, equation (9.18) has to be adjusted again. In this case it can be applied, see equation (9.52). Here not the *change* of the values of the actuating variable, but again the actuating variable itself is evaluated. Thus, the new cost function can be written as equation (9.53).

滞后

在实现数字控制时，通常会在控制器确定参考值的时刻与该参考值真正施加到被控系统的时刻之间存在一个采样周期的滞后。基于被控系统的传递函数模型实现的 MPC 方法可以很好地应对上述情形，因为滞后因素可以很容易地以 CARIMA 模型中的多项式 $B(z^{-1})$ 或多项式矩阵 $\mathbf{B}(z^{-1})$ 乘以 z^{-1} 的方式考虑进来，如章节 6.1.1 第 75 页所述。虽说基于状态空间模型的预测控制器无法使用上述简单方法，但是它可以通过扩展状态向量的方式将滞后因素添加到被控系统模型。为此，需要对系统的状态方程进行修正，见方程 (9.46)。从而，可以得到新的模型描述方式，见方程 (9.47) 和 (9.48)，其中，有方程 (9.49) 至 (9.51)。

由于此时的状态向量 \mathbf{x}_d 中也包含控制变量 \mathbf{u} ，所以成本函数也应当惩罚状态向量本身。为此，需要对方程 (9.18) 进行再次调整，见方程 (9.52)。在这一情况下，成本函数惩罚的并非是控制增量，而再次是控制变量本身。新的成本函数可以写为方程 (9.53)。

mit

$$\mathbf{Q}_d = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{R}_d = \mathbf{R} \quad (9.54)$$

Verwendet man die Zustandsgleichungen (9.47) und (9.48) und die dazugehörige Bewertungsfunktion (9.53) und löst dann das Problem mit Hilfe von mp-QP, so erhält man eine explizite Lösung des Optimierungsproblems für den Fall, dass das zu regelnde System ein Verzögerung von einem Abtastschritt besitzt.

Verzögerung mit Tracking

Es ist davon auszugehen, dass Antriebsregelungen im Allgemeinen sowohl eine Verzögerung aufweisen, als auch, dass der Systemzustand nicht in den Ursprung geregelt werden, sondern einem Istwert folgen soll. Daher sind bei einer praktischen Regelung für elektrische Antriebe die beiden oben genannten Verfahren zu kombinieren. Erzielt wird dies durch eine geringfügige, weitere Modifikation von (9.37),

$$\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{u}(t) \\ \mathbf{w}(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t-1) \\ \mathbf{w}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} \cdot \Delta \mathbf{u}(t) \quad (9.55)$$

wodurch sich die Modellgleichungen

$$\mathbf{x}_{t,d}(t+1) = \mathbf{A}_{t,d}\mathbf{x}_{t,d}(t) + \mathbf{B}_{t,d}\mathbf{u}_{t,d}(t) \quad (9.56)$$

$$\mathbf{y}_{t,d}(t) = \mathbf{C}_{t,d}\mathbf{x}_{t,d}(t) + \mathbf{D}_{t,d}\mathbf{u}_{t,d}(t) \quad (9.57)$$

mit

$$\mathbf{x}_{t,d}(t) = \begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{u}(t) \\ \mathbf{w}(t+1) \end{bmatrix}, \quad \mathbf{u}_{t,d}(t) = \Delta \mathbf{u}(t), \quad (9.58)$$

$$\mathbf{A}_{t,d} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{B}_{t,d} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \quad (9.59)$$

$$\mathbf{C}_{t,d} = \mathbf{I}, \quad \mathbf{D}_{t,d} = \mathbf{0} \quad (9.60)$$

ergeben. Die Bewertungsfunktion (9.44) braucht nicht verändert werden, d. h. $\mathbf{Q}_{t,d} = \mathbf{Q}_t$ und $\mathbf{R}_{t,d} = \mathbf{R}_t$, wobei wiederum der Bewertungsfaktor \mathbf{R}_Δ die Stellgrößenänderung gewichtet.

Wie die obigen Gleichungen zeigen, ist für die Einbindung einer Verzögerung von einem Abtastschritt in eine Regelung mit Tracking keine Erweiterung des Zustandsvektors gegenüber der reinen Tracking-Variante \mathbf{x}_t notwendig. Folglich lässt sich die Berücksichtigung der Verzögerung in diesem Fall ohne weiteren Mehraufwand in das Streckenmodell integrieren.

with equation (9.54). If the state equations (9.47) and (9.48) and the corresponding cost function (9.53) are used and the problem is solved via mp-QP, an explicit solution of the optimization problem is obtained for the case that the system to be controlled has a delay of one sampling cycle.

Delay with tracking

It can be assumed that drive controls commonly have a delay as well as that the system state should not be driven into the origin, but it should follow a reference value. Thus, for an applicable control for electrical drives both methods mentioned above have to be combined. This can be achieved by a further slight modification of (9.37), see equation (9.55), which leads to the model equations (9.56) and (9.57), with equations (9.58) to (9.60).

The evaluation function (9.44) does not have to be modified, i.e. $\mathbf{Q}_{t,d} = \mathbf{Q}_t$ and $\mathbf{R}_{t,d} = \mathbf{R}_t$, in which again the weighting factor \mathbf{R}_Δ weights the change of the value the actuating variables.

As the above equations show, for the integration of a delay of one sampling cycle into a control with tracking no extension of the state vector compared to the mere tracking version \mathbf{x}_t is necessary. Hence, the consideration of the delay can, in this case, be integrated into the plant model without any further effort.

其中，有方程 (9.54)。如果使用状态方程 (9.47) 和 (9.48) 以及对应的成本函数 (9.53)，并借助于 mp-QP 求解这一问题，那么即可得到优化问题的显式解，且被控系统包含了一个采样周期的滞后。

滞后结合跟踪

可以假定驱动控制系统通常包含滞后因素，而且也并非是将系统状态控制到原点，而是跟踪一个变化的参考值。因此，对于电力驱动领域实际的控制场合来说，需结合使用前面介绍的两种方法。为此，需进一步对方程 (9.37) 作出微小的修正，可得方程 (9.55)，并由此可得模型方程 (9.56) 和 (9.57)，其中方程 (9.58) 至 (9.60)。

无需对成本函数 (9.44) 进行修正，也意味着 $\mathbf{Q}_{t,d} = \mathbf{Q}_t$ 和 $\mathbf{R}_{t,d} = \mathbf{R}_t$ ，且权重系数 \mathbf{R}_Δ 惩罚的是控制增量，而非控制变量本身。

正如前面一系列的方程所示，结合了一个采样周期滞后的跟踪控制无需对状态向量进行扩展。从而，在上述情况下，将滞后因素考虑至被控系统模型无需付出额外的代价。

9.5.1 Standard-Algorithmus/Standard algorithm/标准算法

Hat man die explizite Lösung einer MPC-Regelung nach dem oben beschriebenen Verfahren berechnet, so muss man, um die Regelung praktisch zu realisieren, zuerst herausfinden, in welchem der berechneten Polytope der momentane Systemzustand \mathbf{x} liegt, um danach das entsprechende affine Regelgesetz anwenden zu können. Da die Polytope im Allgemeinen nicht sortiert sind, bleibt kein anderer Weg, als eine erschöpfende Suche über alle existierenden Polytope durchzuführen. Für jedes Polytop \mathcal{P}_i gilt für alle Systemzustände \mathbf{x} , die innerhalb dieses Polytops liegen, die Gleichung

$$\mathbf{H}_i \mathbf{x} \leq \mathbf{K}_i \quad (9.61)$$

Nun untersucht man der Reihe nach für alle ermittelten Polytope, ob Gl. (9.61) zutrifft. Wenn ja, so hat man das Polytop \mathcal{P}_j gefunden, welches \mathbf{x} beinhaltet. Das dazugehörige Regelgesetz ist dann anzuwenden.

Bedingt durch die vollständige Enumeration aller vorhandenen Polytope ist dieses Verfahren selbstverständlich nicht sehr effizient, weshalb sich für eine Antriebsregelung intelligentere Suchstrategien empfehlen.

9.5.2 Minimum-Time-Controller/Minimum Time Controller/ 最小时间控制器

Deutlich schneller als das Standard-Verfahren arbeitet der von Grieder/Morari vorgestellte sogenannte *Minimum-Time-Controller*. Dieser kann sowohl für lineare, zeitinvariante (LTI) [45] als auch für stückweise affine (PWA) oder hybride Systeme [46] verwendet werden. Ziel des Verfahrens ist, eine zwar suboptimale, dafür aber in der Ausführung weniger aufwändige Lösung des Optimierungsproblems zu bestimmen. Eine geringere Komplexität der expliziten Reglerstruktur erreicht man am einfachsten durch niedrige Prädiktionshorizonte und wenig Eingangsvariablen. Leider sind diese Größen i. A. von dem zu regelnden System abhängig und daher nicht frei wählbar. Der hier alternativ vorgestellte Weg besteht darin, einstufige Optimierungsprobleme mit sich veränderndem Zielgebiet \mathcal{T}_{set} iterativ zu lösen. Dazu wird jeweils folgende Optimeraufgabe gelöst:

$$J_1(\mathbf{x}(t)) = \min_{\mathbf{u}(t)} \left\{ \mathbf{x}(t+1)^T \mathbf{Q} \mathbf{x}(t+1) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) \right\} \quad (9.62)$$

9.5.1 Standard algorithm

If the explicit solution of MPC is calculated with the method described above, for a practical realization of the control one first has to find out in which one of the calculated polytopes the current system state \mathbf{x} lies in order to apply the corresponding affine control law afterwards. As the polytopes are usually not sorted, there is no other way than an exhaustive search over all existing polytopes. For every polytope \mathcal{P}_i for all system states

\mathbf{x} being within this polytope the equation (9.61) is valid. Then all determined polytopes have to be checked one after each other if equation (9.61) is valid. If this is the case, the polytope \mathcal{P}_j that contains \mathbf{x} is found. Then the corresponding control law has to be applied.

Due to the complete enumeration of all existent polytopes this method is, of course, not very efficient. Thus, more intelligent search strategies are recommended for drive control.

9.5.2 Minimum Time Controller

The so called *Minimum Time Controller* proposed by Grieder/Morari is considerably faster than the standard method. It can be used for linear time invariant (LTI) [45] as well as for piecewise affine (PWA) or hybrid systems [46]. The intention of this method is to determine an indeed sub-optimal, but in the execution less complex solution of the optimization task. A lower complexity of the explicit controller structure can in the easiest way be obtained with short prediction horizons and with a lesser number of input variables. Unfortunately, these variables are normally dependent of the system to be controlled and thus cannot be chosen freely. The alternative way proposed here consists of the iterative solution of single-step optimization problems with a varying target region \mathcal{T}_{set} . For that in each case the following optimization task is solved, see equation (9.62).

9.5.1 标准算法

如果已经根据前面描述的方法求出了 MPC 控制问题的显式解, 那么, 为了能够实际实现控制任务, 必须首先找出当前的系统状态 \mathbf{x} 究竟位于哪一多胞体, 然后才有可能使用所对应的仿射控制律。由于多胞体通常是无序的, 所以除了对所有多胞体进行穷举搜索的方式之外, 别无他法。对于任一多胞体 \mathcal{P}_i , 所有位于这一多胞体的系统状态满足不等式 (9.61)。随后, 逐一对所得到的多胞体进行考察, 检测是否满足不等式 (9.61)。如果满足, 那么则找到了包含 \mathbf{x} 的多胞体 \mathcal{P}_j , 然后即可应用其所属的控制律。

显然, 对所有多胞体进行完全遍历的方式的效率非常低, 正因如此, 建议采用更为智能的搜索策略来实现实际的驱动控制任务。

9.5.2 最小时间控制器

学者 Grieder/Morari 提出的所谓最小时间控制器要明显快于上一小节介绍的标准方法, 它不仅可以用于线性时不变系统 [45], 也可用于分段仿射系统或混合系统 [46]。这一方法的目标是确定优化问题的一个实现起来更为容易的解, 尽管它有可能是一个次优解。最简单的方式是采取较小的预测步长和更少的输入变量以得到一个具有较低复杂度的显式控制器结构。遗憾的是, 上述变量通常取决于被控系统, 因此无法自由选择。此处提出的另一种方式是以迭代方式求解一个目标区域变化的单级优化问题。为此, 在每一情况下, 均需求解下述优化任务, 见方程 (9.62)。

unter Berücksichtigung von

$$\mathbf{x}(t+1) \in \mathcal{T}_{set} \subseteq \mathbf{X} \quad (9.63)$$

$$\mathbf{u}(t) \in \mathbf{U} \quad (9.64)$$

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (9.65)$$

$$\mathbf{Q} = \mathbf{Q}^T \succ 0 \quad (9.66)$$

$$\mathbf{R} = \mathbf{R}^T \succ 0 \quad (9.67)$$

Ziel dieses Verfahrens ist es, das zu regelnde System so schnell wie möglich in einen Bereich \mathcal{X}_{LQR} zu fahren, wo als Regelung eine optimale, unbeschränkte LQR-Regelung möglich ist, ohne dabei die tatsächlich vorhandenen Beschränkungen zu überschreiten. Diese Menge sei nun das erste Zielgebiet:

$$\mathcal{T}_{set} = (\mathcal{X}_f^1)^0 = \mathcal{X}_{LQR} \quad (9.68)$$

Innerhalb dieser Region gilt das LQR-Regelgesetz \mathbf{F}_{LQR} , so dass $\mathbf{F}_0^0 = \mathbf{F}_{LQR}$ und $\mathbf{G}_0^0 = 0$.

Um zu einem PWA-Regelgesetz für den gesamten Zustandsraum zu kommen, wird das auf (9.62) basierende mp-QP gelöst. Es ergibt sich eine Polytopstruktur

$$\mathcal{P}_k^{iter} = \{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{H}_k^{iter} \mathbf{x} \leq \mathbf{K}_k^{iter} \} \quad k = 1 \dots R \quad (9.69)$$

Dabei stellt *iter* den Iterationsschritt, beginnend bei 0 für \mathcal{X}_{LQR} dar; *R* ist die Anzahl der verschiedenen Polytope im aktuellen Iterationsschritt. Die zu den einzelnen Polytopen gehörenden Regelgesetze können durch die Gleichung

$$\mathbf{u}(t) = \mathbf{F}_k^{iter} \mathbf{x}(t) + \mathbf{G}_k^{iter} \quad (9.70)$$

ausgedrückt werden.

Iterativ kann nun die explizite Lösung für den gesamten Bereich berechnet werden. Dazu wird der Lösungsraum des letzten Iterationsschritts als neues Zielgebiet gewählt:

$$\mathcal{T}_{set} = (\mathcal{X}_f^1)^{iter} = \bigcup_{k=1}^R \mathcal{P}_k^{iter} \quad (9.71)$$

Über dieses Gebiet wird wiederum ein einstufiges mp-QP-Problem gelöst, wodurch sich der nächste Regelschritt *iter* + 1 ergibt. Dies wird so lange wiederholt, bis sich das Zielgebiet nicht mehr verändert. Nach jedem Iterationsschritt werden selbstverständlich alle Regionen und die dazugehörenden Regelgesetze abgespeichert.

under consideration of equations (9.63) to (9.67). The intention here is to drive the system to be controlled as fast as possible into a region \mathcal{X}_{LQR} in which an optimal unconstrained LQR control is possible without exceeding the constraints. This set shall be the first target region, see equation (9.68). Within this region the LQR control law \mathbf{F}_{LQR} is valid so that $\mathbf{F}_0^0 = \mathbf{F}_{LQR}$ and $\mathbf{G}_0^0 = 0$.

In order to obtain a PWA control law for the whole state space, the mp-QP based on (9.62) is solved. This results in the following polytope structure, see equation (9.69). In this case, $iter$ is the iteration step, beginning from 0 for \mathcal{X}_{LQR} ; R is the number of different polytopes in the current iteration step. The control laws belonging to the individual polytopes can be expressed by the equation (9.70).

Now, the explicit solution can be calculated iteratively for the whole region. For this purpose, the solution region of the last iteration step is selected as the new target region, see equation (9.71). For this region, a single-step mp-QP problem is solved again whereby the new control step $iter + 1$ results. This is repeated until the target region does not change anymore. After every iteration step, all regions and the corresponding control laws are saved.

考虑以下约束条件 (9.63) 至 (9.67)。这一方法的目标是使被控系统以尽可能快的速度到达一个区域 \mathcal{X}_{LQR} ，在该区域内才有可能实现最优的、无约束的 LQR 控制，同时不违背现有约束条件。假设第一个目标区域可表示为方程 (9.68)，且在该区域内满足 LQR 控制律，从而有 $\mathbf{F}_0^0 = \mathbf{F}_{LQR}$ 和 $\mathbf{G}_0^0 = 0$ 。

为了得到一个适用于整个状态空间的 PWA 控制律，需要求解一个基于方程 (9.62) 的 mp-QP 问题，得到具有下述形式的多胞体结构，见方程 (9.69)，其中， $iter$ 表示迭代步数，并以 0 作为 \mathcal{X}_{LQR} 的起始点； R 表示当前迭代步数下的不同多胞体的数量。每一多胞体所对应的控制律可表示为方程 (9.70)。

整个区域内的显式解可以通过迭代方式计算得到。为此，需选择前一迭代步数的解空间作为新的目标区域，见方程 (9.71)。在这一区域内需再次求解一个单级的 mp-QP 问题，从而得到了新的控制步数 $iter + 1$ 。不断重复上述过程，直至目标区域不再发生变化。当然，在每一迭代步数之后，需要存储所有的区域及相对应的控制律。

Durch die schrittweise Lösung des Optimierungsproblems erhält die explizite Lösung eine Baumstruktur; dies kann für die Konstruktion einer effizienten Suchstrategie zur online-Auswertung des PWA-Regelgesetzes ausgenutzt werden. Dazu wird zuerst die Region mit dem niedrigsten Iterationsschritt, in dem der momentane Systemzustand $\mathbf{x}(t)$ enthalten ist ermittelt, d. h.

$$iter_{min} = \min_{iter} iter, \text{ wobei } \mathbf{x}(t) \in (\mathcal{X}_f^1)^{iter} \quad (9.72)$$

Innerhalb dieser Region wird nun mittels erschöpfender Suche das aktive Polytop $\mathcal{P}_j^{iter_{min}}$ bestimmt, was deutlich schneller geschieht als bei der direkten erschöpfenden Suche, da die Anzahl der Polytope pro Region sehr gering ist. Sind nun $iter$ und j bekannt, kann aus der Gleichung

$$\mathbf{u}(t) = \mathbf{F}_j^{iter_{min}} \mathbf{x}(t) + \mathbf{G}_j^{iter_{min}} \quad (9.73)$$

die an die Strecke anzulegende Stellgröße bestimmt werden.

Hintergedanke des Minimum-Time-Controllers ist, dass weit weg vom Sollwert unabhängig von der Kostenfunktion die maximale Stellgröße immer auch optimal ist. Dadurch erhält man eine in zweifacher Weise vereinfachte Polytopstruktur: Zum einen wird die Anzahl der Regionen gegenüber dem Standard-Verfahren reduziert; zum anderen werden die einzelnen Polytope und Regelgesetze automatisch in einer Baumstruktur angeordnet. Durch die zweistufige Auswertung des expliziten Regelgesetzes wird die online notwendige Rechenzeit gegenüber dem Standard-Verfahren deutlich reduziert, allerdings auf Kosten einer nicht-optimalen Regelung in Bezug auf die Kostenfunktion. Statt dessen liegt nun das Gewicht darauf, den Systemzustand $\mathbf{x}(t)$ möglichst schnell in den Ursprung bzw. die Region \mathcal{X}_{LQR} zu regeln.

Because of the iterative solution of the optimization task, the explicit solution gets a tree-like structure; this can be used for the setup of an efficient search strategy for the online evaluation of the PWA control law. Therefore, at first the region with the lowest iteration step containing the current system state $\mathbf{x}(t)$ is determined, i. e. see equation (9.72).

Within this region now the active polytope $\mathcal{P}_j^{iter\ min}$ is determined via exhaustive search. This can be done much faster than with direct exhaustive search because the number of polytopes per region is very small. If *iter* and *j* are known, the value of the actuating variable to be applied to the plant can be determined with the equation (9.73).

The ulterior motive of the Minimum Time Controller is, that, independent from the cost function, far away from the reference value, the maximum value of the actuating variable is in any case also the optimal one. This leads to a polytope structure simplified in two ways: On the one hand, the number of regions compared to the standard method is reduced; on the other hand, the individual polytopes and control laws are automatically sorted in a tree-like structure. The two-stage evaluation of the explicit control law significantly reduces the necessary online computational time compared to the standard method, however, for the cost of a non-optimum control concerning the cost function. Instead, the strategy is focused to move the system state $\mathbf{x}(t)$ as fast as possible into the origin or the region \mathcal{X}_{LQR} , respectively.

通过优化问题的迭代求解方式可以得到一个具有树状结构的显式解，基于此可以构建一个可以对 PWA 控制律进行在线评判的有效搜索策略。为此，首先要确定在最小迭代步数下包含当前系统状态 $\mathbf{x}(t)$ 的区域，也即方程 (9.72)。

在上述区域内基于穷举搜索方式所确定的有效多胞体 $\mathcal{P}_j^{iter\ min}$ 要明显快于直接的穷举搜索方式，因为在每一区域内多胞体数量是很有限的。如果目前 *iter* 和 *j* 是已知的，那么可基于方程 (9.73) 确定施加给被控系统的控制变量。

最优控制器所隐含的基本思想：在被控变量远未达到参考值时，始终选择最大的控制变量作为最优解，其选取与成本函数无关。如此即可得到一个以两种方式简化的多胞体结构：一是区域的数量相比于标准方法来说变少了；二是每一多胞体和相应的控制律自动被排序为树状结构。相较于标准方法，显式控制律的两级评判方式所需的计算时间明显减小。当然，计算时间的减少是以成本函数评判方式下的非最优控制为代价。相反，当前的重心是以最快的速度将系统状态 $\mathbf{x}(t)$ 控制到原点或区域 \mathcal{X}_{LQR} 。

9.5.3 Binärer Suchbaum/Binary search tree/二叉搜索树

Ein anderes Verfahren zur Verkürzung der online-Auswertung versucht, die aus dem Standard-Verfahren gewonnene Polytopstruktur in einen binären Suchbaum zu überführen. Tøndel, Johansen und Bemporad haben in [117] eine solche Strategie vorgestellt. Dazu werden zuerst sämtliche Hyperebenen, also die Grenzflächen aller Polytope, betrachtet. Die Gesamtanzahl der Hyperebenen sei L , und sie werden jeweils durch die Gleichung $\mathbf{a}_j^T \mathbf{x} = \mathbf{b}_j$, $j = 1 \dots L$ beschrieben. Mit Hilfe dieser Formel wird eine Beschreibungsfunktion $d_j(\mathbf{x})$ wie folgt definiert.

$$d_j(\mathbf{x}) = \mathbf{a}_j^T \mathbf{x} - \mathbf{b}_j \quad (9.74)$$

Durch Auswerten des Vorzeichens von $d_j(\mathbf{x})$ kann für jeden Punkt im Zustandsraum angegeben werden, ob er sich „oberhalb“ oder „unterhalb“ der entsprechenden Hyperebene j befindet². Folglich lässt sich durch eine sogenannte Indexmenge \mathbf{J} ein beliebiges Polytop $\mathcal{P}(\mathbf{J})$ im Zustandsraum beschreiben; eine Indexmenge $\mathbf{J} = \{1^+, 2^-, 3^-\}$ würde dabei gleichbedeutend sein mit $d_1(\mathbf{x}) \geq 0$, $d_2(\mathbf{x}) < 0$ und $d_3(\mathbf{x}) < 0$; sie beschreibt, wie sich das Polytop $\mathcal{P}(\mathbf{J})$ zu jeder der aufgezählten Hyperebenen verhält. Zusätzlich wird eine Indexmenge \mathbf{I} definiert, die beschreibt, welche Polytope \mathcal{P}_i der ursprünglichen Polytopstruktur zumindest teilweise innerhalb des durch $\mathcal{P}(\mathbf{J})$ beschriebenen Raums liegen:

$$\mathbf{I}(\mathbf{J}) = \{i | \mathcal{P}_i \cap \mathcal{P}(\mathbf{J}) \text{ ist volldimensional}\} \quad (9.75)$$

Für jede Indexmenge \mathbf{I} existiert ferner eine Menge $\mathbf{F}(\mathbf{I})$, welche die Regelgesetze der innerhalb von $\mathcal{P}(\mathbf{J})$ liegenden Polytope aufzählt:

$$\mathbf{F}(\mathbf{I}) = \{k | F_k \text{ gehört zu } \mathcal{P}_i, i \in \mathbf{I}\} \quad (9.76)$$

Diese letzte Definition ist wichtig, da verschiedene Polytope ein und dasselbe Regelgesetz besitzen können. Besonders interessant ist dies bei Receding Horizon Control, weil in diesem Fall nur der erste Schritt der vorausberechneten Stellgrößenfolge an das System übergeben wird. Folglich braucht bei der Konstruktion des Suchbaums nur nach dem ersten Element des Regelgesetzes unterschieden werden, wodurch sich die Baumstruktur deutlich vereinfacht.

9.5.3 Binary search tree

Another method to reduce the online evaluation time is to transform the polytope structure obtained with the standard method into a binary search tree. Tøndel, Johansen and Bemporad have proposed such a strategy in [117]. For this purpose, all hyperplanes,

² Die Begriffe „oberhalb“ und „unterhalb“ sind hier in Anführungszeichen gesetzt, da die Definition von „oben“ und „unten“ in höherdimensionalen Räumen nicht mehr dem natürlichen Empfinden entspricht, sondern nur noch durch die Mathematik gegeben ist.

i. e. the boundaries of all polytopes, are considered in a first step. The whole number of hyperplanes shall be L and they are each represented by the equation $\mathbf{a}_j^T \mathbf{x} = \mathbf{b}_j$, $j = 1 \dots L$. With the help of this equation, a describing function $d_j(\mathbf{x})$ is defined as follows, see equation (9.74). By evaluating the sign of $d_j(\mathbf{x})$ for every point in state space, it can be stated if it is “above” or “below”² the corresponding hyperplane j . Consequently, any polytope $\mathcal{P}(\mathbf{J})$ in state space can be described by a so called index set \mathbf{J} ; an index set $\mathbf{J} = \{1^+, 2^-, 3^-\}$ would be equivalent to $d_1(\mathbf{x}) \geq 0$, $d_2(\mathbf{x}) < 0$ and $d_3(\mathbf{x}) < 0$; it describes how the polytope $\mathcal{P}(\mathbf{J})$ behaves with each of the enumerated hyperplanes. Additionally, an index set \mathbf{I} is defined that describes which polytopes \mathcal{P}_i of the original polytope structure lie at least partly within the region described by $\mathcal{P}(\mathbf{J})$, see equation (9.75). Furthermore, for every index set \mathbf{I} , a set $\mathbf{F}(\mathbf{I})$ exists which enumerates the control laws of the polytopes lying within $\mathcal{P}(\mathbf{J})$, see equation (9.76).

This last definition is important, because different polytopes can have the same control law. This is particularly interesting considering Receding Horizon Control since, in this case, only the first value of the sequence of values of the actuating variables will be passed on to the system. Hence, for the construction of the search tree one only has to differentiate with respect to the first element of the control law which leads to a significantly simplified tree structure.

9.5.3 二叉搜索树

另外一种减少在线评估时间的方法是尝试将基于标准方法所得到的多胞体结构转换为一个二叉搜索树。学者 Tøndel, Johansen 和 Bemporad 在文献 [117] 中提出了上述方法。为此, 需要首先考虑所有的超平面, 也即所有多胞体的边界。假设超平面的总数量为 L , 并且它们分别可由方程 $\mathbf{a}_j^T \mathbf{x} = \mathbf{b}_j$, $j = 1 \dots L$ 进行描述。基于上述方程可定义如下的描述函数 $d_j(\mathbf{x})$, 见方程 (9.74)。通过判断 $d_j(\mathbf{x})$ 的符号, 即可确定状态空间内的任一点究竟位于相应超平面的“上方”或“下方”²。从而, 状态空间内的任一多胞体 $\mathcal{P}(\mathbf{J})$ 均可使用一个所谓的索引集 \mathbf{J} 进行描述。索引集 $\mathbf{J} = \{1^+, 2^-, 3^-\}$ 等效于 $d_1(\mathbf{x}) \geq 0$, $d_2(\mathbf{x}) < 0$ 和 $d_3(\mathbf{x}) < 0$, 它描述了多胞体 $\mathcal{P}(\mathbf{J})$ 与每一个列出的超平面之间的关系。另外定义一个索引集 \mathbf{I} , 它描述原始多胞体结构中的哪些多胞体 \mathcal{P}_i 至少部分位于 $\mathcal{P}(\mathbf{J})$ 所表示的空间, 见方程 (9.75)。更进一步, 对于每一个索引集 \mathbf{I} , 还存在一个集合 $\mathbf{F}(\mathbf{I})$, 它列举了位于 $\mathcal{P}(\mathbf{J})$ 内部的多胞体所对应的控制律, 见方程 (9.76)。

前述最后一个定义相当重要, 因为不同的多胞体可能具有相同的控制律。尤其值得注意的是基于滚动优化控制策略的情形, 此时仅将预测计算得到的控制变量序列的第一个元素施加给被控对象。从而, 在构造搜索树时, 仅需要区分的是控制律的第一个元素, 如此即可大大简化树结构。

² Here the terms “above” and “below” are used in quotation marks because the definition of “above” and “below” in multidimensional spaces does not correspond to the natural feeling, but is only given by mathematics.

² “上方”和“下方”这一概念使用引号进行标记, 是因为在多维空间内上方和下方的定义不再与三维空间内的实际感受一致, 而纯粹指的是一种数学描述。

Idee des Verfahrens ist es nun, einen Suchbaum zu entwerfen, bei dem in jedem Knoten N_k für einen gegebenen Zustandsvektor $\mathbf{x}(t) \in \mathbf{X}$ die Beschreibungsfunktion $d_j(\mathbf{x})$ aufgerufen wird. Abhängig von dem Vorzeichen von $d_j(\mathbf{x})$ wird dann entweder in den nachfolgenden linken oder in den rechten Teilbaum verzweigt. Die Indizes der noch auszuwertenden Knoten werden in einer Liste \mathbf{K} abgespeichert. Zu jedem dieser Knoten N_k gehören die beiden Indexmengen \mathbf{J}_k , die die Ergebnisse der bereits angewendeten Bewertungsfunktionen $d_j(\mathbf{x})$ enthält und \mathbf{I}_k , die die Indizes der in $\mathcal{P}(\mathbf{J}_k)$ enthaltenen Polytope beinhaltet, d. h. $\mathbf{I}_k = \mathbf{I}(\mathbf{J}_k)$. Ausgewertete Knoten enthalten, wenn es sich um Endknoten handelt, zusätzlich die Information über das für diesen Betriebspunkt anzuwendende affine Regelgesetz F_k . Andernfalls wird ein Feld mit dem Index der Hyperebene j_k , deren Beschreibungsfunktion $d_{j_k}(\mathbf{x})$ zur Entscheidung über die Auswahl des nachfolgenden Teilbaums dient, hinzugefügt. Die Konstruktion dieses Suchbaums erfolgt nach folgendem Muster:

Zuerst werden für alle Hyperebenen die Indexmengen $\mathbf{I}(j^+)$ und $\mathbf{I}(j^-)$ für $j = 1 \dots L$ bestimmt, d. h. es wird ermittelt, welche Polytope der ursprünglichen Polytopbeschreibung zumindest teilweise „oberhalb“ bzw. „unterhalb“ der betreffenden Hyperebene liegen.

Danach wird die Wurzel des Baums als Knoten N_1 initialisiert. Da die Wurzel alle vorhandenen Polytope umfasst, so gilt folglich, dass \mathbf{J}_1 leer ist und \mathbf{I}_1 sämtliche Polytope beinhaltet. Da die Wurzel noch nicht ausgewertet ist, wird sie als erstes Element in die Menge der noch auszuwertenden Knoten eingetragen. Damit ist $\mathbf{K} = \{N_1\}$. Nun werden iterativ die weiteren Baumknoten ermittelt. Dazu wird die folgende Schleife so lange durchlaufen, bis keine weiteren noch auszuwertenden Knoten mehr existieren, d. h. $\mathbf{K} = \emptyset$.

The concept of the method is to design a search tree in which for every node N_k for a given state vector $\mathbf{x}(t) \in \mathbf{X}$, the describing function $d_j(\mathbf{x})$ is called. Depending on the sign of $d_j(\mathbf{x})$, one will branch either in the following left or the right subtree. The indices of the nodes which do still have to be evaluated are stored in a list \mathbf{K} . For each of the nodes N_k , two index sets \mathbf{J}_k exist, which contain the results of the describing functions $d_j(\mathbf{x})$ already applied and one set \mathbf{I}_k , which contains the indices of the polytopes contained in $\mathcal{P}(\mathbf{J}_k)$, i. e. $\mathbf{I}_k = \mathbf{I}(\mathbf{J}_k)$. Evaluated nodes additionally contain the information about the affine control law F_k to be applied in this operating point if they are end nodes. Otherwise an array with the index of the hyperplane j_k whose describing function $d_{j_k}(\mathbf{x})$ is needed for the decision about the selection of the following part of the tree is added. The construction of this search tree is done in the following way:

At first, for all hyperplanes, the index sets $\mathbf{I}(j^+)$ and $\mathbf{I}(j^-)$ for $j = 1 \dots L$ are determined, i. e. the polytopes of the original description of the polytopes which lie at least partly “above” or “below” the concerning hyperplane have to be identified.

Afterwards, the root of the tree is initialized as node N_1 . As the root encloses all available polytopes, it applies consequently that \mathbf{J}_1 is an empty set and \mathbf{I}_1 contains all polytopes. Since the root has not been evaluated yet, it is set as the first element in the set of the nodes which still have to be evaluated. Hence, $\mathbf{K} = \{N_1\}$. Then, the further nodes of the tree are determined iteratively. For this purpose, the following loop is run through until there are no more nodes to be evaluated, i. e. $\mathbf{K} = \emptyset$.

因此，该方法理念是设计一个搜索树，并在其每一个结点 N_k 处调用所给定状态向量 $\mathbf{x}(t) \in \mathbf{X}$ 所属的描述函数 $d_j(\mathbf{x})$ 。接下来，究竟是在左子树还是右子树进行搜索呢？它取决于 $d_j(\mathbf{x})$ 的符号。待评估结点的索引存储在列表 \mathbf{K} 中。对于每一个结点 N_k 来说，同时存在两个索引集 \mathbf{J}_k ，它们包含应用描述函数 $d_j(\mathbf{x})$ 后所得的结果以及单位集 \mathbf{I}_k ，单位集中包含 $\mathcal{P}(\mathbf{J}_k)$ 中多胞体的索引集，也即 $\mathbf{I}_k = \mathbf{I}(\mathbf{J}_k)$ 。如果已评判的结点为叶结点，那么它还额外包含用于上述工作点的仿射控制律 F_k 信息。否则，则需添加一个超平面索引为 j_k 的区域，其描述函数 $d_{j_k}(\mathbf{x})$ 用于决策后续子树的选择。这一搜索树的构建遵循以下模式：

首先，当 $j = 1 \dots L$ 时，需确定所有超平面的索引集 $\mathbf{I}(j^+)$ 和 $\mathbf{I}(j^-)$ 。换言之，需要清楚的是原始多胞体结构中的哪些多胞体至少部分位于相应超平面的上方或下方。

随后，将树的根结点初始化为结点 N_1 。由于根结点包含了所有存在的多胞体，所以可知 \mathbf{J}_1 为空集， \mathbf{I}_1 包含全部的多胞体。由于还未对根结点进行评判，于是它被看作为待评估结点集合的首个元素，从而有 $\mathbf{K} = \{N_1\}$ 。随后则以迭代的方式确定剩余的结点，从而需要不断地重复下述循环，直至不存在待评估的结点，也即 $\mathbf{K} = \emptyset$ 。

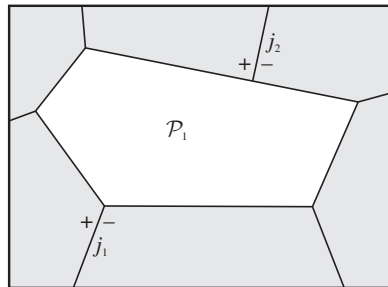
Aus der Menge \mathbf{K} wird ein beliebiger Knoten N_k ausgewählt und \mathbf{K} wird zu $\mathbf{K} \setminus N_k$. Die innerhalb der bisher untersuchten Hyperebenen \mathbf{J}_k liegenden Polytope \mathcal{P}_i mit $i \in \mathbf{I}_k$ werden nun durch sämtliche, noch nicht abgearbeitete Hyperebenen erneut geteilt und die verbleibenden Bereiche annäherungsweise mit $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^\pm)$ für alle $j = 1 \dots L$ bestimmt. Die Ergebnisse werden nach $\max(|\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^+))|, |\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^-))|)$ sortiert. Für die ersten n_j Elemente dieser sortierten Liste werden dann die exakten Indexmengen $\mathbf{I}_k^\pm = \mathbf{I}(\mathbf{J}_k \cup j^\pm)$ berechnet. Der Wert von n_j wird dabei so gewählt, dass alle Hyperebenen aus der sortierten Liste, welche die Anzahl der maximal übrigbleibenden Regelgesetze, d. h. $\max(|\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^+))|, |\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^-))|)$ minimieren, in die erneute Berechnung eingehen. Diese wiederholte Berechnung ist notwendig, da die Abschätzung der verbleibenden Bereiche mit $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^\pm)$ ein größeres Polytop ergeben kann, als die mathematisch exakte, aber auch aufwändigere Methode über $\mathbf{I}_k^\pm = \mathbf{I}(\mathbf{J}_k \cup j^\pm)$; das folgende Beispiel diene als Illustration des Sachverhalts.

Gegeben sei eine Polytopstruktur gemäß Abb. 9.11, dabei werden nur das Polytop \mathcal{P}_1 und die beiden Hyperebenen j_1 und j_2 betrachtet. Offensichtlich gilt:

$$\mathbf{I}(1^+) = \{1\} \qquad \mathbf{I}(2^+) = \{1\} \qquad (9.77)$$

$$\mathbf{I}(1^-) = \{1\} \qquad \mathbf{I}(2^-) = \{1\} \qquad (9.78)$$

Nun sei in einem der vorherigen Rekursionsschritte zur Berechnung des Suchbaums bereits 1^+ angewendet worden, d. h. $\mathbf{I}(\mathbf{J}_k) = \{1^+, \dots\}$. Für weitere Schritte existiert von Polytop \mathcal{P}_1 folglich nur noch der Bereich links von j_1 . Wendet man jetzt die Abschätzung des verbleibenden Bereichs bezüglich der Hyperebene j_2 an, so ist $\{1\}$ sowohl ein Element von $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(2^+)$ als auch von $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(2^-)$, da $\{1\} \in \mathbf{I}(2^+)$ und $\{1\} \in \mathbf{I}(2^-)$. In Wirklichkeit liegt „unterhalb“, also rechts von j_2 jedoch kein Teil von \mathcal{P}_1 mehr, da dieser Teil bereits durch die Anwendung von 1^+ weggeschnitten wurde. Die Berechnung mit $\mathbf{I}_k^- = \mathbf{I}(\mathbf{J}_k \cup 2^-)$ erhält $\{1\}$ richtigerweise nicht.



9.11: Funktionsprinzip Suchbaum/Functional principle of the search tree/搜索树工作原理

From the set \mathbf{K} , one node N_k is selected and \mathbf{K} will become $\mathbf{K} \setminus N_k$. The polytopes \mathcal{P}_i with $i \in \mathbf{I}_k$ being in the hyperplanes \mathbf{J}_k which have already been evaluated are now separated again by all hyperplanes which have not been evaluated yet and the remaining regions are determined approximately with $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^\pm)$ for all $j = 1 \dots L$. The results are sorted by $\max(|\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^+))|, |\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^-))|)$. Then, for the first n_j elements of this sorted list, the exact index sets $\mathbf{I}_k^\pm = \mathbf{I}(\mathbf{J}_k \cup j^\pm)$ are calculated. The value of n_j is selected in such a way that all hyperplanes of the sorted list which minimize the number of the maximum remaining control laws, i. e. $\max(|\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^+))|, |\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^-))|)$, are used for the calculation again. This repeated calculation is necessary since the estimation of the remaining regions with $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^\pm)$ can result in a larger polytope than the mathematically exact, but also more complex method with $\mathbf{I}_k^\pm = \mathbf{I}(\mathbf{J}_k \cup j^\pm)$; the following example shall serve as an illustration of this fact.

When a polytope structure according to Figure 9.11 is given and only the polytope \mathcal{P}_1 and both hyperplanes j_1 and j_2 are considered, apparently the following statements are valid, see equations (9.77) and (9.78). Now suppose that in one of the previous recursion steps for the calculation of the search tree 1^+ has already been applied, i. e. $\mathbf{I}(\mathbf{J}_k) = \{1^+, \dots\}$. Hence, for further steps, only the area of the polytope left of j_1 is existent. If the estimation of the remaining region concerning the hyperplane j_2 is applied, one can see that $\{1\}$ is an element of $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(2^+)$ as well as of $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(2^-)$ because $\{1\} \in \mathbf{I}(2^+)$ and $\{1\} \in \mathbf{I}(2^-)$. In reality, however, “below”, i. e. right of j_2 , there is no further part of \mathcal{P}_1 , because this part has already been cut off by the application of 1^+ . The calculation with $\mathbf{I}_k^- = \mathbf{I}(\mathbf{J}_k \cup 2^-)$ correctly does not contain $\{1\}$.

从集合 \mathbf{K} 中任选一个结点 N_k , 那么集合 \mathbf{K} 则变为 $\mathbf{K} \setminus N_k$. 迄今为止所考察的位于超平面 \mathbf{J}_k 内的多胞体 \mathcal{P}_i 需重新借助于所有仍未被处理的超平面进行分割, 其中 $i \in \mathbf{I}_k$. 对于全部的 $j = 1 \dots L$ 来说, 剩余的区域可近似地根据 $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^\pm)$ 进行确定. 所得结果将根据 $\max(|\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^+))|, |\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^-))|)$ 进行排序. 对于排序后列表中的前 n_j 个元素来说, 可通过计算得到精确的索引集 $\mathbf{I}_k^\pm = \mathbf{I}(\mathbf{J}_k \cup j^\pm)$. n_j 值得选取原则是: 排序后列表中的可以使得最大剩余控制律数量最小化的那些超平面, 也即最小化 $\max(|\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^+))|, |\mathbf{F}(\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^-))|)$, 并重新用于计算. 这一重复计算是必要的, 因为相较于从数学角度来看虽说更为精确, 但却更加复杂的基于 $\mathbf{I}_k^\pm = \mathbf{I}(\mathbf{J}_k \cup j^\pm)$ 的方式, 借助于 $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(j^\pm)$ 对剩余区域进行估计可以得到一个范围更大的多胞体. 接下来的举例可以更好地阐述这一事实.

图 9.11 中给出了一个多胞体结构, 我们仅考察多胞体 \mathcal{P}_1 以及两个超平面 j_1 和 j_2 . 很明显, 满足条件 (9.77) 和 (9.78). 现假设在之前的迭代步数中已使用过 1^+ 用于搜索树的计算, 也即 $\mathbf{I}(\mathbf{J}_k) = \{1^+, \dots\}$. 因此, 在之后的步数中, 多胞体 \mathcal{P}_1 中还剩余的仅仅是 j_1 左侧的部分. 如果接下来对剩余的区域进行评估时考虑超平面 j_2 , 可以看出, $\{1\}$ 既是 $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(2^+)$ 的元素, 也是 $\mathbf{I}(\mathbf{J}_k) \cap \mathbf{I}(2^-)$ 的元素, 因为 $\{1\} \in \mathbf{I}(2^+)$, 且 $\{1\} \in \mathbf{I}(2^-)$. 实际上, 在 j_2 的下方或右侧已经没有 \mathcal{P}_1 的任何部分了, 因为该部分已经在应用 1^+ 时被切除了. 所以, 基于 $\mathbf{I}_k^- = \mathbf{I}(\mathbf{J}_k \cup 2^-)$ 的计算无法准确地包含 $\{1\}$.

Für die exakten Indexmengen wird nun diejenige Hyperebene j_k bestimmt, für die der Wert von $\max(|\mathbf{F}(\mathbf{I}_k^+)|, |\mathbf{F}(\mathbf{I}_k^-)|)$ minimal wird. Dadurch wird ermittelt, welche Hyperebene die bestmögliche Verteilung von Regelgesetzen ergibt, mit dem Ziel, die Baumtiefe zur Verringerung der online-Rechenzeit möglichst gering zu halten. Sollte mehr als eine Hyperebene diese Optimalitätsbedingung erfüllen, so wird als zusätzliches Kriterium die Anzahl der verbleibenden Polytope $\max(|\mathbf{I}_k^+|, |\mathbf{I}_k^-|)$ betrachtet, d. h. es wird diejenige Hyperebene ausgewählt, die neben der Anzahl der verbleibenden Regelgesetze auch die Menge der noch zu untersuchenden Polytope reduziert. Bleiben immer noch mehrere, optimale Hyperebenen übrig, wird aus Komplexitätsgründen nicht weiter untersucht, sondern einfach eine der möglichen Hyperebenen ausgewählt.

Da nun feststeht, durch welche Hyperebene der Zustandsraum am besten aufgeteilt wird, so kann die Information des Baumknotens N_k vervollständigt werden. Die oben ermittelte optimale Hyperebene j_k wird zur Teilung des Baums verwendet. Die beiden entstehenden Folgeknoten N_k^\pm werden mit $\mathbf{J}_k^\pm = \mathbf{J}_k \cup j^\pm$ und den oben bestimmten \mathbf{I}_k^\pm initialisiert. Ist die Anzahl von Regelgesetzen in den Folgeknoten $|\mathbf{F}(\mathbf{I}^\pm)| > 1$, so werden sie der Menge der noch zu untersuchenden Knoten hinzugefügt. Andernfalls ist der Folgeknoten ein Endknoten, da für den gesamten übrigbleibenden Bereich ein und dasselbe Regelgesetz gilt. Dem Endknoten wird daher die Nummer des entsprechenden Regelgesetzes zugewiesen.

Wie bereits weiter oben erwähnt, werden diese Schritte solange wiederholt, bis alle Knoten untersucht sind. Der Suchbaum ist dann vollständig bekannt. Die Berechnung des binären Suchbaums ist mathematisch relativ aufwändig; da dies jedoch offline erfolgt, kann der Aufwand in Kauf genommen werden.

Ist der Suchbaum einmal berechnet, vereinfacht sich die online-Auswertung des expliziten Regelgesetzes deutlich. Dazu braucht lediglich der folgende Algorithmus abgearbeitet werden:

For the exact index sets, the hyperplane j_k is now determined, for which the value of $\max(|\mathbf{F}(\mathbf{I}_k^+)|, |\mathbf{F}(\mathbf{I}_k^-)|)$ becomes minimal. Thus, it can be determined which hyperplane produces the best possible distribution of control laws with the intent to keep the depth of the search tree as small as possible in order to reduce the necessary online computation time. If more than one hyperplane satisfies this optimality constraint the number of remaining polytopes, $\max(|\mathbf{I}_k^+|, |\mathbf{I}_k^-|)$, is considered as additional criterion, i. e. the hyperplane which reduces the set of polytopes still to be evaluated and also the number of remaining control laws, is selected. If there are still several optimum hyperplanes remaining, there is no further evaluation due to reasons of complexity, then simply one of the possible hyperplanes is selected.

As it is determined now by which hyperplane the state space is separated in the best way, the information about the tree node N_k can be completed. The optimum hyperplane j_k , determined above, is used for the separation of the tree. Both originating

subsequent nodes N_k^\pm are initialized with $\mathbf{J}_k^\pm = \mathbf{J}_k \cup j^\pm$ and the \mathbf{I}_k^\pm determined above. If the number of control laws in the subsequent node is $|\mathbf{F}(\mathbf{I}_k^\pm)| > 1$, they are added to the set of nodes which still have to be evaluated. Otherwise the subsequent node is an end node, as for the whole remaining region one single control law is valid. Thus, the number of the corresponding control law is assigned to this end node.

As already mentioned above, these steps are repeated until all nodes are evaluated. Then, the search tree is completely known. The calculation of the binary search tree is mathematically rather complex; due to the fact that the calculation can be done offline, this complexity can be accepted.

Once the calculation of the search tree is done, the online evaluation of the explicit control law is significantly simplified. For this purpose, only the following algorithm has to be processed:

对于精确的索引集来说, 现需确定那些能够最小化 $\max(|\mathbf{F}(\mathbf{I}_k^+)|, |\mathbf{F}(\mathbf{I}_k^-)|)$ 的超平面 j_k 。从而可以知晓, 哪一超平面可以得到最佳的控制律分布, 目标是使树的深度尽可能地小, 从而可以减小在线计算时间。假使多于一个超平面可以满足最优条件, 那么则需考虑剩余超平面 $\max(|\mathbf{I}_k^+|, |\mathbf{I}_k^-|)$ 的数量作为附加的评判条件。换言之, 一方面需要选择那些能够减小剩余控制律数量的超平面; 一方面还需选择能够减少待评估多胞体数量的超平面。应用上述附加评判条件之后, 如果仍存在多个最优的超平面, 那么出于复杂性的缘故将不再继续考察, 而是直接从可能的超平面中随意选择一个。

由于目前已经确定使用哪一超平面可以对状态空间进行最佳分割, 那么树结点 N_k 的信息也随之得到了完善。前面所确定的最优超平面 j_k 用于树的划分, 所形成的两个子结点 N_k^\pm 分别使用 $\mathbf{J}_k^\pm = \mathbf{J}_k \cup j^\pm$ 和之前确定的 \mathbf{I}_k^\pm 进行初始化。如果子节点上控制律的数量 $|\mathbf{F}(\mathbf{I}_k^\pm)| > 1$, 那么该节点则被添加至待考察节点的集合中。否则, 这一子节点为叶节点, 因为此时对于全部剩余的区域来说均满足一个相同的控制律。因此, 相应控制律的编号则分配给此叶节点。

如前所述, 上述步骤需不断重复, 直至遍历所有的结点。从而, 搜索树是完全已知的。从数学上来看, 二叉搜索树的计算相当复杂, 但是考虑到其计算过程是离线完成的, 所以这一复杂性可以接受。

一旦计算出了搜索树, 那么显式控制律的在线评估则明显简化。为此, 只需逐步完成下述算法即可:

1. Wähle als aktuellen Knoten N_k die Wurzel N_1 .
2. Werte die Beschreibungsfunktion $d_j(\mathbf{x})$ aus Gleichung (9.74) aus, wobei $j = j_k$ und \mathbf{x} der aktuelle Systemzustand sind.
3. Verzweige abhängig vom Vorzeichen von $d_j(\mathbf{x})$ zum entsprechenden Folgeknoten, d. h. $N_k = N_k^+$ oder $N_k = N_k^-$.
4. Wenn N_k ein Endknoten ist, so berechne aus dem Regelgesetz des Endknotens die Stellgröße $\mathbf{u}(t)$. Andernfalls gehe zurück zu Schritt 2.

