Fast MPC Solvers for Systems with Hard Real-Time Constraints

Xi Zhang

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Fast MPC Solvers for Systems with Hard Real-Time Constraints

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Xi Zhang

November 18, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Fast MPC Solvers for Systems with Hard Real-Time Constraints

by

Xi Zhang

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: November 18, 2016

Committee Member(s):

Dr. Ir. Tamás Keviczky

Dr. Ir. Sander Wahls

Laura Ferranti

Abstract

Model predictive control (MPC) is an advanced control technique that offers an elegant framework to solve a wide range of control problems (regulation, tracking, supervision, etc) and handle constraints on the plant. The control objectives and constraints are usually formulated as an optimization problem that the MPC controller has to solve (either offline or online) to return the control command for the plant.

The main contribution of this master thesis is that it proposes a novel primal-dual interior-point (PDIP) method for solving quadratic programming problems with linear inequality constraints that typically arise from MPC applications. Convergence of PDIP is studied both in primal and dual framework. We show that the solver converges quadratically to a suboptimal solution of the MPC problem. PDIP solvers rely on two phases: the damped and the pure Newton phases. Compared to state-of-the-art PDIP method, this new solver replaces the initial (linearly convergent) damped Newton phase (usually used to compute a medium-accuracy solution) with a dual solver based on Nesterov's fast gradient scheme (DFG) that converges super-linearly to a medium-accuracy solution. The switching strategy to the pure Newton phase, compared to the state of the art, is computed in the dual space to exploit the dual information provided by the DFG in the first phase. Removing the damped Newton phase has the additional advantage that this solver saves the computational effort required by backtracking line search. The effectiveness of the proposed solver is demonstrated by simulating it on a 2-dimensional discrete-time unstable system.

Table of Contents

Acknowledgements

First and foremost, I would like express my sincere gratitude to my supervisor Dr. Ir. Tamás Keviczky for his continuous support during my master thesis project and writing of this thesis. He guided me in all the time to solve numerous professional problems with his patience and immense knowledge. In addition, I really appreciate the opportunities he afforded me to help me to broaden my eyes in the field related to my thesis project.

Besides, I would like to thank my daily supervisor Laura Ferranti for her assistant during this thesis work. She supported me in regular meetings and discussions with her kindness and encouragement. I learned a lot from the meetings and her feedback. She is a dear friend to me as well.

Last but not the least, I would like to thank my parents for their unconditional love, understanding and support during my studies in TU Delft, and my friends for their accompany and friendship.

Delft, University of Technology *Xi Zhang*
November 18, 2016

"I do not know what I may appear to the world, but to myself I seem to have been only like a boy playing on the sea-shore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me."

— *Isaac Newton*

Chapter 1

Introduction

Model predictive control (MPC) was developed and used in industry for nearly 30 years and is continuously drawing attention from industry by being able to solve a wide range of control problems, e.g., regulation, tracking, supervision, etc., taking into account physical system constraints with optimized closed-loop performance. State and output constraints arise from practical restrictions on the allowed operating range of the system [9]. As it can never be excluded that the state and output of the system move outside the constrained range chosen for the controller design, special provisions must be made to move the state back into the range. This is difficult is exactly what one wanted to avoid by choosing MPC in the very first place comparing to classical control methods, for instance, PID control.

The main contribution of this master thesis focuses on fast MPC problems associated with linear time-invariant (LTI) systems. In these problems, a quadratic programming (QP) problem is recursively solved over a given prediction horizon and new state information is updated from measurements of the plant, at each time instant. The presence of this optimization problem has traditionally limited the use of MPC to *slow* processes, that is, processes with no hard real-time constraints. Recently, MPC has received increasing attention in fields, such as aerospace and automotive, where the real-time aspects are critical and the computation time for the controller is limited. Hence, offline [4] and online solutions have been investigated to overcome the computational issue related to the MPC controller. In this work, we focus on online solutions that allow one to handle a wider range of problems.

1-1 Computational Methods for Model Predictive Control

Efficient methods have been developed to cope desired requirements in embedded MPC. These methods can be roughly divided into two different categories, offline and online solvers.

Among the techniques proposed to solve MPC problems offline, we have, for example, the explicit MPC designed in [4] to solve multi-parametric QPs (mp-QP), where the parameters are the components of the state vector. This approach removes the drawbacks of solving an optimization problem online to compute the control action. In this approach, the optimal

solution for all possible states are computed offline for a given range of operating condition of interest. Therefore, the MPC law is converted into a continuous and piecewise affine (PWA) function of the state vector by partitioning the space of the states and inputs into polyhedral regions and associating with each region a different linear state-update equation [22]. In this way, the online control computation reduces to the simple evaluation of an explicitly defined PWA function which maps the MPC control law into a *look-up table* of linear gains. The main drawback of explicit MPC is that explicit solution is often limited by memory demand and restricted to small-scale systems since the look-up table might grow exponentially in the number of states.

Due to the memory requirement and application restriction of explicit MPC, during the last years, various researchers have spent considerable efforts on accelerating online QP solvers that contribute to fast MPC embedded in high-frequency dynamics [10, 27, 37, 39].

The main issue that arises solving the MPC problem online is related to the limited amount of computational time. Thus, these kinds of applications posed a research challenge for developing optimization algorithms, and in particular quadratic programming solvers, which enable the use of MPC in commercial products. Hence, the study of fast solvers to compute their solution online within guaranteed computation time has gained growing attention. Specifically, an embedded optimization solver is required to:

- provide a sub-optimal solution within the available sampling interval;
- occupy small memory to store the data defining the optimization problem;
- be easy-to-code and software certifiable;
- have predictable worst-case execution time in order to satisfy real-time system requirements.

Online optimization solvers can be divided into two main families: first- and second-order methods. First-order methods, such as gradient or splitting methods and their accelerated versions like Nestrov's fast gradient method [20, 29, 31], have simpler theoretical requirements, such as, Lipschitz continuity only on the first derivative of the cost, and converge to a medium-accuracy solutions within few iterations. Fast gradient methods appear to be significantly easy-to-code and software certifiable. In addition, availability of analytical upper bound on the number of iterations and *super-linear* convergence rate make these methods stand out from other classical first-order methods [29]. One short coming of fast gradient methods is that, when there are constraints to be handled, the problem is related to the projection step that can be computationally challenging. Moreover, when there are ill-conditioned problems, that is to say, if the condition number is large, the solvers easily becomes unstable. Many researchers have explored this field to guarantee primal feasibility while using a dual method [2, 11, 25].

Second-order algorithms for QP problems, such as, active-set methods [10, 33, 35] and interior-point methods [8, 23, 37, 40], can be very efficient in speed and provide solutions in high quality. Among them, interior-point methods appear to be the most appealing. Practical applications have shown that these methods are able to converge to the optimal solution within only a few number of iterations [37]. Besides, these methods perform efficiently to guarantee

a primal feasible solution by introducing *residual* variables according to Karsh-Kuhn-Tucker (KKT) conditions. Compared to first-order methods, however, interior-point methods usually require more advanced algebraic operations and, consequently, more computational effort. Moreover, the analytical upper bound on the number of iterations that guarantees to reach a desired level of accuracy is too conservative to be used in practice.

Motivated by the studies on fast MPC solvers, this master project aims to develop a novel algorithm that combines the benefits of interior-point and fast gradient solvers to provide a certified control law in terms of the level of sub-optimality and worst-case computation time.

1-2 Contribution

The main contribution of this master thesis is a novel primal-dual interior-point solver for solving inequality-constrained QP problems that commonly arise from MPC applications. The proposed solver combines the advantages of the Nesterov's fast gradient method [29] and of primal-dual interior-point solvers. In this work, we are mainly interested in primal-dual interior-point solvers presented in [9]. In [5], their convergence can be divided into two phases:

- Damped Newton Phase. The algorithm performs a linear convergence and is used to reach a medium-accuracy solution.
- Pure Newton Phase. This phase is characterized by a quadratic convergence rate with unit step size selected in backtracking line search.

In particular, we exploit the ability of the fast gradient method to converge to medium-accuracy solutions within few iterations, for which the algorithm can be performed efficiently, and the ability of primal-dual interior-point solvers to converge to high-accuracy solutions in pure Newton phase. The proposed combination aims to improve the convergence of the primal-dual interior-point method by replacing damped Newton phase, in which the algorithm converges linearly, with the fast gradient method, that converges super-linearly. We modify the classical analysis of the primal-dual interior-point method [5] to take into account the presence of (active) inequality constraints and allow the switch between the two solvers. This is done by moving the convergence analysis of the solver and the formulation of the switching strategy into dual framework. As a consequence, we provide bounds on the level of primal suboptimality and infeasibility achieved with our proposed algorithm. An additional feature of the proposed solver is that the computational effort related to the backtracking line search, which is required in damped Newton phase, is removed, given that pure Newton phase uses a unit step size. Finally, the solver is validated on an MPC application, the regulation of an unstable input and output constrained planar system.

We also try to analyze the condition for primal-dual interior-point method in primal space. In this respect, in primal space, the convergence of primal-dual interior-point method can be analyzed using its similarity to the barrier method [5]. The main limitation, however, is that the algorithm has to be initialized with a strictly feasible starting point. And this condition is essential for deriving the coincidence of barrier and primal-dual interior-point methods. A strictly feasible starting point can be guaranteed by introducing either softening constraint approach [19] in the problem solved by primal-dual interior-point solver, or by

tightening the constraints [25] in the problem solved by the dual fast gradient method. These switching condition in primal space, however, becomes undefined when there are constraints active at the optimum. We do not consider the convergence analysis in [5, Chapter 10] based on the convergence of residual variable with infeasible start approach since in the presence of inequality, convergence in [5, Chapter 10] will become too conservative to be used in practice.

This master thesis work is submitted for publication at The 20th World Congress of the International Federation of Automatic Control (IFAC WC 2017).

1-3 Structure of the Thesis

This master thesis is organized in the following structure: Chapter 2 gives a brief introduction on model predictive control and describes how a general MPC problem can be reformulated as a QP. Chapter 3 introduces Nestrov's fast gradient method. Chapter 4 provides an overview on interior-point methods, including primal barrier method and primal-dual interior-point methods. Convergence analysis and relations between two methods are studied. Chapter 5 proposes our novel solvers. Chapter 6 presents the numerical results. Appendix provides detailed mathematical proofs.

Chapter 2

Model Predictive Control: an Overview

Originated in the late seventies, model predictive control is an advanced control methodology which has made a significant impact on industrial control engineering for its capability of optimizing closed-loop performance subject to operating constraints on input and output variables. The theory of MPC is well developed in nearly all aspects, such as stability, nonlinearity, and robustness in [3], [6] and [24]. It has been mostly applied in the petrochemical industry, but it is currently being increasingly applied in other sectors of the process industry [21]. The main reasons for its success in these applications are:

- It handles multi-variable control problems naturally.
- It can take account of actuator limitations.
- It allows operation closer to constraints (compared with conventional control), which frequently leads to more profitable operation.

Loosely speaking, under some assumptions, MPC allows one to ensure closed-loop performance while satisfying the constraints on the plant.

In this chapter, we first provide a brief mathematical description of the MPC problem. After that, this problem is formulated as a standard QP which can be solved by common optimization solvers at present.

2-1 Problem Formulation

In model predictive control, the following *discrete, linear, time-invariant* plant is usually considered

$$x(t+1) = Ax(t) + Bu(t) \tag{2-1a}$$

$$y(t) = Cx(t) + Du(t) \tag{2-1b}$$

with the state $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$, input $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ and output $u \in \mathcal{Y} \subseteq \mathbb{R}^{n_y}$ where t denotes the sampling instant, and the system matrices $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, where (A, B) are stabilizable.

Typically, the input u, state x and output y should be constrained in each step to closed, convex sets $\mathcal{U}$, $\mathcal{X}$ and $\mathcal{Y}$ containing the origin. $\mathcal{U}$, $\mathcal{X}$ and $\mathcal{Y}$ can also be extended to constraints set such as boxes, polytopes, etc [29].

In this master thesis project, we consider model predictive control of system (2-1) subject to polytopic constraints. We consider, over all, prediction horizon of length $N \geq 1$, a quadratic cost criterion. The quadratic cost consists of a stage cost:

$$J_t(x_j, u_j) = \frac{1}{2} \|x_j\|_Q^2 + \frac{1}{2} \|u_j\|_R^2 + u_j^{\mathrm{T}} S x_j + q^{\mathrm{T}} x_j + r^{\mathrm{T}} u_j, \quad j = 0, 1, ..., N-1 \tag{2-2}$$

where weighting matrices $Q \succeq 0$, $Q \in \mathbb{R}_+^{n_x \times n_x}$, where $(A, \sqrt{Q})$ is detectable. $R \succeq 0, R \in \mathbb{R}^{n_u \times n_u}$, and $S \in \mathbb{R}^{n_u \times n_x}$ has full row rank. We also define terminal cost as:

$$J_N(x_N, u_N) = \frac{1}{2} \|x_N\|_P^2 + p^{\mathrm{T}} x_N \tag{2-3}$$

where $P \succeq 0$, $P \in \mathbb{R}_+^{n_x \times n_x}$ is the weight on terminal state. P is associated with the algebraic Riccati equation (ARE) for the unconstrained problem (A, B, Q, R). P can be easily computed using MPT3 toolbox with Q, R and S known [15].

The objective function of the quadratic programming problem formed by a model predictive control problem contains stage and terminal cost in (2-2) and (2-3):

$$J(x, u) = \sum_{j=0}^{N-1} J_t(x_j, u_j) + J_N(x_N, u_N) \tag{2-4}$$

This framework is rather general and contains several special cases. In particular classical stabilization i.e. the regulation control problem towards the origin yields $q = r = p = 0$ [20].

In this work, we focus on regulation problem of systems. As usual in MPC, the applied input is given as the first part of the optimal input sequence resulting from the following QP

$$\min_{u \in \mathcal{U},\ x \in \mathcal{X}} \quad J(x, u) = \sum_{j=0}^{N-1} J_t(x_j, u_j) + J_N(x_N, u_N) \tag{2-5a}$$

$$s.t. \quad x_{j+1} = Ax_j + Bu_j \tag{2-5b}$$

$$j = 0, ..., N \tag{2-5c}$$

$$y \in \mathcal{Y},\ x(t) = x_0 \tag{2-5d}$$

At each time instant, the current plant state is measured or estimated and Problem (2-5) is solved to obtain the optimal solution of control input sequence over prediction horizon N. Afterwards, only the first control input is implemented onto the system plant. At next sampling instant, the system is sampled again and the horizon is shifted towards the future

and the previous procedure is repeated. This principle is called *receding horizon control* as shown in Figure 2-1. The system is controlled by MPC controller to track a reference trajectory (red dotted solid line). At time instant t, the system output is measured (orange dotted solid line). With this measured output, current dynamic state can be obtained via an observer and the system dynamic over prediction horizon N can be predicted, as well as the system output (brown dotted solid line). Afterwards, the QP in (2-5) is solved and the optimal sequence of control input is obtained (light green solid line). The first control input is implemented and the sampling time is shifted to $t+1$. Closed-loop stability and dynamic performance can be guaranteed, meanwhile, constraints on system state, input and output are satisfied.

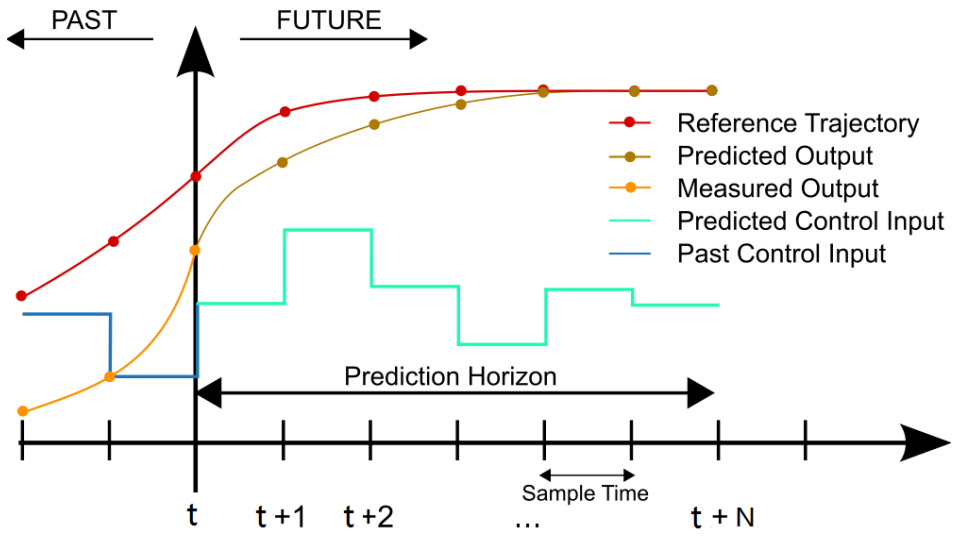


Figure 2-1: Receding horizon control principle [9].

2-2 Model Predictive Control as a Quadratic Programming Problem

We can reformulate Problem (2-5) into a more general way, that is, a standard QP problem of the following form:

$$\min_{\mathbf{z}\in\mathcal{Z}} \quad \frac{1}{2}\mathbf{z}^{\mathrm{T}}H\mathbf{z} + (hx_0)^{\mathrm{T}}\mathbf{z} \tag{2-6a}$$

$$s.t. \quad g(\mathbf{z}) \leq 0 \tag{2-6b}$$

$$f(\mathbf{z}) = 0 \tag{2-6c}$$

where the size of the information matrices and vectors H and h depends on the employed formulation. Moreover, $g(\mathbf{z})$ and $f(\mathbf{z})$ are affine functions. As explained in the next chapters, many solvers are available to compute an optimal solution for (2-6).

2-2-1 Non-Condensed Formulation

We use $\{x(t|j),\ j = 0, ..., N\}$ to denote the j step prediction at sampling instant t.

The *non-condensed* approach keeps the future states as decision variables and incorporates the system dynamics into the problem by enforcing equality constrains [30, 38]. In this case,

$$\mathbf{z} := \begin{bmatrix} x(t|0) \\ u(t|0) \\ x(t|1) \\ u(t|1) \\ \vdots \\ u(t|N-1) \\ x(t|N) \end{bmatrix}. \tag{2-7}$$

Assume $x(t|0) = x_0$, consider the discrete difference equation in (2-1). We can derive the following equation for $\mathbf{z}$:

$$\underbrace{\begin{bmatrix} I & & & & & & \\ -A & -B & I & & & & \\ & & -A & -B & & & \\ & & & & \ddots & & \\ & & & & & I & \\ & & & & & -A & -B & I \end{bmatrix}}_{A_N^{-1}} \mathbf{z} = \begin{bmatrix} x_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{2-8}$$

with $\mathbf{z}$ defined in (2-7). In this way, $h := 0$ in (2-6a) and the coefficient matrices in $g(\mathbf{z})$ obtained from non-condensed formulation is a lower triangular matrix which has sparse and banded structures to describe the optimal control problem in (2-6) exactly [17]. Tailored QP solvers can do better than the general QP solvers.

In non-condensed approach, the computational and memory requirements grow linearly with respect to the prediction horizon length N, which suggests that the non-condensed formulation is preferred by applications that require long prediction horizons [17].

2-2-2 Condensed Formulation

The standard linear MPC approach makes use of (2-6c) to eliminate the states from the decision variables by expressing them as an explicit function of the current state and future control inputs [21]. This is usually known as *condensed* QP formulation, which leads to compact and dense QPs.

In condensed QP formulation, only the control sequence is considered to be the decision variable:

$$\mathbf{z} := \begin{bmatrix} u(t|0) \\ u(t|1) \\ \vdots \\ u(t|N-1) \end{bmatrix} \tag{2-9}$$

Therefore, there will be no (2-6c) in Problem (2-6) with condensed formulation and Problem (2-6) only contains inequality constraint (2-6b). Then we obtain a set of inequalities with dense matrices expressing the decision variables as a function of the current state and input sequence:

$$\mathbf{x} = A_N x_0 + B_N \mathbf{z} \tag{2-10a}$$

$$\mathbf{y} = C_N x_0 + D_N \mathbf{z} \tag{2-10b}$$

where

$$\mathbf{x} = \begin{bmatrix} x(t|0)^{\mathrm{T}} & x(t|1)^{\mathrm{T}} & \dots & x(t|N)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \quad x_0 = x(t|0)$$

$$A_N = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^N \end{bmatrix}, \qquad B_N = \begin{bmatrix} 0 & & & & \\ B & 0 & & & \\ AB & B & 0 & & \\ \vdots & & \ddots & \ddots & \\ A^{N-2}B & & & B & 0 \\ A^{N-1}B & A^{N-2}B & \dots & AB & B \end{bmatrix}$$

$$C_N = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-1} \\ CA^N \end{bmatrix}, \qquad D_N = \begin{bmatrix} D & & & & & \\ CB & D & & & & \\ CAB & CB & D & & & \\ \vdots & & \ddots & \ddots & & \\ CA^{N-2}B & & & CB & D & \\ CA^{N-1}B & CA^{N-2}B & \dots & & CB & D \end{bmatrix}$$

Compared to non-condensed formulation approach, condensed formulation leads to a less-dimensional optimization problem.

In this master thesis project, we focus on condense formulation in which the decision variable contains only the control input sequence over the prediction horizon. Meanwhile, we consider mainly the regulation problem, but the approach can be easily extended to more practical tracking problems. Therefore, the coefficient vectors in (2-2) and (2-3) $q = r = p = 0$. With condensed formulation approach, the optimization problem in (2-5) can be written in to the following standard form which only consists of inequality constraints:

$$\min_{\mathbf{z}\in\mathcal{Z}} \quad f_0(\mathbf{z}) = \frac{1}{2}\mathbf{z}^{\mathrm{T}} H \mathbf{z} + (hx_0)^{\mathrm{T}} \mathbf{z} \tag{2-11a}$$

$$s.t. \quad g(\mathbf{z}) = G\mathbf{z} + Ex_0 + g \leq 0 \tag{2-11b}$$

where $\mathbf{z}$ is defined in (2-9) and

$$H = B_N^{\mathrm{T}} \hat{Q} B_N + \hat{R}, \qquad h = B_N^{\mathrm{T}} \hat{Q} A_N$$

$$\hat{Q} = \begin{bmatrix} I_N \otimes Q & 0 \\ 0 & P \end{bmatrix}, \qquad \hat{R} = I_N \otimes R$$

with $H \in \mathbb{R}^{Nn_u \times Nn_u}$, $h \in \mathbb{R}^{Nn_u \times n_x}$ and $G \in \mathbb{R}^{m \times Nn_u}$. For simplicity, denote $Nn_u = n$.

