

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Structure exploiting optimization methods for model predictive control

EMIL KLINTBERG



Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2017

Structure exploiting optimization methods for model predictive control

EMIL KLINTBERG

ISBN 978-91-7597-567-2

© EMIL KLINTBERG, 2017.

Doktorsavhandlingar vid Chalmers tekniska högskola

Ny serie nr 4248/2017

ISSN 0346-718X

Department of Signals and Systems

CHALMERS UNIVERSITY OF TECHNOLOGY

SE-412 96 Göteborg

Sweden

Telephone: +46 (0)31 – 772 1000

Email: kemil@chalmers.se

Typeset by the author using L^AT_EX.

Chalmers Reproservice
Göteborg, Sweden 2017

Abstract

This thesis considers optimization methods for Model Predictive Control (MPC). MPC is the preferred control technique in a growing set of applications due to its flexibility and to the natural way in which constraints can be incorporated in the control policy. Its applicability is, however, limited by the high computational burden associated with the solution of the underlying optimization problems. To alleviate this drawback we study structures in the MPC problems, which can enhance their solution.

The first topic of the thesis is numerical structures in matrices arising in gradient-based optimization methods for MPC. The idea is that due to the numerical structures, dense matrices can be approximated by sparse matrices to reduce the computational cost per iteration, and also for the overall solution of the MPC problem.

The second topic of the thesis is parallelizable optimization methods for multi-stage MPC. Multi-stage MPC is a popular framework used to increase the robustness of MPC schemes. One major drawback, however, is that the underlying optimization problems become very large. In this context, we consider parallel implementations of two different classes of optimization methods. First, we propose a parallelizable linear algebra for a primal-dual interior point method for two-stage MPC problems, i.e. for multi-stage MPC problems where the scenario tree is restricted to only branch in its root node. Secondly, we consider Newton's method to solve the Lagrange dual problem of multi-stage MPC problems. We show that the Hessian of the dual function is permutation similar to a block-tridiagonal matrix, propose a strategy for reducing the need for regularization, and reduce the cost of globalization strategies for problems with simple constraints and a diagonal cost.

The third topic of the thesis is optimization methods for solving distributed MPC problems in a distributed fashion using dual decomposition. Dual decomposition is commonly used with gradient-based methods to achieve a completely distributed method. In this thesis, however, we use dual decomposition together with Newton's method to achieve semi-distributed methods with a fast practical convergence. We study the occurrence of a singular dual Hessian and propose a constraint relaxation to prevent it. Additionally, we propose a distributed dual Newton strategy which can be viewed as a distributed primal-dual interior point method, and study the numerical structure of the dual Hessian for problems stemming from MPC deployed on multi-agent systems that are interacting via non-delayed couplings.

ABSTRACT

Keywords: Model Predictive Control, Multi-stage robust MPC, Distributed optimization, Optimization methods.

Acknowledgments

As my time as a PhD student is coming to an end, it is time to acknowledge some of the people that have offered support on the way.

First, I would like to express my gratitude towards Bo Egardt for giving me the opportunity to start as a PhD student in his group. In hindsight, this was a crossroad in my career where I believe that I chose the right path.

I have also been lucky to have many fruitful and enjoyable collaborations during my time at Chalmers. I have learned a lot about optimization from my supervisor Sebastien Gros. Moritz Diehl has hosted me twice, once in Leuven and once in Freiburg, where I have had the opportunity to collaborate with Attila Kozma and Dimitris Kouzoupis. At Chalmers I had the opportunity to collaborate with John Dahl and Anton Klintberg. I am very thankful for all these collaborations.

Last and most importantly, I would like to express my gratitude towards my family and friends.

Emil Klintberg
Göteborg, April 2017

List of publications

This thesis is based on the following seven appended papers.

Standard MPC

Paper 1

Emil Klintberg, Sebastien Gros. *Approximate inverses in preconditioned fast dual gradient methods for MPC*. Accepted to the 20th IFAC World Congress, July 2017.

Multi-stage MPC

Paper 2

Emil Klintberg, Sebastien Gros. *A Parallelizable Interior Point Method for Two-Stage Robust MPC*. IEEE Transactions on Control Systems Technology, vol. PP, 2017.

Paper 3

Emil Klintberg, John Dahl, Jonas Fredriksson, Sebastien Gros. *An improved dual Newton strategy for scenario-tree MPC*. IEEE Conference on Decision and Control, pp. 3675-3681, December 2016.

Paper 4

Emil Klintberg, Dimitris Kouzoupis, Moritz Diehl, Sebastien Gros. *A dual Newton strategy with fixed iteration complexity for multi-stage MPC*. Technical report. CPL Publication id: 248820. Chalmers University of Technology.

Distributed MPC

Paper 5

Attila Kozma, **Emil Klintberg**, Sebastien Gros, Moritz Diehl. *An Improved Distributed Dual Newton-CG Method for Convex Quadratic Programming Problems*. American Control Conference, pp. 2324-2329, July 2014.

Paper 6

Emil Klintberg, Sebastien Gros. *A Primal-Dual Newton Method for Distributed Quadratic Programming*. IEEE Conference on Decision and Control, pp. 5843-5848, December 2014.

Paper 7

Emil Klintberg, Sebastien Gros. *Numerical structure of the Hessian of the Lagrange dual function for a class of convex problems*. SIAM Journal on Control and Optimization, vol. 55, issue 1, pp. 574-593, 2017.

Other publications

In addition to the appended papers, the following papers are also written by the author of the thesis.

Emil Klintberg, Sebastien Gros. *An inexact interior point method for optimization of differential algebraic systems*. Computers and Chemical Engineering, vol. 92, pp. 163-171, 2016.

Anton Klintberg, **Emil Klintberg**, Björn Fridholm, Hannes Kuusisto, Torsten Wik. *Statistical modeling of OCV-curves for aged battery cells*. Accepted to the 20th IFAC World Congress, July 2017.

Notations and abbreviations

Notation	Meaning
\mathbb{R}	Set of real numbers
$[a, b]$	Interval of real numbers z such that $a \leq z \leq b$
(a, b)	Interval of real numbers z such that $a < z < b$
$\mathbb{R}^{m \times n}$	Set of real matrices of dimension $m \times n$
\mathbb{S}^n	Set of symmetric real matrices of dimension $n \times n$
\mathbb{S}_+^n	Set of symmetric positive semidefinite matrices of dimension $n \times n$
\mathbb{S}_{++}^n	Set of symmetric positive definite matrices of dimension $n \times n$
$A > 0$	The matrix A is positive definite
$A \geq 0$	The matrix A is positive semidefinite
$\text{diag}(a_1, \dots, a_n)$	Diagonal matrix with diagonal elements a_1, \dots, a_n
$\mathbf{1}$	Column vector of ones
$\frac{\partial f}{\partial z} \in \mathbb{R}^{n \times m}$	Jacobian of the function $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ with respect to the variable z
$\nabla f \in \mathbb{R}^{m \times n}$	Gradient of the function $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$
$\nabla^2 f \in \mathbb{S}^n$	Hessian of the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$
$\text{dom } f$	Domain of the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$

Abbreviation	Meaning
CQ	Constraint Qualification
CPU	Central Processing Unit
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
min	minimize
MPC	Model Predictive Control
QP	Quadratic Program
s.t.	subject to

Contents

Abstract	i
Acknowledgments	iii
List of publications	v
Notations and abbreviations	vii
Contents	ix

I Introductory chapters

1 Introduction	1
1.1 Background	1
1.2 Overview of optimization techniques for MPC	2
1.3 Robust MPC	3
1.4 Distributed MPC	4
1.5 Motivation and contributions	5
1.5.1 Standard MPC	5
1.5.2 Multi-stage MPC	5
1.5.3 Distributed optimization	6
1.6 Outline	7
2 Model Predictive Control	9
2.1 Standard MPC	9
2.2 Multi-stage MPC	12
2.2.1 Two-stage MPC	14
2.3 Distributed MPC via distributed optimization	15
3 Convex optimization	17
3.1 Convex sets and functions	17
3.2 Convex optimization problems	19

CONTENTS

3.2.1	Lagrangian duality	21
3.2.2	KKT optimality conditions	23
3.3	Dual decomposition	24
3.4	Optimization methods	26
3.4.1	Newton's method	26
3.4.2	Primal-dual interior point methods	29
3.4.3	Proximal gradient methods	31
4	Summary of included papers	33
5	Concluding remarks and future research directions	37
	References	39

II Included papers

Paper 1	Approximate inverses in preconditioned fast dual gradient methods for MPC	49
1	Introduction	49
2	Preliminaries and notations	50
2.1	Notations and terminology	50
2.2	Preconditioned dual proximal gradient methods	51
2.3	Model predictive control	53
2.4	Exponentially off-diagonally decaying matrices	54
3	Approximate inverses in dual proximal gradient methods for MPC	55
4	Numerical experiments	57
5	Conclusions and future work	62
	References	62
Paper 2	A Parallelizable Interior Point Method for Two-Stage Robust MPC	67
1	Introduction	67
1.1	Notations	69
2	Two-stage MPC	69
3	Primal-dual interior point methods	71
4	Efficient solution of the KKT system	73
4.1	Alternative KKT system	73
4.2	Parallel computation of the search direction Δw_k	74
4.3	Computation of the search direction $\Delta \lambda$	76
4.4	Calculating J	78
4.5	Algorithm	80

5	Numerical experiments	82
5.1	Example: Lateral control of an autonomous vehicle . . .	82
5.2	Evaluation of computational cost	86
6	Conclusions	89
	References	89
Paper 3 An improved dual Newton strategy for scenario-tree MPC		95
1	Introduction	95
1.1	Notations	97
2	Multi-stage MPC	97
3	Dual decomposition and second-order derivatives	100
3.1	Dual decomposition	100
3.2	Singularity of the dual Hessian	101
3.3	Block-tridiagonal dual Hessian	104
3.4	Algorithm	105
4	Numerical experiments	106
5	Conclusions	109
	References	110
Paper 4 A dual Newton strategy with fixed iteration complexity for multi-stage MPC		115
1	Introduction	115
2	Preliminaries	116
2.1	Multi-stage MPC	117
2.2	Dual decomposition	119
3	Calculating derivatives	121
3.1	Solving the subproblems	121
3.2	Calculating the dual Hessian	122
4	Efficient solution of the Newton system	124
4.1	Alternative Newton system	124
4.2	Regularization	125
4.3	Finding search directions $\Delta\mu_k$	126
4.4	Finding the search direction $\Delta\lambda$	127
4.5	Forming matrix J	128
5	A dual Newton method with a fixed complexity per iteration . .	129
5.1	Choice of step size	130
5.2	Elimination of redundant constraints	130
5.3	Warm-starting	131
6	Numerical experiments	131
7	Conclusions	134
	References	134

Paper 5	An Improved Distributed Dual Newton-CG Method for Convex Quadratic Programming Problems	141
1	Introduction	141
2	Dual Decomposition with Second-order Information	143
	2.1 Dual decomposition	143
	2.2 Singularity of the Dual Hessian	144
3	Non-smooth Newton Method in the Dual Space	146
	3.1 Calculation of the dual sensitivities	147
	3.2 Products between vectors and the dual Hessian	149
	3.3 Conjugate gradient method and line-search strategy	149
4	Numerical Example: Optimal Control of Connected Masses	150
5	Conclusions and Future Work	152
	References	153
Paper 6	A Primal-Dual Newton Method for Distributed Quadratic Programming	159
1	Introduction	159
2	Dual Decomposition with Second-order Information	161
	2.1 Dual decomposition	161
	2.2 Singularity of the dual Hessian	162
3	Relaxing Constraints and Computing Derivatives	163
	3.1 Constraint relaxation in a primal-dual framework	164
	3.2 Gradient and Hessian of the dual function	165
	3.3 Predictor	165
4	Algorithm	166
	4.1 Full convergence	166
	4.2 Path-following	166
	4.3 Fast path-following	168
5	Experimental Results	168
	5.1 Algorithm 10	168
	5.2 Fast path-following vs path-following	169
	5.3 Benefits of predictor step	169
6	Conclusions and Future Work	171
	References	171
Paper 7	Numerical structure of the Hessian of the Lagrange dual function for a class of convex problems	177
1	Introduction	177
	1.1 Notations and terminology	178
2	Problem formulation	179
3	Dual decomposition	180

3.1	Numerical structure of the dual Hessian	182
3.2	A distributed quasi-Newton strategy	188
4	Extension to inequality constrained problems	191
4.1	Dual decomposition using the relaxed dual function . . .	192
5	Numerical experiments	197
6	Conclusions and future research directions	198
	References	200

Part I

Introductory chapters

Chapter 1

Introduction

This chapter provides a brief background on Model Predictive Control (MPC). The concepts of robust MPC and distributed MPC are also introduced, and the main contributions of the thesis are summarized.

1.1 Background

In the early 1970s, it was observed that the performance of the complex and multivariable systems in process industry could be improved by using predictive, receding horizon controllers. Several control techniques were proposed where the arguably most well-known is the so-called Dynamic Matrix Control (DMC) developed at Shell Oil [1].