Bei der praktischen Umsetzung eines nach dem Receding-Horizon-Prinzips arbeitenden MPC-Reglers ist es wichtig, den Baum immer nur an solchen Hyperebenen zu verzweigen, welche Polytope voneinander trennen, deren zugeordnete Regelgesetze sich bereits im ersten Schritt unterscheiden. Auf diese Art und Weise wird eine zusätzliche Vereinfachung des Suchbaums bei weiterhin vollständig optimalem Regelverfahren bewirkt.

1. Select the root N_1 as current node N_k .
2. Evaluate the describing function $d_j(\mathbf{x})$ from equation (9.74) in which \mathbf{x} is the current system state and $j = j_k$.
3. Branch to the corresponding subsequent node dependent on the sign of $d_j(\mathbf{x})$, i. e. $N_k = N_k^+$ or $N_k = N_k^-$
4. If N_k is an end node, then calculate the value of the actuating variable $\mathbf{u}(t)$ from the control law of the end node. Otherwise go back to step 2.

For a practical realization of an MPC controller based on the receding horizon principle, it is important to branch out the tree only at those hyperplanes that separate polytopes from each other whose corresponding control laws differ already in the first step. In this way, an additional simplification of the search tree can be achieved still granting an optimum control method.

1. 选择根节点 N_1 作为当前节点 N_k ;
2. 评估第 305 页方程 (9.74) 所示的描述函数, 其中 $j = j_k$, \mathbf{x} 为当前系统状态;
3. 根据 $d_j(\mathbf{x})$ 的符号决定应当向哪一个子节点进行分支, 也即 $N_k = N_k^+$ 或 $N_k = N_k^-$;
4. 如果 N_k 为叶节点, 那么则根据此叶节点的控制律计算控制变量 $\mathbf{u}(t)$ 。否则, 返回步骤 2.

在实际实现基于滚动优化原则的 MPC 控制器时, 核心是始终在能够分割多胞体的那些超平面对树进行分支, 其所分配的控制律在第一步时已作出区分。如此, 搜索树的额外简化过程则能够在随后的完全最优控制方法中得到保证。

9.5.4 Optimale Komplexitäts-Reduzierung/Optimal complexity reduction/最优的复杂度降低

In vielen Fällen wird die explizite Lösung eines MPC-Reglers für einen Prädiktionshorizont größer als eins berechnet. Gerade bei einem diskreten Regler ergibt sich dann, dass eine Vielzahl von Polytopen existiert, deren zugehörige Regelgesetze zwar prinzipiell unterschiedlich sind, sich im ersten Schritt jedoch nicht unterscheiden. Da bei einer echten Regelung nur der erste Wert der vorausberechneten Stellgrößenfolge tatsächlich an die zu regelnde Strecke ausgegeben wird, ist es in einem solchen Fall sinnvoll, Polytope, deren Regelgesetze im ersten Schritt gleich sind, miteinander zu verschmelzen, um die Komplexität der Struktur und damit auch die zur online-Auswertung notwendige Rechenzeit zu reduzieren. Leider kann dieses Zusammenfassen nicht durch eine einfache Vereinigung aller in Frage kommenden Polytope realisiert werden, da sichergestellt werden muss, dass die resultierende Struktur wiederum aus konvexen Polytopen besteht (siehe weiter vorne in diesem Kapitel). In [43] schlagen Geyer, Torrisi und Morari ein Verfahren vor, um Regionen mit gleichem Regelgesetz möglichst geschickt zusammenzufassen, wobei die Konvexitätsbedingung beachtet wird. Wird diese Strategie in Abhängigkeit von dem ersten Schritt eines jeden Regelgesetzes angewendet, so ist eine deutliche Vereinfachung der Polytopstruktur zu erwarten. Dazu geht man folgendermaßen vor:

Liegt die berechnete stückweise affine Reglerstruktur als Gruppe von Hyperebenen vor, so lässt sich über einen Vorzeichenvektor, der im Prinzip angibt, wie sich eine bestimmte Region zu den verschiedenen Hyperebenen verhält, eine eindeutige Markierung M für jede Region bestimmen. Ein Algorithmus, der eine solche *cell enumeration* vornimmt, wird z. B. von Geyer et al. in [42] ausführlich erläutert; auf eine detaillierte Beschreibung wird an dieser Stelle verzichtet. Die erhaltenen Markierungen sind die Basis für die im Folgenden vorzunehmende optimale Reduktion der Komplexität.

Wenn sämtliche Grenzen nur über Zustandsvariablen und Eingangsvariablen definiert sind, sind die Hyperebenen und die Markierungen ebenfalls über den Raum der Zustands- und Eingangsvariablen definiert. Einen solchen Fall nennt man *global hyperplane arrangement*.

9.5.4 Optimal complexity reduction

In many cases the explicit solution of an MPC controller is calculated for a prediction horizon greater than one. Considering especially a discrete controller, a multiplicity of polytopes whose corresponding control laws being basically different, however, not in the first step, do exist. As, in a real control, only the first value of the precalculated sequence of values for the actuating variables is really applied to the plant, it makes sense to combine polytopes whose control laws are the same for the first step in order to reduce the complexity of the structure and therewith also to reduce the necessary computation time. Unfortunately, this combination cannot be realized by simply combining all possible

polytopes, since it has to be made sure that the resulting structure again consists of convex polytopes (see further in this chapter). In [43], Geyer, Torrisi and Morari propose a method to combine regions with the same control law as skillfully as possible under consideration of the convexity constraint. If this strategy is applied in dependence of the first step of every control law, a significant simplification of the polytope structure can be expected. Thus one can proceed as follows:

If the calculated piecewise affine controller structure is available as a collection of hyperplanes, a unique marking M for every region can be determined via a sign vector which in principle shows the relation between a certain region and the different hyperplanes. An algorithm doing such a *cell enumeration* is e.g. described in detail by Geyer et al. in [42]; here a detailed description is omitted. The determined markings are the basis for the optimal complexity reduction to be done in the following.

If all boundaries are only defined on state and input variables, the hyperplanes and markings are also defined on the state and input variable space. Such a case is called *global hyperplane arrangement*.

9.5.4 最优的复杂度降低

在大多情况下, 针对预测步长大于 1 的情形才计算一个 MPC 控制器的显式解。特别是在离散控制器的情况下, 实际会存在大量的多胞体, 尽管从原则上来看, 其所包含的控制律不同, 但在第一步时却还未产生区别。由于在实际控制时, 仅将预测计算出的控制变量序列的第一个元素施加给被控系统。所以, 为了降低结构的复杂度并同时减小在线评判所需的时间, 有必要合并那些在第一步时控制律相同的多胞体。遗憾的是, 上述合并是无法通过所有相关多胞体的简单组合得以实现的, 因为还要确保所形成的结构同样是凸多胞体。学者 Geyer, Torrisi 和 Morari 提出了一种尽可能合并那些具有相同控制律的区域, 并且同时可以考虑到凸性条件的方法 [43]。如果将上述策略应用于任意控制律的第一步, 可以预期的是, 多胞体结构将会有明显的简化, 相应处理方式如下所述。

如果计算出的分段仿射控制器结构可以作为超平面的组合, 那么从原则上来看, 可以借助于一个符号向量来判断固定的区域与不同超平面之间的关系, 并可对每一区域确定其特有的标记 M 。可以实现这一细胞遍历的算法已由学者 Geyer 等在文献 [42] 作了详细阐述, 此处不再赘述。前面所得到的标记 M 是实现最优的复杂度降低的基础。

如果所有的边界仅通过状态变量和输入变量进行定义, 那么超平面和其标记也同样是通过对状态变量和输入变量所在空间进行定义。这一情况被称为全局超平面排列。

Die Markierungen M des Hyperplane-Arrangements sind dann identisch mit den bereits in Kap. 9.5.3 beschriebenen Indexmengen \mathbf{J} der stückweise affinen Reglerstruktur und es können sowohl die einzelnen Polytope als auch die einzelnen Dynamikbeschreibungen gleichgesetzt werden. Sollten jedoch die Grenzen darüberhinaus von zusätzlichen Variablen abhängen, die ihrerseits von anderen Grenzen abhängig sind, so ergibt der in [42] beschriebene Algorithmus eine Sammlung von Hyperplane-Arrangements, die sequentiell ineinander definiert sind. Folglich deckt die Menge der Regionen nicht den gesamten Zustandsraum ab, sondern ist eine polytopische Untermenge davon. In diesem Fall spricht man von *local hyperplane arrangement*. Dieser Fall tritt im Allgemeinen auf, wenn das System aus mehreren sogenannten *Diskreten Hybriden Automaten* (DHA) besteht, die sequentiell voneinander abhängig sind, d.h. ein DHA definiert eine Hyperebenen-Gruppe innerhalb eines Polytops der Hyperebenen-Gruppe der vorhergehenden PWA-Beschreibung. Dieser sehr komplexe Fall ist aber bei der hier besprochenen Anwendung in Form einer Antriebsregelung mit Umrichter nicht anzunehmen, so dass davon ausgegangen werden kann, dass es sich um ein globales Hyperplane-Arrangement handelt.

Für eine gegebene PWA-Beschreibung, z. B. eine Reglerstruktur für einen expliziten MPC-Regler, muss nun ein Optimierungsproblem gelöst werden. Das bedeutet, dass aus einer gegebenen Menge $\{\mathcal{P}_i\}$, $i = 1 \dots P$ von Polytopen mit gleicher Dynamikbeschreibung eine neue Menge $\{\mathcal{Q}_j\}$, $j = 1 \dots Q$ mit folgenden Eigenschaften gebildet werden muss:

- Die Vereinigung der neuen Polytope ist gleich der Vereinigung der ursprünglichen, d. h. $\bigcup_{i=1}^P \mathcal{P}_i = \bigcup_{j=1}^Q \mathcal{Q}_j$.
- Die neuen Polytope sind gegenseitig disjunkt (elementfremd), d. h. es gilt $\mathcal{Q}_i \neq \mathcal{Q}_j$ für alle $i, j \in \{1 \dots Q\}$, $i \neq j$.
- Die neuen Polytope sind Vereinigungen der alten, d. h. für jedes \mathcal{Q}_j , $j \in \{1 \dots Q\}$ gibt es eine Indexmenge $\mathbf{I} \subseteq \{1 \dots P\}$, so dass \mathcal{Q}_j die Vereinigung aller \mathcal{P}_i , $i \in \mathbf{I}$ ist.
- Die Zahl Q ist minimal, d. h. es gibt keine Menge $\{\mathcal{Q}_j\}$, $j = 1 \dots Q$ mit einer geringeren Anzahl von Polytopen.

The markings M of the hyperplane arrangement are then identical to the index sets \mathbf{J} of the piecewise affine controller structure that has already been described in Chapter 9.5.3 and the single polytopes as well as the single descriptions of the dynamics can be set equal to each other. However, if the boundaries should furthermore be dependent on additional variables which themselves are again dependent on other boundaries, the algorithm described in [42] results in a collection of hyperplane arrangements being sequentially defined into one another. Hence, the set of regions does not cover the whole state space, but it is a polytope subset of it. Such a case is called *local hyperplane arrangement*. It commonly occurs if the system consists of several so called *discrete hybrid*

automata (DHA) which are sequentially dependent on each other, i. e. one DHA defines a group of hyperplanes within a polytope of the group of hyperplanes of the previous PWA description. However, this very complex case is not to be assumed in the application of a drive control with inverter as discussed here. Thus, it can be assumed that it has to be dealt with a global hyperplane arrangement here.

For a given PWA description, e. g. a controller structure for an explicit MPC controller, now an optimization problem has to be solved. This means that a new set $\{\mathcal{Q}_j\}$, $j = 1 \dots Q$ with the following characteristics has to be created from a given set $\{\mathcal{P}_i\}$, $i = 1 \dots P$ of polytopes which describe the same dynamics:

- The union of the new polytopes is equal to the union of the original ones, i. e. $\bigcup_{i=1}^P \mathcal{P}_i = \bigcup_{j=1}^Q \mathcal{Q}_j$.
- The new polytopes are mutually disjoint, i. e. the following statement is valid: $\mathcal{Q}_i \neq \mathcal{Q}_j$ for all $i, j \in \{1 \dots Q\}$, $i \neq j$.
- The new polytopes are unions of the old ones, i. e. for each \mathcal{Q}_j , $j \in \{1 \dots Q\}$ there is an index set $I \subseteq \{1 \dots P\}$ so that \mathcal{Q}_j is the union of all \mathcal{P}_i , $i \in I$.
- The number Q is minimal, i. e. there is no set $\{\mathcal{Q}_j\}$, $j = 1 \dots Q$ with a lesser number of polytopes.

超平面排列的标记 M 与已在章节 9.5.3 描述的分段仿射控制器结构的指数集 J 一致, 所以它既可以看作为单个多胞体, 也可以看作单个动态描述。然而, 如果上述边界还依赖于其他变量, 而这些变量又同时依赖于其他边界, 文献 [42] 中描述的算法给出了一个以串联方式定义的超平面排列组合。因此, 区域的集合并不覆盖整个状态空间, 而只是属于多胞体的子集。这一情况被称为局部超平面排列。上述情况通常会在由多个所谓的离散混合自动机 (英文缩写 DHA) 组成的系统中出现, 它们串联性地前后相互依赖, 也就是说, 一个 DHA 定义了一个在其上一个 PWA 描述的超平面组的多胞体内的新的超平面组。这一非常复杂的情形并不适用于本章所讨论的基于变流器实现驱动控制的场合, 所以我们仅讨论全局超平面排列的情况。

对于一个给定的 PWA 描述, 如针对显式 MPC 控制器的控制器结构, 必须求解一个优化问题。这意味着必须依据给定的多胞体集合 $\{\mathcal{P}_i\}$, $i = 1 \dots P$ 构建一个具有相同动态特性描述的新的多胞体集合 $\{\mathcal{Q}_j\}$, $j = 1 \dots Q$, 其特征如下:

- 新的多胞体的并集与初始多胞体的并集一致, 也即 $\bigcup_{i=1}^P \mathcal{P}_i = \bigcup_{j=1}^Q \mathcal{Q}_j$;
- 新的多胞体之间是相互不相交的, 也即对于所有的 $i, j \in \{1 \dots Q\}$, $i \neq j$, $\mathcal{Q}_i \neq \mathcal{Q}_j$;
- 新的多胞体是旧的多胞体的并集, 也就是说, 对于每一个 \mathcal{Q}_j , $j \in \{1 \dots Q\}$, 存在一个指数集 $I \subseteq \{1 \dots P\}$ 使得 \mathcal{Q}_j 是所有 \mathcal{P}_i , $i \in I$ 的并集;
- 数 Q 是最小的, 也即不存在包含更少数量多胞体的集合 $\{\mathcal{Q}_j\}$, $j = 1 \dots Q$ 。

Die Optimieraufgabe ist nicht-trivial, denn die Vereinigung aller Polytope mit gleicher Dynamikbeschreibung ist im Allgemeinen nicht konvex; außerdem sind überlappende Regionen nicht gestattet.

Zuerst wird nun angenommen, dass neben der PWA-Beschreibung ein entsprechendes globales Hyperebenen-Arrangement zusammen mit den dazugehörigen Markierungen M existiert, was für die vorhandene Regelstruktur als gegeben angenommen bzw. leicht ermittelt werden kann. Mit Hilfe dieser Markierungen lassen sich folgende Eigenschaften von Polytopen schnell ermitteln:

Trennende Hyperebenen: Gegeben seien eine Hyperebene j und zwei Polytope \mathcal{P}_1 und \mathcal{P}_2 mit den dazugehörigen Markierungen M_1 und M_2 . Unterscheiden sich M_1 und M_2 im j -ten Element, so ist die Hyperebene j eine trennende Hyperebene bezüglich der Polytope \mathcal{P}_1 und \mathcal{P}_2 .

Einhüllende: Gegeben seien zwei Polytope \mathcal{P}_1 und \mathcal{P}_2 mit den dazugehörigen Markierungen M_1 und M_2 , wobei $M_1(i) = M_2(i)$ für alle $i \in \mathbf{J}$ und $M_1(i) \neq M_2(i)$ für alle $i \in \{1 \dots L\} \setminus \mathbf{J}$. L stellt dabei die Gesamtanzahl aller Hyperebenen dar. Bildet man nun eine neue Markierung M mit $M(i) = M_1(i)$ für alle $i \in \mathbf{J}$ und $M(i) = *$ für alle $i \in \{1 \dots L\} \setminus \mathbf{J}$, so wird die Einhüllende $\text{env}(\mathcal{P}_1, \mathcal{P}_2)$ durch M bestimmt.

Konvexität: Gegeben seien zwei Polytope \mathcal{P}_1 und \mathcal{P}_2 mit den dazugehörigen Markierungen M_1 und M_2 , $M_1 \neq M_2$. Die Vereinigung $\mathcal{P}_1 \cup \mathcal{P}_2$ ist konvex, wenn die Markierungen M_1 und M_2 sich genau in einem Element unterscheiden.

Benachbarte Polytope: Zwei Polytope bezeichnet man als benachbart, wenn sie eine gemeinsame Seitenfläche (Facette) haben.

Verbindungen: Gegeben sei eine Menge von Polytopen $\{\mathcal{P}_i\}$, $i \in \mathbf{I}$ mit den dazugehörigen Markierungen M_i . Die Polytope sind verbunden, wenn für jedes Polytop \mathcal{P}_i , $i \in \mathbf{I}$ ein anderes Polytop \mathcal{P}_j , $j \in \mathbf{I}$, $j \neq i$ existiert, so dass sich die dazugehörigen Markierungen M_i und M_j in genau einem Element unterscheiden.

The optimization task is non-trivial because the union of all polytopes with the same description of the dynamics is normally not convex; furthermore, overlapping regions are not allowed.

First, it is to be assumed that besides the PWA description a corresponding global hyperplane arrangement exists together with the associated markings M which can be given for the existent control structure or be determined easily. With the help of these markings the following characteristics of polytopes can be determined quickly:

Separating hyperplanes: A hyperplane j and two polytopes \mathcal{P}_1 and \mathcal{P}_2 with the associated markings M_1 and M_2 shall be given. The hyperplane j is a separating

hyperplane with regard to the polytopes \mathcal{P}_1 and \mathcal{P}_2 if M_1 and M_2 differ in the j th element.

Envelope: Two polytopes \mathcal{P}_1 and \mathcal{P}_2 with the associated markings M_1 and M_2 shall be given, in which $M_1(i) = M_2(i)$ for all $i \in \mathbf{J}$ and $M_1(i) \neq M_2(i)$ for all $i \in \{1 \dots L\} \setminus \mathbf{J}$. L is the whole number of all hyperplanes. If a new marking M with $M(i) = M_1(i)$ for all $i \in \mathbf{J}$ and $M(i) = *$ for all $i \in \{1 \dots L\} \setminus \mathbf{J}$ is constructed, the envelope $\text{env}(\mathcal{P}_1, \mathcal{P}_2)$ is given by M .

Convexity: Two polytopes \mathcal{P}_1 and \mathcal{P}_2 with the associated markings M_1 and M_2 , $M_1 \neq M_2$ shall be given. The combination $\mathcal{P}_1 \cup \mathcal{P}_2$ is convex if, and only if, the markings M_1 and M_2 differ in exactly one element.

Neighboring polytopes: Two polytopes are called neighboring if they have a common facet.

Connections: A set of polytopes $\{\mathcal{P}_i\}$, $i \in \mathbf{I}$ with the associated markings M_i shall be given. The polytopes are connected if another polytope \mathcal{P}_j , $j \in \mathbf{I}$, $j \neq i$ exists for every polytope \mathcal{P}_i , $i \in \mathbf{I}$, so that the associated markings M_i and M_j differ in exactly one element.

上述优化任务并不繁琐，因为具有相同动态特性描述的所有多胞体的并集通常是非凸的。此外，有重叠的区域也不存在。

首先，我们假定除了 PWA 描述之外还存在一个相应的包含其所属的标记 M 全局超平面排列，可认为它是由当前的控制器结构所给定或者可以轻易地获取。借助于上述标记可以迅速地确定以下几点特性：

分离性超平面： 给定一个超平面 j ，以及两个标记分别为 M_1 和 M_2 的多胞体 \mathcal{P}_1 和 \mathcal{P}_2 。如果 M_1 和 M_2 的第 j 个元素不相同，那么，超平面 j 可以看作是两胞体 \mathcal{P}_1 和 \mathcal{P}_2 的分离性超平面；

包络： 给定两个多胞体 \mathcal{P}_1 和 \mathcal{P}_2 ，其标记分别为 M_1 和 M_2 ，其中对于所有的 $i \in \mathbf{J}$ ， $M_1(i) = M_2(i)$ ，且对于剩余的 $i \in \{1 \dots L\} \setminus \mathbf{J}$ ， $M_1(i) \neq M_2(i)$ 。 L 表示所有超平面的数量。如果现在构建一个新的标记 M ，对于所有的 $i \in \mathbf{J}$ ， $M(i) = M_1(i)$ ，且对于剩余的 $i \in \{1 \dots L\} \setminus \mathbf{J}$ ， $M(i) = *$ 。如此，即可根据 M 来确定包络 $\text{env}(\mathcal{P}_1, \mathcal{P}_2)$ ；

凸性： 给定两个多胞体 \mathcal{P}_1 和 \mathcal{P}_2 ，其标记分别为 M_1 和 M_2 ，且 $M_1 \neq M_2$ 。当标记 M_1 和 M_2 中的所有元素均不相同，所以并集 $\mathcal{P}_1 \cup \mathcal{P}_2$ 是凸的；

相邻多胞体： 如果两个多胞体拥有一个共同的面，那么则称其为相邻多胞体；

关联性： 给定一个多胞体集合 $\{\mathcal{P}_i\}$ ， $i \in \mathbf{I}$ ，以及其所属的标记 M_i 。如果对于任意的多胞体 \mathcal{P}_i ， $i \in \mathbf{I}$ ，存在另外一个多胞体 \mathcal{P}_j ， $j \in \mathbf{I}$ ， $j \neq i$ 使得其所属标记 M_i 和 M_j 中的对应元素不相同，那么则称多胞体是具有关联性。

Wie unschwer zu erkennen ist, lässt sich allein aus den Markierungen des Hyperebenen-Arrangements auf einfache Weise herauslesen, ob zwei Polytope verbunden sind, ob ihre Einhüllende konvex ist oder ähnliches. Dies wird nun ausgenutzt, um mit Hilfe eines rekursiven Branch-and-Bound-Algorithmus jene Polytope der Reglerstruktur in optimaler Weise zusammenzufassen, die das gleiche Regelgesetz beinhalten. Das klingt auf den ersten Blick kompliziert, ist aber relativ einfach.

Sollen z. B. die Polytope mit dem Regelgesetz x optimal zusammengefasst werden, so wird zuerst die Einhüllende $\mathcal{P}_x = \text{env}(\mathbf{M}_x)$ gebildet, wobei \mathbf{M}_x die Menge aller Markierungen dieser Polytope darstellt. Wie oben erläutert, kann die Markierung M_x der Einhüllenden einfach durch Setzen der entsprechenden Elemente von M_x auf den Wert '*' bestimmt werden. Wenn \mathcal{P}_x keine Polytope mit anderen Regelgesetzen beinhaltet, ist das Problem bereits gelöst, da dies bedeutet, dass die Vereinigung aller Polytope mit dem Regelgesetz x konvex ist; die optimale Zusammenfassung besteht in diesem Fall einfach aus der Vereinigung aller dieser Polytope.

Im Allgemeinen wird $\text{env}(\mathbf{M}_x)$ jedoch auch Polytope anderer Regelgesetze einschließen, d. h. die Vereinigung von \mathbf{M}_x ist nicht konvex. In diesem Fall muss die Einhüllende \mathcal{P}_x sinnvoll geteilt werden. Sei dazu \mathbf{J}_x die Menge der Indizes aller trennenden Hyperebenen bezüglich der Polytope innerhalb von \mathcal{P}_x . Wie die oben angeführten Definitionen belegen, ist \mathbf{J}_x einfach die Aufzählung aller Stellen, an denen die Elemente von M_x den Wert '*' haben und daher ebenfalls einfach zu bestimmen. Für jede Hyperebene $i \in \mathbf{J}_x$ lässt sich nun die Einhüllende in zwei neue konvexe Bereiche teilen, die als $\mathcal{P}_{x,i+}$ und $\mathcal{P}_{x,i-}$ mit den dazugehörigen Mengen von Markierungen $\mathbf{M}_{x,i+}$ bzw. $\mathbf{M}_{x,i-}$ bezeichnet werden. Der Branch-and-Bound-Algorithmus verzweigt nun an der Hyperebene i , indem er sich selbst zweimal aufruft; einmal mit $\mathbf{M}_{x,i+}$ und einmal mit $\mathbf{M}_{x,i-}$ als zu optimierende Polytopstruktur. Dies wird für alle übrigen Hyperebenen $i \in \mathbf{J}_x$ ebenfalls durchgeführt.

Having a closer look at the markings of the hyperplane arrangements, it can easily be figured out if two polytopes are connected with each other, if their envelope is convex or similar things. This is used in order to merge those polytopes of the controller structure which contain the same control law with the help of a recursive Branch and Bound algorithm. At first look, this seems to be complicated, however, it is quite simple.

If e. g. the polytopes with the control law x should be optimally combined, the envelope $\mathcal{P}_x = \text{env}(\mathbf{M}_x)$ is constructed first, in which \mathbf{M}_x describes the set of all markings of these polytopes. As described above, the marking M_x of the envelope can simply be determined by setting the corresponding elements of M_x to the value '*'. If \mathcal{P}_x does not contain any polytopes with other control laws the problem is already solved due to the fact that the combination of all polytopes with the control law x is convex; in this case, the optimum combination simply consists of the combination of all polytopes.

Usually, $\text{env}(\mathbf{M}_x)$ will, however, include polytopes with other control laws, i. e. the combination of \mathbf{M}_x is not convex. In this case the envelope \mathcal{P}_x has to be reasonably separated. Thereto, \mathbf{J}_x shall be the set of indices of all separating hyperplanes with regard to the polytopes within \mathcal{P}_x . As the definitions stated above prove, \mathbf{J}_x is simply the enumeration of all positions on which the elements of \mathbf{M}_x have the value '*' and thus it can be easily determined. For every hyperplane $i \in \mathbf{J}_x$, the envelope can now be separated into two new convex regions which are named $\mathcal{P}_{x,i+}$ and $\mathcal{P}_{x,i-}$ with the associated markings $\mathbf{M}_{x,i+}$ and $\mathbf{M}_{x,i-}$. The Branch and Bound algorithm now branches at the hyperplane i by calling itself two times; once with $\mathbf{M}_{x,i+}$ and once with $\mathbf{M}_{x,i-}$ as the polytope structure that should be optimized. This step is repeated for all other hyperplanes $i \in \mathbf{J}_x$.

不难看出，仅通过超平面排列的标记即可轻易确定两个多胞体是否具有关联性，其包络是否具有凸性，以及其他类似特性。可以充分利用上述特性并结合递归的分支定界算法以最优的方式来整合具有相同控制律的控制器结构中的多胞体。初看之下，上述方式似乎非常复杂，但实际上是简单的。

假使需要对具有相同控制律 x 的多胞体进行最优程度的整合，首先需要构建的是包络 $\mathcal{P}_x = \text{env}(\mathbf{M}_x)$ ，其中 \mathbf{M}_x 表示以上多胞体所有标记的集合。如前所述，可以通过简单的设置 M_x 中相应元素的值为 '*' 来确定包络的标记 M_x 。如果 \mathcal{P}_x 中不包含具有其他控制律的多胞体，那么问题则得到了解决。这也意味着，包含控制律 x 的所有多胞体的并集是凸的，在这一情况下，最优的整合可简单地由所有上述多胞体的并集完成。

一般来说，包络 $\text{env}(\mathbf{M}_x)$ 中会包含具有其他控制律的多胞体，也就是说， \mathbf{M}_x 的并集是非凸的。在这一情况下，需要合理地分离包络 \mathcal{P}_x 。假定 \mathbf{J}_x 是 \mathcal{P}_x 内相关多胞体的所有分离性超平面的指数集，正如前面的定义所述， \mathbf{J}_x 是所有那些 \mathbf{M}_x 中的元素具有 '*' 值的位置的列表，从而可以轻易地被确定。对于任一超平面 $i \in \mathbf{J}_x$ ，现将包络分成两个新的凸区域，并使用 $\mathcal{P}_{x,i+}$ 和 $\mathcal{P}_{x,i-}$ 以及其所属的标记 $\mathbf{M}_{x,i+}$ 和 $\mathbf{M}_{x,i-}$ 的集合进行表示。分支定界算法通过两次调用以在超平面 i 处进行分支：一次是 $\mathbf{M}_{x,i+}$ ，另一次是 $\mathbf{M}_{x,i-}$ 作为待优化的多胞体结构。对于剩余的 $i \in \mathbf{J}_x$ 超平面重复上述步骤。

Ein Knoten des Baums ist in zwei Fällen Endknoten: Einmal wenn die Einhüllende der aktuellen Polytopstruktur keine Polytope mit anderen Regelgesetzen mehr enthält, wenn die Vereinigung also konvex ist. Der zweite Fall ist der, wenn die aktuelle Polytopstruktur keine Polytope mit dem Regelgesetz der zu optimierenden Polytopstruktur mehr enthält, da in diesem Fall keine weiteren Polytope zur Vereinigung mehr zur Verfügung stehen. Des Weiteren werden die üblichen Techniken angewendet, um Baumzweige, die erkennbar nicht zu einem Optimum führen, nicht weiterzuverfolgen, sondern aus dem Baum auszuschneiden.

Da bei einer praktischen Implementierung eines Explicit-MPC-Reglers nur der erste Wert der Stellgröße \mathbf{u} interessiert, findet eine Reduktion der Reglerstruktur nicht nur unter Berücksichtigung des Regelgesetzes an sich statt, sondern es werden jeweils alle Bereiche zusammengefasst, die im nächsten Abtastschritt die gleiche Stellgröße an die Strecke ausgeben. Da ein Zweipunktumrichter mit Spannungszwischenkreis, wie bereits in Kap. 9.2 erläutert, nur acht verschiedene Schaltzustände besitzt, existieren für $\mathbf{u}(t)$ lediglich acht verschiedene Werte. Für Regelungen mit Tracking, also mit beliebigen Stellgrößen, wird $\Delta\mathbf{u}(t)$ statt $\mathbf{u}(t)$ als Eingangsgröße verwendet. Folglich erhöht sich die Zahl der maximal möglichen Werte auf 27, da für die drei Halbbrücken gilt, dass sowohl Δa also auch Δb und Δc ein Element von $\{-1, 0, 1\}$ sind. Damit einhergehend gibt es nur 27 verschiedene Gruppen von Polytopen. Da jedoch die Konvexitätsbedingung ebenfalls erfüllt werden muss, ist leider nicht davon auszugehen, dass nach erfolgter Optimierung die Reglerstruktur nur noch aus 27 verschiedenen Polytopen bestehen wird; die tatsächliche Zahl wird vielmehr deutlich höher liegen.

Es sei darauf hingewiesen, dass mit dem Verfahren der optimalen Komplexitätsreduzierung berechnete Lösungen vom Prinzip her online einen höheren Rechenaufwand benötigen als ein binärer Suchbaum gemäß Kap. 9.5.3, da zwar beide Algorithmen nur diejenigen Hyperebenen berücksichtigen, die tatsächlich zu verschiedenen Steuergrößen im nächsten Abtastschritt führen, der Suchbaum aber darüberhinaus die online-Suche nach dem anzuwendenden Regelgesetz dadurch beschleunigt, dass hier anhand der Baumstruktur vorgegangen werden kann, während bei der mit der optimalen Komplexitätsreduzierung ermittelten vereinfachten Struktur nach wie vor eine erschöpfende Suche über die verbleibenden Polytope durchgeführt werden muss. Im Experiment hat sich jedoch gezeigt, dass für sehr komplexe Lösungsstrukturen eine direkte Berechnung des Suchbaums wegen des hohen Speicherbedarfs nicht möglich ist; allerdings ist es möglich, die Struktur zuerst mit Hilfe des hier beschriebenen Algorithmus zu vereinfachen und dann den binären Suchbaum zu bestimmen.

In two cases, a node of the tree is an end node: First, if the envelope of the current polytope structure contains no more polytopes with other control laws, i. e. when the combination is convex. The second case applies if the current polytope structure contains no further polytopes with the control law of the polytope structure that should be optimized, since in this case no further polytopes are available for merging. Furthermore, the usual techniques for excluding branches from the tree that will recognizable not lead to an optimum, are used.

Since for a practical implementation of an explicit MPC controller, only the first value of the sequence of actuating variables \mathbf{u} is of interest, a reduction of the controller structure is not only done under consideration of the control law itself, but all regions, which will in the next sampling cycle apply the same value of the actuating variable to the plant, are merged. As a two-level inverter with DC link has only eight different switching states, as already explained in Chapter 9.2, $\mathbf{u}(t)$ can take one out of only eight values. For controls with tracking, i. e. with any values for the actuating variables, $\Delta\mathbf{u}(t)$ is used as input variable instead of $\mathbf{u}(t)$. Hence, the maximum number of possible values increases to 27 because for the three half bridges applies that Δa is, as well as Δb and Δc , an element of $\{-1, 0, 1\}$. According to this, only 27 different groups of polytopes exist. However, since the convexity constraint has to be regarded, too, it cannot be expected that the controller structure consists of only 27 different polytopes after the optimization process has been performed; the real number of polytopes will be significantly higher.

It has to be mentioned that solutions calculated with the method of Optimal Complexity Reduction have in principle a higher online computational demand than a binary search tree according to Chapter 9.5.3, although both algorithms only consider those hyperplanes which lead to different values for the actuating variables in the next sampling cycle, but the search three furthermore accelerates the online-search for the control law to be applied through the fact that here an easier approach because of the tree-like structure is possible, while, considering the simplified structure obtained by optimum complexity reduction, still an exhaustive search over the remaining polytopes has to be done. However, in experiments it was proven that due to the high memory requirements for very complex solution structures a direct calculation of the search tree is not possible; however, it is possible to determine the structure with the help of the algorithm described here first and then to determine the binary search tree.

Notizen/Notes/笔记

两种情况下的树节点可以看作作为结束节点：一种是当前多胞体结构的包络中不再包含具有其他控制律的多胞体，而且其并集也是凸的；另一种是当前多胞体结构不再包含与待优化多胞体结构的控制律相同的多胞体，因为这一情况下不存在更多可以合并的多胞体。此外还可使用一般的技术性措施，不再继续追踪那些明显无法得到最优结果的树分支，而是直接从树中剪除。

对于显式 MPC 控制器的实际实现来说，我们仅关注控制变量序列 \mathbf{u} 的第一个元素，所以控制器结构的简化并非仅仅考虑了控制律本身，而且同样整合了那些在下一采样周期施加给被控系统相同控制变量的所有区域。如章节 9.2 所述，由于包含中间直流环节的两电平变流器仅包含 8 个不同的开关状态，所以 $\mathbf{u}(t)$ 只可能为上述数值中的一个。对于可实现跟踪的控制来说，使用 $\Delta\mathbf{u}(t)$ 代替 $\mathbf{u}(t)$ 作为输入变量，从而最大可能的数量增加至 27，因为对于三个半桥来说， Δa ， Δb 和 Δc 均可取 $\{-1, 0, 1\}$ 中元素的一个。同样，也存在 27 个不同的多胞体组。此外，由于必须满足凸性条件，因此并不能认为优化后的控制器结构也仅由 27 个不同的多胞体组成，实际的数量要明显地增多。

必须要提及的是，从原则上来看，使用最优复杂度降低方法求解要比章节 9.5.3 所述的二叉搜索树方式具有更高的在线计算需求，尽管两种算法都只考虑那些在下一个采样周期中出现不同被控变量的超平面。由于树结构本身的特性，根据所使用的控制律可以加速搜索树在线搜索过程；而对于最优复杂度降低所得到的简化结构来说，仍然需要对剩余的多胞体进行穷举搜索。实际实验表明，对于搜索树的计算方式来说，非常复杂的求解结构对计算机系统有很高的存储要求，所以是行不通的；当然，可以首先使用本节所描述的算法来简化结构，然后再使用二叉搜索树的方式。

9.5.5 Experimentelle Ergebnisse/Experimental results/实验结果

LTI-System

Zur Evaluierung der expliziten Regelung wurde der Antrieb zunächst als lineares, zeitinvariantes (LTI) System betrachtet, d. h. die hybride Struktur des Antriebs wurde vernachlässigt. Damit kann das in Kap. 7 für den mehrdimensionalen GPC-Regler hergeleitete Maschinenmodell verwendet werden; allerdings entfällt die Umrechnung in das CARIMA-Modell, da für die explizite Lösung das zeitdiskrete Zustandsraummodell verwendet wird. Auch wurde auf die in Kap. 7.3 beschriebene Berücksichtigung der Kreuzkopplung in Form bekannter Störgrößen verzichtet, so dass es sich bei dem verwendeten Modell de facto um eine diskretisierte Form des in Abb. 3.4 auf Seite 21 gezeigten vereinfachten Maschinenmodells handelt. Mit Hilfe dieses Modells kann der Zeitaufwand für eine explizite MPC-Regelung vergleichend untersucht werden. Tabelle 9.5 zeigt die benötigten Rechenzeiten für die verschiedenen Algorithmen. Zum Vergleich ist die Berechnungsdauer für einen online gerechneten MIMO-GPC-Regler ebenfalls angegeben.

Verfahren	Struktur	max. Rechenzeit
vollst. Enumeration	225 Polytope	163 μ s
Minimum-Time-Controller	9 Polytope, 2 Iterationen	27 μ s
opt. Komplexitäts-Reduzierung	60 Polytope	94 μ s
binärer Suchbaum	225 Knoten, Tiefe: 10	4,7 μ s
MIMO-GPC	– analytische Lösung –	4,1 μ s

9.5: Rechenzeit für MIMO-Stromregler

9.5.5 Experimental results

LTI system

For the evaluation of the explicit control, the drive was first considered as a linear time invariant (LTI) system, i. e. the hybrid structure of the drive was neglected. So the machine model derived in Chapter 7 for a multidimensional GPC controller can be used; however, the transformation into the CARIMA model can be omitted, as for the explicit solution the time discrete state space model is used. Additionally, the consideration of the cross coupling in form of known disturbances as described in Chapter 7.3, was omitted so that the model used is in fact a discretized form of the simplified model shown in Figure 3.4 on page 21. With the help of this model, the computation time for an explicit MPC control can be examined comparatively. Table 9.6 shows the needed computation times for the different algorithms. For reasons of comparison, the computation time for an online computed MIMO GPC controller is also given.

Type	Structure	Max. calculation time
Exhaustive search	225 polytopes	163 μ s
Minimum-Time-Controller	9 polytopes, 2 iterations	27 μ s
Opt. complexity reduction	60 polytopes	94 μ s
Binary search tree	225 nodes, branches: 10	4.7 μ s
MIMO-GPC	– analytical solution –	4.1 μ s

9.6: Calculation times for MIMO current controllers

9.5.5 实验结果

LTI 系统

为了评判显式控制的优劣，首先考察一个以线性时不变系统方式建模的驱动系统，这意味着忽略了驱动系统的混合结构。如此，即可使用第 7 章所推导的用于多维 GPC 控制器的电机模型，它显然省去了向 CARIMA 模型的变换过程，因为显式求解所需的是一个时间离散的状态空间模型。同样，也放弃使用章节 7.3 所述的以已知扰动变量形式考虑的交叉耦合因素，因为所使用的模型实际是第 21 页图 3.4 所示的简化电机模型。借助于这一模型可以对比性地考察显式 MPC 控制所需的计算时间。表 9.7 给出了不同算法下 MIMO 电流控制器所需的计算时间。为了对比需要，表中同样给出了一个在线计算的 MIMO-GPC 控制器所需的计算时间。

方法	结构	最大计算时间
完全遍历	225 个多胞体	163 μ s
最小时间控制器	9 个多胞体，2 个迭代步数	27 μ s
最优复杂度降低	60 个多胞体	94 μ s
二叉搜索树	225 个结点深度：10	4.7 μ s
MIMO-GPC	– 解析解 –	4.1 μ s

9.7: 基于 MIMO 电流控制器的计算时间

Wie unschwer zu erkennen ist, ist die modellbasierte Prädiktivregelung mit expliziter Lösung der Optimieraufgabe (s. Kap. 9.5.1) für die Antriebstechnik ohne weitere Maßnahmen nicht brauchbar, denn die vollständige Enumeration über die 225 Polytope der Reglerstruktur benötigt deutlich zu viel Zeit. Anwendbar ist hingegen der Minimum-Time-Controller (s. Kap. 9.5.2), dessen Ausführungszeit deutlich niedriger liegt, da hier lediglich zwei Iterationen mit insgesamt neun Polytopen auszuwerten sind. Nachteil ist jedoch, dass dieser Reglertyp keine optimale Regelung bezogen auf die Bewertungsfunktion bietet. Die optimale Komplexitäts-Reduzierung (s. Kap. 9.5.4) führt durch die Vereinfachung der Reglerstruktur von 225 auf nur noch 60 Polytope ebenfalls zu einer Beschleunigung der online-Suche; bedingt durch die immer noch notwendige erschöpfende Suche über die verbleibenden 60 Polytope ist die Rechenzeit allerdings weiterhin zu hoch. Unter Verwendung eines binären Suchbaums (s. Kap. 9.5.3) sinkt die online-Rechenzeit gegenüber dem Standard-Verfahren um einen Faktor von fast 35, so dass hiermit eine Implementierung problemlos möglich ist. Interessant ist dabei der Fakt, dass der Suchbaum ebenfalls 225 Elemente besitzt, d. h. genauso viel, wie die ursprüngliche Polytopstruktur. Durch die Organisation der Optimallösung als Baumstruktur sind aber im ungünstigsten Fall nur zehn Entscheidungen zu treffen; bedeutend weniger als die 225 bei der erschöpfenden Suche. Darüberhinaus ist jede Entscheidung beim Suchbaum immer nur an die Auswertung *einer* Hyperebenengleichung gekoppelt; bei der Polytopsuche müssen aber u. U. *alle* Hyperebenen eines Polytops abgefragt werden, um festzustellen ob ein gegebener Punkt innerhalb dieses Polytops liegt oder nicht.

Es lässt sich zusammenfassend feststellen, dass die Durchführung einer expliziten MPC-Regelung für einen als LTI-System modellierten elektrischen Antrieb von der Rechenzeit her kein Problem darstellt. Allerdings ist gegenüber der analytischen Lösung kein Rechenzeitgewinn feststellbar, weshalb in diesem Fall die analytische GPC-Methode vorzuziehen ist.

As it can easily be seen, model-based predictive control with explicit solution of the optimization task (see chapter 9.5.1) is not feasible for drive control without further modifications, since the complete enumeration over the 225 polytopes of the controller structure needs significantly too much time. However, a minimum-time-controller whose execution time is significantly smaller, because here only two iterations with altogether nine polytopes have to be evaluated, is feasible (see chapter 9.5.2). Its drawback is, however, that it does not provide an optimum control considering the cost function. The use of optimum complexity reduction (see chapter 9.5.4) also leads to an acceleration of the online search because of the simplification of the controller structure from 225 to only 60 remaining polytopes; due to the exhaustive search over the remaining 60 polytopes which is still necessary the computation time is still too high. When a binary search tree (see chapter 9.5.3) is used, the online calculation time can be reduced by a factor of nearly 35 compared to the standard method which makes an implementation feasible. An interesting fact is that the search tree also contains 225 elements, i. e. the same amount of elements as the original polytope structure. However, by arranging the optimum solution as a tree-like structure, in the worst case only ten decisions have to be made, significantly less than the 225 for the exhaustive search. Furthermore, every decision in the search tree is always coupled to the evaluation of only *one* hyperplane equation; for the polytope search, *all* hyperplanes of a polytope have to be evaluated, if possible, in order to determine if a given point lies within this polytope or not.

In summary, it can be stated that explicit MPC for an electrical drive modeled as LTI system considering the computation time is feasible. However, in comparison with the analytical solution, no reduction of the computation time could be determined. That is why, in this case, the analytical GPC method should be preferred.

不难看出, 在没有其他措施的前提下, 基于优化任务的显式求解(见章节 9.5.1)方式实现的模型预测控制很难被应用到驱动技术领域, 因为完全遍历控制器结构所包含的 225 个多胞体需要太多的时间。可以使用的是最小时间控制器(见章节 9.5.2), 其计算时间明显变少, 因为它仅评判了在两个迭代步数下的 9 个多胞体。然而, 这一方式的缺点是无法提供一个对于成本函数来说的最优控制。最优的复杂度降低(见章节 9.5.4)方法通过将控制器结构所包含的 225 个多胞体简化至 60 个多胞体, 如此即可加速在线搜索过程。但是, 对于简化后的多胞体数量进行穷举搜索所需的计算时间也相对较高。相对于标准方法, 通过使用二叉搜索树(见章节 9.5.3)可以将计算时间降低至大约三十五分之一, 所以实际实现起来是完全可行的。一个有趣的事实是, 二叉搜索树同样包含 225 个元素, 其数量与原始的多胞体结构数量一致。即使在最不理想的情况下, 基于树结构也仅需完成 10 次决策, 也即深度为 10, 这一数量明显小于穷举搜索下的 225。此外, 与搜索树的每一次决策有关联的始终是对单个超平面方程的判断; 而在穷举搜索下, 则需涉及多胞体的所有超平面, 其目的是确定所给定的点是否位于多胞体之内。

基于上述分析可知, 从计算时间的角度看, 以 LTI 系统方式建模的电力驱动系统应用 MPC 控制的显式求解是毫无问题的。显然, 相对于解析解的方式, 它也无法进一步降低计算时间, 这也是通常优先使用解析解形式的 GPC 方法的原因。

Hybrides System

Da mit Hilfe der in Tabelle 9.5 angegebenen gemessenen Rechenzeiten die prinzipielle Anwendbarkeit der expliziten MPC-Regelung auch für die elektrische Antriebstechnik bewiesen ist, wird nun eine *direkte* modellbasierte Prädiktivregelung mit expliziter Lösung vorgestellt. Dazu ist zunächst das in Kap. 9.3.1 erläuterte vereinfachte Maschinenmodell zu diskretisieren; die Vorgehensweise ist dabei dieselbe, wie schon in Kap. 7 beschrieben. Es ergeben sich für das zeitdiskrete, vereinfachte Maschinenmodell in Statorkoordinaten gemäß Formel (9.15) und (9.16) folgende Systemmatrizen:

$$\mathbf{A} = \begin{bmatrix} 1 - \frac{T_0}{\tau_{\sigma'}} & 0 \\ 0 & 1 - \frac{T_0}{\tau_{\sigma'}} \end{bmatrix} = \begin{bmatrix} 0,9873 & 0 \\ 0 & 0,9873 \end{bmatrix} \quad (9.79)$$

$$\mathbf{B} = \begin{bmatrix} \frac{2T_0}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & -\frac{T_0}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & -\frac{T_0}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} \\ 0 & \frac{T_0}{r_{\sigma}\tau_{\sigma'}} & -\frac{T_0}{r_{\sigma}\tau_{\sigma'}} \end{bmatrix} \quad (9.80)$$

$$= \begin{bmatrix} 0,1713 & -0,08566 & -0,08566 \\ 0 & 0,1484 & -0,1484 \end{bmatrix}$$

$$\mathbf{C} = \mathbf{I} \quad (9.81)$$

$$\mathbf{D} = \mathbf{0} \quad (9.82)$$

Wendet man auf diese Matrizen die üblichen Algorithmen zur Berechnung einer expliziten Lösung an, so erhält man eine Reglerstruktur, die weder die von der digitalen Regelung erzeugte Verzögerungszeit von einem Abtastschritt berücksichtigt, noch zur Regelung eines anderen Sollwerts als $\mathbf{u}(t) = [0 \ 0]^T$ geeignet ist. Daher müssen die Matrizen aus (9.79)–(9.82) gemäß Formel (9.56) und (9.57) zu den entsprechenden Gegenständen mit Berücksichtigung von Verzögerung und Tracking umgeformt werden:

Hybrid system

As the measured computation times given in Table 9.6 generally prove the feasibility of explicit MPC also for electrical drive technology, now a *direct* model-based predictive control scheme with explicit solution is presented. Therefore, at first, the simplified machine model explained in chapter 9.3.1 has to be discretized; the procedure is the same as it has already been described in chapter 7. For the time-discrete simplified machine model in stator coordinates according to the equations (9.15) and (9.16), the following system matrices result, see equations (9.79) to (9.82).

If the usual algorithms for the calculation of an explicit solution are applied to these matrices, a controller structure results which neither considers the delay of one sampling cycle caused by the digital control nor which is feasible for the control of another reference value than $\mathbf{u}(t) = [0 \ 0]^T$. Thus, the matrices of the equations (9.79)–(9.82) have to be transformed to the corresponding matrices under consideration of delay and tracking according to the equations (9.56) and (9.57), see equations (9.83) to (9.86).