Proof. [21] In condensed formulation, we use the predicted state sequence (2-5b) to eliminate state variables in the objective function. Therefore, we can get rid of equality constraints.

In a regulation problem, we consider the following objective function to be minimized:

$$J(x,u) = \frac{1}{2}\sum_{j=0}^{N-1}\left(\|x_j\|_Q^2 + \|u_j\|_R^2\right) + \frac{1}{2}\|x_N\|_P^2 \tag{2-12}$$

Due to the space limitation, we use the a short notation in the following parts of this proof. At sampling time instant t, we use $x(j),\ j = 0,1,...,N$ instead of $x(t|j),\ j = 0,1,...,N$ to denote the state over prediction horizon N, and $u(j),\ j = 0,1,...,N-1$ for input sequence $u(t|j),\ j = 0,1,...,N-1$.

Substituting (2-10a) into (2-12) and obtain the following:

$$\begin{aligned}
J(x,u) =& \frac{1}{2}\begin{bmatrix} x(0)^{\mathrm{T}} & x(1)^{\mathrm{T}} & \dots & x(N-1)^{\mathrm{T}} & x(N)^{\mathrm{T}}\end{bmatrix}\begin{bmatrix} Q & & & & \\ & Q & & & \\ & & \ddots & & \\ & & & Q & \\ & & & & P\end{bmatrix}\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \\ x(N)\end{bmatrix} \\
&+ \frac{1}{2}\begin{bmatrix} u(0)^{\mathrm{T}} & u(1)^{\mathrm{T}} & \dots & u(N-1)^{\mathrm{T}}\end{bmatrix}\begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R\end{bmatrix}\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1)\end{bmatrix} \\
=& \frac{1}{2}\mathbf{x}^{\mathrm{T}}\begin{bmatrix} I_N \otimes Q & \\ & P\end{bmatrix}\mathbf{x} + \frac{1}{2}\mathbf{z}^{\mathrm{T}}\cdot I_N \otimes R \cdot \mathbf{z} \\
=& \frac{1}{2}\left[A_N x_0 + B_N \mathbf{z}\right]^{\mathrm{T}}\hat{Q}\left[A_N x_0 + B_N \mathbf{z}\right] + \frac{1}{2}\mathbf{z}^{\mathrm{T}}\hat{R}\mathbf{z} \\
=& \frac{1}{2}\mathbf{z}^{\mathrm{T}}\underbrace{\left(B_N^{\mathrm{T}}\hat{Q}B_N + \hat{R}\right)}_{H}\mathbf{z} + \underbrace{\left(x_0^{\mathrm{T}}A_N^{\mathrm{T}}\hat{Q}B_N\right)^{\mathrm{T}}}_{(hx_0)^{\mathrm{T}}}\mathbf{z} + \underbrace{\frac{1}{2}x_0^{\mathrm{T}}A_N^{\mathrm{T}}\hat{Q}A_N x_0}_{\text{constant}}
\end{aligned}$$

In this way, we obtain the optimization problem in (2-11). □

2-3 Basic Assumptions

In this master thesis, we assume from here on for Problem (2-11):

Assumption 2.1 (Strong Duality)**.** The Slater's condition hold for (2-11), i.e., there exist $\tilde{\mathbf{z}} \in \mathbf{relint}\mathcal{Z}$ with $g(\tilde{\mathbf{z}}) < 0$.

Assumption 2.2 (Strong Convexity)**.** Function $f_0(\mathbf{z})$ is strongly convex on $\mathcal{Z}$, which means that there exists an $m_p > 0$ such that

$$m_p I_n \preceq \nabla^2 f_0(\mathbf{z}) \leq M_p I_n$$

for all $\mathbf{z} \in \mathcal{Z}$.

Assumption 2.3 (Lipschitz Continuity)**.** The Hessian of $f_0(\mathbf{z})$ is Lipschitz continuous on $\mathcal{Z}$ with constant L_p

$$\|\nabla^2 f_0(\mathbf{z}) - \nabla^2 f_0(\tilde{\mathbf{z}})\|_2 \leq L_p \|\mathbf{z} - \tilde{\mathbf{z}}\|_2$$

for all $\mathbf{z}$, $\tilde{\mathbf{z}} \in \mathcal{Z}$.

Assumption 2.4. Matrix G in the function of inequality constraints has full row rank.

Chapter 3

Fast Gradient Methods for Quadratic Programming

This chapter focuses on Nesterov's gradient method and its extensions [27, 28, 29]. Compared to second-order solvers, fast gradient methods have the advantage of being very simple to implement, as well as division free and matrix-inversion free. Given that the main computational load lies in a matrix-vector multiplication, these methods are practically very parallelizable. In addition, the key benefit is that, in fast gradient methods, a tight upper bound can be found on the number of iterations to reach a desired degree of solution accuracy [26]. The primal version of fast gradient methods is well suited to implementation using fixed-point arithmetic [16]. Moreover, further developments to handle constraints with augmented Lagrange multipliers [20, 26] and combination with the alternating direction of multipliers method [32]. The field of distributed model predictive control are also explored [14], in which a stopping condition that guarantees feasibility and stability is developed for the duality-based distributed optimization algorithm. Moreover, improvement has been made in this field as well [13], where a distribute optimization algorithm with super-linear convergence rate is proposed. Online tools, such as, FiOrDoS [36] and μAO-MPC [41], generate highly portable C code tailored for embedded applications. These two toolboxes implements provably Nesterov's fast gradient method based on Lagrange relaxation. Both toolboxes provide a tailored MEX-interface for calling the generated solvers in MATLAB/Simulink.

This section starts from a brief introduction of Nesterov's fast gradient method in primal. Afterwards, considering more general cases of convex optimization problems with inequality and equality constraints, the dual fast gradient method is explained.

3-1 Fast Gradient Method

Fast gradient methods were developed by Y. Nesterov in 1983 [27]. It is similar to a gradient descent method and can be applied to constrained optimization problems. Therefore, we shortly review fast gradient method as well as the required background information of *gradient projection*.

3-1-1 Gradient Projection

We start with primal space where we only consider the following optimization problem constrained to a *convex*, *close* set $\mathcal{Z}$,

$$\min_{\mathbf{z}\in\mathcal{Z}} \quad J(\mathbf{z}) \tag{3-1}$$

with $J(\mathbf{z}) : \mathbb{R}^n \to \mathbb{R}$ is a smooth, convex function with a globally Lipschitz continuous gradient with constant L_J.

The update procedure of sequence $\{\mathbf{z}_k\}$ in the well-known gradient descent method is the following:

$$\mathcal{G}(\mathbf{z}_k) \coloneqq \mathbf{z}_{k+1} = \mathbf{z}_k - \frac{1}{L_J}\nabla J(\mathbf{z}_k) \tag{3-2}$$

where $\frac{1}{L_J}$ is chosen to be the step size according to [29].

Since the optimization variables are constrained in a convex feasible set $\mathcal{Z}$, Problem (3-1) can be solved using gradient projection [29]. Therefore, compared with traditional gradient descent method, Nesterov's fast gradient method solves the problem via an additional gradient projection step. Instead of the update law in (3-2), fast gradient method uses the projected gradient step $\mathcal{G}_{\mathcal{Z}}(\mathbf{z}_k)$:

$$\begin{cases} \mathcal{G}_{\mathcal{Z}}(\mathbf{z}_{k+1}) \coloneqq \arg\min\limits_{v\in\mathcal{Z}} \ \|\mathbf{z}_{k+1} - v\|^2 \\ \mathbf{z}_{k+1} = \mathcal{G}(\mathbf{z}_k) \end{cases} \tag{3-3}$$

Note that, the projected gradient step $\mathcal{G}_{\mathcal{Z}}(\mathbf{z}_{k+1})$ requires an Euclidean projection into $\mathcal{Z}$. This projection is generally nontrivial. However, it can be done analytically. Thus, this operation is central to compute the gradient mapping and defines a *simple set* as a convex set for which one can compute the projection easily. Such simple sets are for instance the n-dimensional Euclidean ball, positive orthant, and box.

Further more, the projected gradient step delivers feasible iterates and converges to the minimizer with the rate of convergence which is the best or either a linear or a sub-linear convergence rate [31]. In Nesterov's fast gradient method, an additional step (Step 2 in Algorithm 1) that leads to faster convergence than the gradient projection method, is used. With this further step, super-linear convergence in a rate of $\mathcal{O}(1/k^2)$ can be achieved [29].

3-1-2 Algorithm Description

Algorithm 1 Fast gradient method [29].

Require: $\mathbf{z}_0 \in \mathcal{Z}$, number of iterations $k_{\max}$, Lipschitz constant L_J, sequence c_k.
  Set $\mathbf{z}_{-1} = \hat{\mathbf{z}}_0 = \mathbf{z}_0$
  for $k = 1, ..., k_{\max}$ **do**
    1. Compute $\mathbf{z}_{k+1} = \mathcal{G}_{\mathcal{Z}}$ given (3-3).
    2. Compute $\hat{\mathbf{z}}_{k+1} = \mathbf{z}_{k+1} + c_k(\mathbf{z}_{k+1} - \mathbf{z}_k)$.
  end for
  return $\mathbf{z}_{k_{\max}}$.

Algorithm 1 shows the fast gradient method framework. It is always possible to choose $c_k = \frac{k-1}{k+2}$ [20]. Under Assumption 2.2, c_k can also be chosen constant: $c_k = \frac{\sqrt{L_J}-\sqrt{m_J}}{\sqrt{L_J}+\sqrt{m_J}}$, where $m_J = \sigma_{\min}(\nabla^2 J(\mathbf{z}))$ is the minimum eigenvalue of the Hessian $\nabla^2 J(\mathbf{z})$. Clearly, m_J is unique and $L_J \geq m_J$. It is advantageous to use a m_J as small as possible.

3-2 Dual Fast Gradient Method

In most cases, the optimization problem involved in a MPC problem is a constrained problem, e.g., Problem (2-11), where the projection on to the set defined by constraint $g(\mathbf{z}) \leq 0$ is hard to compute. Therefore, instead of using fast gradient method in Section 3-1 only, additional approaches has to be introduced. One approach to deal with optimization problems consisting of equality or inequality constraints relies on the use of Lagrange multiplier.

3-2-1 Dual Formulation

For Problem (2-11), we assume that the projection onto the set defined by the coupling constraints $g(\mathbf{z}) \leq 0$ is hard to compute. Therefore, the complicating constraints are moved into the cost via Lagrange multipliers by the following Lagrangian function:

$$\mathcal{L}(\mathbf{z}, \lambda) = f_0(\mathbf{z}) + \lambda^T g(\mathbf{z}) \tag{3-4}$$

where $\lambda \in \mathbb{R}^m_+$ is the Lagrange multiplier, and we can define the following dual function:

$$d(\lambda) = \min_{\mathbf{z}\in\mathcal{Z}} \mathcal{L}(\mathbf{z}, \lambda) \tag{3-5}$$

Since $f_0(\mathbf{z})$ is strongly convex, the unique optimal solution of the primal (inner) subproblem can also be denoted as:

$$\mathbf{z}^*(\lambda) = \arg\min_{\mathbf{z}\in\mathbb{R}^n} \mathcal{L}(\mathbf{z}, \lambda) \tag{3-6}$$

In our work, the constraints on input, state and output in sets $\mathcal{U}$, $\mathcal{X}$ and $\mathcal{Y}$ are substituted into the function $g(\mathbf{z}) \leq 0$, which leads to an unconstrained least square problem w.r.t λ, e.g., Problem (3-6), which can be solved exactly.

We made the following assumption as well.

Assumption 3.1. The inner problem (3-6) can be solved exactly.

By Assumption 2.1, we have for the dual (outer) problem

$$f_0(\mathbf{z}^*) = d(\lambda^*) = \max_{\lambda\in\mathbb{R}^m_+} d(\lambda) \tag{3-7}$$

and we use λ^* to denote the optimal solution of (3-7).

3-2-2 Optimal Step Size

Because $f_0(\mathbf{z})$ is strongly convex, it can be proved that the gradient of the dual function $d(\lambda)$ is given by the following:

$$\nabla d(\lambda) = g(\mathbf{z}(\lambda)) = G\mathbf{z}(\lambda) + Ex_0 + g \tag{3-8}$$

and it is Lipschitz continuous with constant $L_d = \left\|GH^{-1}G^T\right\|_2$ [12].

This Lipschitz constant is tighter than that provided in [26] and [29].

3-2-3 Algorithm Description

Applying the fast gradient scheme (Algorithm 1) to the dual of Problem (2-11) leads to Algorithm 2 [25].

Algorithm 2 Dual fast gradient method [25].

Require: $H,\ h,\ x_0,\ G,\ E,\ g,\ \lambda_0 = \hat{\lambda}$ and L_d.
for $k = 1, ..., k_{\max}$ **do**
1. Compute $\mathbf{z}_k = \arg\min_{\mathbf{z}\in\mathbb{R}^n} \mathcal{L}(\mathbf{z}, \lambda)$.
2. Compute $\hat{\lambda}_k = \left[\lambda_k + \frac{1}{L_d}\nabla d(\lambda_k)\right]_+$.
3. Compute $\lambda_{k+1} = \frac{k+1}{k+3}\hat{\lambda}_k + \frac{2}{L_d(k+3)}\left[\sum_{j=0}^{k}\frac{j+1}{2}\nabla d(\lambda_j)\right]_+$.

end for
return $\mathbf{z}_{k_{\max}}$.

In particular, the initial Lagrange multiplier λ can be set to any value greater or equal than zero, i.e., $\lambda^0 \geq 0$. A complete analysis for Algorithm 2 starting from $\lambda^0 \neq 0$ can be found in [26]. For simplicity of exposition, we start Algorithm 2 here with $\lambda^0 = 0$. At every iteration, the algorithm first computes a minimizer of (3-6) (Step 1). Then, it performs a linear update of the dual variables (Step 2-3).

Define the following average sequence for the primal variables:

$$\hat{\mathbf{z}}_k = \sum_{j=0}^{k}\frac{2(j+1)}{(k+1)(k+2)}\mathbf{z}_j \tag{3-9}$$

then the following theorem on estimation of primal feasibility violation and suboptimality can be derived.

Theorem 3.1. *[25] Let $f_0(\boldsymbol{z})$ be strongly convex the sequence $\left(\boldsymbol{z}_k, \hat{\lambda}_k, \lambda_k\right)$ be generated by Algorithm 2, and $\hat{\boldsymbol{z}}_k$ be given by* (3-9)*. Then, an estimate on primal feasibility violation for the original problem* (2-11) *is given by the following:*

$$\left\|[G\hat{\boldsymbol{z}}_k + Ex_0 + g]_+\right\|_2 \leq \frac{8L_dR_d}{(k+1)^2} \tag{3-10}$$

where $R_d = \|\lambda^*\|$. Moreover, an estimate on primal suboptimality is given by the following:

$$-\frac{8L_dR_d}{(k+1)^2} \leq f_0(\hat{\boldsymbol{z}}_k) - f_0(\boldsymbol{z}^*) \leq 0 \tag{3-11}$$

From Theorem 3.1, it is easy to see that, dual fast gradient method converges in a rate of $\mathcal{O}(\frac{1}{k^2})$, and if we take $k_{\max} = \left\lfloor 2\sqrt{\frac{2L_dR_d}{\epsilon}} \right\rfloor$, the following estimates on feasibility violation and primal suboptimality hold:

$$\left\| [G\hat{\mathbf{z}}_k + Ex_0 + g]_+ \right\| \leq \epsilon \quad \text{and} \quad -R_d\epsilon \leq f_0(\hat{\mathbf{z}}_k) - f_0(\mathbf{z}^*) \leq 0 \tag{3-12}$$

Therefore, after $k_{\max}$ iterations, Algorithm 2 is able to find an ϵ-solution of MPC problem (2-11).

Chapter 4

Interior-Point Methods for Quadratic Programming

Today interior-point (IP) methods are among the most widely used numerical second-order methods for solving medium- and large-scale convex optimization problems [9]. This approach was developed by Karmarkar in his seminal paper in 1984 on solving linear programming problems [18], and it was generalized to nonlinear convex problems afterwards. Compared with other second-order solvers, such as, active set methods, interior-point methods have a particular advantage given that they are largely insensitive to the number of active constraints [1]. Moreover, with interior-point methods, one pays a fixed price for the factorization of the KKT matrix regardless of the number of active constraints. Hence, as the number of constraints increases, the interior-point solver becomes more competitive.

Consider the QP formed in (2-11) that arises from MPC applications. Interior-point methods solve it by applying Newton's method to a sequence of equality constrained problems, or to a sequence of modified (relaxed) versions of the KKT conditions.

In this chapter, first we summarize different variations of interior-point methods, i.e., primal barrier and primal-dual interior-point methods, based on [5, 9]. Loosely speaking, interior-point methods can be divided into a large number of different categories. In this thesis, we focus on primal barrier method and feasible starting primal-dual interior-point method in [5, Chapter 11], and infeasible starting primal-dual interior-point method in [9, Chapter 4]. In Sections 4-1 and 4-2, the basic knowledges of interior-point methods are summarized. In Section 4-3, we focus on the application of interior-point method solving QPs, e.g., Problem (2-11), in MPC problems. And specifically, we present an analysis of relations between search directions in each variations when solving the QP involved in a MPC problem. With these analysis, in Section 4-4, we are able to draw the conclusion that, when the parameters required in interior-point methods are selected according to some guidances (which is the common setting as well), the search directions of these variations coincide. Finally, convergence analysis is given for interior-point method solving QPs in MPC problems with inequality constraints.

4-1 Primal Barrier Method

The main idea of the barrier method is to handle the inequality constraints in Problem (2-11) by using a *barrier function*. Since we use condensed formulation to form the QP problem in MPC scheme, the re-formulated optimization problem is an unconstrained problem in barrier method.

$$\min_{\mathbf{z}\in\mathbb{R}^n} \; F_b(\mathbf{z}) = \theta f_0(\mathbf{z}) + \phi(\mathbf{z}) \tag{4-1}$$

where θ is called *barrier parameter*, and $\phi(\mathbf{z})$ is the logarithmic barrier function:

$$\phi(\mathbf{z}) = -\sum_{i=1}^{m} \log\left(-g_i(\mathbf{z})\right) \tag{4-2}$$

which is convex and twice differentiable. For future reference, the gradient and Hessian of the logarithmic barrier function $\phi(\mathbf{z})$ are given by the following:

$$\nabla\phi(\mathbf{z}) = -\sum_{i=1}^{m} \frac{1}{g_i(\mathbf{z})}\nabla g_i(\mathbf{z}) \tag{4-3a}$$

$$\nabla^2\phi(\mathbf{z}) = \sum_{i=1}^{m} \frac{1}{g_i(\mathbf{z})^2}\nabla g_i(\mathbf{z})\nabla g_i(\mathbf{z})^{\mathrm{T}} - \sum_{i=1}^{m} \frac{1}{g_i(\mathbf{z})}\nabla^2 g_i(\mathbf{z}) \tag{4-3b}$$

Since in condensed formulation, the equality constraints on state dynamics are eliminated, at the optimum, the following first-order optimality condition has to be satisfied:

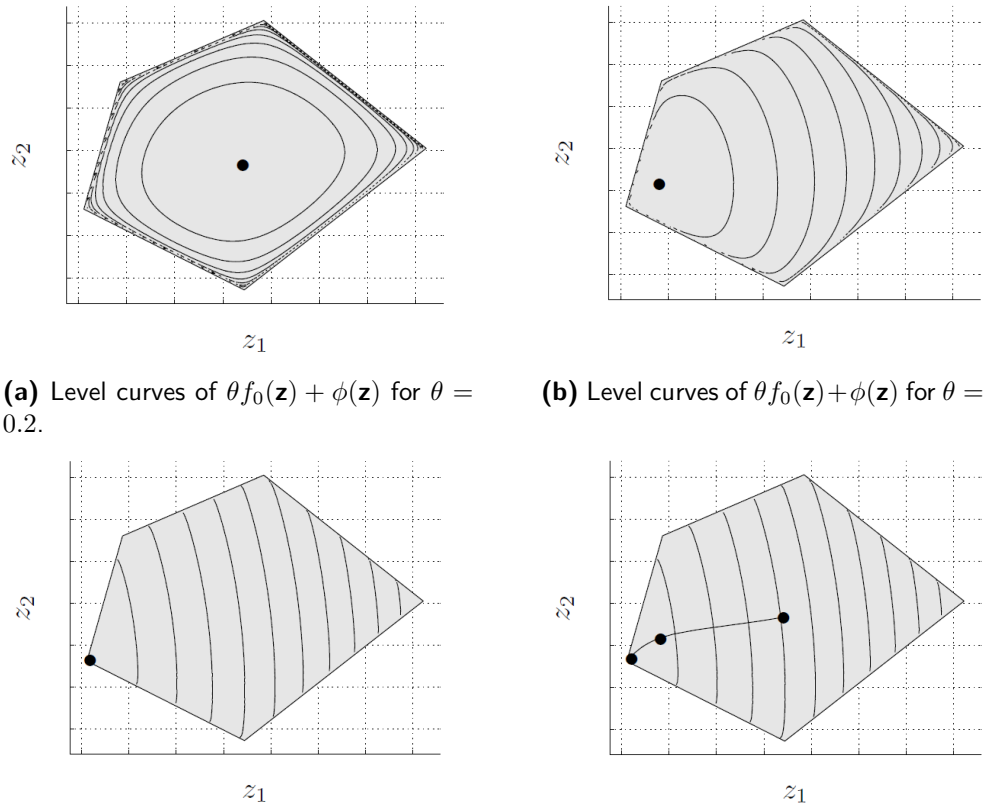
$$\theta\nabla f_0(\mathbf{z}) + \nabla\phi(\mathbf{z}) = 0 \tag{4-4}$$

4-1-1 Central Path

The solution $\mathbf{z}^*(\theta)$ of (4-4) is called *central point* and is located on a curve called *central path*. Figure 4-1 shows the central path of a QP problem. As a result, $\mathbf{z}^*(\theta)$ if feasible for Problem (4-1), but it differs from $\mathbf{z}^*$ since we have perturbed the objective function by the barrier term. The purpose of the barrier function $\phi(\mathbf{z})$ is to "trap" an optimal solution of problem (4-1) in the feasible set [9]. The barrier function in (4-2) takes on the value $+\infty$ whenever $g_i(\mathbf{z}) > 0$ for some i. Therefore, when the objective function $F_b(\mathbf{z})$ in Problem (4-1) is minimized, the inequality constraints are guaranteed to be satisfied.