In DMC, a linear step response of the system was repeatedly simulated over a finite horizon and optimal control actions were calculated as the solution to a least-squares problem. Due to the simulation, unusual dynamic behaviors could be forecasted and complex control problems, which were unsolvable with PID controllers, could be handled. In its first version, the DMC was unable to incorporate system constraints in the control law, although this was later addressed by formulating the problem as a Quadratic Program (QP) [2]. The ability of handling system constraints was particularly important, since the economically most beneficial operating point of processing units is often located at the intersection of constraints [3].

The early receding horizon controllers, such as the DMC controller, are direct ancestors of modern MPC, which nowadays is the preferred control technique in a growing set of applications. In MPC, control actions are repeatedly calculated to optimize the predicted system behaviour, and the benefits over traditional control techniques are thus inherited from the DMC. In addition to its practical benefits, an important research effort has been dedicated to develop a rigorous theoretical foundation for MPC controllers [4].

However, the benefits compared to traditional control techniques come at the expense of solving an optimization problem at every sampling instant. This weakness initially limited the use of MPC to systems with long sampling times and to applications where expensive computing units could be afforded. Hardware improvements and an intensive research effort to design tailor-made optimization techniques have, however, alleviated these drawbacks and have made MPC feasible for increasingly faster systems and for some embedded applications. Despite the improvements, MPC is still intractable or very challenging in many demanding applications.

In the following sections, we provide an overview of optimization methods for MPC, and introduce two challenging problem classes; robust MPC and distributed MPC.

1.2 Overview of optimization techniques for MPC

In the early 2000s, so-called *explicit MPC* was proposed where the optimization problem is solved offline and the result is saved as a look-up table for online use [5]. This strategy quickly proved to be useful in some challenging applications and earned a significant attention in the literature [6–9]. However, the approach is in general limited to short horizons and linear systems with few states due to the excessive memory requirements otherwise. The significance of explicit MPC has therefore decreased with the development of customized online optimization methods, although some recent results have been directed towards reducing the memory requirements of the method [10].

Interior point methods are often preferred due to their robustness, practically consistent convergence and their polynomial (although not necessarily tight) bounds on the runtime. An early interior point method that exploited the specific structure of optimization problems arising from MPC was proposed in [11]. The method used a Riccati recursion to find the search directions and showed a linearly increasing computational cost with the horizon, compared to a cubically increasing cost for a naive implementation. After this, a variety of tailor-made interior point methods with decreasing runtimes [12–15] and computational complexities [16] have followed.

It has also been proposed to exploit the similarity between subsequent optimization problems in MPC by using an *active set method*. Specifically, based on the preceding problem a good initial guess for the active set can be provided, which often significantly improves the convergence of the method. This idea was successfully implemented in [17], and further exploited in [18, 19] where a parallelizable method was presented. However, active set methods lack the practically consistent convergence of interior point methods, and have no useful bound on the runtime.

A considerable part of the recent developments has been devoted to *first-order methods*. In particular the Alternating Direction Method of Multipliers [20] and the fast gradient method [21–25] are commonly identified as favourable for embedded MPC. Their popularity is stemming from their easy implementation and from the fact that, in contrast to interior point methods and active set methods, they have practically useful bounds on the runtime. However, their practical performance is heavily affected by the problem data. This is partly alleviated by the generalized fast gradient method which was presented in [26], and used in the context of MPC in [27, 28].

1.3 Robust MPC

An important property of a control system is its ability of handling plant variations and uncertainties, i.e. its so-called robustness. Being a feedback control technique, MPC has in its linear case some intrinsic robustness [29, 30]. Constraint satisfaction can, however, not be guaranteed if uncertainties are present in the system, and in the case of nonlinear systems the resulting control scheme may be non-robust [30]. As a remedy, several formulations of robust MPC schemes have been proposed to handle uncertainties in the employed model.

The early formulations of robust MPC used a min-max formulation [31], where the worst-case of several possible output trajectories was optimized, while the control inputs were feasible for all possible realizations of the uncertainty. It was, however, observed that the formulation could lead to infeasible optimization problems and was overly conservative, since it did not consider the fact that a new problem would be solved at the next sampling instant.

To account for the feedback, it was proposed to perform the optimization over feedback control laws instead of control inputs [32]. The resulting optimization problem was, however, infinite dimensional and thus difficult to solve. Several approximations were proposed to obtain a tractable formulation, including so-called *tube-based MPC* where the feedback law is approximated around a nominal trajectory [33, 34].

Another formulation which has recently gained attention in the literature is the so-called *multi-stage MPC* [35, 36]. Stemming from the field of stochastic programming, this formulation represents the uncertainty by a scenario tree, which in contrast to the early min-max formulations accounts for the feedback in the MPC scheme. A major drawback, however, is that the size of the underlying optimization problem grows exponentially with the length of the horizon. To alleviate this, several tailor-made optimization techniques have been proposed.

An early interior point method that exploits the specific structure of optimization problems originating from scenario tree problems was proposed in [37]. The method uses a Riccati recursion to calculate the search direction in normal equa-

tion form, and supports a certain degree of parallelism. Later, it has been proposed to use dual decomposition to achieve parallelizable methods that exploit the specific structure of multi-stage MPC problems. In [38], dual decomposition is used in conjunction with a first-order method whereas in [39] a Newton strategy is used in the dual space.

1.4 Distributed MPC

Many large scale systems are composed of interacting subsystems, which for various reasons are hard to control with a central controller. It is then natural to consider distributed control techniques.

Distributed MPC has for the last 10 years been a popular field of research, which has resulted in a vast collection of methods, where a significant part involves local MPC controllers of each subsystem. Methods of this kind are commonly categorized as *decentralized* [40, 41] if no communication between the subsystems is used, or in the case of communication as *cooperative* [42, 43] or *non-cooperative* [44–46] depending on the local objective functions.

Another strategy to achieve a distributed MPC controller, which we will focus on, is to rely on techniques from distributed optimization to solve the central MPC optimization problem. Although the large-scale optimization problem can be very challenging to solve in real-time, this formulation has the obvious advantage of not sacrificing control performance compared to central control. For this purpose, several methods have been proposed, mostly based on so-called *primal decomposition* or *dual decomposition*.

In primal decomposition, the optimization problem is split into a two level problem, where the local variables are manipulated locally, and the shared variables are manipulated via a so-called master problem. As a result, the iterates are feasible after every iteration, but the approach generally suffers from an increasing number of variables in the master problem for an increasing number of subsystems. Several methods using primal decomposition for MPC have been proposed in the literature, see e.g. [47–49].

In dual decomposition, the optimization problem is also split into a two level problem, where all primal variables are manipulated locally and the subproblems are coordinated via the dual variables which are adjusted in a master problem. In this case, the number of variables in the master problem is usually not increasing as rapidly with the number of subsystems, but the iterates are not feasible until convergence is achieved. This strategy was first proposed for distributed MPC in [50, 51], and has then been proposed in conjunction with a variety of methods [52–55] and for a variety of applications [56].

1.5 Motivation and contributions

Despite the immense algorithmic research, further improvements are still needed to extend the use of MPC to fast systems that in addition are embedded, large-scale, safety critical, uncertain or in some other way complicated. The difficulty lies in extending methods to have seemingly conflicting properties, as e.g. possessing a fast and consistent practical convergence besides being scalable or having tight bounds on the runtime.

The focus of this thesis is on optimization methods for MPC, where the main contributions can be divided into methods for standard MPC, multi-stage MPC, and for distributed optimization, which can be used to form distributed MPC schemes. The contribution for standard MPC aims at decreasing the computational cost of a first-order method and hence to facilitate the usage of MPC in embedded applications. The contributions for multi-stage MPC and distributed MPC aim at improving the scalability of interior-point methods and active set methods for these problem classes. The main contributions are outlined below.

1.5.1 Standard MPC

It is commonly proposed to use first-order methods to solve the Lagrange dual problem in the context of MPC. However, the methods are in general sensitive to ill-conditioning. In [28], this drawback is resolved at the expense of more computationally demanding iterations by using a preconditioned fast gradient method. In the paper:

Emil Klintberg, Sebastien Gros. *Approximate inverses in preconditioned fast dual gradient methods for MPC*. Accepted to the 20th IFAC World Congress, July 2017.

we alleviate the extra computational burden by approximating the dense preconditioner by a banded matrix. This is motivated by the observation that the preconditioner is an exponentially off-diagonally decaying matrix.

1.5.2 Multi-stage MPC

Interior point methods are often preferred in control applications due to their consistent practical convergence, and several interior point methods suitable for multi-stage MPC have been proposed in the literature [37, 57, 58]. In the paper:

Emil Klintberg, Sebastien Gros. *A Parallelizable Interior Point Method for Two-Stage Robust MPC*. IEEE Transactions on Control Systems Technology, vol. PP, 2017.

we propose a parallelizable interior point method for two-stage MPC, i.e. for multi-stage MPC where the scenario tree is restricted to only branch in its root node. In contrast to the methods proposed in [37, 57, 58], we introduce separate decision variables for every scenario in order to increase the degree of parallelism and to enable the use of well-established factorization techniques as building blocks. Note, however, that we focus on two-stage problems, whereas e.g. [37] focuses on multi-stage problems.

In [39], dual decomposition is used in conjunction with Newton's method to achieve a parallelizable method with a fast practical convergence. However, it is observed that the Hessian of the Lagrange dual function has an intricate sparsity pattern, implying that the solution of the Newton system is a severe computational bottleneck for large problems. In the paper:

Emil Klintberg, John Dahl, Jonas Fredriksson, Sebastien Gros. *An improved dual Newton strategy for scenario-tree MPC*. IEEE Conference on Decision and Control, pp. 3675-3681, December 2016.

we show that the Hessian is permutation similar to a block tridiagonal matrix, and consequently that the method has a linear computational cost in the number of scenarios. We also present an inexpensive elimination of redundant constraints in order to reduce the need for regularization.

The globalization can, however, be expensive in the dual Newton strategy, since many backtracking steps may be required at each iteration. This can be a significant drawback since every backtracking step involves solving QPs. In the paper:

Emil Klintberg, Dimitris Kouzoupis, Moritz Diehl, Sebastien Gros. *A dual Newton strategy with fixed iteration complexity for multi-stage MPC*. Technical report. CPL Publication id: 248820. Chalmers University of Technology.

we overcome this drawback for the practically important class of problems with a diagonal cost and simple bounds.

1.5.3 Distributed optimization

In [55], a method for solving distributed QPs based on dual decomposition was proposed. The method performs Newton steps on the dual variables where the Newton system is solved using a Conjugate Gradient (CG) method. In this context it has been observed that the presence of local inequality constraints can yield a singular Hessian of the Lagrange dual function. In the paper:

Attila Kozma, **Emil Klintberg**, Sebastien Gros, Moritz Diehl. *An Improved Distributed Dual Newton-CG Method for Convex Quadratic*

Programming Problems. American Control Conference, pp. 2324-2329, July 2014.

we study this effect and propose a constraint relaxation strategy to address the problem, and the resulting method can be viewed as an exterior point method.

However, it is well-known that interior point methods in general outperform exterior point methods, and distributed interior point methods based on dual decomposition have been proposed in e.g. [59, 60]. In the paper:

Emil Klintberg, Sebastien Gros. *A Primal-Dual Newton Method for Distributed Quadratic Programming*. IEEE Conference on Decision and Control, pp. 5843-5848, December 2014.

we propose a method for solving distributed QPs which closely resembles a path-following primal-dual interior point method. We show that local factorizations can be re-used to compute the dual Hessian and linear predictors that are used to enhance the convergence of the method.

In the paper:

Emil Klintberg, Sebastien Gros. *Numerical structure of the Hessian of the Lagrange dual function for a class of convex problems*. SIAM Journal on Control and Optimization, vol. 55, issue 1, pp. 574-593, 2017.

we show that the Hessian of the Lagrange dual function is numerically structured if a dual Newton method is used to solve a distributed MPC problem where the interactions between the subsystems are non-delayed. Based on this observation we propose to use a banded approximation of the Hessian in order to reduce the computational and communication burden of the method.

1.6 Outline

This thesis consists of two parts, where Part I provides a background and motivation for the second part. In Part II, seven papers are appended which serve as a core for the thesis. Part I comprises five chapters. Chapter 1 provides an introduction to the field and the thesis. In Chapter 2 we provide an introduction to MPC, multi-stage MPC and distributed MPC. In Chapter 3, we recall basic results in convex optimization and introduce some common optimization methods which are instrumental for Part II. In Chapter 4, a summary of the appended papers is provided. Chapter 5 finalizes Part I with some concluding remarks and future research directions.

Chapter 2

Model Predictive Control

In this chapter, we recall the basic principle of Model Predictive Control (MPC) and describe properties of multi-stage MPC and distributed optimization which are instrumental for Part II of the thesis.