混合系统

表 9.7 所给出的计算时间证实了显式 MPC 控制在电力驱动控制领域的可用性。接下来, 将介绍基于显式求解的直接模型预测控制。为此, 我们首先离散化章节 9.3.1 所介绍的简化电机模型, 其处理方式与第 7 章所述方法完全一致。根据方程 (9.15) 和 (9.16), 可得到定子坐标系下的时间离散后的简化电机模型的系统矩阵, 见方程 (9.79) 至 (9.82)。

如果将显式求解计算所需的一般性算法应用于上述矩阵, 那么所得到的是一个既无法考虑到数字控制方式下存在的单个采样周期的滞后, 也无法适用于除了 $\mathbf{u}(t) = [0 \ 0]^T$ 之外的其他参考值的控制器结构。为此, 需根据方程 (9.56) 和 (9.57), 首先将方程 (9.79) – (9.82) 中的矩阵转化为可考虑滞后和跟踪的相应形式, 见方程 (9.83) 至 (9.86)。

$$\mathbf{A}_{t,d} = \begin{bmatrix} 1 - \frac{T_0}{\tau_{\sigma'}} & 0 & \frac{2T_0}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & -\frac{T_0}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & -\frac{T_0}{\sqrt{3}r_{\sigma}\tau_{\sigma'}} & 0 & 0 \\ 0 & 1 - \frac{T_0}{\tau_{\sigma'}} & 0 & \frac{T_0}{r_{\sigma}\tau_{\sigma'}} & -\frac{T_0}{r_{\sigma}\tau_{\sigma'}} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (9.83)$$

$$= \begin{bmatrix} 0,9873 & 0 & 0,1713 & -0,08566 & -0,08566 & 0 & 0 \\ 0 & 0,9873 & 0 & 0,1484 & -0,1484 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B}_{t,d} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (9.84)$$

$$\mathbf{C}_{t,d} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (9.85)$$

$$\mathbf{D}_{t,d} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (9.86)$$

Notizen/Notes/笔记

Da nicht die Stellgröße selbst bewertet werden soll, sondern nur die Stellgrößenänderung, die ja dem Stellaufwand des Umrichters entspricht, sind die Bewertungsmatrizen folgendermaßen zu definieren (wobei λ den Bewertungsfaktor der Stellgrößenänderung darstellt):

$$\mathbf{Q}_{t,d} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (9.87)$$

$$\mathbf{R}_{t,d} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \quad (9.88)$$

Für das obige Modell ergeben sich Reglerstrukturen mit etwa 7500–8500 Polytopen; die genaue Zahl schwankt etwas je nach dem Wert von λ . Allgemein führt jedoch eine höhere Bewertung von $\Delta \mathbf{u}$ zu einer einfacheren Reglerstruktur. Trotzdem ist ein expliziter Regler dieser Komplexität allein aus Gründen des Speicherbedarfs nicht anwendbar, von der Rechenzeit ganz zu schweigen. Die offline-Berechnung eines Suchbaums ist ebenfalls nicht möglich, da das dafür notwendige Speichervolumen auf herkömmlichen PCs nicht verfügbar ist. Abhilfe schafft die vorherige Reduzierung der Komplexität auf etwa 1800–2200 Elemente durch Zusammenfassen von Regionen mit gleichem $\Delta \mathbf{u}$; danach ist die Berechnung eines binären Suchbaums möglich. Es ergibt sich schließlich eine Baumstruktur mit etwa 2100–2300 Knoten und einer maximalen Baumtiefe von 18, die auf dem verwendeten Rechnersystem weniger als 8 μs online-Rechenzeit benötigt. Vergleicht man diese Berechnungsdauer mit den in Tabelle 9.2 angegebenen Zeiten, so stellt man fest, dass eine modellbasierte Prädiktivregelung mit direkter Umrichtersteuerung mit Hilfe einer expliziten Lösung im Gegensatz zu den dort diskutierten online-Optimierverfahren durchaus zu betreiben ist.

As not the actuating variable itself should be evaluated, but only the change of the value of the actuating variable, which corresponds to the switching effort of the inverter, the weighting matrices have to be defined in the following way (λ is the weighting factor for the change of the value of the actuating variables), see equations (9.87) and (9.88).

For the model described above, controller structures with around 7500–8500 polytopes result; the exact number is varying a little bit depending on the value of λ . However, commonly it can be stated that a higher weighing of $\Delta \mathbf{u}$ leads to simpler controller structures. Nevertheless, an explicit controller of this complexity is not feasible due to reasons of memory demands and especially due to the needed computation time. The offline computation of a search tree is also not possible since the memory requirements for these tasks are not available on conventional PCs. A workaround is to reduce the complexity beforehand to around 1800–2200 polytopes by merging regions with the same $\Delta \mathbf{u}$; after that, the calculation of a binary search tree is possible. Finally, a tree-like structure results with around 2100–2300 nodes and a maximum tree depth of 18 which needs less than 8 μs of online computation time on the used computer system. If this computation time is compared with the times given in Table 9.3, one can see that model-based predictive control with direct control of the inverter is feasible with the help of an explicit solution, in contrast to the online optimization methods discussed there.

由于所需惩罚的并非是控制变量本身，而是对应于变流器状态变化的控制增量，所以定义如下权重矩阵（其中 λ 表示控制增量的权重系数），见方程 (9.87) 和 (9.88)。

基于上述模型下的控制器结构包含大约 7500–8500 个多胞体，其准确数值会随着 λ 值发生变化。一般来说， $\Delta \mathbf{u}$ 的权重系数越大，控制器结构则越简单。尽管如此，仅出于存储空间的原因，使得具有如此复杂性的显式控制器无法得到实际应用，更勿提及计算时间。基于搜索树的离线计算也不可行，因为传统的计算机无法提供所必需的存储空间。补救方案是通过将包含相同 $\Delta \mathbf{u}$ 的区域进行整合，从而可以将多胞体数量降低至 1800 – 2200 的范围。如此，基于二叉搜索树的计算方式将变得可行，最终得到的是包含大约 2100 – 2300 个结点的树结构，且最大的树深度为 18。基于与前面所使用的相同计算机系统，所需在线计算时间低于 8 μs 。通过与表 9.4 所给出的计算时间进行对比，可以得出结论：借助于显式求解方式下的直接模型预测控制要优于前面所讨论的在线优化方法。

Das Verhalten einer solchen Regelung wurde am Beispiel einer Stromregelung eines elektrischen Antriebs untersucht; die experimentellen Ergebnisse können Abb. 9.12 entnommen werden. Wie schon in Kap. 9.4.3, werden auch hier die Messergebnisse in stationären Koordinaten angegeben; ebenfalls wird wiederum nur ein Sprung auf $i_{s\alpha}$ geschaltet, um gleichzeitig eine Information über die Kopplung zwischen $i_{s\alpha}$ und $i_{s\beta}$ zu erhalten. Zum Vergleich mit einer konventionellen Regelung ist die gleiche Messung mit PI-Stromreglern für $i_{s\alpha}$ und $i_{s\beta}$ wiederholt worden (Abb. 9.12(c)), wobei hier natürlich zwischen Regler und Umrichter ein Modulator (PWM) eingesetzt werden muss. Um trotzdem einigermaßen vergleichbare Ergebnisse zu erzielen, ist die Schaltfrequenz des Umrichters für diese Messungen auf 1630 Hz reduziert worden, bei der direkten Regelung hingegen beträgt die Abtastfrequenz 9770 Hz.³

Wie aus Abb. 9.12 hervorgeht, besitzt die direkte explizite MPC-Regelung ein ähnliches Verhalten, wie die direkte GPC-Regelung in Abb. 9.8. Die Häufigkeit des Schaltens und damit auch die Größe der Regelabweichung werden über den Bewertungsfaktor λ beeinflusst, während sich die Dynamik des Gesamtsystems hierdurch nicht ändert. Der neue Sollwert wird immer in der kürzestmöglichen Zeit erreicht, d. h. die Dynamik der geregelten Strecke wird lediglich durch die physikalischen Grenzen des Systems selbst begrenzt. Eine geringere Gewichtung des Schaltvorgangs als $\lambda = 0,1$ ist nicht sinnvoll, da bei diesem Wert für eine gewählte Abtastzeit von etwa 100 μs bereits das Minimum an Stromverzerrungen erreicht ist, wie Abb. 9.12(a) zeigt.

Der Einfluss der Kreuzkopplung zwischen den Statorstromkomponenten $i_{s\alpha}$ und $i_{s\beta}$ macht sich nicht bemerkbar, da er bedeutend kleiner ist, als die Amplitude der Stromberschwingungen. Eine Implementierung dieser Kreuzkopplung in das Streckenmodell ist folglich nicht sinnvoll, sondern würde die Komplexität der resultierenden expliziten Reglerstruktur nur unnötig vergrößern.

Die Abb. 9.12(c) zeigt das Regelverhalten eines linearen PI-Reglers bei derselben Regelaufgabe und denselben Sollwertverläufen wie bei der direkten Regelung in den Abbildungen 9.12(a) und 9.12(b). Wie zu erwarten, sind die Stromverzerrungen bei Verwendung eines linearen Regelglieds deutlich geringer, da bei einer Ansteuerung des Umrichters mittels Modulator die Schaltzeitpunkte der Halbleiterventile innerhalb einer Abtastperiode variabel sind, während der Direktregler immer nur zu Beginn eines jeden Abtastzyklus schaltet.

³ Da bei einer modulatorbasierten Umrichtersteuerung sechs Schaltvorgänge pro Periode des Trägersignals stattfinden, wurde die Schaltfrequenz für die Messungen mit PI-Regler auf $1/6$ der Abtastfrequenz des Direktreglers eingestellt

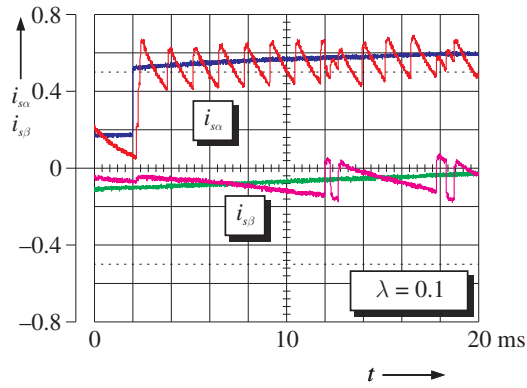
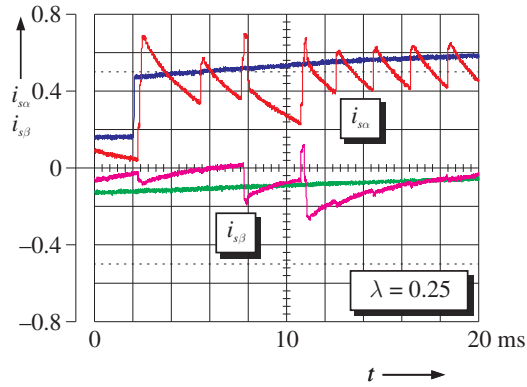
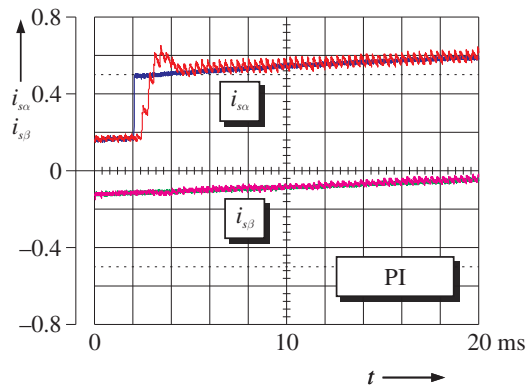
The behavior of such a control was analyzed using the example of current control of an electrical drive; the experimental results can be seen in [Figure 9.12](#). Also here, as in [chapter 9.4.3](#), measurement results are given in a stationary coordinate frame; here again, only a step change of the reference value $i_{s\alpha}$ was done, in order to get some information about the coupling between $i_{s\alpha}$ and $i_{s\beta}$. For a comparison with a conventional control scheme, the same measurement was repeated with PI current controllers for $i_{s\alpha}$ and $i_{s\beta}$ ([Figure 9.12\(c\)](#)), whereas here, of course, a modulator (PWM) has to be inserted between controller and inverter. In order to get, despite of this fact, reasonably comparable results, the switching frequency of the inverter was, for these measurements, reduced to 1630 Hz, for direct control, however, a sampling frequency of 9770 Hz was used.³

As it can be seen in [Figure 9.12](#), the direct explicit MPC has a behavior similar to the direct GPC in [Figure 9.8](#). The switching frequency and thus, the value of the control deviation can be influenced via the weighting factor λ , while the dynamics of the complete system are not affected by this. The new reference value will always be reached in the shortest possible time, i. e. the dynamics of the controlled plant are only limited by the physical limits of the system itself. Weighting the switching events less than $\lambda = 0.1$ does not make sense, because for the selected sampling time of around 100 μs , the minimum of the current disturbances is already reached, as it can be seen in [Figure 9.12\(a\)](#).

An influence of the cross coupling between the stator current $i_{s\alpha}$ and $i_{s\beta}$ cannot be figured out because it is significantly smaller than the amplitude of the current harmonics. Thus, an implementation of this cross coupling in the plant model does not make sense since it would result only in a higher, but unnecessary complexity of the explicit controller structure.

[Figure 9.12\(c\)](#) shows the behavior of the system when using a linear PI-controller for the same control task and the same sequences of reference values as they were used for direct control in [Figure 9.12\(a\)](#) and [Figure 9.12\(b\)](#). As it is to be expected, the current distortions are much smaller when a linear controller is used because using an inverter, controlled via a modulator, the switching times of the semiconductors are variable within a sampling cycle, while direct controllers can only switch at the beginning of every switching cycle.

³ Since six switching instances per period of the carrier signal take place when using modulator-based inverter control, the switching frequency for the measurements with a PI-controller was set to $1/6$ th of the sampling frequency of the direct controller.

(a) $\lambda = 0,1$ (b) $\lambda = 0,25$ 

(c) PI-Regler

接下来以电力驱动系统的电流控制为例来考察上述控制方法的特性，相应的实验结果见图 9.12。正如章节 9.4.3 所述，此处所给出的是在定子坐标系下得到的测量结果。同样，通过给定一个 $i_{s\alpha}$ 的阶跃变化来获取 $i_{s\alpha}$ 和 $i_{s\beta}$ 的耦合特性。为了与传统控制方式作对比，在基于 PI 电流控制器的方式下，我们对 $i_{s\alpha}$ 和 $i_{s\beta}$ 重复上述测量过程，测量结果见图 9.12(c)。要注意的是，此时需要在控制器和变流器之间额外使用一个调制器 (PWM)。由于上述事实，为了得到一个合理的对比结果，对于上述测量来说，变流器的开关频率被降低至 1630 Hz；对于直接控制方式来说，其采样频率为 9770 Hz³。

如图 9.12 所示，与图 9.8 作对比可知，直接的显式 MPC 拥有与直接 GPC 控制类似的特性。开关频率以及控制偏差的大小受权重系数 λ 的影响，但是整体系统的动态特性并未发生改变。系统总是会在最小时间内到达给定的新参考值，这也意味着被控系统的动态特性仅受到系统本身的物理性限制。实际上，小于 $\lambda = 0.1$ 的权重系数是不可取的，因为当 $\lambda = 0.1$ 时，所对应的大约 100 μs 的采样时间已经实现了最小的电流畸变，如图 9.12(a) 所示。

实际很难观察到定子电流 $i_{s\alpha}$ 和 $i_{s\beta}$ 之间的交叉耦合影响，因为这一影响明显小于电流谐波的幅值。正因如此，在被控对象模型中实现交叉耦合并无太大意义，而且还会额外增加最终的显式控制器结构的复杂性。

在与图 9.12(a) 和图 9.12(b) 所示的直接控制方式相同的控制任务以及相同的参考值变化条件下，图 9.12(c) 给出了使用 PI 控制器实现所得到的结果。正如所预期的那样，在使用线性控制器方式下的电流畸变明显变小，因为借助于调制器实现逆变器控制时的半导体开关器件的开关时间在单个采样周期内是变化的；而直接控制器方式下的开关时刻总是位于每个采样周期的起始点。

³ 由于在使用基于调制器的变流器控制时，每个载波信号周期内发生六次开关变化过程，所以在实际测量时，设置基于 PI 控制器方式下的开关频率大约为直接控制器采样频率的 $1/6$ 。

Nachteilig ist jedoch, dass bei einem PI-geregeltem Antrieb die Dynamik des Gesamtsystems an die Schaltfrequenz gekoppelt ist, da sie von der Abtastrate abhängt und es bei einem PI-Regler keinen Sinn macht, höher als mit der doppelten Schaltfrequenz abzutasten. Der Grund dafür liegt darin, dass bei der Parametrisierung eines PI-Reglers die durch die digitale Regelung hervorgerufene Totzeit berücksichtigt werden muss, was zu einer trägeren Regelung führt. Diese Totzeit steht aber in direktem Zusammenhang zur gewählten Abtastrate. Ist nun eine niedrige Schaltfrequenz gewünscht, z. B. bei einem Umrichter hoher Leistung, so verschlechtert sich bei der Stromregelung mit PI-Reglern die Dynamik des Gesamtsystems, da mit der niedrigen Schaltfrequenz auch eine geringe Abtastrate einhergehen muss. Bei Verwendung eines expliziten DMPC-Reglers ist es jedoch möglich, eine vergleichsweise hochfrequente Abtastung zu wählen und durch entsprechend starke Bewertung der Schaltvorgänge in der Kostenfunktion trotzdem eine niedrige mittlere Schaltfrequenz zu gewährleisten. Darüberhinaus kann bei modellbasierten Prädiktivreglern die Regelungstotzeit im Reglerentwurf berücksichtigt werden, weshalb sie einen wesentlich geringeren Einfluss auf die Dynamik des geregelten Gesamtsystems hat.

However, a drawback of a PI controlled drive is that the dynamics of the complete system are coupled to the switching frequency, because the switching frequency is dependent on the sampling frequency and for a PI-controller it does not make sense to use a higher sampling frequency than twice the switching frequency. The reason for this is the fact that, for parameterizing a PI-controller, the dead time caused by the digital control has to be considered, which leads to a more sluggish control. However, this dead time is directly related to the selected sampling rate. If a low switching frequency is desired, e.g. for a high power converter, the dynamics of the complete system are degraded if current control is done by PI-controllers because the low switching frequency also causes a low sampling rate. However, when using an explicit DMPC controller, it is possible to use a comparably high sampling rate and to ensure, despite of this, a low average switching frequency by using a considerably high weighing of the switching events in the cost function. Furthermore, in model-based predictive controllers the dead time caused by digital controllers can be considered in the controller design. Thus, this dead time has a significantly lesser influence on the dynamics of the controlled system.

基于 PI 控制器实现电力驱动的缺点是整个系统的动态特性与开关频率密切相关，因为开关频率取决于采样频率。对于 PI 控制器来说，使用高于两倍开关频率的采样频率是没有意义的。其原因在于，在对 PI 控制器进行参数设计时，必须考虑由于数字控制而形成的死区时间，这将导致控制响应变得迟缓。而死区时间又与所选择的采样率直接相关。如果期望的是一个较低的开关频率，例如在大功率变流器的应用场合，此时基于 PI 控制器实现的电流控制将会严重影响整个系统的动态特性，因为低的开关频率要求的是低的采样频率。然而，当使用显式 DMPC 控制器时，有可能使用一个相对较高的采样频率，并通过在成本函数中使用一个对于开关变化来说较大的权重系数来确保较低的平均开关频率。此外，在基于模型的预测控制器时，可以在控制器设计时考虑到由于数字控制所引起的死区时间，如此即可明显降低死区时间对被控系统动态特性所造成的影响。

10 Verwandte Regelstrukturen/Related Control Structures/相关的控制结构

In der Literatur trifft man immer wieder auf Regelverfahren, die dort teilweise als modellbasierte Prädiktivregler bezeichnet oder gar mit einigen prädiktiven Regelstrategien gleichgesetzt werden. Besonders sind hier *Internal Model Control* und die *Lineare Quadratische Regelung* zu nennen. In der Tat weisen die beiden genannten Regler einige nicht zu übersehende Ähnlichkeiten mit MPC auf. In den folgenden Abschnitten sollen diese Ähnlichkeiten, aber auch die Unterschiede dieser Verfahren im Vergleich zu einer echten modellbasierten Prädiktivregelung aufgezeigt werden.

10.1 Internal Model Control/Internal Model Control/内部模型控制

Internal Model Control (IMC) ist eigentlich kein *Regelverfahren*, sondern ein spezielles *Entwurfsprinzip* für eine Regelung. Allerdings wird es, vermutlich wegen des Namens, von einigen Autoren, z. B. García/Prett/Morari [40], den MPC-Reglern zugeordnet. Auch in der sehr ausführlichen Beschreibung des Verfahrens bei Lunze [84] wird es unter der Bezeichnung „Modellbasierte Regelung“ eingeführt; allerdings wird dort in diesem Zusammenhang nicht von prädiktiver oder vorausberechnender Regelung gesprochen. Um zu zeigen, dass die Grundideen von IMC sich von denen, auf denen MPC fußt, trotz einiger Gemeinsamkeiten doch deutlich unterscheiden, sollen sie hier kurz skizziert werden. Für weiterführende Informationen wird der interessierte Leser auf das Buch von Lunze [84, Kap. 12.2] verwiesen; ein detaillierter Vergleich zwischen MPC und IMC ist bei Dittmar/Pfeiffer [34, Kap. 2.4.2] zu finden. Von Harnefors/Nee [48] wird eine Anwendung von IMC zur Regelung einer Asynchronmaschine ausführlich beschrieben.

Das Grundprinzip eines Reglerentwurfs mit IMC ist, dass ein inneres Modell der zu regelnden Strecke in den Regler selbst integriert wird. Damit steht es im Gegensatz zu herkömmlichen Regelungen, wo ein Streckenmodell nur zur Parametrisierung des Reglers verwendet wird, das eigentliche Regelgesetz selbst jedoch kein Modell beinhaltet. Abb. 10.1 zeigt die typische Struktur eines IMC-Reglers.

In literature, one can quite often find control methods being partly denoted as model-based predictive controllers or which are said to be equal with some predictive control

strategies. Here, especially *Internal Model Control* and *Linear Quadratic Control* have to be mentioned. In fact, both controllers mentioned above have some similarities compared to MPC that cannot be overlooked. In the following sections, these similarities, but also the differences of these methods compared to a real model-based predictive control strategy, will be presented.

10.1 Internal Model Control

Internal Model Control (IMC) is actually no *control method*, but a special *design principle* for a control. However, it is, probably because of its name, assigned to MPC controllers by some authors, e. g. García/Prett/Morari [40]. In a very detailed description of the method by Lunze [84], it is also introduced under the name “model-based control”; however, in this context it is not assumed to be a predictive or precalculating control. In order to show that the basic ideas of IMC are, despite some similarities, significantly different from the ones on which MPC is based, they will shortly be outlined here. For further information, the interested reader is referred to the book of Lunze [84, Chapter 12.2]; a detailed comparison between MPC and IMC can be found in Dittmar/Pfeiffer [34, Chapter 2.4.2]. By Harnefors/Nee [48], the application of IMC for current control of an induction machine is described in detail.

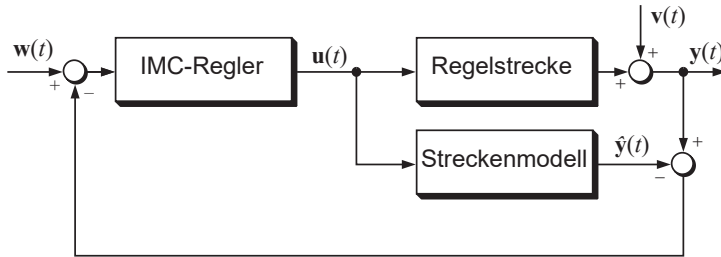
The basic principle of IMC controller design is to integrate an inner model of the plant to be controlled into the controller itself. Thus, it is contrary to conventional controls which use a plant model only for parameterizing the controller; the actual control law itself does, however, not contain a model. Figure 10.1 shows the typical structure of an IMC controller.

在不同文献中，人们会反复遇到一些同样声明是基于模型的预测控制方法，或者被看作是预测控制策略。此处要特别提及的是内部模型控制和线性二次控制。事实上，上述两种控制方法与预测控制方法之间存在着不可忽视的相似之处。本章将具体阐述内部模型控制和线性二次控制与模型预测控制方法之间的相同点和不同点。

10.1 内部模型控制

内部模型控制（英文缩写 IMC）并非是一种控制方法，而是应用于控制领域的一种特殊设计原则。显然由于名称的缘故，使得一些学者将其纳入了模型预测控制器的范畴，如学者 García/Prett/Morari [40]。此外，学者 Lunze 在其著作 [84] 对上述方法做了详细介绍，并称之为基于模型的控制，而并未将上述方法看作是预测控制。为了表明内部模型控制与模型预测控制之间基本思想的区别，尽管它们之间确实存在相似之处，随后将简要设计一种内部模型控制器。更为深入的内容请参考学者 Lunze 的著作 [84, 章节 12.2]；MPC 和 IMC 之间更为详尽的对比可参考学者 Dittmar/Pfeiffer 的著作 [34, 章节 2.4.2]。此外，学者 Harnefors/Nee 详细描述了一种将内部模型控制应用至电机控制的方法 [48]。

内部模型控制器设计的基本原则是将被控对象的内部模型整合到控制器本身。从而，不同于那些基于被控对象模型进行控制器参数调整的传统控制器类型，其控制器本身并不包含模型信息。图 10.1 给出了 IMC 控制器的典型结构。



10.1: IMC-Struktur/IMC structure/IMC 结构

Das Streckenmodell wird der Regelstrecke parallelgeschaltet, so dass nicht mehr die Regelgröße $\mathbf{y}(t)$, sondern die Differenz zwischen echter und per Modell berechnetem Ausgangsvektor $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$ auf den Regler zurückgeführt wird. Ein Wert dieser Rückführung ungleich Null kann nun zwei Ursachen haben:

1. Eine Störgröße $\mathbf{v}(t)$ wirkt auf die Strecke ein.
2. Ein Modellfehler liegt vor.

Geht man davon aus, dass das Modell fehlerfrei ist, so ist bei fehlender Störgröße $\mathbf{v}(t)$ die Rückkopplung Null. Damit ist der IMC-Regler in diesem Fall eine reine Vorwärtssteuerung und man erhält einen ganzheitlichen Ansatz zur direkten Wahl der Reglerstruktur aus dem gewünschten Verhalten des gesamten Systems.

Tritt eine Störgröße $\mathbf{v}(t)$ auf, so ist die Differenz $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$ ungleich Null. Der Regler reagiert auf dieses Signal, indem der Reglerausgang $\mathbf{u}(t)$ entsprechend verändert wird. Durch diese Fähigkeit, auf Störeinflüsse zu reagieren, unterscheidet sich der IMC-Regler von einer reinen Vorwärtssteuerung. Dabei gibt es keinerlei Stabilitätsprobleme; sind Strecke und IMC-Regler stabil, so ist auch der geschlossene Regelkreis stabil.

Um nun eine praktische Realisierung eines IMC-Reglers durchzuführen, wird die IMC-Reglerstruktur (Abb. 10.1) in die konventionelle Struktur nach Abb. 10.2 überführt; es ergibt sich wieder eine Rückführung der Regelgröße $\mathbf{y}(t)$. Das gestrichelte Kästchen entspricht dabei dem konventionellen Regler $F_R(s)$. Sind das Modell der Regelstrecke bekannt und der IMC-Regler entsprechend dem gewünschten Verhalten ausgelegt (s. o.), so kann $F_R(s)$ leicht bestimmt werden.

The plant model is connected in parallel to the plant which leads to the fact that not the control variable $\mathbf{y}(t)$, but the difference between the real and the output vector calculated via the model, $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$, is fed back to the controller. A value of this feedback unequal to zero can be caused by two different things:

1. A disturbance $\mathbf{v}(t)$ affects the plant.
2. A model error is present.

Assuming that the model is not affected with an error and that there is no disturbance $\mathbf{v}(t)$ existing, the feedback is also equal to zero. So the IMC controller is, in this case, a simple open-loop control and offers a holistic approach for the direct selection of the controller structure by deducing it from the desired behavior of the entire system.

If a disturbance $\mathbf{v}(t)$ occurs, the difference $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$ will be unequal to zero. The controller reacts to this signal by correspondingly changing the controller output $\mathbf{u}(t)$. This ability to react to disturbances is a difference of an IMC controller compared to a mere feedforward control. No stability problems will occur; if the plant and the IMC controller are stable, the closed control loop will also be stable.

For a practical realization of an IMC controller, the IMC controller structure (Figure 10.1) is transformed into the conventional structure according to Figure 10.2; again a feedback of the control variable $\mathbf{y}(t)$ results. Thereby, the dashed box corresponds to a conventional controller $F_R(s)$. If the plant model is known and the IMC controller is designed according to the desired behavior (see above), $F_R(s)$ can easily be determined.

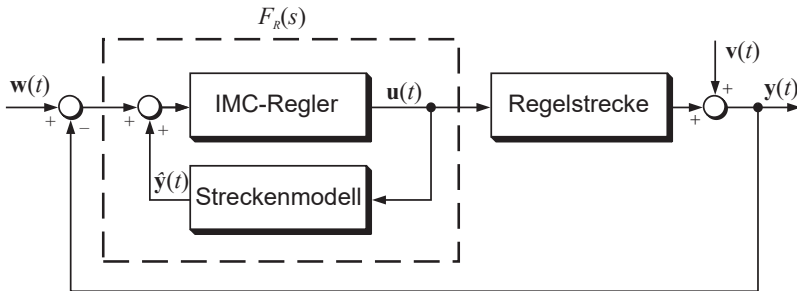
如图所示，被控对象模型与被控对象以并联方式连接，从而反馈给控制器的不再是被控变量 $\mathbf{y}(t)$ ，而是实际被控对象与其模型输出变量之间的差值 $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$ 。引起上述差值不等于零的原因可能是：

1. 一个扰动变量 $\mathbf{v}(t)$ 影响了被控对象；
2. 实际存在模型误差。

假设不存在模型误差，那么当不存在扰动变量 $\mathbf{v}(t)$ 时，反馈量则为零。换言之，在上述条件下得到的 IMC 控制器可以看作是一个纯粹的开环控制器。从而，可以根据所期望的系统特性很容易得到一个用于控制器结构选择的统一方法。

如果存在扰动变量 $\mathbf{v}(t)$ ，那么，差值 $\mathbf{y}(t) - \hat{\mathbf{y}}(t)$ 则不为零。IMC 控制器通过改变其输出变量 $\mathbf{u}(t)$ 以对上述非零反馈量作出反应。IMC 控制器可以对干扰作出反应的能力表明了它不同于纯粹的开环控制器。在这种情况下不存在稳定性问题；如果被控对象和 IMC 控制器是稳定的，那么整个闭环控制回路也是稳定的。

为了实际实现 IMC 控制器，可首先将图 10.1 所示的 IMC 典型控制结构转换为传统形式的结构，如图 10.2 所示。图中虚线框部分对应的是传统控制器 $F_R(s)$ 。假使被控对象模型是已知的，那么即可根据期望的系统特性来设计 IMC 控制器，从而可以很容易地确定 $F_R(s)$ 。



10.2: IMC-Struktur, umgeformt auf klassische Reglerstruktur/IMC structure transformed to the classical controller structure/IMC 结构, 转化为典型的控制器结构

In der Realität kann natürlich nicht davon ausgegangen werden, dass das Streckenmodell in allen Fällen exakt mit der Realität übereinstimmt. Die Möglichkeiten, dies in einem IMC-Regler zu berücksichtigen, werden in den Veröffentlichungen von Harnetors/Nee [48] und Zafriou/Morari [123] beschrieben und daher hier nicht weiter ausgeführt. Dabei ist auch ein adaptives Streckenmodell denkbar.

Gegenüber konventioneller Regelung mit linearen Reglern besitzt IMC eine Reihe von Vorteilen; u. a. ist der Reglerentwurf bedeutend einfacher. Rechnet man beispielsweise die IMC-Struktur in eine konventionelle Reglerstruktur nach Abb. 10.2 um und verwendet einen PI-Regler, so werden die Verstärkung und die Integrationszeit des Reglers direkt durch Maschinenparameter und die gewünschte Regelbandbreite ausgedrückt. Darüberhinaus ist die Stabilität des Gesamtsystems immer gewährleistet, wenn sowohl das zu regelnde System im offenen Kreis als auch der IMC-Regler bereits stabil sind.

Obwohl Internal Model Control offensichtlich ein attraktives Entwurfsprinzip zu sein scheint, treten in der Realität oft Schwierigkeiten auf, die dazu führen, dass sich ein IMC-Regler praktisch nicht realisieren lässt. Neben der Stabilität des eigentlichen IMC-Reglers muss nämlich noch gewährleistet sein, dass das Regelgesetz kausal ist. Dittmar und Pfeiffer [34, Kap. 2.4.2] zeigen, dass selbst für eine Regelstrecke, die lediglich aus einem simplen PT_1 -Glied mit Totzeit besteht, kein realisierbarer IMC-Regler gefunden wird, da die Invertierung der Totzeit dazu führt, dass zur Bestimmung der aktuellen Stellgröße zukünftige Werte der Regelabweichung benötigt werden. Soll ein IMC-Regler dennoch zum Einsatz kommen, muss das Modell der zu regelnden Strecke dergestalt vereinfacht werden, dass die Bestimmung eines praktisch realisierbaren Reglers gemäß der Grundsätze von IMC möglich ist, was in diesen Fällen aber logischerweise zu einer nicht-optimalen Regelung führt.

In reality, one cannot assume that the plant model is in all cases equal to the real plant. The possibilities to consider this in an IMC controller are described in the papers by Harnefors/Nee [48] and Zafiriou/Morari [123] and thus, they are not further discussed here. Thereby, also an adaptive plant model is possible.

Compared to conventional control with linear controllers, IMC has some advantages; among other things, the controller design is significantly easier. If, e. g., the IMC structure is translated into a conventional controller structure according to Figure 10.2 and a PI-controller is used, the proportional and integral gains of the controller are directly expressed in terms of the machine parameters and the desired controller bandwidth. Furthermore, the stability of the complete system can always be granted if the system to be controlled is open-loop stable and if the IMC controller is stable, too.

Although Internal Model Control apparently seems to be an attractive design principle, in reality some problems occur which lead to the fact that an IMC controller can practically not be realized. Besides stability of the actual IMC controller itself, causality of the control law has to be guaranteed, too. Dittmar/Pfeiffer [34, Chapter 2.4.2] show that even for a plant only consisting of a simple PT_1 -block with dead time no realizable IMC controller can be found, since the inversion of the dead time leads to the fact that, for the determination of the current value of the actuating variable, future values of the control deviation are needed. If an IMC controller should be used despite this fact, the model of the plant to be controlled has to be simplified in such a way that the design of a practically realizable controller according to the basics of IMC is possible, which consequently leads to a non-optimum control in this case.

事实上,在大多情况下,被控对象模型难以准确描述真实被控对象,也就是说会存在一定的模型误差。可将模型误差因素考虑至 IMC 控制器的相关内容请参考学者 Harnefors/Nee [48] 和学者 Zafiriou/Morari [123] 的研究论文。由此可知,一种自适应的被控对象模型也是可行的。

相较于使用线性控制器的传统控制方式,IMC 具有一系列的优势,其中最大的优势是控制器的设计明显简单。例如,如果将图 10.1 所示 IMC 结构转换为如图 10.2 所示的传统形式的控制器结构,并使用 PI 控制器,那么即可根据电机参数和期望的控制带宽来确定控制器增益和时间常数。另外,只要被控系统在开环状态以及使用 IMC 控制器时是稳定的,那么也可同样保证转换后的整个系统的稳定性。

尽管内部模型控制看似是一个具有吸引力的设计原则,但在实际情况下会经常出现各种困难,使得很难实现一个简单的 IMC 控制器。除了要保证 IMC 控制器的稳定性之外,还要确保控制律的因果性。学者 Dittmar/Pfeiffer 指出 [34, 章节 2.4.2],即使被控对象仅由一个简单的含有死区时间的一阶滞后环节组成,也无法找到一个可行的 IMC 控制器,因为对死区时间求逆意味着需要使用未来时刻的控制偏差来确定当前的控制变量。如果仍然要使用 IMC 控制器,那么则必须简化被控对象模型,以使得依据 IMC 的基本原理可以确定一个可实际实现的控制器。但逻辑上来看,上述情况下得到的是一个非最优的控制器。

10.2 Lineare Quadratische Regelung/ Linear Quadratic Control/线性二次控制

Vereinzelt hört man die Meinung, GPC, eine in den Kapiteln 6 und 8 für ein- bzw. mehrdimensionale Regelstrecken vorgestellte Untergruppe der modellbasierten Prädiktivregler, sei identisch mit *Linearer Quadratischer Regelung*, abgekürzt *LQR*. Dies ist jedoch definitiv *nicht* der Fall. Die beiden Verfahren sind zwar einander ähnlich und man erhält unter bestimmten Bedingungen mit beiden Methoden die gleichen Ergebnisse, es handelt sich aber eindeutig um zwei unterschiedliche Regelverfahren.

10.2.1 Funktionsweise von LQR/Functional principle of LQR/LQR 工作原理

Bevor die beiden Verfahren GPC und LQR miteinander verglichen werden können, ist es notwendig, die grundlegenden Prinzipien linearer quadratischer Regelung kurz zu erläutern. Der Einfachheit halber wird hier nur der prinzipielle Entwurf einer LQ Regelung für ein deterministisches, zeitinvariantes System beschrieben. Bei tiefergehendem Interesse sei der Leser auf weiterführende Literatur verwiesen. Eine ausführliche Beschreibung des LQR-Prinzips findet sich z. B. bei Isermann [58, Kap. 8.1] oder Kanjilal [62, Kap. 13]; eine gute Erläuterung für den Fall eines zeitkontinuierlichen LQ Reglers hat Lunze [85, Kap. 7] verfasst.

Lineare quadratische Regelung basiert auf einem Zustandsraummodell des zu regelnden Systems gemäß Gln. (9.15) und (9.16) auf Seite 281 sowie der dazugehörigen Kostenfunktion (9.18).¹ Da bei einer LQR-Regelung immer von unbeschränkten Systemgrößen ausgegangen wird, gilt hier jedoch $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^p$ und $\mathbf{u} \in \mathbb{R}^m$; darüberhinaus existiert bei LQR nur ein einziger Horizont. Demnach sind $N_1 = 0$ und $N_2 = N_p = N_u = N$ zu setzen. Um Verwechslungen mit der im Folgenden eingeführten Wichtungsmatrix \mathbf{P} zu vermeiden, wird die Bewertung des Endwerts des Zustandsvektors hier mit \mathbf{S} bezeichnet.

10.2 Linear Quadratic Control

Sometimes, it is told that Generalized Predictive Control, a subset of model-based predictive controllers presented in Chapter 6 and 8 for uni- and multidimensional plants, is identical to *Linear Quadratic Control*. However, this is *definitely not* true. Both methods are similar and in some special cases, one can obtain the same results with both methods, although these are clearly two different control strategies.

¹ Um die mathematische Darstellung von LQR mit den Formeln in Kap. 9 konsistent zu halten, wird hier trotz zeitdiskreter Form auf den Index d verzichtet.

10.2.1 Functional principle of LQR

Before comparing the methods GPC and LQR, it is necessary to explain the basic principles of Linear Quadratic Control. For reasons of simplicity, only the basic design principle of an LQ control for a deterministic time invariant system is described. The interested reader shall be referred to further literature. A detailed description of the LQR principle can be found e. g. in the publication of Isermann [58, Chapter 8.1] or Kanjilal [62, Chapter 13]; a good explanation for the case of a time-continuous LQ controller has been written by Lunze [85, Chapter 7].

Linear Quadratic Control is based on a state space model of the plant according to the equations (9.15) and (9.16) on page 281 as well as the corresponding cost function (9.18)¹. Since for an LQR control it is always assumed that the system variables are not constrained, the following statements are valid: $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^p$ and $\mathbf{u} \in \mathbb{R}^m$; furthermore, for LQR, only one single horizon exists. Thus, $N_1 = 0$ and $N_2 = N_p = N_u = N$. In order to avoid mixing up with the weighting matrix \mathbf{P} that will be introduced in the following, the weighing of the final value of the state vector is denoted with \mathbf{S} here.

10.2 线性二次控制

偶尔人们会听到下述观点：在第 6 章和第 8 章所介绍属于模型预测控制一大子类型的分别适用于单维和多维被控对象的广义预测控制（英文缩写 GPC）其实与线性二次控制（英文缩写 LQR）是等效的。但事实并非如此，虽说 GPC 与 LQR 确实有许多相似之处，并且在某些条件下，基于这两种方法甚至可以得到相同的结果，但它们明显属于两种完全不同的控制方法。

10.2.1 LQR 基本原理

在对比 GPC 与 LQR 之前，有必要简要阐述线性二次控制的基本工作原理。为简化起见，此处仅描述适用于一种确定性时不变系统的线性二次控制。更为深入的内容，读者可参考以下专业书籍：学者 Isermann 的著作 [58, 章节 8.1] 和学者 Kanjilal 的著作 [62, 第 13 章]，二者均详细描述了 LQR 基本原理；学者 Lunze 的著作 [85, 第 7 章] 则阐述了一种线性连续的 LQ 控制器。

线性二次控制基于第 281 页方程 (9.15) 和 (9.16) 所表示的状态空间模型，以及方程 (9.18) 所表示的成本函数¹。对 LQR 控制来说，由于始终假定的是不受约束的系统变量，所以满足条件： $\mathbf{x} \in \mathbb{R}^n$ ， $\mathbf{y} \in \mathbb{R}^p$ 和 $\mathbf{u} \in \mathbb{R}^m$ 。此外，LQR 仅存在一个所谓的步长，因此设置 $N_1 = 0$ 和 $N_2 = N_p = N_u = N$ 。为了避免与后面引入的权重矩阵 \mathbf{P} 相混淆，此时使用符号 \mathbf{S} 来代替之前用于惩罚状态向量终值的符号 \mathbf{P} 。

¹ In order to keep the mathematical representation of LQR consistent to the equations in Chapter 9 the index d is not used despite of the time-discrete form.

¹ 为了使得 LQR 的数学表述与第 9 章所使用的相关方程保持一致，即使是时间离散的形式，本部分也放弃使用下标 d 。

Die Kostenfunktion lautet demnach:

$$\begin{aligned}
 J(N) &= \mathbf{x}(t+N)^T \mathbf{S} \mathbf{x}(t+N) \\
 &+ \sum_{j=1}^N \left(\mathbf{x}(t+j-1)^T \mathbf{Q} \mathbf{x}(t+j-1) \right. \\
 &\quad \left. + \mathbf{u}(t+j-1)^T \mathbf{R} \mathbf{u}(t+j-1) \right)
 \end{aligned} \tag{10.1}$$

Ziel ist es nun, ein optimales Regelgesetz zu erhalten, über welches sich gemäß Gleichung

$$\mathbf{u}(t) = -\mathbf{K}(t)\mathbf{x}(t) \tag{10.2}$$

optimale, zukünftige Stellgrößen berechnen lassen. Dabei bezeichnet \mathbf{K} die sogenannte *Reglermatrix* des LQ Reglers. Gesucht ist nun eine solche Reglermatrix \mathbf{K} , für welche die Kostenfunktion (10.1) minimiert wird.

Nach der Bellman'schen Optimalitätsbedingung [9] ist das Ende einer optimalen Folge ebenfalls optimal. Folglich können, wenn der letzte Wert einer optimalen Folge bekannt ist, rückwärts die anderen Elemente ausgerechnet werden. Zur Lösung des Optimierungsproblems betrachtet man daher nur die letzte Stufe der Kostenfunktion (10.1):

$$J_N = \mathbf{x}(t+N)^T \mathbf{P}_N \mathbf{x}(t+N) \quad \mathbf{P}_N = \mathbf{S} \tag{10.3}$$

Dabei ist \mathbf{P} die neue Wichtungsmatrix. Für obige Optimieraufgabe bestimmt man durch Nullsetzen der Ableitung der Bewertungsfunktion J_N einen optimalen Wert für $\mathbf{u}(t+N-1)$:

$$\begin{aligned}
 \mathbf{u}(t+N-1) &= -(\mathbf{B}^T \mathbf{S} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{S} \mathbf{A} \mathbf{x}(t+N-1) \\
 &= -\mathbf{K}_{N-1} \mathbf{x}(t+N-1)
 \end{aligned} \tag{10.4}$$

Nun können rekursiv die restlichen Gleichungen für $\mathbf{u}(t+j)$, $j = (N-2) \dots 0$ und damit auch Formeln für die übrigen Elemente von \mathbf{K} und \mathbf{P} bestimmt werden.

Im allgemeinen Fall ist davon auszugehen, dass die Reglermatrix nicht konstant ist. Unter Verwendung eines unendlichen Optimierungshorizonts erhält man jedoch ein zeitinvariantes Regelgesetz:

$$\mathbf{u}(t) = -\mathbf{K} \mathbf{x}(t) \tag{10.5}$$

In diesem Fall ergibt sich die Reglermatrix zu

$$\mathbf{K} = (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A} \tag{10.6}$$

mit \mathbf{P} als Lösung der Matrix-Riccati-Gleichung

$$\mathbf{P} = \mathbf{Q} + \mathbf{A}^T \mathbf{P} (\mathbf{I} - \mathbf{B} (\mathbf{R} + \mathbf{B}^T \mathbf{P} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{P}) \mathbf{A} \tag{10.7}$$

Thus, the cost function can be written as equation (10.1). Now, the intention is to obtain an optimum control law with which according to the equation (10.2) optimum future values for the actuating variables can be calculated. Thereby, \mathbf{K} denotes the so called *controller matrix* of the LQ controller. Now, such a controller matrix \mathbf{K} that minimizes the cost function (10.1) has to be found.

According to the optimality constraint by Bellman [9], the end of an optimum sequence is also an optimum. Consequently, if the last value of an optimum sequence is known, the other elements can be calculated backwards. Thus, for the solution of the optimization problem, only the last step of the cost function (10.1) is considered, see equation (10.3). Thereby, \mathbf{P} is the new weighting matrix. For the above optimization task, an optimum value for $\mathbf{u}(t + N - 1)$ is determined by setting the derivation of the cost function J_N equal to zero, see equation (10.4). Now, the remaining equations for $\mathbf{u}(t + j)$, $j = (N - 2) \dots 0$ and with these, also the equations for the remaining elements of \mathbf{K} and \mathbf{P} can be determined recursively.

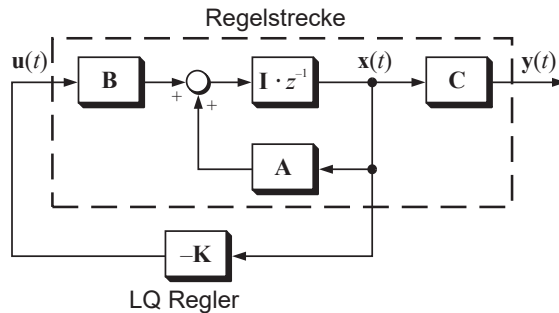
Commonly, one can be sure that the controller matrix is not constant. However, if an infinite optimization horizon is used, a time invariant control law can be obtained, see equation (10.5). In this case, the controller matrix results in equation (10.6), with \mathbf{P} as a solution of the matrix Riccati equation (10.7).

此时，成本函数可以写为方程 (10.1)。当前的目标是获得一个最优的控制律，基于此控制律并结合方程 (10.2)，即可计算出最优的、未来时刻的控制变量。式中的 \mathbf{K} 表示 LQ 控制器的控制矩阵。接下来要寻找的就是这样一个可以使成本函数 (10.1) 最小化的控制矩阵 \mathbf{K} 。

基于 Bellman 最优化条件 [9] 可知：最优路径的子路径也同样是最佳的。因此，如果最优序列的最后一个数值是已知的，那么即可前向递推出最优序列的其他元素。为了求解最优化问题，可首先考察成本函数 (10.1) 的最后一步，见方程 (10.3)，其中， \mathbf{P} 是新的权重矩阵。对于上述优化问题，可以通过将成本函数 J_N 的导数置零来确定一个最优的 $\mathbf{u}(t + N - 1)$ ，见方程 (10.4)。接下来，即可以递归的方式逐一求出其余的被控变量 $\mathbf{u}(t + j)$, $j = (N - 2) \dots 0$ 的表达式，以及 \mathbf{K} 和 \mathbf{P} 剩余元素的表达式。

一般来说，控制矩阵并非保持不变。然而，在使用了一个无限大的优化步长条件下，可以得到一个时不变的控制律，表达式见方程 (10.5)。其中，控制矩阵表达式为方程 (10.6)，式中的 \mathbf{P} 是矩阵黎卡提方程 (10.7) 的解。

Die Lösung der Gl. (10.7) erfolgt über die oben erwähnte rekursive Berechnung. Danach ist die Reglermatrix \mathbf{K} zur Vorausbestimmung optimaler, zukünftiger Stellgrößenfolgen bekannt. Abb. 10.3 zeigt die Struktur der kompletten LQ Regelung. Der Einfachheit halber wurde die Durchgangsmatrix zu $\mathbf{D} = \mathbf{0}$ angenommen, was für die meisten antriebstechnischen Fälle zutreffend sein wird. Beachten Sie außerdem, dass bei dem hier gezeigten LQ Regler die Zustandsgrößen zu Null geregelt werden, d. h. ein Sollwert außer dem Nullvektor ist nicht möglich. Für veränderliche Führungsgrößen muss statt dem Systemzustand \mathbf{x} die Regelabweichung $\mathbf{y} - \mathbf{w}$ bzw. $\mathbf{x} - \mathbf{w}$ minimiert werden. Die entsprechende Vorgehensweise ist analog zu der in Kap. 9.5 auf Seite 291 ff. gezeigten.



10.3: Struktur einer LQ Regelung ($\mathbf{w} = 0$, siehe Text)/Structure of LQ control ($\mathbf{w} = 0$, see text)/LQ 控制的结构 ($\mathbf{w} = 0$, 见正文)

10.2.2 GPC und LQR/GPC and LQR/GPC 和 LQR

Vergleicht man die lineare quadratische Regelung mit GPC, so lassen sich durchaus einige Gemeinsamkeiten feststellen (s. auch [28, 40, 62]):

- Beide Verfahren verwenden ein lineares Modell des zu regelnden Systems und sowohl GPC als auch LQR minimieren eine skalare, quadratische Kostenfunktion.
- Die Lösung des Optimierungsproblems erfolgt in beiden Fällen analytisch, d. h. durch Differenzieren der Kostenfunktion nach der Stellgröße und Suchen der Nullstelle.
- Unter bestimmten Bedingungen liefern sowohl LQR als auch LRPC identische Ergebnisse.
- Sowohl LQ Regelungen mit finitem Horizont als auch GPC-Regler minimieren die gleiche Kostenfunktion und ergeben gleiche Werte für die Stellgrößen. Folglich besitzen sie auch die gleichen Stabilitätseigenschaften.