4-1-2 Outline of Barrier Method

The barrier method starts from a sufficiently large θ_0. In QPs that arise from a MPC problem, the objective function in (4-1) is strongly convex and twice continuously differentiable. In this way, a primal solution $\mathbf{z}^*(\theta_0)$ corresponding to θ_0 can be computed via Newton's method. After $\mathbf{z}_0$ has been computed, the barrier parameter θ is increased by a constant factor, i.e., $\theta_1 = \delta\theta_0,\ \delta > 1$ and $\mathbf{z}^*(\theta_1)$ is computed. This procedure is repeated until θ has been sufficiently increased. Moreover, it can be shown that, under mild conditions, $\mathbf{z}^*(\theta) \to \mathbf{z}^*$ for $\theta \to \infty$.



(a) Level curves of $\theta f_0(\mathbf{z}) + \phi(\mathbf{z})$ for $\theta = 0.2$.

(b) Level curves of $\theta f_0(\mathbf{z}) + \phi(\mathbf{z})$ for $\theta = 4$.

(c) Level curves of $\theta f_0(\mathbf{z}) + \phi(\mathbf{z})$ for $\theta = 8$.

(d) Central path starting at analytic center and moving towards the optimal solution as θ increases.

Figure 4-1: Central path of a QP [9].

For each $\theta > 0$, the first-order optimality condition (4-4) of Problem (4-1) has a unique solution. The central points $\mathbf{z}^*(\theta)$ must satisfy the following *modified* KKT conditions derived from Problem (2-11) [5, § 11.3.4]:

$$\nabla f_0(\mathbf{z}) + Dg(\mathbf{z})^{\mathrm{T}}\lambda = 0 \tag{4-5a}$$

$$-\operatorname{\mathbf{diag}}(g(\mathbf{z}))\,\lambda = \frac{1}{\theta}\mathbf{1}_m \tag{4-5b}$$

$$g(\mathbf{z}) \leq 0 \tag{4-5c}$$

$$\lambda \geq 0 \tag{4-5d}$$

where $\lambda_i = -\frac{1}{\theta g_i(\mathbf{z})}$ denotes the dual points on the central path, , and $Dg(\mathbf{z})$ is the derivative matrix of the inequality constraint function $g(\mathbf{z})$

$$Dg(\mathbf{z}) = \begin{bmatrix} \nabla g_1(\mathbf{z})^{\mathrm{T}} \\ \vdots \\ \nabla g_m(\mathbf{z})^{\mathrm{T}} \end{bmatrix}$$

Therefore, solving nonlinear system (4-4) can be interpreted as solving the linear equations obtained by linearizing (4-4):

$$H_{\text{bar}}\Delta\mathbf{z}_{\text{bar}} = -\nabla F_b(\mathbf{z}) \tag{4-6}$$

where $\Delta\mathbf{z}_{\text{bar}}$ is the search direction computed by Newton's method. Subscript $(\cdot)_{\text{bar}}$ stands for barrier methods and

$$H_{\text{bar}} = \theta\nabla^2 f_0(\mathbf{z}) + \sum_{i=1}^{m} \frac{1}{g_i(\mathbf{z})^2}\nabla g_i(\mathbf{z})\nabla g_i(\mathbf{z})^{\mathrm{T}} \tag{4-7a}$$

$$\nabla F_b(\mathbf{z}) = \theta\nabla f_0(\mathbf{z}) - \sum_{i=1}^{m} \frac{1}{g_i(\mathbf{z})}\nabla g_i(\mathbf{z}) \tag{4-7b}$$

where we use the fact that $\sum_{i=1}^{m} \frac{1}{g_i(\mathbf{z})}\nabla^2 g_i(\mathbf{z}) = 0$ since $g(\mathbf{z})$ is affine.

In barrier method, the primal variable $\mathbf{z}$ is updated by the following sequence:

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \rho_k\Delta\mathbf{z}_{\text{bar}}, \quad k = 0, ... \tag{4-8}$$

where ρ_k is the step size which can be obtained by *backtracking line search* approach. The search direction is obtained by solving the linearized system (4-6). A basic version of the method is described in Algorithm 3.

Algorithm 3 Primal Barrier Method for QPs.

Require: H, h, x_0, G, E, g, strictly feasible initial iterate $\mathbf{z}_0$, i.e., $g(\mathbf{z}_0) < 0$, θ_0, $\delta > 1$, tolerance $\epsilon > 0$.
repeat
1. Compute Newton direction $\Delta\mathbf{z}_{k,\text{bar}}$ by solving (4-6).
3. Backtracking line search $\rho_k := 1$.
while $f_0(\mathbf{z} + \rho_k\Delta\mathbf{z}_{k,\text{bar}}) > f_0(\mathbf{z}) + \alpha\rho_k\nabla f_0(\mathbf{z})\Delta\mathbf{z}_{k,\text{bar}}$ **do**
$\rho_k := \beta\rho_k$.
end while3. Update $\mathbf{z}_{k+1} = \mathbf{z}_k + \rho_k\Delta\mathbf{z}_{k,\text{bar}}$.
4. Increase barrier parameter $\theta_{k+1} = \delta\theta_k$.
until stopping criterion $m/\theta_k \leq \epsilon$
return point close to the optimal solution $\mathbf{z}^*$.

The barrier method is a feasible method, which means that it starts with a feasible point and all iterates remain feasible.

4-2 Primal-Dual Interior-Point Methods

The general idea of primal-dual interior-point methods is to solve the KKT conditions by using a modified version of Newton's method. The nonlinear KKT equations represent necessary and sufficient conditions for optimality for convex problems [9].

In primal-dual interior-point methods, we handle the inequality constraints by Lagrange multipliers. Recall the Lagrangian defined in (3-4), the primal-dual interior-point methods solve the following *relaxed* KKT conditions:

$$\nabla f_0(\mathbf{z}) + Dg(\mathbf{z})^{\mathrm{T}}\lambda = 0 \tag{4-9a}$$

$$g(\mathbf{z}) + s = 0 \tag{4-9b}$$

$$S\lambda = \tau \mathbf{1}_m \tag{4-9c}$$

$$(s, \lambda) > 0 \tag{4-9d}$$

where λ is the Lagrange multiplier and $s \in \mathbb{R}^m$ is the slackness variable and $S = \mathbf{diag}(s)$. The variables $\mathbf{z}$ and s are from the primal, the variable λ from the dual space. If a primal-dual pair of variables $(\mathbf{z}^*, \lambda^*, s^*)$ is found that satisfies these conditions, the corresponding primal variables $(\mathbf{z}^*, s^*)$ are optimal for Problem (2-11).

We introduce the slackness variable s since primal-dual interior-point methods accept infeasible iterates by updating this variable at the same time. The difference between feasible and infeasible iterates will be discussed afterwards.

The parameter τ in (4-9c) modifies the search direction by balancing two extremes [9]:

- Affine scaling direction. $\tau = 0$. The search direction is the pure Newton direction that aims at satisfying the original KKT condition based on a linearization. No centering behavior is performed.
- Centering direction. $\tau = \mu$. A centering step does not decrease the average value, so no progress towards the solution is made. And

  $$\mu = \frac{s^{\mathrm{T}}\lambda}{m} \tag{4-10}$$

  denotes the average duality gap. Therefore, $\mu \to 0$ implies $\tau \to 0$.

At each iteration in primal-dual interior-point method, a *trade-off* between the two directions is achieved by setting

$$\tau = \kappa\mu \tag{4-11}$$

which means at iteration k, we set

$$\tau_{k+1} = \kappa\mu_k \tag{4-12}$$

with centering parameter $\kappa \in (0, 1)$, which allows the algorithm to process towards the optimal solution.

4-2-1 Variations of Primal-Dual Interior-Point Methods

Different variations exist among all kinds of widely-used primal-dual interior-point methods. In this work, we mainly discuss two variations of this method. The first variation is presented in [5, §11.7]. It is a basic version of primal-dual interior-point method. Although the proof of convergence of this variation is not provided in [5], we can derive the detailed convergence proof given the convergence analysis of standard Newton method in [5, Chapter 9] for QP problems in MPC. We do not use this variation in our research. It is mentioned here because based on [5] we will make comparison between this variation and the one we rely on to obtain the convergence results for our solver.

• *Feasible Start Primal-Dual Interior-Point Method [5]*

Define the *residual* based on the modified KKT condition in (4-9):

$$r_\tau(\mathbf{z},\lambda)=\begin{bmatrix}\nabla f_0(\mathbf{z})+Dg(\mathbf{z})^{\mathrm{T}}\lambda\\ -\Lambda g(\mathbf{z})-\tau\mathbf{1}_m\end{bmatrix}=\begin{bmatrix}r_{\mathrm{dual}}\\ r_{\mathrm{cent,f}}\end{bmatrix} \tag{4-13}$$

where $\Lambda=\mathbf{diag}(\lambda)$ and we use the following relation obtained from (4-9b):

$$s=-g(\mathbf{z})$$

Our goal is to drive this residual variable $r_\tau(\mathbf{z},\lambda)$ to zero. Consider the Newton step for solving the nonlinear equations $r_\tau(\mathbf{z},\lambda)=0$ for fixed τ and denote the current point and Newton step as

$$\varphi=(\mathbf{z},\lambda),\qquad \Delta\varphi=(\Delta\mathbf{z},\Delta\lambda)$$

then the Newton step can be characterized by

$$r_\tau(\varphi+\Delta\varphi)\approx r_\tau(\varphi)+Dr_\tau(\varphi)\Delta\varphi=0 \tag{4-14}$$

Therefore, the primal-dual search direction can be computed by the following equation:

$$\underbrace{\begin{bmatrix}\nabla^2 f_0(\mathbf{z}) & Dg(\mathbf{z})^{\mathrm{T}}\\ -\Lambda Dg(\mathbf{z}) & -\mathbf{diag}(g(\mathbf{z}))\end{bmatrix}}_{Dr_\tau(\varphi)}\underbrace{\begin{bmatrix}\Delta\mathbf{z}_{\mathrm{pd}}\\ \Delta\lambda_{\mathrm{pd}}\end{bmatrix}}_{\Delta\varphi_{\mathrm{pd}}}=-\begin{bmatrix}r_{\mathrm{dual}}\\ r_{\mathrm{cent,f}}\end{bmatrix} \tag{4-15}$$

where we use $\sum_{i=1}^m\lambda_i\nabla^2 g_i(\mathbf{z})=0$ since $g(\mathbf{z})$ is affine.

• *Infeasible Start Primal-Dual Interior-Point Method [9]*

Instead of substituting (4-9b) into (4-9c) and defining the primal-dual search direction by (4-15), we keep (4-9b) and define the residual variable in the following way:

$$r_\tau(\mathbf{z},\lambda,s)=\begin{bmatrix}\nabla f_0(\mathbf{z})+Dg(\mathbf{z})^{\mathrm{T}}\lambda\\ g(\mathbf{z})+s\\ S\lambda-\tau\mathbf{1}_m\end{bmatrix}=\begin{bmatrix}r_{\mathrm{dual}}\\ r_{\mathrm{pri}}\\ r_{\mathrm{cent}}\end{bmatrix} \tag{4-16}$$

Linearizing (4-16) at current iterate $\zeta=(\mathbf{z},\lambda,s)$ to obtain the equation for search direction:

$$\underbrace{\begin{bmatrix}\nabla^2 f_0(\mathbf{z}) & Dg(\mathbf{z})^{\mathrm{T}} & 0\\ Dg(\mathbf{z}) & 0 & I_m\\ 0 & S & \Lambda\end{bmatrix}}_{Dr_\tau(\zeta)}\underbrace{\begin{bmatrix}\Delta\mathbf{z}_{\mathrm{pd}}\\ \Delta\lambda_{\mathrm{pd}}\\ \Delta s_{\mathrm{pd}}\end{bmatrix}}_{\Delta\zeta_{\mathrm{pd}}}=-\begin{bmatrix}r_{\mathrm{dual}}\\ r_{\mathrm{pri}}\\ r_{\mathrm{cent}}\end{bmatrix} \tag{4-17}$$

It is obvious that in both primal-dual search directions $\Delta\varphi_{\mathrm{pd}}$ and $\Delta\zeta_{\mathrm{pd}}$, the primal and dual search direction are coupled through the coefficient matrix and the residuals. In this way, we can derive a relation between primal and dual search direction and make a comparison between search directions in the two mentioned interior-point methods.

Algorithm 4 Infeasible Start Primal-Dual Interior Method for QPs.

Require: $H,\ h,\ x_0,\ G,\ E,\ g$, initial iterate $\mathbf{z}_0$, $\lambda_0 > 0$ and $s_0 > 0$, centering parameter $\kappa \in (0,1)$, backtracking line search parameters $\alpha \in (0, 0.5),\ \beta \in (0,1)$, tolerance $\epsilon > 0$.

repeat

1. Determine $\tau_{k+1} = \mu_{k+1} = \kappa\mu_k$.
2. Compute primal-dual search direction $\Delta\zeta_{k,\text{pd}} = (\Delta\mathbf{z}_{k,\text{pd}}, \Delta\lambda_{k,\text{pd}}, \Delta s_{k,\text{pd}})$ solving (4-17).
3. Backtracking line search $\rho_k := 1$.

while $f_0(\mathbf{z} + \rho_k\Delta\mathbf{z}_{k,\text{pd}}) > f_0(\mathbf{z}) + \alpha\rho_k\nabla f_0(\mathbf{z})\Delta\mathbf{z}_{k,\text{pd}}$ **do**

$\rho_k := \beta\rho_k$.

end while

4. Update $\zeta_{k+1} = \zeta_k + \rho_k\Delta\zeta_{k,\text{pd}}$ where $\rho_k > 0$ is the step size.

until stopping criterion $\mu \leq \epsilon$.

return point close to the optimal solution $\mathbf{z}^*$.

4-3 Comparison Between Search Directions for Interior-Point Methods

In [5, §11.7.1], the authors have proved that for general convex optimization problem with equality constraints, as long as the equality constraints are satisfied, the Newton step in barrier method, e.g., as shown in (4-6), coincide with the feasible Newton step. Therefore, with condensed formulation, the search direction in primal-dual interior method used to solve the QP problem derived from a general MPC problem coincides with the search direction in standard Newton step when initialized in a proper way, e.g., with a strictly feasible starting point, and the parameters required, e.g., barrier and centering parameter, are reasonably selected. Hence, it is important to analyze the relation between search directions in primal-dual interior-point methods and in Newton's method in order to derive our convergence analysis for primal-dual interior-point methods solving QP problems that arise from MPC scheme.

First of all, a proof to show the coincidence of primal-dual search direction and barrier search direction is given considering the situation without equality constraints based on the variation of primal-dual interior-point method given in Boyd's book [5, Chap. 11]. Latter, the relation between search directions obtained from two variations of primal-dual interior-point is proposed. Based on these analysis, in next section, a two-phase convergence result of primal-dual interior-point method is drawn.

4-3-1 Comparison Between Barrier and Primal-Dual Interior-Point Methods

The primal-dual search directions are closely related to the search directions used in logarithmic barrier method. Considering the linear equations (4-15), we can eliminate the dual search direction variable $\Delta\lambda_{\text{pd}}$ using the following equation coming from the second block of (4-15):

$$\Delta\lambda_{\text{pd}} = -\mathbf{diag}(g(\mathbf{z}))^{-1}\Lambda Dg(\mathbf{z})\Delta\mathbf{z}_{\text{pd}} + \mathbf{diag}(g(\mathbf{z}))^{-1}r_{\text{cent}} \tag{4-18}$$

Substituting this into the first block of equations gives

$$\begin{aligned} H_{\text{pd}}\Delta\mathbf{z}_{\text{pd}} &= -\left[r_{\text{dual}} + Dg(z)^{\text{T}}\mathbf{diag}(g(\mathbf{z}))^{-1}r_{\text{cent}}\right] \\ &= -\left[\nabla f_0(\mathbf{z}) - \tau\sum_{i=1}^{m}\frac{1}{g_i(\mathbf{z})}\nabla g_i(\mathbf{z})\right] \end{aligned} \tag{4-19}$$

where

$$H_{\text{pd}} = \nabla^2 f_0(\mathbf{z}) - \sum_{i=1}^{m}\frac{\lambda_i}{g_i(\mathbf{z})}\nabla g_i(\mathbf{z})\nabla g_i(\mathbf{z})^{\text{T}} \tag{4-20}$$

If we compare the two systems of equations (4-6) and (4-19), we can see that they are similar. And the coefficient matrices (4-20) and (4-7a) have the same structure. If we divide the system of equations (4-6) by θ, we can obtain

$$\frac{1}{\theta}H_{\text{bar}}\Delta\mathbf{z}_{\text{bar}} = -\left[\nabla f_0(\mathbf{z}) - \frac{1}{\theta}\sum_{i=1}^{m}\frac{1}{g_i(\mathbf{z})}\nabla g_i(\mathbf{z})\right] \tag{4-21}$$

where

$$\frac{1}{\theta}H_{\text{bar}} = \nabla^2 f_0(\mathbf{z}) + \sum_{i=1}^{m}\frac{1}{\theta g_i(\mathbf{z})^2}\nabla g_i(\mathbf{z})\nabla g_i(\mathbf{z})^{\text{T}} \tag{4-22}$$

Therefore, when $\mathbf{z}$ and λ satisfy

$$-g_i(\mathbf{z})\lambda_i = \frac{1}{\theta} \tag{4-23}$$

the coefficient matrices and also the search directions coincide.

4-3-2 Comparison of Variations of Primal-Dual Interior-Point Methods

From the third block in (4-17), we can find the following coupled relationship between Δs_{pd} and $\Delta\lambda_{\text{pd}}$:

$$\Delta s_{\text{pd}} = \Lambda^{-1}\left(-r_{\text{cent}} - S\Delta\lambda_{\text{pd}}\right) \tag{4-24}$$

By substituting this relation into the second block of (4-17), the relation between primal search direction $\Delta\mathbf{z}_{\text{pd}}$ and dual search direction $\Delta\lambda_{\text{pd}}$ can be found:

$$\begin{aligned} Dg(\mathbf{z})\Delta\mathbf{z}_{\text{pd}} + I_m\Delta s_{\text{pd}} &= Dg(\mathbf{z})\Delta\mathbf{z}_{\text{pd}} + I_m\cdot\Lambda^{-1}\left(-r_{\text{cent}} - S\Delta\lambda_{\text{pd}}\right) \\ &= Dg(\mathbf{z})\Delta\mathbf{z}_{\text{pd}} - \Lambda^{-1}\left(r_{\text{cent}} + S\Delta\lambda_{\text{pd}}\right) \\ &= -r_{\text{pri}} \end{aligned}$$

Thus,

$$\Delta\lambda_{\text{pd}} = S^{-1}\Lambda\left(Dg(\mathbf{z})\Delta\mathbf{z}_{\text{pd}} - \Lambda^{-1}r_{\text{cent}} + r_{\text{pri}}\right) \tag{4-25}$$

Then, using the first block, we can derive the following equation for primal search direction $\Delta\mathbf{z}_{\text{pd}}$:

$$\begin{aligned}
\nabla^2 f_0(\mathbf{z})\Delta\mathbf{z}_{\text{pd}} &= -r_{\text{dual}} - Dg(\mathbf{z})^{\text{T}}\Delta\lambda_{\text{pd}} \\
&= -r_{\text{dual}} - Dg(\mathbf{z})^{\text{T}} \cdot S^{-1}\Lambda\left(Dg(\mathbf{z})\Delta\mathbf{z}_{\text{pd}} - \Lambda^{-1}r_{\text{cent}} + r_{\text{pri}}\right) \\
&\Downarrow \\
\hat{H}_{\text{pd}}\Delta\mathbf{z}_{\text{pd}} &= -\left[r_{\text{dual}} - Dg(\mathbf{z})^{\text{T}}S^{-1}\left(r_{\text{cent}} - \Lambda r_{\text{pri}}\right)\right] \\
&= -\left[r_{\text{dual}} + Dg(\mathbf{z})^{\text{T}}S^{-1}\left(\Lambda(g(\mathbf{z}) + s) - (S\lambda - \tau\mathbf{1}_m)\right)\right] \\
&= -\left[r_{\text{dual}} - Dg(\mathbf{z})^{\text{T}}S^{-1}\left(-\tau\mathbf{1}_m - \Lambda g(\mathbf{z})\right)\right] \\
&= -\left[r_{\text{dual}} - Dg(\mathbf{z})^{\text{T}}S^{-1}r_{\text{cent,f}}\right]
\end{aligned} \tag{4-26}$$

where

$$\hat{H}_{\text{pd}} = \nabla^2 f_0(\mathbf{z}) + Dg(\mathbf{z})^{\text{T}}S^{-1}\Lambda Dg(\mathbf{z}) \tag{4-27}$$

If we compare systems (4-19) with (4-26) and the Hessian matrix (4-20) and (4-27), we can conclude that when $g(\mathbf{z}) = -s$, that is, when the iterates are feasible, these two variations are exactly the same.