2.1 Standard MPC

In this section we provide a brief introduction to Model Predictive Control (MPC); for a more detailed description see e.g. [61].

MPC is an optimization-based control technique, where the control inputs are selected based on the current state and predicted state trajectories. Specifically, at each sampling instant, a mathematical model of the system is simulated over a finite horizon and the sequence of control inputs over the horizon is optimized with respect to a performance criterion. The first element of the sequence is then applied to the real plant and a new open-loop optimal control problem is solved at the next sampling instant. Feedback is thus generated into the control scheme. The principle is illustrated in Figure 2.1.

To formulate the open-loop optimal control problem, we consider the following controlled dynamical system:

$$\dot{x}(t) = F(x(t), u(t)) \quad (2.1)$$

where $x(t) \in \mathbb{R}^{n_x}$ denotes the (differential) states of the system, $u(t) \in \mathbb{R}^{n_u}$ denotes the controls representing the actuation in the system, and the system dynamics is determined by $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$. In the context of control, it is often beneficial to represent the system (2.1) explicitly at discrete points, i.e.:

$$x_{i+1} = f(x_i, u_i), \quad x_0 = x(t_0) \quad (2.2)$$

where we have introduced the simplified notation $x_i = x(t_i)$ and $u_i = u(t_i)$, and the controls are assumed to be constant functions between the sampling instants,

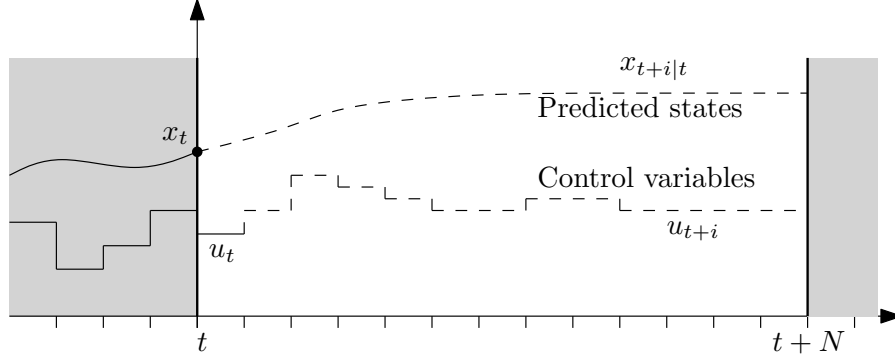


Figure 2.1: Illustration of the finite-horizon prediction. The notation $x_{t+i|t}$ refers to the predicted state at time instant $t + i$ based on information at time instant t .

i.e.:

$$u(t) = u_i, \quad t \in [t_i, t_{i+1}) \quad (2.3)$$

The state transition mapping $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ is often approximated, although it can be represented exactly as:

$$f(x_i, u_i) = x(t_i) + \int_{t_i}^{t_{i+1}} F(x(\tau), u(t_i)) d\tau \quad (2.4)$$

By using (2.2), the current state x_t and a sequence of control variables $\{u_{t+k}\}_{k=0}^{N-1}$, the state trajectory can be predicted over a horizon N as:

$$\begin{cases} x_{t+1|t} = f(x_t, u_t) \\ \vdots \\ x_{t+N|t} = f(x_{t+N-1|t}, u_{t+N-1}) \end{cases} \quad (2.5)$$

where the notation $x_{t+i|t}$ refers to the predicted states at time instant $t + i$ based on information at time instant t . In the following, for notation simplicity, we assume that $t = 0$ and omit to explicitly denote the dependence of the prediction on t , i.e. we use $x_i = x_{t+i|t}$.

Since the control variables can be manipulated, it is natural to aim at selecting them optimally. In this context, we measure optimality by the following performance criterion:

$$V(x, u) = \ell_N(x_N) + \sum_{i=0}^{N-1} \ell(x_i, u_i) \quad (2.6)$$

which takes lower values for favourable states and controls compared to less favourable ones, and where we have introduced the notations $x = [x_0^T, \dots, x_N^T]^T$ and $u = [u_0^T, \dots, u_{N-1}^T]^T$ for the collection of state and control variables over the horizon.

The open-loop optimal control problem, which in the following is referred to as the *MPC problem*, can then be formulated as:

$$\min_{x,u} V(x, u) \quad (2.7a)$$

$$\text{s.t. } x_{i+1} = f(x_i, u_i), \quad i = 0, \dots, N-1 \quad (2.7b)$$

$$x_0 = x(t_0) \quad (2.7c)$$

$$u_i \in \mathcal{U}, \quad i = 0, \dots, N-1 \quad (2.7d)$$

$$x_i \in \mathcal{X}, \quad i = 0, \dots, N-1 \quad (2.7e)$$

$$x_N \in \mathcal{X}_N \quad (2.7f)$$

where the stage constraint sets \mathcal{U} and \mathcal{X} are assumed to be non-empty and closed, and represent control and state constraints such as e.g. physical limitations and safety restrictions of the system. The terminal constraint set \mathcal{X}_N is instrumental to guarantee closed-loop stability [4]. Note that (2) is a *multiparametric program* with the initial state as the parameter. This, together with the highly structured nature of (2), is often heavily exploited in the design of optimization methods for MPC problems.

Let us now remark on a practically important class of MPC problems, often labeled as *linear MPC*. In this class, the dynamics are yielded by a linear system, the cost function is a separable convex quadratic function, and the *stage constraint sets* \mathcal{X} and \mathcal{U} and the *terminal constraint set* \mathcal{X}_N are polyhedral. This means that the MPC problem takes the form of the following structured QP, which has earned a lot of attention in the literature:

$$\min_{x,u} \sum_{i=0}^{N-1} \left(\frac{1}{2} \begin{bmatrix} x_i \\ u_i \end{bmatrix}^T \begin{bmatrix} Q_i & S_i \\ S_i^T & R_i \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} + \begin{bmatrix} q_i \\ r_i \end{bmatrix}^T \begin{bmatrix} x_i \\ u_i \end{bmatrix} \right) + \frac{1}{2} x_N^T P x_N + q_N^T x_N \quad (2.8a)$$

$$\text{s.t. } x_{i+1} = A x_i + B u_i, \quad i = 0, \dots, N-1 \quad (2.8b)$$

$$x_0 = x(t_0) \quad (2.8c)$$

$$C x_i + D u_i \leq d_i, \quad i = 0, \dots, N-1 \quad (2.8d)$$

$$D_N x_N \leq d_N \quad (2.8e)$$

A major challenge in MPC is to solve the underlying optimization problems sufficiently fast. To that end, a variety of customized optimization techniques have emerged to exploit the intrinsic structure of the MPC problems. The structure is mainly stemming from the following two observations.

First, because of the stage-wise structure of the objective function, it is separable with respect to the time stages and its Hessian is *block-diagonal*. Secondly, the constraints are a combination of stage-wise restrictions and dynamic constraints. This means that variables are only directly affected by other variables

at *neighboring time stages*. These structures can, as we shall see, be exploited to significantly reduce the memory requirements and computational times for solving (2.7) and (2.8).

2.2 Multi-stage MPC

In this section we provide a brief introduction to multi-stage MPC and formulate the underlying optimization problem; for a more comprehensive description see [62].

To formulate the MPC problem in this case, we consider a discrete-time, constrained system with uncertain parameters θ :

$$x_{i+1} = f(x_i, u_i, \theta) \quad (2.9a)$$

$$x_i \in \mathcal{X}(\theta), \quad u_i \in \mathcal{U}(\theta) \quad (2.9b)$$

To account for the uncertain parameters, we consider m_d realizations of θ at each time stage. The predicted state trajectory over a horizon N can then be described by a scenario tree, as depicted in Figure 2.2, where each branch corresponds to a specific realization of the uncertainty.

We define a scenario as a path from the root node to a leaf node of the scenario tree. This means that each scenario corresponds to a unique sequence of realizations of the uncertainty, and accordingly that the number of scenarios is growing exponentially with the length of the MPC horizon, yielding very large MPC problems. It is therefore often proposed to treat the uncertain parameters as constant after a certain period of time. This simplification is motivated by the fact that a new MPC problem is solved at the next sampling instant, implying that an accurate model of the far future is not critical. We denote the time period where the parameters can change as the *robust horizon* N_r in contrast to the prediction horizon N . Accordingly, we consider $M = m_d^{N_r}$ scenarios.

Let us consider separate state and control variables for each scenario k , i.e. we introduce $x_k = [x_{k,0}^T \dots x_{k,N}^T]^T \in \mathbb{R}^{\bar{n}_x}$, with $x_{k,i} \in \mathbb{R}^{n_x}$ and $u_k = [u_{k,0}^T \dots u_{k,N-1}^T]^T \in \mathbb{R}^{\bar{n}_u}$, with $u_{k,i} \in \mathbb{R}^{n_u}$ for $k = 1, \dots, M$. However, because the uncertainty cannot be anticipated, control actions are restricted to only depend on historical realizations of the uncertainty, such that the control variables of the scenarios are coupled at their shared nodes. More specifically, if the uncertainty realizations for scenario k and l are identical up to and including time stage i , their control inputs should be identical up to that time stage, i.e. $u_{k,j} = u_{l,j}, \forall j = 0, \dots, i$. This restriction is commonly denoted as *non-anticipativity constraints*.

By considering the following performance criterion for scenario k :

$$V_k(x_k, u_k) = \omega_k V(x_k, u_k) \quad (2.10)$$

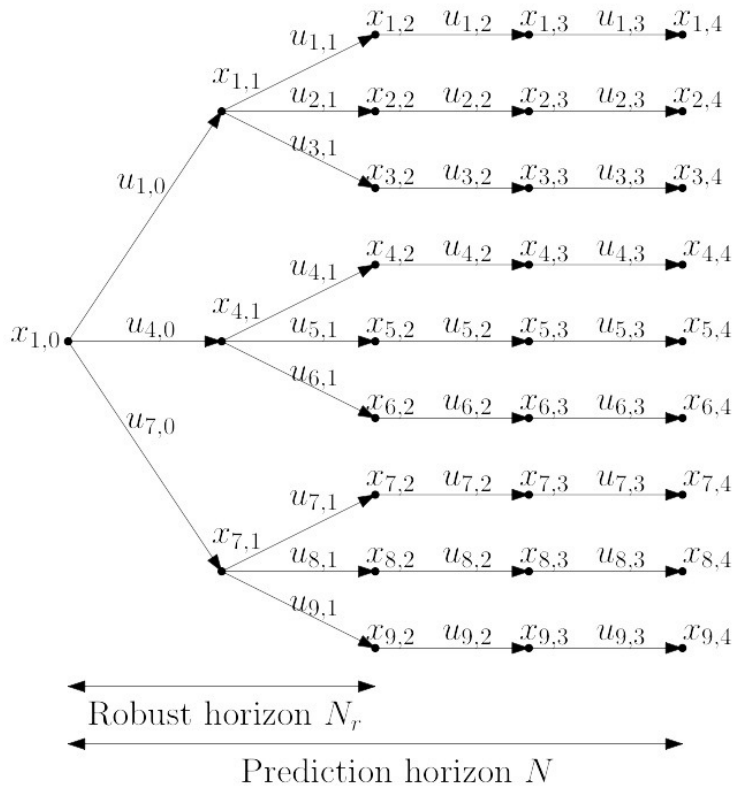


Figure 2.2: The evolution of the system represented as a scenario tree. In this example, $m_d = 3$, $N_r = 2$ and $M = 9$. For nodes and branches that are shared between multiple scenarios, the variable corresponding to the scenario with the lowest index is visualized in the tree.

where V is defined in (2.6) and ω_k denotes the probability of occurrence of scenario k , the multi-stage MPC problem can then be formulated as:

$$\min_{x,u} \sum_{k=1}^M V_k(x_k, u_k) \quad (2.11a)$$

$$\text{s.t. } x_{k,i+1} = f(x_{k,i}, u_{k,i}, \theta_{k,i}), \quad k = 1, \dots, M, \quad i = 0, \dots, N-1 \quad (2.11b)$$

$$x_{k,0} = x(t_0), \quad k = 1, \dots, M \quad (2.11c)$$

$$x_{k,i} \in \mathcal{X}, \quad k = 1, \dots, M, \quad i = 0, \dots, N-1 \quad (2.11d)$$

$$u_{k,i} \in \mathcal{U}, \quad k = 1, \dots, M, \quad i = 0, \dots, N-1 \quad (2.11e)$$

$$x_{k,N} \in \mathcal{X}_N, \quad k = 1, \dots, M \quad (2.11f)$$

$$u_{k,i} = u_{l,i}, \text{ if } x_{k,i} = x_{l,i}, \quad k, l = 1, \dots, M, \quad i = 0, \dots, N_r \quad (2.11g)$$

where we have introduced the notations $x = [x_1^T \dots x_M^T]^T$ and $u = [u_1^T \dots u_M^T]^T$ for the collection of variables over the scenarios, and $\theta_{k,i}$ denotes the k th realization of θ at time instant i .

