ABSTRACT

Title of dissertation: INFEASIBLE-START CONSTRAINT-REDUCED METHODS FOR LINEAR AND CONVEX QUADRATIC OPTIMIZATION

Meiyun He, Doctor of Philosophy, 2011

Dissertation directed by: Professor André L. Tits
Department of Electrical and Computer Engineering

Constraint-reduction schemes have been proposed in interior-point methods for *linear and convex quadratic programs* with many more inequality constraints than variables in standard dual form. Such schemes have been shown to be provably convergent and highly efficient in practice. A critical requirement of these schemes, however, is the availability of a dual-feasible initial point.

The thesis first addresses this requirement for linear optimization. Building on a general framework (which encompasses several previously proposed approaches) for dual-feasible constraint-reduced interior-point optimization, for which we prove convergence to a single point of the sequence of dual iterates, we propose a framework for "infeasible" constraint-reduced interior-point. Central to this framework is an exact (ℓ_1 or ℓ_∞) penalty function scheme endowed with a scheme for iterative adjustment of the penalty parameter aiming to yield, after a finite number of updates, a value that guarantees feasibility (for the original problem) of the minimizers. Finiteness of the sequence of penalty parameter adjustments is proved under mild feasibility assumptions for all algorithms that fit within the framework, including "infeasible" extensions of a "dual" algorithm proposed in the early 1990s and of two recently recently proposed "primal-dual" algorithms. One of the latter two, a constraint-reduced variant of Mehrotra's Predictor Corrector algorithm is then more specifically considered. Further convergence results are proved for the corresponding infeasible method. Furthermore, such an infeasible method is analyzed without feasibility assumptions.

Next, the constraint-reduced scheme with arbitrary initial points is extended for the more general case of convex quadratic optimization. A stronger global convergence result is proven that generalizes the result in the linear case.

Numerical results are reported that demonstrate that the infeasible constraint-reduced algorithms are of practical interest. In particular, in the application of model-predictive-control-based rotorcraft control, our algorithms yield a speed-up of over two times for both altitude control and trajectory following.

INFEASIBLE CONSTRAINT REDUCTION
FOR LINEAR AND CONVEX QUADRATIC OPTIMIZATION

by

Meiyun He

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor André L. Tits, Chair/Advisor
Professor Steve I. Marcus
Professor Nuno C. Martins
Professor William S. Levine
Professor Tobias von Petersdorff, Dean's Representative

Acknowledgments

I am thankful to all the people who have helped me with this thesis. Because of them, my graduate study was smooth and has become a very valuable experience in my life.

First, I'd like to thank my advisor, Professor Andre Tits, for supporting me and giving me the unlimited guidance on my study over the past five and a half years. He had been always available when I needed help and advice; I could drop by his office at any time without an appointment. His brightness always inspired me, and his kindness and thoughtfulness allowed us work like friends. I feel so lucky to be his student and have so much to be grateful for what he has taught me.

I would also like to thank Professor Dianne O'Leary. She gave me priceless advices on the proposal of my work. I am indebted to Professor Steve Marcus, Professor Nuno Martins, Professor William Levine and Professor Tobias von Petersdorff who served on the thesis committee in helping and guiding me finish the Ph.D. study. Thank them for their insights and suggestions, time and energy spent on the supervision of my research.

I would like to thank all my colleagues at Department of Electrical and Computer Engineering. There were so many people, even who may not know any optimization, having helped me through discussions and the group study. They were so nice, and always willing to share with me anything they knew. I am grateful to study together with those nice and smart guys.

My officemates in Room 3181 at A.V.Williams building and roommates are all nice people. They were so kind that we shared laughters and tears together. Without them, my graduate life would not be so colorful.

I love my family very much. It is a pity that my father passed away in 2005 and could not say that he was proud of me, as he always did. My father, who was smart and worked hard, taught me a right attitude towards life which is more important to me than knowledge. I am always thankful to my mother, who could not afford much education in her time, has been always trying her best to give me support, concern and love. Words cannot express my gratitude to them.

While it is impossible to remember all, I would like to thank all the people who helped me in this or that way. Thank you all!

Table of Contents

List of Tables

List of Figures

List of Abbreviations

LP	Linear Programming (or Linear Optimization)
QP	Quadratic Programming (or Quadratic Optimization)
CQP	Convex QP
CO	Convex Optimization
CR	Constraint Reduction
NCR	No Constraint Reduction (namely, without Constraint Reduction)
IPM	Interior-Point Method
PDIP	Primal-Dual Interior-Point
PDIPM	Primal-Dual IPM
MPC	Mehrotra's Predictor-Corrector algorithm
RHC	Receding Horizon Control (namely, Model Predictive Control)
rIPM	Constraint-Reduced Interior-Point method
IrPDIP	Infeasible constraint-Reduced Primal-Dual Interior Point method
IrMPC	Infeasible constraint-Reduced Mehrotra Predictor-Corrector Method
IrQP	Infeasible constraint-Reduced method for Quadratic Programming
LTI	Linear Time-Invariant
COAP	Computational Optimization and Applications

Chapter 1

Introduction

1.1 Problem statement

Linear optimization (LP) is a class of optimization problems involving a linear objective function and linear constraints. *Convex quadratic optimization* (CQP), a more general case of LP, is a class of optimization problems involving a convex quadratic objective function and linear constraints. Any LP problem can be transformed to the standard (dual) problem

$$\max\, b^T y \quad \text{s.t.} \quad A^T y \leq c \tag{D}$$

with the associated standard primal form

$$\min\, c^T x \quad \text{s.t.} \quad Ax = b,\, x \geq 0, \tag{P}$$

where $A \in \mathbb{R}^{m\times n}$. More generally, any CQP problem can be transformed to the *standard form* with only inequality constraints

$$\max \quad f(y) := b^T y - \frac{1}{2} y^T H y \quad \text{s.t.} \quad A^T y \leq c \tag{Dq}$$

with the associated Lagrangian dual

$$\min_{x,\,y} \quad c^T x + \frac{1}{2} y^T H y \quad \text{s.t.} \quad Ax + Hy - b = 0,\, x \geq 0. \tag{Pq}$$

where $H \in \mathbb{R}^{m\times m}$ is symmetric and positive semi-definite. For reasons of consistency with the LP literature, we will refer to (Dq) as the *d*ual and to (Pq) as the *p*rimal.

Throughout the paper, we assume that $[H\ A]$ has full row rank for the quadratic optimization; accordingly, this indicates that we assume the full rankness of A in the LP case where $H = 0$.

The case we consider is $n \gg m$; in other words, there are many more constraints than variables in the dual (D) and (Dq). Such situations are detrimental to classical interior-point methods (IPMs),[1] whose computational cost per iteration is typically proportional to n. Starting in the early 1990's, this has prompted a number of researchers to propose, analyze, and test *constraint-reduced* versions of these methods.

The notation and terminology used in the thesis is mostly standard. We denote the *dual slack variable* by $s \in \mathbb{R}^n$, i.e.,

$$s := c - A^T y.$$

A point $y^0 \in \mathbb{R}^m$ is called (dual) feasible if the associated $s^0 = c - A^T y^0$ has only nonnegative components, and (dual) *strictly* feasible if $s^0 > 0$; otherwise, it is called (dual) infeasible. Note that absolute value, comparison and "max" are meant componentwise. The set of all strictly dual-feasible points is called the strictly feasible set of the dual. By e we denote the vector of all ones with dimension by context. We adopt the Matlab-inspired notation $[v_1; v_2; \cdots; v_p]$ to denote a (vertical) concatenation of vectors (or matrices) v_i, $1 \leq i \leq p$. We write a certain subset of $\mathbf{n} := \{1, 2, \cdots, n\}$ by Q and its complement by $\overline{Q} = \mathbf{n} \backslash Q$. Given a vector x, x^i is its

[1] While the simplex method can be a fast iterative method, when $n \gg m$, it takes many more iterations to solve the problems than the interior point methods, and hence is relatively inefficient [64]; therefore, this method is not considered here.

i-th element, x_- is defined by $x_-^i = \min\{x^i, 0\}$ for all $i \in \mathbf{n}$, and x^Q is a subvector of x with only those elements of x that are indexed in set Q. Given a diagonal matrix $X := \text{diag}(x)$, $X^Q := \text{diag}(x^Q)$. Except when specified, the norm $\|\cdot\|$ is arbitrary. The feasible set of the dual (D) (and (Dq)) is denoted by $\mathcal{F}$, i.e.,

$$\mathcal{F} := \{y \in \mathbb{R}^m : A^T y \leq c\},$$

and the strictly feasible set by $\mathcal{F}^0$, i.e.,

$$\mathcal{F}^0 := \{y \in \mathbb{R}^m : A^T y < c\}.$$

The active set at point y (with y not necessarily in $\mathcal{F}$) of (D) (or (Dq)) is denoted by $I(y)$, i.e.,

$$I(y) := \{i : (a^i)^T y = c^i\}.$$

1.2 Constraint reduction

In most interior point methods [40, 66, 69, 50] for LPs and CQPs, the dominant cost per iteration is the formulation of a matrix of the form ADA^T with some diagonal matrix D. The cost is proportional to the number of constraints. In the case of $n \gg m$, i.e., the number of inequality constraints far exceeds that of decision variables, it is typical that only a small percentage of constraints are active at the solution, the others being, in a sense, redundant. Constraint reduction computes search directions based on a judiciously selected subset Q of the constraints that is updated at each iteration, and thus significantly reduces the work while global and local quadratic convergence can be provably retained. This idea is shown by a simple example in Figure 1.1.

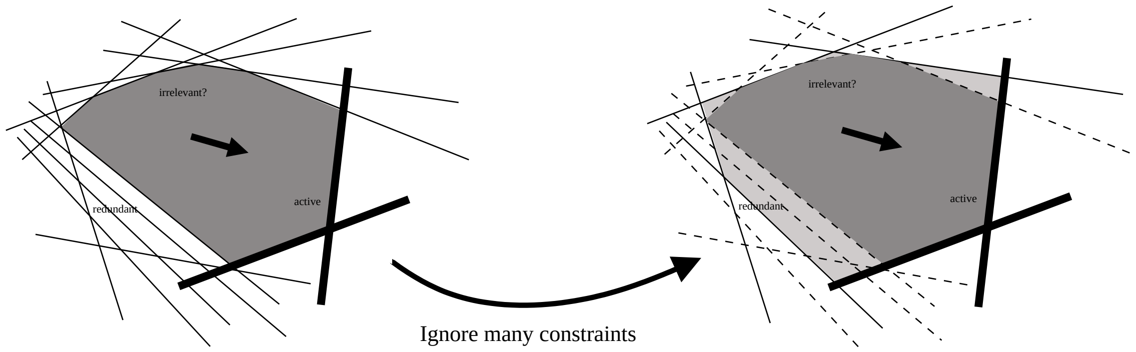


Figure 1.1: An example of constraint reduction: There are $n = 13$ constraints in a dual space of $m = 2$ dimensions. The optimal solution is determined by only two active constraints (in bold lines). Constraint reduction tries to guess a good set Q (with $|Q| = 5$) of constraints and exclude the rest (in dashed lines on the right figure) from computations. Since the computational cost is proportional to the number of constraints, constraint reduction obtains a speedup of $\frac{n}{|Q|} = \frac{13}{5}$. (Drawn by Luke Winternitz)

1.2.1 Constraint-reduced dual interior-point methods for LPs

As early as the 1990's, researchers started to use constraint reduction for LPs that involved only a small subset of constraints in computing search directions for *dual* interior point methods (IPMs).

In paper [15], Dantzig and Ye used constraint reduction within Dikin's algorithm to solve (D). Dikin's algorithm [19, 30] starts with a dual interior point y_k and selects the next iterate y_{k+1} as the solution of the "ellipsoid" problem

$$\max \ b^T y \quad \text{s.t.} \quad \|(y - y_k)AS_k^{-1}\| \leq 1, \tag{1.1}$$

where $S_k := \text{diag}(s_k)$. It can be shown that the solution of (1.1) is

$$y_{k+1} = y_k + (AS_k^{-2}A^T)^{-1}b,$$

and it stays in the interior of the dual feasible set. The process is then repeated with iterate y_{k+1}. Based on the original Dikin's algorithm, constraint reduction in [15] computes the next iterate y_{k+1} by solving the *reduced* "ellipsoid" problem

$$\max\ b^T y \quad \text{s.t.} \quad \|(y - y_k)A^Q(S_k^Q)^{-1}\| \leq 1, \tag{1.2}$$

where $S^Q := \text{diag}(s^Q)$ and A^Q is a submatrix of A with only those columns indexed in Q. Accordingly,

$$y_{k+1} = y_k + (A^Q(S_k^Q)^{-2}(A^Q)^T)^{-1}b.$$

An appropriate set Q is selected such that y_{k+1} is feasible, and with such y_{k+1}, the process is repeated.

At each iterate, an appropriate set Q is "built up" within a "minor cycle" in order to ensure that y_{k+1} is dual-feasible. The minor cycle starts with Q including m "promising" columns (that are deemed to belong to the optimal basis). If the solution of the ellipsoid problem with this Q is infeasible, the cardinality of Q is increased by one. The revised ellipsoid problem, involving $m+1$ columns in the set Q, is re-solved. The columns of A are added to Q like this one by one until the solution of the "built-up" ellipsoid problem is feasible. (When there are more constraints involved, the solution is more likely to be feasible.) Although it is proven in [15] that there always exists a Q with a set of $2m$ columns of A such that the solution of the corresponding ellipsoid problem is feasible, the "minor cycle" does not guarantee that the size of Q as constructed is at most $2m$. This makes choosing Q potentially expensive.

In contrast, the algorithm in [68], due to Ye, starts with Q including all n columns, and *deletes* the columns from Q during the minor cycle. In the end, Q is built-down to an optimal basis. This "build-down" scheme is also used within simplex methods; see [67].

In paper [18], Den Hertog et al. combined "build-up" and "build-down" into one "build up and down" strategy and applied this strategy into a "dual logarithmic-barrier interior-point" method. At each iterate, the build-up process adds those columns to Q, the corresponding constraints of which are violated or almost violated, while the build-down process deletes those columns from Q, the corresponding constraint of which has a large slack. This combined strategy is also used within cutting plane methods for convex optimization [16].

In summary, dual interior-point methods that use constraint reduction have the following features:

- They work on only dual iterates.

- They require a dual strictly-feasible initial point.

- At each iterate, they use a potentially expensive way of selecting the working set Q.

1.2.2 Constraint-reduced primal-dual interior-point methods for LPs

In recent years, a number of researchers have devoted their efforts to constraint reduction for *primal-dual* interior-point methods (PDIPMs) in linear optimization. At each iteration, PDIPMs compute a search direction and perform a linear search. The computational cost of the former is reduced by using constraint reduction (while that of the latter is trivial).

1.2.2.1 Exact directions without constraint reduction

Primal-dual interior-point methods (without constraint reduction) compute a Newton direction from the perturbed Karush-Kuhn-Tucker (KKT) optimality conditions of (P)–(D)

$$A^T y + s = c, \tag{1.3}$$

$$Ax = b, \tag{1.4}$$

$$Xs = \tau e, \tag{1.5}$$

$$x \geq 0, s \geq 0 \tag{1.6}$$

with $X := \operatorname{diag}(x)$, parameter $\tau > 0$ and e the vector of all ones. (This set of conditions is called the "perturbed" KKT conditions because system (1.3)–(1.6) with $\tau = 0$ becomes the set of KKT optimality conditions.) Hence, the Newton direction $(\Delta x_N, \Delta y_N, \Delta s_N)$ solves

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x_N \\ \Delta y_N \\ \Delta s_N \end{bmatrix} = \begin{bmatrix} c - A^T y - s \\ b - Ax \\ \tau e - Xs \end{bmatrix}. \tag{1.7}$$

There are two special cases of the Newton direction. When $\tau = 0$, it is called the *affine-scaling* direction[2] $\Delta p_a = (\Delta x, \Delta y_a, \Delta s)$, i.e.,

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y_a \\ \Delta s \end{bmatrix} = \begin{bmatrix} c - A^T y - s \\ b - Ax \\ -Xs \end{bmatrix}. \tag{1.8}$$

When $\tau \to \infty$, it tends to the *centering* direction $(\Delta x_{cen}, \Delta y_{cen}, \Delta s_{cen})$ which satisfies

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x_{cen} \\ \Delta y_{cen} \\ \Delta s_{cen} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ e \end{bmatrix}. \tag{1.9}$$

Mehrotra's corrector step is a correction to the affine scaling direction Δp_a, and it solves the system

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x_{cor} \\ \Delta y_{cor} \\ \Delta s_{cor} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\Delta X \Delta s \end{bmatrix}, \tag{1.10}$$

[2] Note that for simplicity, we omit the subscript "a" of Δx and Δs, but not that of Δy.

where $\Delta X := \operatorname{diag}(\Delta x)$. Mehrotra's *centering-corrector* direction $\Delta p_c = (\Delta x_c, \Delta y_c, \Delta s_c)$ is a combination of the *centering* and *corrector* direction, and satisfies

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta y_c \\ \Delta s_c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e - \Delta X \Delta s \end{bmatrix} \tag{1.11}$$

with parameter $\sigma \in (0, 1)$ and $\mu = \frac{x^T s}{n}$.

Affine-scaling PDIPMs use only the affine-scaling direction Δp_a as the search direction, and Mehrotra's predictor-corrector PDIPMs use the direction of $\Delta p_a + \Delta p_c$, which is more efficient in practice. (For details, see Wright's book [66], Chapter 10.) Both versions can obtain the search direction by first eliminating $(\Delta x, \Delta s)$ (resp. (Δx_c, Δs_c)) and computing Δy (resp. Δy_c) from a linear system with the coefficient matrix

$$AS^{-1}XA^T = \sum_{i=1}^{n} \frac{x^i}{s^i} a^i (a^i)^T. \tag{1.12}$$

The computational work of obtaining the search direction mainly includes two parts: forming this coefficient matrix and computing its Cholesky factorization. When A is dense, the former takes $m^2 n$ operations and the latter takes $\frac{1}{3}m^3$ operations. In the considered case of $n \gg m$, the former is the dominant cost at each iteration.

1.2.2.2 Inexact directions with constraint reduction

Constraint-reduced PDIPMs use a direction which is obtained from the contribution of a subset Q of the constraints.

Paper [60] is the first one to use constraint reduction within an (*affine-scaling*) PDIPM. (The term *constraint-reduced* PDIPMs was coined there.) The resulting constraint-reduced PDIPM uses a small subset Q of the constraints to compute the affine-scaling search direction. This can be viewed as applying primal-dual interior-point methods to the reduced problem

$$\max\ b^T y \quad \text{s.t. } (A^Q)^T y \leq c^Q, \tag{1.13}$$

and obtaining the affine-scaling search direction $(\Delta x^Q, \Delta y_a, \Delta s^Q)$ by solving the linear system

$$\begin{bmatrix} 0 & (A^Q)^T & I \\ A^Q & 0 & 0 \\ S^Q & 0 & X^Q \end{bmatrix} \begin{bmatrix} \Delta x^Q \\ \Delta y_a \\ \Delta s^Q \end{bmatrix} = \begin{bmatrix} 0 \\ b - A^Q x^Q \\ -X^Q s^Q \end{bmatrix}. \tag{1.14}$$

Eliminating $(\Delta x^Q, \Delta s^Q)$ yields the reduced normal equations

$$A^Q (S^Q)^{-1} X^Q (A^Q)^T \Delta y_a = b. \tag{1.15}$$

It follows that

$$b^T \Delta y_a > 0 \tag{1.16}$$

under a full-row-rank assumption on A^Q and under the positivity of x and s. Inequality (1.16) shows that Δy_a is a direction of ascent for the dual objective $b^T y$, an important property for global convergence.

Paper [65] used constraint reduction schemes within Mehrotra's predictor-corrector (MPC) algorithm. The resulting algorithm rMPC* obtains the reduced affine-scaling direction $(\Delta x^Q, \Delta y_a, \Delta s^Q)$ by solving (1.14), and computes the reduced centering-corrector direction $(\Delta x_c^Q, \Delta y_c, \Delta s_c^Q)$ from

$$\begin{bmatrix} 0 & (A^Q)^T & I \\ A^Q & 0 & 0 \\ S^Q & 0 & X^Q \end{bmatrix} \begin{bmatrix} \Delta x_c^Q \\ \Delta y_c \\ \Delta s_c^Q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma \mu^Q e - \Delta X^Q \Delta s^Q \end{bmatrix} \tag{1.17}$$

where $\mu^Q = \frac{(x^Q)^T s^Q}{|Q|}$. The search direction is then set to

$$(\Delta x_m^Q, \Delta y_m, \Delta s_m^Q) = (\Delta x^Q, \Delta y_a, \Delta s^Q) + \gamma(\Delta x_c^Q, \Delta y_c, \Delta s_c^Q)$$

where $\gamma \in (0,1)$ is carefully chosen to ensure $b^T \Delta y_m > 0$, the same property as in inequality (1.16).

In solving either system (1.14) or (1.17), the dominant cost is to form matrix $A^Q(S^Q)^{-1}X^Q(A^Q)^T$, which takes $|Q|m^2$ operations. This cost is much less than nm^2 of (1.12) when $|Q|$ is small. Thus, the essence of constraint reduction can be summarized as: saving significant cost per iteration by involving much fewer constraints to obtain search directions.

In the constraint-reduced PDIPMs, either the affine-scaling version [60] or the MPC version [65], the rule for choosing set Q is quite simple: at each iterate, include in Q at least $\bar{q}$ constraints that have the smallest slacks such that A^Q has full row rank, where $\bar{q}$ is a known upper bound on the number of active constraints at any solution. Recent work on adaptively choosing Q (see [45] and [32]) satisfies this rule.

In summary, the constraint-reduced PDIPMs of [60] and [65] have the following feastures:

- They work on primal-dual iterates, which is usually faster than working on only dual iterates.

- As constraint-reduced dual IPMs, they also require a strictly dual-feasible initial point.

- Unlike constraint-reduced dual IPMs, they use an inexpensive rule to choose set Q.

1.2.3 Constraint-reduced primal-dual interior-point methods for CQPs

Constraint reduction was extended by Jung et al. in [32] to the general case of convex quadratic optimization problems which have far more inequality constraints than decision variables. As in the linear case, the idea is to reduce the cost of computing the search direction at each iteration.