4-4 Coincidence of Barrier and Primal-Dual Interior-Point Methods

Theorem 4.1. *Consider Problem* (2-11) *derived from MPC problem, indeed the primal-dual interior-point method coincides the logarithmic barrier method with a strictly feasible starting point by selecting*

$$\theta = \frac{1}{\mu} \qquad \textit{and} \qquad \kappa = \frac{1}{\delta} \tag{4-28}$$

Proof. In order to prove the coincidence of search directions in these two methods, we have to prove that the primal variable $\mathbf{z}$ and dual variable λ satisfy (4-23) at each iteration.

Look into details at iteration k:

Suppose that, in barrier method, we increase θ by a constant factor $\kappa > 1$ to guarantee that the inequality constraint $g(\mathbf{z}) \leq 0$ is satisfied recursively.

And in primal-dual interior-point method, at iteration k, the complementarity slackness condition (4-9c) can be denoted as

$$s_{k,i}\lambda_{k,i} = \tau_k, \quad i = 1, ..., m \tag{4-29}$$

Set $\tau_k = \mu_k$, then at each iteration, in order to guarantee centering and convergence, we decrease μ_k by the centering parameter $\delta \in (0, 1)$. Thus,

$$\mu_{k+1} = \delta\mu_k < \mu_k \tag{4-30}$$

By setting $\theta = \frac{1}{\mu}$, at iteration k,

$$\theta_k = \frac{1}{\mu_k} \tag{4-31}$$

then at next iteration,

$$\theta_{k+1} = \frac{1}{\mu_{k+1}} = \frac{1}{\sigma\mu_k} = \frac{1}{\sigma}\theta_k = \kappa\theta_k \tag{4-32}$$

Thus, if we choose $\kappa = \frac{1}{\delta} \in (1, \infty)$, we can derive the following equality from (4-9c) and (4-31)

$$-g_i(\mathbf{z}_k)\lambda_{k,i} = \tau_k = \mu_k = \frac{1}{\theta_k} \tag{4-33}$$

Therefore, the search direction of primal-dual interior-point method and logarithmic barrier method coincides. □

4-5 Convergence Analysis of Primal-Dual Interior-Point Methods

From the proofs in Subsection 4-3-2 and Section 4-4, we can conclude that, the search direction of primal-dual interior-point method in Algorithm 4 when solving Problem (2-11) coincides with search direction in Newton's method solving Problem (4-1) if Algorithm 4 is started with a strictly feasible initial point and Theorem 4.1 satisfied. Therefore, the convergence analysis of primal-dual interior-point method can be done in primal space based on the convergence of standard Newton's method.

Recall the optimization problem in barrier method in (4-1):

$$\min_{\mathbf{z}\in\mathbb{R}^n} \; F_b(\mathbf{z}) = \theta f_0(\mathbf{z}) + \phi(\mathbf{z})$$

Since the search directions of logarithmic barrier method and primal-dual interior-point method coincide, according to convergence analysis in [5, §11.3.3], under some basic assumptions, the convergence of primal-dual interior-point method in Algorithm 4 can be analyzed using the convergence result of logarithmic barrier method, which is equivalent to the convergence of standard Newton's method when no equality constraints are involved.

We need the following assumptions:

Assumption 4.1 (Closed Sub-Level Set)**.** The sub-level set $\mathbb{S} = \{\mathbf{z} | \mathbf{z} \in \mathbf{dom}(F_b), F_b(\mathbf{z}) \leq F_b(\mathbf{z}_0)\}$ is closed, where $\mathbf{z}_0 \in \mathbf{dom}(F_b)$.

Assumption 4.2 (Strong Convexity)**.** On the set $\mathbb{S}$, we have

$$m_{F_b} I \preceq \nabla^2 F_b(\mathbf{z}) \leq M_{F_b} I$$

Assumption 4.3 (Lipschitz Continuity)**.** For $\mathbf{z},\ \tilde{\mathbf{z}} \in \mathbb{S}$, $\nabla^2 F_b(\mathbf{z})$ satisfies the Lipschitz condition

$$\left\| \nabla^2 F_b(\mathbf{z}) - \nabla^2 F_b(\tilde{\mathbf{z}}) \right\|_2 \leq L_{F_b H} \left\| \mathbf{z} - \tilde{\mathbf{z}} \right\|_2$$

Under Assumptions 4.1-4.3, the convergence of primal-dual interior-point method can be given in the following theorem referring to the convergence of logarithmic barrier method.

Theorem 4.2 (Convergence in Primal Space [5])**.** *Under Assumptions 4.1-4.3, there exist numbers η_p and γ_p with $0 < \eta_p \leq \frac{m_{F_b}^2}{L_{F_b H}}$ and $\gamma_p > 0$ such that the following hold.*

- *Damped Newton Phase. If $\|\nabla F_b(\boldsymbol{z})\|_2 > \eta_p$, then for all $k \geq 0$*

$$F_b(\boldsymbol{z}_{k+1}) - F_b(\boldsymbol{z}_k) \leq -\gamma_p \tag{4-34}$$

- *Pure Newton Phase. If $\|\nabla F_b(\boldsymbol{z})\|_2 \leq \eta_p$, then backtracking line search selects the step size $\rho_{\bar{k}} = 1$. For all $\bar{k} > 0$*

$$\frac{L_{F_b H}}{2m_{F_b}^2} \left\|\nabla F_b(\boldsymbol{z}_{\bar{k}+1})\right\|_2 \leq \left(\frac{L_{F_b H}}{2m_{F_b}^2} \|\nabla F_b(\boldsymbol{z}_{\bar{k}})\|_2\right)^2 \tag{4-35}$$

and hence

$$F_b(\boldsymbol{z}_{\bar{k}}) - f_0(\boldsymbol{z}^*) \leq \frac{1}{2m_{F_b}} \|\nabla F_b(\boldsymbol{z}_{\bar{k}})\|_2^2 \leq \frac{3m_{F_b}}{L_{F_b H}} \left(\frac{1}{2}\right)^{2^{\bar{k}+1}} \tag{4-36}$$

Proof. The proof of Theorem 4.2 can be found in [5, §9.5.3]. □

Note that (4-36) shows that, in pure Newton phase quadratic convergence can be achieved from Theorem 4.1 and if (4-23) are satisfied, the search direction of primal-dual interior-point method coincide with the search direction in standard Newton's method when the iterates satisfy (4-9b), that is, Algorithm 4 has to be initialized with a strictly feasible solution. However, a strictly feasible solution is hard to find when initializing the solver without using terminal set, especially when we have active constraints. In the following chapter, we will show that, we can overcome the problem caused by the strictly feasible starting point requirement, and we are able to propose novel initialization strategies for primal-dual interior-point method that allows one to find a better starting point for primal-dual interior-point method compared to *warm-starting* techniques.

Chapter 5

Proposed Solvers in Primal and Dual Space

As stated in previous chapters, when the primal and dual variables in primal-dual interior-point methods are initialized appropriately, and when the centering parameter is chosen properly, the search direction of primal-dual interior-point methods coincides with the search direction of standard Newton's method. Therefore, the convergence procedure can be divided into two phases, damped and pure Newton phase. In the second phase, i.e., pure Newton phase, the algorithm converges quadratically with step size equal to one. Considering the convergence characteristic in the two phases, we can derive an outline to propose an initialization strategy for primal-dual interior-point methods.

As detailed in the remainder of the chapter, by relying on dual fast gradient (DFG) method which converges super-linearly, we show how to initialize primal-dual interior-point (PDIP) methods to remove the damped Newton phase, in which the algorithm converges linearly. Moreover, we derive a switching condition, which is the stopping criterion for dual fast gradient method, and an initialization strategy for PDIP to help us enter quadratic convergence phase directly. In this way, computational effort required by backtracking line search can also be saved.

This chapter is organized as follows: in Section 5-1 and 5-2, we proposed two novel primal-dual interior-point algorithms with a switching condition defined in primal space according to analysis in Chapter 4. Generally, we use the following notations:

- $F_b(\cdot) = \theta J(\cdot)+\phi(\cdot)$: the cost function formed with barrier function $\phi(\cdot)$ corresponding to a QP with only inequality constraints. For example, $F_b(\mathcal{P}(\mathbf{z}))$ denotes the cost function formed in barrier method to solve a QP called $\mathcal{P}(\mathbf{z})$ with decision variable $\mathbf{z}$.
- $\eta_p(\cdot)$: a constant related to condition numbers (refer to Theorem 4.2) corresponding to a QP with only inequality constraints.

Based on the analysis derived in Chapter 4 (in primal space) the following outline to derive our novel algorithm can be given:

- DFG Phase. If $\|\nabla F_b(\mathcal{P}(\mathbf{z}))\|_2 \geq \eta_p(\mathcal{P}(\mathbf{z}))$, dual fast gradient solver runs until the $\eta_p(\mathcal{P}(\mathbf{z}))$-solution $(\mathbf{z}_{\text{DFG}}, \lambda_{\text{DFG}})$ is achieved (after $k_{\max}$) iterations.
- Pure Newton Phase. If
$$\left\|\nabla F_b(\mathcal{P}(\mathbf{z}_{\bar{k}+1}))\right\|_2 \leq \eta_p(\mathcal{P}(\mathbf{z}_{\bar{k}})), \tag{5-1}$$
for all $\bar{k} \geq 0$, with $\rho_{\bar{k}} = 1$ and $\eta_p(\mathcal{P}(\mathbf{z}_{\bar{k}}))$-solution $(\mathbf{z}_{\text{DFG}}, \lambda_{\text{DFG}})$, the primal-dual interior-point solver starts by directly entering pure Newton phase and converges quadratically.

The key for our novel algorithm to work successfully lies in deriving a simple switching condition and an efficient initialization strategy between DFG and pure Newton phase. From Chapter 4, in primal space, a strictly feasible starting point for primal-dual interior-point method in the second phase, is required. In this way, we can use the condition in (5-1) derived from primal barrier method to determine the connecting point between damped and pure Newton phase in primal-dual interior-point method. In order to have a strictly feasible starting point for the primal-dual interior-point solver, soft constraint approach [19] in PDIP phase (Section 5-1) and tightening constraint approach [25] in DFG phase (Section 5-2) are investigated. An initialization strategy for the PDIP phase are proposed for each approach. These two approaches works in most situations. However, the primal switching condition in (5-1) causes issues (undefined structures) when there are constraints active at the optimum.

To overcome this problem, in Section 5-4, we propose a new convergence analysis of primal-dual interior-point method in dual space based on one assumption which can be made on the solution $(\mathbf{z}_{\text{DFG}}, \lambda_{\text{DFG}})$ returned by dual fast gradient solver according to Theorem 3.1. Afterwards, the optimization problem in (2-11) is re-formulated, which leads to a simplified condition for primal-dual interior-point method to enter pure Newton phase. Further more, the proposed strategy does not require any tightening or softening in the MPC formulation.

5-1 Soft Constraints Approach in Primal-Dual Interior-Point Methods

In Chapter 4, we have proved that the coincidence condition of barrier and primal-dual interior-point method requires (4-9b) from KKT condition:

$$g(\mathbf{z}) + s = 0, \quad s > 0$$

which means that primal-dual interior-point method has to be initialized with a strictly feasible starting point. When it is initialized by dual fast gradient method, feasible start cannot be guaranteed without tightening [25]. If we are "lucky" to obtain a primal feasible solution by dual fast gradient method and the condition of pure Newton phase in Theorem 4.2 is active, we can directly enter the second phase of primal-dual interior-point method by initializing it with this primal feasible solution. In another situations, where the solution returned by dual fast gradient is infeasible, then some computational effort is still required by damped Newton phase before primal-dual interior-point method can enter the quadratic convergence phase, reducing the benefits of the initialization.

One way to deal with this problem is introducing soft constraints in primal-dual interior-point method. With soft constraints, primal-dual interior-point method can easily be initialized

using a strictly feasible starting point returned by the solution of dual fast gradient method. Further, with this approach, we are still able to obtain the same optimal solution obtained without softening, using exact penalty functions in the MPC cost.

5-1-1 Overview

Usually, soft constraints are used when we do not know whether a feasible solution exists, that is, to avoid infeasible solutions. In this approach, the constraints in (2-11b) are softened by adding slack variables corresponding to violated or active constraints [7]. The slack variables are added to the MPC cost function and the optimizer searches for a solution which minimizes the original cost function augmented with exact penalty functions while keeping the constraint violation as small as possible [19]. As the value of slack variable converges to zero, the solution obtained with soft constraint approach converges to the optimal solution of the original problem.

Consider the QP problem in (2-11), the soft constraint approach uses slack variables which are defined such that they are non-zero if the corresponding constraints are violated or active. Since in primal-dual interior-point methods, we expect a strictly feasible starting point to satisfied the switching condition in Theorem 4.2, we need to introduce slack variable to soft the active constraints as well. Therefore, the optimization problem can be recast into the following equivalent constrained problem:

$$\min_{\mathbf{z}\in\mathbb{R}^n,\epsilon_s\in\mathbb{R}^{\tilde{m}}} \quad f_0(\mathbf{z}) + \omega\|\epsilon_s\| \tag{5-2a}$$

$$s.t. \quad g(\mathbf{z}) \leq \epsilon_s \tag{5-2b}$$

$$\epsilon_s \geq 0 \tag{5-2c}$$

where $\omega > 0$ is the penalty weight and $\tilde{m}$ denotes the number of violated and active constraints.

The weight on the slack variables in the cost function has to be selected large enough to keep the slack variables at zero, if possible. The following theorem states the condition needed to be satisfied for the objective function in Problem (5-2) being an exact penalty function, and the condition for Problem (5-2) and (2-11) to have the same solution.

Theorem 5.1 ([19]). *If λ^* denotes the optimal Lagrange multiplier for Problem* (2-11)*, the penalty weight satisfies $\omega > \|\lambda^*\|_D$ leads to exact penalty function in Problem* (5-2)*. In addition, if strong duality (Assumption 2.1) holds, the optimal solution of Problem* (5-2) *is equal to the solution of Problem* (2-11)*.*

In Theorem 5.1, the concept of *dual norm* $\|\cdot\|_D$ is used. The dual norm of $\|\cdot\|_1$ is $\|\cdot\|_\infty$ and vice versa, and the dual norm of $\|\cdot\|_2$ is itself. Note that, using the quadratic norm $\|\cdot\|_2^2$ one can express Problem (5-2) as a QP. However, using quadratic norm only does not result in an exact penalty function [19].

5-1-2 Model Predictive Control Problem Reformulation

Recall that our strategy is to initialize the primal-dual interior-point method using the dual fast gradient method. In order to select the initialization of softening parameter, we at first

re-arrange the functions of inequality constraints by defining the following sets:

$$\begin{aligned}
\mathcal{G}_1 &\coloneqq \{i,\ i \in \mathbb{R}^{m_1} \mid i_{k+1} = i_k + 1,\ i_0 = 0 \text{ if } g_j(\mathbf{z}) = \delta_1,\ \delta_1 > 0,\ j = 1, ..., m\} \\
\mathcal{G}_2 &\coloneqq \{i,\ i \in \mathbb{R}^{m_2} \mid i_{k+1} = i_k + 1,\ i_0 = 0 \text{ if } g_j(\mathbf{z}) = \delta_2,\ \delta_2 = 0,\ j = 1, ..., m\} \\
\mathcal{G}_3 &\coloneqq \{i,\ i \in \mathbb{R}^{m_3} \mid i_{k+1} = i_k + 1,\ i_0 = 0 \text{ if } g_j(\mathbf{z}) = \delta_3,\ \delta_3 < 0,\ j = 1, ..., m\}
\end{aligned} \tag{5-3}$$

and define the re-arranged inequality functions as

$$\begin{cases}
g(\mathcal{G}_1) = G_1\mathbf{z} + E_1 x_0 + g_1 \\
g(\mathcal{G}_2) = G_2\mathbf{z} + E_2 x_0 + g_2 \\
g(\mathcal{G}_3) = G_3\mathbf{z} + E_3 x_0 + g_3
\end{cases} \tag{5-4}$$

We can reformulate the optimization problem using the soft constraint approach in the following way:

$$\begin{aligned}
\min_{\mathbf{z}\in\mathbb{R}^n, \epsilon_s \in \mathbb{R}^{\tilde{m}}} \quad & f_0(\mathbf{z}) + \omega_1 ||\epsilon_s||_1 + \omega_2 ||\epsilon_s||_2^2 \\
s.t.\ \ & G_1\mathbf{z} + E_1 x_0 + g_1 \leq \epsilon_1 \\
& G_2\mathbf{z} + E_2 x_0 + g_2 \leq \epsilon_2 \\
& G_3\mathbf{z} + E_3 x_0 + g_3 \leq 0 \\
& \epsilon_s \geq 0
\end{aligned} \tag{5-5}$$

where $\epsilon_s = [\epsilon_1^{\mathrm{T}}\ \epsilon_2^{\mathrm{T}}]^{\mathrm{T}} > 0$, $\omega_1 > ||\lambda^*||_\infty$ with λ^* being the optimal Lagrange multiplier corresponding to inequality constraints in (2-11), $\omega_2 > 0$, $G \in \mathbb{R}^{m\times n}$, $E \in \mathbb{R}^{m\times n_x}$, $g \in \mathbb{R}^m$, $\epsilon_s \in \mathbb{R}^{\tilde{m}}$ and $\tilde{m}$ denotes the number of violated and active constraints.

Problem (5-5) can be written into the standard form of a quadratic programming problem:

$$\begin{aligned}
\min_{\mathbf{z}\in\mathbb{R}^n, \epsilon_s \in \mathbb{R}^{\tilde{m}}} \quad & F_s(\mathbf{z}, \epsilon_s) = f_0(\mathbf{z}) + \omega_1 ||\epsilon_s||_1 + \omega_2 ||\epsilon_s||_2^2 \\
s.t.\ \ & g(\mathbf{z}, \epsilon_s) = \underbrace{\begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}}_{\bar{G}} \mathbf{z} + \underbrace{\begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}}_{\bar{E}} x_0 - \underbrace{\begin{bmatrix} I & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix}}_{\bar{F}} \epsilon_s + \underbrace{\begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}}_{\bar{g}} \leq 0
\end{aligned} \tag{$\mathcal{P}_{\text{soft}}$}$$

Define a new variable $\bar{\mathbf{z}} = [\mathbf{z}^{\mathrm{T}}\ \epsilon_s^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n+\tilde{m}}$, then Problem ($\mathcal{P}_{\text{soft}}$) can be easily written into

$$\min_{\bar{\mathbf{z}}} \quad F_s(\bar{\mathbf{z}}) = \frac{1}{2}\bar{\mathbf{z}}^{\mathrm{T}} H_z \bar{\mathbf{z}} + h_z^{\mathrm{T}} \bar{\mathbf{z}} \tag{5-6a}$$

$$s.t.\ \ G(\bar{\mathbf{z}}) = G_z \bar{\mathbf{z}} + E_z x_0 + g_z \leq 0 \tag{5-6b}$$

where

$$H_z = \begin{bmatrix} H & 0 \\ 0 & \omega_2 I \end{bmatrix}, \quad h_z = \begin{bmatrix} hx_0 \\ \omega_1 \mathbf{1} \end{bmatrix}$$

$$G_z = \left[\begin{array}{c|c} G_1 & -I \\ G_2 & \\ \hline G_3 & 0 \\ 0 & -I \end{array}\right], \quad E_z = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ 0 \end{bmatrix}, \quad g_z = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 0 \end{bmatrix}$$

This QP problem with soft constraint approaching in (5-6) can be solved using Algorithm 4.

Note that in (5-5) we introduce both 1- and 2-norm in order to guarantee exactness of the penalty function and strong convexity of objective function of Problem (5-6), i.e., $H_z \succ 0$.

5-1-3 Initialization Strategy For Primal-Dual Interior-Point Methods

Define $\bar{\lambda} = [\lambda_r^{\mathrm{T}}\ \lambda_{\epsilon_s}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}_+^{m+\tilde{m}}$ for Problem (5-6), where λ_r denotes the **re-arranged** Lagrange multiplier corresponding to the **re-arranged** constraint sets in (5-3) for Problem (2-11). Define the following Lagrangian:

$$\mathcal{L}(\bar{\mathbf{z}}, \bar{\lambda}) = F_s(\bar{\mathbf{z}}) + \bar{\lambda}^{\mathrm{T}} G(\bar{\mathbf{z}})$$

Then we can derive the following relaxed KKT conditions:

$$\nabla_{\bar{\mathbf{z}}} F_s(\bar{\mathbf{z}}) + \nabla_{\bar{\mathbf{z}}} G(\bar{\mathbf{z}})^{\mathrm{T}} \bar{\lambda} = 0 \tag{5-7a}$$

$$G(\bar{\mathbf{z}}) + \bar{s} = 0 \tag{5-7b}$$

$$\bar{S}\bar{\lambda} = \bar{\tau}\mathbf{1}_{m+\tilde{m}} \tag{5-7c}$$

$$(\bar{s}, \bar{\lambda}) > 0 \tag{5-7d}$$

where $\bar{s} = [s_r^{\mathrm{T}}\ s_{\epsilon_s}^{\mathrm{T}}]$, $\bar{S} = \mathbf{diag}(\bar{s})$ and s_r denotes the **re-arranged** slackness variable corresponding to the **re-arranged** constraint sets in (5-3) for Problem (2-11).

In practic, (5-7b) can be written into the following form:

$$\begin{cases} g(\mathcal{G}_1) - \epsilon_1 + s_1 = 0 \\ g(\mathcal{G}_2) - \epsilon_2 + s_2 = 0 \\ g(\mathcal{G}_3) + s_3 = 0 \\ -\epsilon_s + s_{\epsilon_s} = 0 \end{cases} \tag{5-8}$$

where $s_r = [s_1^{\mathrm{T}}\ s_2^{\mathrm{T}}\ s_3^{\mathrm{T}}]^{\mathrm{T}}$.