Let us now note that (2.11) is composed of M ordinary MPC problems that are coupled via the non-anticipativity constraints. This observation can, as we shall see in Part II, be exploited in order to design highly parallelizable optimization methods for (2.11).

2.2.1 Two-stage MPC

A common way of reducing the size of (2.11) is to restrict the scenario tree to only branch in its root node, i.e. to choose $N_r = 1$. The multi-stage MPC problem is then reduced to a *two-stage MPC problem*:

$$\min_{x,u} \sum_{k=1}^{m_d} V_k(x_k, u_k) \quad (2.12a)$$

$$\text{s.t. } x_{k,i+1} = f(x_{k,i}, u_{k,i}, \theta_{k,i}), \quad k = 1, \dots, m_d, \quad i = 0, \dots, N-1 \quad (2.12b)$$

$$x_{k,0} = x(t_0), \quad k = 1, \dots, m_d \quad (2.12c)$$

$$x_{k,i} \in \mathcal{X}, \quad k = 1, \dots, m_d, \quad i = 0, \dots, N-1 \quad (2.12d)$$

$$u_{k,i} \in \mathcal{U}, \quad k = 1, \dots, m_d, \quad i = 0, \dots, N-1 \quad (2.12e)$$

$$x_{k,N} \in \mathcal{X}_N, \quad k = 1, \dots, m_d \quad (2.12f)$$

$$u_{k,0} = u_{l,0}, \quad k, l = 1, \dots, m_d \quad (2.12g)$$

Note that the only structural difference between the two-stage MPC problem and the multi-stage MPC problem lies in the non-anticipativity constraints. This formulation drastically reduces the number of scenarios, whereas a good performance can be obtained in practice [63, 64].

However, let us observe the following. The two-stage MPC problem does not model the fact that a new scenario tree, shifted in time, is considered at the next sampling instant. Hence, the second-stage control variables are less restricted compared to the first-stage control variables at the next sampling instant. This means that the controller could end up in *recursive infeasibility*. We are, however, not aware of an example where this occurs.

2.3 Distributed MPC via distributed optimization

In this section, we formulate a distributed MPC problem as a separable optimization problem.

Let us consider a large-scale system that consists of P subsystems:

$$x_{k,i+1} = f_k(x_{k,i}, u_{k,i}), \quad k = 1, \dots, P \quad (2.13)$$

which are interacting via the following *coupling constraints*:

$$g_i(x_{1,i}, u_{1,i}, \dots, x_{P,i}, u_{P,i}) = 0 \quad (2.14)$$

where $x_{k,i}$ and $u_{k,i}$ represent state and control variables for subsystem k at time instant i . Since the subsystems are often sparsely interconnected, the Jacobian matrix of g_i is typically sparse.

By using a performance criterion which is separable in the subsystems, the resulting MPC problem can be formulated as the following separable problem:

$$\min_{x,u} \sum_{k=1}^P V_k(x_k, u_k) \quad (2.15a)$$

$$\text{s.t. } x_{k,i+1} = f_k(x_{k,i}, u_{k,i}), \quad k = 1, \dots, P, \quad i = 0, \dots, N-1 \quad (2.15b)$$

$$g_i(x_{1,i}, u_{1,i}, \dots, x_{P,i}, u_{P,i}) = 0, \quad i = 0, \dots, N-1 \quad (2.15c)$$

$$x_{k,0} = x_k(t_0), \quad k = 1, \dots, P \quad (2.15d)$$

$$x_{k,i} \in \mathcal{X}_k, \quad k = 1, \dots, P, \quad i = 0, \dots, N-1 \quad (2.15e)$$

$$u_{k,i} \in \mathcal{U}_k, \quad k = 1, \dots, P, \quad i = 0, \dots, N-1 \quad (2.15f)$$

$$x_{k,N} \in \mathcal{X}_{k,N}, \quad k = 1, \dots, P \quad (2.15g)$$

where the sets \mathcal{X}_k and \mathcal{U}_k represent restrictions on the subsystem k , and $\mathcal{X}_{k,N}$ denotes the terminal constraint set of subsystem k . Additionally, we have introduced the notations $x_k = [x_{k,0}^T \dots x_{k,N}^T]^T$ and $u_k = [u_{k,0}^T \dots u_{k,N-1}^T]^T$ for the state and control variables for subsystem k , and the notations $x = [x_1^T \dots x_P^T]^T$ and $u = [u_1^T \dots u_P^T]^T$ for the collection of variables over all subproblems.

Note that (2.15) is composed of P ordinary MPC problems that are coupled via the coupling constraints (2.15c). For geographically distributed systems,

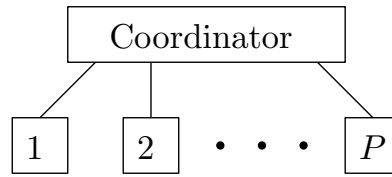


Figure 2.3: Schematic illustration of the coordinator and the distributed subproblems.

there is often only local knowledge of the subsystems (2.13), and distributed optimization techniques are therefore often desirable. In Part II, we consider Newton strategies for solving (2.15) in a distributed fashion. The problem is then decomposed into P subproblems, although in general the resulting methods require a coordinator and can thus be classified as partly distributed as depicted in Figure 2.3.

Chapter 3

Convex optimization

In this chapter, we recall results and methods from convex optimization which are instrumental for the results in Part II of the thesis.

3.1 Convex sets and functions

In this section we provide a brief introduction to convex sets and functions; for a more detailed description see e.g. [65]

A set \mathcal{S} is *convex* if for any points $z_1, z_2 \in \mathcal{S}$ and any scalar $\theta \in [0, 1]$, we have:

$$\theta z_1 + (1 - \theta)z_2 \in \mathcal{S} \quad (3.1)$$

Thus, the line segment between any two points in a convex set is also in the set. Figure 3.1 illustrates a convex and a non-convex set in \mathbb{R}^2 .

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is *convex* if $\text{dom } f$ is a convex set and for any points $z_1, z_2 \in \text{dom } f$ and any scalar $\theta \in [0, 1]$, we have:

$$f(\theta z_1 + (1 - \theta)z_2) \leq \theta f(z_1) + (1 - \theta)f(z_2) \quad (3.2)$$

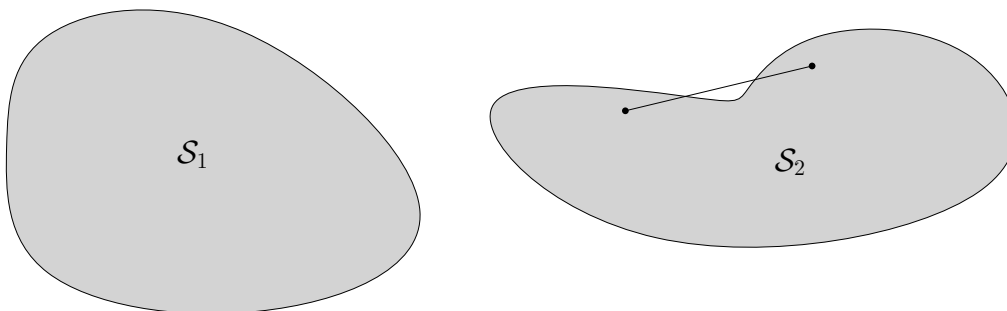


Figure 3.1: The set \mathcal{S}_1 is a convex set whereas the set \mathcal{S}_2 is a non-convex set.

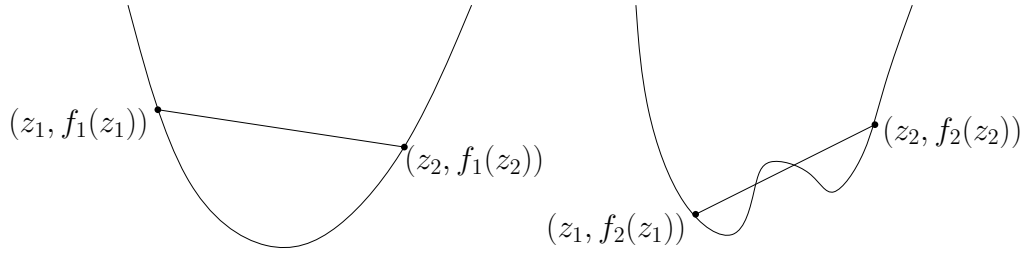


Figure 3.2: The function f_1 is a convex function whereas the function f_2 is a non-convex function.

Geometrically this means that a function is convex if the line segment between any points $(z_1, f(z_1))$ and $(z_2, f(z_2))$ lies above or on the graph of f as illustrated in Figure 3.2.

For differentiable functions, so-called *first-order conditions* can be established for the first-order Taylor expansion of convex functions. A differentiable function f is convex if and only if $\text{dom } f$ is convex and for any points $z_1, z_2 \in \text{dom } f$ the following inequality holds:

$$f(z_1) \geq f(z_2) + \nabla f(z_2)^T (z_1 - z_2) \quad (3.3)$$

This implies that the first-order Taylor expansion of a convex function is a global underestimator of the function. Additionally, we say that a function is *strictly convex* if (3.3) holds with strict inequality for $z_1 \neq z_2$. Moreover, we say that a differentiable function f is *strongly convex* with parameter $\sigma > 0$ if for any $z_1, z_2 \in \text{dom } f$ it holds that:

$$f(z_1) \geq f(z_2) + \nabla f(z_2)^T (z_1 - z_2) + \frac{\sigma}{2} \|z_1 - z_2\|^2 \quad (3.4)$$

Observe that strong convexity implies strict convexity but not vice versa.

Assuming that f is twice differentiable, so-called *second-order conditions* for convexity can be established. A twice differentiable function f is convex if and only if $\text{dom } f$ is convex and the Hessian of f is positive semidefinite, i.e.:

$$\nabla^2 f(z) \geq 0 \quad (3.5)$$

for all $z \in \text{dom } f$. Note that if (3.5) holds with strict inequality f is strictly convex, i.e. if $\text{dom } f$ is convex and the Hessian of f is positive definite. Similarly, a twice differentiable function f is strongly convex with parameter $\sigma > 0$ if $\text{dom } f$ is convex and it holds that:

$$\nabla^2 f(z) \geq \sigma I \quad (3.6)$$

for all $z \in \text{dom } f$. Hence, the minimum eigenvalue of $\nabla^2 f(z)$ is at least σ .

Finally, to exemplify the difference between strictly convex functions and strongly convex functions, let us consider the following scalar valued function:

$$f(z) = e^z \quad (3.7)$$

Note that f is strictly convex since $f''(z) > 0$, although it is not strongly convex since the second derivative can be arbitrary close to zero.

3.2 Convex optimization problems

In this section we provide a brief introduction to convex optimization problems. For a more detailed description see e.g. [65].

Throughout this thesis we consider various forms of the following *convex optimization problem*:

$$\min_z f(z) \quad (3.8a)$$

$$\text{s.t. } Az = b \quad (3.8b)$$

$$h_i(z) \leq 0, \quad i = 1, \dots, m \quad (3.8c)$$

where $z \in \mathbb{R}^n$ is a vector of *decision variables*, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex *objective function*, $A \in \mathbb{R}^{p \times n}$ and $b \in \mathbb{R}^p$, and the *inequality constraint functions* $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ are convex. To simplify notations, we use in the following the vector notation:

$$h(z) = \begin{bmatrix} h_1(z) \\ \vdots \\ h_m(z) \end{bmatrix} \quad (3.9)$$

A point z is called *feasible* if it satisfies the constraints, i.e. if $Az = b$ and $h(z) \leq 0$, and *strictly feasible* if it is feasible and satisfies the inequality constraints with strict inequality, i.e. if $Az = b$ and $h(z) < 0$. For notational convenience, we denote the feasible set as $\mathcal{Z} = \{z : Az = b, h(z) \leq 0\}$. The optimization problem (3.8) is called *feasible* if there exists at least one feasible point, and *infeasible* otherwise. The optimal value of (3.8) is denoted as f^* , where we use the convention of letting $f^* = \infty$ for infeasible problems and $f^* = -\infty$ for problems that are unbounded from below.

A finite value of f^* does, however, not guarantee that there exists an optimal solution z^* such that the optimal value is attained. To exemplify this, let us consider the following unconstrained problem:

$$\min_z e^z \quad (3.10)$$

In this case, it is clear that $f^* = 0$ although there exists no (bounded) z^* such that $e^{z^*} = 0$. The existence of a bounded z^* such that $f(z^*) = f^*$ is attained is

specified in an important theorem due to Weierstrass. Here, we state a simplified version of this theorem:

Proposition 1 *Let \mathcal{Z} be nonempty and closed, and let f be strongly convex. Then there exists a z^* such that:*

$$f(z^*) = \min_{z \in \mathcal{Z}} f(z) \quad (3.11)$$

Proof 1 *See e.g. [66].*

Note that this is a slight modification of Weierstrass' theorem where we have exchanged lower semi-continuity of f and the weak coercivity of f with respect to \mathcal{Z} to strong convexity of f . This is a more restrictive requirement since every convex function is also continuous and every strongly convex function is also coercive but not vice versa. In the rest of the thesis, we will assume that there exists a bounded z^* such that $f(z^*) = f^*$ is attained.