The solution of equation (10.7) is done via the recursive calculation stated above. Afterwards, the controller matrix \mathbf{K} for the advance calculation of optimum future sequences of values for the actuating variables is known. Figure 10.3 shows the structure of the complete LQ control. For reasons of simplicity, the feedforward matrix is considered to be $\mathbf{D} = \mathbf{0}$, an assumption that can be made for most cases in drive control. Furthermore, it has to be considered that for the LQ controller shown here, the state variables are controlled to be zero, i. e. a reference value different from the zero vector is not possible. For changing reference values, the control deviation $\mathbf{y} - \mathbf{w}$ or $\mathbf{x} - \mathbf{w}$, respectively, has to be minimized instead of the system state \mathbf{x} . The corresponding procedure is analog to the one shown in Chapter 9.5 on page 291 et seqq.

10.2.2 GPC and LQR If Linear Quadratic Control is compared to GPC by all means, some similarities can be found (see also [28, 40, 62]):

- Both methods use a linear model of the plant and GPC as well as LQR minimize a scalar quadratic cost function.
- The optimization problem is solved analytically in both cases, i. e. by differentiating the cost function with regard to the actuating variable and then by searching the zero point.
- Under certain circumstances LQR as well as LRPC deliver identical results.
- LQ controllers with finite horizon as well as GPC controllers minimize the same cost function, and the same values for the actuating variables are obtained. Thus, they have the same stability characteristics.

可使用前述递归方式求解方程 (10.7)。随后，即可确定用于预测最优的、未来时刻的控制变量序列的控制矩阵 \mathbf{K} 。图 10.3 给出了一个完整的线性二次控制结构。为简化起见，假设直接输入输出矩阵 $\mathbf{D} = \mathbf{0}$ ，这一假设适用于驱动控制领域的大部分情形。还要特别注意的是，此处给出的线性二次控制只可将状态变量控制为零。换言之，非零的参考变量是不允许的。对于变化的参考变量来说，则需要最小化控制偏差 $\mathbf{y} - \mathbf{w}$ 或者 $\mathbf{x} - \mathbf{w}$ ，而不再是之前的系统状态 \mathbf{x} 。相对应的处理方式类似于第 291 页章节 9.5 所述情形。

10.2.2 GPC 和 LQR

通过对比广义预测控制和线性二次控制，可以确定如下共同点（见 [28, 40, 62]）：

- 两种方法均使用被控对象的线性模型，而且同样最小化一个标量形式的二次成本函数；
- 二者均以解析方式求解优化问题。换言之，首先对成本函数进行求导，然后通过对其导数置零进行求解；
- 在某些确定条件下，基于 LQR 和 GPC 可得到完全相同的结果；
- 如果基于有限步长的 LQ 控制器和 GPC 控制器最小化的是同一个成本函数，那么得到的是相同的控制变量。从而，二者具有相同的稳定性特点。

Neben diesen Gemeinsamkeiten gibt es aber auch deutliche Unterschiede [28, 34, 40, 62]:

- Während GPC ein übertragungsfunktionsbasiertes Modell besitzt (s. Kap. 6.1.1 auf Seite 75), beruhen die Berechnungen von LQR auf einem Zustandsraummodell.
- Wie alle auf LRPC basierenden Strategien verwendet GPC die aktuelle Zeit t als Bezugspunkt und berechnet von hier aus den optimalen Stellgrößenvektor $\mathbf{u}(t)$ auf der Basis von Vorausberechnungen der Regelgrößen $\mathbf{y}(t+j)$ über einen finiten Horizont in die Zukunft. Beim zustandsraumbasierten LQR hingegen wird der zeitliche Bezugspunkt am Ende des Horizonts gewählt, also $t+N$, und dann von dort aus die Stellgröße $\mathbf{u}(t)$ mittels Rückwärts-Rekursionen bestimmt. Der Horizont N kann dabei im Gegensatz zu GPC sowohl endlich als auch unendlich sein.
- Bei der LQ Regelung wird die optimale Rückführstrategie im Entwurfsprozess, also offline, berechnet. Bei GPC wird die optimale Steuergrößenfolge bei jedem Abtastschritt online neu bestimmt.
- GPC ermittelt eine optimale Steuergrößenfolge mit endlichem Horizont, wobei der Kostenhorizont N_u durchaus geringer sein kann als der Prädiktionshorizont N_p . LQR hingegen bestimmt eine optimale Rückführungsmatrix mit meist unendlichem Zeithorizont. Sollte ein endlicher Horizont gewählt werden, so sind Kosten- und Prädiktionshorizont stets gleich.
- MPC-Strategien ohne Berücksichtigung von Beschränkungen – wie es bei GPC-Reglern oft der Fall ist – ergeben ein zeitinvariantes Regelgesetz. LQR hingegen führt im Allgemeinen zu einer nicht-konstanten Reglermatrix $\mathbf{K}(t)$. Ein zeitlich invarianter Regler, wie für die meisten Anwendungsfälle erwünscht, lässt sich auch mit LQR erzielen, sofern die Bewertungsmatrizen \mathbf{S} , \mathbf{Q} und \mathbf{R} zeitlich konstant sind und ein unendlicher Optimierungshorizont gewählt wird.

Besides these similarities, there are also significant differences [28, 34, 40, 62]:

- Whereas for GPC a transfer function based model is used (see Chapter 6.1.1 on page 75), the calculations for LQR are based on a state space model.
- Like all strategies based on LRPC, GPC uses the current time t as reference point and from this point, the optimum vector for the actuating variables $\mathbf{u}(t)$ is calculated based on the advance calculation of the control variables $\mathbf{y}(t+j)$ over a finite horizon into the future. However, for LQR, which is based on a state space representation, the selected reference time point is at the end of the horizon, i. e. at $t+N$, and from that point the values of the actuating variable $\mathbf{u}(t)$ are determined via backward recursions. In this case, the horizon N can, in contrast to GPC, be finite as well as infinite.

- For LQ control, the optimum feedback strategy is calculated in the design process, i. e. offline. For GPC, the optimum sequence of values for the actuating variables is newly determined for every sampling cycle.
- GPC determines an optimum sequence of values for the actuating variables with a finite horizon, in which the cost horizon N_u can in fact be smaller than the prediction horizon N_p . In contrast to that, LQR determines an optimum feedback matrix with a mostly infinite time horizon. If a finite horizon is chosen, cost and prediction horizon are always equal.
- MPC strategies without consideration of constraints—as it is often the case for GPC controllers—result in a time-invariant control law. In contrast to that, LQR commonly leads to a non-constant controller matrix $\mathbf{K}(t)$. A time invariant controller, which is desired for most applications, can also be determined with LQR, as long as the weighting matrices \mathbf{S} , \mathbf{Q} and \mathbf{R} are chosen to be constant over the time and an infinite optimization horizon is selected.

除了上述共同点之外，二者还存在如下明显的差别 [28, 34, 40, 62]:

- GPC 使用一个基于传递函数的模型，如第 75 页章节 6.1.1 所示；LQR 则基于一个状态空间模型；
- 正如所有基于多步长预测控制的策略，GPC 使用当前时刻 t 作为参考点，并根据所预测的有限步长范围内的被控变量 $\mathbf{y}(t+j)$ 来确定最优控制变量 $\mathbf{u}(t)$ ；相反，基于状态空间模型的 LQR 则选取步长的终点作为参考点，也即 $t+N$ 时刻，并使用前向递推的方式来确定控制变量 $\mathbf{u}(t)$ 。不同于 GPC，LQR 的步长既可以是无限大的，也可以是有限的；
- 对于 LQR 来说，最优的反馈策略是在设计阶段计算出来的，也就是说以离线方式；而 GPC 的最优控制变量序列则是在每一个采样周期内在线实时确定的；
- GPC 在有限步长内得到一个最优的控制变量序列，其中的控制步长一定小于预测步长；而 LQR 则大多是基于无限的时间步长来确定一个最优的反馈矩阵。假使 LQR 使用的是有限步长，那么其控制步长和预测步长也始终相同；
- 不考虑约束的 MPC 策略，正如 GPC 控制器的情形，得到的是一个时不变的控制律；而 LQR 得到的通常是一个非恒定的控制器矩阵 $\mathbf{K}(t)$ 。大部分应用场合希望得到一个时不变的控制律，只有当权重矩阵 \mathbf{S} 、 \mathbf{Q} 和 \mathbf{R} 恒定时，且选择一个无限大的步长，LQR 才能得到一个时不变的控制律。

- LQR löst das Optimierungsproblem über eine Riccati-Gleichung. Dies setzt voraus, dass die Steuergröße weder in der Steilheit noch in der Amplitude irgendwelchen Beschränkungen unterworfen ist. Diese Annahme wird in der Realität niemals zutreffend sein. GPC hingegen erlaubt es, eine Optimierung unter Berücksichtigung solcher Beschränkungen durchzuführen.

Wie die Unterschiede zeigen, beruht GPC auf anderen Grundideen als LQR, auch wenn einige Ähnlichkeiten nicht zu übersehen sind. Auch wenn LQR mit finitem Zeithorizont und GPC die gleichen Ergebnisse liefern, so ist GPC doch einem LQR-Verfahren vorzuziehen. Zum einen erlaubt es LQR vom Prinzip her nicht, eventuelle Systembeschränkungen im Reglerentwurf zu berücksichtigen; zum anderen ist ein GPC-Verfahren auch vom praktischen Aspekt her gesehen der LQR-Strategie überlegen. LQ Regelungen berechnen die zukünftigen Steuergrößen nämlich direkt aus dem Zustandsvektor. Sind nun nicht alle Zustandsgrößen direkt messbar, so benötigt man einen Zustandsbeobachter. Dadurch werden zusätzliche Matrixoperationen online notwendig, wodurch LQR mathematisch sehr aufwändig wird.

- LQR solves the optimization problem via a Riccati equation. This requires the actuating variable being neither constrained in the slope nor in the amplitude. This assumption will in reality never be fulfilled. In contrast to that, GPC allows an optimization under consideration of such constraints.

As the differences show, GPC is based on different basic ideas than LQR, although some similarities can clearly be seen. Although LQR with finite time horizon and GPC deliver same results, GPC should be preferred. On the one hand, LQR does in its approach not allow to consider possible system constraints; on the other hand, a GPC method is also superior to an LQR strategy considering the practical aspect. LQ controls calculate future values for the actuating variables directly from the state vector. If not all state variables are directly measurable, a state observer is needed. Thus, additional online matrix operations are necessary, leading to the fact that LQR will mathematically be very complex.

- LQR 基于一个黎卡提方程求解优化问题，其前提条件是控制变量的变化率以及幅值均不受任何形式的约束。很明显，上述前提条件在实际中很难得到满足。不同的是，GPC 可以在满足上述约束条件下实现优化控制。

正如上述差别所示，GPC 和 LQR 所采用的是两种完全不同的基本思想，即使二者存在某种程度上的相似性。尽管基于有限步长的 LQR 可以得到与 GPC 完全相同的结果，但 GPC 还是要优于 LQR：一方面，从原理上看，LQR 在控制器设计时无法考虑系统约束；另一方面，从实际应用角度看，GPC 比 LQR 考虑得更加全面。LQR 依据状态向量计算未来的控制变量，如果所需的状态向量无法通过直接测量的途径获取，那么则需要借助于额外的状态观测器。为此，额外的在线矩阵计算不可避免，从而导致 LQR 变得更为复杂。

11 Zusammenfassung und Ausblick/ Summary and Outlook/总结和展望

Mit der vorliegenden Arbeit konnte die prinzipielle Anwendbarkeit der modellbasierten Prädiktivregelung für die Antriebstechnik mit den heute zur Verfügung stehenden technischen Mitteln nachgewiesen werden. Es wurden Rechenzeiten erzielt, die selbst den hohen Geschwindigkeitsanforderungen für einen Stromregler genügen; die Laufzeiten liegen durchweg unter $10\ \mu\text{s}$. Allerdings sind hierfür Randbedingungen zu beachten, welche die freie Nutzbarkeit von MPC-Reglern vorläufig noch einschränken.

Das GPC-Prinzip, welches auf einer analytischen Lösung des Optimierungsproblems fußt, ist für lineare Systeme problemlos anwendbar. Selbst ein Drehzahlregler mit einem Prädiktionshorizont von $N_p = 200$ weist eine Laufzeit von weniger als $9\ \mu\text{s}$ auf. Als Vorteil des GPC-Verfahrens gilt, dass wegen der übertragungsfunktionsbasierten Modellstruktur eine verzugsfreie Messwertfilterung möglich ist. Darüberhinaus werden konstante Störgrößen vom Regler berücksichtigt, da vergangene Steuer- und Regelgrößen ebenfalls in die Optimierung eingehen.

Bei einem reinen Stromregler (s. Kap. 6.3.3 auf Seite 119 ff.) machen sich die Vorteile des GPC-Reglers nur im Kleinsignalverhalten bemerkbar. Gleichwertig gute Ergebnisse können auch mit konventionellen Mitteln erzielt werden, z. B. indem eine PI-Regelung mit einer Vorsteuerung ausgestattet wird. Für einen Stromregler lohnt sich der mit GPC verbundene Mehraufwand folglich nicht unbedingt. Gleiches gilt für den Einsatz eines Mehrgrößen-Stromreglers für i_{sd} und i_{sq} im Rahmen einer feldorientierten Regelung (s. Kap. 8.4 auf Seite 225 ff.). Zwar zeigt die Berücksichtigung der Kreuzkopplung zwischen dem feldbildenden und dem drehmomentbildenden Anteil des Statorstroms i_s eine Verbesserung des Regelverhaltens; ähnliche Ergebnisse sind jedoch ebenfalls – mit weniger Aufwand – durch entsprechende Erweiterung einer PI-Regelung erreichbar.

In the presented work, the fundamental applicability of model-based predictive control in drive technology with the technical means being available today could be proven. Computation times that do even satisfy the high performance needs of a current controller were achieved; the computation times are consistently lower than $10\ \mu\text{s}$. However, some constraints, which do still limit the applicability of MPC, have to be considered.

The GPC principle, which is based on an analytical solution of the optimization problem, is applicable to linear systems without any problems. Even a speed controller with

a prediction horizon of $N_p = 200$ needs a computation time of less than $9 \mu\text{s}$. An advantage of the GPC method is that because of the transfer function based model structure, filtering of the measured values is possible without delays. Furthermore, constant disturbances are considered by the controller, since former values of the actuating and of the control variables are also used for the optimization.

For a mere current controller (see Chapter 6.3.3 on page 119 et seq.) the advantages of a GPC controller are only noticeable considering small-signal behavior. Equal results can also be achieved by using conventional techniques, i. e. with a PI controller together with a feed-forward control. Thus the additional effort for a GPC current controller is not really rewarding. The same statement can be given for a MIMO current controller for i_{sd} and i_{sq} used in field oriented control (see Chapter 8.4 on page 225 et seq.). Indeed, the consideration of the cross coupling between the flux and torque producing components of the stator current \mathbf{i}_s shows an improvement of the controller dynamics; however, similar results can also be achieved—with less effort—with an adequate extension of the PI control.

基于前述章节的研究结果，我们论证了模型预测控制在驱动控制技术领域实际应用的可行性。即使对于快速性具有高要求的电流控制器来说，计算时间也能够得到满足，因为可以保证计算时间小于 $10 \mu\text{s}$ 。但若考虑约束条件，MPC 控制器的可用性则受到诸多限制。

基于优化问题解析解的广义预测控制非常适用于线性系统，即使是预测步长 $N_p = 200$ 的速度控制器，所需计算时间也小于 $9 \mu\text{s}$ 。GPC 方法的一大优势是有可能实现测量值滤波的无延迟特性，因为这一方法采用的是基于传递函数形式的模型结构。此外，控制器也能处理恒定的扰动因素，因为过去时刻的控制变量和被控变量全部被用于优化过程。

对于单独的电流控制器来说（见第 6.3.3 章第 119 页），GPC 控制器的优势仅能够在小信号响应特性时得到充分体现。当然，借助于传统的方法也能够得到大致相同的结果，例如可使用 PI 控制器结合前馈控制。由此可知，基于 GPC 控制器实现电流控制并非必要。至于磁场定向控制的情形，基于 GPC 实现 i_{sd} 和 i_{sq} 的多变量电流控制其实也无必要（见第 8.4 章第 225 页），即便它能够考虑到定子电流 \mathbf{i}_s 的励磁分量与转矩分量之间的耦合因素，从而可在一定程度上改善控制效果。类似的结果同样可以通过对 PI 控制器进行适当的调整而实现。

Eine GPC-Drehzahlregelung (s. Kap. 6.3.3 auf Seite 127 ff.) hingegen zeigt deutliche Vorteile gegenüber einer linearen Regelung, da die aus einem Inkrementalgebersignal hergeleitete Drehzahlinformation bei herkömmlichen Regelungen relativ stark gefiltert werden muss. Hier macht sich die Möglichkeit, ein verzugsfreies Messwertfilter gemäß Kap. 6.2 auf Seite 93 ff. in eine CARIMA-Modellstruktur zu integrieren, positiv bemerkbar. Für den Einsatz in einer Drehzahlregelung ist eine GPC-Regelung daher vorteilhaft.

Nachteil eines auf GPC basierenden Verfahrens ist jedoch, dass sich wegen des analytischen Lösungswegs Beschränkungen nur relativ schwierig implementieren lassen. Bei einer MIMO-Regelung für Strom und Drehzahl entsteht die Notwendigkeit, dass der Maschinenstrom begrenzt werden muss. Da in einem solchen Fall jedoch der Stromwert außerhalb des Reglers nicht bekannt ist, muss die Begrenzung innerhalb der Reglerstruktur vorgenommen werden, was problematisch ist. Aus diesem Grund wurde von der praktischen Realisation einer solchen Regelung Abstand genommen. Statt dessen wurde der Gedanke einer direkten Maschinenregelung verfolgt.

Eine direkte modellbasierte Prädiktivregelung besitzt viele Vorteile, die mit keinem anderen Regelverfahren erzielt werden können. Kein anderes Verfahren bietet die Möglichkeit, eine echte Bewertung des Schaltaufwands mit so einfachen Mitteln vorzunehmen. Dazu können verbotene Schaltzustände a priori von der Optimierung ausgeschlossen werden (s. Kap. 9.2.1 auf Seite 243 f.). Da bei einem GPC-Verfahren zukünftige Sollwertverläufe dem Regler innerhalb des Prädiktionshorizonts mitgeteilt werden können, kann die Regelung in stationären Koordinaten erfolgen, ohne dass ein Schleppfehler entsteht (s. Kap. 9.4.3 auf Seite 269 ff.). Leider ist ein GPC-basiertes Direktregelverfahren nicht praktisch anwendbar, da die für die Auswertung des Optimierungsproblems notwendige Rechenzeit auch unter Anwendung intelligenter Suchstrategien nicht auf einen akzeptablen Wert reduziert werden kann (s. Kap. 9.4.4 auf Seite 277 f.).

However, GPC speed control (see Chapter 6.3.3 on page 127 et seq.) shows significant advantages compared to linear control, since the speed information derived from an incremental encoder signal has to be filtered quite intensively when using conventional control methods. Here, the possibility to integrate a delay-free filtering into a CARIMA model structure according to Chapter 6.2 on page 93 et seq. becomes positively noticeable. Thus, GPC for use in a speed control application is profitable.

However, a drawback of a GPC based method is that constraints can hardly be implemented because of the analytical solution. Considering a MIMO control for current and speed, it is necessary to limit the machine current. However, as in such a case the current value is not known outside the controller, the limitation has to be done within the controller structure leading to some problems. For this reason, a practical realization of such a control scheme was given up. Instead, the idea of a direct machine control was pursued.

A direct model-based predictive control has many advantages that cannot be achieved with any other control method. No other method offers the possibility to evaluate the switching effort in such a simple way. Furthermore, switching states that are not allowed, can a priori be excluded from the optimization (see Chapter 9.2.1 on page 243 et seq.). Since, considering a GPC method, future values of the actuating variables within the prediction horizon can be forwarded to the controller, control can be done in a stationary coordinate frame without a contouring error (see Chapter 9.4.3 on page 269 et seq.). Unfortunately, a GPC based direct control method can practically not be realized, as the computation time necessary for the evaluation of the optimization problem cannot be reduced to an acceptable value, even if intelligent search strategies are applied (see Chapter 9.4.4 on page 277 et seq.).

然而，相对于线性控制来说，基于 GPC 的转速控制（见章节 6.3.3 第 127 页）则能够明显体现出优势，其原因在于传统控制方式下由编码器信号所生成的速度信号需要经过相对较强的滤波处理，而 GPC 则可以将一个无延迟的测量值滤波器整合到 CARIMA 模型结构。所以，基于 GPC 控制方式来实现速度控制是具有一定优势的。

但是，基于 GPC 方法的一大缺点是：由于使用解析求解的方式而使得实现系统约束变得困难。对于实现电流和速度控制的 MIMO 控制来说，有必要对电机电流进行限制。然而，在这种情况下，位于控制器结构之外的电流参考值是未知的，而限幅又必须在控制器结构内部实现，从而出现了困难。正因如此，在实际中通常放弃使用上述方案，取而代之的是一种直接电机控制的想法。

直接模型预测控制具有其他控制方法无法实现的诸多优势。没有其他任何一种方法能够以如此简单的方式实现对开关变化情况的惩罚。此外，不允许使用的开关状态也可以直接排除在外（见章节 9.2.1 第 243 页）。由于 GPC 方法中未来参考值的变化情况可以在预测步长范围内告知控制器的，从而控制可以在固定坐标系下得到实现，并且不会出现跟踪误差（见章节 9.4.3 第 269 页）。遗憾的是，基于 GPC 的直接控制方法很难实际应用，因为用于处理优化问题所需的计算时间无法降低到可接受的时间范围之内，即使是采用了一些智能的搜索策略（见章节 9.4.4 第 277 页）。

Als Ausweg aus diesem Dilemma bietet sich die explizite Lösung des Optimierungsproblems an. Leider ist dies bisher nur für MPC-Regler hergeleitet, die auf dem Zustandsraummodell basieren; eine explizite GPC-Regelung ist damit vorläufig nicht mit den aktuell zur Verfügung stehenden Mitteln realisierbar. Die GPC-spezifischen Vorteile wie die interne Filterung und die Berücksichtigung der Vergangenheit sind folglich nicht nutzbar. Allerdings kann auch die explizite Lösung normalerweise nur unter Zuhilfenahme weiterer Werkzeuge praktisch angewendet werden, da die Rechenzeiten für eine online-Suche nach dem aktiven Regelgesetz schon für kleine Reglerstrukturen schnell den für eine Antriebsregelung zulässigen Wert überschreiten. Vereinfachungen und Umformulierungen der expliziten Reglerstruktur, wie sie in den Kapiteln 9.5.3 und 9.5.4 erläutert werden, ermöglichen jedoch die Ausführung selbst von komplexen Reglern in äußerst kurzen Zeiten.

Für eine Stromregelung mit linearem Modell, d. h. mit Modulator, ist die explizite MPC-Lösung einem GPC-Regler unterlegen, da der Rechenaufwand ungefähr identisch ist, GPC jedoch aus den oben aufgeführten Gründen weitere Vorteile bietet. Ein expliziter MPC-Regler mit linearem Modell ist daher für die elektrische Antriebstechnik nicht empfehlenswert.

Eine Direktregelung ohne Modulator ist mit einem GPC-basierten Verfahren aus Gründen der Rechenzeit nicht möglich. Eine Überführung des Optimierungsproblems in eine Polytopstruktur mit affinen Regelgesetzen erlaubt jedoch die Ausführung einer MPC-Regelung eines elektrischen Antriebs unter Berücksichtigung der wertdiskreten Eigenschaften des Stellglieds. Eine Anwendung als Stromregler zeigt die Überlegenheit von DMPC über einen linearen Regler im dynamischen Betrieb (s. Kap. 9.5.5 auf Seite 327 ff.).

A way out of this dilemma is the explicit solution of the optimization problem. Unfortunately, this can, until now, only be done for MPC controllers based on state space representation; thus, an explicit GPC control with the methods currently available can at least presently not be realized. Hence, the GPC specific advantages like internal filtering and consideration of past values cannot be used. However, even the explicit solution can only be practically applied when additional tools are used, since the computation times for an online search for the active control law even for small controller structures quickly exceed the acceptable value for drive control. Simplifications and transformations of the explicit controller structure, as they are described in the Chapters 9.5.3 and 9.5.4, allow indeed the execution of even complex controllers in very short times.

For current control with a linear model, i. e. with modulator, the explicit MPC solution is inferior to a GPC controller, since the computational effort is nearly the same. GPC, however, offers further advantages due to the reasons stated above. Thus, an explicit MPC controller with linear model cannot be recommended for electric drive technology.

A direct control with a GPC-based method and without a modulator is not feasible because of the needed computation time. However, a transformation of the optimization problem into a polytope structure with affine control laws allows the execution of MPC control of an electric drive under consideration of the value discrete nature of the actuator. An application as current controller shows the superiority of DMPC compared to a linear controller in dynamic mode (see Chapter 9.5.5 at page 327 et seqq.).

优化问题的显式求解作为一种可摆脱上述困境的方法。遗憾的是，到目前为止，这一方法也仅应用于基于状态空间模型的 MPC 控制器；目前可用的方法仍然无法实现显式 GPC 控制。而且 GPC 方法独特的优势将无法继续使用，如内部滤波、考虑过去的状态等。通常来说，显式求解方式仅可作为辅助性手段在实际中应用，因为即使对于很小的控制器结构来说，在线搜索有效的控制律所需计算时间已经超过了驱动控制领域所允许的数值。如在章节 9.5.3 和章节 9.5.4 所述，通过对显式控制器结构进行简化和变形，使得在极短时间内实现复杂控制器成为可能。

对于基于线性模型的电流控制来说，如含有调制器的情形，显式 MPC 方法确实不如 GPC 方法，因为二者计算时间几乎一致，但 GPC 还包含前面所介绍的其他优势。所以，对于电力驱动技术领域的应用来说，并不推荐使用基于线性模型的显式 MPC 方法。

由于计算时间的限制，使用 GPC 方法无法实现不包含调制器的直接控制。通过将优化问题转换为一个含有仿射控制律的多胞体结构，使得在考虑了执行机构数值离散特性的情况下实现电力驱动的 MPC 控制成为可能。一个实现电流控制的实际应用表明，在动态情况下，DMPC 要优于线性控制器（见章节 9.5.5 第 327 页）。

Allerdings existieren einige Nachteile:

- Übertragungsfunktionsbasierte Streckenmodelle ähneln den IIR-Filtern der Nachrichtentechnik; daher ist eine Filterung von Messsignalen problemlos implementierbar. Zustandsraummodelle bieten keine Möglichkeit, eine verzugsfreie Filterung ohne zusätzlichen Aufwand vorzunehmen. Sollte dies notwendig sein, so muss z. B. ein Kalman-Filter vorgeschaltet werden. Für die Stromregelung eines elektrischen Antriebs mittels DMPC ist dies jedoch nicht notwendig, da durch die Berücksichtigung der hybriden Eigenschaften die Stromverzerrungen quasi zum System dazugehören und daher nicht zu unerwünschtem Verhalten führen, so dass sich der Nachteil der fehlenden verzugsfreien Filterung in diesem Fall nicht auswirkt. Soll die Drehzahlregelung in einen DMPC-Regler integriert werden, sind hier jedoch Probleme zu erwarten.
- Für die experimentellen Untersuchungen wurde ein äußerst einfaches Maschinenmodell verwendet. Selbst mit diesem Modell ergaben sich Reglerstrukturen mit bis zu 10.000 Polytopen. Aufwändigere Streckenmodelle, die z. B. die Kreuzkopplung berücksichtigen oder eine MIMO-Regelung von Strom und Drehzahl erlauben, führen schnell zu noch komplexeren Strukturen, die dann mit der aktuell verfügbaren Hard- und Software nicht mehr beherrschbar sind. Ähnliches gilt für größere Regelhorizonte.
- Bei expliziten MPC-Verfahren ist eine Berücksichtigung der zukünftigen Sollwert-Trajektorien nicht möglich, da sich in diesem Fall für jeden zukünftigen Abtastschritt die Dimension der Reglerstruktur um Eins erhöhen würde, was wiederum zu nicht mehr beherrschbaren Strukturen führt. Da die Stromregelung in Statorkoordinaten erfolgt, sind Schleppfehler daher unvermeidlich.
- Mit dem Ersatz der impliziten Einschnitt-Optimierung durch eine explizite Lösung entfällt der Vorteil, dass dynamische Beschränkungen der Stellgröße, wie sie z. B. durch eine Bootstrap-Treiberschaltung des Umrichters hervorgerufen werden, berücksichtigt werden können. Es bleibt lediglich die von herkömmlichen Direktsteuerverfahren bekannte Vorgehensweise, den von der Regelung ermittelten Schaltzustand bei Bedarf nachträglich zu verändern.
- Auf Zustandsraummodellen basierende MPC-Regler können vergangene Eingangs- und Ausgangsgrößen der Regelstrecke nicht berücksichtigen und scheitern daher an der Ausregelung konstanter Störgrößen. Für Stromregelungen von elektrischen Antrieben stellt jedoch die Gegenspannung der Maschine stets eine solche Störung dar. Um ein offsetfreies Ausregeln zu ermöglichen, ist zusätzlicher Aufwand notwendig.

However, some drawbacks do exist:

- System models based on the transfer function are similar to the IIR filters known from communications technology. Hence, a filtering of measured signals can be easily implemented. State space models, on the other hand, offer no possibility to perform a delay-free filtering without any additional effort. If this is needed, a Kalman filter or similar tools have to be included in the signal path. For realizing a current control of an electric drive via DMPC, this is indeed not necessary, since the current distortions are quasi included in the model by regarding the hybrid characteristics of the system. Hence, the disadvantage of the missing delay-free filtering has no impact on the controller performance in this case. However, if a speed control is to be included in the DMPC controller, problems are to be expected.
- For the experimental investigations, a very simple machine model was used, but even this model resulted in controller structures with up to 10,000 polytopes. More complex plant models, which e. g. consider the cross coupling or which allow MIMO control of current and speed, will quickly lead to even more complex structures which are not feasible anymore with the hard- and software currently available. Similar statements can be made for larger control horizons.
- In explicit MPC methods, a consideration of future reference value trajectories is not possible, since in this case for every future sampling cycle the dimension of the controller structure would be enlarged by one, which would again lead to structures that are not feasible anymore. Since current control is done in stator coordinates contouring errors cannot be avoided.
- With the use of the implicit single-step optimization method by means of an explicit solution, the advantage that dynamic constraints for the values of the actuating variables can be considered, as they can e. g. be caused by a bootstrap driver circuit of the inverter, does not apply. Only the approach known from conventional direct control methods remains, i. e. to change the switching state determined by the control afterwards if it is necessary.
- MPC controllers based on state space models cannot consider past input and output variables of the plant and thus they fail in controlling constant disturbances to zero. Unfortunately, for current control of electrical drives the back emf of the machine is such a disturbance. In order to achieve offset free control additional effort is necessary.

Notizen/Notes/笔记

然而，它也确实存在一些缺点：

- 基于传递函数的被控系统模型与通信技术中已知的 IIR 滤波器相似。因此，测量信号的滤波可以很容易得到实现。另一方面，状态空间模型不可能在不做任何额外努力的情况下进行无延迟滤波。如果必须的话，则必须接入类似卡尔曼滤波器的装置。对于基于 DMPC 实现电力驱动的电流控制来说，上述措施确实没有必要，因为考虑到系统的混合特性，电流畸变已包含在模型中。因此，在这种情况下，缺少无延迟滤波特性这一缺点对控制器性能并没有影响。然而，如果将速度控制也包括在 DMPC 控制器中，那么则有可能出现问题。
- 在实际实验考察时，我们使用了一个非常简单的电机模型。但是，即使如此简单的模型也使得控制器结构包含了将近 10,000 个多胞体。至于更复杂的被控系统模型，例如考虑交叉耦合因素或同时实现电流和速度的 MIMO 控制，则会迅速导致更为复杂的结构，这在目前可用的硬件和软件框架下是无法实现的。对于更大的控制步长情形，会出现类似的结果。
- 对于显式 MPC 方法来说，它无法考虑到未来的参考值轨迹。因为在这一情况下，对于每一个未来的采样周期，控制器的维度将会增加一个数量级，这将导致一个无法实现的结构。由于在定子坐标系下实现电流控制，那么跟踪误差则无法避免。
- 通过使用显式求解来代替隐式的单步优化将会失去原本可以考虑到控制变量的动态约束这一优势，有时这一约束可能是由于变流器的自举电路所引起。目前仅存的也只有传统的直接控制方法所用的处理方式，也即在实际控制需要的情况下延迟地改变开关状态。
- 基于状态空间模型的 MPC 控制无法考虑被控对象过去的输入和输出变量，因此也无法有效地将扰动变量控制为零。不幸的是，对于电力驱动的电流控制来说，电机的反电动势又不可避免地作为一种扰动变量。所以，为了实现无偏移的控制，有必要做出额外的努力。

Zusammenfassend kann gesagt werden, dass sich modellbasierte Prädiktivregler in der Antriebstechnik anwenden lassen. Für den Einsatz in schnell schaltenden Umrichtern scheint jedoch die Verwendung von MPC aus den oben genannten Gründen nicht sinnvoll zu sein. Die direkte Regelung von elektrischen Antrieben ohne zwischengeschalteten Modulator mittels expliziter MPC erscheint hier vielversprechender. Der momentane Forschungsstand erlaubt es jedoch noch nicht, die prinzipiellen Vorteile einer direkten LRPC hier nutzbar zu machen. Die verfügbare Rechenleistung wird allerdings erfahrungsgemäß weiter steigen. Außerdem sind die weltweiten Forschungsarbeiten auf dem Gebiet der hybriden Systeme und der expliziten MPC-Regelung bei weitem noch nicht abgeschlossen. Deshalb sind weitere Verbesserungen der offline-Berechnungen, die zu schlankeren Reglerstrukturen führen werden, zu erwarten. Beides zusammen wird DMPC mittelfristig auch für aufwändigere Regelungen anwendbar machen.

Folgende Ziele scheinen damit erreichbar: 1). In dem Maß, in dem komplexere Strukturen beherrschbar werden, kann das Maschinenmodell verbessert werden. Eine MIMO-Regelung, die auch die Drehzahl- oder gar die Lageregelung einschließt, wird möglich; 2). Da bei MPC die Kostenfunktion frei gewählt werden kann, bietet es sich an, neben dem Stromfehler und dem Schaltaufwand auch den Verzerrungsfaktor d zu bewerten. Mit diesen Vorgaben erhält man *online optimierte Pulsmuster*. Der bisher übliche Weg über offline optimierte Pulsmuster für verschiedene stationäre Betriebspunkte und das mühsamen Ausregeln des Modulationsfehlers im dynamischen Betrieb [15] wären dann nicht mehr notwendig.

Als letzter Punkt sei noch auf die nichtlinearen Prädiktivregler hingewiesen. Bei Verwendung eines DMPC-Reglers werden die durch die digitale Regelung entstehende Totzeit und die wertdiskrete Eigenschaft der Stellgröße (Quantisierung) implizit berücksichtigt. Ein elektrischer Antrieb besitzt allerdings darüberhinaus weitere Nichtlinearitäten, wie z. B. die gegenseitige Beeinflussung der beiden Statorstromkomponenten oder Sättigungseffekte in der Maschine, die in dem hier verwendeten einfachen Maschinenmodell nicht von vorneherein berücksichtigt werden. Durch die Vorgehensweise, die Regelstrecke als stückweise affines System zu beschreiben, könnten diese Nichtlinearitäten jedoch recht gut angenähert werden. Unter der Voraussetzung, dass sich in naher Zukunft auch für komplexe Systeme eine beherrschbare Reglerstruktur bestimmen lässt, sollte die DMPC-Regelung auf diese Weise weiter verbessert werden können.

In summary, it can be stated that model-based predictive controllers can be applied in drive technology. However, the application of MPC in fast switching inverters does not seem to be expedient for the reasons stated above. Direct control of electrical drives without intermediary modulator with explicit MPC seems to be much more promising here. However, the current state of research does not allow to make use of the fundamental advantages of direct LRPC. Indeed, the available processing power will surely increase. Furthermore, the worldwide research activities in the area of hybrid systems

and explicit MPC control are far from being concluded. Thus further improvements of the offline calculations which lead to smaller controller structures can be expected. Both together will in some time make DMPC feasible for more complex calculations, too.

Thus the following intentions seem to be reachable: 1). The machine model can be improved in the same degree as in which more complex structures will become feasible. A MIMO control including speed or even position control will then be realizable; 2). As for MPC, the cost function can be chosen freely, it is also possible to weight the distortion factor d besides the current error and the switching effort. With these demands *online optimized pulse patterns* can be obtained. The conventional method via offline optimized pulse patterns for different stationary operating points and the exhausting compensation of the modulation error in dynamic mode [15] would then not be necessary anymore.

Finally, nonlinear predictive controllers shall be pointed out. If a DMPC controller is used, the dead time and the value-discrete character of the actuating variable (quantization) caused by the digital control are implicitly considered. Moreover, an electric drive has indeed further nonlinearities, like e. g. the cross coupling of both stator current components or saturation effects in the machine, which are not considered in the simple machine model used here. However, by taking the approach to describe the plant as a piecewise affine system, these nonlinearities could be approximated. Assuming the fact that in the near future also for complex systems a feasible controller structure can be determined, it should be possible that further improvements of DMPC control can be made.

综上所述,可以说基于模型的预测控制器可以应用于驱动技术领域。然而,由于前述原因,MPC在快速开关变化的变流器场合下的应用似乎并不合适。借助于显式MPC实现不包含中间调制器的电力驱动直接控制似乎更有前景。然而,目前的研究现状还无法利用直接LRPC的基本优势。根据经验,可用的处理器计算能力将不断增加。此外,世界范围内的学者在混合系统和显式MPC控制领域的研究活动还远远没有结束。因此,可以预期的是,通过对离线计算的进一步改进有可能导致更小的控制器结构。两者结合,在不久的将来,DMPC在更复杂的控制场合中也将是完全可行的方案。

因此,以下两个目标看似是可以实现的:1).在某种程度上改进电机模型,以使其可用于更复杂的结构。此外,包括速度甚至是位置控制的MIMO控制将是可实现的;2).对于MPC来说,目标函数可以自由选择,除了电流误差和开关努力之外,还可以对失真系数 d 进行加权。有了这些要求,就可以得到在线优化脉冲模式。传统的方法是通过离线优化脉冲模式,用于不同的静止工作点,并在动态模式下对调制误差进行详尽的补偿[15],如此则再无必要。

最后要提及的是非线性预测控制器。当使用DMPC控制器时,数字控制引起的死区时间和执行变量的数值离散性(量化)被隐含地考虑在内。此外,电力驱动还包含额外的非线性因素,如定子电流分量的相互耦合或电机的饱和效应,在前面介绍的简化电机模型并没有得到考虑。然而,通过将对象描述为一个分段仿射系统,那么这些非线性因素可以被很好地近似处理。假设在不久的将来也能为复杂的系统确定一个可行的控制器结构,那么就有可能进一步改进DMPC控制。

Literaturverzeichnis/Bibliography/参考文献

- [1] M. Aaltonen, P. Tiitinen, J. Lalu, S. Heikkilä, Direkte Drehmomentregelung von Drehstromantrieben (Direct Torque Control of Three-Phase Drives), *ABB Technik*, No. 3, 1995, pp. 19–24
- [2] A. Ackva, H. Reinold, R. Olesinski, A Simple and Self-Adapting High-Performance Current Control Scheme for Three Phase Voltage Source Inverters, *23rd IEEE Power Electronics Specialists Conference pesc'92*, Vol. 1, pp. 435–442, Toledo, 1992
- [3] P. Albertos, R. Ortega, On Generalized Predictive Control: Two Alternative Formulations, *Automatica*, Vol. 25, No. 5, 1989, pp. 753–755
- [4] V. Ambrožič, R. Fišer, D. Nedeljković, Direct Current Control—A New Current Regulation Principle, *IEEE Transactions on Power Electronics*, Vol. 18, No. 1, 2003, pp. 495–503
- [5] V. Ambrožič, G. S. Buja, R. Menis, Band-Constrained Technique for Direct Torque Control of Induction Motor, *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 4, 2004, pp. 776–784
- [6] G. Amler, A PWM Current-Source Inverter for high Quality Drives, *EPE Journal*, Vol. 1, No. 1, 1991, pp. 21–32
- [7] G. M. Asher, M. Sumner, F. Cupertino, A. Lattanzi, Direct Flux Control of Induction Motor Drives, *European Conference on Power Electronics and Applications EPE 2001*, Graz, 2001
- [8] C. Attaianesi, V. Nardi, A. Perfetto, G. Tomasso, Vectorial Torque Control: A Novel Approach to Torque and Flux Control of Induction Motor Drives, *IEEE Transactions on Industry Applications*, Vol. 35, No. 6, 1999, pp. 1399–1405
- [9] R. Bellman, *Dynamische Programmierung und selbstanpassende Regelprozesse (Dynamic Programming and Self-Adapting Control Processes)*, Munich, Vienna: R. Oldenbourg Verlag, 1967
- [10] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, Vol. 35, No. 3, 1999, pp. 407–427

- [11] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The Explicit Solution of Model Predictive Control via Multiparametric Quadratic Programming, *American Control Conference ACC2000*, pp. 872–876, Chicago, 2000
- [12] A. Bemporad, F. Borrelli, M. Morari, Piecewise Linear Optimal Controllers for Hybrid Systems, *American Control Conference ACC2000*, pp. 1190–1194, Chicago, 2000
- [13] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica*, Vol. 38, No. 1, 2002, pp. 3–20
- [14] R. E. Betz, B. J. Cook, S. J. Henriksen, A Digital Current Controller for Three Phase Voltage Source Inverters, *IEEE IAS Annual Meeting*, Vol. 1, pp. 722–729, New Orleans, 1997
- [15] B. Beyer, *Schnelle Stromregelung für Hochleistungsantriebe mit Vorgabe der Stromtrajektorie durch off-line optimierte Pulsmuster (Fast Current Control for High Power Drives with Presetting of the Current Trajectory via Off-line Optimized Pulse Patterns)*, Dissertation, Wuppertal: Wuppertal University, 1998
- [16] S. Bolognani, M. Zigliotto, Full-digital predictive hysteresis current control for switching losses minimisation in PMSM drives, *Power Electronics, Machines and Drives Conference PEMD'02*, pp. 61–67, Bath, 2002
- [17] F. Bonanno, A. Consoli, A. Raciti, A. Testa, An Innovative Direct Self-Control Scheme for Induction Motor Drives, *IEEE Transactions on Power Electronics*, Vol. 12, No. 5, 1997, pp. 800–806
- [18] C. Bordons, E. F. Camacho, A Generalized Predictive Controller for a Wide Class of Industrial Processes, *IEEE Transactions on Control Systems Technology*, Vol. 6, No. 3, 1998, pp. 372–387
- [19] P. Boucher, D. Dumur, P. Raguenaud, C. Rougebief, Multivariable Generalised Predictive Control for Cognac Distillation, *European Control Conference ECC'99*, Karlsruhe, 1999
- [20] E. F. Camacho, C. Bordons, *Model Predictive Control*, London: Springer-Verlag, 1999
- [21] M. S. Carmeli, F. Castelli-Dezza, G. Superti-Furga, Smart Modulation: A new Approach to Power Converter Control, *European Conference on Power Electronics and Applications EPE 2001*, Graz, 2001
- [22] D. Casadei, G. Serra, A. Tani, Implementation of a Direct Torque Control Algorithm for Induction Motors Based on Discrete Space Vector Modulation, *IEEE Transactions on Power Electronics*, Vol. 15, No. 4, 2000, pp. 769–777

-
- [23] M. Chandorkar, New Techniques for Inverter Flux Control, *IEEE Transactions on Industry Applications*, Vol. 37, No. 3, 2001, pp. 880–887
- [24] Y. A. Chapuis, C. Pelissou, D. Roze, Direct Torque Control of Induction Machine under Square Wave Conditions, *IEEE IAS Annual Meeting*, Vol. 1, pp. 343–349, Orlando, 1995
- [25] S. Chattopadhyay, V. Ramanarayanan, V. Jayashankar, A Predictive Switching Modulator for Current Mode Control of High Power Factor Boost Rectifier, *IEEE Transactions on Power Electronics*, Vol. 18, No. 1, Part I, 2003, pp. 114–123
- [26] J.-W. Choi, S.-K. Sul, New Current Control Concept—Minimum Time Current Control in the Three-Phase PWM Converter, *IEEE Transactions on Power Electronics*, Vol. 12, No. 1, 1997, pp. 124–131
- [27] D. W. Clarke, C. Mohtadi, P. S. Tuffs, Generalized Predictive Control—Part I. The Basic Algorithm, *Automatica*, Vol. 23, No. 2, 1987, pp. 137–148
- [28] D. W. Clarke, C. Mohtadi, P. S. Tuffs, Generalized Predictive Control—Part II. Extensions and Interpretations, *Automatica*, Vol. 23, No. 2, 1987, pp. 149–160
- [29] D. W. Clarke, R. Scattolini, Constrained receding-horizon predictive control, *IEE Proceedings*, Vol. 138, Part D, No. 4, 1991, pp. 347–354
- [30] D. W. Clarke, Adaptive Predictive Control, *Annual Review in Automatic Programming*, Vol. 20, 1996, pp. 83–94
- [31] L. Cooper, M. W. Cooper, *Introduction to Dynamic Programming*, Oxford: Pergamon Press, 1981
- [32] C. R. Cutler, B. L. Ramaker, Dynamic Matrix Control—A computer control algorithm, *Joint Automatic Control Conference 1980*, Vol. 1, pp. 806–811, San Francisco, 1980
- [33] M. Depenbrock, Direkte Selbstregelung (DSR) für hochdynamische Drehfeldantriebe mit Stromrichterspeisung (Direct Self Control for Inverter Supplied High Dynamic AC Drives), *etzArchiv*, Vol. 7, 1985, pp. 211–218
- [34] R. Dittmar, B.-M. Pfeiffer, *Modellbasierte prädiktive Regelung (Model Based Predictive Control)*, Munich, Vienna: R. Oldenbourg Verlag, 2004
- [35] S. V. Emeljanov, *Automatische Regelsysteme mit veränderlicher Struktur (Automatic Control Systems with Variable Structure)*, Munich, Vienna: R. Oldenbourg Verlag, 1969

- [36] J. Faßnacht, P. Mutschler, Direct Mean Torque Control with improved flux control, *European Conference on Power Electronics and Applications EPE 2003*, Toulouse, 2003
- [37] E. Flach *Direkte Regelung des Drehmomentmittelwerts einer Induktionsmaschine (Direct Control of the Mean Torque Value of an Induction Machine)*, Dissertation, Darmstadt: Technial University Darmstadt, 1999
- [38] O. Föllinger, *Regelungstechnik (Control Technology)*, Heidelberg: Hüthig, 1994
- [39] F. R. Gantmacher, *Matrizentheorie (Matrix Theory)*, Berlin, Heidelberg: Springer-Verlag, 1986
- [40] C. E. García, D. M. Prett, M. Morari, Model Predictive Control: Theory and Practice—a Survey, *Automatica*, Vol. 25, No. 3, 1989, pp. 335–348
- [41] H. P. Geering, *Berechnung von Zustandsraummodellen minimaler Ordnung aus der Übertragungsmatrix $G(s)$ (Calculation of State Space Models with Minimal Order from the Transfer Matrix $G(s)$)*, <http://www.imrtweb.ethz.ch/users/geering/SGA1983v2.pdf>, Zurich: IMRT Press, 1999
- [42] T. Geyer, F. D. Torrisi, M. Morari, Efficient Mode Enumeration of Compositional Hybrid Systems, *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, Vol. 2623, April 2003, pp. 216–232
- [43] T. Geyer, F. D. Torrisi, M. Morari, Optimal Complexity Reduction of Piecewise Affine Models based on Hyperplane Arrangements, *American Control Conference ACC2004*, Boston, 2004
- [44] G. C. Goodwin, K. S. Sin, *Adaptive Filtering, Prediction and Control*, Englewood Cliffs: Prentice-Hall, 1984
- [45] P. Grieder, M. Morari, Complexity Reduction of Receding Horizon Control, *IEEE Conference on Decision and Control CDC2003*, pp. 3179–3184, Maui, 2003
- [46] P. Grieder, M. Kvasnica, M. Baotić, M. Morari, Low Complexity Control of Piecewise Affine Systems with Stability Guarantee, *American Control Conference ACC2004*, Boston, 2004
- [47] G. Hadley, *Nichtlineare und dynamische Programmierung (Non-linear and Dynamic Programming)*, Würzburg: Physica-Verlag, 1969
- [48] L. Harnefors, H.-P. Nee, Model-Based Current Control of AC Machines Using the Internal Model Control Method, *IEEE Transactions on Industry Applications*, Vol. 34, No. 1, 1998, pp. 133–141