1.2.3.1 Exact Newton directions without constraint reduction

The KKT conditions for (Pq)–(Dq) are

$$\begin{aligned} A^T y + s &= c, \\ Ax + Hy &= b, \\ Sx &= 0, \\ s \geq 0, x &\geq 0. \end{aligned}$$

By applying Newton's method on the KKT conditions, the primal-dual affine-scaling search direction $(\Delta x, \Delta y, \Delta s)$ solves the system

$$\begin{bmatrix} 0 & A^T & I \\ A & H & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ b - Ax - Hy \\ -Sx \end{bmatrix} \tag{1.18}$$

where we have assumed that y is dual feasible. After block Gaussian elimination, this system is reduced to normal equations

$$(H + AS^{-1}XA^T)\Delta y = b - Hy, \tag{1.19}$$

$$\Delta s = -A^T \Delta y, \tag{1.20}$$

$$\Delta x = -x - S^{-1}X\Delta s. \tag{1.21}$$

As in the linear case, the dominant cost of computing the direction $(\Delta x, \Delta y, \Delta s)$ is to form $AS^{-1}XA^T$ in (1.19), which takes m^2n flops. This cost is significantly reduced by constraint reduction below.

1.2.3.2 Inexact Newton directions with constraint reduction

The constraint-reduced PDIPM [32] for CQPs applies primal-dual interior-point methods to the reduced problem

$$\max \ f(y) \quad \text{s.t.} \quad (A^Q)^T y \leq c^Q, \tag{1.22}$$

and computes the affine-scaling search direction $(\Delta x^Q, \Delta y, \Delta s^Q)$ from solving the system

$$\begin{bmatrix} 0 & (A^Q)^T & I \\ A^Q & H & 0 \\ S^Q & 0 & X^Q \end{bmatrix} \begin{bmatrix} \Delta x^Q \\ \Delta y \\ \Delta s^Q \end{bmatrix} = \begin{bmatrix} 0 \\ b - A^Q x^Q - Hy \\ -S^Q x^Q \end{bmatrix} . \tag{1.23}$$

Analogously to (1.19)–(1.21), this system can be rearranged into the normal equations

$$\begin{aligned} \left(H + A^Q (S^Q)^{-1} X^Q (A^Q)^T\right) \Delta y &= b - Hy & (1.24) \\ \Delta s^Q &= -(A^Q)^T \Delta y & (1.25) \\ \Delta x^Q &= -x^Q - (S^Q)^{-1} X^Q \Delta s^Q. & (1.26) \end{aligned}$$

The dominant cost of computing Δy is to form the matrix in (1.24):

$$A^Q (S^Q)^{-1} X^Q (A^Q)^T = \sum_{i \in Q} \frac{x^i}{s^i} a^i (a^i)^T, \tag{1.27}$$

which takes $|Q|m^2$ operations. These are significantly fewer operations, compared to nm^2 operations in (1.19), when $|Q|$ is as small as a multiple of m. After Δy is solved, $(\Delta x, \Delta s)$ can be trivially retrieved from (1.20)–(1.21).

The constraint reduced PDIPM for CQPs enjoys the same features as that for LPs. In particular, it also requires a strictly dual-feasible initial point.

1.2.4 Motivation: need for a dual-feasible initial point

Though the idea of constraint reduction has been proved successful in previous work for both linear and quadratic optimization, to the best of our knowledge, all existing constraint-reduced IPMs (see section 1.2.1–1.2.3) are obtained by implanting a constraint-reduction scheme into a *dual-feasible* method. Accordingly, they all require a dual feasible initial point. This is an important limitation because such a point is often unavailable in practice and in other cases it may be available but poorly centered, resulting in slow progress of the algorithm.

To obtain a feasible initial point, one can solve the feasibility problem of the dual without considering the objective in a “Phase I” process. With the feasible initial point from “Phase I”, “Phase II” then solves the problem again with consideration of the objective. One common approach for “Phase I” is to solve a relaxed dual problem

$$\max_{y,\,\eta} \; -\eta \quad \text{s.t.} \quad A^T y - \eta e \leq c,$$

where the scalar variable η relaxes the constraints, so that feasible methods can be used. While the objective is to maximize the negative of η, feasible algorithms can be terminated when a negative η is found. Another approach [14] for “Phase I” finds a feasible point that is on the primal-dual central path. Such a feasible point is obtained from solving the relaxed perturbed KKT conditions

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ Xs &= \mu e, \\ x \geq -\eta_1,\; s &\geq -\eta_2, \end{aligned}$$

where in order to allow feasible points for those conditions, the positivity constraints are relaxed by parameters η_1 and η_2. These two parameters are updated at each iterate, and are driven to zero in a finite number of iterations, enforcing the feasibility.

However, solving a "Phase I" problem is equivalent to solving another linear program of the same dimension in many situations. So the idea of using "Phase I" to get a feasible initial point for constraint reduction schemes is impractical.

Another idea is to implant constraint-reduction schemes into *infeasible* PDIPMs. Such attempts were made in [60] with affine-scaling methods and [45] with Mehrotra's Predictor-Corrector methods [39], and also in [45] with an algorithm from [47]. They have some numerical success, but no supporting analysis was provided. Indeed, it appears unlikely that these methods enjoy guaranteed global convergence.

In the present thesis, we show how the need to allow for infeasible initial points can be addressed by making use of exact penalty functions.[3] Exact ℓ_1/ℓ_∞ penalty functions have been used in connection with IPMs in nonlinear programming [3, 61, 8], in particular on problems with complementarity constraints [9, 35, 54], and in at least one instance in linear programming [10]. The dearth of instances of the use of penalty functions in linear and quadratic programming is probably due to the availability of powerful algorithms, both of the simplex variety and of the interior-point variety, that accommodate infeasible initial points in a natural fashion, even guaranteeing polynomial complexity in the case of interior point methods, e.g., [46, 47, 41, 17, 1]. Combining such (possibly polynomially convergent) infeasible

[3] Paper [32] also uses exact penalty functions for infeasible points, but there the penalty parameter must be given by users. Thus, there is no guarantee that optimization problems will be solved.

interior-point methods with constraint-reduction schemes has so far proved elusive, and the use of exact penalty functions is a natural avenue to consider. We have proven global convergence of infeasible constraint-reduced algorithms that use exact penalty functions and have obtained promising numerical results.

The remainder of this thesis is organized as follows: We develop infeasible constraint-reduced algorithms for linear optimization in Chapter 2, and for convex quadratic optimization in Chapter 3. In Chapter 4, these algorithms are applied to the model predictive control applications for rotorcraft altitude control and trajectory following. Future research is discussed in Chapter 5.

Chapter 2

Infeasible Constraint Reduction for Linear Optimization

This chapter focuses mainly on developing an infeasible constraint-reduced IPM framework for linear optimization. The organization is as follows: Based on feasible constraint-reduced PDIPMs, our proposed dual-feasible constraint-reduced IPM (rIPM) framework is layed out and analyzed in section 2.1. In section 2.2, the framework is extended, by incorporating an exact penalty function, to allow for infeasible initial points in the case of constraint-reduced *primal-dual* interior point. The resulting framework, IrPDIP, is specialized to the case of Algorithm rMPC* of [65] (a constraint-reduced variant of Mehrotra's Predictor Corrector algorithm), yielding algorithm IrMPC. IrMPC is analyzed in section 2.3, and further studied in section 2.4 under weaker feasibility assumptions. Numerical results are reported in section 2.5.

2.1 A framework for dual-feasible constraint-reduced IPMs

Many interior-point methods for the solution of (P)–(D), including the current "champion", Mehrotra's Predictor Corrector [39], and its many variants [71, 53, 66], make use of an *affine scaling* direction Δy_a. That is the solution of

$$ADA^T \Delta y_a = b \tag{2.1}$$

for some diagonal positive-definite matrix D, usually updated from iteration to iteration. For such methods, when $n \gg m$, the main computational cost at each iteration resides in forming the matrix

$$ADA^T = \sum_{i=1}^{n} d^i a^i (a^i)^T \tag{2.2}$$

where d^i is the ith diagonal entry of D and a^i the ith column of A. Forming ADA^T takes up roughly nm^2 multiplications and as many additions. If the sum in the right-hand side of (2.2) is reduced by dropping all terms except those associated with a certain small working index set Q, the cost of forming it reduces from nm^2 to roughly $|Q|m^2$. Conceivably, the cardinality $|Q|$ of Q could be as small as m in nondegenerate situations, leading to a potential computational speedup factor of n/m. Ideas along these lines are explored in [15, 18, 62, 60, 65] where schemes are proposed that enjoy strong theoretical properties and work well in practice. (Interestingly, in many cases, it has been observed that using a small working set does not significantly increase the total number of iterations required to solve the problem, and sometimes even reduces it.) Several of these methods [15, 60, 65] fit within the following general iteration framework.

Iteration rIPM

Parameters: $\theta \in (0,1)$ and $\tau > 0$.

Data: $y \in \mathbb{R}^m$ such that $s := c - A^T y > 0$; $Q \subseteq \mathbf{n}$ such that A^Q is full row rank; $D \in \mathbb{R}^{|Q|\times|Q|}$, diagonal and positive definite.

Step 1: Computation of the dual search direction Δy

(i) Let Δy_a solve

$$A^Q D(A^Q)^T \Delta y_a = b. \tag{2.3}$$

(ii) Select Δy to satisfy

$$b^T \Delta y \geq \theta b^T \Delta y_a, \quad \|\Delta y\| \leq \tau \|\Delta y_a\|. \tag{2.4}$$

Step 2: Updates

(i) Update the dual variables by choosing a stepsize $t \in (0, 1]$ such that

$$s_+ := c - A^T y_+ > 0$$

where

$$y_+ := y + t\Delta y. \tag{2.5}$$

(ii) Pick Q_+ such that A^{Q_+} has full row rank.

(iii) Select $D_+ \in \mathbb{R}^{|Q_+| \times |Q_+|}$, diagonal and positive definite.

Since A^Q has full row rank, the linear system (2.3) has a unique solution. Hence Iteration rIPM is well defined and, since $x_+ > 0$ and $s_+ > 0$, it can be repeated indefinitely to generate infinite sequences. We attach subscript k to denote the kth iterate. Since for all k, $x_k > 0$ and $s_k > 0$, it also follows from (2.3) that

$$b^T \Delta y_{a,k} > 0, \tag{2.6}$$

and further from (2.4) and (2.5) that the sequence $\{b^T y_k\}$ is increasing.

An important property of Iteration rIPM, established in Proposition 2.1 below, is that if the dual-feasible sequence $\{y_k\}$ remains bounded, then it must converge, and if it is unbounded, then $b^T y_k \to +\infty$. The proof makes use of the following lemma, a direct consequence of results in [52] (see also [51]).

Lemma 2.1. *Let G be a full row rank matrix and b be in the range of G. Then, (i) there exists $\phi > 0$ (depending only on G and b) such that, given any positive-definite diagonal matrix D, the solution Δy to*

$$GDG^T \Delta y = b,$$

satisfies

$$\|\Delta y\| \leq \phi b^T \Delta y;$$

and (ii) if a sequence $\{y_k\}$ is such that $\{b^T y_k\}$ is bounded and, for some $\omega > 0$, satisfies

$$\|y_{k+1} - y_k\| \leq \omega b^T (y_{k+1} - y_k) \quad \forall k, \tag{2.7}$$

then $\{y_k\}$ converges.

Proof. The first claim immediately follows from Theorem 5 in [52], noting (as in [51], section 4) that, for some $\zeta > 0$, $\zeta \Delta y$ solves

$$\max\{\ b^T u \mid u^T GDG^T u \leq 1\}.$$

(See also Theorem 7 in [51].) The second claim is proved using the central argument of the proof of Theorem 9 in [52]:

$$\sum_{k=0}^{N-1} \|y_{k+1} - y_k\| \leq \omega \sum_{k=0}^{N-1} b^T (y_{k+1} - y_k) = \omega b^T (y_N - y_0) \leq 2\omega v \quad \forall N > 0,$$

where v is a bound on $\{|b^T y_k|\}$, implying that $\{y_k\}$ is Cauchy, and thus converges. (See also Theorem 9 in [51].) □

Proposition 2.1. *Suppose* (D) *is strictly feasible. Then, if* $\{y_k\}$ *generated by Iteration rIPM is bounded, then* $y_k \to y_*$ *for some* $y_* \in \mathcal{F}$*, and if it is not, then* $b^T y_k \to \infty$*.*

Proof. We first show that $\{y_k\}$ satisfies (2.7) for some $\omega > 0$. In view of (2.5), it suffices to show that, for some $\omega > 0$,[1]

$$\|\Delta y_k\| \leq \omega b^T \Delta y_k \quad \forall k. \tag{2.8}$$

Now, since $\Delta y_{a,k}$ solves (2.3) and since A^{Q_k} has full row rank, and $Q_k \subseteq \mathbf{n}$, a finite set, it follows from Lemma 2.1 (i) that, for some $\phi > 0$,

$$\|\Delta y_{a,k}\| \leq \phi b^T \Delta y_{a,k} \quad \forall k.$$

With this in hand, we obtain, for all k, using (2.4),

$$\|\Delta y_k\| \leq \tau \|\Delta y_{a,k}\| \leq \tau \phi b^T \Delta y_{a,k} \leq \tau \frac{\phi}{\theta} b^T \Delta y_k \quad \forall k,$$

so (2.8) holds with $\omega := \tau \frac{\phi}{\theta}$. Hence (2.7) holds (with the same ω).

To complete the proof, first suppose that $\{y_k\}$ is bounded. Then so is $\{b^T y_k\}$ and, in view of Lemma 2.1 (ii) and of the fact that $\{y_k\}$ is feasible, we have $y_k \to y_*$, for some $y_* \in \mathcal{F}$. On the other hand, if $\{y_k\}$ is unbounded, then $\{b^T y_k\}$ is also unbounded (since, in view of Lemma 2.1 (ii), having $\{b^T y_k\}$ bounded together with (2.7) would lead to the contradiction that the unbounded sequence $\{y_k\}$ converges). Since $\{b^T y_k\}$ is nondecreasing, the claim follows. □

[1] Inequality (2.8) is an angle condition: existence of $\omega > 0$ means that the angle between b and Δy is bounded away from 90 °. This condition, which is weaker than (2.4), is sufficient for Proposition 2.1 to hold.

The "build-up" algorithm in [15], algorithm rPDAS in [60], and rMPC* in [65] all fit within the rIPM framework. In [15], D is $\mathrm{diag}(s^Q)^{-2}$, and in rPDAS and rMPC*, D is $\mathrm{diag}\left((x^i/s^i)^{i\in Q}\right)$. In [15] and rPDAS, Δy is Δy_a, and in rMPC*, Δy satisfies (2.4) with $\tau = 1+\psi$, where $\psi > 0$ is a parameter of rMPC*. Hence, Proposition 2.1 provides a simpler proof for the convergence of the dual sequence $\{y_k\}$ of [15] than that used in proving Theorem 3 of that paper; it strengthens the convergence result for rPDAS (Theorem 12 in [60]) by establishing convergence of the dual sequence to a single optimal point; and it is used in [65] (provisionally accepted for publication). Proposition 2.1 is also used in the next section, in the analysis of the expanded framework IrPDIP (see Proposition 2.2).

2.2 A framework for infeasible constraint-reduced PDIPs

2.2.1 Basic ideas and algorithm statement

As mentioned in chapter 1, previously proposed constraint-reduced interior-point methods ([15], [68], [18], [60] and [65]) for LPs require a strictly dual-feasible initial point. Here, we show how the limitation can be circumvented by means of an ℓ_1 or ℓ_∞ exact penalty function. Specifically, in the ℓ_1 case, we consider relaxing (D) with

$$\left.\begin{array}{ll} \max & b^Ty - \rho e^T z \\ \text{s.t.} & A^Ty - z \le c,\ z \ge 0, \end{array}\right\} \qquad (\mathrm{D}_\rho)$$

where $z \in \mathbb{R}^n$, maximization is with respect to (y, z), and $\rho > 0$ is a scalar penalty parameter, with "primal"

$$\left.\begin{array}{rc} \min & c^T x \\ \text{s.t.} & Ax = b,\, x + u = \rho e, \\ & x \geq 0,\, u \geq 0. \end{array}\right\} \qquad (\mathrm{P}_\rho)$$

Strictly feasible initial points for (D_ρ) are trivially available, and any of the algorithms just mentioned can be used to solve this primal-dual pair.

It is well known (e.g. Theorem 40 in [23], Theorem 1 in [10]) that the ℓ_1 penalty function is "exact", i.e., there exists a threshold value ρ_* such that for any $\rho > \rho_*$, (y_*^ρ, z_*^ρ) solves (D_ρ), then y_*^ρ solves (D) and $z_*^\rho = 0$. But such a ρ_* is not known *a priori*. We propose a scheme inspired by that used in [61] (in a nonlinear optimization context) for iteratively identifying an appropriate value for ρ. A key difference is that, unlike that of [61] (see Lemma 4.1 and Proposition 4.2 in that paper), our scheme requires no *a priori* assumption on the boundedness of the sequences of iterates (y_k in our case, x_k in [61]). As seen from the toy example

$$\max y \quad \text{s.t.} \quad y \leq 0,\, 2y \leq 2, \tag{2.9}$$

when too small a value of ρ is used, such boundedness is not guaranteed. Indeed, the penalized problem associated to (2.9) is

$$\max y - \rho z^1 - \rho z^2 \quad \text{s.t.} \quad y - z^1 \leq 0,\, 2y - z^2 \leq 2,\, z^1 \geq 0,\, z^2 \geq 0,$$

or equivalently,

$$\min_y \quad \{-y + \rho \max\{0, y\} + 2\rho \max\{0, y - 1\}\}. \tag{2.10}$$

As seen from Figure 2.1, when $\rho < \frac{1}{3}$, problem (2.10) is unbounded, even though problem (2.9) itself is bounded.

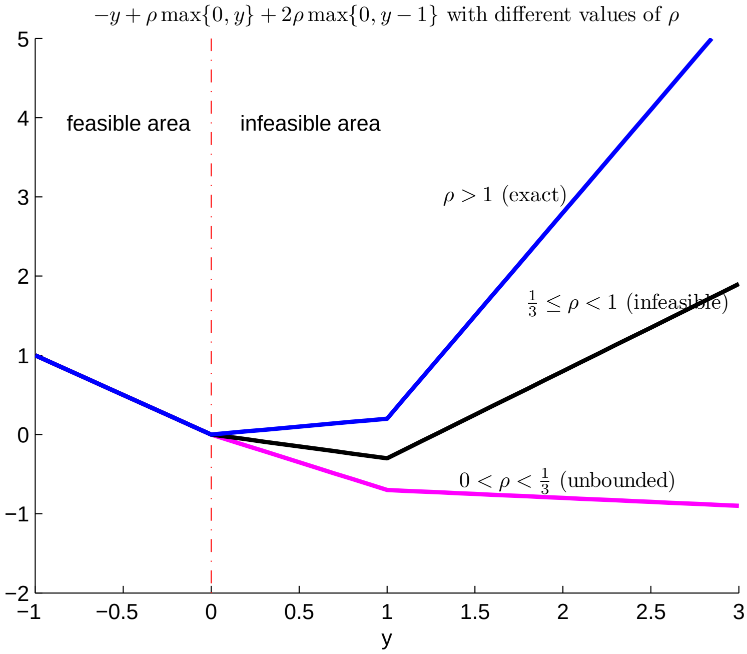


Figure 2.1: The objective function of problem (2.10) with different penalty parameter values. When $\rho < \frac{1}{3}$, problem (2.10) is unbounded. When $\rho \in [\frac{1}{3}, 1)$, it is bounded but the minimizer $y_*^\rho = 1$ is infeasible for (2.9). When $\rho > \rho_* = 1$, $y_*^\rho = 0$ solves (2.9) as desired.

In the ℓ_1 version of our proposed scheme, the penalty parameter ρ is increased if either

$$\|z_+\| > \gamma_1 \frac{\|z_0\|}{\rho_0} \rho \tag{2.11}$$

or

$$\|[\Delta y_a; \Delta z_a]\| \leq \frac{\gamma_2}{\rho}, \text{ and (ii) } \tilde{x}^Q \geq -\gamma_3 e, \text{ and (iii) } \tilde{u}^Q \not\geq \gamma_4 e \tag{2.12}$$

is satisfied, where $\gamma_i > 0$, $i = 1, 2, 3, 4$ are parameters, z_+ is the just computed next value of z, and $\tilde{x}^Q$ and $\tilde{u}^Q$ (defined in (2.17) and (2.18) below) are the most recently computed Karush-Kuhn-Tucker (KKT) multipliers for constraints $(A^Q)^T y - z^Q \leq c^Q$ and $z^Q \geq 0$ respectively, and where the factor $\|z_0\|/\rho_0$ has been introduced for scaling purposes. Note that these conditions involve both the dual and primal sets of variables. As we will see though, the resulting algorithm framework IrPDIP is proved to behave adequately under rather mild restrictions on how primal variables are updated.

Condition (2.11) is new. It ensures boundedness of $\{z_k\}$ (which is necessary for $\{y_k\}$ to be bounded), whenever $\{\rho_k\}$ is bounded; with such a condition, the situation just described where $\{z_k\}$ is unbounded due to $\{\rho_k\}$ being too small cannot occur. Condition (2.12) is adapted from [61] (see Step 1 (ii) in Algorithm A of [61], as well as the discussion preceding the algorithm). Translated to the present context, the intuition is that ρ should be increased if a stationary point for (D_ρ) is approached ($\|[\Delta y_a; \Delta z_a]\|$ small) at which not all components of the constraints $z \geq 0$ are binding (not all components of $\tilde{u}^Q$ are significantly positive), and no component of $\tilde{x}^Q$ or $\tilde{u}^Q$ takes a large negative value, which would indicate that the stationary point is not a dual maximizer. Two adaptations were in order: First, closeness to a stationary point for (D_ρ) is rather related to the size of $\rho\|[\Delta y_a; \Delta z_a]\|$; in [61], this makes no difference because the sequence of multiplier estimates ((x, u) in the present context) is bounded by construction, even when ρ grows without bound; second, the lower bound on $\tilde{u}^Q$ turns out not to be needed in the present context due to the special structure of the $z \geq 0$ constraints (compared to the general $c(x) \geq 0$ in [61]).