If the objective function f is differentiable and a bounded optimal point $z^* \in \mathcal{Z}$ exists, then the optimal point fulfills the following so-called *first-order optimality condition*:

$$\nabla f(z^*)^T (z - z^*) \geq 0, \quad \forall z \in \mathcal{Z} \quad (3.12)$$

Geometrically this means that if $\nabla f(z^*) \neq 0$, there is no feasible direction $z - z^*$ which is also a descent direction in f .

Finally, as a motivation for the next subsection, let us define a *relaxation* of the problem (3.8). Consider the following problem:

$$\min_{z \in \mathcal{Z}_R} f_R(z) \quad (3.13)$$

We say that (3.13) is a relaxation of (3.8) if $f_R : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function such that $f_R(z) \leq f(z)$, $\forall z \in \mathcal{Z}$, and $\mathcal{Z} \subseteq \mathcal{Z}_R$. If we denote the optimal value of (3.13) as f_R^* , we can state the following result for the pair of problems:

Theorem 1 *The following properties hold:*

1. $f_R^* \leq f^*$.
2. If (3.13) is infeasible, then so is (3.8).
3. If (3.13) has an optimal solution z_R^* such that $z_R^* \in \mathcal{Z}$ and $f_R(z_R^*) = f(z_R^*)$, then z_R^* is an optimal solution to (3.8) as well.

Proof 2 *See [66].*

As we shall see, this is an important result since many optimization algorithms seek a solution not to the considered problem but to a relaxation of the considered problem.

3.2.1 Lagrangian duality

In many problems, there is a subset of the constraints that makes the problem significantly harder to solve, so-called *complicating constraints*. The idea in Lagrangian duality is to augment the objective function with the constraints, and it can therefore often serve as a vital tool in the design of optimization methods for problems involving complicating constraints.

Let us denote a subset of the constraints (3.8b) and (3.8c) as:

$$\bar{A}z = \bar{b} \quad (3.14a)$$

$$\bar{h}(z) \leq 0 \quad (3.14b)$$

where $\bar{A} \in \mathbb{R}^{\bar{p} \times n}$, $\bar{b} \in \mathbb{R}^{\bar{p}}$ for $\bar{p} \leq p$, and $\bar{h} : \mathbb{R}^n \rightarrow \mathbb{R}^{\bar{m}}$ for $\bar{m} \leq m$. Additionally, we introduce the notation \mathcal{Z}^C for the feasible set defined by the constraints in (3.8b) and (3.8c) that are not included in (3.14).

By introducing the dual variables $\mu \in \mathbb{R}^{\bar{p}}$ and $\lambda \in \mathbb{R}^{\bar{m}}$ corresponding to the constraints (3.14a) and (3.14b) respectively, we define the (partial) *Lagrange function* as:

$$\mathcal{L}(z, \mu, \lambda) = f(z) + \mu^T (\bar{A}z - \bar{b}) + \lambda^T \bar{h}(z) \quad (3.15)$$

Hence, the Lagrange function is defined by the objective function augmented with a weighted sum of constraint functions. The (negated) Lagrange dual function is then defined as:

$$d(\mu, \lambda) = - \min_{z \in \mathcal{Z}^C} \mathcal{L}(z, \mu, \lambda) \quad (3.16)$$

Under some (rather weak) conditions, the Lagrange dual function is differentiable. This is summarized in the following proposition.

Proposition 2 *Assume that $\bar{h}(z)$ is continuous and convex and that (3.8) is feasible. Then the dual function $d(\mu, \lambda)$ is differentiable with gradient:*

$$\nabla d(\mu, \lambda) = - \begin{bmatrix} \bar{A}z^*(\mu, \lambda) - \bar{b} \\ \bar{h}(z^*(\mu, \lambda)) \end{bmatrix} \quad (3.17)$$

where $z^*(\mu, \lambda) = \arg \min_{z \in \mathcal{Z}^C} \mathcal{L}(z, \mu, \lambda)$.

Proof 3 *See e.g. [67].*

We usually restrict the problem data further, and can then obtain a continuously differentiable dual function:

Proposition 3 *Assume that $\bar{h}(z) = Cz - d$, that (3.8) is feasible and that f is continuous and strongly convex. Then the dual function $d(\mu, \lambda)$ has a Lipschitz continuous gradient.*

Proof 4 See e.g. [68].

One important property of the Lagrange dual function is that it yields a lower bound on the optimal value f^* of (3.8). More specifically, for any μ and $\lambda \geq 0$ and any feasible $\tilde{z} \in \mathcal{Z}$, we have that:

$$\begin{aligned} -d(\mu, \lambda) &= \min_{z \in \mathcal{Z}^c} \left(f(z) + \mu^T (\bar{A}z - \bar{b}) + \lambda^T h(z) \right) \\ &\leq f(\tilde{z}) + \mu^T (\bar{A}\tilde{z} - \bar{b}) + \lambda^T h(\tilde{z}) \leq f(\tilde{z}) \end{aligned} \quad (3.18)$$

where the second inequality is stemming from the fact that:

$$\mu^T (\bar{A}\tilde{z} - \bar{b}) + \lambda^T h(\tilde{z}) \leq 0 \quad (3.19)$$

Consequently, the Lagrange dual function is a relaxation of (3.8) whenever $\lambda \geq 0$, and (3.16) is then commonly referred to as a *Lagrangian relaxation*. The best possible lower bound $-d^*$ that can be obtained from the Lagrange dual function is given by the following *Lagrange dual problem*:

$$\min_{\mu, \lambda} d(\mu, \lambda) \quad (3.20a)$$

$$\text{s.t. } \lambda \geq 0 \quad (3.20b)$$

where we refer to the difference between the optimal value f^* and $-d^*$ as the *optimal duality gap*. If the optimal duality gap is zero, i.e. if $f^* = -d^*$, we say that *strong duality* holds. Strong duality is an important property since it implies that the primal solution can be recovered from the dual solution:

Proposition 4 Suppose that strong duality holds and that μ^* and λ^* solve the Lagrange dual problem. Then, z^* is a primal optimal solution if and only if z^* is feasible in (3.8) and:

$$z^* \in \arg \min_{z \in \mathcal{Z}^c} \mathcal{L}(z, \mu^*, \lambda^*) \quad (3.21)$$

Proof 5 See [66].

As we shall see, this is an important result since the Lagrange dual problem and (3.21) often are significantly easier to solve compared to directly solving the primal problem.

There are several so-called *constraint qualifications* (CQs) to ensure that strong duality holds. In this thesis, we consider the following two commonly used CQs.

Definition 1 The system of constraints describing the feasible region \mathcal{Z} is said to satisfy Slater's constraint qualification if A has full row rank and there exists a point $z \in \mathcal{Z}$ such that $h(z) < 0$.

Definition 2 *The Linear Independence Constraint Qualification (LICQ) is said to be satisfied at a point $z \in \mathcal{Z}$ if the gradients $\nabla h_i(z)$, $i = 1, \dots, m$ together with A^T are linearly independent.*

Let us now assume that strong duality holds. Under this assumption, we can conclude that:

$$\begin{aligned}
f(z^*) &= -d(\mu^*, \lambda^*) = \min_{z \in \mathcal{Z}^C} \mathcal{L}(z, \mu^*, \lambda^*) \\
&= \min_{z \in \mathcal{Z}^C} \left(f(z) + (\mu^*)^T (\bar{A}z - \bar{b}) + \sum_{i=1}^{\bar{m}} \lambda_i^* \bar{h}_i(z) \right) \\
&\leq f(z^*) + (\mu^*)^T (\bar{A}z^* - \bar{b}) + \sum_{i=1}^{\bar{m}} \lambda_i^* \bar{h}_i(z^*) \\
&\leq f(z^*)
\end{aligned} \tag{3.22}$$

where the first equality is due to strong duality, the second equality follows from the definition of the dual function, and the last inequality holds since z^* is primal feasible, i.e. since $\bar{A}z^* - \bar{b} = 0$ and $\bar{h}(z^*) \leq 0$. Accordingly, we can conclude that the inequalities in (3.22) can be replaced by equalities, and hence that:

$$\sum_{i=1}^{\bar{m}} \lambda_i^* \bar{h}_i(z^*) = 0 \tag{3.23}$$

Since every term in the sum is non-negative, it follows that:

$$\lambda_i^* \bar{h}_i(z^*) = 0, \quad i = 1, \dots, \bar{m} \tag{3.24}$$

This condition is referred to as *complementary slackness*, and holds for any primal optimal solution z^* and dual optimal solution μ^* and λ^* when strong duality holds.

3.2.2 KKT optimality conditions

It is often of importance to have easily verifiable criteria to check whether a certain point is optimal or not. In this subsection we use the results from Lagrangian duality to develop the classical Karush-Kuhn-Tucker (KKT) conditions.

Let us assume that f and h in (3.8) are differentiable and that all constraints are dualized, i.e. that $\mathcal{Z}^C = \mathbb{R}^n$. Now, recall that an optimal point z^* minimizes $\mathcal{L}(z, \mu^*, \lambda^*)$ over $z \in \mathbb{R}^n$. This means that the gradient of the Lagrange function must vanish at the minimizer z^* , i.e. at any optimal point z^* the following must hold:

$$\nabla f(z^*) + A^T \mu^* + \sum_{i=1}^m \lambda_i^* \nabla h_i(z^*) = 0 \tag{3.25}$$

Additionally, assuming that strong duality holds, we recall that complementary slackness must be satisfied for an optimal primal-dual solution $z^* \in \mathcal{Z}$, μ^* and $\lambda^* \geq 0$. This implies that any pair of primal and dual optimal points must satisfy:

$$\nabla f(z^*) + A^T \mu^* + \nabla h(z) \lambda^* = 0 \quad (3.26a)$$

$$Az^* - b = 0 \quad (3.26b)$$

$$h_i(z^*) \lambda_i^* = 0, \quad i = 1, \dots, p \quad (3.26c)$$

$$h_i(z^*) \leq 0, \quad i = 1, \dots, p \quad (3.26d)$$

$$\lambda_i^* \geq 0, \quad i = 1, \dots, p \quad (3.26e)$$

The conditions (3.26) are called the *KKT conditions*. Following from the discussion in Section 3.2.1, the KKT conditions are clearly sufficient and necessary conditions for optimality provided that the problem is convex and strong duality holds. However, without strong duality, i.e. without a constraint qualification, the KKT conditions remain only sufficient for convex problems. To exemplify this, let us consider the following convex problem:

$$\min_z z_1 \quad (3.27a)$$

$$\text{s.t. } z_1^2 + z_2 \leq 0 \quad (3.27b)$$

$$-z_2 \leq 0 \quad (3.27c)$$

Note that there is only one feasible point, i.e. $z = 0$, and Slater's CQ is hence not satisfied. Additionally, at this point both inequality constraints are active and LICQ is violated. The KKT conditions are:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix} \lambda = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.28a)$$

$$\lambda \geq 0 \quad (3.28b)$$

which clearly are unsolvable. Hence, this shows a situation in which the KKT conditions are not fulfilled at the optimal solution, and thus demonstrates the importance of constraint qualifications.

3.3 Dual decomposition

In this section we recall dual decomposition on which some of the results in Part II are based.

Let us consider the following separable convex optimization problem:

$$\min_z \sum_{k=1}^N f_k(z_k) \quad (3.29a)$$

$$\text{s.t.} \quad \sum_{k=1}^N A_k z_k = b \quad (3.29b)$$

$$z_k \in \mathcal{Z}_k, \quad k = 1, \dots, N \quad (3.29c)$$

where $z_k \in \mathbb{R}^{n_k}$ are vectors of local decision variables, $f_k : \mathbb{R}^{n_k} \rightarrow \mathbb{R}$ are convex functions, $A_k \in \mathbb{R}^{p \times n_k}$, and the sets \mathcal{Z}_k are convex and fulfill a constraint qualification such that strong duality holds. Additionally, for notational simplicity, we have introduced the notation $z = [z_1^T \dots z_N^T]^T$ and use $\mathcal{Z} = \mathcal{Z}_1 \times \dots \times \mathcal{Z}_N$.