-
- [49] J. Hecht, *Design and Simulation of a New Predictive Current Control Strategy for DC Drives*, Diploma Thesis, Toronto, Karlsruhe: University of Toronto/Technical University Karlsruhe, 1991
- [50] D. Hintze, *Asynchroner Vierquadranten-Drehstromantrieb mit Stromzwischenkreisumrichter und Oberschwingungsarmen Maschinengrößen (Asynchronous Four-Quadrant AC Drive with Current Source Inverter and Machine Values with Low Harmonics)*, Dissertation, Munich: Technical University Munich, 1993
- [51] U. Hoffmann, *Entwurf und Erprobung einer adaptiven Zweipunktregelung (Design and Proving of an Adaptive Two-Step Control)*, Dissertation, Aachen: Technical University Aachen, 1984
- [52] U. Hoffmann, Eine adaptive Zweipunktregelung für Prozesse mit schaltenden Stellgliedern (An Adaptive Two-Step Control for Processes with Switching Actuators), *at – Automatisierungstechnik*, 35. Jahrgang, No. 5, 1987, pp. 184–191
- [53] D. G. Holmes, D. A. Martin, Implementation of a Direct Digital Predictive Current Controller for Single and Three Phase Voltage Source Inverters, *IEEE IAS Annual Meeting*, Vol. 2, pp. 906–913, San Diego, 1996
- [54] J. Holtz, U. Schwellenberg, A new Fast-Response Current Control Scheme for Line Controlled Converters, *International Semiconductor Power Converter Conference*, pp. 175–183, Orlando, 1982
- [55] J. Holtz, S. Stadtfeld, A Predictive Controller for the Stator Current Vector of AC-Machines fed from a Switched Voltage Source, *International Power Electronics Conference IPEC*, Vol. 2, pp. 1665–1675, Tokio, 1983
- [56] J. Holtz, The Dynamic Representation of AC Drive Systems by Complex Signal Flow Graphs, *International Symposium on Industrial Electronics ISIE'94*, pp. 1–6, Santiago de Chile, 1994
- [57] J. Holtz, The Induction Motor—A Dynamic System, *IEEE IECON'94*, Vol. 1, pp. P1–P6, Bologna, 1994
- [58] R. Isermann, *Digitale Regelsysteme (Digital Control Systems)*, Volume I, Berlin, Heidelberg: Springer-Verlag, 1987
- [59] R. Isermann, *Digitale Regelsysteme (Digital Control Systems)*, Volume II, Berlin, Heidelberg: Springer-Verlag, 1987
- [60] B.-J. Kang, C.-M. Liaw, A Robust Hysteresis Current-Controlled PWM Inverter for Linear PMSM Driven Magnetic Suspended Positioning System, *IEEE Transactions on Industrial Electronics*, Vol. 48, No. 5, 2001, pp. 956–967

- [61] J.-K. Kang, S.-K. Sul, New Direct Torque Control of Induction Motor for Minimum Torque Ripple and Constant Switching Frequency, *IEEE Transactions on Industry Applications*, Vol. 35, No. 5, 1999, pp. 1076–1082
- [62] P. P. Kanjilal, *Adaptive prediction and predictive control*, London: Peter Peregrinus, 1995
- [63] A. Kaufmann, A. Henry-Labordère, *Integer and Mixed Programming: Theory and Applications*, New York, San Francisco, London: Academic Press, 1977
- [64] M. P. Kaźmierkowski, M. A. Dzieniakowski, W. Sulkowski, Novel Space Vector Based Current Controllers for PWM-Inverters, *IEEE Transactions on Power Electronics*, Vol. 6, No. 1, 1991, pp. 158–166
- [65] R. Kennel, *Prädiktives Führungsverfahren für Stromrichter (Predictive Control Strategy for Inverters)*, Dissertation, Kaiserslautern: Kaiserslautern University, 1984
- [66] R. Kennel, A. Linder, M. Linke, Generalized Predictive Control (GPC)—Ready for Use in Drive Applications?, *32nd IEEE Power Electronics Specialists Conference pesc2001*, Vol. 4, pp. 1839–1844, Vancouver, 2001
- [67] R. Kennel, A. Linder, Predictive Control for Electrical Drives—A Survey, *Korea-Germany Advanced Power Electronics Symposium 2001*, pp. 39–43, Seoul, 2001
- [68] R. M. C. de Keyser, G. A. van de Velde, F. A. G. Dumortier, A Comparative Study of Self-adaptive Long-range Predictive Control Methods, *Automatica*, Vol. 24, No. 2, 1988, pp. 149–163
- [69] H. Kohlmeier, O. Niermeyer, D. Schröder, High dynamic four-quadrant AC-motor drive with improved power-factor and on-line optimized pulse pattern with PROMC, *European Conference on Power Electronics and Applications EPE'85*, Vol. 3, pp. 3.173–3.178, Brüssel, 1985
- [70] K. P. Kovács, I. Rácz, *Transiente Vorgänge in Wechselstrommaschinen (Transient Behavior of AC Machines)*, Volume I, Budapest: Akadémiai Kiadó, 1959
- [71] K. P. Kovács, I. Rácz, *Transiente Vorgänge in Wechselstrommaschinen (Transient Behavior of AC Machines)*, Volume II, Budapest: Akadémiai Kiadó, 1959
- [72] P. Krauss, K. Daß, T. Bunte, Prädiktiver Regler mit Kalman-Filter zur Zustandsschätzung (Predictive Controller with Kalman Filter for State Estimation), *at – Automatisierungstechnik*, Vol. 42, No. 12, 1994, pp. 533–539

-
- [73] M. Kvasnica, P. Grieder, M. Baotić, M. Morari, Multi-Parametric Toolbox (MPT), *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, Vol. 2993, March 2004, pp. 448–462
- [74] K.-K. La, M.-H. Shin, D.-S. Hyun, Direct Torque Control of Induction Motor with Reduction of Torque Ripple, *IEEE IECON2000*, pp. 1087–1092, Nagoya, 2000
- [75] P. Lancaster, M. Tismenetsky, *The Theory of Matrices*, Orlando: Academic Press, 1985
- [76] C. Lascu, I. Boldea, F. Blaabjerg, A Modified Direct Torque Control (DTC) for Induction Motor Sensorless Drive, *IEEE IAS Annual Meeting*, Vol. 1, pp. 415–422, St. Louis, 1998
- [77] C. Lascu, A. M. Trzynadlowski, Combining the Principles of Sliding Mode, Direct Torque Control and Space-Vector Modulation in a High-Performance Sensorless AC Drive, *IEEE Transactions on Industry Applications*, Vol. 40, No. 1, 2004, pp. 170–177
- [78] J. B. Lasserre, An explicit equivalent positive semidefinite program for nonlinear 0-1 programs, *SIAM Journal on Optimization*, Vol. 12, No. 3, 2002, pp. 756–769
- [79] W. Leonhard, *Control of Electrical Drives*, Berlin, Heidelberg: Springer-Verlag, 1985
- [80] I. J. Leontaritis, S. A. Billings, Input-output parametric models for non-linear systems Part I: deterministic non-linear systems, *International Journal of Control*, Vol. 41, No. 2, 1985, pp. 303–328
- [81] I. J. Leontaritis, S. A. Billings, Input-output parametric models for non-linear systems Part II: stochastic non-linear systems, *International Journal of Control*, Vol. 41, No. 2, 1985, pp. 329–344
- [82] K. K. S. Leung, J. Y. C. Chiu, H. S. H. Chung, Boundary Control of Inverters Using Second-Order Switching Surface, *36th IEEE Power Electronics Specialists Conference pesc2005*, pp. 936–942, Recife, 2005
- [83] P. C. Loh, D. G. Holmes, A Variable Band Universal Flux/Charge Modulator for VSI and CSI Modulation, *IEEE IAS Annual Meeting*, Chicago, 2001
- [84] J. Lunze, *Regelungstechnik 1 (Control Technology 1)*, Berlin, Heidelberg: Springer-Verlag, 1996
- [85] J. Lunze, *Regelungstechnik 2 (Control Technology 2)*, Berlin, Heidelberg: Springer-Verlag, 1997

- [86] J. Maes, J. Melkebeek, Discrete Time Direct Torque Control of Induction Motors using Back-EMF measurement, *IEEE IAS Annual Meeting*, Vol. 1, pp. 407–414, St. Louis, 1998
- [87] H. R. Mayer, G. Pfaff, Direct Control of Induction Motor Currents—Design and Experimental Results, *European Conference on Power Electronics and Applications EPE 1985*, Vol. 2, pp. 3.7–3.12, Brüssel, 1985
- [88] M. Morari, J. H. Lee, Model predictive control: past, present and future, *Computers and Chemical Engineering*, Vol. 23, No. 4–5, 1999, pp. 667–682
- [89] C. El Moucary, E. Mendes, T. Raithel, R. Ortega, A New Strategy for Direct Torque and Flux Control of Induction Motor Drives, *International Conference on Electrical Machines ICEM2000*, Vol. 2, pp. 923–927, Espoo, 2000
- [90] P. Mutschler, Verfahren zur direkten Regelung der Geschwindigkeit eines elektrischen Antriebs (Principle for Direct Control of the Speed of an Electric Drive), *German Patent DE 196 35 981 C 2*, 1998
- [91] P. Mutschler, M. Marcks, A Direct Control Method for Matrix Converters, *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 2, 2002, pp. 362–369
- [92] I. Nagy, Novel Adaptive Tolerance Band Based PWM for Field-Oriented Control of Induction Machines, *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 4, 1994, pp. 406–417
- [93] K. Nakano, T. Yamamoto, T. Hinamoto, A Design of Robust Self-Tuning GPC-Based PID Controllers, *IEEE IECON2003*, pp. 285–290, Roanoke, 2003
- [94] M. E. Nillesen, J. L. Duarte, M. Pasquariello, A. Del Pizzo, Direct Torque Control with the Application of a Predictive Pulse Width Control, *IEEE IAS Annual Meeting*, Vol. 3, pp. 1375–1379, Rome, 2000
- [95] T. Noguchi, M. Yamamoto, S. Kondo, I. Takahashi, Enlarging Switching Frequency in Direct Torque-Controlled Inverter by Means of Dithering, *IEEE Transactions on Industry Applications*, Vol. 35, No. 6, 1999, pp. 1358–1366
- [96] T. Noguchi, H. Kodachi, I. Saito, High-Speed Current Vector Control of PWM Inverter Minimizing Current Error at Every Sampling Point, *IEEE IECON2002*, Sevilla, 2002
- [97] I.-H. Oh, Y.-S. Jung, M.-J. Youn, A Source Voltage-Clamped Resonant Link Inverter for a PMSM Using a Predictive Current Control Technique, *IEEE Transactions on Power Electronics*, Vol. 14, No. 6, 1999, pp. 1122–1132

-
- [98] G. Ose, G. Schiemann, H. Baumann, F. Stopp, *Ausgewählte Kapitel der Mathematik (Selected Units of Mathematics)*, Leipzig: VEB Fachbuchverlag, 1975
- [99] G. Pfaff, A. Wick, Direkte Stromregelung bei Drehstromantrieben mit Pulswechsellrichter (Direct Current Control of AC Drives with PWM Inverter), *rtp – Regelungstechnische Praxis*, Vol. 24, No. 11, 1983, pp. 472–477
- [100] A. Purcell, P.P. Acarnley, Multilevel hysteresis comparator forms for direct torque control schemes, *Electronic Letters*, Vol. 34, No. 6, 1998, pp. 601–603
- [101] A. Purcell, P.P. Acarnley, Enhanced Inverter Switching for Fast Response Direct Torque Control, *IEEE Transactions on Power Electronics*, Vol. 16, No. 3, 2001, pp. 382–389
- [102] J. B. Rawlings, Tutorial Overview of Model Predictive Control, *IEEE Control Systems Magazine*, June 2000, pp. 38–52
- [103] J. Rodríguez, J. Pontt, C. Silva, P. Cortés, U. Ammann, S. Rees, Predictive Direct Torque Control of an Induction Machine, *11th International Power Electronics and Motion Control Conference EPE-PEMC2004*, Riga, 2004
- [104] S. Salama, T. Yehia, Verfahren zur verbesserten Führung eines Drehstrom-Pulswechsellrichters (Principle for better Control of an Three-Phase PWM Inverter), *German Patent DE 40 21 766 C 2*, 1993
- [105] P. Schmitz, *Entwurf und Erprobung einer adaptiven prädiktiven Dreipunktregelung (Design and Proving of an Adaptive Predictive Three-Step Control)*, Fortschritt-Berichte VDI Series 8, No. 210, Düsseldorf: VDI-Verlag, 1990
- [106] M. Schoch, *Das Erweiterungsprinzip (The Expansion Strategy)*, Berlin: VEB Deutscher Verlag der Wissenschaften, 1976
- [107] H. Schwarz, *Mehrfachregelungen (Multi-Dimensional Control)*, Volume 1, Berlin, Heidelberg: Springer-Verlag, 1967
- [108] H. Schwarz, *Mehrfachregelungen (Multi-Dimensional Control)*, Volume 2, Berlin, Heidelberg: Springer-Verlag, 1971
- [109] D.E. Seborg, The Prospects for Advanced Process Control, *10th IFAC World Congress*, Munich, 1987
- [110] A. Sikorski, M. Korzeniewski, Improvement of Torque and Flux Control in DTC Method, *11th International Power Electronics and Motion Control Conference EPE-PEMC2004*, Riga, 2004

- [111] A. Steimel, J. Wiesemann, Further Development of Direct Self Control for Application in Electric Traction, *International Symposium on Industrial Electronics ISIE'96*, Vol. 1, pp. 180–185, Warschau, 1996
- [112] H. Stöcker, *DeskTop Stöcker: Taschenbuch mathematischer Formeln und moderner Verfahren (Pocket Book of Mathematical Formulas and Modern Methods)*, Thun: Harri Deutsch, 1999
- [113] I. Takahashi, T. Noguchi, A New Quick Response and High Efficiency Control Strategy of an Induction Motor, *IEEE IAS Annual Meeting*, pp. 496–502, Toronto, 1985
- [114] D. Telford, M. W. Dunnigan, B. W. Williams, A Novel Torque-Ripple Reduction Strategy for Direct Torque Control, *IEEE Transactions on Industrial Electronics*, Vol. 48, No. 4, 2001, pp. 867–870
- [115] J. Terno, *Numerische Verfahren der diskreten Optimierung (Numerical Methods of Discrete Optimization)*, Leipzig: B. G. Teubner, 1981
- [116] H. du Toit Mouton, J. H. R. Enslin, An Optimal On-Line-Tuning Current Regulator for High-Power IGBT Converters, *IEEE Transactions on Industry Applications*, Vol. 35, No. 5, 1999, pp. 1132–1140
- [117] P. Tøndel, T. A. Johansen, A. Bemporad, Evaluation of piecewise affine control via binary search tree, *Automatica*, Vol. 39, No. 5, 2003, pp. 945–950
- [118] A. M. Trzynadlowski, M. M. Bech, F. Blaabjerg, J. K. Pedersen, An Integral Space-Vector PWM Technique for DSP-Controlled Voltage-Source Inverters, *IEEE Transactions on Industry Applications*, Vol. 35, No. 5, 1999, pp. 1091–1097
- [119] T. T. C. Tsang, D. W. Clarke, Generalised predictive control with input constraints, *IEE Proceedings*, Vol. 135, Part D, No. 6, 1988, pp. 451–460
- [120] P. Vas, A. F. Stronach, M. Rashed, M. Zordan, B. C. Chew, DSP implementation of torque pulsation reduction techniques in sensorless DTC induction and P. M. synchronous motor drives, *Intelligent Motion Proceedings*, June 1999, pp. 107–111
- [121] H. Warmer, D. Schröder, An improved method of predictive control for line commutated DC-drives, *International Conference on Electrical Machines ICEM'86*, pp. 566–571, Munich, 1986
- [122] D. Wuest, F. Jenni, Space vector based current control schemes for voltage source inverters, *24th IEEE Power Electronics Specialists Conference pesc'93*, pp. 986–992, Seattle, 1993

- [123] E. Zafiriou, M. Morari, Design of robust digital controllers and sampling-time selection for SISO systems, *International Journal of Control*, Vol. 44, No. 3, 1986. pp. 711–735
- [124] R. Zurmühl, S. Falk, *Matrizen und ihre Anwendungen (Matrices and their Applications)*, Part 1: Basics, Berlin, Heidelberg: Springer-Verlag, 1984
- [125] unknown, *Polynomial and polynomial matrix glossary*, PolyX Ltd., <http://www.polyx.cz/glossary.htm>, Prague, 2002

Anhang/Appendix/附录 A

Glossar Polynommatrizen/Glossary Polynomial Matrices/词汇表 多项式矩阵

Nach [44] und [125].

Polynommatrix: Eine $k \times m$ -Polynommatrix ist eine Matrix der Form

$$\mathbf{P}(s) = \mathbf{P}_0 + \mathbf{P}_1 s + \mathbf{P}_2 s^2 + \cdots + \mathbf{P}_n s^n \quad (\text{A.1})$$

wobei s eine undefinierte, normalerweise komplexwertige Variable ist, und

$$\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n \quad (\text{A.2})$$

die konstanten $k \times m$ -Koeffizientenmatrizen darstellen. Wenn nicht anders angegeben, behandeln wir *reelle* Polynommatrizen, d. h. Polynommatrizen, deren Koeffizientenmatrizen reellwertig sind.

Ist \mathbf{P}_n keine Nullmatrix, hat $\mathbf{P}(s)$ den Grad n .

Ist \mathbf{P}_n eine Einheitsmatrix, wird $\mathbf{P}(s)$ *monisch* genannt.

Hoch und breit: Eine Polynommatrix ist *hoch*, wenn sie mindestens so viele Zeilen wie Spalten hat. Sie ist *breit*, wenn sie mindestens so viele Spalten wie Zeilen hat.

Rang: Eine Polynommatrix $\mathbf{P}(s)$ hat *vollen Spaltenrang* (oder *vollen normalen Spaltenrang*), wenn sie den vollen Spaltenrang überall außer an einer endlichen Zahl von Punkten in der komplexen Ebene hat. Ähnliche Definitionen gelten für *vollen Zeilenrang* und *vollen Rang*.

Der *normale Rang* einer Polynommatrix $\mathbf{P}(s)$ bedeutet:

$$\max_{s \in \mathbb{C}} \text{rg } \mathbf{P}(s) \quad (\text{A.3})$$

Ähnliche Definitionen betreffen den *normalen Spaltenrang* und *normalen Zeilenrang*.

Eine quadratische Polynommatrix ist *nichtsingulär*, wenn sie für fast alle komplexen Werte von s einen vollen normalen Rang hat.

From [44] and [125]:

Polynomial matrix: A $k \times m$ polynomial matrix is a matrix of the form (A.1), in which s is an undefined, normally complex variable and $\overline{\text{symbol (A.2)}}$ represent the constant $k \times m$ coefficient matrices. If nothing else is declared, *real* polynomial matrices, i. e. polynomial matrices whose coefficient matrices are real are treated.

If \mathbf{P}_n is not a zero matrix, then $\mathbf{P}(s)$ is of the degree n .

If \mathbf{P}_n is an identity matrix, $\mathbf{P}(s)$ is said to be *monic*.

Tall and wide: A polynomial matrix is *tall*, if it has at least as many rows as columns. It is *wide*, if it has at least as many columns as rows.

Rank: A polynomial matrix $\mathbf{P}(s)$ has *full column rank* (or *full normal column rank*), if it has full column rank everywhere in the complex plane, except at a finite number of points. Similar definitions can be given for *full row rank* and *full rank*.

The *normal rank* of a polynomial matrix $\mathbf{P}(s)$ is equal to $\overline{\text{symbol (A.3)}}$.

Similar definitions apply to the notions of *normal column rank* and *normal row rank*.

A square polynomial matrix is *nonsingular*, if it has full normal rank for almost all complex values of s .

依据 [44] 和 [125]:

多项式矩阵: 一个 $k \times m$ 多项式矩阵是具有形式 (A.1) 的矩阵, 其中 s 表示未定义的, 一般的复数变量, 并且符号 $\overline{\text{symbol (A.2)}}$ 表示 $k \times m$ 系数矩阵。除非特别说明, 我们讨论的是实数多项式矩阵, 其系数矩阵全部为实数。

如果 \mathbf{P}_n 为非零矩阵, 则 $\mathbf{P}(s)$ 是 n 阶的;

如果 \mathbf{P}_n 为单位矩阵, 则 $\mathbf{P}(s)$ 被称为首一的。

高和宽: 当一个多项式矩阵的行数不小于列数时, 则称之为高的; 反之, 当一个多项式矩阵的列数不小于行数时, 则称之为宽的。

秩: 当一个多项式矩阵 $\mathbf{P}(s)$ 在复平面内有限的几个点之外是列满秩的, 则它是列满秩的。类似的定义也适用于行满秩和满秩。

多项式矩阵 $\mathbf{P}(s)$ 的标准秩可表示为符号 $\overline{\text{symbol (A.3)}}$ 。类似的定义也适用于标准列秩和标准行秩。

如果一个二次的多项式矩阵的几乎全部复数值 s 具有标准秩, 那么, 它也是非奇异的。

Zeilen- und Spaltengrad: Seien die Elemente der $k \times m$ -Polynommatrix $\mathbf{P}(s)$

$$P_{i,j}(s), \quad i = 1, 2, \dots, k \quad j = 1, 2, \dots, m \quad (\text{A.4})$$

Dann sind die Werte

$$\rho_i = \max_j \deg P_{i,j}(s), \quad i = 1, 2, \dots, k \quad (\text{A.5})$$

$$\gamma_j = \max_i \deg P_{i,j}(s), \quad j = 1, 2, \dots, m \quad (\text{A.6})$$

der *Zeilen-* und *Spaltengrad* von $\mathbf{P}(s)$.

Führende Koeffizientenmatrix: Die Polynommatrix $\mathbf{P}(s)$ habe die Zeilen- und Spaltengrade

$$\rho_i, \quad i = 1, 2, \dots, k$$

$$\gamma_j, \quad j = 1, 2, \dots, m$$

Die *führende Spaltenkoeffizientenmatrix* von $\mathbf{P}(s)$ ist eine konstante Matrix, deren Element $e_{i,j}$ der Koeffizient des Terms zur Potenz γ_j aus dem polynomen Element $P_{i,j}(s)$ der Matrix $\mathbf{P}(s)$ ist. Die *führende Zeilenkoeffizientenmatrix* von $\mathbf{P}(s)$ ist eine konstante Matrix, deren Element $e_{i,j}$ der Koeffizient des Terms zur Potenz ρ_j aus dem polynomen Element $P_{i,j}(s)$ der Matrix $\mathbf{P}(s)$ ist.

Spalten- und zeilenreduziert: Eine Polynommatrix $\mathbf{P}(s)$ ist *spaltenreduziert*, wenn ihre führende Spaltenkoeffizientenmatrix vollen Spaltenrang hat. Sie ist *zeilenreduziert*, wenn ihre führende Zeilenkoeffizientenmatrix vollen Zeilenrang hat. Eine nichtsinguläre Polynommatrix $\mathbf{P}(s)$ ist *zeilenreduziert*, wenn und nur wenn ihre führende Zeilenkoeffizientenmatrix regulär (nichtsingulär) ist. Sie ist *spaltenreduziert*, wenn und nur wenn ihre führende Spaltenkoeffizientenmatrix regulär ist.

Jede nichtsinguläre Polynommatrix $\mathbf{P}(s)$ kann durch elementare Zeilenoperationen in zeilenreduzierte Form transformiert werden. Darüberhinaus kann die sich ergebende führende Zeilenkoeffizientenmatrix als eine untere Dreiecksmatrix mit Einsen auf der Hauptdiagonalen angenommen werden. Genauso kann $\mathbf{P}(s)$ durch elementare Spaltenoperationen in spaltenreduzierte Form umgewandelt werden. Dann kann die sich ergebende führende Spaltenkoeffizientenmatrix als eine obere Dreiecksmatrix mit Einsen auf der Hauptdiagonalen angenommen werden.

Row and column degrees: Let the elements of the $k \times m$ polynomial matrix $\mathbf{P}(s)$ be symbol (A.4). Then the numbers (A.5) and (A.6) are the *row* and the *column degrees* of $\mathbf{P}(s)$, respectively.

Leading coefficient matrix: Supposing that the polynomial matrix $\mathbf{P}(s)$ has the row and column degrees (A.5) and (A.6). The *leading column coefficient matrix* of $\mathbf{P}(s)$ is a constant matrix whose element $e_{i,j}$ is the coefficient of the term with power γ_j of the polynomial element $P_{i,j}(s)$ of the matrix $\mathbf{P}(s)$. The *leading row coefficient matrix* of $\mathbf{P}(s)$ is a constant matrix whose element $e_{i,j}$ is the coefficient of the term with power ρ_j of the polynomial element $P_{i,j}(s)$ of the matrix $\mathbf{P}(s)$.

Column and row reduced: A polynomial matrix is *column reduced* if its leading column coefficient matrix has full column rank. It is *row reduced* if its leading row coefficient matrix has full row rank. A nonsingular polynomial matrix $\mathbf{P}(s)$ is *row reduced* if, and only if, its leading row coefficient matrix is regular (non-singular). It is *column reduced* if, and only if, its leading column coefficient matrix is regular.

Any nonsingular polynomial matrix $\mathbf{P}(s)$ can be transformed into row reduced form by elementary row operations. Additionally, the resulting leading row coefficient matrix can assumed to be a lower triangular matrix with unity entries on the main diagonal. In the same way, $\mathbf{P}(s)$ can be converted into column-reduced form by elementary column operations. Then, the resulting leading column coefficient matrix can assumed to be an upper triangular matrix with unity entries on the main diagonal.

行阶数与列阶数: 如果一个 $k \times m$ 多项式矩阵 $\mathbf{P}(s)$ 的元素为符号 (A.4)。

那么, 数值 (A.5) 和 (A.6) 则分别表示 $\mathbf{P}(s)$ 的行阶数与列阶数。

前导系数矩阵: 假设多项式矩阵 $\mathbf{P}(s)$ 的行阶数与列阶数为数值 (A.5) 和 (A.6)。

$\mathbf{P}(s)$ 的前导列系数矩阵为常矩阵, 其元素 $e_{i,j}$ 是矩阵 $\mathbf{P}(s)$ 的多项式元素 $P_{i,j}(s)$ 的 γ_j 次幂的系数; $\mathbf{P}(s)$ 的前导行系数矩阵为常矩阵, 其元素 $e_{i,j}$ 是矩阵 $\mathbf{P}(s)$ 的多项式元素 $P_{i,j}(s)$ 的 ρ_j 次幂的系数;

列减的与行减的: 如果一个多项式矩阵 $\mathbf{P}(s)$ 的前导列系数矩阵是列满秩的, 那么它是列减的; 如果一个多项式矩阵 $\mathbf{P}(s)$ 的前导行系数矩阵是行满秩的, 那么它是行减的。当前仅当一个非奇异的多项式矩阵 $\mathbf{P}(s)$ 的前导行系数矩阵为非奇异的, 那么它才是行减的; 当前仅当一个非奇异的多项式矩阵 $\mathbf{P}(s)$ 的前导列系数矩阵为非奇异的, 那么它才是列减的;

任一非奇异的多项式矩阵 $\mathbf{P}(s)$ 可以通过基本的行运算转换为行减的形式。此外, 由此形成的前导行系数矩阵可以看作是主对角线上的元素全为 1 的下三角矩阵; 同理, 任一非奇异的多项式矩阵 $\mathbf{P}(s)$ 可以通过基本的列运算转换为列减的形式。此外, 由此形成的前导列系数矩阵可以看作是主对角线上的元素全为 1 的上三角矩阵。

Wurzeln: Die *Wurzeln* oder *Nullstellen* einer Polynommatrix $\mathbf{P}(s)$ sind die Stellen in der komplexen Ebene, wo $\mathbf{P}(s)$ einen Rangabfall erleidet.

Wenn $\mathbf{P}(s)$ quadratisch ist, dann sind die Wurzeln von $\mathbf{P}(s)$ gleichzeitig die Wurzeln der Determinante $\det \mathbf{P}(s)$, einschließlich Vielfachheit.

Prim: Eine Polynommatrix $\mathbf{P}(s)$ ist *links prim*, wenn sie überall in der komplexen Ebene einen vollen Zeilenrang hat. Die Matrix ist *rechts prim*, wenn überall in der komplexen Ebene einen vollen Spaltenrang hat.

Koprim (auch: relativ prim): Die N Polynommatrizen $\mathbf{P}_1(s), \mathbf{P}_2(s), \dots, \mathbf{P}_N(s)$ mit der gleichen Zeilenanzahl sind *links koprim* oder *relativ links prim*, wenn

$$\begin{bmatrix} \mathbf{P}_1(s) & \mathbf{P}_2(s) & \cdots & \mathbf{P}_N(s) \end{bmatrix} \quad (\text{A.7})$$

links prim ist. Wenn die N Polynommatrizen alle die gleiche Spaltenanzahl haben, sind sie *rechts koprim* oder *relativ rechts prim*, wenn

$$\begin{bmatrix} \mathbf{P}_1(s) \\ \mathbf{P}_2(s) \\ \vdots \\ \mathbf{P}_N(s) \end{bmatrix} \quad (\text{A.8})$$

rechts prim ist.

Zwei Matrizen $\mathbf{D}_L(s)$ und $\mathbf{N}_L(s)$ mit der gleichen Zeilenanzahl sind *relativ links prim* oder *links koprim*, wenn und nur wenn ihr größter gemeinsamer linker Teiler unimodular ist.

Unimodular: Eine *unimodulare Matrix* $\mathbf{U}(s)$ wird als eine beliebige, quadratische Matrix definiert, die aus der Einheitsmatrix durch eine endliche Anzahl elementarer Zeilen- und Spaltenoperationen gewonnen werden kann. Die Determinante $\det \mathbf{U}(s)$ ist daher ein reeller oder komplexwertiger, skalarer Wert ungleich Null und umgekehrt ist jede Polynommatrix, deren Determinante ein reeller oder komplexwertiger Wert ungleich Null ist, eine unimodulare Matrix.

Das Inverse einer Polynommatrix, eine *im Ringbereich liegende Kehrmatrix*, also eine solche, die wiederum Polynommatrix ist, existiert dann und nur dann, wenn die ursprüngliche Matrix unimodular ist [124].

Roots: The *roots* or *zeros* of a polynomial matrix $\mathbf{P}(s)$ are those points in the complex plane where $\mathbf{P}(s)$ loses rank.

If $\mathbf{P}(s)$ is square, then its roots are the roots of its determinant $\det \mathbf{P}(s)$, including multiplicity.

Prime: A polynomial matrix $\mathbf{P}(s)$ is *left prime* if it has full row rank everywhere in the complex plane. The matrix is *right prime* if it has full column rank everywhere in the complex plane.

Coprime (also: relative prime): N polynomial matrices $\mathbf{P}_1(s), \mathbf{P}_2(s), \dots, \mathbf{P}_N(s)$ with the same number of rows are *left coprime* or *relatively left prime*, if matrix (A.7) is left prime. If all N polynomial matrices have the same number of columns, then they are *right coprime* or *relatively right prime* if matrix (A.8) is right prime.

Two matrices $\mathbf{D}_L(s)$ and $\mathbf{N}_L(s)$ with the same number of rows are *relatively left prime* or *left coprime* if, and only if, their greatest common left divisor is unimodular. Two matrices $\mathbf{D}_R(s)$ and $\mathbf{N}_R(s)$ with the same number of columns are *relatively right prime* or *right coprime* if, and only if, their greatest common right divisor is unimodular.

Unimodular: A *unimodular matrix* $\mathbf{U}(s)$ is defined as an arbitrary square matrix which can be derived from the identity matrix by a finite number of elementary row and column operations. Therefore, the determinant $\det \mathbf{U}(s)$ is a nonzero real or complex scalar, and conversely any polynomial matrix whose determinant is a nonzero real or complex scalar is a unimodular matrix.

The inverse of a polynomial matrix, an *inverse matrix being in the ring area*, i. e. again a polynomial matrix, exists if, and only if, the original matrix is unimodular [124].

根: 多项式矩阵 $\mathbf{P}(s)$ 的根或零点表示其在复平面内丢失秩的点。如果 $\mathbf{P}(s)$ 是二次的, 那么 $\mathbf{P}(s)$ 的根同时也是其行列式 $\det \mathbf{P}(s)$ 的根, 且服从倍数关系。

素数: 如果一个多项式矩阵在整个复平面内是行满秩的, 那么它是左素的; 如果一个多项式矩阵在整个复平面内是列满秩的, 那么它是右素的。

互素 (或相对素数): 行数相同的 N 个多项式矩阵 $\mathbf{P}_1(s), \mathbf{P}_2(s), \dots, \mathbf{P}_N(s)$ 是左互素的, 只有当矩阵 (A.7) 是左素的; 列数相同的 N 个多项式矩阵是右互素的, 只有当矩阵 (A.8) 是右素的。

当且仅当两个行数相同的矩阵 $\mathbf{D}_L(s)$ 和 $\mathbf{N}_L(s)$ 的左最大公约数是单模的, 它们才是左互素的; 当且仅当两个行数相同的矩阵 $\mathbf{D}_R(s)$ 和 $\mathbf{N}_R(s)$ 的右最大公约数是单模的, 它们才是右互素的。

单模: 单模矩阵 $\mathbf{U}(s)$ 定义为一个任意的方阵, 该方阵是由单位矩阵经过有限数量的基本行运算和列运算变化得到的。因此, 行列式 $\det \mathbf{U}(s)$ 为非零的实数或复数值。反之, 如果一个多项式矩阵的行列式为非零的实数或复数值, 那么此多项式矩阵为单模矩阵。

当且仅当初始矩阵为单模的情况, 一个多项式矩阵的逆, 或者环形区域内的逆矩阵, 才同样是一个多项式矩阵 [124]。

Elementare Rechenregeln: Es gibt drei elementare Zeilenoperationen:

- Multiplikation einer Zeile mit einer reellen oder komplexwertigen Konstanten ungleich Null

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{Zeile 1} \cdot 3} \begin{bmatrix} 3 & 3s \\ 2 & s^2 \end{bmatrix}$$

- Vertauschen von zwei Zeilen

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{Zeile 1 und Zeile 2 vertauschen}} \begin{bmatrix} 2 & s^2 \\ 1 & s \end{bmatrix}$$

- Addieren eines polynomischen Vielfachen einer Zeile zu einer anderen

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{Zeile 1} = \text{Zeile 1} + s \cdot \text{Zeile 2}} \begin{bmatrix} 1 + 2s & s + s^3 \\ 2 & s^2 \end{bmatrix}$$

Rechenregeln für elementare Spaltenumformungen sind analog definiert.

Es sei darauf hingewiesen, dass elementare Zeilen- oder Spaltenoperationen einer Polynommatrix $\mathbf{P}(s)$ auch realisiert werden können, indem die identischen Operationen mit einer Einheitsmatrix \mathbf{I} durchgeführt werden und dann $\mathbf{P}(s)$ mit dem Ergebnis prämultipliziert wird.

Elementary row and column operations: Three elementary row operations exists:

- Multiplication of a row with a real or complex-valued constant unequal to zero:

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{row 1} \cdot 3} \begin{bmatrix} 3 & 3s \\ 2 & s^2 \end{bmatrix}$$

- Interchanging of two rows:

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{interchange row 1 and row 2}} \begin{bmatrix} 2 & s^2 \\ 1 & s \end{bmatrix}$$

- Addition of a polynomial multiple of one row to another one:

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{row 1} = \text{row 1} + s \cdot \text{row 2}} \begin{bmatrix} 1 + 2s & s + s^3 \\ 2 & s^2 \end{bmatrix}$$

Elementary column operations are defined analogously.

It shall be noted that the elementary row or column operations of a polynomial matrix $\mathbf{P}(s)$ can also be realized if identical operations are carried out on an identity matrix \mathbf{I} and then $\mathbf{P}(s)$ is premultiplied with the result.

基本运算法则: 存在三种基本的行运算法则:

- 某一行乘以一个非零的实数或复数值

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{行 1} \cdot 3} \begin{bmatrix} 3 & 3s \\ 2 & s^2 \end{bmatrix}$$

- 行与行之间的互换

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{第 1 行和第 2 行互换}} \begin{bmatrix} 2 & s^2 \\ 1 & s \end{bmatrix}$$

- 将某一行的多项式倍数加到另外一行

$$\begin{bmatrix} 1 & s \\ 2 & s^2 \end{bmatrix} \xrightarrow{\text{第 1 行} = \text{第 1 行} + s \cdot \text{第 2 行}} \begin{bmatrix} 1 + 2s & s + s^3 \\ 2 & s^2 \end{bmatrix}$$

基本的列运算法则的定义与行运算类似。

需要指出的是, 多项式矩阵 $\mathbf{P}(s)$ 的基本行运算或列运算也是可以实现的, 只需首先对单位矩阵 \mathbf{I} 实施相同的运算法则, 然后将其结果与 $\mathbf{P}(s)$ 相乘即可。

Diophantische Gleichung: Die einfachste Form einer linearen, skalaren Polynomgleichung – die sogenannte *Diophantische Gleichung*, benannt nach dem Alexandrinischen Mathematiker Diophantos (275 v. Chr.) – ist:

$$A(s)X(s) + B(s)Y(s) = C(s) \quad (\text{A.9})$$

Die Polynome $A(s)$, $B(s)$ und $C(s)$ sind gegeben, während die Polynome $X(s)$ und $Y(s)$ unbekannt sind. Die Gleichung ist lösbar, wenn und nur wenn der größte gemeinsame Teiler von $A(s)$ und $B(s)$ auch Teiler von $C(s)$ ist. Dies impliziert, dass wenn $A(s)$ und $B(s)$ koprim sind, die Gleichung für jede rechte Gleichungsseite, einschließlich $C(s) = 1$ lösbar ist.

Immer, wenn die Diophantische Gleichung lösbar ist, besitzt sie unendlich viele Lösungen. Wenn $(X_0(s), Y_0(s))$ eine (partikuläre) Lösung ist, dann lautet die allgemeine Lösung

$$X(s) = X_0(s) + \overline{B}(s)T(s) \quad (\text{A.10})$$

$$Y(s) = Y_0(s) - \overline{A}(s)T(s) \quad (\text{A.11})$$

wobei $T(s)$ ein beliebiges Polynom (der Parameter) ist und $\overline{A}(s)$ und $\overline{B}(s)$ koprim Polynome sind, für die gilt

$$\frac{\overline{B}(s)}{\overline{A}(s)} = \frac{B(s)}{A(s)} \quad (\text{A.12})$$

Wenn $A(s)$ und $B(s)$ selbst koprim sind, kann man natürlich auch wählen

$$\overline{A}(s) = A(s), \quad \overline{B}(s) = B(s) \quad (\text{A.13})$$

Unter allen Lösungen einer Diophantischen Gleichung existiert ein einmaliges Lösungspaar $(X(s), Y(s))$, gekennzeichnet durch

$$\deg X(s) < \deg \overline{B}(s) \quad (\text{A.14})$$

Es gibt ein anderes – üblicherweise verschiedenes – Lösungspaar mit

$$\deg Y(s) < \deg \overline{A}(s) \quad (\text{A.15})$$

Die beiden Lösungspaare sind nur identisch, wenn

$$\deg A(s) + \deg B(s) \geq \deg C(s) \quad (\text{A.16})$$

Diophantine equation: The most simple form of a linear, scalar polynomial equation—the so-called *Diophantine equation* named after the Alexandrian mathematician Diophantos (A. D. 275)—is equation (A.9). The polynomials $A(s)$, $B(s)$ and $C(s)$ are given, while the polynomials $X(s)$ and $Y(s)$ are unknown. The equation is solvable if, and only if, the greatest common divisor of $A(s)$ and $B(s)$ is also a divisor of $C(s)$. This implies that the equation is solvable for any right hand side polynomial including $C(s) = 1$, if $A(s)$ and $B(s)$ are coprime.

If a Diophantine equation is solvable, it always has an infinite number of solutions. If $(X_0(s), Y_0(s))$ is any (particular) solution, then the general solution is equations (A.10) and (A.11), where $T(s)$ is an arbitrary polynomial (the parameter) and $\bar{A}(s)$ and $\bar{B}(s)$ are coprime polynomials to which the following applies, see equation (A.12).

If $A(s)$ and $B(s)$ themselves are coprime, then one can select equation (A.13). Among all solutions of a Diophantine equation, a unique solution pair $(X(s), Y(s))$ exists, characterized by equation (A.14). There is another—usually different—solution pair with equation (A.15). Both solution pairs are only identical if (A.16).

丢番图方程: 一个具有最简单形式的线性的标量多项式方程被称之为丢番图方程, 它是以亚历山大港的数学家丢番图 (公元前 275 年) 命名的, 表达式为方程 (A.9)。其中的多项式 $A(s)$, $B(s)$ 和 $C(s)$ 是已知的, 而多项式 $X(s)$ 和 $Y(s)$ 则是未知的。丢番图方程可解的充要条件是: $A(s)$ 和 $B(s)$ 的最大公约数为 $C(s)$ 的约数。这也亦为之, 当 $A(s)$ 和 $B(s)$ 互素时, 上述方程对于任意的 $C(s)$ (包含 $C(s) = 1$) 都是有解的。

只要丢番图方程是有解的, 那么它总是有无穷多个解。假使 $(X_0(s), Y_0(s))$ 为其中的一个特解, 那么其通解形式为方程 (A.10) 和 (A.11), 其中 $T(s)$ 为任意的多项式, $\bar{A}(s)$ 和 $\bar{B}(s)$ 是互素的多项式, 且满足方程 (A.12)。

如果 $A(s)$ 和 $B(s)$ 本身也是互素的, 显然可以选择方程 (A.13)。在丢番图方程的所有解中存在一个唯一的解对数 $(X(s), Y(s))$, 其中一个可表示为方程 (A.14)。通常来说, 另外的一个解对数可表示为方程 (A.15)。只有满足条件 (A.16) 时, 上述两个解对数才可能是相同的。

Bézout-Gleichung: Eine Diophantische Gleichung mit 1 auf der rechten Seite wird *Bézout-Gleichung* genannt. Sie könnte z. B. lauten

$$A(s)X(s) + B(s)Y(s) = 1 \tag{A.17}$$

wobei $A(s)$ und $B(s)$ gegebene Polynome sowie $X(s)$ und $Y(s)$ unbekannt sind.

Zwei Polynommatrizen $\mathbf{D}_L(s)$ und $\mathbf{N}_L(s)$ mit der gleichen Zeilenzahl sind links koprim, wenn und nur wenn zwei Polynommatrizen $\mathbf{A}(s)$ und $\mathbf{B}(s)$ existieren, die die Bézout-Gleichung

$$\mathbf{D}_L(s)\mathbf{A}(s) + \mathbf{N}_L(s)\mathbf{B}(s) = \mathbf{I} \tag{A.18}$$

erfüllen.

Zwei Polynommatrizen $\mathbf{D}_R(s)$ und $\mathbf{N}_R(s)$ mit der gleichen Spaltenzahl sind rechts koprim, wenn und nur wenn zwei Polynommatrizen $\mathbf{A}(s)$ und $\mathbf{B}(s)$ existieren, die die Bézout-Gleichung

$$\mathbf{A}(s)\mathbf{D}_R(s) + \mathbf{B}(s)\mathbf{N}_R(s) = \mathbf{I} \tag{A.19}$$

erfüllen.

Teiler und Vielfache: Seien die Polynome $A(s), B(s)$ und $C(s)$ gegeben, so dass die Beziehung $A(s) = B(s) \cdot C(s)$ lautet. In diesem Fall wird $B(s)$ Teiler von $A(s)$ genannt und $A(s)$ ein Vielfaches von $B(s)$. Man schreibt $B(s)|A(s)$. Man sagt auch $B(s)$ teilt $A(s)$.

Wenn ein Polynom $G(s)$ sowohl $A(s)$ als auch $B(s)$ teilt, dann wird $G(s)$ als gemeinsamer Teiler von $A(s)$ und $B(s)$ bezeichnet. Wenn darüberhinaus $G(s)$ ein Vielfaches jedes gemeinsamen Teilers von $A(s)$ und $B(s)$ ist, dann ist $G(s)$ der größte gemeinsame Teiler (ggT) von $A(s)$ und $B(s)$. Wenn die einzigen gemeinsamen Teiler von $A(s)$ und $B(s)$ Konstanten sind, dann sind $A(s)$ und $B(s)$ koprim.

Wenn ein Polynom $M(s)$ ein Vielfaches sowohl von $A(s)$ als auch von $B(s)$ ist, dann wird $M(s)$ gemeinsames Vielfaches von $A(s)$ und $B(s)$ genannt. Wenn darüberhinaus $M(s)$ ein Teiler aller gemeinsamen Vielfachen von $A(s)$ und $B(s)$ ist, so ist $M(s)$ das kleinste gemeinsame Vielfache (kgV) von $A(s)$ und $B(s)$.

Nun seien Polynommatrizen $\mathbf{A}(s), \mathbf{B}(s)$ und $\mathbf{C}(s)$ von zueinander passender Größe gegeben, so dass $\mathbf{A}(s) = \mathbf{B}(s) \cdot \mathbf{C}(s)$ gilt. In diesem Fall wird $\mathbf{B}(s)$ linker Teiler von $\mathbf{A}(s)$ und $\mathbf{A}(s)$ rechtes Vielfaches von $\mathbf{B}(s)$ genannt.

Bézout equation: A Diophantine equation with 1 on the right hand side is called *Bézout equation*. It could e.g. have the form (A.17), in which $A(s)$ and $B(s)$ are given polynomials and $X(s)$ and $Y(s)$ are unknown. Two polynomial matrices $\mathbf{D}_L(s)$ and $\mathbf{N}_L(s)$ with the same number of rows are left coprime if, and only if, two

polynomial matrices $\mathbf{A}(s)$ and $\mathbf{B}(s)$ exist, which fulfil the Bézout equation (A.18). Two polynomial matrices $\mathbf{D}_R(s)$ and $\mathbf{N}_R(s)$ with the same number of columns are right coprime if, and only if, two polynomial matrices $\mathbf{A}(s)$ and $\mathbf{B}(s)$ exist, which fulfil the Bézout equation (A.19).

Divisors and multiples: Let the polynomials $A(s)$, $B(s)$ and $C(s)$ be given such that $A(s) = B(s) \cdot C(s)$. In this case, $B(s)$ is called divisor of $A(s)$ and $A(s)$ is called multiple of $B(s)$. It can be written as $B(s)|A(s)$. This is also stated as $B(s)$ divides $A(s)$.

If a polynomial $G(s)$ divides both $A(s)$ and $B(s)$, then $G(s)$ is called common divisor of $A(s)$ and $B(s)$. If additionally, $G(s)$ is a multiple of every common divisor of $A(s)$ and $B(s)$, then $G(s)$ is the greatest common divisor (GCD) of $A(s)$ and $B(s)$. If the only common divisors of $A(s)$ and $B(s)$ are constants, then $A(s)$ and $B(s)$ are coprime.

If a polynomial $M(s)$ is a multiple of both $A(s)$ and $B(s)$, then $M(s)$ is called common multiple of $A(s)$ and $B(s)$. If additionally, $M(s)$ is a divisor of all common multiples of $A(s)$ and $B(s)$, then $M(s)$ is the least common multiple (LCM) of $A(s)$ and $B(s)$.

Now let the polynomial matrices $\mathbf{A}(s)$, $\mathbf{B}(s)$ and $\mathbf{C}(s)$ of compatible size be given such that $\mathbf{A}(s) = \mathbf{B}(s) \cdot \mathbf{C}(s)$. In this case, $\mathbf{B}(s)$ is called the left divisor of $\mathbf{A}(s)$ and $\mathbf{A}(s)$ is called the right multiple of $\mathbf{B}(s)$.

贝祖方程: 等式右侧为 1 的丢番图方程被称为贝祖方程, 它具有形式 (A.17), 其中 $A(s)$ 和 $B(s)$ 是已知的多项式, $X(s)$ 和 $Y(s)$ 是未知的。具有相同行数的两个多项式矩阵 $\mathbf{D}_L(s)$ 和 $\mathbf{N}_L(s)$ 为左互素的充要条: 两个多项式矩阵 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 满足贝祖方程 (A.18); 具有相同列数的两个多项式矩阵 $\mathbf{D}_R(s)$ 和 $\mathbf{N}_R(s)$ 为右互素的充要条: 两个多项式矩阵 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 满足贝祖方程 (A.19);

约数与倍数: 假设给定的多项式 $A(s)$, $B(s)$ 和 $C(s)$ 满足 $A(s) = B(s) \cdot C(s)$, 那么则称 $B(s)$ 为 $A(s)$ 的约数, 同时 $A(s)$ 为 $B(s)$ 的倍数。通常可表示为 $B(s)|A(s)$, 并可理解为 $B(s)$ 整除 $A(s)$ 。

如果一个多项式 $G(s)$ 既可以整除 $A(s)$, 也可以整除 $B(s)$, 那么 $G(s)$ 则为 $A(s)$ 和 $B(s)$ 的公约数。此外, 如果 $G(s)$ 是 $A(s)$ 和 $B(s)$ 任一公约数的倍数, 那么 $G(s)$ 为 $A(s)$ 和 $B(s)$ 的最大公约数。如果 $A(s)$ 和 $B(s)$ 唯一的公约数为常数, 那么 $A(s)$ 和 $B(s)$ 互素。

如果一个多项式 $M(s)$ 同时为 $A(s)$ 和 $B(s)$ 的倍数, 那么则称 $M(s)$ 为 $A(s)$ 和 $B(s)$ 的公倍数。此外, 如果 $M(s)$ 是 $A(s)$ 和 $B(s)$ 所有公倍数的约数, 那么 $M(s)$ 为 $A(s)$ 和 $B(s)$ 的最小公倍数。

接下来, 假设多项式矩阵 $\mathbf{A}(s)$, $\mathbf{B}(s)$ 和 $\mathbf{C}(s)$ 具有合适的大小, 使得满足 $\mathbf{A}(s) = \mathbf{B}(s) \cdot \mathbf{C}(s)$ 。此时, 则称 $\mathbf{B}(s)$ 为 $\mathbf{A}(s)$ 的左约数, $\mathbf{A}(s)$ 为 $\mathbf{B}(s)$ 的右倍数。

Teiler und Vielfache: Wenn eine Polynommatrix $\mathbf{G}(s)$ ein linker Teiler sowohl von $\mathbf{A}(s)$ als auch von $\mathbf{B}(s)$ ist, so nennt man $\mathbf{G}(s)$ gemeinsamen linken Teiler von $\mathbf{A}(s)$ und $\mathbf{B}(s)$. Wenn darüberhinaus $\mathbf{G}(s)$ ein rechtes Vielfaches jedes gemeinsamen linken Teilers von $\mathbf{A}(s)$ und $\mathbf{B}(s)$ ist, dann nennt man $\mathbf{G}(s)$ größten gemeinsamen linken Teiler von $\mathbf{A}(s)$ und $\mathbf{B}(s)$. Wenn die einzigen gemeinsamen linken Teiler von $\mathbf{A}(s)$ und $\mathbf{B}(s)$ unimodulare Matrizen sind, dann sind die Polynommatrizen $\mathbf{A}(s)$ und $\mathbf{B}(s)$ links koprim. Wenn eine Polynommatrix $\mathbf{M}(s)$ ein rechtes Vielfaches sowohl von $\mathbf{A}(s)$ als auch von $\mathbf{B}(s)$ ist, nennt man $\mathbf{M}(s)$ gemeinsames rechtes Vielfaches von $\mathbf{A}(s)$ und $\mathbf{B}(s)$. Wenn darüberhinaus $\mathbf{M}(s)$ ein linker Teiler aller gemeinsamen rechten Vielfachen von $\mathbf{A}(s)$ und $\mathbf{B}(s)$ ist, so ist $\mathbf{M}(s)$ das kleinste gemeinsame rechte Vielfache von $\mathbf{A}(s)$ und $\mathbf{B}(s)$. Rechte Teiler, linke Vielfache, gemeinsame rechte Teiler, größte gemeinsame rechte Teiler, gemeinsame linke Vielfache und kleinste gemeinsame linke Vielfache sind ähnlich definiert.