Iteration IrPDIP, stated next, amounts to rIPM applied to (D$_\rho$), rather than (D), with ρ updated as just discussed (Step 2 (iv)), as well as a specific D matrix (primal-dual affine scaling: Step 1 (i)) and rather general bounds on how the primal variables x and u should be updated (Step 2 (ii)).

Iteration IrPDIP

Parameters: $\theta \in (0,1), \tau > 0, \alpha > 0$, $\chi > 0$, $\sigma > 1$, $\gamma_i > 0$, for $i = 1, 2, 3, 4$.

Data: $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ such that $z > \max\{0, A^T y - c\}$; $x \in \mathbb{R}^n$, $u \in \mathbb{R}^n$ and $\rho \in \mathbb{R}$ such that $x > 0$, $u > 0$ and $\rho > 0$; $Q \subseteq \mathbf{n}$ such that A^Q has full row rank; $s := c - A^T y + z$.

Step 1: Computation of search direction:

(i) Let $\left(\Delta x^Q, \Delta u, \Delta y_a, \Delta z_a, \Delta s^Q\right)$ be the primal-dual affine-scaling direction (see (1.14)) for problem[2]

$$\left.\begin{array}{ll} \max & b^T y - \rho e^T z \\ \text{s.t.} & (A^Q)^T y - z^Q \leq c^Q,\ z \geq 0. \end{array}\right\} \qquad (\mathrm{D}_\rho^Q)$$

(ii) Select $(\Delta y, \Delta z)$ to satisfy

$$b^T \Delta y - \rho e^T \Delta z \geq \theta(b^T \Delta y_a - \rho e^T \Delta z_a), \quad \|[\Delta y; \Delta z]\| \leq \tau \|[\Delta y_a; \Delta z_a]\|. \quad (2.13)$$

Step 2. Updates.

[2] Constraints $z \geq 0$ are not "constraint-reduced" in (D$_\rho^Q$). The reason is that they are known to be active at the solution, and furthermore their contribution to the normal matrix (2.2) is computed at no cost.

(i) Update the dual variables by choosing a stepsize $t \in (0,1]$ such that

$$s_+ := c - A^T y_+ + z_+ > 0, \quad z_+ > 0 \tag{2.14}$$

where

$$y_+ := y + t\Delta y, \quad z_+ := z + t\Delta z. \tag{2.15}$$

(ii) Select $[x_+; u_+] > 0$ to satisfy

$$\|[x_+; u_+]\| \leq \max\left\{\|[x; u]\|, \alpha\|[\tilde{x}^Q; \tilde{u}]\|, \chi\right\} \tag{2.16}$$

where

$$\begin{aligned} \tilde{x}^Q &:= x^Q + \Delta x^Q, & (2.17) \\ \tilde{u} &:= u + \Delta u. & (2.18) \end{aligned}$$

(iii) Pick $Q_+ \subseteq \mathbf{n}$ such that A^{Q_+} has full row rank.

(iv) Check the two cases (2.11) and (2.12). If either case is satisfied, set

$$\rho_+ := \sigma\rho;$$

otherwise $\rho_+ := \rho$.

Note that to guarantee that direction $\left(\Delta x^Q, \Delta u, \Delta y_a, \Delta z_a, \Delta s^Q\right)$ (see (2.19) below) is well defined, it is sufficient that A^Q have full row rank (see Step 2 (iii) in Iteration IrPDIP). Indeed, this makes $[A^Q \;\, 0; -E^Q \;\, -I]$ full row rank, so that the solution $(\Delta y_a, \Delta z_a)$ to (2.20) below is well defined.

2.2.2 Computational issues

The main computation in Iteration IrPDIP is the calculation of the affine-scaling direction in Step 1 (i). The primal-dual affine-scaling direction $\left(\Delta x^Q, \Delta u, \Delta y_a, \Delta z_a, \Delta s^Q\right)$ for (D_ρ^Q) is obtained by solving system (derived from (1.14))

$$\begin{bmatrix} 0 & 0 & 0 & (A^Q)^T & -I & 0 & I \\ A^Q & 0 & 0 & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 \\ S^Q & 0 & 0 & 0 & 0 & 0 & X^Q \\ 0 & Z^Q & 0 & 0 & U^Q & 0 & 0 \\ 0 & 0 & Z^{\overline{Q}} & 0 & 0 & U^{\overline{Q}} & 0 \end{bmatrix} \begin{bmatrix} \Delta x^Q \\ \Delta u^Q \\ \Delta u^{\overline{Q}} \\ \Delta y_a \\ \Delta z_a^Q \\ \Delta z_a^{\overline{Q}} \\ \Delta s^Q \end{bmatrix} = \begin{bmatrix} 0 \\ b - A^Q x^Q \\ \rho e - x^Q - u^Q \\ \rho e - u^{\overline{Q}} \\ -X^Q s^Q \\ -Z^Q u^Q \\ -Z^{\overline{Q}} u^{\overline{Q}} \end{bmatrix} \tag{2.19}$$

where $Z := \operatorname{diag}(z)$ and $U := \operatorname{diag}(u)$. Eliminating $(\Delta x^Q, \Delta u)$ and Δs^Q in system (2.19), we obtain the reduced normal system

$$\begin{bmatrix} A^Q & 0 \\ -E^Q & -I \end{bmatrix} \begin{bmatrix} X^Q & 0 \\ 0 & U \end{bmatrix} \begin{bmatrix} S^Q & 0 \\ 0 & Z \end{bmatrix}^{-1} \begin{bmatrix} A^Q & 0 \\ -E^Q & -I \end{bmatrix}^T \begin{bmatrix} \Delta y_a \\ \Delta z_a \end{bmatrix} = \begin{bmatrix} b \\ -\rho e \end{bmatrix} \tag{2.20}$$

$$\Delta s^Q = -(A^Q)^T \Delta y_a + \Delta z_a^Q, \tag{2.21}$$

$$\begin{bmatrix} \Delta x^Q \\ \Delta u \end{bmatrix} = - \begin{bmatrix} x^Q \\ u \end{bmatrix} - \begin{bmatrix} X^Q & 0 \\ 0 & U \end{bmatrix} \begin{bmatrix} S^Q & 0 \\ 0 & Z \end{bmatrix}^{-1} \begin{bmatrix} \Delta s^Q \\ \Delta z_a \end{bmatrix} \tag{2.22}$$

where E^Q is a submatrix of the $n \times n$ identity matrix consisting of only those columns that are indexed in set Q. Further eliminating Δz_a, we can reduce (2.20) to

$$\begin{aligned} A^Q D^{(Q)}(A^Q)^T \Delta y_a &= b - A^Q X^Q (S^Q)^{-1}(E^Q)^T (D_2^{(Q)})^{-1} \rho e, \qquad (2.23) \\ D_2^{(Q)} \Delta z_a &= -\rho e + E^Q X^Q (S^Q)^{-1} (A^Q)^T \Delta y_a \end{aligned}$$

where diagonal positive definite matrices $D^{(Q)}$ and $D_2^{(Q)}$ are given as

$$\begin{aligned} D^{(Q)} &:= X^Q(S^Q)^{-1} - X^Q(S^Q)^{-1}(E^Q)^T(D_2^{(Q)})^{-1}E^Q X^Q (S^Q)^{-1}, \\ D_2^{(Q)} &:= UZ^{-1} + E^Q X^Q (S^Q)^{-1}(E^Q)^T. \qquad (2.24) \end{aligned}$$

(Since Q is selected such that A^Q is full row rank, (2.23) yields a unique Δy_a.) By using the Sherman-Morrison-Woodbury matrix identity, $D^{(Q)}$ can be simplified to

$$D^{(Q)} = \left(S^Q(X^Q)^{-1} + (E^Q)^T U^{-1} Z E^Q\right)^{-1} = \left(S^Q(X^Q)^{-1} + Z^Q(U^Q)^{-1}\right)^{-1}. \quad (2.25)$$

The dominant cost in computing $\left(\Delta x^Q, \Delta u, \Delta y_a, \Delta z_a, \Delta s^Q\right)$ is to solve (2.20), with cost dominated by forming the coefficient matrix $A^Q D^{(Q)}(A^Q)^T$ of (2.23). When A is dense, this operation takes $|Q|m^2$ multiplications. In the case of $n \gg m$, this can be much less than nm^2. In particular, the same speedup factor can be obtained as in the case of the dual-feasible rIPM.

2.2.3 Convergence analysis

Iteration IrPDIP can be repeated indefinitely, generating an infinite sequence of iterates with the dual sequence $\{(y_k, z_k, s_k)\}$ feasible for problem (D$_\rho$). In section 2.1, the sole assumption on (P)–(D) was that A has full row rank. Below, we further selectively assume (strict) feasibility of (P)–(D).

In this section, we show that under mild assumptions the penalty parameter ρ in Iteration IrPDIP will be increased no more than a finite number of times. First, as a direct application of (2.6) transposed to problem (D_ρ), and of (2.13), $(\Delta y, \Delta z)$ is an ascent direction for (D_ρ). We state this as a lemma.

Lemma 2.2. *Step 1(i) of IrPDIP is well defined and* $b^T\Delta y - \rho e^T \Delta z > 0$*.*

In view of (2.11), a necessary condition for ρ_k to remain bounded is that $\{z_k\}$ be bounded. The latter does hold, as we show next. A direct consequence is boundedness of $\{b^T y_k\}$ from above.

Lemma 2.3. *Suppose* (P) *is feasible, then sequence* $\{z_k\}$ *is bounded, and* $\{b^T y_k\}$ *is bounded from above.*

Proof. We first show that $\{z_k\}$ is bounded. If ρ_k is increased finitely many times to a finite value, say ρ_∞, then condition (2.11) must fail for k large enough, which implies that $\|z_k\| \leq \gamma_1 \frac{\|z_0\|}{\rho_0}\rho_\infty$ for k large enough, proving the claim. It remains to prove that $\{z_k\}$ is bounded when ρ_k is increased infinitely many times, i.e., when $\rho_k \to \infty$ as $k \to \infty$.

By assumption, (P) has a feasible point, say x^0, i.e.,

$$Ax^0 = b,\, x^0 \geq 0. \tag{2.26}$$

Since $\rho_k \to \infty$ as $k \to \infty$, there exists k_0 such that

$$\rho_k > \|x^0\|_\infty,\, \forall k \geq k_0. \tag{2.27}$$

Since $(y_k,\, z_k)$ is feasible for (D_ρ) for all k, we have

$$\begin{aligned} A^T y_k &\le z_k + c \quad \forall k, & (2.28)\\ z_k &\ge 0 \quad \forall k. & (2.29) \end{aligned}$$

Left-multiplying by $(x^0)^T \ge 0$ on both sides of (2.28) and using (2.26) yields

$$b^T y_k \le (x^0)^T z_k + c^T x^0 \quad \forall k. \tag{2.30}$$

Adding $\rho_k e^T z_k$ to both sides of (2.30), we get

$$(\rho_k e - x^0)^T z_k \le \pi_k + \rho_k e^T z_k \quad \forall k, \tag{2.31}$$

where we have defined

$$\pi_k := c^T x^0 - b^T y_k. \tag{2.32}$$

In view of (2.27) and (2.29), we conclude that z_k satisfies

$$0 \le z_k^i \le \frac{\pi_k + \rho_k e^T z_k}{\rho_k - (x^0)^i} \le \frac{\pi_k + \rho_k e^T z_k}{\rho_k - \|x^0\|_\infty} =: \nu_k,\ \forall i,\ \forall k \ge k_0,$$

so that

$$\|z_k\|_\infty \le \nu_k. \tag{2.33}$$

Hence, in order to show that $\{z_k\}$ is bounded, it suffices to prove that $\{\nu_k\}$ is bounded. We show next that $\nu_{k+1} \le \nu_k$, $\forall k \ge k_0$. Since in view of (2.27), ν_k is positive for all k, this proves the boundness of $\{\nu_k\}$.

To this end, first note that for each k, Lemma 2.2 implies that

$$b^T y_{k+1} - \rho_k e^T z_{k+1} = b^T y_k - \rho_k e^T z_k + t_k(b^T \Delta y_k - \rho_k e^T \Delta z_k) \ge b^T y_k - \rho_k e^T z_k,$$

where we have used (2.15). Together with (2.27), this implies that

$$\nu_k = \frac{\pi_k + \rho_k e^T z_k}{\rho_k - \|x^0\|_\infty} \geq \frac{\pi_{k+1} + \rho_k e^T z_{k+1}}{\rho_k - \|x^0\|_\infty},\ \forall k \geq k_0. \tag{2.34}$$

Since $\rho_{k+1} \geq \rho_k$ and since

$$\nu_{k+1} = \frac{\pi_{k+1} + \rho_{k+1} e^T z_{k+1}}{\rho_{k+1} - \|x^0\|_\infty}, \tag{2.35}$$

in order to conclude that $v_{k+1} \leq v_k$ for $k \geq k_0$, it is sufficient to verify that the function g given by

$$g(\rho) := \frac{\pi_{k+1} + \rho e^T z_{k+1}}{\rho - \|x^0\|_\infty}$$

has a nonpositive derivative $g'(\rho)$ for all ρ satisfying (2.27). Since

$$\begin{aligned}
\pi_{k+1} + \|x^0\|_\infty e^T z_{k+1} &= c^T x^0 - b^T y_{k+1} + \|x^0\|_\infty e^T z_{k+1} && \text{(using (2.32))} \\
&= (x^0)^T c - (x^0)^T A^T y_{k+1} + \|x^0\|_\infty e^T z_{k+1} && \text{(using (2.26))} \\
&\geq -(x^0)^T z_{k+1} + \|x^0\|_\infty e^T z_{k+1} && \text{(using (2.28) and (2.26))} \\
&\geq 0, && \text{(using (2.29))}
\end{aligned}$$

it is readily checked using (2.27) that

$$g'(\rho) = -\frac{\pi_{k+1} + \|x^0\|_\infty e^T z_{k+1}}{(\rho - \|x^0\|_\infty)^2} \leq 0.$$

Hence $\{z_k\}$ is bounded, proving the first claim. It follows immediately from (2.30) that $\{b^T y_k\}$ is bounded above, proving the second claim. □

With boundedness of $\{z_k\}$ in hand, the possibility that $\{\rho_k\}$ be unbounded will be ruled out by a contradiction argument. But first, we prove that the primal variables are bounded by a linear function of ρ_k.

Lemma 2.4. *There exists a constant $C > 0$ such that*

$$\left\| [\tilde{x}_k^{Q_k}; \tilde{u}_k; x_k; u_k] \right\| \;\; \leq \;\; C\rho_k. \tag{2.36}$$

Proof. By the triangle inequality, it suffices to show that there exist C_1 and C_2 such that

$$\|[\tilde{x}_k^{Q_k}; \tilde{u}_k]\| \leq C_1\rho_k, \;\; \|[x_k; u_k]\| \leq C_2\rho_k. \tag{2.37}$$

Substituting (2.21) into (2.22), and using (2.17) and (2.18), we have

$$\begin{bmatrix} \tilde{x}_k^{Q_k} \\ \\ \tilde{u}_k \end{bmatrix} = \begin{bmatrix} X_k^{Q_k}\left(S_k^{Q_k}\right)^{-1} & 0 \\ \\ 0 & U_k(Z_k)^{-1} \end{bmatrix} \begin{bmatrix} A^{Q_k} & 0 \\ \\ -E^{Q_k} & -I \end{bmatrix}^T \begin{bmatrix} \Delta y_{a,k} \\ \\ \Delta z_{a,k} \end{bmatrix}. \tag{2.38}$$

Solving (2.20) for $[\Delta y_{a,k}; \Delta z_{a,k}]$ and substituting it into (2.38) yields

$$\begin{bmatrix} \tilde{x}_k^{Q_k} \\ \\ \tilde{u}_k \end{bmatrix} = H_k \begin{bmatrix} b \\ \\ -\rho_k e \end{bmatrix} \tag{2.39}$$

with

$$H_k := \begin{bmatrix} X_k^{Q_k}\left(S_k^{Q_k}\right)^{-1} & 0 \\ 0 & U_k(Z_k)^{-1} \end{bmatrix} \begin{bmatrix} A^{Q_k} & 0 \\ -E^{Q_k} & -I \end{bmatrix}^T \left(\begin{bmatrix} A^{Q_k} & 0 \\ -E^{Q_k} & -I \end{bmatrix} \begin{bmatrix} X_k^{Q_k}\left(S_k^{Q_k}\right)^{-1} & 0 \\ 0 & U_k(Z_k)^{-1} \end{bmatrix} \begin{bmatrix} A^{Q_k} & 0 \\ -E^{Q_k} & -I \end{bmatrix}^T \right)^{-1}.$$

Because diagonal matrices $X_k^{Q_k}, S_k^{Q_k}, U_k$ and Z_k are positive definite for all k, it follows from Theorem 1 in [55] that the sequence $\{H_k\}$ is bounded. Therefore, (2.39) implies that there exist $C' > 0$ and $C_1 > 0$, both independent of k, such that

$$\left\| \begin{bmatrix} \tilde{x}_k^{Q_k} \\ \\ \tilde{u}_k \end{bmatrix} \right\| \leq C' \left\| \begin{bmatrix} b \\ \\ -\rho_k e \end{bmatrix} \right\| \leq C_1\rho_k, \, \forall k, \tag{2.40}$$

proving the first inequality in (2.37). Now, without loss of generality, suppose

$$C_1 \geq \frac{\max(\|[x_0; u_0]\|, \chi)}{\alpha\rho_0},$$

where α is a parameter in Iteration IrPDIP, and let $C_2 \geq \alpha C_1$. That $\|[x_k; u_k]\| \leq C_2\rho_k$ follows by induction. Indeed, clearly, it holds at $k = 0$, and if $\|[x_k; u_k]\| \leq C_2\rho_k$ at some iterate k, then since $\{\rho_k\}$ is nondecreasing, it follows from (2.16) and (2.40) that

$$\|[x_{k+1}; u_{k+1}]\| \leq \max\{C_2\rho_k, \alpha C_1\rho_k, \chi\} \leq C_2 \max\{\rho_k, \rho_0\} \leq C_2\rho_{k+1}.$$

□

If (P) is feasible, then Lemma 2.3 rules out the possibility that condition (2.11) is satisfied on an infinite sequence. Therefore, if, as we will assume by contradiction, ρ_k goes to infinity as k goes to infinity, conditions (2.12) must be satisfied on an infinite subsequence. The next lemma exploits this. From that lemma on, K_ρ denotes the index sequence on which ρ_k is updated, i.e.,

$$K_\rho = \{k : \rho_{k+1} > \rho_k\}. \tag{2.41}$$

Lemma 2.5. *Suppose $\rho_k \to \infty$ and* (P) *is feasible, then $\{Z_k\tilde{u}_k\}$ and $\{S_k^{Q_k}\tilde{x}_k^{Q_k}\}$ are bounded on K_ρ. If in addition* (D) *is feasible, then $z_k \to 0$ as $k \to \infty$, $k \in K_\rho$, and if furthermore* (P) *is strictly feasible, then $\{y_k\}$ is bounded on K_ρ.*

Proof. Since ρ_k goes to infinity on K_ρ and (P) is feasible, Lemma 2.3 implies that conditions (2.11) is eventually violated, so condition (2.12) must be satisfied for $k \in K_\rho$ large enough. In particular, there exists k_0 such that for all $k \geq k_0$, $k \in K_\rho$,

$$\|[\Delta y_{a,k}; \Delta z_{a,k}]\| \leq \frac{\gamma_2}{\rho_k}, \tag{2.42}$$

and

$$\tilde{x}_k^{Q_k} \geq -\gamma_3 e. \tag{2.43}$$

Since (first block row of (2.19))

$$\Delta s_k^{Q_k} = -(A^{Q_k})^T \Delta y_{a,k} + \Delta z_{a,k}^{Q_k},$$

it follows from (2.42) that there exists $\delta > 0$ such that

$$\|\Delta s_k^{Q_k}\| \leq \frac{\delta}{\rho_k},\ k \geq k_0, k \in K_\rho. \tag{2.44}$$

Using Lemma 2.4, equations (2.42) and (2.44), and the last three block rows of (2.19), we get

$$\|Z_k \tilde{u}_k\| = \|U_k \Delta z_{a,k}\| \leq C\rho_k \cdot \frac{\gamma_2}{\rho_k} = C\gamma_2, \quad k \geq k_0,\ k \in K_\rho, \tag{2.45}$$

and

$$\left\|S_k^{Q_k} \tilde{x}_k^{Q_k}\right\| = \left\|X_k^{Q_k} \Delta s_k^{Q_k}\right\| \leq C\rho_k \cdot \frac{\delta}{\rho_k} = C\delta, \quad k \geq k_0,\ k \in K_\rho, \tag{2.46}$$

which proves the first claim. Now, without loss of generality, assume that $\rho_{k_0} > \|x^0\|_\infty$ with x^0 a feasible point of (P), so that

$$u_k^0 := \rho_k e - x^0 > 0, \text{ for } k \geq k_0. \tag{2.47}$$

Then, by our assumption in the second claim that (P)–(D) is feasible, there exist y^0 and $s^0 \geq 0$ which, together with x^0, satisfy

$$\begin{aligned} A^{Q_k}(x^0)^{Q_k} + A^{\overline{Q}_k}(x^0)^{\overline{Q}_k} = Ax^0 &= b, \\ x^0 + u_k^0 &= \rho_k e, \\ A^T y^0 + s^0 &= c. \end{aligned}$$

On the hand other, from the second, third and fourth block rows of (2.19), and definitions (2.17), (2.18) and (2.14), we get

$$\begin{aligned} A^{Q_k}\tilde{x}_k^{Q_k} &= b, \\ (\tilde{x}_k + \tilde{u}_k)^{Q_k} &= \rho_k e, \quad \tilde{u}_k^{\overline{Q}_k} = \rho_k e, \\ A^T y_k + s_k - z_k &= c. \end{aligned} \tag{2.48}$$

These two groups of equations yield

$$\begin{bmatrix} A^{Q_k} & A^{\overline{Q}_k} & 0 & 0 \\ I & 0 & I & 0 \\ 0 & I & 0 & I \end{bmatrix} \begin{bmatrix} (\tilde{x}_k - x^0)^{Q_k} \\ -(x^0)^{\overline{Q}_k} \\ (\tilde{u}_k - u_k^0)^{Q_k} \\ (\tilde{u}_k - u_k^0)^{\overline{Q}_k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

and

$$\begin{bmatrix} A^{Q_k} & A^{\overline{Q}_k} & 0 & 0 \\ I & 0 & I & 0 \\ 0 & I & 0 & I \end{bmatrix}^T \begin{bmatrix} y^0 - y_k \\ z_k^{Q_k} \\ z_k^{\overline{Q}_k} \end{bmatrix} = \begin{bmatrix} (s_k - s^0)^{Q_k} \\ (s_k - s^0)^{\overline{Q}_k} \\ z_k^{Q_k} \\ z_k^{\overline{Q}_k} \end{bmatrix}.$$

This implies that

$$[(\tilde{x}_k - x^0)^{Q_k}; -(x^0)^{\overline{Q}_k}; (\tilde{u}_k - u_k^0)] \perp [(s_k - s^0)^{Q_k}; (s_k - s^0)^{\overline{Q}_k}; z_k],$$

i.e.,

$$(\tilde{x}_k^{Q_k})^T (s_k - s^0)^{Q_k} - (x^0)^T (s_k - s^0) + (\tilde{u}_k - u_k^0)^T z_k = 0. \tag{2.49}$$

Hence, for C' large enough, we obtain

$$\begin{aligned} (u_k^0)^T z_k + (x^0)^T s_k &= (x^0)^T s^0 + (\tilde{x}_k^{Q_k})^T s_k^{Q_k} - (\tilde{x}_k^{Q_k})^T (s^0)^{Q_k} + \tilde{u}_k^T z_k \\ &\le (x^0)^T s^0 + C'\delta + \gamma_3 e^T s^0 + C'\gamma_2 \end{aligned} \tag{2.50}$$

where the equality comes from the expansion of (2.49), and the inequality from (2.46), (2.43), and (2.45). Since u_k^0, z_k, x^0 and s_k are nonnegative for $k \geq k_0$, we get

$$z_k^i \leq \frac{(x^0)^T s^0 + C'\delta + \gamma_3 e^T s^0 + C'\gamma_2}{(u_k^0)^i}, \quad \forall i,\, k \geq k_0,\, k \in K_\rho.$$

Since (see (2.47)) $(u_k^0)^i \to \infty$, $i \in \mathbf{n}$ as $k \to \infty$ on K_ρ, this proves that

$$\lim_{k\to\infty, k\in K_\rho} z_k = 0,$$

proving the second claim. Finally, if in addition (P) is strictly feasible, then we can select $x^0 > 0$, and (2.50) yields

$$s_k^i \leq \frac{(x^0)^T s^0 + C'\delta + \gamma_3 e^T s^0 + C'\gamma_2}{(x^0)^i}, \quad \forall i, k \geq k_0, k \in K_\rho,$$

proving that $\{s_k\}$ is bounded on K_ρ. Boundednesses of $\{s_k\}$ and $\{z_k\}$, together with equation (2.48) and full-rankness of A, imply that $\{y_k\}$ is bounded on K_ρ.