Therefore, to initialize PDIP with a strictly feasibly starting point to solve Problem (5-6), variables in KKT conditions in (5-7) can be initialized in the following way according to (5-3):

$$\mathbf{z}_{0,\mathrm{PDIP}} = \mathbf{z}_{\mathrm{DFG}}, \qquad \epsilon_{s,0,\mathrm{PDIP}} = \begin{bmatrix} \delta_1 + \tilde{\delta}\mathbf{1}_{m_1} \\ \tilde{\delta}\mathbf{1}_{m_2} \end{bmatrix} \tag{5-9a}$$

$$\bar{s}_{0,\mathrm{PDIP}} = \begin{bmatrix} \tilde{\delta}\mathbf{1}_{m_1} \\ \tilde{\delta}\mathbf{1}_{m_2} \\ -\delta_3\mathbf{1}_{m_3} \\ \delta_1 + \tilde{\delta}\mathbf{1}_{m_1} \\ \tilde{\delta}\mathbf{1}_{m_2} \end{bmatrix}, \qquad \bar{\lambda}_{0,\mathrm{PDIP}} = \begin{bmatrix} \lambda(\mathcal{G}_1) \\ \lambda(\mathcal{G}_2) \\ \lambda(\mathcal{G}_3) \\ \lambda(\mathcal{G}_1) \\ \lambda(\mathcal{G}_2) \end{bmatrix} \tag{5-9b}$$

with $\tilde{\delta} > 0$ being efficiently small. Note that the information on the duality gap is fully fed into the PDIP phase by (5-9b) according to the following:

$$\begin{aligned} (\bar{s}_{0,\mathrm{PDIP}})^{\mathrm{T}}\bar{\lambda}_{0,\mathrm{PDIP}} &= (\tilde{\delta}\mathbf{1}_{m_1})^{\mathrm{T}}\lambda(\mathcal{G}_1) + (\tilde{\delta}\mathbf{1}_{m_2})^{\mathrm{T}}\lambda(\mathcal{G}_2) - (\delta_3\mathbf{1}_{m_3})^{\mathrm{T}}\lambda(\mathcal{G}_3) \\ &\quad + [(\delta_1 + \tilde{\delta})\mathbf{1}_{m_1}]^{\mathrm{T}}\lambda(\mathcal{G}_1) + (\tilde{\delta}\mathbf{1}_{m_2})^{\mathrm{T}}\lambda(\mathcal{G}_2) \\ &\approx (\delta_1\mathbf{1}_{m_1})^{\mathrm{T}}\lambda(\mathcal{G}_1) \\ &= s_r^{\mathrm{T}}\lambda_r \end{aligned} \tag{5-10}$$

where we use the fact that $\tilde{\delta}$ is efficiently small and $\lambda(\mathcal{G}_3) = 0$.

5-1-4 Algorithm Description

The nonlinear system we have to solve to obtain the primal-dual search direction consists of (5-7a) - (5-7c). Since the algorithm is initialized with a strictly feasible starting point, linearizing the nonlinear system, we can eliminate the equation

$$G(\bar{\mathbf{z}}) + \bar{s} = 0 \tag{5-11}$$

Thus the final nonlinear system is:

$$\begin{bmatrix} \nabla_{\bar{\mathbf{z}}} F_s(\bar{\mathbf{z}}) + \nabla_{\bar{\mathbf{z}}} G(\bar{\mathbf{z}})^{\mathrm{T}} \bar{\lambda} \\ -\mathbf{diag}\left(G(\bar{\mathbf{z}})\right) \bar{\lambda} - \bar{\tau}\mathbf{1} \end{bmatrix} = \begin{bmatrix} r_{\text{dual,s}} \\ r_{\text{cent,f,s}} \end{bmatrix} \tag{5-12}$$

Linearize the nonlinear system above:

$$\begin{bmatrix} \nabla^2_{\mathbf{zz}} F_s(\mathbf{z},\epsilon_s) + \bar{\Lambda}\nabla^2_{\mathbf{zz}} G(\mathbf{z},\epsilon_s) & \nabla^2_{\mathbf{z}\epsilon_s} F_s(\mathbf{z},\epsilon_s) + \bar{\Lambda}\nabla^2_{\mathbf{z}\epsilon_s} G(\mathbf{z},\epsilon_s) & \nabla_{\mathbf{z}} G(\mathbf{z},\epsilon_s) \\ \nabla^2_{\mathbf{z}\epsilon_s} F_s(\mathbf{z},\epsilon_s) + \bar{\Lambda}\nabla^2_{\mathbf{z}\epsilon_s} G(\mathbf{z},\epsilon_s) & \nabla^2_{\epsilon_s\epsilon_s} F_s(\mathbf{z},\epsilon_s) + \bar{\Lambda}\nabla^2_{\epsilon_s\epsilon_s} G(\mathbf{z},\epsilon_s) & \nabla_{\epsilon_s} G(\mathbf{z},\epsilon_s) \\ -\bar{\Lambda}\nabla_{\mathbf{z}} G(\mathbf{z},\epsilon_s) & -\bar{\Lambda}\nabla_{\epsilon_s} G(\mathbf{z},\epsilon_s) & -\mathbf{diag}\left(G(\mathbf{z},\epsilon_s)\right) \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z}_{\text{pd}} \\ \Delta\epsilon_{s,\text{pd}} \\ \Delta\bar{\lambda}_{\text{pd}} \end{bmatrix} = - \begin{bmatrix} r_{\text{dual,s},\mathbf{z}} \\ r_{\text{dual,s},\epsilon_s} \\ r_{\text{cent,f,s}} \end{bmatrix} \tag{5-13}$$

with $\mathbf{diag}(\bar{\lambda}) = \bar{\Lambda}$. Thus,

$$\begin{bmatrix} \nabla^2 f_0(\mathbf{z}) + \omega_2 I_n & 0 & \bar{G}^{\mathrm{T}} \\ 0 & 0 & -\bar{F}^{\mathrm{T}} \\ -\bar{\Lambda}\bar{G} & -\bar{\Lambda}(-\bar{F}) & -\mathbf{diag}\left(g(\mathbf{z},\epsilon_s)\right) \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z}_{\text{pd}} \\ \Delta\epsilon_{s,\text{pd}} \\ \Delta\bar{\lambda}_{\text{pd}} \end{bmatrix} = - \begin{bmatrix} \nabla f_0(\mathbf{z}) + \bar{G}^{\mathrm{T}}\bar{\lambda} \\ \omega_1 \mathbf{1}_{\tilde{m}} - \bar{F}^{\mathrm{T}}\bar{\lambda} \\ -\mathbf{diag}\left(g(\mathbf{z},\epsilon_s)\right)\bar{\lambda} - \bar{\tau}\mathbf{1}_{m+\tilde{m}} \end{bmatrix} \tag{5-14}$$

Besides (5-14), one can also take the slackness variable $\bar{s}$ in to consideration when computing the primal-dual search direction. Therefore, the following linear system that is similar to (4-17) is solved.

$$\begin{bmatrix} \nabla^2_{\bar{\mathbf{z}}} F_s(\bar{\mathbf{z}}) & DG(\bar{\mathbf{z}})^{\mathrm{T}} & 0 \\ DG(\bar{\mathbf{z}}) & 0 & I_{m+\tilde{m}} \\ 0 & \bar{S} & \bar{\Lambda} \end{bmatrix} \begin{bmatrix} \Delta\bar{\mathbf{z}}_{\text{pd}} \\ \Delta\bar{\lambda}_{\text{pd}} \\ \Delta\bar{s}_{\text{pd}} \end{bmatrix} = - \begin{bmatrix} \nabla_{\bar{\mathbf{z}}} F_s(\bar{\mathbf{z}}) + DG(\bar{\mathbf{z}})^{\mathrm{T}}\bar{\lambda} \\ G(\bar{\mathbf{z}}) + \bar{s} \\ \bar{S}\bar{\lambda} - \bar{\tau}\mathbf{1}_{m+\tilde{m}} \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} H_z & G_z^{\mathrm{T}} & 0 \\ G_z & 0 & I_{m+\tilde{m}} \\ 0 & \bar{S} & \bar{\Lambda} \end{bmatrix}}_{Dr_{\bar{\tau}}(\bar{\zeta})} \underbrace{\begin{bmatrix} \Delta\bar{\mathbf{z}}_{\text{pd}} \\ \Delta\bar{\lambda}_{\text{pd}} \\ \Delta\bar{s}_{\text{pd}} \end{bmatrix}}_{\Delta\bar{\zeta}_{\text{pd}}} = - \begin{bmatrix} H_z\bar{\mathbf{z}} + h_z + G_z^{\mathrm{T}}\bar{\lambda} \\ G_z\bar{\mathbf{z}} + E_z x_0 + g_z + \bar{s} \\ \bar{S}\bar{\lambda} - \bar{\tau}\mathbf{1}_{m+\tilde{m}} \end{bmatrix} \tag{5-15}$$

where $DG(\bar{\mathbf{z}})$ denotes the first derivative of $G(\bar{\mathbf{z}})$ in (5-6b)

$$DG(\bar{\mathbf{z}}) = \begin{bmatrix} \nabla G_1(\bar{\mathbf{z}})^{\mathrm{T}} \\ \vdots \\ \nabla G_{m+\tilde{m}}(\bar{\mathbf{z}})^{\mathrm{T}} \end{bmatrix}$$

Hence, Problem (5-6) can be solved by primal-dual interior-point methods in Subsection 4-2-1. Further, the convergence of primal-dual interior-point solver can be analyzed via standard Newton's method solving the following corresponding optimization problem involving barrier function:

$$\min_{\bar{\mathbf{z}}\in\mathbb{R}^{n+\tilde{m}}} \quad F_{b,s}(\bar{\mathbf{z}}) = F_s(\bar{\mathbf{z}}) + \phi(\bar{\mathbf{z}}) \tag{5-16}$$

where $\phi(\bar{\mathbf{z}}) = -\sum_{i=1}^{m+\tilde{m}} \log\left(-G_i(\bar{\mathbf{z}})\right)$. Therefore, the primal-dual interior-point method can enter directly into pure Newton phase when initialized by dual fast gradient method and the condition

$$\left\|\nabla F_{b,s}(\bar{\mathbf{z}}_k)\right\|_2 := \left\|\nabla F_b(\mathcal{P}_{\text{soft}})\right\|_2 \leq \eta_p(\mathcal{P}_{\text{soft}}) \tag{5-17}$$

is satisfied, which is derived based on Theorem 4.2 and (5-16). Further, the following combined algorithm can be derived.

The detailed algorithm, in which primal-dual interior-point method with soft constraint approach starts from a medium-accuracy solution returned by dual fast gradient method, is described in Algorithm 5.

Algorithm 5 Combined Algorithm: Dual Fast Gradient Method Initializing Primal-Dual Interior-Point Method with Soft Constraint Approach.

Given H, h, g, E, G, x_0, $\hat{\lambda}$, L_d, η_p, $\varepsilon < \eta_p$.
Initialize $\lambda_0 = \hat{\lambda}$, $k = 0$.
repeat
  Compute $\mathbf{z}_k = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z}, \lambda_k)$.
  Compute $\hat{\lambda}_k = \left[\lambda_k + \frac{1}{L_d}\nabla d(\lambda_k)\right]_+$.
  Compute $\lambda_{k+1} = \frac{k+1}{k+3}\hat{\lambda}_k + \frac{2}{L_d(k+3)}\left[\sum_{j=0}^{k} \frac{j+1}{2}\nabla d(\lambda_j)\right]_+$.
  $k = k + 1$.
until (5-17) is satisfied.
return $(\mathbf{z}_{\text{DFG}}, \lambda_{\text{DFG}})$
Reformulate the QP with soft constraint approach in (5-6) according to (5-3), and initialize $\bar{\zeta}_0$ according to (5-9) and $k = 0$.
repeat
  Determine $\bar{\tau}_{k+1} = \bar{\mu}_{k+1} = \kappa\bar{\mu}_k = (\bar{s}_k)^{\mathrm{T}}\bar{\lambda}_k/m$.
  Compute search direction $\Delta\bar{\zeta}_{k,\text{pd}} = (\Delta\bar{\mathbf{z}}_{k,\text{pd}}, \Delta\bar{\lambda}_{k,\text{pd}}, \Delta\bar{s}_{k,\text{pd}})$ by solving (5-15).
  Update $\bar{\zeta}_{k+1} = \bar{\zeta}_k + \Delta\bar{\zeta}_{k,\text{pd}}$.
  $k = k + 1$.
until stopping criterion $\bar{\mu}_k \leq \varepsilon$.
return Point close to $\mathbf{z}^*$

5-2 Constraint Tightening Approach in Dual Fast Gradient Methods

One main drawback of dual methods is that they can only ensure primal feasibility asymptotically. This short-coming can be solved by using a constraint tightening approach. Dual

first-order methods with tightening constraint approach has been proposed in [14, 34]. An extended approach with adaptively chosen tightening parameters is presented in [25].

5-2-1 Overview

When solving a QP problem, such as, Problem (2-11), with constraint tightening approach, instead of solving the problem directly, the following one is solved:

$$\begin{aligned} \min_{\mathbf{z}} \quad & f_0(\mathbf{z}) \\ s.t. \quad & g(\mathbf{z}) + \epsilon_c \mathbf{1} \leq 0 \end{aligned} \tag{5-18}$$

where $\epsilon_c > 0$ is the *tightening parameter* and decides the amount of relative constraint tightening. We assume here that the Slater's condition also holds for the tightened problem (5-18).

5-2-2 Model Predictive Control Problem Reformulation

Considering Problem (2-11) arises from MPC scheme, the following problem with tightened constraints is solved:

$$\begin{aligned} \min_{\mathbf{z}\in\mathbb{R}^n} \quad & f_0(\mathbf{z}) = \frac{1}{2}\mathbf{z}^{\mathrm{T}} H \mathbf{z} + (hx_0)^{\mathrm{T}} \mathbf{z} \\ s.t. \quad & g_c(\mathbf{z}) = G\mathbf{z} + Ex_0 + g + \epsilon_c \mathbf{1}_m \leq 0 \end{aligned} \tag{$\mathcal{P}_{\text{tight}}$}$$

If ϵ_c is selected as

$$\frac{1}{2}\epsilon_c = \max\left[G\mathbf{z} + Ex_0 + g\right]_+ \tag{5-19}$$

a strictly feasible solution can be obtained [25].

According to Problem ($\mathcal{P}_{\text{tight}}$), we define the following Lagrangian,

$$\mathcal{L}(\mathbf{z}, \lambda_c) = f_0(\mathbf{z}) + \lambda_c^{\mathrm{T}} g_c(\mathbf{z}) \tag{5-20}$$

the dual function

$$d(\lambda_c) = \min_{\mathbf{z}\in\mathbb{R}^n} \mathcal{L}(\mathbf{z}, \lambda_c) \tag{5-21}$$

and the inner problem

$$\mathbf{z}^*(\lambda_c) = \arg\min_{\mathbf{z}\in\mathbb{R}^n} \mathcal{L}(\mathbf{z}, \lambda_c) \tag{5-22}$$

Therefore, Problem ($\mathcal{P}_{\text{tight}}$) can be solved using a dual method, e.g., dual fast gradient method.

5-2-3 Initialization Strategy for Primal-Dual Interior-Point Methods

By constraint tightening approach in dual fast gradient phase, a strictly feasible solution can be obtained. Thus, the solution returned by dual fast gradient can be directly fed into primal-dual interior-point method. Therefore, the initialization strategy of primal-dual interior-point method can be given as follows:

$$\begin{aligned} \mathbf{z}_{0,\text{PDIP}} &= \mathbf{z}_{\text{DFG}}, \quad \lambda_{0,\text{PDIP}} = \lambda_{\text{DFG}} \\ s_{0,\text{PDIP}} &= -(G\mathbf{z}_{\text{DFG}} + Ex_0 + g) \end{aligned} \tag{5-23}$$

5-2-4 Algorithm Description

In primal space, according to (5-1), with a strictly feasible solution returned by dual fast gradient method, the condition in primal space for primal-dual interior-point method to enter pure Newton phase can be derived based the the barrier function associate with Problem (2-11), that is:

$$\|\nabla F_{b,t}(\mathbf{z}_k)\|_2 := \|\nabla F_b(\mathcal{P}_{\text{tight}})\|_2 \leq \eta_p(\mathcal{P}_{\text{tight}}) \tag{5-24}$$

where $F_{b,t}(\mathbf{z}) = f_0(\mathbf{z}) + \phi(\mathbf{z})$, and $\phi(\mathbf{z}) = -\sum_{i=1}^{m} \log\left(g_{c,i}(\mathbf{z})\right)$ is the barrier function.

The detailed algorithm, in which primal-dual interior-point method starting from a medium-accuracy solution returned by dual fast gradient method with constraint tightening approach, is described in Algorithm 6.

Algorithm 6 Combined Algorithm: Dual Fast Gradient Method with Constraint Tightening Approach Initializing Primal-Dual Interior-Point Method.

Given H, h, g, E, G, x_0, $\hat{\lambda}_c$, L_d, η_p, $\varepsilon < \eta_p$.
Initialize $\lambda_{c,0} = \hat{\lambda}_c$, $k = 0$.
repeat
  Compute $\mathbf{z}_k = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z}, \lambda_{c,k})$.
  Compute $\hat{\lambda}_{c,k} = \left[\lambda_{c,k} + \frac{1}{L_d}\nabla d(\lambda_{c,k})\right]_+$.
  Compute $\lambda_{c,k+1} = \frac{k+1}{k+3}\hat{\lambda}_{c,k} + \frac{2}{L_d(k+3)}\left[\sum_{j=0}^{k} \frac{j+1}{2}\nabla d(\lambda_{c,j})\right]_+$.
  $k = k + 1$.
until (5-24) is satisfied.
return $(\mathbf{z}_{\text{DFG}}, \lambda_{\text{DFG}})$
Initialize ζ_0 according to (5-23) and $k = 0$.
repeat
  Determine $\tau_{k+1} = \mu_{k+1} = \kappa\mu_k = (s_k)^{\mathrm{T}}\lambda_k/m$.
  Compute search direction $\Delta\zeta_{k,\text{pd}}$ by solving (4-17).
  Update $\zeta_{k+1} = \zeta_k + \Delta\zeta_{k,\text{pd}}$.
  $k = k + 1$.
until stopping criterion $\mu_k \leq \varepsilon$.
return Point close to $\mathbf{z}^*$

5-3 Primal Switching Condition from Dual Fast Gradient to Primal-Dual Interior-Point Methods

One drawback of applying the switching condition (5-1) defined using barrier function is that the condition involves the inverse of $g(\mathbf{z})$ which will become undefined when there are constraints active at the optimum. However, the main reason why MPC are preferred for solving control problem in dynamic system with constraints is that MPC can exploit the full range of the actuators, states and outputs.

Figure 5-1 shows how the primal condition in (5-1) drifts when the feasible starting primal-dual interior-point solver converges to the optimum. The optimization problem $\mathcal{P}(\mathbf{z})$ in this

case is the mp-QP in Problem (2-11) with current state

$$x_0 = \begin{bmatrix} -0.3010 & -1.5380 \end{bmatrix}^{\mathrm{T}}$$

This current state x_0 leads to constraints on output $\|y\|_\infty \leq 1$ active at the optimum.

In Figure 5-1, the left axis shows the value of $\|\nabla F_b(\mathcal{P}(\mathbf{z}))\|_2$ that plays an important role when determining the switching between damped and pure Newton phase in primal-dual interior-point methods. The right axis shows the difference between the current and optimal cost. The optimal cost $f_0(\mathbf{z}^*)$ is computed by MATLAB function `quadprog`. Both axises are in logarithmic scale. From Figure 5-1 we can see that the value of $\|\nabla F_b(\mathcal{P}(\mathbf{z}))\|_2$ decreases and after 12 iterations, the switching condition of primal-dual interior-point method entering the pure Newton phase is satisfied with $\|\nabla F_b(\mathcal{P}(\mathbf{z}))\|_2 \leq \eta_p = 0.0093$. However, after 30 iterations, the value of $\|\nabla F_b(\mathcal{P}(\mathbf{z}))\|_2$ begins to increase while the difference between the current and optimal cost is still decreasing, which means that the solution converges to an active set, on which $g_i(\mathbf{z}) = 0$ for some i. With 30 iterations, the algorithm reaches the accuracy around 10^{-8}. Afterwards, the algorithm continues converging to higher accuracy while the value of $\|\nabla F_b(\mathcal{P}(\mathbf{z}))\|_2$ flows back the greater values. Around 50 iterations the value of $\|\nabla F_b(\mathcal{P}(\mathbf{z}))\|_2$ exceeds η_p, however, the algorithm has already converged to the optimum.

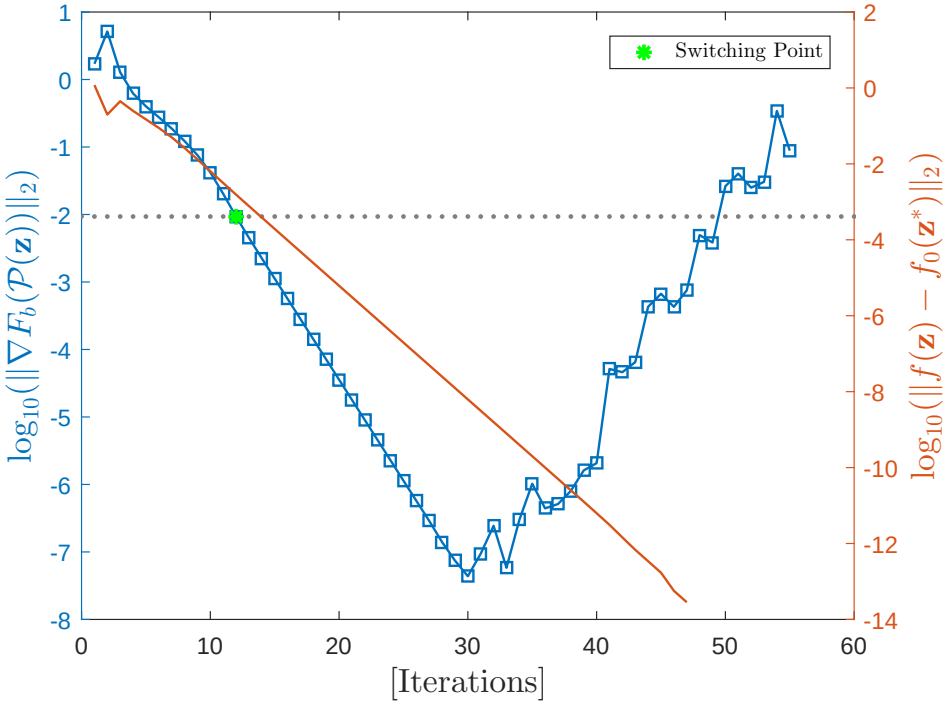


Figure 5-1: Primal switching condition in (5-1) for primal-dual interior-point methods.