Weitere Eigenschaften: Aus Gründen der Einfachheit werden hier nur die Eigenschaften entweder für linke oder rechte Matrixoperationen beschrieben. Die zugehörigen Ergebnisse für den anderen Fall können durch Ersetzen von rechts, Zeile, Spalte, prämultiplizieren usw. durch links, Spalte, Zeile bzw. postmultiplizieren hergeleitet werden. Jede $m \times l$ -Polynommatrix $\mathbf{P}(s)$ vom Rang r kann durch elementare Spaltenoperationen auf eine untere linke Dreiecksmatrix reduziert werden, in der gilt:

- Wenn $l > s$ so sind die letzten $(l - s)$ Spalten alle Null.
- In Zeile j mit $1 \leq j \leq s$ ist das Diagonalelement monisch und von höherem Grad als alle Elemente ungleich Null links von ihm.
- Wenn in Zeile j mit $1 \leq j \leq s$ das Diagonalelement Eins ist, so sind alle Elemente links davon gleich Null.

Gegeben seien zwei $m \times m$ - und $m \times r$ -Polynommatrizen $\mathbf{D}_L(s)$ und $\mathbf{N}_L(s)$. Wenn die Matrix $\begin{bmatrix} \mathbf{D}_L(s) & \mathbf{N}_L(s) \\ \mathbf{R}(s) & 0 \end{bmatrix}$ gemäß obiger Beschreibung auf eine untere linke Dreiecksform $\begin{bmatrix} \mathbf{R}(s) & 0 \end{bmatrix}$ reduziert wird, dann ist $\mathbf{R}(s)$ größter gemeinsamer linker Teiler von $\mathbf{D}_L(s)$ und $\mathbf{N}_L(s)$.

Divisors and multiples: If a polynomial matrix $\mathbf{G}(s)$ is a left divisor of $\mathbf{A}(s)$ and $\mathbf{B}(s)$, then $\mathbf{G}(s)$ is called common left divisor of $\mathbf{A}(s)$ and $\mathbf{B}(s)$. If additionally, $\mathbf{G}(s)$ is a right multiple of every common left divisor of $\mathbf{A}(s)$ and $\mathbf{B}(s)$, then $\mathbf{G}(s)$ is the largest common left divisor of $\mathbf{A}(s)$ and $\mathbf{B}(s)$. If the only common left divisors of $\mathbf{A}(s)$ and $\mathbf{B}(s)$ are unimodular matrices, then the polynomial matrices $\mathbf{A}(s)$ and $\mathbf{B}(s)$ are left coprime. If a polynomial matrix $\mathbf{M}(s)$ is a right multiple of $\mathbf{A}(s)$ and $\mathbf{B}(s)$, then $\mathbf{M}(s)$ is called common right multiple of $\mathbf{A}(s)$ and $\mathbf{B}(s)$. If additionally, $\mathbf{M}(s)$ is a left divisor of all common right multiples of $\mathbf{A}(s)$ and $\mathbf{B}(s)$, then $\mathbf{M}(s)$ is the smallest common right multiple of $\mathbf{A}(s)$ and $\mathbf{B}(s)$. Right divisors, left multiples,

common right divisors, greatest common right divisors, common left multiples and smallest common left multiples are defined in a similar way.

Other properties: For the sake of simplicity, the properties described here are either for left or for right matrix operations. The corresponding results for the other case can be derived by substituting right, row, column, premultiplication, etc. with left, column, row or postmultiplication. Any $m \times l$ polynomial matrix $\mathbf{P}(s)$ of the rank r can be reduced by elementary column operations to a lower left triangular matrix in which the following applies:

- If $l > s$, then the last $(l - s)$ columns are all zero.
- In row j with $1 \leq j \leq s$ the diagonal element is monic and of a higher degree than any (nonzero) element to the left of it.
- If in row j with $1 \leq j \leq s$ the diagonal element is unity, then all elements to the left of it are zero.

Given are the two $m \times m$ and $m \times r$ polynomial matrices $\mathbf{D}_L(s)$ and $\mathbf{N}_L(s)$. If the matrix $[\mathbf{D}_L(s) \quad \mathbf{N}_L(s)]$ is reduced to lower left triangular form $[\mathbf{R}(s) \quad 0]$ according to the above description, then $\mathbf{R}(s)$ is the greatest common left divisor of $\mathbf{D}_L(s)$ and $\mathbf{N}_L(s)$.

约数与倍数: 如果一个多项式矩阵 $\mathbf{G}(s)$ 同时为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 的左约数, 那么则称 $\mathbf{G}(s)$ 为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 的左公约数。此外, 如果 $\mathbf{G}(s)$ 为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 任一左公约数的右倍数, 那么则称 $\mathbf{G}(s)$ 为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 的最大左公约数。如果 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 唯一的左公约数为单模矩阵, 那么多项式矩阵 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 是左互素的。如果一个多项式矩阵 $\mathbf{M}(s)$ 同时为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 的右倍数, 那么则称 $\mathbf{M}(s)$ 为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 的右公倍数。此外, 如果 $\mathbf{M}(s)$ 为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 任一右公倍数的左约数, 那么则称 $\mathbf{G}(s)$ 为 $\mathbf{A}(s)$ 和 $\mathbf{B}(s)$ 的最大右公倍数。右约数、左倍数、右公约数、最大右公约数、左公倍数、最小左公倍数的定义与前述类似, 此处不再赘述。

其他特性: 为简化起见, 本部分仅描述左或右矩阵运算的相关特性。至于其他的情况所对应的结果, 可以简单地使用右、行、列、前乘等来代替左、列、行、后乘等直接推导得出。任一为秩为 r 的 $m \times l$ 多项式矩阵 $\mathbf{P}(s)$ 可以通过基本的列运算缩减为下左三角矩阵, 且满足下述条件:

- 如果 $l > s$, 那么最后 $(l - s)$ 列的元素全为零;
- 当 $1 \leq j \leq s$ 时, 第 j 行中对角线上的元素是单调的且在其左侧的、具有更高阶数的所有元素均不等于零。
- 当 $1 \leq j \leq s$ 时, 第 j 行中对角线上的元素是全为 1, 那么在其左侧的所有元素均等于零。

给定两个 $m \times m$ 和 $m \times r$ 多项式矩阵 $\mathbf{D}_L(s)$ 和 $\mathbf{N}_L(s)$, 如果基于上面的描述可将矩阵 $[\mathbf{D}_L(s) \quad \mathbf{N}_L(s)]$ 缩减为一个下左三角矩阵 $[\mathbf{R}(s) \quad 0]$, 那么 $\mathbf{R}(s)$ 为 $\mathbf{D}_L(s)$ 和 $\mathbf{N}_L(s)$ 的最大左公约数。

Anhang/Appendix/附录 B

Nomenklatur/Nomenclature/术语表

Die Bezeichnung der einzelnen Variablen, Parameter etc. erfolgt nach folgender Konvention:

skalare Werte	: kursive Schrift (a, b, c)
komplexe Werte	: kursive Fettschrift ($\mathbf{a}, \mathbf{b}, \mathbf{c}$)
Vektoren	: fette Kleinbuchstaben ($\mathbf{a}, \mathbf{b}, \mathbf{c}$)
Matrizen	: fette Großbuchstaben ($\mathbf{A}, \mathbf{B}, \mathbf{C}$)
Mengen	: kursive, fette Großbuchstaben ($\mathbf{A}, \mathbf{B}, \mathbf{C}$)
Polynome in z	: Großbuchstaben ($A(z^{-1}), B(z^{-1}), C(z^{-1})$)
Polynommatrizen in z	: fette Großbuchst. ($\mathbf{A}(z^{-1}), \mathbf{B}(z^{-1}), \mathbf{C}(z^{-1})$)
Polytope	: geschwungene Großbuchstaben ($\mathcal{A}, \mathcal{B}, \mathcal{C}$)
Zahlenbereiche	: Großbuchstaben mit Doppelstrich ($\mathbb{A}, \mathbb{B}, \mathbb{C}$)
Sollwerte	: hochgestelltes Sternchen ($a^*, \mathbf{a}^*, \mathbf{a}^*$)

Übersicht über die verwendeten Formelzeichen:

Maschinengrößen

a	Beschleunigung
a	Schaltzustand der Halbbrücke zu Strang a
b	Schaltzustand der Halbbrücke zu Strang b
c	Schaltzustand der Halbbrücke zu Strang c
d	Verzerrungsfaktor
e	Regelabweichung, Regelfehler
f_s	Schaltfrequenz des Umrichters, nicht normiert
I	Strom, allgemein
I_R	Bemessungsstrom der Maschine
$I_{Str,R}$	Bemessungs-Strangstrom
i_s	Statorstrom
i_{sa}	Statorstrom im Strang a
i_{sb}	Statorstrom im Strang b
i_{sc}	Statorstrom im Strang c
i_{sd}	feldbildender Anteil des Statorstroms

i_{sq}	drehmomentbildender Anteil des Statorstroms
$i_{s\alpha}$	Realteil von i_s in Statorkoordinaten
$i_{s\beta}$	Imaginärteil von i_s in Statorkoordinaten
k_r	Kopplungsfaktor des Rotors
k_s	Kopplungsfaktor des Ständers
l_h	Hauptinduktivität der Maschine
l_r	Rotorinduktivität
l_r'	transiente Rotorinduktivität
$l_{r\sigma}$	Streuinduktivität des Rotors
l_s	Ständerinduktivität
l_s'	transiente Ständerinduktivität
$l_{s\sigma}$	Streuinduktivität des Ständers
m	Modulationsgrad
p	Polpaarzahl
r_r	ohmscher Widerstand des Rotors
r_s	ohmscher Widerstand des Ständers
r_σ	wirksamer Widerstandswert beider Wicklungen
S	Schaltzustand des Umrichters
T_0	Abtastrate, nicht normiert
U_R	Bemessungsspannung der Maschine
$U_{Str,R}$	Bemessungs-Strangspannung
u_d	Zwischenkreisspannung
u_s	Statorspannung
u_{sd}	feldbildender Anteil der Statorspannung
u_{sq}	drehmomentbildender Anteil der Statorspannung
$u_{s\alpha}$	Realteil von u_s in Statorkoordinaten
$u_{s\beta}$	Imaginärteil von u_s in Statorkoordinaten
δ	Feldwinkel
σ	totale Streuziffer
τ_0	Abtastrate
τ_f	Zeitkonstante des Istwertfilters
τ_m	mechanische Zeitkonstante
τ_r	Rotorzeitkonstante
τ_r'	transiente Rotorzeitkonstante
τ_s	Ständerzeitkonstante
τ_s'	transiente Ständerzeitkonstante
τ_{tot}	Totzeit
τ_σ'	transiente Zeitkonstante
φ	mechanische Lage des Rotors
ψ_r, ψ_{rd}	Rotorfluss

$\psi_{r\alpha}$	Realteil von ψ_r in Statorkoordinaten
$\psi_{r\beta}$	Imaginärteil von ψ_r in Statorkoordinaten
ψ_s	Ständerfluss
ω	mechanische Geschwindigkeit des Rotors
ω_k	Winkelgeschwindigkeit des Koordinatensystems
ω_r	Winkelgeschwindigkeit des Rotorstroms
ω_s	Winkelgeschwindigkeit des Statorstroms
$\omega_{s,R}$	Bemessungs-Winkelgeschwindigkeit des Statorstroms

Reglergrößen PI-Regler

T_i	Integrationszeitkonstante der Stromregler
T_ω	Integrationszeitkonstante des Drehzahlreglers
V_i	Verstärkung der Stromregler
V_ω	Verstärkung des Drehzahlreglers

MPC-Variablen (SISO)

$A(z^{-1})$	Nennerpolynom der Übertragungsfunktion $G(z^{-1})$
$\hat{A}(z^{-1})$	modifiziertes Nennerpolynom; $\hat{A}(z^{-1}) = \Delta A(z^{-1})$
$B(z^{-1})$	Zählerpolynom der Übertragungsfunktion $G(z^{-1})$
$C(z^{-1})$	Noise-Polynom
d	diskrete Verzögerungszeit
$E_j(z^{-1})$	Hilfspolynom
$F_j(z^{-1})$	Hilfspolynom
$\mathbf{F}(z^{-1})$	Übertragungs-Polynommatrix für vergangene Regelgrößen
\mathbf{FG}'	Hilfsmatrix zur Berechnung der freien Antwort
$\mathbf{F}\Gamma$	Hilfsmatrix zur Berechnung der freien Antwort b. Filterung
$f(t)$	freie Antwort
\mathbf{f}	Vektor der freien Antworten $f(t)$
$f'(t)$	freie Antwort bei Filterung
\mathbf{f}'	Vektor der freien Antworten bei Filterung
\mathbf{G}	Übertragungsmatrix für die erzwungene Antwort
$G(z^{-1})$	Übertragungsfunktion des Systems
\mathbf{G}'	Übertragungsmatrix für erzwungene Antwort bei Filterung
$\mathbf{G}'(z^{-1})$	Übertragungs-Polynommatrix für vergangene Steuergrößen
$G_j(z^{-1})$	Übertragungspolynom für GPC-Regler
$G_j'(z^{-1})$	Übertragungsfunktion für zukünftige Steuergrößen
\mathbf{g}_i^T	erste Zeile der Matrix $(\mathbf{G}'^T \mathbf{G} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}^T$
\mathbf{g}^T	erste Zeile der Matrix $(\mathbf{G}'^T \mathbf{G}' + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}'^T$
\mathbf{I}	Einheitsmatrix
i	Zählvariable

J	Bewertungskriterium
j	Zählvariable
k	Zählvariable
N	Horizont, allgemein
N_1	unterer Kostenhorizont
N_2	oberer Kostenhorizont
N_p	Prädiktionshorizont
N_u	Regelhorizont
na	Grad des Nennerpolynoms $A(z^{-1})$
nb	Grad des Zählerpolynoms $B(z^{-1})$
nc	Grad des Noise-Polynoms $C(z^{-1})$
nt	Grad des Design-Polynoms $T(z^{-1})$
r_j	Hilfsgröße zur Berechnung von $E_j(z^{-1})$
$\bar{R}(z^{-1})$	Hilfpolynom zur Berechnung von $E_j(z^{-1})$
$T(z^{-1})$	Design-Polynom zur Filterung
$u(t)$	Stellgröße
$u^f(t)$	mit $T(z^{-1})$ gefilterte Stellgröße
\mathbf{u}	Vektor der Stellgrößendifferenzen $\Delta u(t)$
$w(t)$	Führungsgröße
\mathbf{w}	Vektor der Führungsgrößen $w(t)$
$y(t)$	Regelgröße
$y^f(t)$	mit $T(z^{-1})$ gefilterte Regelgröße
$\hat{y}(t)$	prädizierte Regelgröße
\mathbf{y}	Vektor der prädizierten Regelgrößen $\hat{y}(t)$
\mathbf{yu}	Vektor vergangener Regel- und Stellgrößen $y(t)$ und $u(t)$
\mathbf{yu}^f	Vektor gefilterter Regel- und Stellgrößen $y^f(t)$ und $u^f(t)$
$\Gamma(z^{-1})$	Übertragungs-Polynommatrix für gefilterte Steuergrößen
$\Gamma_j(z^{-1})$	Übertragungsfunktion für gefilterte Steuergrößen
Δ	Differenzialoperator $1 - z^{-1}$
λ_j	Gewichtungsfaktoren für Stellgrößen
μ_j	Gewichtungsfaktoren für Regelabweichungen
$\xi(t)$	Störuschen

MPC-Variablen (MIMO)

$\mathbf{0}$	Nullmatrix
\mathbf{A}	Systemmatrix
$\mathbf{A}(z^{-1})$	Nenner-Polynommatrix für CARIMA-Modell
$\mathbf{A}(z^{-1})$	modifizierte Polynommatrix; $\mathbf{A}(z^{-1}) = \Delta \mathbf{A}(z^{-1})$
\mathbf{A}_d	Systemmatrix im zeitdiskreten Modell
\mathbf{B}	Eingangsmatrix

$\mathbf{B}(z^{-1})$	Zähler-Polynommatrix für CARIMA-Modell
\mathbf{B}_d	Eingangsmatrix im zeitdiskreten Modell
\mathbf{C}	Ausgangsmatrix
$\mathbf{C}(z^{-1})$	Noise-Polynommatrix für CARIMA-Modell
\mathbf{C}_d	Ausgangsmatrix im zeitdiskreten Modell
\mathbf{D}	Durchgangsmatrix
$\mathbf{D}(z^{-1})$	Störgrößen-Polynommatrix für CARIMA-Modell
\mathbf{D}_d	Durchgangsmatrix im zeitdiskreten Modell
\mathbf{E}_d	Störgrößen-Eingangsmatrix im zeitdiskreten Modell
$\mathbf{E}_j(z^{-1})$	Hilfspolynommatrix
\mathbf{F}	Matrix der freien Antworten $\mathbf{f}(t)$
\mathbf{F}'	Matrix der modifizierten freien Antworten $\mathbf{H}\mathbf{V} + \mathbf{F}$
\mathbf{F}'	(bei Filterung) Matrix der freien Antworten $\mathbf{f}'(t)$
\mathbf{F}''	Matrix der modifizierten freien Antw. $\mathbf{H}\mathbf{V} + \mathbf{F}'$ b. Filt.
$\mathbf{F}_{N_{12}}$	Matrix \mathbf{F} unter Berücksichtigung des Regelhorizonts
$\mathbf{F}'_{N_{12}}$	Matrix \mathbf{F}' unter Berücksichtigung des Regelhorizonts
$\mathbf{F}''_{N_{12}}$	Matrix \mathbf{F}'' unter Berücksichtigung des Regelhorizonts
$\mathbf{F}(z^{-1})$	Übertragungs-Polynommatrix für vergangene Regelgrößen
\mathbf{F}_d	Störgrößen-Durchgangsmatrix im zeitdiskreten Modell
$\mathbf{F}_j(z^{-1})$	Hilfspolynommatrix
$\mathbf{F}\mathbf{G}_p$	Hilfsmatrix zur Berechnung der freien Antwort
$\mathbf{F}\mathbf{G}_p_{N_{12}}$	Matrix $\mathbf{F}\mathbf{G}_p$ unter Berücksichtigung des Regelhorizonts
$\mathbf{F}\mathbf{G}_p\mathbf{H}_p$	Hilfsmatrix für freie Antwort mit Störeinflüssen
$\mathbf{F}\mathbf{G}_p\mathbf{H}_p_{N_{12}}$	Matrix $\mathbf{F}\mathbf{G}_p\mathbf{H}_p$ unter Berücksichtigung des Regelhorizonts
$\mathbf{F}\Gamma\Theta$	wie $\mathbf{F}\mathbf{G}_p\mathbf{H}_p$, jedoch bei Filterung
$\mathbf{F}\Gamma\Theta_{N_{12}}$	Matrix $\mathbf{F}\Gamma\Theta$ unter Berücksichtigung des Regelhorizonts
$\mathbf{f}(t)$	freie Antworten
$\mathbf{f}'(t)$	freie Antworten bei Filterung
\mathbf{f}_j	Vektor der freien Antwort
\mathbf{G}	Übertragungsmatrix für die erzwungene Antwort
\mathbf{G}'	Übertragungsmatrix für erzwungene Antwort bei Filterung
$\mathbf{G}_{N_{12u}}$	Matrix \mathbf{G} unter Berücksichtigung des Regelhorizonts
$\mathbf{G}'_{N_{12u}}$	Matrix \mathbf{G}' unter Berücksichtigung des Regelhorizonts
$\mathbf{G}_d(z^{-1})$	zeitdiskrete Übertragungsmatrix (Matrizenpolynom)
$\mathbf{G}_j(z^{-1})$	Übertragungs-Polynommatrix für die erzwungene Antwort
$\mathbf{G}'_j(z^{-1})$	wie $\mathbf{G}_j(z^{-1})$, jedoch bei Filterung
$\mathbf{G}_{jp}(z^{-1})$	Übertragungs-Polynommatrix für vergangene Steuergrößen
$\mathbf{G}_p(z^{-1})$	Übertragungs-Polynommatrix für vergangene Steuergrößen
\mathbf{H}	Übertragungsmatrix für zukünftige Störeinflüsse
\mathbf{H}'	Übertragungsmatrix für zukünft. Störeinflüsse bei Filterung

$\mathbf{H}_{N_{12u}}$	Matrix \mathbf{H} unter Berücksichtigung des Regelhorizonts
$\mathbf{H}'_{N_{12u}}$	Matrix \mathbf{H}' unter Berücksichtigung des Regelhorizonts
$\mathbf{H}_d(z^{-1})$	zeitd. Störgrößen-Übertragungsmatrix (Matrizenpolynom)
$\mathbf{H}_j(z^{-1})$	Übertragungs-Polynommatrix für zukünftige Störeinflüsse
$\mathbf{H}'_j(z^{-1})$	wie $\mathbf{H}_j(z^{-1})$, jedoch bei Filterung
$\mathbf{H}_{jp}(z^{-1})$	Übertragungs-Polynommatrix für vergangene Störeinflüsse
$\mathbf{H}_p(z^{-1})$	Übertragungs-Polynommatrix für vergangene Störeinflüsse
$\mathbf{HFG}_p\mathbf{H}_p$	Hilfsmatr. zur Berechnung der modifizierten freien Antwort
$\mathbf{H}'\mathbf{F}\mathbf{\Gamma}\mathbf{\Theta}$	wie $\mathbf{HFG}_p\mathbf{H}_p$, jedoch bei Filterung
\mathbf{I}	Einheitsmatrix
i	Zählvariable
J	Bewertungskriterium
j	Zählvariable
\mathbf{K}	Reglermatrix (LQR)
k	Zählvariable
l	Anzahl der Eingänge der Störeinflüsse
m	Anzahl der Systemeingänge
N	Horizont, allgemein
N_1	unterer Kostenhorizont
N_2	oberer Kostenhorizont
N_p	Prädiktionshorizont
N_u	Regelhorizont
n	Anzahl der Systemausgänge
na	Grad des Nenner-Matrizenpolynoms $\mathbf{A}(z^{-1})$
nb	Grad des Zähler-Matrizenpolynoms $\mathbf{B}(z^{-1})$
nc	Grad des Noise-Matrizenpolynoms $\mathbf{C}(z^{-1})$
nd	Grad des Störgrößen-Matrizenpolynoms $\mathbf{D}(z^{-1})$
nt	Grad des Design-Matrizenpolynoms $\mathbf{T}(z^{-1})$
\mathbf{P}	Gewichtungsmatrix (LQR)
$\mathbf{P}(z^{-1})$	beliebige Polynommatrix
\mathbf{Q}	Gewichtungsmatrix für Zustandsgrößen
\mathbf{R}	Gewichtungsmatrix für Stellgrößen
\mathbf{R}_j	reelle Hilfsmatrix zur Berechnung von $\mathbf{E}_j(z^{-1})$
$\mathbf{R}(z^{-1})$	Hilfpolynommatrix zur Berechnung von $\mathbf{E}_j(z^{-1})$
\mathbf{S}	Gewichtungsmatrix für Endwert der Zustandsgrößen
$\mathbf{T}(z^{-1})$	Design-Polynommatrix zur Filterung
$\mathbf{U}(z)$	Z-Transformierte des Systemeingangs $\mathbf{u}(t)$
\mathbf{U}	Matrix der Stellgrößendifferenzen $\Delta\mathbf{u}(t)$
\mathbf{U}_{N_u}	Matrix \mathbf{U} unter Berücksichtigung des Regelhorizonts
$\mathbf{u}(t)$	Stellgrößen- bzw. Systemeingänge

$\mathbf{u}^f(t)$	mit $\mathbf{T}(z^{-1})$ gefilterte Stellgrößen
\mathbf{V}	Matrix der Störgrößendifferenzen $\Delta \mathbf{v}(t)$
\mathbf{VYUV}	Matrix diverser zukünftiger und vergangener Größen
\mathbf{VYUV}^f	Matrix div. zukünftiger und gefilterter, vergangener Größen
$\mathbf{v}(t)$	Störgrößen
$\mathbf{v}^f(t)$	mit $\mathbf{T}(z^{-1})$ gefilterte Störgrößen
\mathbf{W}	Matrix der zukünftigen Führungsgrößen $\mathbf{w}(t)$
$\mathbf{W}_{N_{12}}$	Matrix \mathbf{W} unter Berücksichtigung des Regelhorizonts
$\mathbf{w}(t)$	Führungsgrößen
$\mathbf{x}(t)$	Zustandsgrößen
\mathbf{Y}	Matrix der prädizierten Regelgrößen $\hat{\mathbf{y}}(t)$
$\mathbf{Y}_{N_{12}}$	Matrix \mathbf{Y} unter Berücksichtigung des Regelhorizonts
$\mathbf{Y}(z)$	Z-Transformierte des Systemausgangs $\mathbf{y}(t)$
\mathbf{YU}	Matrix vergangener Regel- und Stellgrößen $\mathbf{y}(t)$ und $\mathbf{u}(t)$
\mathbf{YUV}	Matrix vergangener Regel-, Stell- und Störgrößen
\mathbf{YUV}^f	Matrix vergangener, gefilt. Regel-, Stell- und Störgrößen
$\mathbf{y}(t)$	Regelgrößen bzw. Systemausgänge
$\mathbf{y}^f(t)$	mit $\mathbf{T}(z^{-1})$ gefilterte Regelgrößen
$\hat{\mathbf{y}}(t)$	prädizierte Regelgrößen
$\Gamma(z^{-1})$	wie $\mathbf{G}_p(z^{-1})$, jedoch bei Filterung
$\Gamma_j(z^{-1})$	wie $\mathbf{G}_{jp}(z^{-1})$, jedoch bei Filterung
Δ	Differenzialoperator $1 - z^{-1}$
$\Theta(z^{-1})$	wie $\mathbf{H}_p(z^{-1})$, jedoch bei Filterung
$\Theta_j(z^{-1})$	wie $\mathbf{H}_{jp}(z^{-1})$, jedoch bei Filterung
λ_j	Gewichtungsfaktoren für Stellgrößen
μ_j	Gewichtungsfaktoren für Regelabweichungen
$\xi(t)$	Störerauschen

DMPC-Variablen

$\mathbf{0}$	Nullmatrix
\mathbf{A}	Systemmatrix (zeitdiskret)
\mathbf{A}_d	Systemmatrix (zeitdiskret), Verzögerung berücksichtigt
\mathbf{A}_t	Systemmatrix (zeitdiskret), Tracking berücksichtigt
$\mathbf{A}_{t,d}$	Systemmatrix (zeitdiskret), Tracking und Verzögerung
$\mathbf{A}(z^{-1})$	Nenner-Polynommatrix für CARIMA-Modell (GPC)
\mathbf{a}_j	Beschreibungsvektor für Hyperebene j
\mathbf{B}	Eingangsmatrix (zeitdiskret)
\mathbf{B}_d	Eingangsmatrix (zeitdiskret), Verzögerung berücksichtigt
\mathbf{B}_t	Eingangsmatrix (zeitdiskret), Tracking berücksichtigt
$\mathbf{B}_{t,d}$	Eingangsmatrix (zeitdiskret), Tracking und Verzögerung

$\mathbf{B}(z^{-1})$	Zähler-Polynommatrix für CARIMA-Modell (GPC)
\mathbf{b}_j	Beschreibungsvektor für Hyperebene j
\mathbf{C}	Ausgangsmatrix (zeitdiskret)
\mathbf{C}_d	Ausgangsmatrix (zeitdiskret), Verzögerung berücksichtigt
\mathbf{C}_t	Ausgangsmatrix (zeitdiskret), Tracking berücksichtigt
$\mathbf{C}_{t,d}$	Ausgangsmatrix (zeitdiskret), Tracking und Verzögerung
\mathbf{D}	Durchgangsmatrix (zeitdiskret)
\mathbf{D}_d	Durchgangsmatrix (zeitdiskret), Verzögerung berücksichtigt
\mathbf{D}_t	Durchgangsmatrix (zeitdiskret), Tracking berücksichtigt
$\mathbf{D}_{t,d}$	Durchgangsmatrix (zeitdiskret), Tracking und Verzögerung
d_j	Beschreibungsfunktion für Hyperebene j
\mathbf{E}	Matrix für Randbed. modifiziertes Bewertungskriterium
\mathbf{F}	Gewichtungsmatrix modifiziertes Bewertungskriterium
\mathbf{F}	Matrix der freien Antworten $\mathbf{f}(t)$ (GPC)
\mathbf{F}	Menge der Indizes der Regelgesetze
\mathbf{F}_i	Matrix Regelgesetz innerhalb Polytop \mathcal{P}_i
F_k	affines Regelgesetz
$\mathbf{f}(t)$	freie Antworten (GPC)
\mathbf{G}	Matrix für Randbed. modifiziertes Bewertungskriterium
\mathbf{G}	Übertragungsmatrix für die erzwungene Antwort (GPC)
$\mathbf{G}(z^{-1})$	zeitd. Übertragungsmatrix (Matrizenpolynom) (GPC)
\mathbf{G}_i	Matrix Regelgesetz innerhalb Polytop \mathcal{P}_i
\mathbf{H}	Gewichtungsmatrix modifiziertes Bewertungskriterium
\mathbf{H}_i	Beschreibungsmatrix für Polytop \mathcal{P}_i in HK -Notation
\mathbf{I}	Einheitsmatrix
I	Menge der Indizes i von Polytopen \mathcal{P}_i
I_k	Menge der Indizes aller in $\mathcal{P}(\mathbf{J}_k)$ enthaltenen Polytope \mathcal{P}_i
i	Zählvariable
$iter$	Iterationsschritt bei Minimum-Time-Controller
J	Bewertungskriterium
\mathbf{J}	Menge von Indizes j von Hyperebenen
J'	modifiziertes Bewertungskriterium
J'_z	modifiziertes Bewertungskriterium für explizite Lösung
J_1	Einschritt-Bewertungskriterium für Min-Time-Controller
\mathbf{J}_k	Erg. d. in Knoten N_k bereits ausgewerteten Hyperebenen
\mathbf{J}_x	Menge der Indizes aller trennenden Hyperebenen bez. \mathcal{P}_x
j	Zählvariable
\mathbf{K}	Verstärkungsmatrix des Reglers
\mathbf{K}	Menge der noch auszuwertenden Knoten N_k
\mathbf{K}_i	Beschreibungsmatrix für Polytop \mathcal{P}_i in HK -Notation

k	Zählvariable
L	Anzahl der Hyperebenen
M	Markierung für Lage des Polytops bez. der Hyperebenen
M_x	Menge der Markierungen M des Polytops \mathcal{P}_x
m	Anzahl der Systemeingänge
\mathbb{N}	Menge der natürlichen Zahlen
N_1	unterer Kostenhorizont
N_2	oberer Kostenhorizont
N_k	Knoten k des binären Suchbaums
N_p	Prädiktionshorizont
N_u	Regelhorizont
n	Anzahl der Systemzustände
na	Grad des Nenner-Matrizenpolynoms $\mathbf{A}(z^{-1})$ (GPC)
nb	Grad des Zähler-Matrizenpolynoms $\mathbf{B}(z^{-1})$ (GPC)
nt	Grad des Design-Matrizenpolynoms $\mathbf{T}(z^{-1})$ (GPC)
P	Anzahl der Polytope der expliziten Lösung
\mathbf{P}	Gewichtungsmatrix für Endwert der Zustandsgrößen
\mathcal{P}_j	Polytop der expliziten Lösung
p	Anzahl der Systemausgänge
Q	Anzahl der Polytope der reduzierten expliziten Lösung
\mathbf{Q}	Gewichtungsmatrix für Zustandsgrößen
\mathbf{Q}_d	wie \mathbf{Q} , jedoch Verzögerung berücksichtigt
\mathbf{Q}_t	wie \mathbf{Q} , jedoch Tracking berücksichtigt
$\mathbf{Q}_{t,d}$	wie \mathbf{Q} , jedoch Tracking und Verzögerung berücksichtigt
Q_j	Polytop der reduzierten expliziten Lösung
R	Anzahl der Polytope einer Region bei Min-Time-Controller
\mathbf{R}	Gewichtungsmatrix für Stellgrößen
\mathbf{R}_d	wie \mathbf{R} , jedoch Verzögerung berücksichtigt
\mathbf{R}_t	wie \mathbf{R} , jedoch Tracking berücksichtigt
$\mathbf{R}_{t,d}$	wie \mathbf{R} , jedoch Tracking und Verzögerung berücksichtigt
\mathbf{R}_Δ	Gewichtungsmatrix für Stellgrößenänderung
\mathbb{R}	Menge der reellen Zahlen
\mathbf{R}	beliebige Menge
\mathbf{S}	Matrix für Randbedingung für explizite Lösung
\mathbf{S}	beliebige Menge
s	Dimension von \mathbf{U}
$\mathbf{T}(z^{-1})$	Design-Polynommatrix zur Filterung (GPC)
\mathcal{T}_{set}	Zielgebiet für Minimum-Time-Controller
\mathbf{U}	Vektor aus allen Stellgrößen \mathbf{u}
\mathbf{U}	Matrix der Stellgrößendifferenzen $\Delta\mathbf{u}(t)$ (GPC)

U	Menge der möglichen Werte der Stellgrößen
U_k	Teilmenge für Erweiterungsprinzip
$\mathbf{u}, \mathbf{u}(t)$	Eingangsvektor, Stellgrößen
$\mathbf{u}_d, \mathbf{u}_d(t)$	Eingangsvektor, Stellgrößen, Verzögerung berücksichtigt
$\mathbf{u}_t, \mathbf{u}_t(t)$	Eingangsvektor, Stellgrößen, Tracking berücksichtigt
$\mathbf{u}_{t,d}, \mathbf{u}_{t,d}(t)$	Eingangsvektor, Stellgrößen, Tracking und Verzögerung
W	Matrix für Randbed. modifiziertes Bewertungskriterium
W	Matrix der zukünftigen Führungsgrößen $\mathbf{w}(t)$ (GPC)
$\mathbf{w}, \mathbf{w}(t)$	Führungsgrößen
$\mathbf{w}_d, \mathbf{w}_d(t)$	Führungsgrößen, Verzögerung berücksichtigt
$\mathbf{w}_t, \mathbf{w}_t(t)$	Führungsgrößen, Tracking berücksichtigt
$\mathbf{w}_{t,d}, \mathbf{w}_{t,d}(t)$	Führungsgrößen, Tracking und Verzögerung
X	Menge der möglichen Werte der Zustandsgrößen
\mathcal{X}_{LQR}	Zentralbereich mit LQR-Regelung für Min-Time-Controller
\mathcal{X}_f^N	Feasible Region für N Abtastschritte
$\mathbf{x}, \mathbf{x}(t)$	Zustandsvektor
$\mathbf{x}_d, \mathbf{x}_d(t)$	Zustandsvektor, Verzögerung berücksichtigt
$\mathbf{x}_t, \mathbf{x}_t(t)$	Zustandsvektor, Tracking berücksichtigt
$\mathbf{x}_{t,d}, \mathbf{x}_{t,d}(t)$	Zustandsvektor, Tracking und Verzögerung
Y	Gewichtungsmatrix modifiziertes Bewertungskriterium
Y	Menge der möglichen Werte der Regelgrößen
$\mathbf{y}, \mathbf{y}(t)$	Ausgangsvektor, Regelgrößen
$\mathbf{y}_d, \mathbf{y}_d(t)$	Ausgangsvektor, Regelgrößen, Verzögerung berücksichtigt
$\mathbf{y}_t, \mathbf{y}_t(t)$	Ausgangsvektor, Regelgrößen, Tracking berücksichtigt
$\mathbf{y}_{t,d}, \mathbf{y}_{t,d}(t)$	Ausgangsvektor, Regelgrößen, Tracking und Verzögerung
\mathbf{z}	Hilfsvektor für explizite Lösung
λ	Gewichtungsfaktor für Stellgrößen

Regelungstechnik, allgemein

A	Systemmatrix
A_d	Systemmatrix im zeitdiskreten Modell
B	Eingangsmatrix
B_d	Eingangsmatrix im zeitdiskreten Modell
C	Ausgangsmatrix
C_d	Ausgangsmatrix im zeitdiskreten Modell
D	Durchgangsmatrix
D_d	Durchgangsmatrix im zeitdiskreten Modell
E	Eingangsmatrix für Störgrößen
E_d	Störgrößen-Eingangsmatrix im zeitdiskreten Modell
F	Durchgangsmatrix für Störgrößen

\mathbf{F}_d	Störgrößen-Durchgangsmatrix im zeitdiskreten Modell
$F_R(s)$	Übertragungsfunktion des Reglers
\mathbf{I}	Einheitsmatrix
k	zeitdiskreter Zeitpunkt kt
\mathbf{k}	beliebiger konstanter Vektor
t	Zeit, allgemein
T_0	Abtastzeit
\mathbf{u}	Eingangsvektor
\mathbf{v}	Eingangsvektor der Störgrößen
\mathbf{x}	Zustandsvektor
\mathbf{y}	Ausgangsvektor
τ	normierte Zeit
Φ	Übergangs-, Transitions- oder Fundamentalmatrix

Matrixalgebra

\mathbf{A}	beliebige Matrix
\mathbf{B}	Hilfsmatrix zum Zwischenspeichern von Ergebnissen
\mathbf{I}	Einheitsmatrix
\mathbf{L}	linke (untere) Dreiecksmatrix
\mathbf{P}	Permutationsmatrix
\mathbf{R}	rechte (obere) Dreiecksmatrix
\mathbf{x}, \mathbf{y}	beliebige Vektoren

Nomenclature

The notation of the individual variables, parameters etc. is based on the following convention:

Scalar values	: Italic font (a, b, c)
Complex values	: Italic bold font ($\mathbf{a}, \mathbf{b}, \mathbf{c}$)
Vectors	: Bold lowercase letters ($\mathbf{a}, \mathbf{b}, \mathbf{c}$)
Matrices	: Bold capitals ($\mathbf{A}, \mathbf{B}, \mathbf{C}$)
Sets	: Italic, bold capitals ($\mathbf{A}, \mathbf{B}, \mathbf{C}$)
Polynomials in z	: Capitals ($A(z^{-1}), B(z^{-1}), C(z^{-1})$)
Polynomial matrices in z	: Bold capitals ($\mathbf{A}(z^{-1}), \mathbf{B}(z^{-1}), \mathbf{C}(z^{-1})$)
Polytopes	: Curved capitals ($\mathcal{A}, \mathcal{B}, \mathcal{C}$)
Number ranges	: Bold capitals with double line ($\mathbb{A}, \mathbb{B}, \mathbb{C}$)
Reference values	: Star superscript ($a^*, \mathbf{a}^*, \mathbf{a}^*$)

Overview of the formula symbols:

Machine values

a	Acceleration
a	Switching state of the half bridge of phase a
b	Switching state of the half bridge of phase b
c	Switching state of the half bridge of phase c
d	Distortion factor
e	Control deviation, control error
f_s	Inverter switching frequency, non-normalized
I	Current (general)
I_R	Rated current
$I_{ph,R}$	Rated phase current
\mathbf{i}_s	Stator current
i_{sa}	Stator current in phase a
i_{sb}	Stator current in phase b
i_{sc}	Stator current in phase c
i_{sd}	Field producing component of the stator current
i_{sq}	Torque producing component of the stator current
$i_{s\alpha}$	Real part of \mathbf{i}_s in stator coordinates
$i_{s\beta}$	Imaginary part of \mathbf{i}_s in stator coordinates
k_r	Rotor coupling factor
k_s	Stator coupling factor
l_h	Mutual machine inductance
l_r	Rotor inductance

l_r'	Transient rotor inductance
$l_{r\sigma}$	Leakage inductance of the rotor
l_s	Stator inductance
l_s'	Transient stator inductance
$l_{s\sigma}$	Leakage inductance of the stator
m	Modulation index
p	Number of pole pairs
r_r	Rotor resistance
r_s	Stator resistance
r_σ	Effective resistance of both windings
S	Inverter switching state
T_0	Sampling rate, non-normalized
U_R	Nominal machine voltage
$U_{ph,R}$	Nominal phase voltage
u_d	DC link voltage
\mathbf{u}_s	Stator voltage
u_{sd}	Field producing component of the stator voltage
u_{sq}	Torque producing component of the stator voltage
$u_{s\alpha}$	Real part of \mathbf{u}_s in stator coordinates
$u_{s\beta}$	Imaginary part of \mathbf{u}_s in stator coordinates
δ	Field angle
σ	Total leakage factor
τ_0	Sampling rate
τ_f	Filter time constant for the actual values
τ_m	Mechanical time constant
τ_r	Rotor time constant
τ_r'	Transient rotor time constant
τ_s	Stator time constant
τ_s'	Transient stator time constant
τ_{tot}	Dead time
τ_σ'	Transient time constant
φ	Mechanical rotor position
ψ_r, ψ_{rd}	Rotor flux
$\psi_{r\alpha}$	Real part of ψ_r in stator coordinates
$\psi_{r\beta}$	Imaginary part of ψ_r in stator coordinates
ψ_s	Stator flux
ω	Mechanical rotor speed
ω_k	Angular speed of the coordinate system
ω_r	Angular speed of the rotor current
ω_s	Angular speed of the stator current

$\omega_{s,R}$ Nominal angular speed of the stator current

PI-controller constants

T_i Integrator time constant of the current controllers
 T_ω Integrator time constant of the speed controller
 V_i Gain factor of the current controllers
 V_ω Gain factor of the speed controller

MPC variables (SISO)

$A(z^{-1})$ Denominator polynomial of the transfer function $G(z^{-1})$
 $\tilde{A}(z^{-1})$ Modified denominator polynomial; $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$
 $B(z^{-1})$ Numerator polynomial of the transfer function $G(z^{-1})$
 $C(z^{-1})$ Noise polynomial
 d Discrete delay time
 $E_j(z^{-1})$ Auxiliary polynomial
 $F_j(z^{-1})$ Auxiliary polynomial
 $\mathbf{F}(z^{-1})$ Transfer polynomial matrix for past controlled variables
 $\mathbf{F}\mathbf{G}'$ Aux. matrix for the calculation of the free response
 $\mathbf{F}\Gamma$ Aux. matrix for the calc. of the free response with filt.
 $f(t)$ Free response
 \mathbf{f} Vector of the free response $f(t)$
 $f'(t)$ Free response with filtering
 \mathbf{f}' Vector of the free response with filtering
 \mathbf{G} Transfer matrix for the forced response
 $G(z^{-1})$ Transfer function of the system
 \mathbf{G}' Transfer matrix for the forced response with filtering
 $\mathbf{G}'(z^{-1})$ Transfer polynomial matrix for past actuating variables
 $G_j(z^{-1})$ Transfer polynomial for GPC controller
 $G_j'(z^{-1})$ Transfer function for future actuating variables
 \mathbf{g}^T First row of the matrix $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}^T$
 \mathbf{g}'^T First row of the matrix $(\mathbf{G}'^T\mathbf{G}' + \lambda\mathbf{I})^{-1} \cdot \mathbf{G}'^T$
 \mathbf{I} Identity matrix
 i Counter variable
 J Cost function
 j Counter variable
 k Counter variable
 N Horizon (general)
 N_1 Lower cost horizon
 N_2 Upper cost horizon
 N_p Prediction horizon

N_u	Control horizon
na	Degree of the denominator polynomial $A(z^{-1})$
nb	Degree of the numerator polynomial $B(z^{-1})$
nc	Degree of the noise polynomial $C(z^{-1})$
nt	Degree of the design polynomial $T(z^{-1})$
r_j	Auxiliary values for the calculation of $E_j(z^{-1})$
$\bar{R}(z^{-1})$	Auxiliary polynomial for the calculation of $E_j(z^{-1})$
$T(z^{-1})$	Design polynomial for filtering
$u(t)$	Actuating variable
$u^f(t)$	Actuating variable filtered with $T(z^{-1})$
\mathbf{u}	Vector of actuating variable differences $\Delta u(t)$
$w(t)$	Reference variable
\mathbf{w}	Vector of reference variables $w(t)$
$y(t)$	Controlled variable
$y^f(t)$	Controlled variable filtered with $T(z^{-1})$
$\hat{y}(t)$	Predicted value of the controlled variable
\mathbf{y}	Vector of the predicted controlled variables $\hat{y}(t)$
\mathbf{yu}	Vector of past controlled and actuating v. $y(t)$ and $u(t)$
\mathbf{yu}^f	Vector of filt. controlled and actuating v. $y^f(t)$ and $u^f(t)$
$\Gamma(z^{-1})$	Transfer polynomial matrix for filtered actuating variables
$\Gamma_j(z^{-1})$	Transfer function for filtered actuating variables
Δ	Differential operator $1 - z^{-1}$
λ_j	Weighting factors for actuating variables
μ_j	Weighting factors for control errors
$\xi(t)$	Noise variable

MPC variables (MIMO)

$\mathbf{0}$	Zero matrix
\mathbf{A}	State matrix
$\mathbf{A}(z^{-1})$	Denominator polynomial matrix for the CARIMA model
$\mathbf{\bar{A}}(z^{-1})$	Modified polynomial matrix; $\mathbf{\bar{A}}(z^{-1}) = \Delta \mathbf{A}(z^{-1})$
\mathbf{A}_d	State matrix in the time-discrete model
\mathbf{B}	Input matrix
$\mathbf{B}(z^{-1})$	Numerator polynomial matrix for the CARIMA model
\mathbf{B}_d	Input matrix in the time-discrete model
\mathbf{C}	Output matrix
$\mathbf{C}(z^{-1})$	Noise polynomial matrix for the CARIMA model
\mathbf{C}_d	Output matrix in the time-discrete model
\mathbf{D}	Feedforward matrix
$\mathbf{D}(z^{-1})$	Noise polynomial matrix for the CARIMA model

\mathbf{D}_d	Feedforward matrix in the time-discrete model
\mathbf{E}_d	Noise input matrix in the time-discrete model
$\mathbf{E}_j(z^{-1})$	Auxiliary polynomial matrix
\mathbf{F}	Matrix of the free responses $\mathbf{f}(t)$
\mathbf{F}'	Matrix of the modified free responses $\mathbf{H}\mathbf{V} + \mathbf{F}$
\mathbf{F}'	Matrix of the free responses $\mathbf{f}'(t)$ (with filtering)
\mathbf{F}''	Matrix of the modified free resp. $\mathbf{H}\mathbf{V} + \mathbf{F}'$ (with filt.)
$\mathbf{F}_{N_{12}}$	Matrix \mathbf{F} under consideration of the control horizon
$\mathbf{F}'_{N_{12}}$	Matrix \mathbf{F}' under consideration of the control horizon
$\mathbf{F}''_{N_{12}}$	Matrix \mathbf{F}'' under consideration of the control horizon
$\mathbf{F}(z^{-1})$	Transfer polynomial matrix for past controlled variables
\mathbf{F}_d	Noise feedforward matrix in the time-discrete model
$\mathbf{F}_j(z^{-1})$	Auxiliary polynomial matrix
$\mathbf{F}\mathbf{G}_p$	Auxiliary matrix for the calculation of the free response
$\mathbf{F}\mathbf{G}_p$	Matrix $\mathbf{F}\mathbf{G}_p$ under consideration of the control horizon
$\mathbf{F}\mathbf{G}_p\mathbf{H}_p$	Auxiliary matrix for the free response with noise
$\mathbf{F}\mathbf{G}_p\mathbf{H}_p$	Matrix $\mathbf{F}\mathbf{G}_p\mathbf{H}_p$ under cons. of the control horizon
$\mathbf{F}\Gamma\Theta$	Like $\mathbf{F}\mathbf{G}_p\mathbf{H}_p$, but with filtering
$\mathbf{F}\Gamma\Theta_{N_{12}}$	Matrix $\mathbf{F}\Gamma\Theta$ under consideration of the control horizon
$\mathbf{f}(t)$	Free response
$\mathbf{f}'(t)$	Free response with filtering
\mathbf{f}_j	Vector of the free response
\mathbf{G}	Transfer matrix for the forced response
\mathbf{G}'	Transfer matrix for the forced response with filtering
$\mathbf{G}_{N_{12u}}$	Matrix \mathbf{G} under consideration of the control horizon
$\mathbf{G}'_{N_{12u}}$	Matrix \mathbf{G}' under consideration of the control horizon
$\mathbf{G}_d(z^{-1})$	Time-discrete transfer matrix (matrix polynomial)
$\mathbf{G}_j(z^{-1})$	Transfer polynomial matrix for the forced response
$\mathbf{G}'_j(z^{-1})$	Like $\mathbf{G}_j(z^{-1})$, but with filtering
$\mathbf{G}_{jp}(z^{-1})$	Transfer polynomial matrix for past actuating variables
$\mathbf{G}_p(z^{-1})$	Transfer polynomial matrix for past actuating variables
\mathbf{H}	Transfer matrix for future disturbances
\mathbf{H}'	Like \mathbf{H} , but with filtering
$\mathbf{H}_{N_{12u}}$	Matrix \mathbf{H} under consideration of the control horizon
$\mathbf{H}'_{N_{12u}}$	Matrix \mathbf{H}' under consideration of the control horizon
$\mathbf{H}_d(z^{-1})$	Time-discrete dist. trans. matrix (matrix polynomial)
$\mathbf{H}_j(z^{-1})$	Transfer polynomial matrix for future disturbances
$\mathbf{H}'_j(z^{-1})$	Like $\mathbf{H}_j(z^{-1})$, but with filtering
$\mathbf{H}_{jp}(z^{-1})$	Transfer polynomial matrix for past disturbances
$\mathbf{H}_p(z^{-1})$	Transfer polynomial matrix for past disturbances