□

We are now ready to prove that ρ_k is increased at most finitely many times. The proof uses the fact that if (D) has a strictly feasible point, then for all $y \in \mathcal{F}$, $\{a_i : i \in I(y)\}$ is a positively linearly independent set of vectors.

Proposition 2.2. *If* (P)–(D) *is strictly feasible, then ρ_k is increased at most finitely many times, i.e., K_ρ is finite. Furthermore, $\{y_k\}$ and $\{z_k\}$ converge to some y_* and z_*.*

Proof. If the first claim holds, then after finitely many iterations, IrPDIP reduces to rIPM applied to (D_ρ), so the second claim follows by Proposition 2.1. It remains to prove the first claim. Proceeding by contradiction, suppose K_ρ is infinite. Then there exists an infinite index set K and some $Q \subseteq \mathbf{n}$ such that $Q_k = Q$, for all $k \in K$. In view of Lemma 2.3, since $K \subseteq K_\rho$, there must exist $k_0 > 0$ such that conditions (2.12) are satisfied for $k \geq k_0$, $k \in K$; in particular,

$$\begin{aligned} \tilde{x}_k^Q &\geq -\gamma_3 e, \quad k \geq k_0,\, k \in K. && (2.51) \\ \tilde{u}_k^Q &\ngeq \gamma_4 e, \quad k \geq k_0,\, k \in K, && (2.52) \end{aligned}$$

Since $\lim_{k\to\infty} \rho_k = \infty$, it follows from (2.17), (2.18), the third block row of (2.19), and (2.52) that

$$\lambda_k := \|\tilde{x}_k^Q\|_\infty = \|\rho_k e - \tilde{u}_k^Q\|_\infty \to \infty, \quad \text{as } k \to \infty,\, k \in K. \tag{2.53}$$

Hence

$$\|\hat{x}_k^Q\|_\infty = 1, \quad k \geq k_0,\, k \in K \tag{2.54}$$

where we have defined

$$\hat{x}_k^Q := \frac{\tilde{x}_k^Q}{\lambda_k}, \quad k \geq k_0, \forall k \in K. \tag{2.55}$$

(Without loss of generality, we have assumed that $\lambda_k \neq 0$, $\forall k \geq k_0, k \in K$.) Now, in view of Lemma 2.5, we have for certain constant $C > 0$ large enough,

$$\begin{aligned} \|S_k^Q \tilde{x}_k^Q\| &\leq C, \quad \forall k \in K, && (2.56) \\ \|y_k\| &\leq C, \quad \forall k \in K, && (2.57) \\ \lim_{k\to\infty} z_k &= 0, \quad k \in K. && (2.58) \end{aligned}$$

Note that by (2.57) and (2.54), $\{y_k\}$ and $\{\hat{x}_k^Q\}$ are bounded on K, so in view of (2.54) and (2.58), there exists an infinite index set $K' \subseteq K$ such that

$$\hat{x}_k^Q \to \hat{x}_*^Q \neq 0, \quad y_k \to y_*, \quad z_k \to z_* = 0, \quad \text{as } k \to \infty, k \in K', \tag{2.59}$$

for some $\hat{x}_*^Q$ and some $y_* \in \mathcal{F}$ (since $z_* = 0$). Dividing by λ_k and taking the limit on both sides of (2.56), we obtain

$$S_k^Q \hat{x}_k^Q \to 0, \quad \text{as } k \to \infty,\, k \in K'$$

which implies that

$$\hat{x}_*^i = 0, \quad \forall i \in Q \backslash I(y_*). \tag{2.60}$$

On the other hand, the second block equation in (2.19) and equation (2.17) give

$$A^Q \tilde{x}_k^Q = b \quad \forall k.$$

Dividing by λ_k and taking the limit of both sides, and using (2.60), we obtain

$$\sum_{i \in I(y_*) \cap Q} \hat{x}_*^i a^i = 0. \tag{2.61}$$

Now note from (2.51), (2.55) and (2.53) that

$$\hat{x}_*^Q = \lim_{k \to \infty, k \in K'} \frac{\tilde{x}_k^Q}{\lambda_k} \geq \lim_{k \to \infty, k \in K'} \frac{-\gamma_3 e}{\lambda_k} = 0. \tag{2.62}$$

Since the strict feasibility of (D) implies positive linear independence of vectors $\{a^i : i \in I(y_*) \cap Q,\, y_* \in \mathcal{F}\}$, it follows from (2.61) and (2.62) that

$$\hat{x}_*^i = 0, \quad \forall i \in I(y_*) \cap Q.$$

Together with (2.60), we therefore have

$$\hat{x}_*^Q = 0,$$

which is a contradiction to (2.59). □

2.2.4 An ℓ_∞ version

Instead of the ℓ_1 exact penalty function used in (P$_\rho$)–(D$_\rho$), we can use an ℓ_∞ exact penalty function and consider the problem

$$\left.\begin{array}{rc} \max & b^Ty - \rho z \\ \text{s.t.} & A^Ty - ze \le c,\; z \ge 0 \end{array}\right\} \tag{2.63}$$

with its associated primal

$$\left.\begin{array}{rc} \min & c^Tx \\ \text{s.t.} & Ax = b,\; e^Tx + u = \rho, \\ & x \ge 0,\; u \ge 0 \end{array}\right\} \tag{2.64}$$

where $z \in \mathbb{R}$ and $u \in \mathbb{R}$. Again, strictly feasible points for (2.63) are readily available. Conditions akin to (2.11)–(2.12) can again be used to iteratively obtain an appropriate value of ρ. Since both z and u are scalar variables, the scheme can be slightly simplified: Increase ρ if either

$$z_+ > \gamma_1 \frac{z_0}{\rho_0} \rho, \tag{2.65}$$

or

$$\text{(i)}\;\; \|[\Delta y_a; \Delta z_a]\| \le \frac{\gamma_2}{\rho},\;\text{ and (ii) } \tilde{x}^Q \ge -\gamma_3 e, \;\text{ and }\; \text{(iii) } \tilde{u} < \gamma_4. \tag{2.66}$$

An analysis very similar to that of section 2.2.3 shows that the resulting ℓ_∞ variant IrPDIP-ℓ_∞ enjoys the same theoretical properties as the ℓ_1 version IrPDIP; see Appendix B. Minor changes includes substitution of the ℓ_∞-dual norm $\|\cdot\|_1$ for the ℓ_1-dual norm $\|\cdot\|_\infty$.

2.3 Infeasible constraint-reduced MPC: IrMPC

As an instance of IrPDIP, we apply rMPC* of [65] to (P$_\rho$)–(D$_\rho$), and dub the resulting full algorithm IrMPC. (Indeed the search direction in rMPC* satisfies condition (2.4) of rIPM and condition (2.16) of IrPDIP.) In view of Proposition 2.2, subject to strict feasibility of (P)–(D), after finitely many iterations, IrMPC reduced to rMPC* applied to problem (D$_\rho$) with ρ equal to a fixed value $\bar{\rho}$. Thus, we can invoke results from [65] under appropriate assumptions.

Proposition 2.3. *Suppose* (P)-(D) *is strictly feasible. Then* $\{(y_k, z_k)\}$ *converges to a stationary point* (y_*, z_*) *of problem* (D$_\rho$) *with* $\rho = \bar{\rho}$.

Proof. It follows from Theorem 3.8 in [65] that $\{(y_k, z_k)\}$ converges to a stationary point of problem (D$_\rho$) if and only if the penalized dual objective function is bounded. To conclude the proof, we now establish that $\{b^T y_k - \rho_k e^T z_k\}$ is bounded indeed. Lemma 2.2 implies that $\{b^T y_k - \rho_k e^T z_k\}$ is increasing for k large enough that $\rho_k = \bar{\rho}$, so it is sufficient to prove that $\{b^T y_k - \rho_k e^T z_k\}$ is bounded above. Since Lemma 2.3 implies that $\{b^T y_k\}$ is bounded above, this claim follows from boundedness of $\{z_k\}$ and $\{\rho_k\}$ (from Lemma 2.3 and Proposition 2.2 respectively).

□

Under a non-degeneracy assumption,[3] $\{z_k\}$ converges to zero, and thus $\{y_k\}$ converges to an optimal solution of (D). The proof of the following lemma is routine

[3] The question of whether Theorem 3.8 and Proposition 3.9 in [65] hold without assuming linear independence of gradients of active constraints is open. If the answer is positive, then the results established in our Theorem 1 will hold under the sole assumption that (P)–(D) is strictly feasible.

and hence omitted.

Lemma 2.6. *The gradients of active constraints of problem* (D_ρ) *are linearly independent for all* (y, z) *if and only if* $\{a^i : (a^i)^T y = c^i\}$ *is a linearly independent set of vectors for all* $y \in \mathbb{R}^m$*.*

Theorem 2.1. *Suppose* (P)-(D) *is strictly feasible, and for all* $y \in \mathbb{R}^m$*,* $\{a^i : i \in I(y)\}$ *is a linearly independent set of vectors. Then* $z_k \to 0$ *and* $\{y_k\}$ *converges to an optimal solution of problem* (D)*.*

Proof. Lemma 2.6 implies that the gradients of active constraints of problem (D_ρ) are linearly independent for all feasible (y, z). Applying the latter portion of Theorem 3.8 in [65], we conclude that (y_k, z_k) converges to a maximizer (y_*, z_*) of problem $(\mathrm{D}_{\bar\rho})$. Next, Proposition 3.9 of [65] implies that there exists an infinite subsequence K on which $[\tilde{x}_k; \tilde{u}_k]$ converges to an optimal solution $[\tilde{x}_*; \tilde{u}_*]$ of problem $(\mathrm{P}_{\bar\rho})$ and on which

$$[\Delta y_{a,k}; \Delta z_{a,k}] \to 0, \text{ as } k \to \infty, \ k \in K.$$

Thus conditions (i) and (ii) of (2.12) are satisfied on K. On the other hand, since $\rho_k = \bar\rho$ for $k \in K$ large enough, one condition in (2.12) must fail. It follows that $\tilde{u}_k^{Q_k} \geq \gamma_4 e$ for $k \in K$ large enough. Since $\tilde{u}_k^{\overline{Q}_k} = \rho_k e$, from the fourth block row of (2.19) and definition (2.18), we conclude that

$$\tilde{u}_k \geq \min(\gamma_4, \bar\rho)e, \ k \in K \text{ large enough.}$$

It follows that

$$\tilde{u}_* \geq \min(\gamma_4, \bar\rho)e.$$

Hence, complementary slackness implies that $z_* = 0$, and as a consequence, y_* is an optimal solution of problem (D). □

2.4 Analysis without primal or dual feasibility

In practice it is often hard, even impossible, to ascertain *a priori* the (strict) feasibility of the primal and dual problems. This motivates the present section. In this section, we study the behavior of Iteration IrMPC without feasibility assumptions. Our first result deals with the case when $\{\rho_k\}$ is bounded.

Theorem 2.2. *Suppose that, at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set. If ρ_k is increased finitely many times, then either of the following occurs*

(a) $\{y_k\}$ *is unbounded. In this case,* (P) *is infeasible.*

(b) $\{y_k\}$ *is bounded. In this case, every limit point of* $\{y_k\}$ *is optimal for* (D)*; and* (P) *is feasible and* (D) *strictly feasible.*

Proof. We show case (a) by contradiction. Suppose (P) is feasible, then it is implied from Lemma 2.3 that $\{z_k\}$ is bounded and $\{b^T y_k\}$ is bounded from above. Because $\{\rho_k\}$ is increased finitely many times to $\bar{\rho}$, it follows that $\{b^T y_k - \bar{\rho} e^T z_k\}$ is bounded from above. Since Lemma 2.2 implies that $\{b^T y_k - \bar{\rho} e^T z_k\}$ is increasing for k large enough, it follows that $\{b^T y_k - \bar{\rho} e^T z_k\}$ is bounded. By noting that with $\bar{\rho}$, Iteration IrMPC is a special case of Iteration rIPM applied to problem (D$_\rho$), it follows from Lemma 2.1 (ii) that $[y_k; z_k]$ converges, contradicting to the assumption of unboundedness of $\{y_k\}$.

In case (b), since ρ_k is increased only finitely many times, condition (2.11) must be violated, implying that $\{z_k\}$ is bounded. Note that the assumption that (P)–(D) is strictly feasible is used to show that $[y_k; z_k]$ is bounded (see Lemma 2.5) and that ρ_k is increased only finitely many times (see Proposition 2.2), all of which are satisfied in the present claim. Hence, by a similar argument as Proposition 2.3 and Theorem 2.1, we can have that any limit point of $\{y_k\}$ solves (D). It then implies that both (P) and (D) are feasible. To show (D) is strictly feasible, it is equivalent to show there exists y^0 and $\xi > 0$ such that $A^T y^0 - c\xi < 0$. From Motzkin's theorem (page 28, [37]), it suffices to show that there does not exist $x^0 \neq 0$ such that $Ax^0 = 0$, $c^T x^0 \leq 0$ and $x^0 \geq 0$, for which a sufficient condition is that the solution set of (P) is nonempty and bounded. Since the feasibility of (P)–(D) implies the non-emptiness of the solution set of (P), we will next show the solution set of (P) is bounded indeed. Given any solution x_* of (P), there exists y_* as a solution of (D) such that $X_*(c - A^T y_*) = 0$ and $Ax_* = b$, which implies that

$$A^{I(y_*)} x_*^{I(y_*)} = b.$$

Because $\{a^i : I(y_*)\}$ is a linearly independent set, it follows that

$$\|x_*\| = \|x_*^{I(y_*)}\| \leq \|(A^{I(y_*)})^T A^{I(y_*)}\|^{-1} \|(A^{I(y_*)})^T b\|. \tag{2.67}$$

Since there are finitely many possible sets of active constraints, and since (2.67) holds for arbitrary primal solution x_*, this proves the boundedness of the solution set of (P). □

Sequence $\{\rho_k\}$ can tend to infinity, i.e., K_ρ can be infinite, due to infeasibility or the lack of the constraint qualification. This case is studied in the following lemma and theorem.

Lemma 2.7. *Suppose $\{z_k\}$ is bounded, and $\{y_k\}$ has a limit point y_* on K_ρ. If $\rho_k \to \infty$, as $k \to \infty$, then for any limit point $[y_*, z_*]$ of $\{[y_k; z_k]\}$, there exists $\bar{x}_* \neq 0$ with $s_* = c - A^T y_* + z_*$ such that*

$$A\bar{x}_* = 0, \tag{2.68}$$

$$Z_*(e - \bar{x}_*) = 0, \tag{2.69}$$

$$S_*\bar{x}_* = 0, \tag{2.70}$$

$$\bar{x}_* \geq 0. \tag{2.71}$$

Proof. Since $\{z_k\}$ is bounded, condition (2.11) will be violated eventually. Hence, since $\rho_k \to \infty$ as $k \to \infty$, without loss of generality, conditions (2.12) must be satisfied for all $k \in K_\rho$, i.e.,

$$\begin{aligned} \|[\Delta y_k; \Delta z_k]\| &\leq \frac{\gamma_2}{\rho_k}, \quad k \in K_\rho, & (2.72)\\ \tilde{x}_k^{Q_k} &\geq -\gamma_3 e, \quad k \in K_\rho, & (2.73)\\ \tilde{u}_k^{Q_k} &\ngeq \gamma_4 e, \quad k \in K_\rho. & (2.74) \end{aligned}$$

Now, for each k, x_k and u_k is bounded by $C\rho_k$ for some $C > 0$ (see Lemma 2.4), and by construction (see Step 2(ii) of IrMPC),

$$\tilde{x}_k^i = 0 \quad \forall i \notin Q_k. \tag{2.75}$$

It follows from (2.22) and (2.72) that there exists $C > 0$ such that

$$\|Z_k \tilde{u}_k\| = \|U_k \Delta z_k\| \le C\rho_k \frac{\gamma_2}{\rho_k} = C\gamma_2,\ k \in K_\rho, \tag{2.76}$$

$$\|S_k \tilde{x}_k\| = \|S_k^{Q_k} \tilde{x}_k^{Q_k}\| = \|X_k^{Q_k} \Delta s_k^{Q_k}\| \le C\rho_k \frac{\gamma_2}{\rho_k} = C\gamma_2,\ k \in K_\rho, \tag{2.77}$$

and from (2.19) and (2.17) that

$$A\tilde{x}_k = A^{Q_k} \tilde{x}_k^{Q_k} \;=\; b, \tag{2.78}$$

$$\tilde{x}_k + \tilde{u}_k \;=\; \rho_k e. \tag{2.79}$$

Define

$$\bar{x}_k = \frac{\tilde{x}_k}{\rho_k}, \quad \text{and} \quad \bar{u}_k = \frac{\tilde{u}_k}{\rho_k}.$$

Lemma 2.4 then implies that $\{\bar{x}_k\}$ and $\{\bar{u}_k\}$ are bounded. Noting that by assumption, z_k is also bounded, let $[y_*, z_*, s_*, \bar{x}_*, \bar{u}_*]$ be a limit point of $[y_k, z_k, s_k, \bar{x}_k, \bar{u}_k]$ on K_ρ. Dividing ρ_k on both sides of (2.73)–(2.79), and taking limit, we derive

$$\bar{x}_* \ge 0,$$

$$\bar{u}_* \not\ge 0, \tag{2.80}$$

$$\|Z_* \bar{u}_*\| = 0,$$

$$\|S_* \bar{x}_*\| = 0,$$

$$A\bar{x}_* = 0,$$

$$\bar{x}_* + \bar{u}_* = e, \tag{2.81}$$

proving the claim. (Note that (2.80) and (2.81) imply $x_* \ne 0$.) □

Theorem 2.3. *Suppose $\rho_k \to \infty$ as $k \to \infty$, then one of the cases must occur:*

(a) $\{\|y_k\|\} \to \infty$ *as* $k \to \infty, k \in K_\rho$. *In this case,* (P) *is not strictly feasible or* (D) *is infeasible.*

(b) $\{y_k\}$ *has a limit point on* K_ρ *and* $\{z_k\}$ *is unbounded. In this case,* (P) *is infeasible.*

(c) $\{y_k\}$ *has a limit point on* K_ρ, $\{z_k\}$ *is bounded, and there exists an infinite set* $K \subseteq K_\rho$ *such that* $\lim_{k\in K, k\to\infty}\{y_k\} = y_*$ *for some* x_* *and* $\lim_{k\to\infty}\inf_{k\in K}\|z_k\| = 0$. *In this case,* (D) *is feasible but not strictly feasible.*

(d) $\{y_k\}$ *has a limit point on* K_ρ, $\{z_k\}$ *is bounded, and for any infinite set* $K \subseteq K_\rho$ *such that* $\lim_{k\to\infty, k\in K} y_k = y_*$ *for some* $y_* \in \mathbb{R}^m$, *it satisfies that* $\lim_{k\to\infty}\sup_{k\in K}\|z_k\| > 0$. *In this case,* (D) *is infeasible.*

Proof. Claim (a) follows from last claim of Lemma 2.5 and claim (b) follows from the inverse negative proposition of Lemma 2.3.