Note that the equality constraint (3.29b) acts as a complicating constraint, since it involves all decision variables in contrast to the local constraints (3.29c). To decompose the problem (3.29), we introduce the dual variables $\lambda \in \mathbb{R}^p$ corresponding to the complicating constraint (3.29b) and define the Lagrange function as:

$$\mathcal{L}(z, \lambda) = \sum_{k=1}^N f_k(z_k) + \lambda^T \left(\sum_{k=1}^N A_k z_k - b \right) \quad (3.30)$$

where we note that $\mathcal{L}(z, \lambda)$ is separable in z , i.e. it can be expressed as:

$$\mathcal{L}(z, \lambda) = \sum_{k=1}^N \mathcal{L}_k(z_k, \lambda) \quad (3.31)$$

where:

$$\mathcal{L}_k(z_k, \lambda) = f_k(z_k) + \lambda^T \left(A_k z_k - \frac{1}{N} b \right) \quad (3.32)$$

The Lagrange dual function $d(\lambda) = -\min_{z \in \mathcal{Z}} \mathcal{L}(z, \lambda)$ can thus be evaluated in parallel as:

$$d(\lambda) = -\sum_{k=1}^N \min_{z_k \in \mathcal{Z}_k} \mathcal{L}_k(z_k, \lambda) \quad (3.33)$$

where we have interchanged the order of the minimization and summation. Assuming that the Lagrange dual function is differentiable, according to Proposition 2, its gradient can be calculated as:

$$\nabla d(\lambda) = -\sum_{k=1}^N A_k z_k^*(\lambda) + b \quad (3.34)$$

where $z_k^*(\lambda) = \arg \min_{z_k \in \mathcal{Z}_k} \mathcal{L}_k(z_k, \lambda)$ can be calculated in parallel. As a result, the Lagrange dual problem can be decomposed and solved in a parallel fashion using e.g. gradient-based optimization methods.

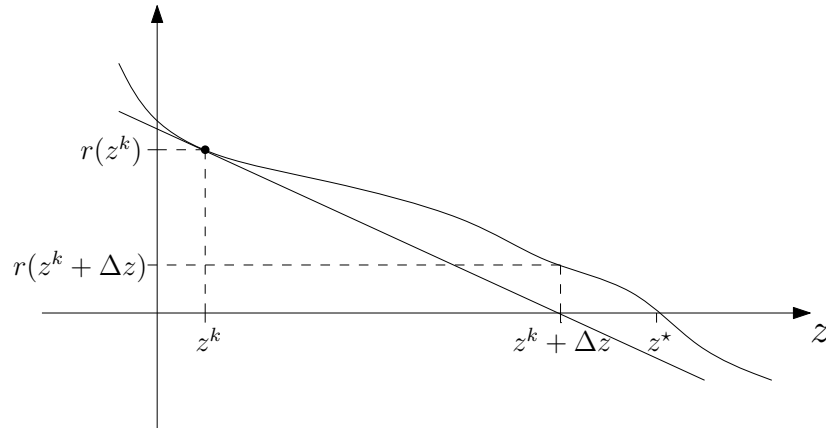


Figure 3.3: Illustration of Newton's method.

3.4 Optimization methods

In this section, we recall the optimization methods on which the results in Part II are based.

3.4.1 Newton's method

The essence of many optimization routines is to find a solution to a nonlinear system of equations, i.e. the problem of finding $z^* \in \mathbb{R}^n$ such that:

$$r(z^*) = 0 \quad (3.35)$$

where $r : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a smooth vector valued function, i.e.:

$$r(z) = \begin{bmatrix} r_1(z) \\ \vdots \\ r_n(z) \end{bmatrix} \quad (3.36)$$

In practice, one of the most consistently efficient methods for finding a solution z^* to a system of the form (3.35) is Newton's method. In Newton's method a solution is searched by iteratively linearizing r and performing steps that fulfill the linearizations as illustrated in Figure 3.3. The core of the method is thus to solve the following so-called *Newton system* in order to find the *Newton direction* Δz :

$$\frac{\partial r(z^k)}{\partial z} \Delta z = -r(z^k) \quad (3.37)$$

The variables are then updated in the direction of the Newton direction according to:

$$z^{k+1} = z^k + t\Delta z \quad (3.38)$$

where $t \in (0, 1]$ is an appropriately chosen *step size* to enforce convergence and speed of the iteration.

Provided that there exists a solution z^* to the system (3.35) and that $\frac{\partial r(z^k)}{\partial z}$ is full rank, Newton's method provides a contractive iteration for finding z^* if we provide an initial guess sufficiently close to z^* . Here, we omit to further specify what *close enough* means, and only state that it depends on how nonlinear r is, for details see e.g. [69].

To exemplify the usefulness of Newton's method in optimization, let us consider the following unconstrained convex problem:

$$\min_z f(z) \quad (3.39)$$

Due to the absence of constraints, the first-order optimality condition simplifies to:

$$\nabla f(z^*) = 0 \quad (3.40)$$

Hence, the solution z^* to (3.39) is also solving the nonlinear system of equations (3.40). Applying Newton's method on the system (3.40) results in the following Newton system:

$$\nabla^2 f(z^k) \Delta z = -\nabla f(z^k) \quad (3.41)$$

A useful interpretation of this system is that the Newton direction is the minimizer of a quadratic model of f around the current point z^k . Specifically, Δz is the solution to the following unconstrained Quadratic Program (QP):

$$\min_z f(z^k) + \nabla f(z^k)^T z + \frac{1}{2} z^T \nabla^2 f(z^k) z \quad (3.42)$$

This interpretation often provides insights into when Newton's method may be successful in solving an unconstrained optimization problem and when it may fail. In cases where f can only be inaccurately modeled by a quadratic function at z^k , the selection of the step size t is instrumental.

In the context of unconstrained optimization, the step size t is commonly chosen by a *backtracking line-search*. This means that the step size is initialized at a relatively large value and iteratively reduced until a condition measuring progress is fulfilled. In this thesis, we use the *Armijo condition* in the backtracking line-search. This means that for every search direction Δz , we initialize the step size at $t = 1$ and backtrack according to:

$$t := \beta t \quad (3.43)$$

until the Armijo condition is fulfilled, i.e. until:

$$f(z^k + t\Delta z) \leq f(z^k) + \alpha t \nabla f(z^k)^T \Delta z \quad (3.44)$$

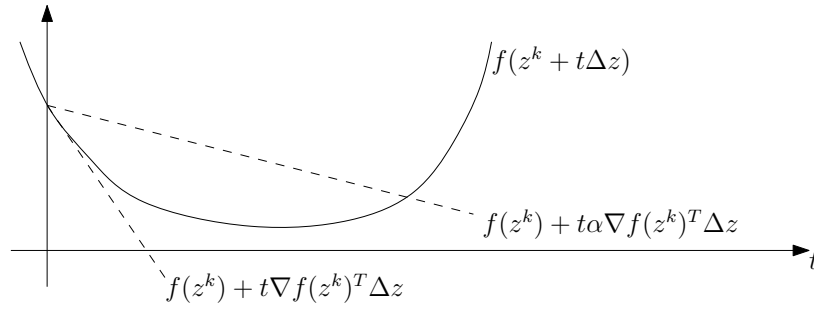


Figure 3.4: Illustration of the Armijo condition.

for some constants $\alpha, \beta \in (0, 1)$. To gain some insights into the choice of the parameters α and β , let us recall (3.3) and note that if f is nonlinear (3.44) would never be fulfilled for $\alpha = 1$, since a linearization is a global underestimator of a convex function. By reducing α , however, the slope is reduced and the linear function will always intersect f . Evidently, for convex problems it is always possible to find a step size $t \in (0, 1]$ such that (34) is fulfilled. Following from the discussion, we also note that large values of α and β correspond to a crude line-search whereas small values correspond to a careful search. An illustration of the Armijo condition is provided in Figure 3.4.

Note that for a twice differentiable convex function f , $\nabla^2 f(z^k)$ is only positive semidefinite, i.e. there could be directions in which f lacks curvature. Such cases are problematic since a rank deficient $\nabla^2 f(z)$ leads to an inconsistent Newton system. This problem is in general addressed by adding a regularization matrix $F \in \mathbb{S}_+^n$ to $\nabla^2 f(z^k)$ to ensure that:

$$\nabla^2 f(z^k) + F > 0 \tag{3.45}$$

One common choice is the so-called *Levenberg-Marquardt regularization* with $F = \delta I$.

Algorithm 1 Newton’s method for unconstrained minimization

- 1: **for** $k \geq 0$ **do**
 - 2: Calculate F
 - 3: Solve $(\nabla^2 f(z^k) + F)\Delta z = -\nabla f(z^k)$
 - 4: Perform line-search to find an appropriate t
 - 5: $z^{k+1} = z^k + t\Delta z$
 - 6: **end for**
-

A basic Newton method for unconstrained convex minimization is provided in Algorithm 1. The computational bottleneck in Newton’s method typically lies in solving the Newton system. This means that any structure that can enhance its solution should be exploited.

If the function f lacks enough differentiability or if the Newton system despite structure exploitation is too costly to solve, $\nabla^2 f(z^k)$ is typically approximated in the Newton system by a matrix that is cheaper to compute. The resulting method is then referred to as a *quasi-Newton method*.

3.4.2 Primal-dual interior point methods

Due to their robustness and consistent practical performance, interior point methods are often the preferred choice to solve problems of the form (3.8). In this subsection, we recall the basics of *primal-dual interior point methods*.

In primal-dual interior point methods, Newton's method is used to solve a modified version of the KKT conditions. To motivate the modification, we recall that Newton's method offers a practically efficient way of solving a system of smooth nonlinear equations. This means that the non-smooth complementary relationship between $h(z^*)$ and λ^* in the KKT conditions is problematic. Thus, to smoothen the KKT conditions, the complementary slackness condition is relaxed as:

$$h_i(z)\lambda_i = \tau, \quad i = 1, \dots, p \quad (3.46)$$

where $\tau \geq 0$ is referred to as the *barrier parameter*.

Additionally, to facilitate the enforcing of the inequality constraints, slack variables $s \in \mathbb{R}^p$ are typically introduced, and the inequality constraints are reformulated as:

$$h(z) + s = 0 \quad (3.47a)$$

$$s \geq 0 \quad (3.47b)$$

The resulting *interior point KKT conditions* are given by:

$$0 = \nabla \mathcal{L}(z, s, \mu, \lambda) = \nabla f(z) + A^T \mu + \nabla h(z)\lambda \quad (3.48a)$$

$$0 = Az - b \quad (3.48b)$$

$$0 = h(z) + s \quad (3.48c)$$

$$0 = S\lambda - \tau \mathbf{1} \quad (3.48d)$$

$$\lambda \geq 0, \quad s \geq 0 \quad (3.48e)$$

where we have introduced the notation $S = \text{diag}(s)$ and $\mathbf{1} \in \mathbb{R}^p$ represents a vector of ones. Note that by decreasing τ , (3.48) can be an arbitrary good approximation of the KKT conditions.

By fixing τ and applying Newton's method on the system (3.48a)-(3.48d) in the variables z, μ, λ and s we obtain the so-called *interior-point KKT system* or

primal-dual system:

$$\begin{bmatrix} \nabla_{zz}^2 \mathcal{L}^k & A^T & \nabla h(z^k) \\ A & & \\ \nabla h(z^k)^T & S^k & \Lambda^k \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \mu \\ \Delta \lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} \nabla_z \mathcal{L}^k \\ Az^k - b \\ h(z^k) + s^k \\ \Lambda^k s^k - \tau \mathbf{1} \end{bmatrix} \quad (3.49)$$

where we have introduced the notations $\mathcal{L}^k = \mathcal{L}(z^k, s^k, \mu^k, \lambda^k)$ and $\Lambda^k = \text{diag}(\lambda^k)$. The primal-dual variables are then updated according to:

$$z^{k+1} = z^k + t\Delta z \quad (3.50a)$$

$$s^{k+1} = s^k + t\Delta s \quad (3.50b)$$

$$\mu^{k+1} = \mu^k + t\Delta \mu \quad (3.50c)$$

$$\lambda^{k+1} = \lambda^k + t\Delta \lambda \quad (3.50d)$$

where $t \in (0, 1]$ is chosen to enforce $s^{k+1} > 0$ and $\lambda^{k+1} > 0$. A basic primal-dual interior point method is summarized in Algorithm 2, where r^k denotes the coefficient vector of (3.49) and $\epsilon > 0$ serves as a tuning parameter. State-of-the-art primal-dual interior point methods are, however, somewhat more subtle [69].

Algorithm 2 Primal-dual interior point method

- 1: **for** $k \geq 0$ **do**
 - 2: Solve interior point KKT system
 - 3: Find appropriate step size t
 - 4: Update variables
 - 5: **if** $\|r^k\| \leq \epsilon$ **then**
 - 6: $\tau := \alpha\tau$
 - 7: **end if**
 - 8: **end for**
-

The computational bottleneck in any modern interior point method lies in the solution of the interior point KKT system, which motivates the development of efficient techniques for factorizing its coefficient matrix. Finally, we recall two reformulations of (3.49) that are commonly used to reduce the computational cost of its solution.