$\mathbf{HFG}_p\mathbf{H}_p$	Aux. matrix for the calculation of the mod. free resp.
$\mathbf{H}'\Gamma\Theta$	Like $\mathbf{HFG}_p\mathbf{H}_p$, but with filtering
\mathbf{I}	Identity matrix
i	Counter variable
J	Cost function
j	Counter variable
\mathbf{K}	Controller matrix (LQR)
k	Counter variable
l	Number of disturbance variables
m	Number of system inputs
N	Horizon (general)
N_1	Lower cost horizon
N_2	Upper cost horizon
N_p	Prediction horizon
N_u	Control horizon
n	Number of system outputs
na	Degree of the denominator matrix polynomial $\mathbf{A}(z^{-1})$
nb	Degree of the numerator matrix polynomial $\mathbf{B}(z^{-1})$
nc	Degree of the noise matrix polynomial $\mathbf{C}(z^{-1})$
nd	Degree of the disturbance matrix polynomial $\mathbf{D}(z^{-1})$
nt	Degree of the design matrix polynomial $\mathbf{T}(z^{-1})$
\mathbf{P}	Weighting matrix (LQR)
$\mathbf{P}(z^{-1})$	Arbitrary polynomial matrix
\mathbf{Q}	Weighting matrix for the state variables
\mathbf{R}	Weighting matrix for the actuating variables
\mathbf{R}_j	Real auxiliary matrix for the calculation of $\mathbf{E}_j(z^{-1})$
$\mathbf{R}(z^{-1})$	Auxiliary polynomial matrix for the calculation of $\mathbf{E}_j(z^{-1})$
\mathbf{S}	Weighting matrix for the final value of the state variables
$\mathbf{T}(z^{-1})$	Design polynomial matrix for filtering
$\mathbf{U}(z)$	Z transformed system input $\mathbf{u}(t)$
\mathbf{U}	Matrix of actuating variable differences $\Delta\mathbf{u}(t)$
\mathbf{U}_{N_u}	Matrix \mathbf{U} under consideration of the control horizon
$\mathbf{u}(t)$	Actuating variables or systems inputs
$\mathbf{u}^f(t)$	Actuating variables, filtered with $\mathbf{T}(z^{-1})$
\mathbf{V}	Matrix of disturbance variable differences $\Delta\mathbf{v}(t)$
\mathbf{VYUV}	Matrix of misc. future and past variable values
\mathbf{VYUV}^f	Matrix of misc. future and past values with filter
$\mathbf{v}(t)$	Disturbance variables
$\mathbf{v}^f(t)$	Disturbance variables filtered with $\mathbf{T}(z^{-1})$
\mathbf{W}	Matrix of future reference variables $\mathbf{w}(t)$

$\mathbf{W}_{N_{12}}$	Matrix \mathbf{W} under consideration of the control horizon
$\mathbf{w}(t)$	Reference variables
$\mathbf{x}(t)$	State variables
\mathbf{Y}	Matrix of the predicted controlled variables $\hat{\mathbf{y}}(t)$
$\mathbf{Y}_{N_{12}}$	Matrix \mathbf{Y} under consideration of the control horizon
$\mathbf{Y}(z)$	Z transformation of system output variables $\mathbf{y}(t)$
\mathbf{YU}	Matrix of past controlled and actuating v. $\mathbf{y}(t)$ and $\mathbf{u}(t)$
\mathbf{YUV}	Matrix of past controlled, actuating and dist. v.
\mathbf{YUV}^f	Matrix of past controlled, control and dist. v. with filt.
$\mathbf{y}(t)$	Controlled variables or system output variables
$\mathbf{y}^f(t)$	Controlled variables, filtered with $\mathbf{T}(z^{-1})$
$\hat{\mathbf{y}}(t)$	Predicted values of the controlled variables
$\Gamma(z^{-1})$	Like to $\mathbf{G}_p(z^{-1})$, but with filtering
$\Gamma_j(z^{-1})$	Like to $\mathbf{G}_{jp}(z^{-1})$, but with filtering
Δ	Differential operator $1 - z^{-1}$
$\Theta(z^{-1})$	Like to $\mathbf{H}_p(z^{-1})$, but with filtering
$\Theta_j(z^{-1})$	Like to $\mathbf{H}_{jp}(z^{-1})$, but with filtering
λ_j	Weighting factors for actuating variables
μ_j	Weighting factors for control errors
$\xi(t)$	Noise variable

DMPC variables

$\mathbf{0}$	Zero matrix
\mathbf{A}	State matrix (time-discrete)
\mathbf{A}_d	State matrix (time-discrete), cons. delay
\mathbf{A}_t	State matrix (time-discrete), cons. tracking
$\mathbf{A}_{t,d}$	State matrix (time-discrete), cons. tr. and delay
$\mathbf{A}(z^{-1})$	Den. polynomial matrix for CARIMA model (GPC)
\mathbf{a}_j	Describing vector for hyperplane j
\mathbf{B}	Input matrix (time-discrete)
\mathbf{B}_d	Input matrix (time-discrete), cons. delay
\mathbf{B}_t	Input matrix (time-discrete), cons. tracking
$\mathbf{B}_{t,d}$	Input matrix (time-discrete), cons. tr. and delay
$\mathbf{B}(z^{-1})$	Num. polynomial matrix for CARIMA model (GPC)
\mathbf{b}_j	Describing vector for hyperplane j
\mathbf{C}	Output matrix (time-discrete)
\mathbf{C}_d	Output matrix (time-discrete), cons. delay
\mathbf{C}_t	Output matrix (time-discrete), cons. tracking
$\mathbf{C}_{t,d}$	Output matrix (time-discrete), cons. tr. and delay
\mathbf{D}	Feedforward matrix (time-discrete)

\mathbf{D}_d	Feedforward matrix (time-discrete), cons. delay
\mathbf{D}_t	Feedforward matrix (time-discrete), cons. tracking
$\mathbf{D}_{t,d}$	Feedforward matrix (time-discrete), cons. tr. and delay
d_j	Describing function for hyperplane j
\mathbf{E}	Matrix of constraints modified cost function
\mathbf{F}	Weighting matrix modified cost function
\mathbf{F}	Matrix of free responses $\mathbf{f}(t)$ (GPC)
\mathbf{F}	Set of indices of the control laws
\mathbf{F}_i	Matrix control law within polytope \mathcal{P}_i
F_k	Affine control law
$\mathbf{f}(t)$	Free responses (GPC)
\mathbf{G}	Matrix of constraints modified cost function
\mathbf{G}	Transfer matrix for forced response (GPC)
$\mathbf{G}(z^{-1})$	Time-discrete transfer matrix (matrix polyn.) (GPC)
\mathbf{G}_i	Matrix control law within polytope \mathcal{P}_i
\mathbf{H}	Weighting matrix modified cost function
\mathbf{H}_i	Describing matrix for polytope \mathcal{P}_i in HK notation
\mathbf{I}	Identity matrix
I	Set of indices i of polytopes \mathcal{P}_i
I_k	Set of indices of all polytopes \mathcal{P}_i being in $\mathcal{P}(\mathbf{J}_k)$
i	Counter variable
$iter$	Iteration step with minimum-time controller
J	Cost function
\mathbf{J}	Set of indices j of hyperplanes
J'	Modified cost function
J'_z	Modified cost function for explicit solution
J_1	Single-step cost function for minimum-time controller
\mathbf{J}_k	Result of the hyperplanes already evaluated in node N_k
\mathbf{J}_x	Set of indices of all separating hyperplanes rel. to \mathcal{P}_x
j	Counter variable
\mathbf{K}	Controller gain matrix
\mathbf{K}	Set of nodes N_k to be evaluated
\mathbf{K}_i	Describing matrix for the polytope \mathcal{P}_i in HK notation
k	Counter variable
L	Number of hyperplanes
M	Marking for position of polytope rel. to the hyperpl.
M_x	Set of markings M of the polytope \mathcal{P}_x
m	Number of system inputs
\mathbb{N}	Set of natural numbers
N_1	Lower cost horizon

N_2	Upper cost horizon
N_k	Node k of the binary search tree
N_p	Prediction horizon
N_u	Control horizon
n	Number of the system states
na	Degree of the den. matrix polynomial $\mathbf{A}(z^{-1})$ (GPC)
nb	Degree of the num. matrix polynomial $\mathbf{B}(z^{-1})$ (GPC)
nt	Degree of the design matrix polynomial $\mathbf{T}(z^{-1})$ (GPC)
P	Number of polytopes of the explicit solution
\mathbf{P}	Weight. matrix for the final value of the state variables
\mathcal{P}_j	Polytope of the explicit solution
p	Number of system outputs
Q	Number of polytopes of the reduced explicit solution
\mathbf{Q}	Weighting matrix for the state variables
\mathbf{Q}_d	Like \mathbf{Q} , but under consideration of delay
\mathbf{Q}_t	Like \mathbf{Q} , but under consideration of tracking
$\mathbf{Q}_{t,d}$	Like \mathbf{Q} , but under consideration of tr. and delay
\mathcal{Q}_j	Polytope of the reduced explicit solution
R	Number of polytopes in a region with min-time contr.
\mathbf{R}	Weighting matrix for the actuating variables
\mathbf{R}_d	Like \mathbf{R} , but under consideration of delay
\mathbf{R}_t	Like \mathbf{R} , but under consideration of tracking
$\mathbf{R}_{t,d}$	Like \mathbf{R} , but under consideration of tracking and delay
\mathbf{R}_Δ	Weighting matrix for changes of the actuating variables
\mathbb{R}	Set of real numbers
\mathbf{R}	Arbitrary set
\mathbf{S}	Matrix for constraints for the explicit solution
\mathbf{S}	Arbitrary set
s	Dimension of \mathbf{U}
$\mathbf{T}(z^{-1})$	Design polynomial matrix with filter (GPC)
\mathcal{T}_{set}	Target area for minimum-time controller
\mathbf{U}	Vector of all actuating variables \mathbf{u}
\mathbf{U}	Matrix of actuating variable differences $\Delta\mathbf{u}(t)$ (GPC)
U	Set of possible values for the actuating variables
U_k	Subset for the expansion principle
$\mathbf{u}, \mathbf{u}(t)$	Input vector, actuating variables
$\mathbf{u}_d, \mathbf{u}_d(t)$	Input vector, actuating variables, cons. delay
$\mathbf{u}_t, \mathbf{u}_t(t)$	Input vector, actuating variables, cons. tracking
$\mathbf{u}_{t,d}, \mathbf{u}_{t,d}(t)$	Input vector, actuating variables, cons. tr. and delay
\mathbf{W}	Matrix for constraints modified cost function

W	Matrix of future reference variables $\mathbf{w}(t)$ (GPC)
$\mathbf{w}, \mathbf{w}(t)$	Reference variables
$\mathbf{w}_d, \mathbf{w}_d(t)$	Reference variables, cons. of delay
$\mathbf{w}_t, \mathbf{w}_t(t)$	Reference variables, cons. of tracking
$\mathbf{w}_{t,d}, \mathbf{w}_{t,d}(t)$	Reference variables, cons. of tracking and delay
X	Set of possible values for the state variables
\mathcal{X}_{LQR}	Central region with LQR control for min-time controllers
\mathcal{X}_f^N	Feasible region for N sampling cycles
$\mathbf{x}, \mathbf{x}(t)$	State vector
$\mathbf{x}_d, \mathbf{x}_d(t)$	State vector, cons. of delay
$\mathbf{x}_t, \mathbf{x}_t(t)$	State vector, cons. of tracking
$\mathbf{x}_{t,d}, \mathbf{x}_{t,d}(t)$	State vector, cons. of tracking and delay
Y	Weighting matrix modified cost function
Y	Set of possible values for the controlled variables
$\mathbf{y}, \mathbf{y}(t)$	Output vector, controlled variables
$\mathbf{y}_d, \mathbf{y}_d(t)$	Output vector, controlled variables, cons. of delay
$\mathbf{y}_t, \mathbf{y}_t(t)$	Output vector, controlled variables, cons. of tracking
$\mathbf{y}_{t,d}, \mathbf{y}_{t,d}(t)$	Source vector, controlled variables, cons. of tr. and delay
z	Auxiliary vector for the explicit solution
λ	Weighting factor for the actuating variables

Control Engineering (general)

A	State matrix
A_d	State matrix in a time-discrete model
B	Input matrix
B_d	Input matrix in a time-discrete model
C	Output matrix
C_d	Output matrix in a time-discrete model
D	Feedforward matrix
D_d	Feedforward matrix in a time-discrete model
E	Input matrix for disturbance variables
E_d	Disturbance input matrix in a time-discrete model
F	Feedforward matrix for disturbance variables
F_d	Disturbance feedf. matrix in a time-discrete model
$F_R(s)$	Controller transfer function
I	Identity matrix
k	Time-discrete point in time kt
k	Arbitrary constant vector
t	Time (general)
T_0	Sampling time

u	Input vector
v	Input vector of the disturbance variables
x	State vector
y	Output vector
τ	Normalized time
Φ	Transition or fundamental matrix

Matrix algebra

A	Arbitrary matrix
B	Auxiliary matrix for intermediate results
I	Identity matrix
L	Left (lower) triangular matrix
P	Permutation matrix
R	Right (upper) triangular matrix
x, y	Arbitrary vectors

术语表

各个变量、参数等表示方式遵循以下约定：

标量	: 斜体 (a, b, c)
复数	: 加粗斜体 ($\mathbf{a}, \mathbf{b}, \mathbf{c}$)
向量	: 粗体小写字母 ($\mathbf{a}, \mathbf{b}, \mathbf{c}$)
矩阵	: 粗体大写字母 ($\mathbf{A}, \mathbf{B}, \mathbf{C}$)
集合	: 斜体的粗体大写字母 ($\mathbf{A}, \mathbf{B}, \mathbf{C}$)
变量 z 的多项式	: 大写字母 ($A(z^{-1}), B(z^{-1}), C(z^{-1})$)
变量 z 的多项式矩阵	: 粗体大写字母 ($\mathbf{A}(z^{-1}), \mathbf{B}(z^{-1}), \mathbf{C}(z^{-1})$)
多胞体	: 弧形大写字母 ($\mathcal{A}, \mathcal{B}, \mathcal{C}$)
数域	: 双线式大写字母 ($\mathbb{A}, \mathbb{B}, \mathbb{C}$)
参考值	: 星形上标 ($a^*, \mathbf{a}^*, \mathbf{a}^*$)

所使用的符号概览：

电机量

a	加速度
a	a 相半桥开关状态
b	b 相半桥开关状态
c	c 相半桥开关状态
d	畸变因数
e	控制偏差, 控制误差
f_s	变流器开关频率, 非标么化的
I	电流, 一般情况
I_R	额定电流
$I_{ph,R}$	额定相电流
i_s	定子电流
i_{sa}	a 相定子电流
i_{sb}	b 相定子电流
i_{sc}	c 相定子电流
i_{sd}	定子电流励磁分量
i_{sq}	定子电流转矩分量
$i_{s\alpha}$	定子坐标系下 i_s 的实部
$i_{s\beta}$	定子坐标系下 i_s 的虚部
k_r	转子耦合系数
k_s	定子耦合系数
l_h	电机主电感
l_r	转子电感
l_r'	暂态转子电感

$l_{r\sigma}$	转子漏感
l_s	定子电感
l'_s	暂态定子电感
$l_{s\sigma}$	定子漏感
m	调制比
p	极对数
r_r	转子电阻
r_s	定子电阻
r_σ	双绕组的有效电阻
S	变流器开关状态
T_0	采样频率, 非标么化的
U_R	额定机电电压
$U_{ph,R}$	额定相电压
u_d	中间直流电压
u_s	定子电压
u_{sd}	定子电压励磁分量
u_{sq}	定子电压转矩分量
$u_{s\alpha}$	定子坐标系下 u_s 的实部
$u_{s\beta}$	定子坐标系下 u_s 的虚部
δ	磁链角
σ	整体漏磁因数
τ_0	采样频率
τ_f	实际值滤波器的时间常数
τ_m	机械时间常数
τ_r	转子时间常数
τ'_r	暂态转子时间常数
τ_s	定子时间常数
τ'_s	暂态定子时间常数
τ_{tot}	死区时间
τ'_σ	暂态时间常数
φ	转子的机械位置
ψ_r, ψ_{rd}	转子磁链
$\psi_{r\alpha}$	定子坐标系下 ψ_r 的实部
$\psi_{r\beta}$	定子坐标系下 ψ_r 的虚部
ψ_s	定子磁链
ω	转子的机械转速
ω_k	坐标系的角速度
ω_r	转子电流的角速度
ω_s	定子电流的角速度
$\omega_{s,R}$	定子电流的额定角速度

PI 控制器常数

T_i	电流控制器的积分时间常数
T_w	速度控制器的积分时间常数
V_i	电流控制器的比例作用系数
V_w	速度控制器的比例作用系数

MPC 变量 (单输入单输出)

$A(z^{-1})$	传递函数 $G(z^{-1})$ 的分母多项式
$\tilde{A}(z^{-1})$	修正的分母多项式; $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$
$B(z^{-1})$	传递函数 $G(z^{-1})$ 的分子多项式
$C(z^{-1})$	噪声多项式
d	离散的滞后时间
$E_j(z^{-1})$	辅助多项式
$F_j(z^{-1})$	辅助多项式
$\mathbf{F}(z^{-1})$	用于过去被控变量的传递多项式矩阵
\mathbf{FG}'	用于计算自由响应的辅助矩阵
$\mathbf{F}\Gamma$	用于计算包含滤波的自由响应的辅助矩阵
$f(t)$	自由响应
\mathbf{f}	自由响应 $f(t)$ 的向量
$f'(t)$	滤波后的自由响应
\mathbf{f}'	滤波后的自由响应的向量
\mathbf{G}	用于强制响应的传递矩阵
$G(z^{-1})$	系统的传递函数
\mathbf{G}'	用于包含滤波的强制响应的传递矩阵
$\mathbf{G}'(z^{-1})$	用于过去控制变量的传递多项式矩阵
$G_j(z^{-1})$	GPC 控制器的传递多项式
$G_j'(z^{-1})$	用于未来控制变量的传递函数
\mathbf{g}^T	矩阵 $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}^T$ 的首行
\mathbf{g}'^T	矩阵 $(\mathbf{G}'^T \mathbf{G}' + \lambda \mathbf{I})^{-1} \cdot \mathbf{G}'^T$ 的首行
\mathbf{I}	单位矩阵
i	计数变量
J	成本函数
j	计数变量
k	计数变量
N	步长, 一般的
N_1	下成本步长
N_2	上成本步长
N_p	预测步长
N_u	控制步长

na	分母多项式 $A(z^{-1})$ 的阶数
nb	分子多项式 $B(z^{-1})$ 的阶数
nc	噪声多项式 $C(z^{-1})$ 的阶数
nt	设计多项式 $T(z^{-1})$ 的阶数
r_j	用于计算 $E_j(z^{-1})$ 的辅助量
$\tilde{R}(z^{-1})$	用于计算 $E_j(z^{-1})$ 的辅助多项式
$T(z^{-1})$	用于滤波的设计多项式
$u(t)$	控制变量
$u^f(t)$	使用 $T(z^{-1})$ 滤波后的控制变量
\mathbf{u}	控制变量增量 $\Delta u(t)$ 的向量
$w(t)$	参考变量
\mathbf{w}	参考变量 $w(t)$ 的向量
$y(t)$	被控变量
$y^f(t)$	使用 $T(z^{-1})$ 滤波后的被控变量
$\hat{y}(t)$	预测出的被控变量
\mathbf{y}	预测出的被控变量 $\hat{y}(t)$ 的向量
\mathbf{yu}	过去的被控变量 $y(t)$ 和控制变量 $u(t)$ 组合后的向量
\mathbf{yu}^f	滤波后的过去的被控变量 $y(t)$ 和控制变量 $u(t)$ 组合后的向量
$\Gamma(z^{-1})$	用于滤波后控制变量的传递多项式矩阵
$\Gamma_j(z^{-1})$	用于滤波后控制变量的传递函数
Δ	差分算子 $1 - z^{-1}$
λ_j	用于控制变量的权重系数
μ_j	用于控制偏差的权重系数
$\xi(t)$	扰动变量

MPC 变量 (多输入多输出)

$\mathbf{0}$	零矩阵
\mathbf{A}	系统矩阵
$\mathbf{A}(z^{-1})$	CARIMA 模型的分母多项式矩阵
$\mathbf{A}(z^{-1})$	修正的多项式矩阵; $\mathbf{A}(z^{-1}) = \Delta \mathbf{A}(z^{-1})$
\mathbf{A}_d	时间离散模型的系统矩阵
\mathbf{B}	输入矩阵
$\mathbf{B}(z^{-1})$	CARIMA 模型的分子多项式矩阵
\mathbf{B}_d	时间离散模型的输入矩阵
\mathbf{C}	输出矩阵
$\mathbf{C}(z^{-1})$	CARIMA 模型的噪声多项式矩阵
\mathbf{C}_d	时间离散模型的输出矩阵
\mathbf{D}	直接输入输出矩阵
$\mathbf{D}(z^{-1})$	CARIMA 模型的扰动多项式矩阵
\mathbf{D}_d	时间离散模型的直接输入输出矩阵

E_d	时间离散模型的扰动输入矩阵
$E_j(z^{-1})$	辅助多项式矩阵
F	自由响应 $f(t)$ 的矩阵
F'	修正后的自由响应 $HV + F$ 的矩阵
F''	自由响应 $f'(t)$ 的矩阵 (包含滤波)
F'''	修正后的自由响应 $HV + F''$ 的矩阵 (包含滤波)
$F_{N_{12}}$	考虑控制步长后的矩阵 F
$F'_{N_{12}}$	考虑控制步长后的矩阵 F'
$F''_{N_{12}}$	考虑控制步长后的矩阵 F''
$F(z^{-1})$	用于过去被控变量的传递多项式矩阵
F_d	时间离散模型的扰动直接输入输出矩阵
$F_j(z^{-1})$	辅助多项式矩阵
FG_p	用于计算自由响应的辅助矩阵
$FG_{pN_{12}}$	考虑控制步长后的矩阵 FG_p
FG_pH_p	用于计算包含噪声的自由响应的辅助矩阵
$FG_pH_{pN_{12}}$	考虑控制步长后的矩阵 FG_pH_p
$F\Gamma\Theta$	类似 FG_pH_p , 但是含有滤波
$F\Gamma\Theta_{N_{12}}$	考虑控制步长后的矩阵 $F\Gamma\Theta$
$f(t)$	自由响应
$f'(t)$	滤波后的自由响应
f_j	自由响应的向量
G	用于强制响应的传递矩阵
G'	用于滤波后的强制响应的传递矩阵
$G_{N_{12u}}$	考虑控制步长后的矩阵 G
$G'_{N_{12u}}$	考虑控制步长后的矩阵 G'
$G_d(z^{-1})$	时间离散传递矩阵 (矩阵多项式)
$G_j(z^{-1})$	用于强制响应的传递多项式矩阵
$G'_j(z^{-1})$	类似 $G_j(z^{-1})$, 但是含有滤波
$G_{jp}(z^{-1})$	用于过去控制变量的传递多项式矩阵
$G_p(z^{-1})$	用于过去控制变量的传递多项式矩阵
H	用于未来扰动因素的传递矩阵
H'	类似 H , 但是含有滤波
$H_{N_{12u}}$	考虑控制步长后的矩阵 H
$H'_{N_{12u}}$	考虑控制步长后的矩阵 H'
$H_d(z^{-1})$	时间离散扰动传递矩阵 (矩阵多项式)
$H_j(z^{-1})$	用于未来扰动因素的传递多项式矩阵
$H'_j(z^{-1})$	类似 $H_j(z^{-1})$, 但是含有滤波
$H_{jp}(z^{-1})$	用于过去扰动因素的传递多项式矩阵
$H_p(z^{-1})$	用于过去扰动因素的传递多项式矩阵
HFG_pH_p	用于计算修正自由响应的辅助矩阵

$\mathbf{H}'\Gamma\Theta$	类似 $\mathbf{HFG}_p\mathbf{H}_p$, 但是含有滤波
\mathbf{I}	单位矩阵
i	计数变量
J	成本函数
j	计数变量
\mathbf{K}	控制器矩阵 (LQR)
k	计数变量
l	扰动变量的数量
m	系统输入的数量
N	步长, 一般的
N_1	下成本步长
N_2	上成本步长
N_p	预测步长
N_u	控制步长
n	系统输出的数量
na	分母矩阵多项式 $\mathbf{A}(z^{-1})$ 的阶数
nb	分子矩阵多项式 $\mathbf{B}(z^{-1})$ 的阶数
nc	噪声矩阵多项式 $\mathbf{C}(z^{-1})$ 的阶数
nd	扰动矩阵多项式 $\mathbf{D}(z^{-1})$ 的阶数
nt	设计矩阵多项式 $\mathbf{T}(z^{-1})$ 的阶数
\mathbf{P}	权重矩阵 (LQR)
$\mathbf{P}(z^{-1})$	任意的多项式矩阵
\mathbf{Q}	用于状态变量的权重矩阵
\mathbf{R}	用于控制变量的权重矩阵
\mathbf{R}_j	用于计算 $\mathbf{E}_j(z^{-1})$ 的实数辅助矩阵
$\mathbf{R}(z^{-1})$	用于计算 $\mathbf{E}_j(z^{-1})$ 的辅助多项式矩阵
\mathbf{S}	用于状态变量终值的权重矩阵
$\mathbf{T}(z^{-1})$	用于滤波的设计多项式
$\mathbf{U}(z)$	系统输入 $\mathbf{u}(t)$ 的 Z 变换
\mathbf{U}	控制增量矩阵 $\Delta\mathbf{u}(t)$
\mathbf{U}_{N_u}	考虑控制步长后的矩阵 \mathbf{U}
$\mathbf{u}(t)$	控制变量或系统输入
$\mathbf{u}^f(t)$	使用 $\mathbf{T}(z^{-1})$ 滤波后的控制变量
\mathbf{V}	扰动增量 $\Delta\mathbf{v}(t)$ 的矩阵
\mathbf{VYUV}	由不同的未来和过去变量组成的矩阵
\mathbf{VYUV}^f	由不同的未来和经滤波后的过去变量组成的矩阵
$\mathbf{v}(t)$	扰动变量
$\mathbf{v}^f(t)$	使用 $\mathbf{T}(z^{-1})$ 滤波后的扰动变量
\mathbf{W}	未来参考变量 $\mathbf{w}(t)$ 的矩阵
$\mathbf{W}_{N_{12}}$	考虑控制步长后的矩阵 \mathbf{W}

$w(t)$	参考变量
$x(t)$	状态变量
Y	预测出的被控变量 $\hat{y}(t)$ 的矩阵
$Y_{N_{12}}$	考虑控制步长后的矩阵 Y
$Y(z)$	系统输出 $y(t)$ 的 Z 变换
YU	由过去的被控变量 $y(t)$ 和控制变量 $u(t)$ 组成的矩阵
YUV	由过去的被控变量、控制变量和扰动变量组成的矩阵
YUV^f	由过去的, 滤波后的被控变量、控制变量和扰动变量组成的矩阵
$y(t)$	被控变量或系统输出
$y^f(t)$	使用 $T(z^{-1})$ 滤波后的被控变量
$\hat{y}(t)$	预测出的被控变量
$\Gamma(z^{-1})$	类似 $G_p(z^{-1})$, 但是含有滤波
$\Gamma_j(z^{-1})$	类似 $G_{jp}(z^{-1})$, 但是含有滤波
Δ	微分算子 $1 - z^{-1}$
$\Theta(z^{-1})$	类似 $H_p(z^{-1})$, 但是含有滤波
$\Theta_j(z^{-1})$	类似 $H_{jp}(z^{-1})$, 但是含有滤波
λ_j	用于控制变量的权重系数
μ_j	用于控制偏差的权重系数
$\xi(t)$	扰动变量

DMPC 变量

0	零矩阵
A	系统矩阵 (时间离散)
A_d	系统矩阵 (时间离散), 考虑滞后
A_t	系统矩阵 (时间离散), 考虑跟踪
$A_{t,d}$	系统矩阵 (时间离散), 综合考虑滞后和跟踪
$A(z^{-1})$	CARIMA 模型的分母多项式矩阵 (GPC)
a_j	超平面 j 的描述向量
B	输入矩阵 (时间离散)
B_d	输入矩阵 (时间离散), 考虑滞后
B_t	输入矩阵 (时间离散), 考虑跟踪
$B_{t,d}$	输入矩阵 (时间离散), 综合考虑滞后和跟踪
$B(z^{-1})$	CARIMA 模型的分子多项式矩阵 (GPC)
b_j	超平面 j 的描述向量
C	输出矩阵 (时间离散)
C_d	输出矩阵 (时间离散), 考虑滞后
C_t	输出矩阵 (时间离散), 考虑跟踪
$C_{t,d}$	输出矩阵 (时间离散), 综合考虑滞后和跟踪
D	直接输入输出矩阵 (时间离散)
D_d	直接输入输出矩阵 (时间离散), 考虑滞后

D_t	直接输入输出矩阵 (时间离散), 考虑跟踪
$D_{t,d}$	直接输入输出矩阵 (时间离散), 综合考虑滞后和跟踪
d_j	超平面 j 的描述函数
E	用于修正评判准则后约束条件的矩阵
F	修正评判准则后的权重矩阵
F	自由响应 $f(t)$ 的矩阵 (GPC)
F	控制律的指数集合
F_i	多胞体 \mathcal{P}_i 内的控制律矩阵
F_k	仿射控制律
$f(t)$	自由响应 (GPC)
G	用于修正成本函数后约束条件的矩阵
G	用于强制响应的传递矩阵 (GPC)
$G(z^{-1})$	时间离散传递矩阵 (矩阵多项式) (GPC)
G_i	多胞体 \mathcal{P}_i 内的控制律矩阵
H	修正评判准则后的权重矩阵
H_i	在 HK 符号下的用于多胞体 \mathcal{P}_i 的描述矩阵
I	单位矩阵
I	多胞体 \mathcal{P}_i 的指数 i 的集合
I_k	所有位于 $\mathcal{P}(J_k)$ 内的多胞体 \mathcal{P}_i 的指数集合
i	计数变量
$iter$	最小时间控制器下的递归步数
J	成本函数
J	超平面 j 的指数集合
J'	修正后的传递函数
J'_z	用于显示求解的修正传递函数
J_1	用于最小时间控制器的单步传递函数
J_k	在已评判超平面内结点 N_k 的结果
J_x	与 \mathcal{P}_x 相关的所有分离超平面的指数集合
j	计数变量
K	控制器增益矩阵
K	待评判结点 N_k 的集合
K_i	在 HK 符号下的用于多胞体 \mathcal{P}_i 的描述矩阵
k	计数变量
L	超平面的数量
M	与超平面相关的多胞体位置的标记
M_x	多胞体 \mathcal{P}_x 的标记 M 的集合
m	系统输入的数量
\mathbb{N}	自然数
N_1	下成本步长
N_2	上成本步长

N_k	二叉搜索树的结点 k
N_p	预测步长
N_u	控制步长
n	系统状态的数量
na	分母矩阵多项式 $\mathbf{A}(z^{-1})$ 的阶数 (GPC)
nb	分子矩阵多项式 $\mathbf{B}(z^{-1})$ 的阶数 (GPC)
nt	设计矩阵多项式 $\mathbf{T}(z^{-1})$ 的阶数 (GPC)
P	显式求解下的多胞体的数量
\mathbf{P}	用于状态变量终值的权重矩阵
\mathcal{P}_j	显式求解下的多胞体
p	系统输出的数量
Q	降阶的显式求解下的多胞体的数量
\mathbf{Q}	用于状态变量的权重矩阵
\mathbf{Q}_d	用于状态变量的权重矩阵, 考虑滞后
\mathbf{Q}_t	用于状态变量的权重矩阵, 考虑跟踪
$\mathbf{Q}_{t,d}$	用于状态变量的权重矩阵, 综合考虑滞后和跟踪
\mathcal{Q}_j	降阶的显式求解下的多胞体
R	最小时间控制器下的一个区域内的多胞体数量
\mathbf{R}	用于控制变量的权重矩阵
\mathbf{R}_d	用于控制变量的权重矩阵, 考虑滞后
\mathbf{R}_t	用于控制变量的权重矩阵, 考虑跟踪
$\mathbf{R}_{t,d}$	用于控制变量的权重矩阵, 综合考虑滞后和跟踪
\mathbf{R}_Δ	用于控制增量的权重矩阵
\mathbb{R}	实数集
\mathbf{R}	任意的集合
\mathbf{S}	用于显式求解中的约束条件的矩阵
\mathbf{S}	任意的集合
s	\mathbf{U} 的维度
$\mathbf{T}(z^{-1})$	用于滤波的设计多项式矩阵 (GPC)
\mathcal{T}_{set}	最小时间控制器下的目标区域
\mathbf{U}	所有控制变量 \mathbf{u} 组成的向量
\mathbf{U}	控制增量 $\Delta\mathbf{u}(t)$ 的矩阵 (GPC)
\mathbf{U}	控制变量可能数值的集合
U_k	扩展原则下的子集
$\mathbf{u}, \mathbf{u}(t)$	输入向量, 控制变量
$\mathbf{u}_d, \mathbf{u}_d(t)$	输入向量, 控制变量, 考虑滞后
$\mathbf{u}_t, \mathbf{u}_t(t)$	输入向量, 控制变量, 考虑跟踪
$\mathbf{u}_{t,d}, \mathbf{u}_{t,d}(t)$	输入向量, 控制变量, 综合考虑滞后和跟踪
\mathbf{W}	用于修正成本函数后约束条件的矩阵
\mathbf{W}	未来的参考变量 $\mathbf{w}(t)$ 矩阵 (GPC)

$\mathbf{w}, \mathbf{w}(t)$	参考变量
$\mathbf{w}_d, \mathbf{w}_d(t)$	参考变量, 考虑滞后
$\mathbf{w}_t, \mathbf{w}_t(t)$	参考变量, 考虑跟踪
$\mathbf{w}_{t,d}, \mathbf{w}_{t,d}(t)$	参考变量, 综合考虑滞后和跟踪
\mathbf{X}	状态变量的可能值的集合
\mathcal{X}_{LQR}	最小时间控制器下使用 LQR 控制的中央区域
\mathcal{X}_f^N	N 个采样周期下的可行域
$\mathbf{x}, \mathbf{x}(t)$	状态向量
$\mathbf{x}_d, \mathbf{x}_d(t)$	状态向量, 考虑滞后
$\mathbf{x}_t, \mathbf{x}_t(t)$	状态向量, 考虑跟踪
$\mathbf{x}_{t,d}, \mathbf{x}_{t,d}(t)$	状态向量, 考虑跟踪
\mathbf{Y}	用于修正成本函数后的权重矩阵
\mathbf{Y}	被控变量的可能值的集合
$\mathbf{y}, \mathbf{y}(t)$	输出向量, 被控变量
$\mathbf{y}_d, \mathbf{y}_d(t)$	输出向量, 被控变量, 考虑滞后
$\mathbf{y}_t, \mathbf{y}_t(t)$	输出向量, 被控变量, 考虑跟踪
$\mathbf{y}_{t,d}, \mathbf{y}_{t,d}(t)$	输出向量, 被控变量, 考虑跟踪
\mathbf{z}	显式求解下的辅助向量
λ	用于控制变量的权重系数

控制技术, 一般的

\mathbf{A}	系统矩阵
\mathbf{A}_d	时间离散模型下的系统矩阵
\mathbf{B}	输入矩阵
\mathbf{B}_d	时间离散模型下的输入矩阵
\mathbf{C}	输出矩阵
\mathbf{C}_d	时间离散模型下的输出矩阵
\mathbf{D}	直接输入输出矩阵
\mathbf{D}_d	时间离散模型下的直接输入输出矩阵
\mathbf{E}	用于扰动变量的输入矩阵
\mathbf{E}_d	时间离散模型下的扰动输入矩阵
\mathbf{F}	用于扰动变量的直接输入输出矩阵
\mathbf{F}_d	时间离散模型下的扰动直接输入输出矩阵
$F_R(s)$	控制器的传递函数
\mathbf{I}	单位矩阵
k	时间离散时刻 kt
\mathbf{k}	任意的恒定向量
t	时间, 一般的
T_0	采样时间
\mathbf{u}	输入向量

v	扰动输入向量
x	状态向量
y	输出向量
τ	标么化后的时间
Φ	传递矩阵或基本矩阵

矩阵代数

A	任意矩阵
B	用于结果中间存储的辅助矩阵
I	单位矩阵
L	左(下)三角矩阵
P	置换矩阵
R	右(上)三角矩阵
x, y	任意向量

Anhang/Appendix/附录 C

Normierungsgrößen/Normalization Values/标么量

Zur Normierung werden die Strang-Nenngrößen verwendet.

$$\begin{array}{ll} \text{bei Sternschaltung} & U_{Str,R} = \frac{1}{\sqrt{3}} U_R \\ & I_{Str,R} = I_R \end{array} \quad \begin{array}{ll} \text{bei Dreieckschaltung} & U_{Str,R} = U_R \\ & I_{Str,R} = \frac{1}{\sqrt{3}} I_R \end{array}$$

Dadurch ergeben sich folgende Normierungsgrößen (Zeit = $\frac{1}{\omega_{s,R}}$):

Spannung	$\sqrt{2} \cdot U_{Str,R}$	Flussverkettung	$\frac{\sqrt{2} \cdot U_{Str,R}}{\omega_{s,R}}$
Strom	$\sqrt{2} \cdot I_{Str,R}$	Leistung	$3 \cdot U_{Str,R} \cdot I_{Str,R}$
Impedanz	$\frac{U_{Str,R}}{I_{Str,R}}$	Drehmoment	$\frac{3 \cdot p \cdot U_{Str,R} \cdot I_{Str,R}}{\omega_{s,R}}$
Induktivität	$\frac{U_{Str,R}}{\omega_{s,R} \cdot I_{Str,R}}$	Drehfrequenz	$\frac{\omega_{s,R}}{p}$

Normalization values/标么量

The nominal phase values are used for normalization.
 额定的相值用于标么化。

Star connection/星形联接	$U_{ph,R} = \frac{1}{\sqrt{3}}U_R$ $I_{ph,R} = I_R$
----------------------	---

Y-connection/三角形联接	$U_{ph,R} = U_R$ $I_{ph,R} = \frac{1}{\sqrt{3}}I_R$
--------------------	---

Thereby, the following normalization values result (Time = $\frac{1}{\omega_{s,R}}$):
 因此, 可得到如下的标么量 (时间 = $\frac{1}{\omega_{s,R}}$):

Voltage/电压	$\sqrt{2} \cdot U_{Str,R}$	Flux linkage/磁链	$\frac{\sqrt{2} \cdot U_{Str,R}}{\omega_{s,R}}$
Current/电流	$\sqrt{2} \cdot I_{Str,R}$	Power/功率	$3 \cdot U_{Str,R} \cdot I_{Str,R}$
Impedance/阻抗	$\frac{U_{Str,R}}{I_{Str,R}}$	Torque/转矩	$\frac{3 \cdot p \cdot U_{Str,R} \cdot I_{Str,R}}{\omega_{s,R}}$
Inductance/电感	$\frac{U_{Str,R}}{\omega_{s,R} \cdot I_{Str,R}}$	Rotating speed/转速	$\frac{\omega_{s,R}}{p}$

Anhang/Appendix/附录 D

Physikalische Maschinengrößen/Physical Machine Constants/电机参数

Technische Daten der in den Experimenten zu den Kapiteln 3.4, 6.1.2, 6.3.3, 7.2, 8.4, 9.4.3 und 9.5.5 verwendeten Asynchronmaschine:

Maschinengröße	Wert
U_R	380 V
I_R	4,9 A
$\omega_{s,R}$	50 Hz
l_s	2,67
τ_r	95,2
r_s	0,0447
τ_s	59,7
$l_{s\sigma} = l_{r\sigma}$	0,0863
$l_h = l_s + l_{s\sigma}$	2,58
$r_r = \frac{l_r}{\tau_r}$	0,0280
σ	0,0663
$l_s' = l_r'$	0,177
$k_r = k_s$	0,966
τ_s'	3,96
r_σ	0,0708
τ_σ'	2,50
τ_m	550

Physical machine constants/电机参数

Technical data of the asynchronous machine used in the experiments in the chapters 3.4, 6.1.2, 6.3.3, 7.2, 8.4, 9.4.3 and 9.5.5:

在章节 3.4, 6.1.2, 6.3.3, 7.2, 8.4, 9.4.3 和 9.5.5 中的实验中所使用的异步电机的技术参数如下:

Machine constant/电机参数	Value/数值
U_R	380 V
I_R	4.9 A
$\omega_{s,R}$	50 Hz
l_s	2.67
τ_r	95.2
r_s	0.0447
τ_s	59.7
$l_{s\sigma} = l_{r\sigma}$	0.0863
$l_h = l_s + l_{s\sigma}$	2.58
$r_r = \frac{l_r}{\tau_r}$	0.0280
σ	0.0663
$l_s' = l_r'$	0.177
$k_r = k_s$	0.966
τ_s'	3.96
r_σ	0.0708
τ_σ'	2.50
τ_m	550

Anhang/Appendix/附录 E

Polynome und Matrizen bei GPC/ Polynomials and Matrices for GPC/ 用于 GPC 的多项式和矩阵

E.1 SISO-System/SISO-System/单输入单输出系统

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{nb} z^{-nb}$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_{nc} z^{-nc}$$

$$E_j(z^{-1}) = e_{j,0} + e_{j,1} z^{-1} + e_{j,2} z^{-2} + \cdots + e_{j,j-1} z^{-(j-1)} \quad j = 1 \dots N_p$$

$$= e_0 + e_1 z^{-1} + e_2 z^{-2} + \cdots + e_{j-1} z^{-(j-1)} \quad (\text{s. Kap. 6.1.1})$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1} z^{-1} + f_{j,2} z^{-2} + \cdots + f_{j,na} z^{-na} \quad j = 1 \dots N_p$$

$$\mathbf{F}(z^{-1}) = \begin{bmatrix} F_1(z^{-1}) \\ F_2(z^{-1}) \\ \vdots \\ F_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} f_{1,0} + f_{1,1} z^{-1} + \cdots + f_{1,na} z^{-na} \\ f_{2,0} + f_{2,1} z^{-1} + \cdots + f_{2,na} z^{-na} \\ \vdots \\ f_{N_p,0} + f_{N_p,1} z^{-1} + \cdots + f_{N_p,na} z^{-na} \end{bmatrix}$$

$$\mathbf{F}\mathbf{G}' = \begin{bmatrix} f_{1,0} & f_{1,1} & \cdots & f_{1,na} & g'_{1,0} & g'_{1,1} & \cdots & g'_{1,nb-1} \\ f_{2,0} & f_{2,1} & \cdots & f_{2,na} & g'_{2,0} & g'_{2,1} & \cdots & g'_{2,nb-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{N_p,0} & f_{N_p,1} & \cdots & f_{N_p,na} & g'_{N_p,0} & g'_{N_p,1} & \cdots & g'_{N_p,nb-1} \end{bmatrix}$$

$$\mathbf{F}\mathbf{\Gamma} = \begin{bmatrix} f_{1,0} & f_{1,1} & \cdots & f_{1,na} & \gamma_{1,0} & \gamma_{1,1} & \cdots & \gamma_{1,nb-1} \\ f_{2,0} & f_{2,1} & \cdots & f_{2,na} & \gamma_{2,0} & \gamma_{2,1} & \cdots & \gamma_{2,nb-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{N_p,0} & f_{N_p,1} & \cdots & f_{N_p,na} & \gamma_{N_p,0} & \gamma_{N_p,1} & \cdots & \gamma_{N_p,nb-1} \end{bmatrix}$$

$$\begin{aligned}
 \mathbf{f} &= \begin{bmatrix} f(t+1) \\ f(t+2) \\ \vdots \\ f(t+N_p) \end{bmatrix} \\
 \mathbf{f}' &= \begin{bmatrix} f'(t+1) \\ f'(t+2) \\ \vdots \\ f'(t+N_p) \end{bmatrix} \\
 \mathbf{G} &= \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_p-1} & g_{N_p-2} & \cdots & g_0 \end{bmatrix} \\
 \mathbf{G}' &= \begin{bmatrix} g'_{0} & 0 & 0 & \cdots & 0 \\ g'_{1} & g'_{0} & 0 & \cdots & 0 \\ g'_{2} & g'_{1} & g'_{0} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g'_{N_p-1} & g'_{N_p-2} & g'_{N_p-3} & \cdots & g'_{0} \end{bmatrix} \\
 \mathbf{G}'(z^{-1}) &= \begin{bmatrix} g'_{1,0} + g'_{1,1}z^{-1} + \cdots + g'_{1,nb-1}z^{-(nb-1)} \\ g'_{2,0} + g'_{2,1}z^{-1} + \cdots + g'_{2,nb-1}z^{-(nb-1)} \\ \vdots \\ g'_{N_p,0} + g'_{N_p,1}z^{-1} + \cdots + g'_{N_p,nb-1}z^{-(nb-1)} \end{bmatrix} \\
 T(z^{-1}) &= 1 + t_1z^{-1} + t_2z^{-2} + \cdots + t_{nt}z^{-nt} \\
 \mathbf{u} &= \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N_p-1) \end{bmatrix} \\
 \mathbf{w} &= \begin{bmatrix} w(t+1) \\ w(t+2) \\ \vdots \\ w(t+N_p) \end{bmatrix} \\
 \mathbf{y} &= \begin{bmatrix} \hat{y}(t+1) \\ \hat{y}(t+2) \\ \vdots \\ \hat{y}(t+N_p) \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{y}\mathbf{u} &= \begin{bmatrix} y(t) \\ y(t-1) \\ \dots \\ y(t-na) \\ \Delta u(t-1) \\ \Delta u(t-2) \\ \dots \\ \Delta u(t-nb) \end{bmatrix} \\
 \mathbf{y}\mathbf{u}^f &= \begin{bmatrix} y^f(t) \\ y^f(t-1) \\ \dots \\ y^f(t-na) \\ \Delta u^f(t-1) \\ \Delta u^f(t-2) \\ \dots \\ \Delta u^f(t-nb) \end{bmatrix} \\
 \Gamma(z^{-1}) &= \begin{bmatrix} \Gamma_1(z^{-1}) \\ \Gamma_2(z^{-1}) \\ \vdots \\ \Gamma_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \gamma_{1,0} + \gamma_{1,1}z^{-1} + \dots + \gamma_{1,nb-1}z^{-(nb-1)} \\ \gamma_{2,0} + \gamma_{2,1}z^{-1} + \dots + \gamma_{2,nb-1}z^{-(nb-1)} \\ \vdots \\ \gamma_{N_p,0} + \gamma_{N_p,1}z^{-1} + \dots + \gamma_{N_p,nb-1}z^{-(nb-1)} \end{bmatrix}
 \end{aligned}$$