We next prove (c) and (d), for which we mainly use Lemma 2.7. We first show (c). Let $[y_*; z_*; s_*]$ be any limit point of $\{[y_k; z_k; s_k]\}$ on K such that $z_* = 0$. Since for all k, $s_k \geq 0$, we have from the dual feasibility of $[y_k; z_k]$,

$$c - A^T y_* = s_* - z_* = s_* \geq 0, \tag{2.82}$$

i.e., y_* is a feasible point of (D). To show (D) is not strictly feasible, given any $y \in \mathbb{R}^m$, we obtain from (2.82)

$$s_* + A^T(y_* - y) = c - A^T y. \tag{2.83}$$

From Lemma 2.7, there exists $\bar{x}_* \neq 0$ satisfying (2.68)–(2.71). Left-multiplying

(2.83) by $\bar{x}_*^T$ yields

$$\bar{x}_*^T(c - A^T y) = 0 \quad \forall y \in \mathbb{R}^m,$$

where we have used equation (2.70) and (2.68). Since $\bar{x}_* \geq 0$ from (2.71), and since $\bar{x}_* \neq 0$, it follows that the set $\{y : A^T y \leq c\}$ has no strictly point, proving the claim.

Next, we show (d). Let $[y_*; z_*; s_*]$ be any limit point of $\{[y_k; z_k; s_k]\}$ on K, so (since $z_k \geq 0$, $\forall k$)

$$z_* \neq 0 \text{ with } z_* \geq 0. \tag{2.84}$$

To show that (D) is infeasible, i.e., there does not exist y such that $A^T y \geq c$, by Farkas' Lemma, it suffices to show that there exists $\bar{x}_*$ satisfying

$$A\bar{x}_* = 0, \quad \bar{x}_* \geq 0, \quad c^T\bar{x}_* < 0.$$

Again, let $\bar{x}_*$ satisfy (2.68)–(2.71). It remains to show $c^T\bar{x}_* < 0$, which can be derived as

$$c^T\bar{x}_* \;\; = \;\; (A^T y_* + s_* - z_*)^T\bar{x}_* = -z_*^T\bar{x}_* = -e^T z_* < 0,$$

where the first equality comes from dual feasibility of $[y_k; z_k]$, the second one from (2.68) and (2.70), the last one from (2.69), and the inequality from (2.84).

□

Theorem 2.3 summarizes the feasibility of (P)–(D) with respect to the behavior of Iteration IrMPC. On the other direction, the following result gives the behavior of Iteration IrMPC with respect to the feasibility of (P)–(D).

Theorem 2.4. *Suppose for every $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set, then the following properties hold*

(a) *If* (P) *is feasible, then $\{z_k\}$ is bounded.*

(b) *If* (P) *and* (D) *are both strictly feasible, then ρ_k is increased only finitely many times and $z_k \to 0$ as $k \to \infty$, $k \in K_\rho$, and every limit point of the bounded sequence $\{y_k\}$ is optimal for* (D).

(c) *If* (P) *is infeasible or* (D) *is not strictly feasible, then $\{[y_k; z_k]\}$ is unbounded.*

(d) *If* (P) *is feasible and* (D) *is not strictly feasible, then $\{y_k\}$ is unbounded.*

Proof. Claim (a) is a restatement of Lemma 2.3, and claim (b) is a restatement of Theorem 2.1.

We show claim (c) by contradiction. Suppose $\{[y_k; z_k]\}$ is bounded. Note that the strict feasibility of (P)–(D) is used to show that $[y_k; z_k]$ is bounded in Proposition 2.2, and note that linear independence of active constraints at any point is a stronger condition than positive linear independence of active constraints at feasible points of (D), thus a similar argument as Proposition 2.2 can show that ρ_k is increased finitely many times. It then follows from Theorem 2.2 (b) that (D) is strictly feasible and (P) is feasible, contradicting to the assumption. In view of claim (a), claim (d) follows immediately from claim (c).

□

2.5 Numerical results

2.5.1 Implementation

IrMPC was implemented in MATLAB R2009a. All tests were run on a laptop machine (Intel R / 1.83G Hz, 1GB of RAM, Windows XP professional 2002). To eliminate random errors in measured CPU time, we report averages over 10 repeated runs.

The initial points were set as follows. we adopted (typical infeasible) initial conditions (x_0, y_0, s_0) from [39] for problems (P)–(D). Namely, we first computed

$$\tilde{y} := (AA^T)^{-1}Ac, \;\; \tilde{s} := c - A^T\tilde{y}, \;\; \tilde{x} := A^T(AA^T)^{-1}b,$$

$$\delta_x := \max(-1.5 * \min(\tilde{x}), 0), \;\; \delta_s := \max(-1.5 * \min(\tilde{s}), 0),$$

$$\tilde{\delta}_x := \delta_x + 0.5 * \frac{(\tilde{x} + \delta_x e)^T(\tilde{s} + \delta_s e)}{\sum_{i=1}^n(\tilde{x}^i + \delta_x)}, \;\; \tilde{\delta}_s := \delta_s + 0.5 * \frac{(\tilde{x} + \delta_x e)^T(\tilde{s} + \delta_s e)}{\sum_{i=1}^n(\tilde{s}^i + \delta_s)}$$

and selected (x_0, y_0, s_0) to be

$$x_0 := \tilde{x} + \tilde{\delta}_x e, \;\; y_0 := \tilde{y}, \;\; s_0 := \tilde{s} + \tilde{\delta}_s e. \tag{2.85}$$

The vector z_0 (for the penalized problem) was set to be

$$z_0 := A^T y_0 - c + s_0$$

and under the idea of centrality, initial point u_0 was computed as

$$(u_0)_i := \frac{\mu_0}{(z_0)_i}, \; i \in \mathbf{n}$$

where $\mu_0 := \frac{(x_0)^T s_0}{n}$. The penalty parameter was initialized with $\rho_0 := \|x_0 + u_0\|_\infty$ for the version with the ℓ_1 exact penalty function, and with $\rho_0 := e^T x_0 + u_0$ for the ℓ_∞ version.

The parameters for rMPC* (in Step 1 (ii) and Step 2 (i)-(iii) of IrMPC) were set to the same values as in section 5 ("Numerical Experiments") of [65]. As for the adaptive scheme (2.11)–(2.12), parameters were set to $\sigma := 10$, $\gamma_1 := 10$, $\gamma_2 := 1$, $\gamma_3 := 100$, $\gamma_4 := 100$, and the Euclidean norm was used in (2.11) and (2.12). We chose Q according to the most active rule (Rule 2.1 in [65] with $\epsilon = \infty$), which selects the constraints that have smallest slacks s. Analogously to [65], we terminated when

$$\max\left\{\frac{\|[b - Ax;\, \rho e - x - u]\|}{1 + \|[x; u]\|}, \frac{c^T x - b^T y + \rho e^T z}{1 + |b^T y - \rho e^T z|}\right\} < tol$$

where we used $tol = 10^{-8}$.

We applied IrMPC on problems from the Netlib LP test problem set collection and randomly generated problems.

2.5.2 Problems from COAP

We applied IrMPC to the selected LP problems (those in standard forms) from the Netlib LP test problem set [56]. The Matlab-data format version of this problem set, needed in our Matlab environment to save data conversion, is available at the COAP collection [57]. This problem set was solved because they have known solutions, and hence, provided us benchmark references.

We have verified that IrMPC generates correct solutions as listed in [57]. Furthermore, Table 2.1 shows certain specific properties of results by the ℓ_1 version of IrMPC for $|Q| = n$. The first column lists the names of selected problems, and the next two columns their sizes: number of variables (m) and number of constraints (n). The fourth column shows the feasibility of point y_0 initialized as (2.85) for corresponding problems: feasible (F) and infeasible (IF). In the last two columns, x_* denotes the primal optimal solution, and $\bar{\rho}$ the final value of the penalty parameter. As can be seen from the table, $\bar{\rho}$ is greater than $\|x_*\|_\infty$, consistent with our analysis. These benchmark tests demonstrate that IrMPC obtains the optimal solutions for those test programs with either dual feasible or infeasible initial points. Similar behaviors are exhibited by the ℓ_∞ version, and hence omitted. Comparison of the CPU time and the number of iterations between these two versions is shown in Table 2.2. They have not much difference in the number of iterations. In terms of the CPU time, the ℓ_∞ version is slightly faster. This is because it has less overhead, resulting from less variables and hence, less time in the line searches and variable updates.

2.5.3 Randomly generated problems

We generated standard linear problems of size $m = 100$ and $n = 20000$. Entries of matrix A and vectors b are independently normally distributed. We set vector $c := A^T y + s$ with a normally distributed vector $y \sim \mathcal{N}(0, 1)$ and with a vector s uniformly generated on $[0, 1]$, which guarantees that the dual problem is strictly feasible. We generated 10 random problems. The average CPU time and iteration counts for solving those 10 problems are shown in Figures 2.2 and 2.3 for various values of $|Q|$ for the ℓ_1 and ℓ_∞ versions, respectively. Point y_0 initialized in (2.85) was infeasible for (D) for all generated problems. The fraction of kept constraints, defined by $|Q|/n$, is showed in the horizontal axis with a logarithmic scale. The rightmost

Table 2.1: Results of the Netlib LP test problems by the ℓ_1 version of IrMPC

Problem	m	n	Initial Feasibility	$\|x_*\|_\infty$	$\bar{\rho}$
SCRS8	491	1169	IF	286.2776	350.5974
SCSD1	78	760	F	0.7073	2.9960
SCSD6	148	1350	F	0.9488	2.5005
SCSD8	398	2750	F	22.5000	42.5128
SHIP04L	403	2118	IF	187.3789	213.8128
SHIP04S	403	1458	IF	190.9176	213.7949
SHIP08L	779	4283	IF	126.0000	196.6515
SHIP08S	779	2387	IF	126.0000	196.6692
SHIP12L	1152	5427	IF	135.9453	223.9894
SHIP12S	1152	2763	IF	134.2941	224.7715
TRUSS	334	587	F	725.6492	1222.8
WOODW	1099	8405	IF	0.6234	5.9225
WOOD1P	245	2594	IF	1.4429	2.7623

Table 2.2: Comparison of the results by the ℓ_1 and ℓ_∞ versions for the Netlib LP test problems

Problem	Iter (ℓ_∞)	CPU Time (ℓ_∞)	Iter (ℓ_1)	CPU time (ℓ_1)
SCRS8	23	0.4250	51	1.3391
SCSD1	9	0.1063	11	0.1500
SCSD6	11	0.1375	11	0.2328
SCSD8	10	0.2250	10	0.3719
SHIP04L	14	1.3938	14	1.7172
SHIP04S	15	1.0547	15	1.3539
SHIP08L	16	8.5594	16	9.0609
SHIP08S	15	4.5297	15	4.7188
SHIP12L	18	20.1750	18	23.0625
SHIP12S	17	7.9234	17	8.0672
TRUSS	20	10.2016	20	11.3250
WOODW	31	1.5300	31	1.8204
WOOD1P	29	6.3563	29	10.1047

point, corresponding to $|Q| = n$, is the result without constraint reduction. As can be seen from both Figures 2.2 and 2.3, CPU time decreases as $|Q|/n$ decreases, till as little as 1% of constraints. As was already observed in [60] and [65], the number of iterations remains constant for a large range of fractions $|Q|/n$. Note that the CPU time at $|Q| = n$ in the ℓ_∞ version is less than that in the ℓ_1 version. While the cost per iteration for these two exact penalty functions is about the same, as seen in Figure 2.3, that difference comes from the fewer iterations the ℓ_∞ version takes.

We compared the result with the infeasible Mehrotra's predictor-corrector (MPC) method in [66]. Initial points for MPC were set the same as in (2.85). Figure 2.2 and 2.3 denote the CPU time and the number of iterations obtained from MPC by a dashed magenta line. It shows that for a large range of fractions $|Q|/n$ (from 0.7% to 99% in the ℓ_1 version, and nearly 0 to 100% in the ℓ_∞ version), algorithm IrMPC takes less time to solve the problem than algorithm MPC does. In particular, the time obtained by IrMPC using fractions from 1% to 10% is reduced to approximately one sixth of that by MPC.

Furthermore, we give the statistical data by the ℓ_1 penalty function for 1000 randomly generated problems. Figure 2.4 show the average of the CPU time and number of iterations, and Figure 2.5 shows the corresponding average of speedup gain. Note that the speedup gain for CPU time (resp. number of iterations) is defined as the ratio of time (resp. number of iterations) with n constraints and $|Q|$ constraints. As we can see, with more than 1% constraints, constraint reduction is better than no constraint reduction. In particular, using 1% $\sim$ 10% constraints yields an average of more than 8 times speed-up. Figure 2.6 and 2.7 give the worst

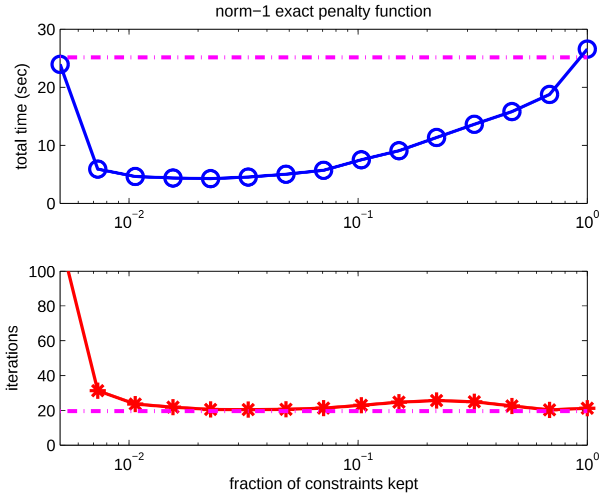


Figure 2.2: CPU time and iterations with the ℓ_1 penalty function.

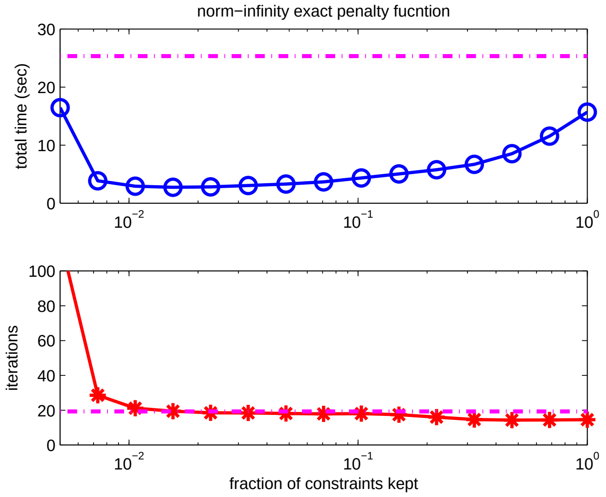


Figure 2.3: CPU time and iterations with the ℓ_∞ penalty function.

case and standard deviation of the speedup gain, respectively. As can be seen in the worst case, constraint reduction takes less time than no constraint reduction with 1% $\sim$ 8% constraints, even when the corresponding number of iterations is about four times of that without constraint reduction. The standard deviation of CPU time is less than 1.4, and that of number of iterations is rather small.

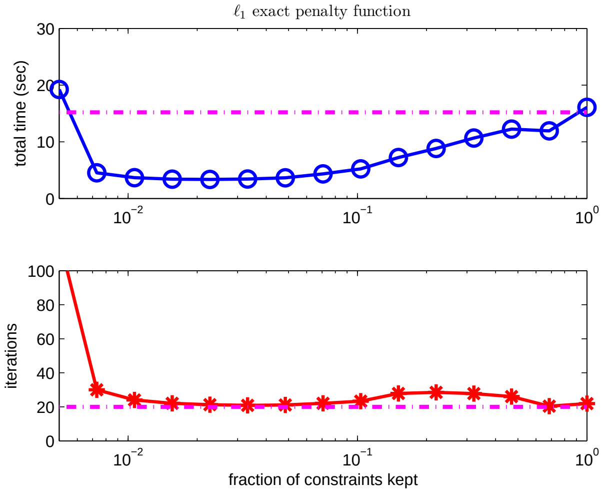


Figure 2.4: The average of CPU time and iterations for 1000 randomly generated problems.

2.6 Conclusion

We have two contributions in constraint reduction for linear optimization (see a published version [27]). At first, we have outlined a general framework (rIPM) for a class of constraint-reduced, dual-feasible interior-point methods that encompasses several previously proposed algorithms, and proved, for all methods in that class, that the dual sequence converges to a single point. In order to accommodate important classes of problems for which an initial dual-feasible point is not read-

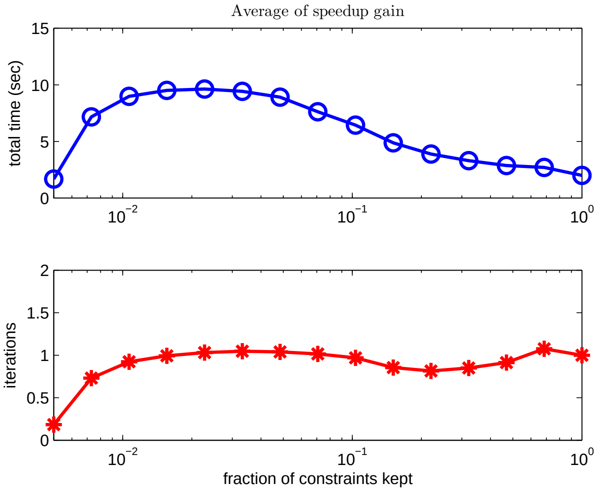


Figure 2.5: The average speedup gain of CPU time and iterations for 1000 randomly generated problems.

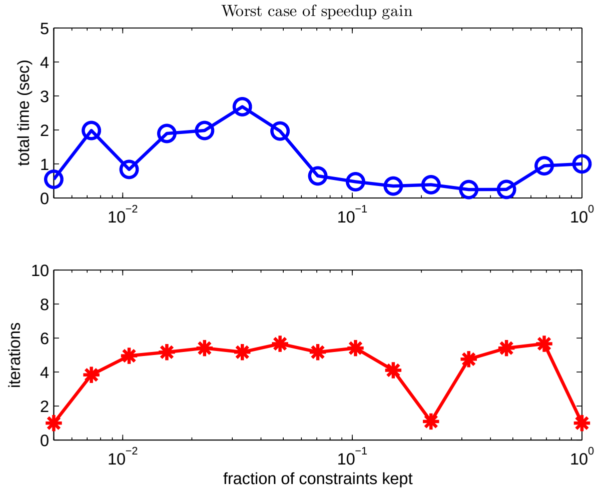


Figure 2.6: The worst speedup gain of CPU time and iterations for 1000 randomly generated problems.

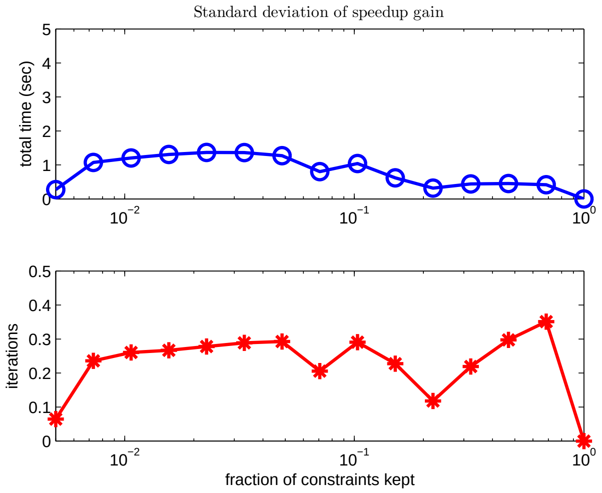


Figure 2.7: The standard deviation of the speedup gain of CPU time and iterations for 1000 randomly generated problems.

ily available, we have then proposed an ℓ_1/ℓ_∞ penalty-based extension (IrPDIP) of this framework for *infeasible* constraint-reduced primal-dual interior point. We have shown that the penalty adjustment scheme in IrPDIP has the property that, under the sole assumption that the primal-dual pair is strictly feasible, the penalty parameter remains bounded.

An infeasible constraint-reduced variant of Mehrotra's Predictor Corrector (specifically, an infeasible variant of rMPC* from [65]), dubbed IrMPC, was then considered, as an instance of IrPDIP. IrMPC was analyzed, and tested on the Netlib problem set of linear problems and randomly generated problems. The results show promise that IrMPC obtains major speed-ups while handling infeasible initial points.

It is an open problem about the time complexity of IrPDIP and IrMPC. Time complexity is a hard problem here because the algorithms here don't force strict restrictions on the step-size.

Chapter 3

Infeasible Constraint Reduction for Convex Quadratic Optimization

In this chapter, we extend the *infeasible* constraint-reduced algorithm to convex quadratic optimization, and derive the results that generalize those in linear optimization. Previous work of Jung et al. [32] used exact penalty functions to allow for infeasible initial points in the constraint-reduced algorithm. The unsolved issue, about how to choose an appropriate penalty parameter, is addressed here.

3.1 An infeasible constraint-reduced IPM

3.1.1 Basic ideas

The constraint reduction PDIPM (see section 1.2.3) for CQPs requires a strictly dual-feasible initial point. To address this limitation, as in the linear programming problems, we introduce an ℓ_1 or ℓ_∞ exact penalty function. Specifically in the ℓ_1 case, as in [32], we consider the relaxed problem of (Dq)

$$\left.\begin{array}{ll} \max_{y,z} & b^T y - \rho e^T z - \frac{1}{2} y^T H y \\ \text{s.t.} & A^T y - z \leq c,\ z \geq 0, \end{array}\right\} \qquad (\mathrm{Dq}_\rho)$$

with its associated primal problem

$$\left.\begin{aligned} \min \quad & c^T x + \tfrac{1}{2} y^T H y \\ \text{s.t.} \quad & Ax + Hy = b, \\ & x + u = \rho e, \\ & x \geq 0,\ u \geq 0. \end{aligned}\right\} \tag{Pq$_\rho$}$$

Strictly feasible points are readily available for the penalized problem (Dq$_\rho$): Given any y, feasible or infeasible for (Dq), selecting $z > \max\{0, A^T y - c\}$ makes (y, z) strictly feasible for (Dq$_\rho$).