By eliminating Δs and $\Delta \lambda$ using the bottom two block rows of (3.49), the so-called *augmented system* is obtained:

$$\begin{bmatrix} \Phi^k & A^T \\ A & \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta \mu \end{bmatrix} = - \begin{bmatrix} r_d^k \\ Az^k - b \end{bmatrix} \quad (3.51)$$

where we have introduced:

$$\Phi^k = \nabla^2 \mathcal{L}^k + \nabla h(z^k)^T (S^k)^{-1} \Lambda^k \nabla h(z^k) \quad (3.52a)$$

$$r_d^k = \nabla \mathcal{L}^k + A^T \mu^k + \nabla h(z^k)^T (S^k)^{-1} \Lambda^k (h(z^k) + s^k) \quad (3.52b)$$

Due to the symmetric structure of the coefficient matrix in (3.51), a sparse symmetric indefinite factorization routine can be used to solve the augmented system. This formulation is often attractive for problems with few equality constraints.

By forming the Schur complement of the coefficient matrix of (3.51), the so-called *normal equation form* is obtained:

$$Y^k \Delta \mu = r_n^k \quad (3.53a)$$

$$\Phi^k \Delta z = -r_d - A^T \Delta \mu \quad (3.53b)$$

where we have introduced the notations:

$$Y^k = A(\Phi^k)^{-1}A^T \quad (3.54a)$$

$$r_n^k = Az^k - b - A(\Phi^k)^{-1}r_d^k \quad (3.54b)$$

The normal equation form is often preferred if Y^k is not too dense compared to Φ^k , or if there are many equality constraints resulting in a large augmented system. In control applications, the coefficient matrices in (3.53) can typically be factorized by sparse Cholesky factorization routines.

Although very similar search directions are generated, primal-dual interior point methods often outperform classical *barrier methods*, especially when high accuracy is required [65]. This is partly stemming from the fact that the iterates generated by a primal-dual interior point method are in general not feasible. However, the primal-dual inequality constraints are fulfilled for every iteration, i.e.:

$$h(z^k) \leq 0 \quad (3.55a)$$

$$\lambda^k \geq 0 \quad (3.55b)$$

which suggests that *interior point* is not a misleading label for this class of methods.

3.4.3 Proximal gradient methods

Let us consider the following composite optimization problem:

$$\min_z f(z) + g(z) \quad (3.56)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex and differentiable, ∇f is Lipschitz continuous with constant L , and $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is convex and lower semi-continuous. To illustrate the generality of (3.56), we note that it can capture the convex optimization problem (3.8) if g is chosen as the indicator function of the feasible set, i.e. if:

$$I_{\mathcal{Z}}(z) = \begin{cases} 0, & z \in \mathcal{Z} \\ \infty, & \text{otherwise} \end{cases} \quad (3.57)$$

where $\mathcal{Z} = \{z : Az = b, h(z) \leq 0\}$.

Problems of the form (3.56) are commonly tackled via proximal gradient methods, see e.g. [70]. One iteration of the proximal gradient method deployed on (3.56) consists of a forward step based on the gradient of f , and then a backward step based on the proximity operator of g . The resulting forward-backward step is therefore given by:

$$z^{k+1} = \text{prox}_{\frac{1}{L}g} \left(z^k - \frac{1}{L} \nabla f(z^k) \right) \quad (3.58)$$

where the proximity operator is defined as:

$$\text{prox}_f(z) = \arg \min_{y \in \mathbb{R}^n} f(y) + \frac{1}{2} \|y - z\|_2^2 \quad (3.59)$$

To obtain a simple interpretation of the prox-step (3.58), we note that it is equivalently given as the solution to the following optimization problem:

$$\min_z f(z^k) + \nabla f(z^k)^T (z - z^k) + \frac{L}{2} \|z - z^k\|_2^2 + g(z) \quad (3.60)$$

Accordingly, f is approximated around the point z^k by a second-order function with a uniform curvature. We also note that it is in general non-trivial to calculate the prox-step since it involves solving a convex optimization problem. However, the non-differentiable term g is in many cases simple such that the solution to (3.60) can be expressed in closed form.

To gain some more insights into the iteration (3.58), let us again consider the special case when the non-smooth term in (3.56) is the indicator function for a set \mathcal{Z} , i.e. when $g(z) = I_{\mathcal{Z}}(z)$. In this case, (3.58) simplifies to:

$$z^{k+1} = \text{proj}_{\mathcal{Z}} \left(z^k - \frac{1}{L} \nabla f(z^k) \right) \quad (3.61)$$

and the ordinary projected gradient method is recovered.

Algorithm 3 Fast proximal gradient method

- 1: **for** $k \geq 0$ **do**
 - 2: $w^k = z^k + \frac{k-1}{k+2}(z^k - z^{k-1})$
 - 3: $z^{k+1} = \text{prox}_{\frac{1}{L}g} \left(w^k - \frac{1}{L} \nabla f(w^k) \right)$
 - 4: **end for**
-

A common approach to improve the convergence of the iteration (3.58) is to deploy its accelerated variant given in Algorithm 3. Note that the difference between (3.58) and Algorithm 3 lies in basic arithmetic operations, which are cheap compared to evaluating the prox-step, which is the computational bottleneck in both methods.

Chapter 4

Summary of included papers

This Chapter provides a brief summary of the papers included in the thesis. Full versions of the papers can be found in Part II. The papers have been reformatted to ease readability and to comply with the layout of the rest of the thesis.

Standard MPC

Paper 1

Emil Klintberg, Sebastien Gros. *Approximate inverses in preconditioned fast dual gradient methods for MPC*. Accepted to the 20th IFAC World Congress, July 2017.

The paper considers the usage of approximate inverses to reduce the computational cost per iteration in a preconditioned dual proximal gradient method for MPC. The motivation is stemming from the observation that preconditioned or generalized fast gradient methods usually perform significantly better on ill-conditioned problems compared to ordinary fast gradient methods, at the cost of more computationally expensive iterations. We show that for a dualization of the dynamic constraints, the dense preconditioner is an exponentially off-diagonally decaying matrix. By approximating the preconditioner by a banded matrix, we show that the computational cost per iteration can be decreased while numerical experiments indicate that the number of iterations is almost unaffected in cases where the off-diagonal decay is rapid. In such cases, the approach may result in significant performance improvements.

The basic idea and the work is due to Emil Klintberg. Useful comments were provided by Sebastien Gros.

Multi-stage MPC

Paper 2

Emil Klintberg, Sebastien Gros. *A Parallelizable Interior Point Method for Two-Stage Robust MPC*. IEEE Transactions on Control Systems Technology, vol. PP, 2017.

The paper considers a parallelizable interior point method for two-stage MPC problems, i.e. for multi-stage MPC problems where the scenario tree is restricted to only branch in its root node. To enhance parallel computations, we introduce separate state and control variables for each scenario, and solve the interior-point KKT system on normal equation form. The extra dimensionality allows for decomposing the normal equations into a small and highly structured linear system and several decomposable systems that commonly arise when interior point methods are deployed in the context of MPC. State-of-the-art factorization techniques are then cornerstones of the proposed method.

The basic idea and the work is due to Emil Klintberg. Useful comments were provided by Sebastien Gros.

Paper 3

Emil Klintberg, John Dahl, Jonas Fredriksson, Sebastien Gros. *An improved dual Newton strategy for scenario-tree MPC*. IEEE Conference on Decision and Control, pp. 3675-3681, December 2016.

The paper considers a dual Newton strategy for multi-stage MPC problems. In this context, it has been observed that the Hessian of the dual function has an intricate sparsity structure and can be rank deficient, hence requiring a computationally expensive linear algebra and a regularization strategy. We show that a re-ordering of the non-anticipativity constraints results in a block-tridiagonal dual Hessian, hence dramatically reducing the cost of its factorization. Moreover, a simple and inexpensive strategy for eliminating redundant constraints is proposed for ensuring positive definiteness of the dual Hessian, making regularization superfluous.

The basic ideas of reordering the non-anticipativity constraints and eliminating redundant constraints are due to Emil Klintberg. Most algorithmic details are due to Emil Klintberg. Control design, simulations and useful comments are due to John Dahl. The proof showing that constraint elimination results in a positive definite Hessian is due to Sebastien Gros. Useful comments were provided by Jonas Fredriksson.

Paper 4

Emil Klintberg, Dimitris Kouzoupis, Moritz Diehl, Sebastien Gros.
A dual Newton strategy with fixed iteration complexity for multi-stage MPC. Technical report. CPL Publication id: 248820. Chalmers University of Technology.

This paper considers a dual Newton strategy for the practically important class of multi-stage MPC problems with a diagonal cost and simple bounds. In the context of dual Newton strategies, the globalization can be expensive since every back-tracking step involves solving QPs which often becomes the computational bottleneck of the method. In this paper, we dualize both the non-anticipativity constraints and the dynamical constraints. As a result, solving the QPs become computationally negligible whereas the resulting dual Newton system becomes larger. To enhance the solution of the Newton system it is reformulated into several small highly structured linear systems, which to a large extent can be solved in parallel. Consequently, computational effort is moved from the solution of the QPs to the solution of the Newton system. This is an improvement since the Newton system is only solved once every iteration whereas the QPs may be solved multiple times. As a result, we obtain a dual Newton strategy with a computational cost per iteration that is essentially constant.

The basic idea and most algorithmic details are due to Emil Klintberg. Control design, simulations and useful comments are due to Dimitris Kouzoupis. Useful comments were provided by Moritz Diehl and Sebastien Gros.

Distributed MPC

Paper 5

Attila Kozma, **Emil Klintberg**, Sebastien Gros, Moritz Diehl. *An Improved Distributed Dual Newton-CG Method for Convex Quadratic Programming Problems*. American Control Conference, pp. 2324-2329, July 2014.

This paper considers a dual Newton strategy to solve distributed QPs. In this context, it has been observed that the Hessian of the Lagrange dual function can be singular when a poor initial guess is provided for the dual variables, leading to an inconsistent Newton system and hence a failure of the linear algebra. In this paper, we study this effect and propose a constraint relaxation strategy to prevent this problem. It is shown both formally and experimentally that the relaxation prevents rank deficiency of the dual Hessian. To avoid communicating matrices between subsystems, the dual Newton system is solved using a Conjugate Gradient (CG) method.

The basic idea is due to Moritz Diehl. Most algorithmic details are due to Attila Kozma and Sebastien Gros. Control design and simulations are due to Emil Klintberg.

Paper 6

Emil Klintberg, Sebastien Gros. *A Primal-Dual Newton Method for Distributed Quadratic Programming*. IEEE Conference on Decision and Control, pp. 5843-5848, December 2014.

This paper considers a dual Newton strategy for solving distributed QPs. To avoid singularity of the dual Hessian, the local inequality constraints are augmented with the objective function using a log-barrier, making the method similar to a distributed interior point method. We show that Hessian of the dual function and linear predictors can be calculated inexpensively by re-using local factorizations. Numerical experiments indicate that it is sufficient to perform one iteration on the local subproblems, i.e. to solve one linear system, instead of solving the subproblems at every dual Newton step. This is at a large extent possible since every iteration is warm-started using the linear predictors. In the case of a single iteration on each subproblem, the method can be viewed as a distributed primal-dual interior point method.

The basic idea is due to Emil Klintberg and Sebastien Gros. The work is due to Emil Klintbergs and useful comments were provided by Sebastien Gros.

Paper 7

Emil Klintberg, Sebastien Gros. *Numerical structure of the Hessian of the Lagrange dual function for a class of convex problems*. SIAM Journal on Control and Optimization, vol. 55, issue 1, pp. 574-593, 2017.

This paper investigates the numerical structure of the Hessian of the Lagrange dual function for a class of separable convex problems. The problem class is motivated by the deployment of MPC on multi-agent systems that interact via non-delayed couplings. We show that the dual decomposition of this problem yields a numerical structure in the Hessian of the dual function. This numerical structure allows for deploying a quasi-Newton method in the dual space. We show that the structure holds also for log-barrier methods. The quasi-Newton method helps reducing the communication and computational burden for this class of problems.

The basic idea and the work is due to Emil Klintberg. Useful comments were provided by Sebastien Gros.

Chapter 5

Concluding remarks and future research directions

In this thesis, we have studied optimization methods for three problem classes; standard MPC, multi-stage MPC and methods for distributed optimization problems. The contributions are mostly directed towards lowering the computational burden of solving such problems. In this chapter, we provide some comments regarding possible extensions of the results, and directions for further research.

Off-diagonally decaying matrices arise in many contexts of control and signal processing, and are often overlooked in the quest for useful structures. As an example, the results of Paper 1 can be extended to the multi-stage MPC problem in Paper 3 and Paper 4. This is a direct consequence of the banded dual Hessian in Paper 3, and the banded matrices in the linear systems that the Newton system is reformulated into in Paper 4.

The results regarding off-diagonally decaying matrices in Paper 1 and Paper 7 can significantly reduce the computational burden of solving problems in practice. The rate of the off-diagonal decay can, however, vary strongly between seemingly very similar problem instances, and thus affect the practical benefits of the methods. In Paper 1, this could be considered as a minor drawback since the rate of the decay can be evaluated off-line. The benefits of the results would, however, increase if practically useful guidelines are developed on how to perform the approximations in the methods.