E.2 MIMO-System/MIMO-System/多输入多输出系统

E.2.1 Definitionen/Definitions/定义

$$\begin{aligned}
 \mathbf{A}(z^{-1}) &= \mathbf{I} + \mathbf{A}_1 z^{-1} + \mathbf{A}_2 z^{-2} + \dots + \mathbf{A}_{na} z^{-na} \\
 \mathbf{B}(z^{-1}) &= \mathbf{B}_0 + \mathbf{B}_1 z^{-1} + \mathbf{B}_2 z^{-2} + \dots + \mathbf{B}_{nb} z^{-nb} \\
 \mathbf{C}(z^{-1}) &= \mathbf{I} + \mathbf{C}_1 z^{-1} + \mathbf{C}_2 z^{-2} + \dots + \mathbf{C}_{nc} z^{-nc} \\
 \mathbf{D}(z^{-1}) &= \mathbf{D}_0 + \mathbf{D}_1 z^{-1} + \mathbf{D}_2 z^{-2} + \dots + \mathbf{D}_{nd} z^{-nd} \\
 \mathbf{E}_j(z^{-1}) &= \mathbf{E}_{j,0} + \mathbf{E}_{j,1} z^{-1} + \mathbf{E}_{j,2} z^{-2} + \dots + \mathbf{E}_{j,j-1} z^{-(j-1)} \quad j = 1 \dots N_p \\
 &= \mathbf{E}_0 + \mathbf{E}_1 z^{-1} + \mathbf{E}_2 z^{-2} + \dots + \mathbf{E}_{j-1} z^{-(j-1)} \quad (\text{s. Kap. 8.1.3}) \\
 \mathbf{F} &= \begin{bmatrix} \mathbf{f}(t+1) \\ \mathbf{f}(t+2) \\ \vdots \\ \mathbf{f}(t+N_p) \end{bmatrix} \quad \begin{array}{l} \mathbf{F}' \text{ und } \mathbf{F}'' \text{ entsprechend} \\ \text{über } \mathbf{f}'(t) \text{ bzw. } \mathbf{f}''(t) \\ \text{definiert} \end{array}
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{F}_{N_{12}} &= \begin{bmatrix} \mathbf{f}(t + N_1) \\ \mathbf{f}(t + N_1 + 1) \\ \vdots \\ \mathbf{f}(t + N_2) \end{bmatrix} && \mathbf{F}'_{N_{12}} \quad \text{und} \quad \mathbf{F}''_{N_{12}} \\
 &&& \text{entsprechend über } \mathbf{f}(t) \\
 &&& \text{bzw. } \mathbf{f}'(t) \text{ definiert} \\
 \mathbf{F}(z^{-1}) &= \begin{bmatrix} \mathbf{F}_1(z^{-1}) \\ \mathbf{F}_2(z^{-1}) \\ \vdots \\ \mathbf{F}_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{1,0} + \mathbf{F}_{1,1}z^{-1} + \cdots + \mathbf{F}_{1,na}z^{-na} \\ \mathbf{F}_{2,0} + \mathbf{F}_{2,1}z^{-1} + \cdots + \mathbf{F}_{2,na}z^{-na} \\ \vdots \\ \mathbf{F}_{N_p,0} + \mathbf{F}_{N_p,1}z^{-1} + \cdots + \mathbf{F}_{N_p,na}z^{-na} \end{bmatrix} \\
 \mathbf{F}_j(z^{-1}) &= \mathbf{F}_{j,0} + \mathbf{F}_{j,1}z^{-1} + \mathbf{F}_{j,2}z^{-2} + \cdots + \mathbf{F}_{j,na}z^{-na} \quad j = 1 \dots N_p \\
 \mathbf{F}\mathbf{G}_p &= \begin{bmatrix} \mathbf{F}_{1,0} & \cdots & \mathbf{F}_{1,na} & \mathbf{G}_{1p,0} & \cdots & \mathbf{G}_{1p,nb-1} \\ \mathbf{F}_{2,0} & \cdots & \mathbf{F}_{2,na} & \mathbf{G}_{2p,0} & \cdots & \mathbf{G}_{2p,nb-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \cdots & \mathbf{F}_{N_p,na} & \mathbf{G}_{N_pp,0} & \cdots & \mathbf{G}_{N_pp,nb-1} \end{bmatrix} \\
 \mathbf{F}\mathbf{G}_p\mathbf{N}_{12} &= \begin{bmatrix} \mathbf{F}_{N_1,0} & \cdots & \mathbf{F}_{N_1,na} & \mathbf{G}_{N_1p,0} & \cdots & \mathbf{G}_{N_1p,nb-1} \\ \mathbf{F}_{N_1+1,0} & \cdots & \mathbf{F}_{N_1+1,na} & \mathbf{G}_{(N_1+1)p,0} & \cdots & \mathbf{G}_{(N_1+1)p,nb-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_2,0} & \cdots & \mathbf{F}_{N_2,na} & \mathbf{G}_{N_2p,0} & \cdots & \mathbf{G}_{N_2p,nb-1} \end{bmatrix} \\
 \mathbf{F}\mathbf{G}_p\mathbf{H}_p &= \begin{bmatrix} \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\ \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,na} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\ \mathbf{G}_{1p,0} & \mathbf{G}_{1p,1} & \cdots & \mathbf{G}_{1p,nb-1} \\ \mathbf{G}_{2p,0} & \mathbf{G}_{2p,1} & \cdots & \mathbf{G}_{2p,nb-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{N_pp,0} & \mathbf{G}_{N_pp,1} & \cdots & \mathbf{G}_{N_pp,nb-1} \\ \mathbf{H}_{1p,0} & \mathbf{H}_{1p,1} & \cdots & \mathbf{H}_{1p,nd-1} \\ \mathbf{H}_{2p,0} & \mathbf{H}_{2p,1} & \cdots & \mathbf{H}_{2p,nd-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{N_pp,0} & \mathbf{H}_{N_pp,1} & \cdots & \mathbf{H}_{N_pp,nd-1} \end{bmatrix}
 \end{aligned}$$

$$\mathbf{F}\mathbf{G}_p\mathbf{H}_p\mathbf{H}_{N_{12}} = \begin{bmatrix}
 \mathbf{F}_{N_1,0} & \mathbf{F}_{N_1,1} & \cdots & \mathbf{F}_{N_1,na} \\
 \mathbf{F}_{N_1+1,0} & \mathbf{F}_{N_1+1,1} & \cdots & \mathbf{F}_{N_1+1,na} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{F}_{N_2,0} & \mathbf{F}_{N_2,1} & \cdots & \mathbf{F}_{N_2,na} \\
 \mathbf{G}_{N_1p,0} & \mathbf{G}_{N_1p,1} & \cdots & \mathbf{G}_{N_1p,nb-1} \\
 \mathbf{G}_{(N_1+1)p,0} & \mathbf{G}_{(N_1+1)p,1} & \cdots & \mathbf{G}_{(N_1+1)p,nb-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{G}_{N_2p,0} & \mathbf{G}_{N_2p,1} & \cdots & \mathbf{G}_{N_2p,nb-1} \\
 \mathbf{H}_{N_1p,0} & \mathbf{H}_{N_1p,1} & \cdots & \mathbf{H}_{N_1p,nd-1} \\
 \mathbf{H}_{(N_1+1)p,0} & \mathbf{H}_{(N_1+1)p,1} & \cdots & \mathbf{H}_{(N_1+1)p,nd-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{H}_{N_2p,0} & \mathbf{H}_{N_2p,1} & \cdots & \mathbf{H}_{N_2p,nd-1}
 \end{bmatrix}$$

$$\mathbf{F}\mathbf{\Gamma}\mathbf{\Theta} = \begin{bmatrix}
 \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\
 \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,na} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\
 \mathbf{\Gamma}_{1,0} & \mathbf{\Gamma}_{1,1} & \cdots & \mathbf{\Gamma}_{1,nb-1} \\
 \mathbf{\Gamma}_{2,0} & \mathbf{\Gamma}_{2,1} & \cdots & \mathbf{\Gamma}_{2,nb-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{\Gamma}_{N_p,0} & \mathbf{\Gamma}_{N_p,1} & \cdots & \mathbf{\Gamma}_{N_p,nb-1} \\
 \mathbf{\Theta}_{1,0} & \mathbf{\Theta}_{1,1} & \cdots & \mathbf{\Theta}_{1,nd-1} \\
 \mathbf{\Theta}_{2,0} & \mathbf{\Theta}_{2,1} & \cdots & \mathbf{\Theta}_{2,nd-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{\Theta}_{N_p,0} & \mathbf{\Theta}_{N_p,1} & \cdots & \mathbf{\Theta}_{N_p,nd-1}
 \end{bmatrix}$$

$$\begin{aligned}
 \mathbf{F}\Gamma\Theta_{N_{12}} &= \begin{bmatrix} \mathbf{F}_{N_1,0} & \mathbf{F}_{N_1,1} & \cdots & \mathbf{F}_{N_1,na} \\ \mathbf{F}_{N_1+1,0} & \mathbf{F}_{N_1+1,1} & \cdots & \mathbf{F}_{N_1+1,na} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_{N_2,0} & \mathbf{F}_{N_2,1} & \cdots & \mathbf{F}_{N_2,na} \\ \Gamma_{N_1,0} & \Gamma_{N_1,1} & \cdots & \Gamma_{N_1,nb-1} \\ \Gamma_{N_1+1,0} & \Gamma_{N_1+1,1} & \cdots & \Gamma_{N_1+1,nb-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma_{N_2,0} & \Gamma_{N_2,1} & \cdots & \Gamma_{N_2,nb-1} \\ \Theta_{N_1,0} & \Theta_{N_1,1} & \cdots & \Theta_{N_1,nd-1} \\ \Theta_{N_1+1,0} & \Theta_{N_1+1,1} & \cdots & \Theta_{N_1+1,nd-1} \\ \vdots & \vdots & \vdots & \vdots \\ \Theta_{N_2,0} & \Theta_{N_2,1} & \cdots & \Theta_{N_2,nd-1} \end{bmatrix} \\
 \mathbf{G} &= \begin{bmatrix} \mathbf{G}_0 & 0 & \cdots & 0 \\ \mathbf{G}_1 & \mathbf{G}_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_{N_p-1} & \mathbf{G}_{N_p-2} & \cdots & \mathbf{G}_0 \end{bmatrix} \\
 \mathbf{G}' &= \begin{bmatrix} \mathbf{G}'_0 & 0 & \cdots & 0 \\ \mathbf{G}'_1 & \mathbf{G}'_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}'_{N_p-1} & \mathbf{G}'_{N_p-2} & \cdots & \mathbf{G}'_0 \end{bmatrix} \\
 \mathbf{G}_{N_{12u}} &= \begin{bmatrix} \mathbf{G}_{N_1-1} & \mathbf{G}_{N_1-2} & \cdots & \mathbf{G}_{N_1-N_u} \\ \mathbf{G}_{N_1} & \mathbf{G}_{N_1-1} & \cdots & \mathbf{G}_{N_1-N_u+1} \\ \vdots & \vdots & & \vdots \\ \mathbf{G}_{N_2-1} & \mathbf{G}_{N_2-2} & \cdots & \mathbf{G}_{N_2-N_u} \end{bmatrix} \\
 \mathbf{G}'_{N_{12u}} &= \begin{bmatrix} \mathbf{G}'_{N_1-1} & \mathbf{G}'_{N_1-2} & \cdots & \mathbf{G}'_{N_1-N_u} \\ \mathbf{G}'_{N_1} & \mathbf{G}'_{N_1-1} & \cdots & \mathbf{G}'_{N_1-N_u+1} \\ \vdots & \vdots & & \vdots \\ \mathbf{G}'_{N_2-1} & \mathbf{G}'_{N_2-2} & \cdots & \mathbf{G}'_{N_2-N_u} \end{bmatrix} \\
 \mathbf{G}_p(z^{-1}) &= \begin{bmatrix} \mathbf{G}_{1p,0} + \mathbf{G}_{1p,1}z^{-1} + \cdots + \mathbf{G}_{1p,nb-1}z^{-(nb-1)} \\ \mathbf{G}_{2p,0} + \mathbf{G}_{2p,1}z^{-1} + \cdots + \mathbf{G}_{2p,nb-1}z^{-(nb-1)} \\ \vdots \\ \mathbf{G}_{N_pp,0} + \mathbf{G}_{N_pp,1}z^{-1} + \cdots + \mathbf{G}_{N_pp,nb-1}z^{-(nb-1)} \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{H} &= \begin{bmatrix} \mathbf{H}_0 & 0 & \cdots & 0 \\ \mathbf{H}_1 & \mathbf{H}_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{N_p-1} & \mathbf{H}_{N_p-2} & \cdots & \mathbf{H}_0 \end{bmatrix} \\
 \mathbf{H}' &= \begin{bmatrix} \mathbf{H}'_0 & 0 & \cdots & 0 \\ \mathbf{H}'_1 & \mathbf{H}'_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}'_{N_p-1} & \mathbf{H}'_{N_p-2} & \cdots & \mathbf{H}'_0 \end{bmatrix} \\
 \mathbf{H}_{N_{12u}} &= \begin{bmatrix} \mathbf{H}_{N_1-1} & \mathbf{H}_{N_1-2} & \cdots & \mathbf{H}_{N_1-N_u} \\ \mathbf{H}_{N_1} & \mathbf{H}_{N_1-2} & \cdots & \mathbf{H}_{N_1-N_u+1} \\ \vdots & \vdots & & \vdots \\ \mathbf{H}_{N_2-1} & \mathbf{H}_{N_2-2} & \cdots & \mathbf{H}_{N_2-N_u} \end{bmatrix} \\
 \mathbf{H}'_{N_{12u}} &= \begin{bmatrix} \mathbf{H}'_{N_1-1} & \mathbf{H}'_{N_1-2} & \cdots & \mathbf{H}'_{N_1-N_u} \\ \mathbf{H}'_{N_1} & \mathbf{H}'_{N_1-2} & \cdots & \mathbf{H}'_{N_1-N_u+1} \\ \vdots & \vdots & & \vdots \\ \mathbf{H}'_{N_2-1} & \mathbf{H}'_{N_2-2} & \cdots & \mathbf{H}'_{N_2-N_u} \end{bmatrix} \\
 \mathbf{H}_p(z^{-1}) &= \begin{bmatrix} \mathbf{H}_{1p}(z^{-1}) \\ \mathbf{H}_{2p}(z^{-1}) \\ \vdots \\ \mathbf{H}_{N_pp}(z^{-1}) \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{H}_{1p,0} + \mathbf{H}_{1p,1}z^{-1} + \cdots + \mathbf{H}_{1p,nd-1}z^{-(nd-1)} \\ \mathbf{H}_{2p,0} + \mathbf{H}_{2p,1}z^{-1} + \cdots + \mathbf{H}_{2p,nd-1}z^{-(nd-1)} \\ \vdots \\ \mathbf{H}_{N_pp,0} + \mathbf{H}_{N_pp,1}z^{-1} + \cdots + \mathbf{H}_{N_pp,nd-1}z^{-(nd-1)} \end{bmatrix}
 \end{aligned}$$

$$\mathbf{HFG}_p\mathbf{H}_p = \begin{bmatrix}
 \mathbf{H}_0 & 0 & \cdots & 0 \\
 \mathbf{H}_1 & \mathbf{H}_0 & \cdots & 0 \\
 \vdots & \vdots & \ddots & \vdots \\
 \mathbf{H}_{N_p-1} & \mathbf{H}_{N_p-2} & \cdots & \mathbf{H}_0 \\
 \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\
 \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,na} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\
 \mathbf{G}_{1p,0} & \mathbf{G}_{1p,1} & \cdots & \mathbf{G}_{1p,nb-1} \\
 \mathbf{G}_{2p,0} & \mathbf{G}_{2p,1} & \cdots & \mathbf{G}_{2p,nb-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{G}_{N_pp,0} & \mathbf{G}_{N_pp,1} & \cdots & \mathbf{G}_{N_pp,nb-1} \\
 \mathbf{H}_{1p,0} & \mathbf{H}_{1p,1} & \cdots & \mathbf{H}_{1p,nd-1} \\
 \mathbf{H}_{2p,0} & \mathbf{H}_{2p,1} & \cdots & \mathbf{H}_{2p,nd-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{H}_{N_pp,0} & \mathbf{H}_{N_pp,1} & \cdots & \mathbf{H}_{N_pp,nd-1}
 \end{bmatrix}$$

$$\mathbf{H}'\mathbf{F}\mathbf{G}\Theta = \begin{bmatrix}
 \mathbf{H}'_0 & 0 & \cdots & 0 \\
 \mathbf{H}'_1 & \mathbf{H}'_0 & \cdots & 0 \\
 \vdots & \vdots & \ddots & \vdots \\
 \mathbf{H}'_{N_p-1} & \mathbf{H}'_{N_p-2} & \cdots & \mathbf{H}'_0 \\
 \mathbf{F}_{1,0} & \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,na} \\
 \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \cdots & \mathbf{F}_{2,na} \\
 \vdots & \vdots & \vdots & \vdots \\
 \mathbf{F}_{N_p,0} & \mathbf{F}_{N_p,1} & \cdots & \mathbf{F}_{N_p,na} \\
 \Gamma_{1,0} & \Gamma_{1,1} & \cdots & \Gamma_{1,nb-1} \\
 \Gamma_{2,0} & \Gamma_{2,1} & \cdots & \Gamma_{2,nb-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \Gamma_{N_p,0} & \Gamma_{N_p,1} & \cdots & \Gamma_{N_p,nb-1} \\
 \Theta_{1,0} & \Theta_{1,1} & \cdots & \Theta_{1,nd-1} \\
 \Theta_{2,0} & \Theta_{2,1} & \cdots & \Theta_{2,nd-1} \\
 \vdots & \vdots & \vdots & \vdots \\
 \Theta_{N_p,0} & \Theta_{N_p,1} & \cdots & \Theta_{N_p,nd-1}
 \end{bmatrix}$$

$$\mathbf{T}(z^{-1}) = \mathbf{T}_0 + \mathbf{T}_1 z^{-1} + \mathbf{T}_2 z^{-2} + \cdots + \mathbf{T}_{nt} z^{-nt}$$

$$\begin{aligned}
 \mathbf{U} &= \begin{bmatrix} \Delta \mathbf{u}(t) \\ \Delta \mathbf{u}(t+1) \\ \dots \\ \Delta \mathbf{u}(t+N_p-1) \end{bmatrix} \\
 \mathbf{U}_{N_u} &= \begin{bmatrix} \Delta \mathbf{u}(t) \\ \Delta \mathbf{u}(t+1) \\ \dots \\ \Delta \mathbf{u}(t+N_u-1) \end{bmatrix} \\
 \mathbf{V} &= \begin{bmatrix} \Delta \mathbf{v}(t+1) \\ \Delta \mathbf{v}(t+2) \\ \dots \\ \Delta \mathbf{v}(t+N_p) \end{bmatrix} \\
 \mathbf{VYUV} &= \begin{bmatrix} \Delta \mathbf{v}(t+1) \\ \Delta \mathbf{v}(t+2) \\ \vdots \\ \Delta \mathbf{v}(t+N_p) \\ \mathbf{y}(t) \\ \mathbf{y}(t-1) \\ \vdots \\ \mathbf{y}(t-na) \\ \Delta \mathbf{u}(t-1) \\ \Delta \mathbf{u}(t-2) \\ \vdots \\ \Delta \mathbf{u}(t-nb) \\ \Delta \mathbf{v}(t) \\ \Delta \mathbf{v}(t-1) \\ \vdots \\ \Delta \mathbf{v}(t-nd+1) \end{bmatrix} \\
 \mathbf{W} &= \begin{bmatrix} \mathbf{w}(t+1) \\ \mathbf{w}(t+2) \\ \vdots \\ \mathbf{w}(t+N_p) \end{bmatrix} \\
 \mathbf{W}_{N_{12}} &= \begin{bmatrix} \mathbf{w}(t+N_1) \\ \mathbf{w}(t+N_1+1) \\ \vdots \\ \mathbf{w}(t+N_2) \end{bmatrix}
 \end{aligned}$$

\mathbf{VYUV}^f entsprechend
über $\Delta \mathbf{v}(t+1)$, $\mathbf{y}^f(t)$,
 $\Delta \mathbf{u}^f(t-1)$ und $\Delta \mathbf{v}^f(t)$
definiert

$$\begin{aligned}
 \mathbf{Y} &= \begin{bmatrix} \hat{\mathbf{y}}(t+1) \\ \hat{\mathbf{y}}(t+2) \\ \vdots \\ \hat{\mathbf{y}}(t+N_p) \end{bmatrix} \\
 \mathbf{Y}_{N_{12}} &= \begin{bmatrix} \hat{\mathbf{y}}(t+N_1) \\ \hat{\mathbf{y}}(t+N_1+1) \\ \vdots \\ \hat{\mathbf{y}}(t+N_2) \end{bmatrix} \\
 \mathbf{YU} &= \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{y}(t-1) \\ \vdots \\ \mathbf{y}(t-na) \\ \Delta \mathbf{u}(t-1) \\ \Delta \mathbf{u}(t-2) \\ \vdots \\ \Delta \mathbf{u}(t-nb) \end{bmatrix} \\
 \mathbf{YUV} &= \begin{bmatrix} \mathbf{y}(t) \\ \mathbf{y}(t-1) \\ \vdots \\ \mathbf{y}(t-na) \\ \Delta \mathbf{u}(t-1) \\ \Delta \mathbf{u}(t-2) \\ \vdots \\ \Delta \mathbf{u}(t-nb) \\ \Delta \mathbf{v}(t) \\ \Delta \mathbf{v}(t-1) \\ \vdots \\ \Delta \mathbf{v}(t-nd+1) \end{bmatrix} \quad \begin{array}{l} \mathbf{YUV}^f \text{ entsprechend über} \\ \mathbf{y}^f(t), \Delta \mathbf{u}^f(t-1) \text{ und} \\ \Delta \mathbf{v}^f(t) \text{ definiert} \end{array} \\
 \Gamma(z^{-1}) &= \begin{bmatrix} \Gamma_1(z^{-1}) \\ \Gamma_2(z^{-1}) \\ \vdots \\ \Gamma_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \Gamma_{1,0} + \Gamma_{1,1}z^{-1} + \cdots + \Gamma_{1,nb-1}z^{-(nb-1)} \\ \Gamma_{2,0} + \Gamma_{2,1}z^{-1} + \cdots + \Gamma_{2,nb-1}z^{-(nb-1)} \\ \vdots \\ \Gamma_{N_p,0} + \Gamma_{N_p,1}z^{-1} + \cdots + \Gamma_{N_p,nb-1}z^{-(nb-1)} \end{bmatrix}
 \end{aligned}$$

$$\Theta(z^{-1}) = \begin{bmatrix} \Theta_1(z^{-1}) \\ \Theta_2(z^{-1}) \\ \vdots \\ \Theta_{N_p}(z^{-1}) \end{bmatrix} = \begin{bmatrix} \Theta_{1,0} + \Theta_{1,1}z^{-1} + \dots + \Theta_{1,nd-1}z^{-(nd-1)} \\ \Theta_{2,0} + \Theta_{2,1}z^{-1} + \dots + \Theta_{2,nd-1}z^{-(nd-1)} \\ \vdots \\ \Theta_{N_p,0} + \Theta_{N_p,1}z^{-1} + \dots + \Theta_{N_p,nd-1}z^{-(nd-1)} \end{bmatrix}$$

E.2.2 Dimensionen/Dimensions/维度

$\mathbf{A}(z^{-1})$: Matrix($n \times n$)	(Zeilen \times Spalten)
$\mathbf{B}(z^{-1})$: Matrix($n \times m$)	
$\mathbf{C}(z^{-1})$: Matrix($n \times n$)	
$\mathbf{D}(z^{-1})$: Matrix($n \times l$)	
$\mathbf{E}_j(z^{-1})$: Matrix($n \times n$)	
$\mathbf{F}_j(z^{-1})$: Matrix($n \times n$)	
$\mathbf{f}(t), \mathbf{f}^f(t)$: Vektor($n \times 1$)	
$\mathbf{G}_d(z^{-1}), \mathbf{G}_j(z^{-1}), \mathbf{G}_{jp}(z^{-1})$: Matrix($n \times m$)	
$\mathbf{G}_j'(z^{-1})$: Matrix($n \times m$)	Hinweis beachten!
$\mathbf{H}_d(z^{-1}), \mathbf{H}_j(z^{-1}), \mathbf{H}_{jp}(z^{-1})$: Matrix($n \times l$)	
$\mathbf{H}_j'(z^{-1})$: Matrix($n \times l$)	Hinweis beachten!
$\mathbf{T}(z^{-1})$: Matrix($n \times n$)	
$\mathbf{u}(t), \mathbf{u}^f(t)$: Vektor($m \times 1$)	
$\mathbf{v}(t), \mathbf{v}^f(t)$: Vektor($l \times 1$)	
$\mathbf{w}(t)$: Vektor($n \times 1$)	
$\mathbf{y}(t), \mathbf{y}^f(t)$: Vektor($n \times 1$)	
$\Gamma_j(z^{-1})$: Matrix($n \times m$)	Hinweis beachten!
$\Theta_j(z^{-1})$: Matrix($n \times l$)	Hinweis beachten!
$\boldsymbol{\xi}(t)$: Vektor($n \times 1$)	

Hinweis: Bei *Anwendung der Filterung* müssen der Stellgrößenvektor $\mathbf{u}(t)$, der Störgrößenvektor $\mathbf{v}(t)$ und der Regelgrößenvektor $\mathbf{y}(t)$ alle *die gleiche Dimension besitzen* (s. Kapitel 8.3.3 ab Seite 209). Daher muss in diesem Fall $n = m = l$ gelten.

Note: If *filtering is applied* the vector of actuating variables $\mathbf{u}(t)$, the vector of disturbances $\mathbf{v}(t)$ and the vector of controlled variables $\mathbf{y}(t)$ must all have *the same dimension* (see Chapter 8.3.3 on page 209 et seqq.). Therefore, in all cases $n = m = l$ applies.

提示：在应用滤波时，需满足控制变量向量 $\mathbf{u}(t)$ 、扰动变量向量 $\mathbf{v}(t)$ 和被控变量向量 $\mathbf{y}(t)$ 具有相同的维度（见章节 8.3.3 第 209 页）。所以，此时需满足条件 $n = m = l$ 。

Anhang/Appendix/附录 F

Verfahren zur Matrixinversion/Methods for Matrix Inversion/矩阵求逆方法

Für die Inversion einer Matrix bieten sich verschiedene Algorithmen an, deren Eignung für eine beim GPC-Regler notwendige Matrizeninversion im Folgenden untersucht wird. Besonderes Augenmerk wird dabei die die für die Matrixinversion benötigte Rechenzeit gelegt.

F.1 Gauß-Verfahren/Gauss algorithm/高斯法

Hauptidee des Gauß-Verfahrens ist, die Matrixgleichung $\mathbf{A} \cdot \mathbf{A}' = \mathbf{I}$ zu lösen. Der Algorithmus funktioniert dabei nach folgenden Gesichtspunkten:

1. Zuerst muss die Matrix \mathbf{A} auf Dreiecksform gebracht werden. Dazu wird nacheinander jeweils ein Diagonalelement aus den darunter liegenden Matrixzeilen entfernt, so dass sich die Matrix schließlich in Dreiecksform präsentiert. Die Multiplikatoren für jeden Fall werden dabei in einer Hilfsmatrix \mathbf{B} festgehalten. Eine Pivottisierung zur Optimierung der Berechnung wurde nicht implementiert.
2. Nach der Umformung der Eingangsmatrix \mathbf{A} wird dasselbe nun mit der Einheitsmatrix \mathbf{I} durchgeführt. Dabei werden die in der Matrix \mathbf{B} zwischengespeicherten Faktoren verwendet.
3. Da nun \mathbf{A} und \mathbf{I} beide in gleicher Weise in \mathbf{A} bzw. \mathbf{I} umgeformt sind, haben wir nun eine Gleichung $\mathbf{A} \cdot \mathbf{A}' = \mathbf{I}$, wobei \mathbf{A} eine obere Dreiecksmatrix darstellt. Die Einheitsmatrix \mathbf{I} hat sich in eine untere Dreiecksmatrix \mathbf{I} verwandelt, deren Diagonalelemente alle 1 sind. Das Gleichungssystem kann leicht gelöst werden und damit sind die Elemente der Matrix $\mathbf{A}' = \mathbf{A}^{-1}$ bestimmt.

Methods for matrix inversion

For matrix inversion there exist several algorithms whose suitability for a matrix inversion that is necessary for a GPC controller will be examined in the following. Special attention is given to the computation time that is necessary for the matrix inversion.

F.1 Gauss algorithm

The basic idea of the Gauss algorithm is to solve the matrix equation $\mathbf{A} \cdot \mathbf{A}' = \mathbf{I}$. The algorithm is described in the following:

1. First the matrix \mathbf{A} has to be transformed into triangular form. For this purpose, each diagonal element is removed from the underneath matrix rows one after each other so that then, the matrix has triangular form. The multipliers for each case are stored in an auxiliary matrix \mathbf{B} . Pivoting for the optimization of the calculation was not implemented.
2. After having transformed the input matrix \mathbf{A} , the same will also be carried out with the identity matrix \mathbf{I} using the factors stored in matrix \mathbf{B} .
3. Since now \mathbf{A} and \mathbf{I} are both transformed in identical manner into $\mathbf{\hat{A}}$ and $\mathbf{\hat{I}}$, the equation $\mathbf{\hat{A}} \cdot \mathbf{A}' = \mathbf{\hat{I}}$ results, in which $\mathbf{\hat{A}}$ is an upper triangular matrix. The identity matrix \mathbf{I} has transformed into a lower triangular matrix $\mathbf{\hat{I}}$ whose diagonal elements are all 1. The system of equations can easily be solved and then, the elements of the matrix $\mathbf{A}' = \mathbf{A}^{-1}$ are determined.

矩阵求逆的方法

存在多种不同的算法可用于矩阵求逆，接下来将考察不同算法对于 GPC 控制器所必需的矩阵求逆的适用性。特别关注的是矩阵求逆所需的计算时间。

F.1 高斯法

高斯法的基本思想是求解矩阵方程 $\mathbf{A} \cdot \mathbf{A}' = \mathbf{I}$ 。这一算法的执行基于以下几点考虑：

1. 首先，必须将矩阵 \mathbf{A} 变换为三角形式。为此，每个对角线上的元素都要从其下面的矩阵行中一个接一个地去除，如此矩阵就变成了三角形式。而且，每一情况下的乘数都被存储在辅助矩阵 \mathbf{B} 。用于计算优化所需的寻找主元过程则无需实施。
2. 在对输入矩阵 \mathbf{A} 进行上述变换之后，还需对单位矩阵 \mathbf{I} 实施相同的变换过程。为此，需要使用前面在辅助矩阵 \mathbf{B} 中存储的相关系数。
3. 由于已经将 \mathbf{A} 和 \mathbf{I} 以相同的方式变换为 $\mathbf{\hat{A}}$ 和 $\mathbf{\hat{I}}$ ，所以有方程 $\mathbf{\hat{A}} \cdot \mathbf{A}' = \mathbf{\hat{I}}$ ，其中， $\mathbf{\hat{A}}$ 表示一个上三角矩阵。单位矩阵 \mathbf{I} 也被变换为一个下三角矩阵 $\mathbf{\hat{I}}$ ，而且其对角线上的元素全为 1。如此既可以轻易地求解上述方程，并随之可确定矩阵 $\mathbf{A}' = \mathbf{A}^{-1}$ 的元素。

F.2 Gauß-Jordan-Verfahren/Gauss Jordan algorithm/高斯-若尔当法

Das Verfahren nach Gauß-Jordan entspricht dem in Kap. F.1 erläuterten Verfahren nach Gauß mit dem Unterschied, dass die Matrix \mathbf{A} nicht auf Dreiecksform, sondern auf Diagonalform gebracht wird. Man erreicht dies, indem die Diagonalelemente nicht nur aus den darunter liegenden, sondern auch aus allen darüber liegenden Matrixzeilen eliminiert werden.

F.3 Austauschverfahren/Exchange algorithm/互换法

Dem Austauschverfahren liegt die Überlegung zugrunde, dass die Gleichung $\mathbf{y} = \mathbf{A}\mathbf{x}$ sich unter Verwendung der inversen Matrix $\mathbf{A}' = \mathbf{A}^{-1}$ schreiben lässt als $\mathbf{x} = \mathbf{A}'\mathbf{y}$. Demnach lässt sich die Inversion einer Matrix durch den Austausch der unabhängigen Variable gegen die abhängige realisieren. Der dazu notwendige Weg wird bei Ose et al. [98] sehr gut erläutert.

F.4 LR-Zerlegung/LR decomposition/LR 分解

Die LR-Zerlegung, auch als LU-Zerlegung bekannt, zerlegt die Matrix \mathbf{A} in je eine untere (linke) und eine obere (rechte) Dreiecksmatrix \mathbf{L} bzw. \mathbf{R} , so dass gilt $\mathbf{A} = \mathbf{L}\mathbf{R}$. Führt man vor der Zerlegung eine Pivotisierung durch, werden die Tauschbewegungen in einer Permutationsmatrix \mathbf{P} festgehalten, so dass schließlich $\mathbf{P}\mathbf{A} = \mathbf{L}\mathbf{R}$ gilt. Demnach gilt auch $\mathbf{A} = \mathbf{P}^T\mathbf{L}\mathbf{R}$. Da \mathbf{L} und \mathbf{R} Dreiecksmatrizen sind, können sie bequem invertiert werden und aus \mathbf{L}^{-1} und \mathbf{R}^{-1} erhält man dann $\mathbf{A}^{-1} = \mathbf{P}^T\mathbf{R}^{-1}\mathbf{L}^{-1}$. Beachte den Tausch von \mathbf{L} und \mathbf{R} wegen $(\mathbf{L}\mathbf{R})^{-1} = \mathbf{R}^{-1}\mathbf{L}^{-1}$.

F.5 Verfahren von Cholesky/Algorithm of Cholesky/科列斯基法

Der Vollständigkeit halber sei noch das Verfahren von Cholesky erwähnt, welches gegenüber dem LR-Verfahren eine deutliche Verringerung des Berechnungsaufwands verspricht [112, 124]. Da es aber nur für die Inversion symmetrischer Matrizen geeignet ist, dies aber bei der Berechnung der MPC-Systemmatrizen nicht vorausgesetzt werden kann, entfällt eine nähere Betrachtung.

F.2 Gauss Jordan algorithm The algorithm of Gauss Jordan corresponds to the algorithm by Gauss, explained in section F.1, with the difference that the matrix \mathbf{A} is not transformed into triangular form, but into diagonal form. This can be achieved if the diagonal elements are not only eliminated from the lower, but also from the above

matrix rows.

F.3 Exchange algorithm The exchange algorithm is based on the assumption that the equation $\mathbf{y} = \mathbf{Ax}$ can be written as $\mathbf{x} = \mathbf{A}'\mathbf{y}$ using the inverse matrix $\mathbf{A}' = \mathbf{A}^{-1}$. Therefore, the inversion of a matrix can be realized via the exchange of an independent variable with a dependent one. The procedure is very well explained by Ose et al. [98].

F.4 LR decomposition The LR decomposition, also known as LU decomposition, decomposes the matrix \mathbf{A} into a lower (left) and into an upper (right) triangular matrix \mathbf{L} and \mathbf{R} , such that $\mathbf{A} = \mathbf{LR}$ applies. If pivoting is done before the decomposition, the exchange movements are stored in a permutation matrix \mathbf{P} so that in the end $\mathbf{PA} = \mathbf{LR}$ applies. Therefore, $\mathbf{A} = \mathbf{P}^T\mathbf{LR}$ does apply, too. Since \mathbf{L} and \mathbf{R} are triangular matrices, they can be easily inverted and from \mathbf{L}^{-1} and \mathbf{R}^{-1} the inverse matrix $\mathbf{A}^{-1} = \mathbf{P}^T\mathbf{R}^{-1}\mathbf{L}^{-1}$ can be obtained. The exchange of \mathbf{L} and \mathbf{R} because of $(\mathbf{LR})^{-1} = \mathbf{R}^{-1}\mathbf{L}^{-1}$ has to be noted.

F.5 Algorithm of Cholesky For reasons of completeness, the algorithm of Cholesky should be mentioned, too, which, compared to the LR decomposition, promises a significant reduction of the calculation complexity [112, 124]. But, since it is suitable only for the inversion of symmetrical matrices, it cannot be used for the calculation of MPC system matrices and so it is omitted from further discussions.

F.2 高斯-若尔当法 高斯-若尔当法与上一小节 F.1 所介绍的高斯法相对应，其不同之处在于：此时将矩阵 \mathbf{A} 转换为对角形式，而并非高斯法中的三角形式。为此，不仅每个对角线上的元素都要从其下面的矩阵行中一个接一个地去除，而且其上方的矩阵行中也要同样去除，如此矩阵就变成了对角形式。

F.3 互换法 互换法的基本思想是借助于逆矩阵 $\mathbf{A}' = \mathbf{A}^{-1}$ 可将方程 $\mathbf{y} = \mathbf{Ax}$ 写为 $\mathbf{x} = \mathbf{A}'\mathbf{y}$ 的形式。从而，矩阵的求逆可以通过自变量与因变量的互换而得到，学者 Ose 等详细阐述了上述过程 [98]。

F.4 LR 分解 LR 分解将矩阵 \mathbf{A} 分解为一个下 (左) 三角矩阵 \mathbf{L} 和一个上 (右) 三角矩阵 \mathbf{R} ，从而满足 $\mathbf{A} = \mathbf{LR}$ ，这一分解方式也被称为 LU 分解。如果在分解之前执行寻找主元的过程，那么相应的变换过程将被存储在一个置换矩阵 \mathbf{P} 中，从而最后有 $\mathbf{PA} = \mathbf{LR}$ 。因此， $\mathbf{A} = \mathbf{P}^T\mathbf{LR}$ 。由于 \mathbf{L} 和 \mathbf{R} 为三角矩阵，对其求逆也较为方便。基于 \mathbf{L}^{-1} 和 \mathbf{R}^{-1} 即可得到 $\mathbf{A}^{-1} = \mathbf{P}^T\mathbf{R}^{-1}\mathbf{L}^{-1}$ 。要注意的是，由于 $(\mathbf{LR})^{-1} = \mathbf{R}^{-1}\mathbf{L}^{-1}$ 的缘故， \mathbf{L} 和 \mathbf{R} 需要交换位置。

F.5 科列斯基法 为完整性起见，此处也应当提及科列斯基法。相对于 LR 分解，科列斯基法能够明显地降低计算量 [112, 124]。但是，这一方法也仅适用于对称矩阵的求逆，而在 MPC 系统矩阵的计算时并不满足对称性的前提条件，所以此处不再详细探讨。

F.6 Rechenzeitaufwand/Computation times/计算时间

Bei der Implementation von GPC ist es äußerst wichtig, für die Matrixoperationen möglichst effiziente Rechenmethoden einzusetzen. Daher wurden bei der Programmierung der oben beschriebenen Verfahren Optimierungen durchgeführt, d. h. es wurden alle Divisionen durch denselben Divisor innerhalb von Schleifen durch Zuweisungen *vor* der Schleife und *Multiplikationen innerhalb der Schleife* ersetzt. Danach wurden die oben beschriebenen Verfahren einem Test unterzogen. Die Ergebnisse sind in Tabelle F.1 dargestellt. Für den Test wurde ein Rechner mit AMD Duron[®] Prozessor (900 MHz) und 128 MB RAM verwendet.

Verfahren	Matrixgröße				
	3×3	10×10	15×15	25×25	100×100
Gauß	1,9	18,9	56,2	239,4	14390
Gauß-Jordan	1,1	11,8	35,5	395,1	26010
Austauschverfahren	1,2	19,9	58,7	268,6	16980
LR (mit Pivot)	2,4	20,9	54,9	206,9	10010
LR (ohne Pivot)	1,8	16,1	45,6	177,0	9420

F.1: Rechenzeiten für Matrixinversionen in μs

Wie die gezeigten Ergebnisse verdeutlichen, hängt die Auswahl, welche Vorgehensweise optimal ist, von der Größe der zu invertierenden Matrix ab. Für kleine Matrizen bis etwa 15×15 ist das Gauß-Jordan-Verfahren das schnellste, danach zeigt sich die LR-Zerlegung überlegen.

F.6 Computation times

For the implementation of GPC, it is extremely important to use very efficient methods for matrix operations. Therefore, for the programming of the methods described above, optimizations were carried out, i. e. all divisions with the same divisor within loops were replaced by allocations *before* and *multiplications within* the loop. Then the methods described above were tested. The results are presented in Table F.2. For the examinations a computer with AMD Duron[®] processor (900 MHz) and 128 MB RAM was used.

Algorithm	Matrix size				
	3 × 3	10 × 10	15 × 15	25 × 25	100 × 100
Gauss	1.9	18.9	56.2	239.4	14390
Gauss-Jordan	1.1	11.8	35.5	395.1	26010
Exchange algorithm	1.2	19.9	58.7	268.6	16980
LR (with pivoting)	2.4	20.9	54.9	206.9	10010
LR (without pivoting)	1.8	16.1	45.6	177.0	9420

F.2: Calculation times for matrix inversion in μs

As the presented results clarify, the choice of the optimum method depends on the size of the matrix that should be inverted. For small matrices with sizes up to 15×15 , the Gauss-Jordan algorithm is the fastest one, whereas for larger matrices, the LR decomposition is superior.

F.6 计算时间

对于 GPC 的实现来说, 应用尽可能有效的计算方法来进行矩阵运算是至关重要的。因此, 在编程实现前面所述的方法时进行了相应的优化, 也就是说, 单个循环内所有具有相同除数的除法都由循环前的提示和循环内的乘法所取代。从而, 可以对前面描述的方法进行测试。表 F.3 给出了相应的结果, 所使用的计算机含有 AMD Duron[®] 处理器 (900 MHz) 以及 128 MB RAM。

算法	矩阵大小				
	3 × 3	10 × 10	15 × 15	25 × 25	100 × 100
高斯法	1.9	18.9	56.2	239.4	14390
高斯-若尔当法	1.1	11.8	35.5	395.1	26010
互换法	1.2	19.9	58.7	268.6	16980
LR 分解 (含寻找主元)	2.4	20.9	54.9	206.9	10010
LR 分解 (不含寻找主元)	1.8	16.1	45.6	177.0	9420

F.3: 矩阵求逆所需的计算时间, 单位 μs

正如表中结果所示, 最优处理方式的选择取决于待求逆矩阵的大小。对于小于 15×15 的矩阵来说, 高斯-若尔当法是最快速的, 当矩阵更大时, LR 分解法则体现出了优势。

Anhang/Appendix/附录 G

Alternativverfahren Matrizenzerlegung/ Alternative Method for Matrix Decomposition/矩阵分解的替代性方法

Das Goodwin/Sin [44, Kap. 2.3.5] entlehnte und von Camacho/Bordons [20, Kap. 6.2.1] erläuterte Verfahren zur Zerlegung der Übertragungs-Polynommatrix $\mathbf{G}_d(z^{-1})$ in zwei links kopprime Matrizen $\mathbf{A}(z^{-1})$ und $\mathbf{B}(z^{-1})$ beruht auf folgender Vorgehensweise:

Zu Beginn wird $\mathbf{G}_d(z^{-1})$ in einen beliebigen rechten Matrixbruch umgewandelt.

$$\mathbf{G}_d(z^{-1}) = \mathbf{N}_R(z^{-1})\mathbf{D}_R(z^{-1})^{-1} \quad (\text{G.1})$$

Analog der Zerlegung in einen linken Bruch (s. Kap. 8.1.2 auf Seite 169) wird dazu $\mathbf{D}_R(z^{-1})$ als Diagonalmatrix gewählt, deren Diagonalelemente jeweils gleich dem kleinsten gemeinsamen Nenner der entsprechenden Spalten von $\mathbf{G}_d(z^{-1})$ sind. $\mathbf{N}_R(z^{-1})$ kann dann einfach über $\mathbf{N}_R(z^{-1}) = \mathbf{G}_d(z^{-1})\mathbf{D}_R(z^{-1})$ bestimmt werden. Dabei brauchen $\mathbf{N}_R(z^{-1})$ und $\mathbf{D}_R(z^{-1})$ nicht rechts koprim zu sein.

Nun wird eine unimodulare Matrix $\mathbf{U}(z^{-1})$ gesucht, die die Gleichung

$$\begin{bmatrix} \mathbf{U}_{11}(z^{-1}) & \mathbf{U}_{12}(z^{-1}) \\ \mathbf{U}_{21}(z^{-1}) & \mathbf{U}_{22}(z^{-1}) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{D}_R(z^{-1}) \\ \mathbf{N}_R(z^{-1}) \end{bmatrix} = \begin{bmatrix} \mathbf{R}(z^{-1}) \\ 0 \end{bmatrix} \quad (\text{G.2})$$

wobei

$$\mathbf{U}(z^{-1}) = \begin{bmatrix} \mathbf{U}_{11}(z^{-1}) & \mathbf{U}_{12}(z^{-1}) \\ \mathbf{U}_{21}(z^{-1}) & \mathbf{U}_{22}(z^{-1}) \end{bmatrix} \quad (\text{G.3})$$

$$\mathbf{P}(z^{-1}) = \begin{bmatrix} \mathbf{D}_R(z^{-1}) \\ \mathbf{N}_R(z^{-1}) \end{bmatrix} \quad (\text{G.4})$$

$$\mathbf{R}(z^{-1}) = \text{größter gemeinsamer rechter Teiler von } \mathbf{D}_R(z^{-1}) \text{ und } \mathbf{N}_R(z^{-1}) \quad (\text{G.5})$$

erfüllt. Bringt man $\mathbf{P}(z^{-1})$ durch elementare Zeilenumformungen auf obere rechte Dreiecksform, erhält man gemäß der in Anhang A genannten Eigenschaften von Polynommatrizen genau den gesuchten größten gemeinsamen rechten Teiler von $\mathbf{D}_R(z^{-1})$ und $\mathbf{N}_R(z^{-1})$. Wendet man nun dieselben Umformungen auf eine Einheitsmatrix an, ergibt sich die unimodulare Matrix $\mathbf{U}(z^{-1})$.

Alternative method for matrix decomposition

The method for decomposition of the transfer polynomial matrix $\mathbf{G}_d(z^{-1})$ into two left coprime matrices $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$, originally developed by Goodwin/Sin [44, Chapter 2.3.5] and explained by Camacho/Bordons [20, Chapter 6.2.1], is based on the following approach:

At the beginning $\mathbf{G}_d(z^{-1})$ is converted into an arbitrary right matrix fraction, see equation (G.1). Analog to the decomposition into a left fraction (see Chapter 8.1.2 on page 169) $\mathbf{D}_R(z^{-1})$ is chosen as a diagonal matrix whose diagonal elements are equal to the smallest common denominator of the corresponding columns of $\mathbf{G}_d(z^{-1})$. Then, $\mathbf{N}_R(z^{-1})$ can easily be determined via $\mathbf{N}_R(z^{-1}) = \mathbf{G}_d(z^{-1})\mathbf{D}_R(z^{-1})$. Here, $\mathbf{N}_R(z^{-1})$ and $\mathbf{D}_R(z^{-1})$ do not have to be right coprime.

Now, a unimodular matrix $\mathbf{U}(z^{-1})$ has to be determined which fulfils the equation (G.2), in which equation (G.3), (G.3) and

$$\mathbf{R}(z^{-1}) = \text{Greatest common right divisor of } \mathbf{D}_R(z^{-1}) \text{ and } \mathbf{N}_R(z^{-1}) \quad (\text{G.5})$$

If $\mathbf{P}(z^{-1})$ is transformed into upper right triangular form by elementary row transformations, the greatest common right divisor of $\mathbf{D}_R(z^{-1})$ and $\mathbf{N}_R(z^{-1})$ can be obtained according to the properties of polynomial matrices mentioned in appendix A. If the same transformations are applied to an identity matrix, the unimodular matrix $\mathbf{U}(z^{-1})$ will result.

矩阵分解的替代性方法

由学者 Goodwin/Sin [44, 章节 2.3.5] 所提出的, 并由学者 Camacho/Bordons [20, 章节 6.2.1] 所阐述的可用于将传递多项式矩阵 $\mathbf{G}_d(z^{-1})$ 分解为两个左互素的矩阵 $\mathbf{A}(z^{-1})$ 和 $\mathbf{B}(z^{-1})$ 基于如下所述的处理方式:

首先, 将 $\mathbf{G}_d(z^{-1})$ 转换为一个任意的右矩阵分数形式, 见方程 (G.1)。

类似于分解为左分数的情形 (见章节 8.1.2 第 169 页), 为此需要选择一个对角矩阵 $\mathbf{D}_R(z^{-1})$, 其对角线上的元素分别等于 $\mathbf{G}_d(z^{-1})$ 相应列的最小公分母。然后, 即可根据 $\mathbf{N}_R(z^{-1}) = \mathbf{G}_d(z^{-1})\mathbf{D}_R(z^{-1})$ 轻易地确定 $\mathbf{N}_R(z^{-1})$ 。而且, 并不需要 $\mathbf{N}_R(z^{-1})$ 和 $\mathbf{D}_R(z^{-1})$ 是右互素的。

现在要寻找的是一个单模矩阵 $\mathbf{U}(z^{-1})$, 它满足方程 (G.2), 其中, 有方程 (G.3), (G.4) 和

$$\mathbf{R}(z^{-1}) = \mathbf{D}_R(z^{-1}) \text{ 和 } \mathbf{N}_R(z^{-1}) \text{ 的最大右公约数} \quad (\text{G.5})$$

如果通过基本的行变换将 $\mathbf{P}(z^{-1})$ 转换为上右三角矩阵的形式, 从而可以根据附录 A 中所描述的多项式矩阵的特性得到所要寻找的 $\mathbf{D}_R(z^{-1})$ 和 $\mathbf{N}_R(z^{-1})$ 的最大右公约数。如果将相同的变换过程应用于单位矩阵, 即可得到单模矩阵 $\mathbf{U}(z^{-1})$ 。

Wie sich beweisen lässt, sind die Untermatrizen $\mathbf{U}_{21}(z^{-1})$ und $\mathbf{U}_{22}(z^{-1})$ links koprim. Da $\mathbf{U}_{22}(z^{-1})$ darüberhinaus nichtsingulär ist, lässt sich aus (G.2) folgern:

$$\mathbf{N}_{\mathbf{R}}(z^{-1})\mathbf{D}_{\mathbf{R}}(z^{-1})^{-1} = -\mathbf{U}_{22}(z^{-1})^{-1}\mathbf{U}_{21}(z^{-1})$$

Ein Vergleich mit Gleichung (8.10) ergibt:

$$\begin{aligned}\mathbf{A}(z^{-1}) &= \mathbf{U}_{22}(z^{-1}) \\ \mathbf{B}(z^{-1}) &= -\mathbf{U}_{21}(z^{-1})z\end{aligned}$$

Wie bereits erwähnt, sind $\mathbf{U}_{21}(z^{-1})$ und $\mathbf{U}_{22}(z^{-1})$ links koprim, und damit ist auch für $\mathbf{A}(z^{-1})$ und $\mathbf{B}(z^{-1})$ eine links koprimale Beschreibung gefunden.

Das oben erläuterte Verfahren wird auch bei Geering [41] ausführlich und in deutscher Sprache beschrieben.

It can be proven that the sub-matrices $\mathbf{U}_{21}(z^{-1})$ and $\mathbf{U}_{22}(z^{-1})$ are left coprime. Since $\mathbf{U}_{22}(z^{-1})$ is in addition to that non-singular, the following equation can be deduced from (G.2):

$$\mathbf{N}_{\mathbf{R}}(z^{-1})\mathbf{D}_{\mathbf{R}}(z^{-1})^{-1} = -\mathbf{U}_{22}(z^{-1})^{-1}\mathbf{U}_{21}(z^{-1})$$

A comparison with equation (8.10) results in

$$\begin{aligned}\mathbf{A}(z^{-1}) &= \mathbf{U}_{22}(z^{-1}) \\ \mathbf{B}(z^{-1}) &= -\mathbf{U}_{21}(z^{-1})z\end{aligned}$$

As already mentioned, $\mathbf{U}_{21}(z^{-1})$ and $\mathbf{U}_{22}(z^{-1})$ are left coprime and thus, a left coprime description is found for also $\mathbf{A}(z^{-1})$ and $\mathbf{B}(z^{-1})$.

The procedure described above is also described in detail by Geering [41] in German language.

可以证明，两个子矩阵 $\mathbf{U}_{21}(z^{-1})$ 和 $\mathbf{U}_{22}(z^{-1})$ 是左互素的。此外，由于 $\mathbf{U}_{22}(z^{-1})$ 是非奇异的，从而根据方程 (G.2) 可得：

$$\mathbf{N}_{\mathbf{R}}(z^{-1})\mathbf{D}_{\mathbf{R}}(z^{-1})^{-1} = -\mathbf{U}_{22}(z^{-1})^{-1}\mathbf{U}_{21}(z^{-1})$$

与方程 (8.10) 进行对比，可得：

$$\begin{aligned}\mathbf{A}(z^{-1}) &= \mathbf{U}_{22}(z^{-1}) \\ \mathbf{B}(z^{-1}) &= -\mathbf{U}_{21}(z^{-1})z\end{aligned}$$

如前所述， $\mathbf{U}_{21}(z^{-1})$ 和 $\mathbf{U}_{22}(z^{-1})$ 是左互素的，从而，可以发现 $\mathbf{A}(z^{-1})$ 和 $\mathbf{B}(z^{-1})$ 也是左互素的。

学者 Geering 使用德文详细地描述了上述过程 [41]。