Following [32], applying constraint reduction schemes to problem (Dq$_\rho$) yields the reduced penalized problem[1]

$$\max_{y,z} \quad b^T y - \rho e^T z - \frac{1}{2} y^T H y \tag{3.1}$$

$$\text{s.t.} \quad (A^Q)^T y - z^Q \leq c^Q,\ z \geq 0. \tag{3.2}$$

By applying primal-dual interior-point methods to problem (3.1)–(3.2), the reduced affine-scaling direction $(\Delta x^Q, \Delta u, \Delta y, \Delta z, \Delta s^Q)$ can be obtained by solving the sys-

[1] Note that, once again, we don't use constraint-reduction to constraints $z \geq 0$; see the reason in footnote 5 of Chapter 2.

tem

$$\begin{bmatrix} 0 & 0 & 0 & (A^Q)^T & -I & 0 & I \\ A^Q & 0 & 0 & H & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 \\ S^Q & 0 & 0 & 0 & 0 & 0 & X^Q \\ 0 & Z^Q & 0 & 0 & U^Q & 0 & 0 \\ 0 & 0 & Z^{\overline{Q}} & 0 & 0 & U^{\overline{Q}} & 0 \end{bmatrix} \begin{bmatrix} \Delta x^Q \\ \Delta u^Q \\ \Delta u^{\overline{Q}} \\ \Delta y \\ \Delta z^Q \\ \Delta z^{\overline{Q}} \\ \Delta s^Q \end{bmatrix} = \begin{bmatrix} 0 \\ b - A^Q x^Q - Hy \\ \rho e - x^Q - u^Q \\ \rho e - u^{\overline{Q}} \\ -X^Q s^Q \\ -Z^Q u^Q \\ -Z^{\overline{Q}} u^{\overline{Q}} \end{bmatrix}. \qquad (3.3)$$

After block Gaussian elimination, this system can be derived into normal equations

$$M^{(Q)} \begin{bmatrix} \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} b - Hy \\ -\rho e \end{bmatrix}, \qquad (3.4)$$

$$\Delta s^Q = -(A^Q)^T \Delta y + \Delta z^Q, \qquad (3.5)$$

$$\begin{bmatrix} \Delta x^Q \\ \Delta u \end{bmatrix} = -\begin{bmatrix} x^Q \\ u \end{bmatrix} - \begin{bmatrix} (S^Q)^{-1} X^Q & 0 \\ 0 & Z^{-1} U \end{bmatrix} \begin{bmatrix} \Delta s^Q \\ \Delta z \end{bmatrix}, \qquad (3.6)$$

where

$$M^{(Q)} = \begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A^Q & 0 \\ -E^Q & -I \end{bmatrix} \begin{bmatrix} (S^Q)^{-1} X^Q & 0 \\ 0 & Z^{-1} U \end{bmatrix} \begin{bmatrix} A^Q & 0 \\ -E^Q & -I \end{bmatrix}^T. \qquad (3.7)$$

As in [32], further eliminating Δz, we can reduce (3.4) to

$$(H + A^Q D^{(Q)} (A^Q)^T) \Delta y = b - Hy - A^Q X^Q (S^Q)^{-1} (E^Q)^T (D_2^{(Q)})^{-1} \rho e, \qquad (3.8)$$

$$D_2^{(Q)} \Delta z = -\rho e + E^Q X^Q (S^Q)^{-1} (A^Q)^T \Delta y, \qquad (3.9)$$

where diagonal positive-definite matrices $D^{(Q)}$ and $D_2^{(Q)}$ are respectively defined in (2.25) and (2.24). The dominant cost of computing $\left(\Delta x^Q, \Delta u, \Delta y, \Delta z, \Delta s^Q\right)$ is to solve (3.8), which mainly needs to form the matrix

$$A^Q D^{(Q)} (A^Q)^T = \sum_{i \in Q} d^i a^i (a^i)^T.$$

It takes $|Q|m^2$ flops, enjoying the same speed-up as *feasible* constraint-reduced IPMs. (see (1.27).)

The ℓ_1 penalty function is "exact" (see the definition on page 23, section 2.2.1), so that an appropriate value of ρ is needed. Such a value, as in the linear case, can be obtained by the scheme for nonlinear optimization in [61]. Once again, our scheme does not require the assumption on the boundedness of the sequences of iterates y_k. Since quadratic optimization is more general than linear optimization, obviously, boundedness of y_k can not be guaranteed. To illustrate this, a toy example (based on the linear example (2.9)) is constructed as

$$\max_{y^1,\, y^2} \quad y^1 + \frac{1}{2} \begin{bmatrix} y^1 \\ y^2 \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} \quad \text{s.t.} \quad y^1 \leq 0,\, 2y^1 \leq 2. \tag{3.10}$$

This example has no constraints on variable y^2, so $y^2_* = 0$ at the solution. Substituting y^2_* reduces (3.10) to the linear program (2.9) with the only variable y^1. The penalized problem of (2.9) is unbounded when $\rho < \frac{1}{3}$; see Figure 2.1. Therefore, the scheme is: at the end of each iteration, increase ρ when either

$$\|z\|_\infty \geq \gamma_1 \frac{\|z_0\|_\infty}{\rho_0} \rho \tag{3.11}$$

OR

$$\text{(i)} \ \ \|[\Delta y; \Delta z]\| \leq \gamma_2, \ \text{AND (ii)} \ \tilde{x}^Q \geq -\gamma_3 e, \ \text{AND (iii)} \ \tilde{u}^Q \not\geq \gamma_4 e \tag{3.12}$$

is satisfied. This scheme is the same as in the linear case (2.11)–(2.12) except the first condition of (3.12). The difference arises because algorithm IrQP forces the boundedness of primal variables (see (3.21) and (3.22) below), so the first condition in (3.12) is enough to indicate that the current iterate is close to a stationary point. (Such closeness is measured by the product of primal variables $[x; u]$ and dual direction $[\Delta y; \Delta z]$.) But in the linear case, the primal variables are only bounded by a linear function of ρ_k (see Lemma 2.4) where ρ_k might be unbounded.

3.1.2 Algorithm statement

We are now ready to state Iteration IrQP, an infeasible constraint-reduced interior-point algorithm for quadratic program (Dq). It is identical to the iterations considered in [32] except for Step 2 (iii). Besides updating ρ, Step 2 (iii) also forces the dual variables to be centralized when ρ is increased (see (3.23)–(3.24)), which is shown to be much more efficient in our implementation. The updates (3.21)–(3.22) for primal variables are from [32] because [32] is what our convergence analysis is based on. Another method for updates in linear optimization [65] is also suitable.

Iteration IrQP

Parameters: $\beta \in (0,1)$, $\sigma > 1$, $\gamma_i > 0$, for $i = 1,2,3,4$; $w_{\min} > 0$ and $\chi > 0$.

Data: $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ such that $z > \max\{0, A^T y - c\}$; $s := c - A^T y + z$; $x \in \mathbb{R}^n$, $u \in \mathbb{R}^n$ and $\rho \in \mathbb{R}$ such that $x > 0$, $u > 0$ and $\rho > 0$; $Q \subseteq \mathbf{n}$ such that $[H\ A^Q]$ has full rank.

Step 1: Computation of the search direction.

(i). Obtain $\left(\Delta x^Q, \Delta u, \Delta y, \Delta z, \Delta s^Q\right)$ by solving (3.4)–(3.6). Compute

$$\Delta s^{\overline{Q}} := -(A^{\overline{Q}})^T \Delta y + \Delta z^{\overline{Q}}. \tag{3.13}$$

Set

$$\tilde{x}^i := \begin{cases} x^i + \Delta x^i, & i \in Q, \\ 0, & i \notin Q, \end{cases} \tag{3.14}$$

$$\tilde{u} := u + \Delta u, \tag{3.15}$$

$$\tilde{y} := y + \Delta y, \tag{3.16}$$

and set

$$\tilde{x}_- := \min\{\tilde{x}, 0\}, \quad \tilde{u}_- := \min\{\tilde{u}, 0\}. \tag{3.17}$$

(ii). Compute

$$\hat{t} := \arg\max\{\bar{t} \in [0,1] \,|\, s + \bar{t}\Delta s \geq 0, z + \bar{t}\Delta z \geq 0\}; \tag{3.18}$$

Set step sizes

$$t := \max\{\beta\hat{t}, \hat{t} - \|[\Delta y; \Delta z]\|\}; \tag{3.19}$$

Step 2. Updates.

(i). Dual variables: set

$$(y_+, s_+, z_+) := (y, s, z) + t(\Delta y, \Delta s, \Delta z). \tag{3.20}$$

(ii). Primal variables: set

$$x_+^i := \max\{\max\{\min\{\|[\Delta y; \Delta z]\|^2 + \| \, [\tilde{u}_-; \tilde{x}_-] \, \|^2, w_{\min}\}, \tilde{x}^i\}, \chi\} \quad i \in \mathbf{n}; \tag{3.21}$$

$$u_+^i := \max\{\max\{\min\{\|[\Delta y; \Delta z]\|^2 + \| \, [\tilde{u}_-; \tilde{x}_-] \, \|^2, w_{\min}\}, \tilde{u}^i\}, \chi\} \quad i \in \mathbf{n}. \tag{3.22}$$

(iii). The penalty parameter: If either (3.11) or (3.12) is satisfied, set

$$\rho_+ := \sigma\rho \tag{3.23}$$

and

$$\mu := \frac{x^T s + u^T z}{2n}, \quad x^i := \frac{\mu}{s^i}, \quad u^i := \frac{\mu}{z^i} \quad \forall i \in \mathbf{n}; \tag{3.24}$$

otherwise set $\rho_+ := \rho$.

(iv). The working set: select Q_+ such that $[H \;\; A^{Q_+}]$ has full row rank.

□

It is clear from Iteration IrQP that $x_+ > 0$, $u_+ > 0$, $z_+ > \max\{0, A^T y_+ - c\}$, and (see (3.26) below) $s_+ = c - A^T y_+ + z_+$. Further, since $[H \;\; A^Q]$ has full row rank and $(x, u, s, z) > 0$, $M^{(Q)}$ (see (3.7)) is positive definite, so the search direction $(\Delta y, \Delta z)$

in Step 1 (i) of IrQP is well defined. Hence, Iteration IrQP can be repeated indefinitely, generating an infinite sequence of iterates. Once again, k is attached to denote the kth iterate.

Also, note that equations (3.5) and (3.13) imply

$$A^T \Delta y_k - \Delta z_k + \Delta s_k = 0 \quad \forall k. \tag{3.25}$$

Since $s_0 = c - A^T y_0 + z_0$ (see Data section of Iteration IrQP), in view of (3.20), it follows that

$$A^T y_k - z_k + s_k = c, \quad s_k > 0, \quad z_k > 0, \quad \forall k, \tag{3.26}$$

i.e., for all k, the primal iterate (y_k, z_k) is strictly feasible for (Dq_ρ).

3.1.3 Boundedness of the sequence of penalty parameters

In this section, we show that ρ_k (generated by Iteration IrQP) is increased only finitely many times under the assumption that the primal and dual problems are both strictly feasible. This is the same result that can be proven with the same procedure as in linear optimization, but is more complicated to derive because of the difficulties caused by the nonzero Hessian matrix. We will point out the differences from the LP case and explain the resulting difficulties throughout the analysis.

The following lemma states that the objective function of (Dq_ρ) increases when parameter ρ is not changed.

Lemma 3.1. *(Corresponds to Lemma 2.2) For all k, $f(y_{k+1}) - \rho_k e^T z_{k+1} \geq f(y_k) - \rho_k e^T z_k$.*

Proof. At each iterate k, the update of dual iterates can be viewed as applying to (Dq_{ρ_k}) one iteration of Algorithm A of [32] without increasing the parameter ρ_k. Thus, the claim follows from Proposition A. 4 (i) of [32] with α there substituted by t with $0 \leq t \leq 1$ (see (3.18) and (3.19)). □

Remark: Because the value of ρ_k might increase at some iteration due to the scheme (3.11)–(3.12), it is not necessarily true that for all k, $f(y_{k+1}) - \rho_{k+1} e^T z_{k+1} \geq f(y_k) - \rho_k e^T z_k$.

For future reference, note that because H is positive semidefinite, given any $p \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$, it holds that $(y-p)^T H (y-p) \geq 0$, or equivalently

$$y^T Hy - 2p^T Hy \geq -p^T Hp. \tag{3.27}$$

The following lemma shows the boundedness of sequence $\{z_k\}$. Unlike the corresponding Lemma 2.3 of linear optimization, the difficulty is to find a *decreasing* sequence $\{\nu_k\}$ that satisfies the boundedness property $\nu_k \leq z_k$ for all k; see (3.36) below. Such a sequence $\{\nu_k\}$ is hard to construct because we lack the information about the boundedness of sequences $\{y_k^T H y_k\}$ and $\{(y^0)^T H y_k\}$ with the given y^0 (see the right hand side of (3.35) below), both of which simply vanish in linear optimization.

Lemma 3.2. *(Corresponds to Lemma 2.3) Suppose* (Pq) *is feasible, then sequence* $\{z_k\}$ *is bounded.*

Proof. If ρ_k is increased finitely many times to a finite value, say ρ_∞, then condition (3.11) must fail for k large enough, i.e., $\|z_k\|_\infty \leq \gamma_1 \frac{\|z^0\|_\infty}{\rho^0} \rho_\infty$ for k large enough, proving the claim. It remains to prove that $\{z_k\}$ is bounded when ρ_k is increased infinitely many times, i.e., when $\rho_k \to \infty$ as $k \to \infty$.

By assumption that (Pq) has a feasible point, say (x^0, y^0), we have

$$Ax^0 + Hy^0 = b,\ x^0 \geq 0. \tag{3.28}$$

Since $\rho_k \to \infty$ as $k \to \infty$, there exists k_0 such that

$$\rho_k > \|x^0\|_\infty,\ \forall k \geq k_0. \tag{3.29}$$

Since $(y_k,\ z_k)$ is dual feasible for (Dq_{ρ_k}) for all k (see (3.26)), we have

$$A^T y_k \ \leq \ z_k + c, \tag{3.30}$$

$$z_k \ \geq \ 0. \tag{3.31}$$

Left-multiplying by $(x^0)^T \geq 0$ on both sides of (3.30), using (3.28), yields

$$(b - Hy^0)^T y_k \leq z_k^T x^0 + c^T x^0. \tag{3.32}$$

Adding $\rho_k e^T z_k$ to both sides of (3.32), after simple reorganization, we get

$$(\rho_k e - x^0)^T z_k \leq c^T x^0 - (b - Hy^0)^T y_k + \rho_k e^T z_k. \tag{3.33}$$

Next, inequality (3.27) implies that

$$-\frac{1}{2} y_k^T H y_k + (y^0)^T H y_k + c^T x^0 \leq \frac{1}{2}(y^0)^T H y^0 + c^T x^0 =: M, \quad \forall k. \tag{3.34}$$

In view of (3.29) and (3.31), it follows from (3.33) and (3.34) that, for all i,

$$\begin{aligned} 0 \leq z_k^i \ &\leq \ \frac{c^T x_0 - (b - Hy^0)^T y_k + \rho_k e^T z_k}{\rho_k - (x^0)^i} && (3.35)\\ &\leq \ \frac{c^T x_0 - (b - Hy^0)^T y_k + \rho_k e^T z_k}{\rho_k - \|x^0\|_\infty} \\ &= \ \frac{(c^T x^0 + (y^0)^T H y_k - \frac{1}{2} y_k^T H y_k) - (b^T y_k - \frac{1}{2} y_k^T H y_k) + \rho_k e^T z_k}{\rho_k - \|x^0\|_\infty} \\ &\leq \ \frac{M - f(y_k) + \rho_k e^T z_k}{\rho_k - \|x^0\|_\infty} =: \nu_k, && (3.36) \end{aligned}$$

so that

$$\|z_k\|_\infty \leq \nu_k.$$

Hence, in order to show that $\{z_k\}$ is bounded, it suffices to prove that $\{\nu_k\}$ is bounded. We show next that $\nu_{k+1} \leq \nu_k$, $\forall k \geq k_0$. Since ν_k is nonnegative for all k, this proves the boundedness of $\{\nu_k\}$.

In view of (3.29) and (3.36), it is implied from Lemma 3.1 that

$$\nu_k = \frac{M - f(y_k) + \rho_k e^T z_k}{\rho_k - \|x^0\|_\infty} \geq \frac{M - f(y_{k+1}) + \rho_k e^T z_{k+1}}{\rho_k - \|x^0\|_\infty}, \forall k \geq k_0. \tag{3.37}$$

On the other hand,

$$\nu_{k+1} = \frac{M - f(y_{k+1}) + \rho_{k+1} e^T z_{k+1}}{\rho_{k+1} - \|x^0\|_\infty}.$$

Since $\rho_{k+1} \geq \rho_k$, in order to conclude that $\nu_k \geq \nu_{k+1}$ for $k \geq k_0$, it is sufficient to verify that the function g given by

$$g(\rho) := \frac{M - f(y_{k+1}) + \rho e^T z_{k+1}}{\rho - \|x^0\|_\infty}$$

has a nonpositive derivative $g'(\rho)$ for all ρ satisfying (3.29). Note that

$$\begin{aligned} M - f(y_{k+1}) + \|x^0\|_\infty e^T z_{k+1} &\geq c^T x^0 + (y^0)^T H y_{k+1} - \frac{1}{2} y_{k+1}^T H y_{k+1} - f(y_{k+1}) + \|x^0\|_\infty e^T z_{k+1} \\ &= c^T x^0 + (y^0)^T H y_{k+1} - b^T y_{k+1} + \|x^0\|_\infty e^T z_{k+1} \\ &= (x^0)^T c - (x^0)^T A^T y_{k+1} + \|x^0\|_\infty e^T z_{k+1} \\ &\geq -(x^0)^T z_{k+1} + \|x^0\|_\infty e^T z_{k+1} \\ &\geq 0, \end{aligned}$$

where the first inequality comes from (3.34) with k replaced by $k+1$, the first equality from the substitution of $f(y_{k+1})$ by $b^T y_{k+1} - \frac{1}{2} y_{k+1}^T H y_{k+1}$, the second one from (3.28), the second inequality from (3.30) and the non-negativeness of x^0, and the last one from (3.31). It follows from (3.29) that

$$g'(\rho) = -\frac{M - f(y_{k+1}) + \|x^0\|_\infty e^T z_{k+1}}{(\rho - \|x^0\|_\infty)^2} \leq 0,$$

proving $\nu_k \geq \nu_{k+1}$ and hence the boundedness of $\{z_k\}$.

□

Hence, if ρ_k goes to infinity as k goes to infinity, (3.12) must happen infinitely many times. The following lemma (corresponding to Lemma 2.5) studies the boundedness property of iterate sequences on K_ρ. (Recall that K_ρ denotes the index sequence on which ρ_k is updated; see (2.41).) Compared to Lemma 2.5, this lemma additionally shows the boundedness of sequence $\psi_k := (y^0 - y_k)^T H(y^0 - \tilde{y}_k)$ for the given vector y^0, without which it is impossible to derive the boundedness of dual sequences $\{s_k\}$ and $\{z_k\}$; see (3.49) below. The sequence ψ_k, again, vanishes in linear optimization because $H = 0$.

Lemma 3.3. *(Corresponds to Lemma 2.5) Suppose* (Pq) *is feasible. If $\rho_k \to \infty$, then $\{Z_k\tilde{u}_k\}$ and $\{S_k\tilde{x}_k\}$ are bounded on K_ρ. If additionally,* (Dq) *is feasible, then $z_k \to 0$ as $k \to \infty$, $k \in K_\rho$, and if, moreover,* (Pq) *is strictly feasible, then $\{y_k\}$ is bounded on K_ρ.*

Proof. Since (Pq) is feasible, and since ρ_k goes to infinity on K_ρ, Lemma 3.2 implies that (3.12) must be satisfied for all $k \in K_\rho$ large enough. In particular, there exists an integer k_0 such that for all $k \geq k_0$, $k \in K_\rho$,

$$\|[\Delta y_k; \Delta z_k]\| \quad \leq \quad \gamma_2, \tag{3.38}$$

and

$$\tilde{x}_k^{Q_k} \;\; \geq \;\; -\gamma_3 e. \tag{3.39}$$

In view of (3.25), inequality (3.38) implies that $\{\Delta s_k\}$ is bounded on K_ρ. Thus, with boundedness of x_k and u_k (enforced by Iteration IrQP) and boundedness of Δz_k and Δs_k on K_ρ, it follows from (3.6) and definitions (3.14)–(3.15) that for some $C > 0$ large enough,

$$\|Z_k \tilde{u}_k\| = \|U_k \Delta z_k\| \leq C, \quad k \geq k_0,\, k \in K_\rho, \tag{3.40}$$

and

$$\left\| S_k^{Q_k} \tilde{x}_k^{Q_k} \right\| = \left\| X_k^{Q_k} \Delta s_k^{Q_k} \right\| \leq C, \quad k \geq k_0,\, k \in K_\rho, \tag{3.41}$$

which proves the first claim that $\{Z_k \tilde{u}_k\}$ on K_ρ and from (3.14) that $\{S_k \tilde{x}_k\}$ are bounded on K_ρ.