Another convergence analysis based on the convergence of the residual variable r_τ defined in (4-16) has been given in [5, Chapter 10]. By this scenario, the 2-norm of the residual variable has to satisfy the following condition in order to enter pure Newton phase:

$$\|r_\tau(\mathbf{z}, \lambda, s)\| \leq \eta_{Dr} \tag{5-25}$$

where $0 < \eta_{Dr} \leq \frac{m_{Dr}^2}{L_{Dr}}$ with m_{Dr} the lower bound on $Dr_\tau(\zeta)$ and L_{Dr} denotes the Lipschitz constant of $Dr_\tau(\zeta)$. Compared to convergence analysis presented in [5, Chapter 10] we want to account for inequality constraints. In the presence of inequality, the condition in (5-25) becomes too conservative to be used in practice because the condition numbers of $Dr_\tau(\zeta)$ are hard to derive, due to asymmetry and variance of $Dr_\tau(\zeta)$. Therefore, we can conclude by now that by the author's knowledge, the switching condition between damped and pure Newton phase in primal space of primal-dual interior-point method, e.g., Theorem 4.2 ((5-17) and (5-24) are its extensions) and (5-25), will not work out in practice, even theoretically, when inequality constraints are involved in system dynamics.

Loosely speaking, the conditions above state that if the algorithm is far from the optimal solution, it converges more slowly (damped Newton phase), while when it is close to the optimal solution, it converges faster (pure Newton phase). We rely on this observation to propose, in the next section, a new switching condition (in dual space) to enter pure Newton phase. Moreover, we design an improved primal-dual interior-point algorithm that fully replace the damped Newton phase with the dual fast gradient method.

5-4 Proposed Dual Switching Condition from Dual Fast Gradient to Primal-Dual Interior-Point Methods

Recall the QP problem obtained via condensed formulation for a MPC problem in (2-11):

$$\begin{aligned} \min_{\mathbf{z}\in\mathbb{R}^n} \quad & f_0(\mathbf{z}) = \frac{1}{2}\mathbf{z}^{\mathrm{T}} H \mathbf{z} + (hx_0)^{\mathrm{T}}\mathbf{z} \\ s.t. \quad & g(\mathbf{z}) = G\mathbf{z} + Ex_0 + g \leq 0 \end{aligned}$$

in which we substituting all constraints on $\mathbf{z}$ into $g(\mathbf{z}) \leq 0$. In this way, $\mathcal{Z} \coloneqq \mathbb{R}^n$.

5-4-1 Characterization of the Dual Hessian

Recall the dual function $d(\lambda)$ defined in (3-5). Then we can derive the following expression for dual Hessian:

$$\nabla^2 d(\lambda) = -\nabla g(\mathbf{z}(\lambda)) \left[\nabla^2 f_0(\mathbf{z})\right]^{-1} \nabla g(\mathbf{z}(\lambda))^{\mathrm{T}} = -GH^{-1}G^{\mathrm{T}} \tag{5-26}$$

for QP problems derived from a MPC problem. A proof based on the proof of Theorem 2.1 in [26] can be derived.

Proof. Since $f_0(\mathbf{z})$ is strongly convex, it follows that for any $\lambda \geq 0$, $\mathbf{z}(\lambda)$ is unique, and thus $d(\lambda)$ is a differentiable function having the gradient given by:

$$\nabla d(\lambda) = g(\mathbf{z}(\lambda)) \tag{5-27}$$

Give the optimality condition,

$$\nabla f_0(\mathbf{z}(\lambda)) + \nabla g(\mathbf{z}(\lambda))^{\mathrm{T}}\lambda = 0, \quad \lambda \geq 0 \tag{5-28}$$

and differentiate it w.r.t λ we can obtain the following equation:

$$\nabla \mathbf{z}(\lambda)^{\mathrm{T}} \nabla^2 f_0(\mathbf{z}(\lambda)) + \nabla g(\mathbf{z}(\lambda)) + \nabla \mathbf{z}(\lambda)^{\mathrm{T}} \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(\mathbf{z}(\lambda)) = 0 \tag{5-29}$$

from which we can obtain:

$$\mathbf{z}(\lambda)^{\mathrm{T}} = -\nabla g(\mathbf{z}(\lambda)) \left[\nabla^2 f_0(\mathbf{z}(\lambda)) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(\mathbf{z}(\lambda)) \right]^{-1} \tag{5-30}$$

We also assume that $g(\mathbf{z})$ are twice differentiable, and by differentiating (5-27) we have

$$\nabla^2 d(\lambda) = \nabla g(\mathbf{z}(\lambda)) \nabla \mathbf{z}(\lambda) \tag{5-31}$$

Substituting (5-30) into (5-31) and obtain:

$$\begin{aligned} \nabla^2 d(\lambda) &= -\nabla g(\mathbf{z}(\lambda)) \left[\nabla^2 f_0(\mathbf{z}(\lambda)) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(\mathbf{z}(\lambda)) \right]^{-1} \nabla g(\mathbf{z}(\lambda))^{\mathrm{T}} \\ &= -\nabla g(\mathbf{z}(\lambda)) \left[\nabla^2 f_0(\mathbf{z}) \right]^{-1} \nabla g(\mathbf{z}(\lambda))^{\mathrm{T}} \\ &= -G H^{-1} G^{\mathrm{T}} \end{aligned} \tag{5-32}$$

where in the second equation we use that $\sum_{i=1}^{m} \lambda_i \nabla^2 g_i(\mathbf{z}(\lambda)) = 0$ since $g(\mathbf{z})$ is affine. □

Therefore, we can derive the following upper and lower bound on the norm of $\nabla^2 d(\lambda)$

$$m_d = \frac{\|G\|_2^2}{\sigma_{\max}(H)} \leq \|\nabla^2 d(\lambda)\|_2 \leq \frac{\|G\|_2^2}{\sigma_{\min}(H)} = M_d \tag{5-33}$$

where $\sigma_{\max}(H)$ and $\sigma_{\min}(H)$ denote the maximum and minimum eigenvalue of H. And from (5-32), it is obvious that $\nabla^2 d(\lambda)$ is Lipschitz continuous with any $L_{dH} \geq 0$.

5-4-2 Modification of the Switching Condition

Mainly, we care about the pure Newton phase in the convergence of primal-dual interior-point method. Since it is hard to analyze the characteristic of convergence in primal space and dual fast gradient method operates in dual space, we transfer our convergence analysis into the dual space as well. We also make the following assumption on the solution obtained by dual fast gradient method.

Assumption 5.1. After $k_{\max}$ iterations, dual fast gradient method returns a η_d-suboptimal solution $\mathbf{z}_{k_{\max}}$ which provide the following estimation on feasibility violation:

$$\left\| [g(\hat{\mathbf{z}}_{k_{\max}})]_+ \right\|_2 \leq \eta_d$$

where $\hat{\mathbf{z}}_k$ denotes the average sequence for the primal variable

$$\hat{\mathbf{z}}_k = \sum_{j=0}^{k} \frac{2(j+1)}{(k+1)(k+2)} \mathbf{z}_j$$

Based on the convergence analysis in [5, §9.5.3] of standard Newton's method, we can derive a similar outline of convergence analysis in dual space.

Theorem 5.2. *There exist numbers η_d and γ_d with $0 < \eta_d \leq \frac{m_d^2}{L_{dH}}$ and $\gamma_d > 0$ such that the following hold.*

- *Damped Newton Phase. If $\|\nabla d(\lambda)\|_2 > \eta_d$, then for all $k \geq 0$*

$$d(\lambda_{k+1}) - d(\lambda_k) \leq \gamma_d \tag{5-34}$$

- *Pure Newton Phase. If $\|\nabla d(\lambda)\|_2 \leq \eta_d$, the the backtracking line search in dual space will select $\rho_{\bar{k}} = 1$ and for $\bar{k} > 0$*

$$\frac{L_{dH}}{2m_d^2} \left\|\nabla d(\lambda_{\bar{k}+1})\right\|_2 \leq \left(\frac{L_{dH}}{2m_d^2} \|\nabla d(\lambda_{\bar{k}})\|_2\right)^2 \tag{5-35}$$

and an upper bound on the duality gap can be derived as

$$f_0(\boldsymbol{z}^*) - d(\lambda^{\bar{k}}) \leq \frac{1}{2m_d} \left\|\nabla d(\lambda^{\bar{k}})\right\|_2^2 \leq \frac{2m_d^3}{L_{dH}^2} \left(\frac{1}{2}\right)^{2^{\bar{k}+1}} \tag{5-36}$$

which shows that we achieve quadratic convergence.

Proof. Here we provide an outline of the proof. For those readers who are interested, the detailed proof can be found in Appendix A-1.

We must take into account that the dual function (3-5) is a concave function that must be maximized. Hence, the search direction in dual space is an ascent direction, compared to the one in primal space which is a descent one. Similar considerations hold to define the Newton increment for dual problem. Following the convergence proof in [5, Chapter 9], first we show that the following condition leads to a unit step size in backtracking line search in dual space:

$$\|\nabla d(\lambda_k)\|_2 \leq \eta_d. \tag{5-37}$$

Second, by relying on the Lipschitz continuity of the dual Hessian in (5-26) and strong convexity property of function $-d(\lambda)$ (which follow from the assumptions of the theorem), a crucial result on the duality gap can be obtained:

$$f_0(\mathbf{z}^*) - f(\mathbf{z}_{\bar{k}}) \leq f_0(\mathbf{z}^*) - d(\lambda_{\bar{k}}) \leq \frac{1}{2m_d} \|\nabla d(\lambda_{\bar{k}})\|_2^2 \tag{5-38}$$

Finally, using the definition of η_d and substituting (5-37) into (5-38) concludes the proof. □

Therefore, under Assumption 5.1, (3-5) is can be modified as follows:

$$\begin{aligned} d(\lambda) &= \min_{\mathbf{z}\in\mathbb{R}^n} f_0(\mathbf{z}) + \lambda^{\mathrm{T}} g(\mathbf{z}) \\ &= \min_{\mathbf{z}\in\mathbb{R}^n} f_0(\mathbf{z}) + \lambda^{\mathrm{T}} [g(\mathbf{z})]_+ \end{aligned} \tag{5-39}$$

since in the Lagrange multiplier λ, the entries corresponding to inactive constraints equal to zero. Therefore, the first-order derivative of the dual function $d(\lambda)$ for Problem (5-39) is:

$$\nabla d(\lambda) = [g(\mathbf{z})]_+ \tag{5-40}$$

Thus, the switching condition in Theorem 5.2 can be transfered into:

$$\|\nabla d(\lambda)\|_2 = \left\|[g(\mathbf{z})]_+\right\|_2 \leq \eta_d \tag{5-41}$$

5-5 Proposed Solver Description

Algorithm 7 Proposed Solver.

1: Given H, h, g, E, G, x_0, $\hat{\lambda}$, L_d, η_d, $\varepsilon < \eta_d$.
2: Initialize $\lambda_0 = \hat{\lambda}$, $k = 0$.
3: **repeat**
4: Compute $\mathbf{z}_k = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{z}, \lambda_k)$.
5: Compute $\hat{\lambda}_k = \left[\lambda_k + \frac{1}{L_d}\nabla d(\lambda_k)\right]_+$.
6: Compute $\lambda_{k+1} = \frac{k+1}{k+3}\hat{\lambda}_k + \frac{2}{L_d(k+3)}\left[\sum_{j=0}^{k} \frac{j+1}{2}\nabla d(\lambda_j)\right]_+$.
7: $k = k + 1$.
8: **until** (5-41) is satisfied.
9: **return** $(\mathbf{z}_{\text{DFG}}, \lambda_{\text{DFG}})$
10: Initialize ζ_0 according to (5-42) and $k = 0$.
11: **repeat**
12: Determine $\tau_{k+1} = \mu_{k+1} = \kappa\mu_k = (s_k)^{\text{T}}\lambda_k/m$.
13: Compute search direction $\Delta\zeta_{k,\text{pd}}$ by solving (4-17).
14: Update $\zeta_{k+1} = \zeta_k + \Delta\zeta_{k,\text{pd}}$.
15: $k = k + 1$.
16: **until** stopping criterion $\mu_k \leq \varepsilon$.
17: **return** Point close to $\mathbf{z}^*$

Algorithm 7 summarizes the proposed strategy to compute a solution for the MPC problem in (2-11), in which the DFG (Steps 3-9) is used to compute an η_d-solution that allows the PDIP (Steps 10-17) to enter directly in the pure Newton phase and converge quadratically to the optimal solution of the MPC problem. When Algorithm 7 switches from the DFG phase to the pure Newton phase (Step 5), we have to make sure that we preserve the information already computed by the DFG. Hence, the initialization strategy for the PDIP is important to have a successful switch. In this respect, Step 5 of Algorihtm 7 relies on the solution $(\mathbf{z}_{\text{DFG}}, \lambda_{\text{DFG}})$ returned by DFG as follows:

$$\zeta_0 = (\mathbf{z}_{\text{DFG}},\, \lambda_{\text{DFG}}, s_{0,\text{PDIP}}), \tag{5-42a}$$

$$s_{0,\text{PDIP}} = -\left[g(\mathbf{z}_{\text{DFG}})\right]_- + \left[g(\mathbf{z}_{\text{DFG}})\right]_+. \tag{5-42b}$$

Equation (5-42b) provides an initialization for s that guarantees $s > 0$. Note that the information on the duality gap is fully fed into the pure Newton phase by (5-42b), according to

the following

$$\begin{aligned}(s_{0,\mathrm{PDIP}})^{\mathrm{T}}\lambda_{0,\mathrm{PDIP}} &= \Big[-\left[g(\mathbf{z}_{\mathrm{DFG}})\right]_{-}+\left[g(\mathbf{z}_{\mathrm{DFG}})\right]_{+}\Big]^{\mathrm{T}}\lambda_{0,\mathrm{PDIP}}\\ &= \left[g(\mathbf{z}_{\mathrm{DFG}})\right]_{+}^{\mathrm{T}}\lambda_{0,\mathrm{PDIP}},\end{aligned}$$

where we use the fact that $\left[g(\mathbf{z}_{\mathrm{DFG}})\right]_{-}^{\mathrm{T}}\lambda_{0,\mathrm{PDIP}} = 0$.

Chapter 6

Numerical Results

The proposed solver in Algorithm 5, 6 and 7 are tested on the planar discrete-time linear unstable system in [34] with the following system matrices:

$$A = \begin{bmatrix} 1.09 & 0.22 \\ 0.49 & 0.02 \end{bmatrix}, \quad B = \begin{bmatrix} 1.22 & 0.88 \\ -0.78 & -0.34 \end{bmatrix}$$
$$C = \begin{bmatrix} 1.34 & -0.16 \\ -3.19 & -0.56 \end{bmatrix}, \quad D = \begin{bmatrix} 1.60 & 1.01 \\ -0.68 & 0.77 \end{bmatrix}$$

The input and output of the system are subjected to the following constraints: $\|u\|_\infty \leq 1$, $\|y\|_\infty \leq 1$. Furthermore, Q and R are defined as follows according to [34]:

$$Q = \begin{bmatrix} 5.44 & 5.80 \\ 5.80 & 7.01 \end{bmatrix}, \quad R = \begin{bmatrix} 1.14 & 0.68 \\ 0.68 & 0.62 \end{bmatrix}$$

We initialize the system at the following initial condition:

$$x_0 = \begin{bmatrix} -0.3010 & -1.5480 \end{bmatrix}^T$$

that causes active output constraints at the optimum.

We tested the algorithm on a Windows OS, using an Intel(R) Core(TM) i7-4550 CPU (1.50-2.10 GHz) and RAM 8.00GB. The algorithms are implemented in MATLAB.

6-1 Results for Primal Switching Condition

As shown in Section 5-3, the switching condition in (5-1) between damped and pure Newton phase for primal-dual interior-point methods in primal space encounters invalidation when there are constraints active at the optimum. Therefore, the condition (5-1) cannot be used in

practice due to its conservativeness. However, the idea that removing damped Newton phase in primal-dual interior-point methods by dual fast gradient methods can still be verified.

In order to present numerical results for Algorithm 5 and 6, we use a stopping criterion on duality gap ($\mu \leq \epsilon_{\text{DFG}}$) for dual fast gradient method instead of condition (5-1) ((5-17) for Algorithm 5 and (5-24) for Algorithm 6). And we use condition (5-25) to evaluate the iterations in damped and pure Newton phase.

6-1-1 Soft Constraint Approach in Primal-Dual Interior-Point Methods

For Algorithm 5, we compare the following five scenarios:

Scenario 1. Algorithm 4 is *warm-started* with $\bar{\mathbf{z}}_{0,\text{PDIP}} := [\mathbf{z}_{\text{LS}}^{\text{T}} \ \epsilon_s(\mathbf{z}_{\text{LS}})^{\text{T}}]^{\text{T}}$, where $\mathbf{z}_{\text{LS}}$ is the optimal solution of the unconstrained problem, that is, a least-squares (LS) problem, associated with Problem (2-11), and $\epsilon_c(\mathbf{z}_{\text{LS}})$ is the softening parameter associated with solution $\mathbf{z}_{\text{LS}}$. Slack variable $\bar{s}_{0,\text{PDIP}}$ related to Problem (5-6) is initialized according to (5-9) related to $\bar{\mathbf{z}}_{0,\text{PDIP}}$ and $\bar{\lambda}_{0,\text{PDIP}} = \mathbf{1}_{m+\tilde{m}}$.

Scenario 2. Algorithm 4 is *warm-started* with $\bar{\mathbf{z}}_{0,\text{PDIP}} := [\mathbf{z}_{\text{LS}}^{\text{T}} \ \epsilon_s(\mathbf{z}_{\text{LS}})^{\text{T}}]^{\text{T}}$ as described in Scenario 1 and $(\bar{s}_{0,\text{PDIP}}, \bar{\lambda}_{0,\text{PDIP}})$ related to $\bar{\mathbf{z}}_{0,\text{PDIP}}$ in (5-9).

Scenario 3. Algorithm 5 with Step 8 replaced by[1]

$$f_0(\mathbf{z}_{\text{DFG}}) - d(\lambda_{\text{DFG}}) \leq \epsilon_{\text{DFG}} = 10^{-2/-3}$$

Scenario 4. Algorithm 5 with Step 8 replaced by[2]

$$f_0(\mathbf{z}_{\text{DFG}}) - d(\lambda_{\text{DFG}}) \leq \epsilon_{\text{DFG}} = 10^{-4/-5}$$

Scenario 5. Algorithm 5 with Step 8 replaced by

$$f_0(\mathbf{z}_{\text{DFG}}) - d(\lambda_{\text{DFG}}) \leq \epsilon_{\text{DFG}} = 10^{-6}$$

Remark. Scenarios 1 and 2 rely on two different initializations of the dual variables to show the impacts that their initialization have on the behavior of the solver. To the best of our knowledge, many off-the-shelf interior-point solvers do not allow the user to access the dual variables for their initialization (e.g., MATLAB's `quadprog`). A default choice is the one proposed in Scenario 1 in which the dual variables are initialized to $\mathbf{1}$. The warm-starting strategy proposed in Scenario 2 considers $\bar{\lambda}_{0,\text{PDIP}}$ corresponding to the violated (and active) constraints with solution $\bar{\mathbf{z}}_{0,\text{PDIP}} := [\mathbf{z}_{\text{LS}}^{\text{T}} \ \epsilon_s(\mathbf{z}_{\text{LS}})^{\text{T}}]^{\text{T}}$, in order to enable Algorithm 4 to maintain information on duality gap at $\mathbf{z}_{\text{LS}}$ with softening parameter $\epsilon_s(\mathbf{z}_{\text{LS}})$, detailed in (5-10).

[1] With accuracy $\epsilon_{\text{DFG}} = 10^{-2}$ or 10^{-3}, dual fast gradient solver in Algorithm 2 returns the same result when solving Problem (2-11).

[2] With accuracy $\epsilon_{\text{DFG}} = 10^{-4}$ or 10^{-5}, dual fast gradient solver in Algorithm 2 returns the same result when solving Problem (2-11).

Table 6-1 compares Algorithm 5 (Scenario 3, 4 and 5) with Algorithm 4 (Scenarios 1 and 2) in terms of Newton iterations. Considering warm-starting approach applied in Scenario 1 and 2, Algorithm 4 requires fewer Newton iterations with $\bar{\lambda}_{0,\text{PDIP}}$ initialized by (5-9). The reason is that, with soft constraint approach, the primal feasibility violation is introduced in the cost function, i.e., (5-6a). Therefore, compared to initialization strategy of $\bar{\lambda}_{0,\text{PDIP}}$ in Scenario 1 when solving Problem (5-6), Scenario 2 has a starting point that is closer to the optimum. From columns 3-4 in Table 6-1 we can see that as DFG solvers is terminated at a higher accuracy, fewer iterations in damped Newton phase is required. As DFG is terminated at the accuracy $\epsilon_{\text{DFG}} = 10^{-6}$, the damped Newton phase can be completely removed. Figure 6-1 shows the computational time of Scenario 1-5. In Figure 6-1, $f_0(\mathbf{z}^*)$ is the optimal cost corresponding to $\mathbf{z}^*$, computed by MATLAB's `quadprog`. PDIP terminates when the duality gap satisfies $\bar{\mu} \leq 10^{-12}$. The blue dash-dot line shows the computational time of Algorithm 4 in Scenario 1. The red dash-dot line shows the computational time of Algorithm 4 in Scenario 2. The orange solid line shows the computational time of Algorithm 2 reaching a high accuracy ($\approx 10^{-12}$) by 5155 iterations. The purple solid line shows the computational time of Algorithm 5 in Scenario 3. The green line shows the computational time of Algorithm 5 in Scenario 4. The light-blue bold solid line shows the computational time of Algorithm 5 in Scenario 5. Note that Scenario 5 requires 591 DFG iterations to eliminate damped Newton phase in PDIP phase. Compared to Scenario 1 and 2, Scenario 5 shows improvement on time efficiency since damped Newton phase is replaced by DFG iterations and therefore no backtracking line search is needed.