Note that the interior point method for two-stage MPC in Paper 2 can be trivially extended to multi-stage MPC problems. By arranging the non-anticipativity constraints as in Paper 3 and Paper 4, the first normal equation will then take the form of the Newton system in Paper 3. As a result, the first normal equation can be solved using a similar blockwise strategy as described in Paper 3, whereas the rest of the method remain identical as described in Paper 2.

References

- [1] C. R. Cutler. and B. L. Ramaker., “Dynamic matrix control - a computer control algorithm,” in *AICHE national meeting*, 1979.
- [2] C. Cutler., A. Morshedi., and J. Haydel., “An industrial perspective on advanced control,” in *AICHE annual meeting*, 1983.
- [3] D. M. Prett. and R. D. Gillette., “Optimization and constrained multivariable control of a catalytic unit,” in *Joint automatic control conference*, 1980.
- [4] D. Mayne., J. Rawlings., C. Rao., and P. Scokaert., “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 26, pp. 789–814, 2000.
- [5] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, pp. 3–20, 2002.
- [6] A. Bemporad., F. Borrelli., and M. Morari., “Piecewise linear optimal controllers for hybrid systems,” in *American Control Conference*, 2000.
- [7] T. Johansen and A. Grancharova, “Approximate explicit constrained linear model predictive control via orthogonal search tree,” *IEEE Transactions on Automatic Control*, vol. 48, pp. 810–815, 2003.
- [8] P. Tøndel, T. Johansen, and A. Bemporad, “Computation and Approximation of Piecewise Affine Control Laws via Binary Search Trees,” *Automatica*, vol. 39, pp. 945–950, 2003.
- [9] ———, “An Algorithm for Multi-Parametric Quadratic Programming and Explicit MPC Solutions,” *Automatica*, vol. 39, pp. 489–497, 2003.
- [10] I. Nielsen and D. Axehill, “Reduced memory footprint in multiparametric quadratic programming by exploiting low rank structure,” in *IEEE Conference on Decision and Control*, 2016.

REFERENCES

- [11] C. Rao, S. Wright, and J. Rawlings, “Application of Interior-Point Methods to Model Predictive Control,” *Journal of Optimization Theory and Applications*, vol. 99, pp. 723–757, 1998.
- [12] Y. Wang and S. Boyd, “Fast Model Predictive Control Using Online Optimization,” *IEEE Transactions on Control Systems Technology*, vol. 18, pp. 267–278, 2010.
- [13] A. Domahidi, A. Zgraggen, M. Zeilinger, M. Morari, and C. Jones, “Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control,” in *IEEE Conference on Decision and Control*, 2012.
- [14] G. Frison, K. M. Kufoalor, L. Imsland, and J. B. Jorgensen, “Efficient Implementation of Solvers for Linear Model Predictive Control on Embedded Devices,” in *IEEE International Conference on Control Applications*, 2014.
- [15] G. Frison, L. E. Sokoler, and J. Jorgensen, “A Family of High-Performance Solvers for Linear Model Predictive Control,” in *IFAC World Congress*, 2014.
- [16] I. Nielsen and D. Axehill, “A Parallel Structure Exploiting Factorization Algorithm with Application to Model Predictive Control,” in *IEEE Conference on Decision and Control*, 2015.
- [17] H. J. Ferreau, H. G. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit MPC,” *International Journal of Robust and Nonlinear Control*, vol. 18, pp. 816–830, 2008.
- [18] H. Ferreau, A. Kozma, and M. Diehl, “A parallel active-set strategy to solve sparse parametric quadratic programs arising in MPC,” in *IFAC Nonlinear Model Predictive Control Conference*, 2012.
- [19] J. V. Frasch, S. Sager, and M. Diehl, “A Parallel Quadratic Programming Method for Dynamic Optimization Problems,” *Mathematical Programming Computation*, vol. 7, pp. 289–329, 2015.
- [20] B. O’Donoghue., G. Stathopoulos., and S. Boyd., “A Splitting Method for Optimal Control,” *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 2432–2442, 2013.
- [21] S. Richter, C. Jones, and M. Morari, “Real-time input-constrained MPC using fast gradient methods,” in *IEEE Conference on Decision and Control*, 2009.
- [22] M. Kögel. and R. Findeisen, “A Fast Gradient method for embedded linear predictive control,” in *IFAC World Congress*, 2011.

- [23] S. Richter., C. Jones., and M. Morari, “Certification aspects of the fast gradient method for solving the dual of parametric convex programs,” *Mathematical Methods of Operations Research*, vol. 77, pp. 305–321, 2013.
- [24] P. Patrinos. and A. Bemporad., “An accelerated dual gradient-projection algorithm for embedded linear model predictive control,” *IEEE Transactions on Automatic Control*, vol. 59, pp. 18–33, 2014.
- [25] J. Jerez, P. Goulart, S. Richter, G. Constantinides, E. Kerrigan, and M. Morari, “Embedded Online Optimization for Model Predictive Control at Megahertz Rates,” *IEEE Transactions on Automatic Control*, vol. 59, pp. 3238–3251, 2014.
- [26] W. Zuo. and Z. Lin., “A Generalized Accelerated Proximal Gradient Approach for Total-Variation-Based Image Restoration,” *IEEE Transactions on Image Processing*, vol. 20, pp. 2748–2759, 2011.
- [27] P. Giselsson., “Improved Fast Dual Gradient Methods for Embedded Model Predictive Control,” in *IFAC World Congress*, 2014.
- [28] P. Giselsson. and S. Boyd., “Preconditioning in fast dual gradient methods,” in *IEEE Conference on Decision and Control*, 2014.
- [29] J. H. Lee. and Z. H. Yu., “Tuning of model predictive controllers for robust performance,” *Computers & Chemical Engineering*, vol. 18, pp. 15 – 37, 1994.
- [30] G. Grimm., M. Messina., S. Tuna., and A. Teel., “Examples when nonlinear model predictive control is nonrobust,” *Automatica*, vol. 40, pp. 1729–1738, 2004.
- [31] P. J. Campo and M. Morari, “Robust model predictive control,” in *American Control Conference*, 1987, pp. 1021–1026.
- [32] P. O. M. Scokaert and D. Q. Mayne, “Min-max feedback model predictive control for constrained linear systems,” *IEEE Transactions on Automatic Control*, vol. 43, pp. 1136–1142, 1998.
- [33] D. Mayne, M. Seron, and S. Rakovic, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, pp. 219–224, 2005.
- [34] D. Mayne., E. Kerrigan., E. J. van Wyk., and P. Falugi., “Tube-based robust nonlinear model predictive control,” *International Journal of Robust Nonlinear Control*, vol. 21, pp. 1341–1353, 2011.

REFERENCES

- [35] D. Munoz de la Pena, A. Bemporad, and T. Alamo, “Stochastic programming applied to model predictive control,” in *IEEE Conference on Decision and Control*, 2005.
- [36] D. Bernardini and A. Bemporad, “Scenario-based model predictive control of stochastic constrained linear systems,” in *IEEE Conference on Decision and Control*, 2009.
- [37] M. C. Steinbach, “Tree-Sparse Convex Programs,” *Mathematical Methods of Operations Research*, vol. 56, pp. 347–376, 2002.
- [38] R. Marti, S. Lucia, D. Sarabia, R. Paulen, S. Engell, and C. de Prada, “An Efficient Distributed Algorithm for Multi-Stage Robust Nonlinear Predictive Control,” in *European Control Conference*, 2015.
- [39] C. Leidereiter, A. Potschka, and H. G. Bock, “Dual decomposition for QPs in scenario tree NMPC,” in *European Control Conference*, 2015.
- [40] L. Magni and R. Scattolini, “Stabilizing decentralized model predictive control of nonlinear systems,” *Automatica*, vol. 42, pp. 1231–1236, 2006.
- [41] D. Raimondo, L. Magni, and R. Scattolini, “Decentralized MPC of nonlinear systems: An input-to-state stability approach,” *International Journal of Robust Nonlinear Control*, vol. 17, pp. 1651–1667, 2007.
- [42] A. Venkat, J. Rawlings, and S. Wright, “Stability and optimality of distributed model predictive control,” in *IEEE Conference on Decision and Control*, 2005.
- [43] J. Rawlings and B. Stewart, “Coordinating multiple optimization-based controllers: New opportunities and challenges,” *Journal of Process Control*, vol. 18, pp. 839–845, 2008.
- [44] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar, “Distributed model predictive control,” *Control Systems Magazine*, vol. 22, pp. 44–52, 2002.
- [45] T. Keviczky., F. Borrelli., and G. Balas., “Decentralized receding horizon control for large scale dynamically decoupled systems,” *Automatica*, vol. 42, pp. 2105–2115, 2006.
- [46] A. Richards and J. How, “Robust distributed model predictive control,” *International Journal of Control*, vol. 80, pp. 1517–1531, 2007.
- [47] B. Johansson., A. Speranzon., M. Johansson., and K. H. Johansson., “On decentralized negotiation of optimal consensus,” *Automatica*, vol. 44, pp. 1175–1179, 2008.

- [48] N. Marcos., J. F. Forbes., and M. Guay., “Coordination of Distributed Model Predictive Controllers for Constrained Dynamic Processes,” in *IFAC Symposium on Advanced Control of Chemical Processes*, 2009.
- [49] H. Scheu. and W. Marquardt., “Sensitivity-based coordination in distributed model predictive control,” *Journal of Process Control*, vol. 21, pp. 715–728, 2011.
- [50] R. Cheng., J. F. Forbes., and W. S. Yip., “Price-driven coordination for solving plant-wide MPC problems,” *Journal of Process Control*, vol. 17, pp. 429–438, 2007.
- [51] Y. Wakasa., M. Arakawa., K. Tanaka., and T. Akashi., “Decentralized model predictive control via dual decomposition,” in *IEEE Conference on Decision and Control*, 2008.
- [52] J. Mota., J. Xavier., P. Aguiar., and M. Püschel., “Distributed ADMM for Model Predictive Control and Congestion Control,” in *IEEE Conference on Decision and Control*, 2012.
- [53] C. Conte, T. Summers, M. Zeilinger, M. Morari, and C. Jones, “Computational Aspects of Distributed Optimization in Model Predictive Control,” in *IEEE Conference on Decision and Control*, 2012.
- [54] P. Giselsson., M. D. Doan., T. Keviczky., B. D. Schutter., and A. Rantzer., “Accelerated gradient methods and dual decomposition in distributed model predictive control,” *Automatica*, vol. 49, pp. 829 – 833, 2013.
- [55] A. Kozma, J. V. Frasch, and M. Diehl, “A Distributed Method for Convex Quadratic Programming Problems Arising in Optimal Control of Distributed Systems,” in *Conference on Decision and Control*, 2013.
- [56] P. Trnka., J. Pekar., and V. Havlena., “Application of Distributed MPC to Barcelona Water Distribution Network,” in *IFAC World Congress*, 2011.
- [57] J. R. Birge and D. F. Holmes, “Efficient Solution of Two-Stage Stochastic Linear Programs Using Interior Point Methods,” *Computational Optimization and Applications*, vol. 1, pp. 245–276, 1992.
- [58] E. Jessup, D. Yang, and S. Zenios, “Parallel factorization of structured matrices arising in stochastic programming,” *SIAM Journal on Optimization*, vol. 4, pp. 833–846, 1994.
- [59] I. Necoara and J. Suykens, “An interior-point Lagrangian decomposition method for separable convex optimization,” *Journal of Optimization Theory and Applications*, vol. 143, pp. 567–588, 2009.

REFERENCES

- [60] D. Q. Tran, I. Necoara, C. Savorgnan, and M. Diehl, “An inexact perturbed path-following method for lagrangian decomposition in large-scale separable convex optimization,” *SIAM Journal on Optimization*, vol. 23, pp. 95–125, 2013.
- [61] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill, 2009.
- [62] S. Lucia., “Robust Multi-stage Nonlinear Model Predictive Control,” Ph.D. dissertation, Technical University of Dortmund, 2014.
- [63] S. Lucia, T. Finkler, and S. Engell, “Multi-stage Nonlinear Model Predictive Control Applied to a Semi-batch Polymerization Reactor under Uncertainty,” *Journal of Process Control*, vol. 23, pp. 1306–1319, 2013.
- [64] S. Lucia, J. Andersson, H. Brandt, M. Diehl, and S. Engell, “Handling uncertainty in economic model predictive control: A comparative case study,” *Journal of Process Control*, vol. 24, pp. 1247–1259, 2014.
- [65] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: University Press, 2004.
- [66] N. Andreasson., A. Evgrafov., M. Patriksson., E. Gustavsson., and M. Önnheim., *An Introduction to Continuous Optimization*. Studentlitteratur, 2005.
- [67] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.
- [68] Y. Nesterov., “Smooth minimization of non-smooth functions,” *Mathematical Programming*, vol. 103, pp. 127–152, 2005.
- [69] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.
- [70] Y. Nesterov, “Gradient methods for minimizing composite functions,” *Mathematical Programming*, vol. 140, pp. 125–161, 2013.