Now, by assumption that (Pq)–(Dq) is primal-dual feasible, there exist (x^0, y^0, s^0) such that

$$A^T y^0 + s^0 = c, \quad Ax^0 = b - Hy^0, \quad [x^0; s^0] \geq 0. \tag{3.42}$$

Without loss of generality, we assume that $\rho_{k_0} > \|x^0\|_\infty$, so that

$$u_k^0 := \rho_k e - x^0 > 0, \text{ for } k \geq k_0. \tag{3.43}$$

On the other hand, in view of definitions (3.14), (3.15) and (3.16), equation (3.26) and the second to fourth block equations of (3.3) imply that, for all k,

$$A^T y_k - z_k + s_k = c,\; A\tilde{x}_k = b - H\tilde{y}_k,\; \tilde{x}_k + \tilde{u}_k = \rho_k e. \tag{3.44}$$

Equations (3.42)–(3.44) yield that

$$\begin{aligned} A^T(y^0 - y_k) + z_k + s^0 - s_k &= 0, && (3.45)\\ A(x^0 - \tilde{x}_k) &= -H(y^0 - \tilde{y}_k), && (3.46)\\ (x^0 - \tilde{x}_k) + (u_k^0 - \tilde{u}_k) &= 0. && (3.47) \end{aligned}$$

It follows that, for all k,

$$\begin{aligned} (s^0 - s_k)^T(x^0 - \tilde{x}_k) - z_k^T(u_k^0 - \tilde{u}_k) &= (s^0 - s_k + z_k)^T(x^0 - \tilde{x}_k)\\ &= -(y^0 - y_k)^T A(x^0 - \tilde{x}_k) \end{aligned}$$

where the first equality comes from (3.47) and the second one from (3.45). Hence, from (3.46), we obtain

$$(s^0 - s_k)^T(x^0 - \tilde{x}_k) - z_k^T(u_k^0 - \tilde{u}_k) = (y^0 - y_k)^T H(y^0 - \tilde{y}_k). \qquad (3.48)$$

It follows that

$$\begin{aligned} & (y^0 - y_k)^T H(y^0 - \tilde{y}_k) + s_k^T x^0 + z_k^T u_k^0\\ = \;& s_k^T \tilde{x}_k + z_k^T \tilde{u}_k - \tilde{x}_k^T s^0 + (x^0)^T s^0\\ \le \;& 2nC + \gamma_3 e^T s^0 + (x^0)^T s^0, \quad \forall k \in K_\rho, \qquad (3.49) \end{aligned}$$

where the equality comes from the expansion of (3.48), and the inequality from (3.40), (3.41) and (3.39). Now, inequality (3.27) with $y := y^0 - y_k$ and $p := \frac{1}{2}\Delta y_k$ implies that

$$\psi_k := (y^0 - y_k)^T H(y^0 - \tilde{y}_k) = (y^0 - y_k)^T H(y^0 - y_k) - (y^0 - y_k)^T H\Delta y_k \ge -\frac{1}{4}\Delta y_k^T H \Delta y_k$$

where the equality is from (3.16). Note from (3.38) that $\Delta y_k^T H \Delta y_k$ is bounded, so ψ_k is bounded from below. Hence, it follows from (3.49) that there exists $\pi > 0$ large enough such that

$$s_k^T x^0 \leq \pi \quad \forall k \in K_\rho, \tag{3.50}$$

and

$$z_k^i (u_k^0)^i \leq z_k^T u_k^0 \leq \pi \quad \forall i, \ \forall k \in K_\rho.$$

Using (3.43) and the fact that $\rho_k \to \infty$ as $k \to \infty$, it follows from positiveness of z_k and u_k^0 that

$$\lim_{k\to\infty} \sup_{k\in K_\rho} z_k^i \leq \frac{\pi}{\lim_{k\to\infty} \inf_{k\in K_\rho} (u_k^0)^i} = 0 \quad \forall i,$$

proving that z_k converges to zero on K_ρ.

Next, since H is a symmetric, semi-positive definite matrix, there exists a matrix L such that

$$H = L^T L,$$

so ψ_k can be written as

$$\psi_k = \|L(y_k + \frac{\Delta y_k}{2} - y^0)\|^2 - \frac{1}{4} \Delta y_k^T H \Delta y_k. \tag{3.51}$$

Since with $[u_k^0; x^0; z_k; s_k] \geq 0$ for $k \geq k_0$, we obtain from (3.49) that ψ_k is bounded from above, it follows from (3.51) and (3.38) that $\{Ly_k\}$ is bounded on K_ρ, and hence so is $\{Hy_k\}$. If (Pq) is strictly feasible, we can select $x^0 > 0$, and thus boundedness of $\{s_k\}$ on K_ρ follows from (3.50). In view of (3.26), together with boundedness of $\{z_k\}$ and $\{s_k\}$ on K_ρ, we have that $\{A^T y_k\}$ is bounded on K_ρ. With boundedness of $\{Hy_k\}$ and $\{A^T y_k\}$ on K_ρ in hand, full-rankness of $[H \ A]$ implies that $\{y_k\}$ is bounded on K_ρ, proving the last claim.

□

We next state a useful lemma (corresponding to Lemma 2.7) about the case when the sequence $\{\rho_k\}$ is unbounded. The proof procedure is similar to Lemma 2.7, but it is more complicated to show in Proposition A.1 that $\{[\tilde{x}_k; \tilde{u}_k]\}$ is bounded by some sequence (see (3.65) below). This cannot be proven in the same way as Lemma 2.4 because such lemma there implicitly requires the full rankness of A. The full rankness of A is not necessarily true for the quadratic optimization since we only assume the full-rankness of $[H \;\; A]$.

Lemma 3.4. *(Corresponds to Lemma 2.7) Suppose $\rho_k \to \infty$ as $k \to \infty$. If $\{y_k\}$ has a limit point on K_ρ and $\{z_k\}$ is bounded on K_ρ, then for any limit point $\{[y_*; z_*]\}$ of $\{[y_k; z_k]\}$ on K_ρ, there exists $\bar{x}_* \neq 0$ with $s_* = c - A^T y_* + z_* \geq 0$ and $z_* \geq 0$ such that*

$$A\bar{x}_* = 0, \tag{3.52}$$

$$Z_*(e - \bar{x}_*) = 0, \tag{3.53}$$

$$S_*\bar{x}_* = 0, \tag{3.54}$$

$$\bar{x}_* \geq 0. \tag{3.55}$$

Proof. Since $\{z_k\}$ is bounded on K_ρ, condition (3.11) will be violated eventually, and conditions (3.12) must be satisfied for all $k \in K_\rho$ large enough, i.e.,

$$\begin{aligned} \|[\Delta y_k; \Delta z_k]\| &\leq \gamma_2, \quad k \in K_\rho & (3.56)\\ \tilde{x}_k^{Q_k} &\geq -\gamma_3 e, \quad k \in K_\rho, & (3.57)\\ \tilde{u}_k^{Q_k} &\ngeq \gamma_4 e, \quad k \in K_\rho. & (3.58) \end{aligned}$$

Moreover, we have from (3.3) and (3.14)–(3.15) that for all k,

$$A\tilde{x}_k = A^{Q_k}\tilde{x}_k^{Q_k} = b - H(y_k + \Delta y_k), \tag{3.59}$$

$$\tilde{x}_k^{Q_k} + \tilde{u}_k^{Q_k} = \rho_k e, \tag{3.60}$$

$$\tilde{x}_k^i = 0,\ \tilde{u}_k^i = \rho_k, \quad \forall i \notin Q_k \tag{3.61}$$

$$A^T\Delta y_k - \Delta z_k + \Delta s_k = 0, \tag{3.62}$$

$$S_k^{Q_k}\tilde{x}_k^{Q_k} = -X_k^{Q_k}\Delta s_k^{Q_k}, \tag{3.63}$$

$$Z_k\tilde{u}_k = -U_k\Delta z_k. \tag{3.64}$$

Because Q_k can take only finitely many values, it follows from Proposition A.1 (see Appendix A) with

$$G := \begin{bmatrix} A^{Q_k} & 0 \\ I & I \end{bmatrix},\ D := \begin{bmatrix} S_k^{Q_k}(X_k^{Q_k})^{-1} & 0 \\ 0 & Z_k^{Q_k}(U_k^{Q_k})^{-1} \end{bmatrix},$$

and

$$v := \begin{bmatrix} \tilde{x}_k^{Q_k} \\ \tilde{u}_k^{Q_k} \end{bmatrix},\ g := \begin{bmatrix} b - H(y_k + \Delta y_k) \\ \rho_k e \end{bmatrix},\ w := \begin{bmatrix} \Delta s_k^{Q_k} \\ \Delta z_k^{Q_k} \end{bmatrix}$$

that there exists $C > 0$ such that

$$\|[\tilde{x}_k^{Q_k}; \tilde{u}_k^{Q_k}]\| \leq C\|[b - H(y_k + \Delta y_k); \rho_k e]\| \quad \forall k. \tag{3.65}$$

Since by assumption, there exists an infinite sequence $K \subseteq K_\rho$ such that $\{y_k\}$ is bounded on K, and since $\{\Delta y_k\}$ is bounded on K_ρ in view of (3.56), we have for some C' large enough

$$\|[\tilde{x}_k^{Q_k}; \tilde{u}_k^{Q_k}]\| \leq C'\rho_k, \quad k \in K.$$

Together with (3.61), we get that $\{[\bar{x}_k; \bar{u}_k]\}$ is bounded on K, where we have defined

$$\bar{x}_k = \frac{\tilde{x}_k}{\rho_k}, \quad \bar{u}_k = \frac{\tilde{u}_k}{\rho_k}. \tag{3.66}$$

Since both $\{y_k\}$ and $\{z_k\}$ are bounded on K, let $\{(y_*, z_*, s_*, \bar{x}_*, \bar{u}_*)\}$ be any limit point of $\{(y_k, z_k, s_k, \bar{x}_k, \bar{u}_k)\}$ on K with $z_* \geq 0$, $s_* = c - A^T y_* + z_* \geq 0$, and

$$\|\bar{x}_*\|_\infty = \|e - \bar{u}_*\|_\infty \geq 1, \tag{3.67}$$

where the equality comes from (3.60)–(3.61) and (3.66), and the inequality from (3.58). Next, since $\{x_k\}$ and $\{u_k\}$ are bounded by construction (Step 2 (ii) of Iteration IrQP), equations (3.61)–(3.64) imply from (3.62) and (3.56) that there exists $C'' > 0$ such that

$$\begin{aligned} s_k^i \tilde{x}_k^i &= 0, \quad i \notin Q & (3.68)\\ \|S_k^{Q_k} \tilde{x}_k^{Q_k}\| &= \|X_k^{Q_k} \Delta s_k^{Q_k}\| \leq C'' \quad k \in K, & (3.69)\\ \|Z_k \tilde{u}_k\| &= \|U_k \Delta z_k\| \leq C'' \quad k \in K. & (3.70) \end{aligned}$$

Dividing both sides of (3.59)–(3.61), (3.68)–(3.70) and (3.57) by ρ_k and taking limits on K, we conclude that $\bar{x}_* \neq 0$ (as for (3.67)) satisfies

$$\begin{aligned} A\bar{x}_* &= 0,\\ \bar{x}_* + \bar{u}_* &= e,\\ Z_*\bar{u}_* &= 0,\\ S_*\bar{x}_* &= 0,\\ \bar{x}_* &\geq 0, \end{aligned}$$

proving the claim.

□

The following theorem establishes that under the strict primal-dual feasibility,

ρ_k is increased at most finitely many time. The proof procedure is similar to and not harder than Proposition 2.2.

Theorem 3.1. *(Corresponds to Proposition 2.2) Suppose* (Pq)–(Dq) *is strictly feasible,*[2] *then ρ_k is increased at most finitely many times, i.e., K_ρ is finite.*

Proof. By contradiction, suppose K_ρ is infinite, i.e., $\rho_k \to \infty$ as $k \to \infty$. In view of Lemma 3.3, $\{y_k\}$ is bounded, and $\{z_k\} \to 0$ as $k \to \infty, k \in K_\rho$. Let y_* and z_* be the limit points of $\{y_k\}$ and $\{z_k\}$ on K_ρ, so $z_* = 0$, i.e., $y_* \in \mathcal{F}$. It follows from Lemma 3.4 that there exists $\bar{x}_* \neq 0$ that satisfies (3.52)–(3.55). In view of (3.54),

$$\bar{x}_* = 0, \quad \forall i \notin I(y_*). \tag{3.71}$$

Together with (3.52), we get

$$\sum_{i \in I(y_*)} \bar{x}_*^i a^i = 0.$$

Since the strict feasibility of (Dq) implies positive linear independence of vectors $\{a^i : i \in I(y_*), y_* \in \mathcal{F}\}$, it follows from (3.55) that

$$\bar{x}_* = 0, \quad \forall i \in I(y_*).$$

Together with (3.71), we have

$$\bar{x}_* = 0,$$

contradicting to that $\bar{x}_*$ is nonzero. □

Finally, if positive linear independence of $\{a^i : i \in I(y_*), y \in \mathcal{F}\}$ is replaced with the much stronger assumption of linear independence of $\{a^i : i \in I(y), y \in \mathbb{R}^m\}$,

[2] That (Pq) is strictly feasible is equivalent to that the solution set of problem (Dq) is nonempty and bounded (see Theorem 2.1 in [20]). Thus, our assumptions are the same as those in [32].

then boundedness of ρ_k follows without any feasibility assumption, as we state next. (In linear case, we used a similar result in Theorem 2.2, but we did not make it a proposition.)

Proposition 3.1. *Suppose that, at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set. If $\{y_k\}$ has a limit point on K_ρ and $\{z_k\}$ is bounded on K_ρ, then ρ_k is increased at most finitely many times.*

Proof. By contradiction, suppose $\rho_k \to \infty$ as $k \to \infty$. Lemma 3.4 then implies that for any limit point $z_* \geq 0$, $s_* \geq 0$ and y_* of sequences $\{z_k\}$, $\{s_k\}$ and $\{y_k\}$ with $s_* = c - A^T y_* + z_*$, there exists $\bar{x}_* \neq 0$ that satisfies (3.52)–(3.55). We can then conclude the proof with a contradiction argument, exactly as is in the proof of Theorem 3.1, except that the requirement of positive linear independence of $\{a^i : i \in I(y_*),\, y_* \in \mathcal{F}\}$ is replaced by the assumption of linear independence of $\{a^i : i \in I(y_*), y \in \mathbb{R}^m\}$. □

As in IrPDIP, if ρ_k is increased finitely many times, we denote the final value of ρ_k by $\bar{\rho}$.

3.2 Global convergence

In this section, we prove that, under the assumption that (Pq)–(Dq) is strictly feasible, any limit point of $\{y_k\}$ is an optimal solution of (Dq). To that end, we first show that $\{y_k\}$ is bounded when $\{\rho_k\}$ is bounded. In the linear case, the boundedness of $\{y_k\}$ directly follows from Proposition 2.1. Unfortunately, Proposition 2.1 does not hold for the more general algorithm IrQP. Such boundedness is then proven by mainly using a result of [23]; see Lemma 3.5 through Lemma 3.7. The remaining part shows the optimality of any limit point of the bounded sequence $\{y_k\}$, and its proof procedure is similar to the linear case (see section 2.3).

Lemma 3.5. *(Corollary 20, Page 94, [23]) If g_0, $-g_1$, $\cdots$, $-g_n$ are convex functions, and if the set of minima of problem*

$$\min\ g_0(y)\ \ s.t.\ \ g_i(y) \geq 0, \quad i = 1, \cdots, n \tag{3.72}$$

is a nonempty bounded set, then for every finite ξ_0, ξ_1, $\cdots$, ξ_n, the set

$$S_k = \{y|\ g_0(y) \leq \xi_0,\ g_i(y) \geq \xi_i,\ i = 1, \cdots, n\}$$

is a compact set (possibly empty).

To show the boundedness of $\{y_k\}$, we apply Lemma 3.5 to problem (Dq). Thus we need to show that the solution set of (Dq) is nonempty and bounded, so that the assumption of Lemma 3.5 is satisfied.

Lemma 3.6. *Suppose* (Dq) *is feasible. If* (Pq) *is feasible, then the solution set of* (Dq) *is nonempty; If* (Pq) *is strictly feasible, then it is also bounded.*

Proof. Since (Pq) is feasible, the objective function $f(y)$ of (Dq) is bounded from above by the weak duality theorem (see e.g. Proposition 6.2.2 of [11]). In view of the feasibility of (Dq), it follows from a 1956 result of Frank and Wolfe (see Appendix (i) in [25]) that (Dq) has a solution (see also Proposition 6.5.6 of [11]), proving the first claim. Next, we show the second claim. By assumption, (Pq) is strictly feasible, i.e., there exist $x^0 > 0$, $\xi > 0$ and y^0 such that $Hy^0 + Ax^0 - b\xi = 0$. It follows from

Tucker's theorem of the alternative (see e.g., section 2.4 in [37], with $B := [-A \;\; b]^T$, $C := 0$ and $D := H$) that there does *not* exist d such that

$$A^T d \leq 0, \quad Hd = 0, \quad b^T d \leq 0 \quad \text{with } [A \;\; b]^T d \neq 0.$$

Hence, every d that satisfies the first three conditions must satisfy $[A \;\; b]^T d = 0$ which, since $[H \;\; A]$ is full rank, implies that $d = 0$. So the solution set of (Dq) has no recession direction, proving that it is bounded. □

Lemma 3.7. *Suppose* (Pq) *is strictly feasible and* (Dq) *is feasible. If* ρ_k *is increased only finitely many times, then* $\{y_k\}$ *is bounded.*

Proof. Since ρ_k is increased only finitely many times, i.e., $\rho_k = \bar{\rho}$ for k large enough, Lemma 3.1 implies that $\{f(y_k) - \bar{\rho} e^T z_k\}$ is increasing for k large enough. Furthermore, Lemma 3.2 implies that $\{z_k\}$ is bounded. Thus, it follows that $f(y_k)$ is bounded below by some $\underline{f}$. In view of (3.25), boundedness of $\{z_k\}$ implies that the components of $\{c - A^T y_k\}$ are also bounded below by some $\underline{v}^i$ for $i = 1, \cdots, n$. Moreover, Lemma 3.6 implies that the solution set of (Dq) is nonempty and bounded. With all these in hand, it follows from Lemma 3.5 that the set

$$\underline{S} := \{y : f(y) \geq \underline{f},\; c - A^T y \geq \underline{v}\}$$

is bounded. Since $y_k \in \underline{S}$ for all k, $\{y_k\}$ is bounded. □

Lemma 3.8. *Suppose at every point* $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ *is a linearly independent set of vectors. If* $\{[y_k; z_k]\}$ *and* $\{\rho_k\}$ *are bounded, then every limit point of the bounded sequence* $\{[y_k; z_k]\}$ *is optimal for problem (Dq$_{\bar{\rho}}$).*

Proof. Since ρ_k is increased only finitely many times, after finitely many iterates, IrQP reduces to Algorithm A of [32] applied to $(Dq_{\bar{\rho}})$. We next check that all assumptions are satisfied that are needed in [32] on problem $(Dq_{\bar{\rho}})$, so that the claim follows by applying Theorem 2.2 of [32] to problem $(Dq_{\bar{\rho}})$.

The first two assumptions, that $\begin{bmatrix} H & 0 & A & 0 \\ 0 & 0 & -I & -I \end{bmatrix}$ is full rank (Assumption 2.1 of [32]) and that $(Dq_{\bar{\rho}})$ is strictly feasible (Assumption 2.2 of [32]), are trivially satisfied. Assumption 2.3 of [32] that the solution set of $(Dq_{\bar{\rho}})$ is nonempty and bounded is used only to get boundedness of $\{[y_k; z_k]\}$ (see Corollary A.5 in [32]), which is assumed in the statement of the present claim. Assumption 2.4 of [32] is equivalent to that at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linear independent set of vectors. □

Lemma 3.9. *Suppose at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set of vectors. If $\{[y_k; z_k]\}$ and $\{\rho_k\}$ are bounded, then there exists an infinite index set K on which $[\Delta y_k; \Delta z_k] \to 0$ as $k \in K \to \infty$, and $\lim_{k\to\infty} \inf_{k \in K} \tilde{x}_k \geq 0$.*

Proof. Proceeding by contradiction, suppose that

$$\inf\{\|[\Delta y_k; \Delta z_k]\|^2 + \|[\min(\tilde{x}_k, 0); \min(\tilde{u}_k, 0)]\|^2 : \forall k\} > 0. \tag{3.73}$$

Under our assumptions that $\{\rho_k\}$ and $\{[y_k; z_k]\}$ are bounded, it follows from Lemma A. 8 of [32] applied to $(Dq_{\bar{\rho}})$ that

$$[\Delta y_k; \Delta z_k] \to 0, \text{ as } k \to \infty. \tag{3.74}$$

In view of Lemma A. 7 of [32], (3.73) and (3.74) imply that any limit point of the bounded sequence $[y_k; z_k]$ is a non-optimal stationary point for problem $(Dq_{\bar{\rho}})$, contradicting to Lemma 3.8. □

Proposition 3.2. *Suppose at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set of vectors. If $\{[y_k; z_k]\}$ and $\{\rho_k\}$ are bounded, every limit point of the bounded sequence $\{y_k\}$ is optimal for* (Dq)*.*

Proof. Since ρ_k is constant for k large enough, conditions (3.12) must be violated eventually. In view of Lemma 3.9, there exists an infinite index set K on which

$$[\Delta y_k; \Delta z_k] \to 0 \text{ as } k \in K \to \infty, \tag{3.75}$$

and $\lim_{k \in K \to \infty} \tilde{x}_k \geq 0$, thus conditions (i) and (ii) in conditions (3.12) are satisfied on K. It follows that it must be condition (iii) of (3.12) that is eventually violated on K, i.e.,

$$\tilde{u}_k^{Q_k} \geq \gamma_4 e \text{ for } k \in K \text{ large enough.}$$

Note from the fourth row of (3.3) and from (3.15) that $\tilde{u}_k^{\overline{Q}_k} = \bar{\rho}$ when k is large enough, thus,

$$\tilde{u}_k \geq \min(\gamma_4, \bar{\rho})e \text{ for } k \in K \text{ large enough.} \tag{3.76}$$

Since $\{u_k\}$ is bounded as established in IrQP, and since for all k,

$$Z_k \tilde{u}_k = -U_k \Delta z_k,$$

it follows from (3.75) and (3.76) that

$$z_k \to 0, \quad \text{as } k \to \infty,\, k \in K. \tag{3.77}$$

In view of Lemma 3.8, this implies that every limit point of $\{f(y_k) - \rho_k e^T z_k\}$ on K is an optimal value of (Dq). Since Lemma 3.1 implies that $\{f(y_k) - \rho_k e^T z_k\}$ is monotonically increasing when $\rho_k = \bar{\rho}$ for k large enough, it follows that $\{f(y_k) - \rho_k e^T z_k\}$ converges to the optimal value of (Dq). Hence, any limit point of $\{y_k\}$ is an optimal solution of (Dq).