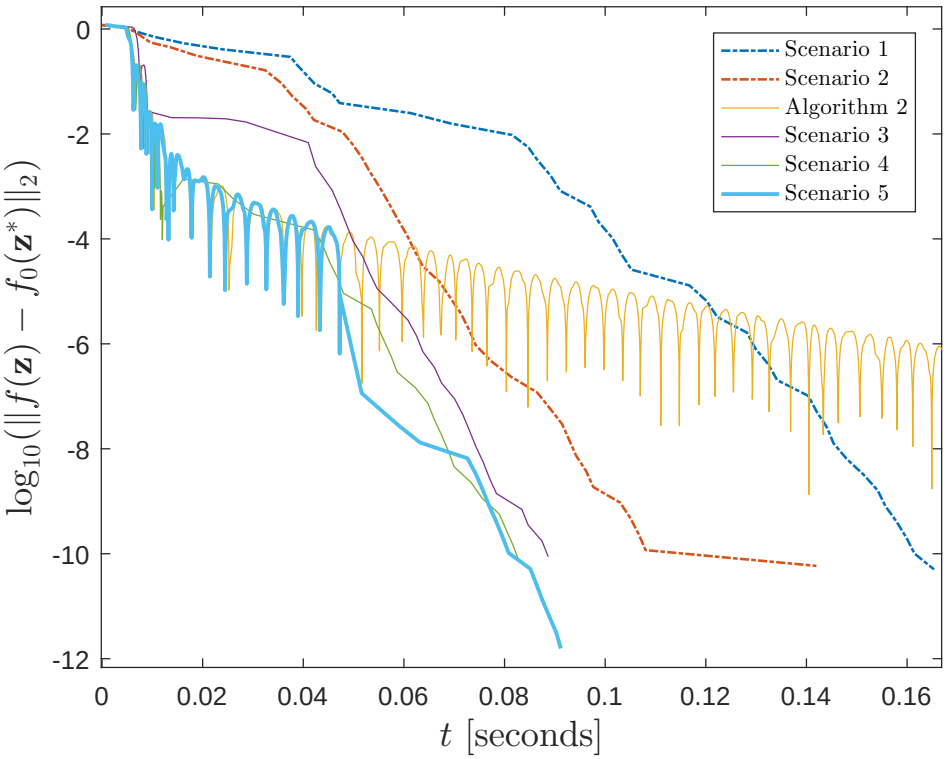


Figure 6-1: Time required for online optimization using Algorithms 2, 4, and 5 in Scenarios 1-5.

Table 6-1: Iterations of Scenarios 1-5.

		Scenario 1	*Scenario 2*	*Scenario 3*	*Scenario 4*	*Scenario 5*
DFG		–	–	35	108	591
PDIP	*Damped Newton Phase*	19	16	9	5	–
	Pure Newton Phase	23	23	22	22	18

6-1-2 Constraint Tightening Approach in Dual Fast Gradient Methods

For Algorithm 6, we compare the following five scenarios:

Scenario 6. Algorithm 4 is *warm-started* with $\mathbf{z}_{0,\text{PDIP}} := \mathbf{z}_{\text{LS}}$, which is the optimal solution of the unconstrained problem, that is, a least-squares (LS) problem, associated with Problem (2-11), and $\lambda_{0,\text{PDIP}} = \mathbf{1}_m$.

Scenario 7. Algorithm 4 is *warm-started* with $\mathbf{z}_{0,\text{PDIP}} := \mathbf{z}_{\text{LS}}$ as in Scenario 6 and $\lambda_{0,\text{PDIP}} = 10^{-6}\mathbf{1}_m$.

Scenario 8. Algorithm 6 with $\epsilon_c = 0.0589$ and Step 8 replaced by[3]

$$f_0(\mathbf{z}_{\text{DFG}}) - d(\lambda_{c,\text{DFG}}) \leq \epsilon_{\text{DFG}} = 10^{-2/-3}$$

Scenario 9. Algorithm 6 with $\epsilon_c = 0.0025$ and Step 8 replaced by[4]

$$f_0(\mathbf{z}_{\text{DFG}}) - d(\lambda_{c,\text{DFG}}) \leq \epsilon_{\text{DFG}} = 10^{-4/-5}$$

Scenario 10. Algorithm 6 with $\epsilon_c = 1.4008 \times 10^{-6}$ and Step 8 replaced by

$$f_0(\mathbf{z}_{\text{DFG}}) - d(\lambda_{c,\text{DFG}}) \leq \epsilon_{\text{DFG}} = 10^{-6}$$

Remark. Scenarios 6 and 7 rely on two different initializations of the dual variables to show the impacts that their initialization have on the behavior of the solver. A default choice is the one proposed in Scenario 6 in which the dual variables are initialized to $\mathbf{1}$. The warm-starting strategy proposed in Scenario 7 considers $\lambda_{0,\text{PDIP}}$ close to zero, which is the optimal value of the multipliers associated with the unconstrained LS problem.

Table 6-2 compares Algorithm 6 (Scenario 8, 9 and 10) with Algorithm 4 (Scenarios 6 and 7) in terms of Newton iterations. Algorithm 4 requires 45 iterations in Scenario 6, while it requires 43 iterations in Scenario 7. Notice, however, that, while the total number of iterations in Scenario 7 is reduced, Algorithm 4 requires more damped-Newton-phase iterations. The main reason is that the initialization in Scenario 7 is farther from the optimal value, given that it is initialized with the pair ($\mathbf{z}_{\text{LS}},\ \lambda_0 \approx 0$), which assumes no active constraints (while

[3]With accuracy $\epsilon_{\text{DFG}} = 10^{-2}$ or 10^{-3}, dual fast gradient solver in Algorithm 2 returns the same result when solving Problem (2-11).

[4]With accuracy $\epsilon_{\text{DFG}} = 10^{-4}$ or 10^{-5}, dual fast gradient solver in Algorithm 2 returns the same result when solving Problem (2-11).

constraints are active at the optimum). If we compare the results in Table 6-1 and 6-2, they are similar. Columns 3-4 in Table 6-2 shows that when DFG solvers is terminated at a higher accuracy, fewer iterations in damped Newton phase is required. As DFG is terminated at the accuracy $\epsilon_{\text{DFG}} = 10^{-6}$, the damped Newton phase can be completely removed. PDIP terminates when the duality gap satisfies $\mu \leq 10^{-12}$. Figure 6-2 shows the computational time of Scenario 6-10. The blue dash-dot line shows the computational time of Algorithm 4 in Scenario 6. The red dash-dot line shows the computational time of Algorithm 4 in Scenario 7. The orange solid line shows the computational time of Algorithm 2 reaching a high accuracy ($\approx 10^{-12}$) by 5155 iterations without tightening constraint approach. The purple solid line shows the computational time of Algorithm 6 in Scenario 8. The green line shows the computational time of Algorithm 6 in Scenario 9. The light-blue bold solid line shows the computational time of Algorithm 6 in Scenario 10. Note that Scenario 10 requires 590 DFG iterations to eliminate damped Newton phase in PDIP phase. Compared to Scenario 6 and 7, Scenario 10 shows similar improvement on time efficiency with constraint tightening approach to Figure 6-1.

Remark. Note that, compared to Figure 6-1, there are gaps in DFG phase between the plots of different scenarios in Figure 6-2, while in Figure 6-1 the computational time of DFG phase in each scenarios matches almost completely. The reason is that in Figure 6-2, Scenario 8-10 solve QP problem with tightened constraints, i.e., Problem (5-18), and the tightening parameters are different for different terminating accuracy of DFG. Thus, in each scenario, the DFG algorithm solves a different QP problem, which leads to difference in computational time.

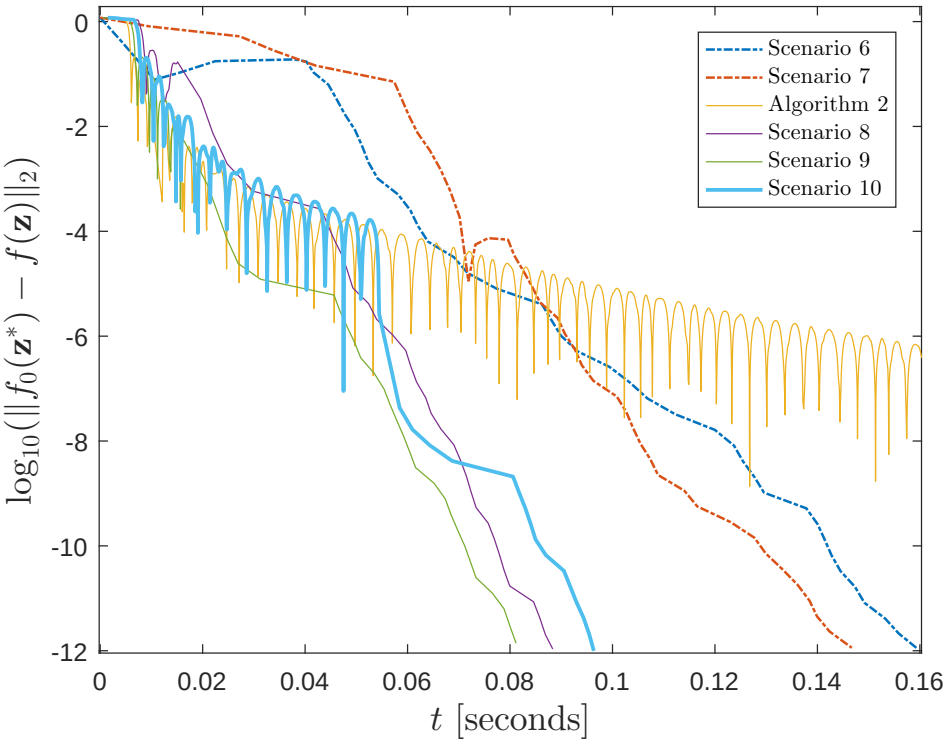


Figure 6-2: Time required for online optimization using Algorithms 2, 4, and 6 in Scenarios 6-10.

Table 6-2: Iterations of Scenarios 6-10.

		Scenario 6	*Scenario 7*	*Scenario 8*	*Scenario 9*	*Scenario 10*
DFG		–	–	68	79	590
PDIP	*Damped Newton Phase*	6	16	13	7	–
	Pure Newton Phase	39	27	21	21	17

6-2 Results for Dual Switching Condition

As mentioned in Section 5-4, L_{dH} can be selected as any real number greater than zero. In this respect, we chose $L_{dH} = 150$. Furthermore, $m_d = 1.4529$ and, consequently $\eta_d = 0.0097$. Finally, the backtracking line search parameters for Algorithm 4 are chosen as $\alpha = \frac{1}{3}$, $\beta = 0.5$.

We compared the Scenario 6, 7 and Algorithm 7.

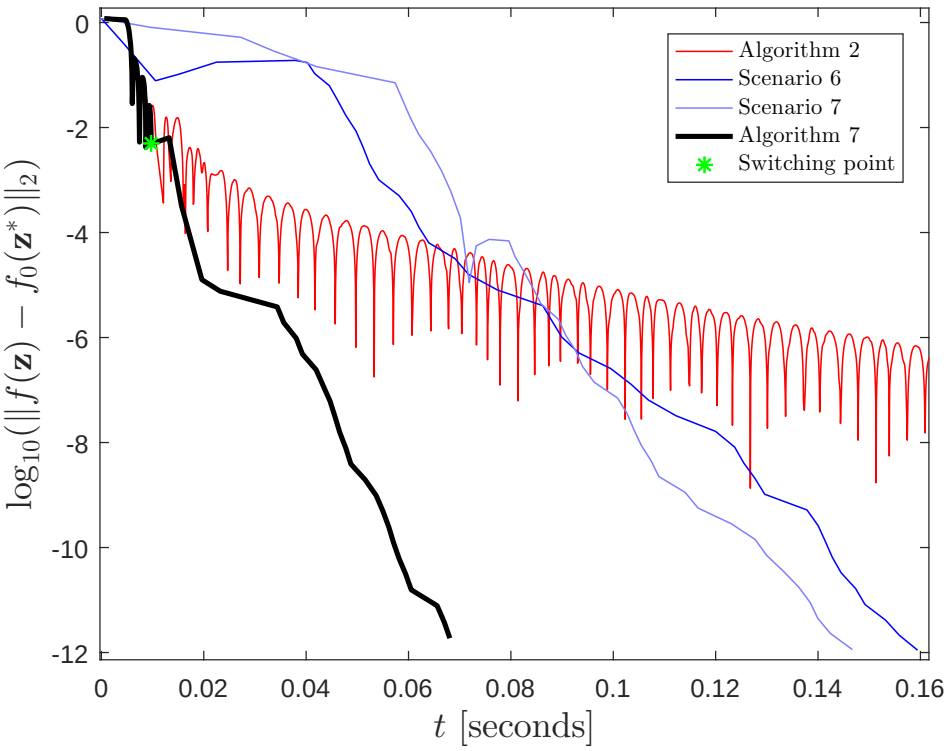


Figure 6-3: Time required for online optimization using Algorithms 2, 4, and 7.

Table 6-3 compares Algorithm 7 with Algorithm 4 (Scenarios 6 and 7) in terms of Newton iterations. As Table 6-3 shows, Algorithm 7 reduces the number of iterations to 28 and completely eliminates the damped Newton phase, thanks to the use of the DFG to initialize the interior-point iterates. This leads to significant improvements also from the computation point of view as Figure 6-3 depicts. In particular, Figure 6-3 shows the computation time required to solve Problem (2-11) online using the proposed algorithm, i.e., Algorithm 7. In Figure 6-3,

Table 6-3: Iterations of Algorithms 4 (columns 1-2) and 7 (column 3).

	Scenario 6	*Scenario 7*	*Algorithm 7*
Damped Newton Phase	6	16	–
Pure Newton Phase	39	27	28

$f_0(\mathbf{z}^*)$ is the optimal cost corresponding to $\mathbf{z}^*$, computed by MATLAB's `quadprog`. PDIP terminates when the duality gap satisfies $\mu \leq \varepsilon = 10^{-12}$. The blue solid line shows the computation time of Algorithm 4 in Scenario 6. The light-blue solid line is the computation time of Algorithm 4 in Scenario 7. The red solid line shows the computation time of Algorithm 2. The black bold line shows our proposed solver, Algorithm 7. The green star indicates the switching point when the solution returned by DFG satisfies condition (5-41) ($\mu \approx 10^{-3}$). Note that, Algorithm 2 requires 5155 iterations to reach a high accuracy ($\approx 10^{-12}$), while DFG phase only requires 65 DFG iterates to reach the medium accuracy needed and enter the pure Newton phase in Algorithm 7. Furthermore, note that the improvements in terms of computation time compared to Scenarios 6 and 7 are related to the following facts: (i) our algorithm does not require backtracking line search (computationally costly), and (ii) the damped Newton phase is replaced by DFG iterates.

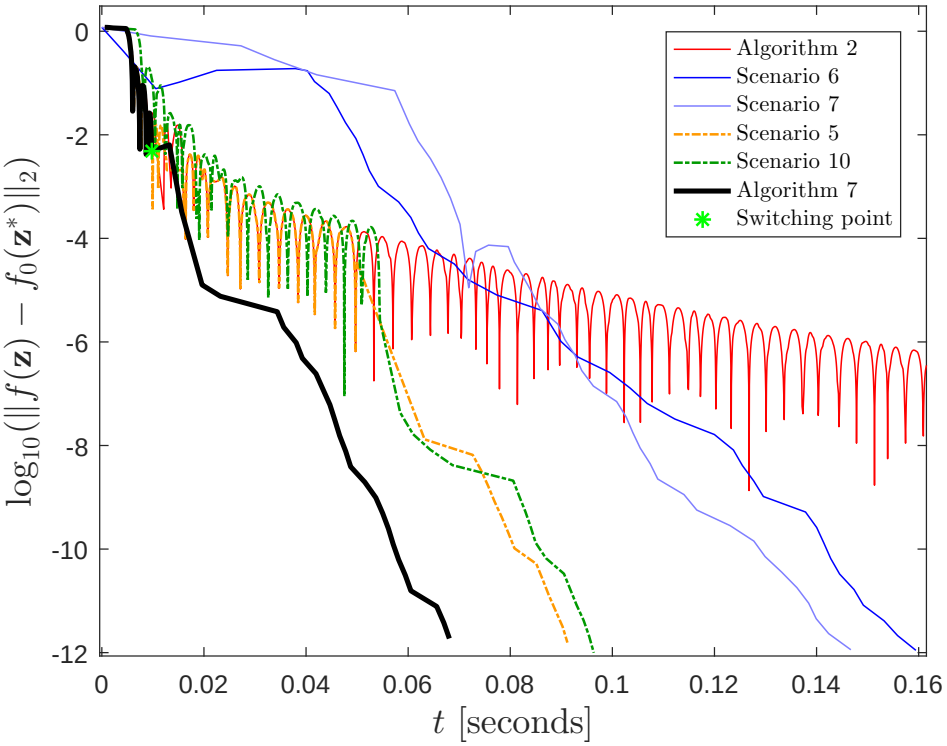


Figure 6-4: Time required for online optimization using Algorithms 2, 4, Algorithm 5, Algorithm 6 and 7.

Finally, we make a comparison on computational time in Figure 6-4 between Algorithm 2, 4 (Scenario 6 and 7), 5 (Scenario 5), 6 (Scenario 10) and Algorithm 7. From Figure 6-4,

we can see that without constraint softening or tightening approach, Algorithm 7 is more computationally efficient by allowing primal infeasible iterations in PDIP phase, and thanks to the switching strategy in (5-41) derived in dual space which can be checked easily in each DFG iteration.

Remark. Figure 6-2, 6-3 and 6-4 monitor $\|f_0(\mathbf{z}_k) - f_0(\mathbf{z}^*)\|_2$ (in logarithmic scale). When Algorithm 4 is initialized with the optimal solution $\mathbf{z}_0 := \mathbf{z}_{\text{LS}}$ (Scenarios 6 and 7), $f_0(\mathbf{z}_0) < f_0(\mathbf{z}^*)$, given that it does not account for the presence of (active) constraints at the optimum. This leads to the nonmonotonic behavior at 0.02 sec (blue line) and 0.08 sec (light-blue line) in Figure 6-3, when PDIP enters the feasible region (and in pure Newton phase).

Chapter 7

Conclusion and Future Work

The most significant contribution of this master thesis is that it proposes a novel primal-dual interior-point method for optimization problems that typically arise from model predictive control applications, i.e., quadratic programming problems with linear inequality constraints. The proposed solver improves the convergence of state-of-the-art primal-dual interior-point methods by replacing the damped Newton phase, which converges linearly, with a dual fast method, that converges, instead, super-linearly. This result is obtained by working in the dual space and modifying the condition to enter the pure Newton phase. Being able to start primal-dual interior-point method directly from pure Newton phase, the computational effort required by backtracking line search can also be saved. Finally, we show the benefits of the proposed algorithm on a discrete unstable planar system. This work has been submitted for publication at IFAC WC 2017.

In addition, the studies on the convergence of primal-dual interior-point method in primal space are explained in details. Although the switching condition for primal-dual interior-point methods remains conservative in primal space, the methods proposed in Section 5-1 and 5-2 can also earn benefits by reducing number of iterations in damped Newton phase. These two methods can be applied by setting the stopping criterion of DFG as

$$f_0(\mathbf{z}^*) - d(\lambda) \leq \epsilon_{\mathrm{DFG}} \tag{7-1}$$

usually, 10^{-6} is a reasonable choice for ϵ_{DFG}. These two method would be preferred in the following situations:

- DFG + PDIP with soft constraint approach (Section 5-1): one does not know whether a feasible solution exists.
- DFG with constraint tightening approach + PDIP (Section 5-2): more strict executional time limitation which leads to high possibility that the algorithm would end prematurely.

We aim to extend the initialization strategy to active set methods, where the dual fast gradient can be used to determine the set of active constraints more efficiently. Furthermore, we aim to test the proposed algorithm on a more realistic (large scale) application.

Appendix A

Appendix

A-1 Proof of Theorem 5.2

Before we prove the theorem, some important definitions and conclusions are presented beforehand.

A-1-1 Newton Increment

For Newton's method in primal space, the search direction is the inverse of the primal Hessian times the first derivative. With step size properly chosen, this search direction will lead to a decrease in the cost function. Therefore, the search direction of Newton's method in primal space is a descent direction.

In dual space, the following dual function corresponding to the primal problem is maximized with Newton's method:

$$\max_{\lambda \in \mathbb{R}^m_+} \; d(\lambda) = \mathcal{L}(\mathbf{z}^*(\lambda), \lambda) \tag{A-1}$$

and we assume that the inner problem

$$\mathbf{z}^*(\lambda) = \arg\min_{\mathbf{z} \in \mathbb{R}^n} \; \mathcal{L}(\mathbf{z}, \lambda)$$

can be solved exactly.

According to the strategy followed in [5, Chapter 9], the first-order approximation of dual function $d(\lambda)$ is used.

$$d(\lambda + \Delta\lambda_{\text{nt}}) = d(\lambda) + \nabla d(\lambda)^T \Delta\lambda_{\text{nt}} \tag{A-2}$$

At the optimum, the first-order optimality condition shows that

$$\nabla d(\lambda + \Delta\lambda_{\text{nt}}) = \nabla d(\lambda) + \nabla^2 d(\lambda)\Delta\lambda_{\text{nt}} = 0 \tag{A-3}$$

From the optimality condition in (A-3), the Newton search direction in dual space can be computed:

$$\begin{aligned} \Delta\lambda_{\text{nt}} &= -\nabla^2 d(\lambda)^{-1}\nabla d(\lambda) \\ &\geq 0 \end{aligned} \tag{A-4}$$

Since dual function is a concave function, the dual Hessian $-\nabla^2 d(\lambda)$ is negative definite. Therefore, $\Delta\lambda_{\text{nt}}$ is an ascent direction in dual space.

Definition A.1. The Newton increment of dual function $d(\lambda)$ can be defined as:

$$\begin{aligned} \nu(\lambda) &= \left[-\nabla d(\lambda)^T \nabla^2 d(\lambda)^{-1}\nabla d(\lambda)\right]^{\frac{1}{2}} \\ &= \left[-\Delta\lambda_{\text{nt}}^T \nabla^2 d(\lambda)\Delta\lambda_{\text{nt}}\right]^{\frac{1}{2}} \end{aligned} \tag{A-5}$$

Thus, the following holds:

$$\begin{aligned} \nu(\lambda)^2 &= -\nabla d(\lambda)^T \nabla^2 d(\lambda)^{-1}\nabla d(\lambda) \\ &= -\Delta\lambda_{\text{nt}}^T \nabla^2 d(\lambda)\Delta\lambda_{\text{nt}} \end{aligned} \tag{A-6}$$

and

$$\nu(\lambda)^2 = \nabla d(\lambda)^T \Delta\lambda_{\text{nt}} \tag{A-7}$$

Another important conclusion used in the proof afterwards is derived based on the assumption of strong convexity (5-33) and (A-6):

$$-\nu(\lambda)^2 \leq -m_d \left\|\Delta\lambda_{\text{nt}}\right\|_2^2 \tag{A-8}$$

Define the second-order Taylor approximation $\hat{d}(\lambda)$ of $d(\lambda)$ at λ as:

$$\hat{d}(\lambda + \Delta\lambda_{\text{nt}}) = d(\lambda) + \nabla d(\lambda)^T \Delta\lambda_{\text{nt}} + \frac{1}{2}\Delta\lambda_{\text{nt}}^T \nabla^2 d(\lambda)\Delta\lambda_{\text{nt}} \tag{A-9}$$

Since strong duality holds (Assumption 2.1), $\sup \hat{d}(\lambda)$ is the optimum of original problem (5-4). Then, Newton increment can be related to the quantity:

$$d(\lambda) - \sup \hat{d}(\lambda + \Delta\lambda_{\text{nt}}) = \frac{1}{2}\Delta\lambda_{\text{nt}}^T \nabla^2 d(\lambda)\Delta\lambda_{\text{nt}} = -\frac{1}{2}\nu(\lambda)^2 \leq 0 \tag{A-10}$$

Therefore, the Newton increment can serve as an estimation of suboptimality.

A-1-2 Proof of Quadratic Convergence

• *Step Size Selected by Backtracking Line Search*

In this part, we prove that, if

$$\left\|\nabla d(\lambda)\right\|_2 \leq \frac{m_d^2}{L_{dH}} \tag{A-11}$$

the step size of backtracking line search in dual space will select the step size as one, i.e., $\rho = 1$.

By Lipschitz condition on the dual Hessian $\nabla^2 d(\lambda)$, then for $\rho \geq 0$, the following relation holds:

$$\left\|\nabla^2 d(\lambda + \rho\Delta\lambda_{\mathrm{nt}}) - \nabla^2 d(\lambda)\right\|_2 \leq \rho L_{dH} \left\|\Delta\lambda_{\mathrm{nt}}\right\|_2 \tag{A-12}$$

Multiply (A-12) by $\Delta\lambda_{\mathrm{nt}}$ on both sides and obtain:

$$\left|\Delta\lambda_{\mathrm{nt}}^T \left[\nabla^2 d(\lambda + \rho\Delta\lambda_{\mathrm{nt}}) - \nabla^2 d(\lambda)\right] \Delta\lambda_{\mathrm{nt}}\right| \leq \rho L_{dH} \left\|\Delta\lambda_{\mathrm{nt}}\right\|_2^3 \tag{A-13}$$

Define a function w.r.t step size variable ρ as $\tilde{d}(\rho) = d(\lambda + \rho\Delta\lambda_{\mathrm{nt}})$ and then we have $\nabla_\rho^2 \tilde{d}(\rho) = \Delta\lambda_{\mathrm{nt}}^T \nabla^2 d(\lambda + \rho\Delta\lambda_{\mathrm{nt}})\Delta\lambda_{\mathrm{nt}}$. Therefore, (A-13) can be written as:

$$\left|\nabla_\rho^2 \tilde{d}(\rho) - \nabla_\rho^2 \tilde{d}(0)\right| \leq \rho L_{dH} \left\|\Delta\lambda_{\mathrm{nt}}\right\|_2^3 \tag{A-14}$$

In order to prove the conclusion on backtracking line search, the following inequality is used to find an lower bound on $\tilde{d}(\rho)$:

$$\begin{aligned} 0 = \nabla_\rho^2 \tilde{d}(\rho) - \nabla_\rho^2 \tilde{d}(0) &\geq -\rho L_{dH} \left\|\Delta\lambda_{\mathrm{nt}}\right\|_2^3 \\ &\geq -\rho \frac{L_{dH}}{m_d^{3/2}} \nu(\lambda)^3 \end{aligned} \tag{A-15}$$

and this will leads to

$$\nabla_\rho^2 \tilde{d}(\rho) \geq -\nu(\lambda)^2 - \rho \frac{L_{dH}}{m_d^{3/2}} \nu(\lambda)^3 \tag{A-16}$$

From (A-15) to (A-15), we use $\nabla_\rho^2 \tilde{d}(0) = -\nu(\lambda)^2$ according to (A-6) and (A-8).

Since $\nabla_\rho \tilde{d}(0) = \nabla d(\lambda)^T \Delta\lambda_{\mathrm{nt}} = \nu(\lambda)^2$ according to (A-7), we integrate this equation over ρ to obtain

$$\begin{aligned} \nabla_\rho \tilde{d}(\rho) &\geq \nabla_\rho \tilde{d}(0) - \rho\nu(\lambda)^2 - \rho^2 \frac{L_{dH}}{m_d^{3/2}} \nu(\lambda^3) \\ &= \nu(\lambda)^2 - \rho\nu(\lambda)^2 - \rho^2 \frac{L_{dH}}{m_d^{3/2}} \nu(\lambda^3) \end{aligned} \tag{A-17}$$

Integrate the equation above once more and get

$$\tilde{d}(\rho) \geq \tilde{d}(0) + \rho\nu(\lambda)^2 - \frac{\rho^2}{2}\nu(\lambda)^2 - \rho^3 \frac{L_{dH}}{6m_d^{3/2}} \nu(\lambda)^3 \tag{A-18}$$

Now suppose the switching condition to entering pure Newton phase is satisfied

$$\left\|\nabla d(\lambda)\right\|_2 \leq \eta_d \leq \frac{3(1-2\alpha)m_d^2}{L_{dH}} \tag{A-19}$$

in which with the backtracking line search parameter $\alpha = \frac{1}{3}$ yields the condition in (A-11). Moreover, by strong convexity, we can derive the following relation:

$$\nu(\lambda) \leq \frac{3(1-2\alpha)m_d^{3/2}}{L_{dH}} \tag{A-20}$$

If the step size ρ is set to one, (A-18) becomes:

$$\begin{aligned} d(\lambda+\Delta\lambda_{\mathrm{nt}}) &\geq d(\lambda)+\frac{1}{2}\nu(\lambda)^2-\frac{L}{6m_d^{3/2}}\nu(\lambda)^3 \\ &\geq d(\lambda)+\frac{1}{2}\nu(\lambda)^2+\frac{L}{6m_d^{3/2}}\nu(\lambda)^2\left[\frac{3(2\alpha-1)m_d^{3/2}}{L_{dH}}\right] \\ &= d(\lambda)+\nu(\lambda)^2\left[\frac{1}{2}+\frac{L}{6m_d^{3/2}}\frac{3(2\alpha-1)m_d^{3/2}}{L_{dH}}\right] \\ &= d(\lambda)+\alpha\nu(\lambda)^2 \\ &= d(\lambda)+\alpha\nabla^2 d(\lambda)^T\Delta\lambda_{\mathrm{nt}} \end{aligned} \tag{A-21}$$

which shows that the unit step $\rho = 1$ is accepted by the backtracking line search in dual space.

• *Quadratic Convergence in Dual Space*

Since with condition in (A-11) the backtracking line search will select unit step, by Lipschitz continuity, we can derive the following relation of the first-order derivative of the dual function between two iterations:

$$\begin{aligned} \|\nabla d(\lambda_+)\|_2 &= \left\|\int_0^1\left[\nabla^2 d(\lambda+\rho\Delta\lambda_{\mathrm{nt}})-\nabla^2 d(\lambda)\right]\Delta\lambda_{\mathrm{nt}}d\rho\right\|_2 \\ &\leq \frac{L_{dH}}{2}\|\Delta\lambda_{\mathrm{nt}}\|_2^2 \\ &= \frac{L_{dH}}{2}\left\|\nabla^2 d(\lambda)^{-1}\nabla d(\lambda)\right\|_2^2 \\ &\leq \frac{L_{dH}}{2m_d^2}\|\nabla d(\lambda)\|_2^2 \end{aligned} \tag{A-22}$$

Therefore, for $\bar{k} > 0$, recursively we can have

$$\frac{L_{dH}}{2m_d^2}\|\nabla d(\lambda_{\bar{k}})\|_2 \leq \left(\frac{L_{dH}}{2m_d^2}\|\nabla d(\lambda)\|_2\right)^2 \leq \left(\frac{L_{dH}}{2m_d^2}\cdot\frac{m_d^2}{L_{dH}}\right)^{2^{\bar{k}}} = \left(\frac{1}{2}\right)^{2^{\bar{k}}} \tag{A-23}$$

From the property of concave function:

$$d(\mu) \leq d(\lambda)+\nabla d(\lambda)^T(\mu-\lambda)-\frac{m}{2}\|\mu-\lambda\|_2^2 \tag{A-24}$$

we can have

$$d(\mu) \leq d(\lambda)+\|\nabla d(\lambda)\|_2\|\mu-\lambda\|_2-\frac{m_d}{2}\|\mu-\lambda\|_2^2 \tag{A-25}$$

Thus,

$$\begin{aligned} d(\lambda_+)-d(\lambda) &\leq \|\nabla d(\lambda)\|_2\|\lambda_+-\lambda\|_2-\frac{m_d}{2}\|\lambda_+-\lambda\|_2 \\ &\leq \|\nabla d(\lambda)\|_2\left\|\nabla^2 d(\lambda)^{-1}\nabla d(\lambda)\right\|_2-\frac{m_d}{2}\left\|\nabla^2 d(\lambda)^{-1}\nabla d(\lambda)\right\|_2 \\ &\leq \frac{1}{2m_d}\|\nabla d(\lambda)\|_2 \end{aligned} \tag{A-26}$$

Combining (A-23) and (A-26), the following convergence result can be obtained:

$$f_0(\mathbf{z}^*) - d(\lambda_{\bar{k}}) = d(\lambda^*) - d(\lambda_{\bar{k}}) \leq \frac{1}{2m_d} \|\nabla d(\lambda)\|_2 \leq \frac{m_d^3}{L_{dH}^2} \left(\frac{1}{2}\right)^{2^{\bar{k}}} = \left(\frac{m_d}{4L_{dH}}\right)^{\bar{k}} \tag{A-27}$$

which shows that we can achieve quadratic convergence.

Bibliography

[1] R. Bartlett, A. Wächter, and L. Biegler. Active set vs. interior point strategies for model predictive control. In *Proceedings of the American Control Conference*, volume 6, pages 4229–4233. IEEE, 2000.

[2] A. Bemporad, D. Bernardini, and P. Patrinos. A convex feasibility approach to anytime model predictive control. *arXiv preprint arXiv:1502.07974*, 2015.

[3] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999.

[4] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] E. F. Camacho and C. Bordons. *Model predictive control in the process industry*. Springer Science & Business Media, 2012.

[7] N. de Oliveira and L. T. Biegler. Constraint handing and stability properties of model-predictive control. *AIChE journal*, 40(7):1138–1155, 1994.

[8] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 668–674. IEEE, 2012.

[9] M. F.Borrelli, A. Bemporad. Predictive control for linear and hybrid systems. http://www.mpc.berkeley.edu/mpc-course-material.

[10] H. J. Ferreau. An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control. *University of Heidelberg*, 2006.

[11] D. Fontanelli, L. Greco, and A. Bicchi. Anytime control algorithms for embedded real-time systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 158–171. Springer, 2008.

[12] P. Giselsson. Improved fast dual gradient methods for embedded model predictive control. *IFAC Proceedings Volumes*, 47(3):2303–2309, 2014.

[13] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829–833, 2013.

[14] P. Giselsson and A. Rantzer. On feasibility, stability and performance in distributed model predictive control. *IEEE Transactions on Automatic Control*, 59(4):1031–1036, 2014.

[15] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013. http://control.ee.ethz.ch/~mpt.

[16] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari. Embedded predictive control on an fpga using the fast gradient method. In *Proc. European Control Conference*, 2013.

[17] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides. A condensed and sparse qp formulation for predictive control. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5217–5222. IEEE, 2011.

[18] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM, 1984.

[19] E. C. Kerrigan and J. M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In *Control 2000 Conference, Cambridge*, 2000.

[20] M. Kögel and R. Findeisen. Fast predictive control of linear systems combining Nesterov's gradient method and the method of multipliers. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 501–506. IEEE, 2011.

[21] J. M. Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[22] L. Magni, D. M. Raimondo, and F. Allgöwer. *Nonlinear model predictive control*. Springer, 2009.

[23] J. Mattingley and S. Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

[24] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.

[25] I. Necoara, L. Ferranti, and T. Keviczky. An adaptive constraint tightening approach to linear model predictive control based on approximation algorithms for optimization. *Optimal Control Applications and Methods*, 36(5):648–666, 2015.

[26] I. Necoara and V. Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5):1232–1243, 2014.

[27] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[28] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[29] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[30] C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. *Journal of optimization theory and applications*, 99(3):723–757, 1998.

[31] S. Richter, C. N. Jones, and M. Morari. Real-time input-constrained MPC using fast gradient methods. In *Proc. of the 48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 7387–7393. IEEE, 2009.

[32] S. Richter, M. Morari, and C. N. Jones. Towards computational complexity certification for constrained mpc based on lagrange relaxation and the fast gradient method. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5223–5229. IEEE, 2011.

[33] N. L. Ricker. Use of quadratic programming for constrained internal model control. *Industrial & Engineering Chemistry Process Design and Development*, 24(4):925–936, 1985.

[34] M. Rubagotti, P. Patrinos, and A. Bemporad. Stabilizing linear model predictive control under inexact numerical optimization. *IEEE Transactions on Automatic Control*, 59(6):1660–1666, 2014.

[35] C. Schmid and L. T. Biegler. Quadratic programming methods for reduced hessian SQP. *Computers & chemical engineering*, 18(9):817–832, 1994.

[36] F. Ullmann. Fiordos: A matlab toolbox for c-code generation for first order methods. *no. December*, 2011.

[37] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.

[38] S. J. Wright. Applying new optimization algorithms to model predictive control. In *AIChE Symposium Series*, volume 93, pages 147–155. Citeseer, 1997.

[39] M. N. Zeilinger, C. N. Jones, and M. Morari. Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization. *IEEE transactions on automatic control*, 56(7):1524–1534, 2011.

[40] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones. On real-time robust model predictive control. *Automatica*, 50(3):683–694, 2014.

[41] P. Zometa, M. Kögel, and R. Findeisen. muAO-MPC: A free code generation tool for embedded real-time linear model predictive control. In *Proc. American Control Conference (ACC), 2013*, pages 5340–5345, Washington D.C., USA, 2013.

Glossary

List of Acronyms

3mE	Mechanical, Maritime and Materials Engineering
ARE	Algebraic Riccati Equation
DCSC	Delft Center for Systems and Control
DFG	Dual Fast Gradient Method
IP	Interior-Point Methods
KKT	Karush-Kuhn-Tucker Conditions
LTI	Linear Time-Invariant
LS	Least-Squares
MPC	Model Predictive Control
mp-QP	Multi-Parametric Quadratic Programming
PID	Proportional-Integral-Derivative Controller
PDIP	Primal-Dual Interior-Point Method
PWA	Piecewise Affine Function
QP	Quadratic Programming

List of Notations

$\|\cdot\|_1$	1-Euclidean norm
$\|\cdot\|_2$	2-Euclidean norm

$\|\cdot\|_\infty$	Infinity norm
$\|\cdot\|_D$	Dual norm
$\|\cdot\|_Q^2$	Quadratic norm with weight matrix Q, e.g., $\|\mathbf{z}\|_Q^2 = \mathbf{z}^{\mathrm{T}} Q \mathbf{z}$
$[\,\cdot\,]_{+/-}$	Projection on nonnegative or negative orthant
$\otimes$	Tensor product
$\nabla^n(\cdot)$	n^{th} derivative of a function
diag$(\cdot)$	Matrix built from a vector with entries on its main diagonal
relint$(\cdot)$	Relative interior of a set
$\lfloor \alpha \rfloor$	Largest integer which is less than or equal to α which is a real number

List of Symbols

t	Sampling instant
$x(t)$	System state
$u(t)$	System input
$y(t)$	System output
$A,\ B,\ C,\ D$	System matrices
$\mathcal{X},\ \mathcal{U},\ \mathcal{Y}$	Feasible sets for system state, input and output
N	Prediction horizon
$J_t(x_j, u_j)$	Stage cost
$J_N(x_N, u_N)$	Terminal cost
$J(x, u)$	Cost function
$Q,\ R$	Weight matrices on system state and input in state cost
P	Weight matrix on terminal state
x_0	Current state, $x_0 := x(t)$
$\mathbf{z}$	Decision variable in the QP problem formulate from MPC problem, primal variable
$\mathcal{Z}$	Feasible set for decision variable $\mathbf{z}$

$A_N,\ B_N,\ C_N,\ D_N$	System matrices over prediction horizon
H	Primal Hessian
h	Coefficient vector in primal cost function
$g(\cdot)$	Function of inequality constraints w.r.t decision variable, affine.
$f(\cdot)$	Function of equality constraints w.r.t decision variable, affine.
$G,\ E,\ g$	Coefficient matrices or vectors in function $g(\cdot)$
$f_0(\mathbf{z}^*)$	Primal optimum
n	Dimension of decision variable
m	Number of independent inequality constraints
$m_p,\ M_p$	Condition numbers associated with strong convexity of the primal optimization problem
L_p	Lipschitz constant of the primal Hessian
$\sigma(\cdot)_{\max\,/\,\min}$	Maximal and minimal eigenvalue of a matrix
$\mathcal{L}(\cdot)$	Lagrangian
λ	Lagrange multiplier
$d(\lambda)$	Dual function
H_d	Dual Hessian
$d(\lambda^*)$	Dual optimum
$m_d,\ M_d$	Condition numbers of dual Hessian
L_d	Lipschitz constant of the first derivative of dual function, i.e., $\nabla d(\lambda)$
L_{dH}	Lipschitz constant of dual Hessian
$\hat{\mathbf{z}}_k$	Average sequence for primal variables
R_d	Variable in dual fast gradient method, defined as $R_d = \|\lambda^* - \lambda_0\|$
ϵ	Tolerance
i	Number of iterations
θ	Barrier parameter
$\phi(\cdot)$	Barrier function
$F_b(\cdot)$	Cost function of unconstrained QP formulated in barrier method

$\mathbf{z}^*(\theta)$	Point on central path
δ	Constant factor
ρ	Step size
$\Delta\mathbf{z}$	Primal search direction
$\Delta\varphi,\ \Delta\zeta$	Primal-dual search directions
s	Slackness variable
μ	Duality gap
κ	Centering parameter
$r_\tau(\cdot)$	Residual variable
r_{dual}	Stationary residual
r_{pri}	Primal residual
r_{cent}	Centering residual with infeasible start approach
$r_{\text{cent,f}}$	Centering residual with feasible start approach
H_{bar}	Primal Hessian in barrier method
$\hat{H}_{\text{pd}},\ H_{\text{pd}}$	Primal Hessian in primal-dual interior point method
$m_{F_b(\cdot)},\ M_{F_b(\cdot)}$	Condition numbers associated with strong convexity of the QP problem formulated in barrier method
$L_{F_b(\cdot)}$	Lipschitz constant of the Hessian of function $F_b(\cdot)$
$\gamma_p(\cdot),\ \eta_p(\cdot)$	Constants related to condition numbers in primal space
$\mathcal{P}_{\text{soft}}$	QP problem with soft constraint approach
ϵ_s	Softening parameter
ω	Weight on primal feasibility violation
$\tilde{m}$	Number of constraints that need to be softened
$\mathcal{G}$	Re-arranged feasible set
$\bar{\mathbf{z}}$	Decision variable consists of primal variable $\mathbf{z}$ and softening parameter ϵ_s
$H_z,\ h_z$	Coefficient matrix and vector in cost function of QP problem with soft constraint approach
$G_z,\ E_z,\ g_z$	Coefficient matrix and vector in function of inequality constraints in QP problem with soft constraint approach

$\bar{\lambda}$	Lagrange multiplier corresponding to constraints in QP with soft constraint approach
$\bar{s}$	Slackness variable corresponding to constraints in QP with soft constraint approach
δ_1	$[g(\mathbf{z})]_+$
δ_3	$[g(\mathbf{z})]_-$
$\tilde{\delta}$	Efficiently small constant
$\mathcal{P}_{\text{tight}}$	QP problem with tightening constraint approach
ϵ_c	Tightening parameter
$\nu(\cdot)$	Newton increment
$\alpha,\ \beta$	Backtracking line search parameters