□

Theorem 3.2. *Suppose at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set of vectors. If* (Pq)–(Dq) *is strictly feasible, then ρ_k is increased at most finitely many times, $z_k \to 0$ as $k \to \infty$, and every limit point of the bounded sequence $\{y_k\}$ is optimal for* (Dq)*.*

Proof. Under the assumption that (Pq)–(Dq) is strictly feasible, Theorem 3.1 and Lemma 3.2 imply that $\{\rho_k\}$ and $\{z_k\}$ are bounded, respectively. It follows from Lemma 3.7 that $\{y_k\}$ is also bounded. The claim then follows immediately from Proposition 3.2. □

3.3 Analysis without primal or dual feasibility

This section provides an analysis result for Iteration IrQP without assuming the feasibility of (Pq) or (Dq). Such result can be shown with a similar proof procedure to that in linear optimization (see section 2.4), except for the additional Lemma 3.10 below (which is trivially true in the linear case because $H = 0$).

Let S_D denote the set

$$S_D := \{y : \exists x \text{ such that } Ax + Hy = b \text{ and } X(c - A^T y) = 0\},$$

where x and $c - A^T y$ are not necessarily nonnegative. Both the stationary set and optimal set of (Dq) are subsets of S_D.

Lemma 3.10. *$\{Hy : y \in S_D\}$ is bounded.*

Proof. We show that Hy is constant for all y that have the same active constraints. Since there are finitely many possible sets of active constraints, the claim will follow. To show that, suppose $y_1 \in S_D$ and $y_2 \in S_D$ associated with multipliers x_1 and x_2 respectively, have the same active constraints, i.e., $I(y_1) = I(y_2) = I$. By definition of S_D, y_1 and y_2 satisfy

$$Ax_1 + Hy_1 = b, \quad X_1(c - A^T y_1) = 0$$

and

$$Ax_2 + Hy_2 = b, \quad X_2(c - A^T y_2) = 0.$$

It follows that

$$A^{I(y_1)} x_1^{I(y_1)} + Hy_1 = b, \quad (A^{I(y_1)})^T y_1 = c^{I(y_1)}$$

and

$$A^{I(y_2)} x_2^{I(y_2)} + Hy_2 = b, \quad (A^{I(y_2)})^T y_2 = c^{I(y_2)}.$$

Since $I(y_1) = I(y_2) = I$, substraction of the above two groups of equations yields

$$A^I (x_1 - x_2)^I + H(y_1 - y_2) = 0, \quad (A^I)^T (y_1 - y_2) = 0.$$

Left-multiplying $(y_1 - y_2)^T$ to the first equation, using the second equation, we obtain

$$(y_1 - y_2)^T H (y_1 - y_2) = 0.$$

Since H is positive semi-definite, it follows that $H(y_1 - y_2) = 0$, proving the constant of Hy for all y with the same active constraints.

□

Theorem 3.3. *(Corresponds to Theorem 2.2) Suppose at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linear independent set of vectors. If ρ_k is increased finitely many times, then either of the following occurs:*

(a) $\{y_k\}$ is unbounded. In this case, (Pq) *is not strictly feasible or* (Dq) *is infeasible.*

(b) $\{y_k\}$ is bounded. In this case, every limit point of $\{y_k\}$ is optimal for (Dq)*; and* (Pq) *is feasible and* (Dq) *is strictly feasible.*

Proof. Case (a) follows from Lemma 3.7. In the case (b), since $\{\rho_k\}$ is increased finitely many times, condition (3.11) must be violated eventually, thus $\{z_k\}$ is bounded. It follows from Proposition 3.2 that every limit of the bounded sequence $\{y_k\}$ is an optimal solution of (Dq), implying both (Pq) and (Dq) are feasible. To show (Dq) is strictly feasible, it is equivalent to show there exists y^0 and $\xi > 0$ such that $c\xi - A^T y^0 > 0$. From Motzkin's theorem (page 29, [37]), it suffices to show that there doesnot exist $x^0 \neq 0$ such that $Ax^0 = 0$, $c^T x^0 = 0$ and $x^0 \geq 0$, for which the sufficient condition is that the solution set of (Pq) is bounded. We will next show the solution set of (Pq) is bounded indeed. Given any solution (x_*, y_*) of (Pq) with y_* as a solution of (Dq), from KKT conditions, it satisfies that $X_*(c - A^T y_*) = 0$ and $Ax_* + Hy_* = b$, which follows that

$$A^{I(y_*)} x_*^{I(y_*)} = b - Hy_*.$$

Because $\{a^i : I(y_*)\}$ is a linear independent set of vectors, it follows that

$$\|x_*\| = \|x_*^{I(y_*)}\| \leq \|(A^{I(y_*)})^T A^{I(y_*)}\|^{-1} \|(A^{I(y_*)})^T (b - Hy_*)\|. \tag{3.78}$$

Since from Lemma 3.10, Hy is bounded for all the optimal solutions y of (Dq), it follows that x_* must be bounded by some value independent of y_*. Because x_* is arbitrary, this proves the boundedness of the solution set of (Pq). □

The following lemma and theorem study this case when $\{\rho_k\}$ tends to infinity.

Theorem 3.4. *(Corresponds to Theorem 2.3) Suppose $\rho_k \to \infty$ as $k \to \infty$, then one of the cases must occur:*

(a) $\|y_k\| \to \infty$ *as* $k \to \infty$, $k \in K_\rho$. *In this case,* (Pq) *is not strictly feasible or* (Dq) *is infeasible.*

(b) $\{y_k\}$ *has a limit point on* K_ρ *and* $\{z_k\}$ *is unbounded on* K_ρ. *In this case,* (Pq) *is infeasible.*

(c) $\{y_k\}$ *has a limit point on* K_ρ, $\{z_k\}$ *is bounded on* K_ρ, *and there exists an infinite set* $K \subseteq K_\rho$ *such that* $\lim_{k\to\infty, k\in K} y_k = y_*$ *for some* $y_* \in \mathbb{R}^m$ *and* $\lim_{k\to\infty} \inf_{k\in K} \|z_k\| = 0$. *In this case,* (Dq) *is feasible but not strictly feasible.*

(d) $\{y_k\}$ *has a limit point on* K_ρ, $\{z_k\}$ *is bounded on* K_ρ, *and for any infinite set* $K \subseteq K_\rho$ *such that* $\lim_{k\to\infty, k\in K} y_k = y_*$ *for some* $y_* \in \mathbb{R}^m$, *it satisfies that* $\lim_{k\to\infty} \inf_{k\in K} \|z_k\| > 0$. *In this case,* (Dq) *is infeasible.*

Proof. Claim (a) follows from Lemma 3.3 and claim (b) from Lemma 3.2.

We next prove (c) and (d), for which we mainly use Lemma 3.4. We first show (c). Let $[y_*; z_*; s_*]$ be any limit point of $\{[y_k; z_k; s_k]\}$ on K with $z_* = 0$. Since for all k, $s_k \geq 0$, we have, using (3.26),

$$c - A^T y_* = s_* - z_* = s_* \geq 0, \tag{3.79}$$

i.e., y_* is a feasible point of (Dq). To show (Dq) is not strictly feasible, given any $y \in \mathbb{R}^m$, we have from (3.79)

$$-s_* + A^T(y - y_*) = A^T y - c. \tag{3.80}$$

From Lemma 3.4, there exists $\bar{x}_* \neq 0$ satisfying (3.52)–(3.55). Left-multiplying (3.80) by $\bar{x}_*^T$ yields

$$\bar{x}_*^T (A^T y - c) = 0 \quad \forall y \in \mathbb{R}^m,$$

where we have used equation (3.54) and (3.52). Since $\bar{x}_* \geq 0$ from (3.55), and since $\bar{x}_* \neq 0$, it follows that the set $\{y : A^T y \leq c\}$ has no strictly feasible point, proving the claim.

Next, we show (d). Let $[y_*; z_*; s_*]$ be any limit point of $\{[y_k; z_k; s_k]\}$, so (since $z_k \geq 0$, $\forall k$)

$$z_* \neq 0 \text{ with } z_* \geq 0. \tag{3.81}$$

To show (Dq) is infeasible, i.e., there does not exist y such that $A^T y \leq c$, by Farkas' Lemma, it suffices to show that there exists $\bar{x}_*$ satisfying

$$A\bar{x}_* = 0, \quad \bar{x}_* \geq 0, \quad c^T \bar{x}_* < 0.$$

Let $\bar{x}_*$ as guaranteed by Lemma 3.4 satisfy (3.52)–(3.55), then it remains to show $c^T\bar{x}_* < 0$, which can be derived as

$$c^T\bar{x}_* \;=\; (A^Ty_* - z_* + s_*)^T\bar{x}_* = -z_*^T\bar{x}_* = -e^Tz_* < 0,$$

where the first equality comes from (3.25), the second one from (3.52) and (3.54), the last one from (3.53), and the inequality from (3.81).

□

Theorem 3.4 summarizes the feasibility of (Pq)–(Dq) with respect to the behavior of Iteration IrQP. On the other direction, the following result gives the behavior of Iteration IrQP with respect to the feasibility of (Pq)–(Dq).

Theorem 3.5. *(Corresponding to Theorem 2.4) If for every $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set of vectors, then the following properties hold*

(a) *If* (Pq) *is feasible, then $\{z_k\}$ is bounded.*

(b) *If* (Dq) *and* (Pq) *are both strictly feasible, then $\{\rho_k\}$ is bounded, and every limit point of the bounded sequence $\{y_k\}$ is optimal for* (Dq)*.*

(c) *If* (Dq) *is not strictly feasible or* (Pq) *is infeasible, then $\{[y_k; z_k]\}$ is unbounded.*

(d) *If* (Dq) *is not strictly feasible and* (Pq) *is feasible, then $\{y_k\}$ is unbounded.*

Proof. Claim (a) is a restatement of Lemma 3.2, and claim (b) is a restatement of Theorem 3.2.

We show claim (c) by contradiction. Suppose $\{[y_k; z_k]\}$ is bounded. It follows from Proposition 3.1 that ρ_k is increased finitely many times. This implies from Theorem 3.3 (b) that (Dq) is strictly feasible and (Pq) is feasible, contradicting to the present assumption. In view of claim (a), claim (d) follows immediately from claim (c).

□

Remarks:

(1) The result of the quadratic optimization in section 3.1-3.3 is stronger than that of the linear case in section 2.3-2.4.

(2) See an ℓ_∞ version in Appendix C.

3.4 Numerical results

3.4.1 Implementation

Iteration IrQP was implemented in MATLAB R2009a for the tests in this chapter. All tests were run on a laptop machine (Intel R / 1.83G Hz, 1GB of RAM, Windows XP Professional 2002).

Parameters in the adaptive scheme (3.11)–(3.12) were set as $\sigma = 10$, $\gamma_1 = 100$, $\gamma_2 = 1$, $\gamma_3 = 100$, $\gamma_4 = 1$. We selected (typically infeasible) initial conditions y_0 for original problem (Dq) to be

$$y_0 := (AA^T)^{-1}Ac. \tag{3.82}$$

To force the strict feasibility for the penalized problem (Dq$_\rho$), the initial point z_0

was set to be

$$z_0 := \max(0, A^T y_0 - c) + \delta$$

where $\delta := 0.01$. Thus, the initial value of s_0, computed by

$$s_0 := c - A^T y_0 + z_0,$$

was strictly positive. The remaining components of x_0 and u_0 were selected so as to achieve perfect balance. First, we select the complementary measure μ_0 as

$$\mu_0 := g(\tilde{s}_0)^T g(\tilde{x}_0)/n$$

where

$$\tilde{s}_0 := c - A^T y_0, \quad \tilde{x}_0 := A^T(AA^T)^{-1}(b - Hy_0), \quad g(v) := v - \min(\min(v), 0)e.$$

The initial conditions of x_0 and u_0 were then computed by

$$x_0^i := \frac{\mu_0}{s_0^i}, \quad i = 1, \cdots, n$$

and

$$u_0^i := \frac{\mu_0}{z_0^i}, \quad i = 1, \cdots, n.$$

The penalty parameter was initialized with

$$\rho_0 := \|x_0\|_\infty. \tag{3.83}$$

We chose Q according to the most active rule, which selects the constraints that have smallest slacks. Analogously to [32], we terminated when the maximum number of iteration 100 is reached or the dual residual and the dual gap for (Pq_ρ)-(Dq_ρ)was less than a specific tolerance, i.e.,

$$\max\left\{\frac{\|b - Hy - Ax\|}{\max(\|H\|_\infty, \|A\|_\infty, \|b\|_\infty)}, \frac{\|\rho e - x - u]\|}{\max(1, \rho)}, \frac{s^T x + z^T u}{2n}\right\} < tol \tag{3.84}$$

where we used $tol = 10^{-6}$. (Because, by construction, all iterates are primal-feasible, the *primal* residual is always zero.)

We applied Iteration IrQP on randomly generated problems.

3.4.2 Randomly generated problems

We generated the standard linear problem of size $m = 50000$ and $n = 100$. Entries of matrix A are normally distributed with a zero mean and 0.01 covariance, denoted by $A \sim \mathcal{N}(0, 0.01)$, and vectors $b \sim \mathcal{N}(0, 1)$. Vector $c := A^T\xi + \eta$ where vector $\xi \sim \mathcal{N}(0, 0.01)$ and where vector η uniformly generated on $[0.05, 1.05]$ ($\eta \sim \mathcal{U}[0.05, 1.05]$), guaranteeing the dual (Dq$_\rho$) is strictly feasible. Let $\tilde{H} \sim \mathcal{U}[0, 10^\kappa]$ where κ is some randomly generated number on $[-1,\ 4]$ so the dual solution can have a large range, matrix H was then set to be $H = \tilde{H}\tilde{H}^T$ to make it positive semi-definite.

The average of CPU time and the number of iterations for 10 random problems are showed in Figure 3.1 with points y_0 (see (3.82)) infeasible for (Dq). For each random problem, we completely solved the same random problem many times with different fractions $|Q|/n$ of constraints that were kept. The rightmost point, corresponding to the fraction 1, is the result with all constraints ($|Q| = n$). As can be seen from Figure 3.1, the CPU time decreases as fractions $|Q|/n$ decreases, down to as little as 2% of constraints, while the number of iteration remains constant for a large range of fractions.

The initial penalty parameter by (3.83) was always too small to force feasibility of the iterates, so we observed it was always increased. The final values $\bar{\rho}$ were large enough from noting that the final values of z were zero.

We further compare the results by the adaptive penalty adjustment scheme and by the *fixed*-penalty scheme where the penalty parameter is not changed during the optimization process, with $|Q| = 10m$ (2% constraints) for both schemes. Figure 3.2 shows the average ratios of 20 problems between the CPU time and iterations obtained by different fixed values and those by the adaptive adjustment scheme. The value of fixed penalty values ranged from $10^{0.8}\rho_*$ to $10^{10}\rho_*$, where the threshold value $\rho_* := \|x_*\|_\infty$ and x_* is the solution of (Pq). (With a fixed penalty value greater than $\rho_* 10^{10}$, algorithm IrQP reached its limit iteration, so the corresponding result was not showed.) The blue (vertical) line corresponds the fixed value set to ρ_* $(\log_{10}(\frac{\rho_*}{\rho_*}) = 0)$, and the red (vertical) line corresponds to the final value $\bar{\rho}$. At those fixed penalty values that have ratios above the horizontal magenta line (denoting value 1), the corresponding CPU time and iterations were more than those by the adaptive scheme. We can see that the adaptive scheme was faster than the fixed scheme with most penalty values.

Figure 3.3 and 3.4 respectively show the average speedup gain of CPU time and iteration counts of 100 problems with different *nonzero* ranks r_H of the Hessian matrix H, where $\tilde{H} \sim \mathcal{N}(m, r_H)$. As can be seen, constraint reduction gains a speed-up with almost all sizes of $|Q|$ for those problems. Moreover, problems with a higher rank of H obtain a larger speed-up, which might due to the less number of active constraints at the solution. It is interesting that with 10% more constraints, the speedup gain seems independent of the rank r_H of the Hessian matrix H.

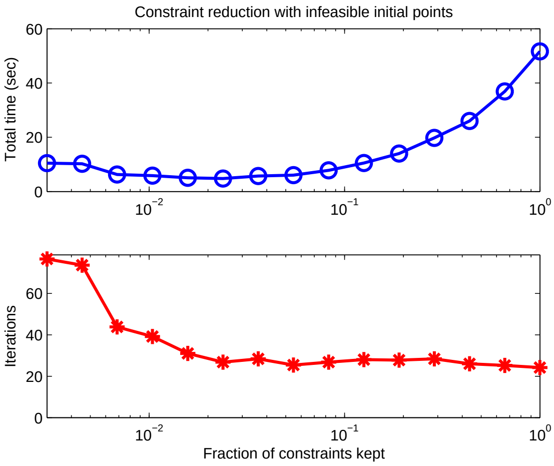


Figure 3.1: CPU time and iterations with infeasible initial points.

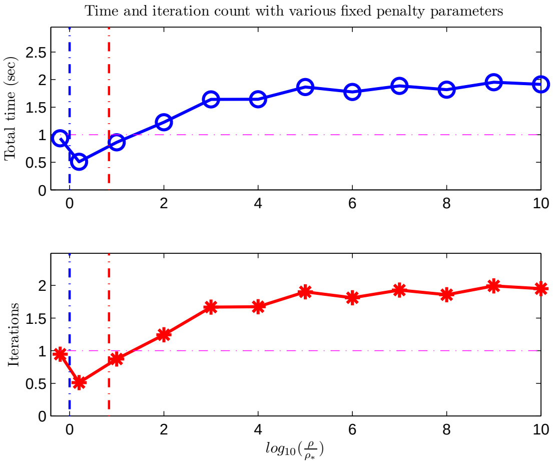


Figure 3.2: Behaviors for various values of fixed penalty parameters ρ with $\frac{|Q|}{n} = 2\%$.

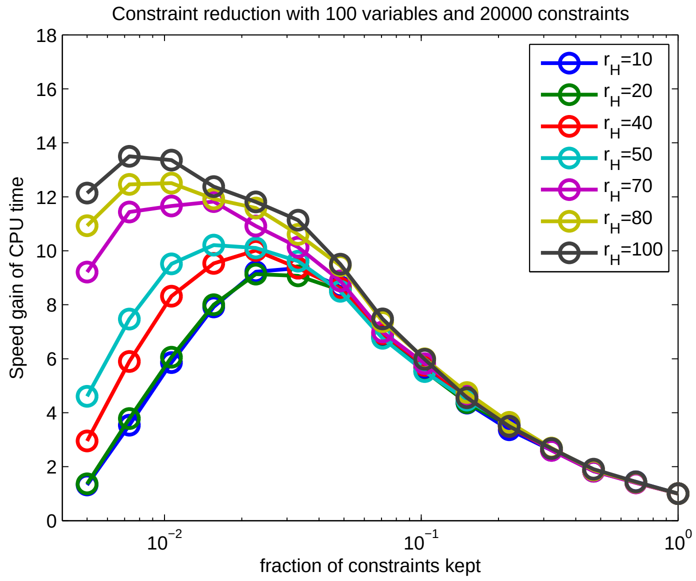


Figure 3.3: The speedup gain of CPU time with different ranks of Hessian H.

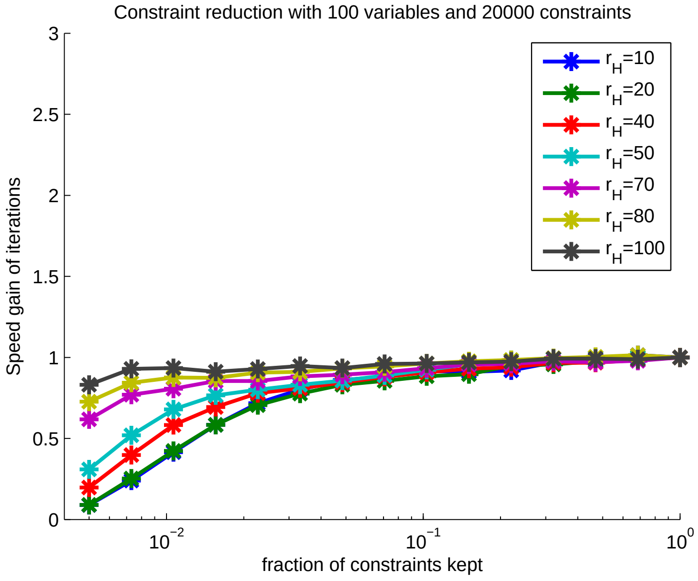


Figure 3.4: The speedup gain of iteration counts with different ranks of Hessian H.

3.5 Conclusion

Our contribution for constraint reduction in convex quadratic optimization allows for the use of infeasible initial points. In most cases, feasible initial points are not readily available. To address that, we extended the adaptation scheme proposed in Chapter 2 for the penalty parameter. We have shown that under the sole assumption of the strict feasibility of the primal-dual pair, the penalty parameter is increased only finitely many times. We have further proven that under a further non-degeneracy assumption, the algorithm is convergent to an optimal KKT point. Numerical results on randomly generated problems show that the proposed algorithm has obtained a major speed-up.

Chapter 4

Application to Model Predictive Control (RHC)

The work in this chapter is joint with Aaron Greenfield and Vineet Sahasrabudhe from Sikorsky Aircraft Corporation.

4.1 Introduction

Model Predictive Control (also known as Receding Horizon Control (RHC))[1] has been highly successful in process control applications; see, e.g., [13, 34, 48]. Its broader use has been hindered by its high computational requirements: determination of the control value to be applied at the next sample time involves the solution of an optimization problem with many practical constraints, which is slightly different from time step to time step. (See, e.g. [26, 36], for background on constrained model-predictive control.) In particular, few studies of the possible use of RHC in aerospace applications have been conducted. Exceptions include [38], where RHC-based control for spacecraft formation keeping and attitude control is studied, [63] where RHC is used in conjunction with a neural network feedback controller, to

[1] We will mostly refrain from using the acronym MPC, ubiquitous in the model predictive control literature. This is because, by an unfortunate coincidence, that same acronym is commonly used with a different meaning—Mehrotra's Predictor Corrector algorithm, also briefly considered in the present paper—in the interior-point optimization literature. Indeed the present thesis is targeted at both audiences.

control a six-degree-of-freedom model of an autonomous helicopter, and [33] where RHC is successfully applied in flight tests on unmanned aerial vehicles.

Here we consider model predictive control of a rotorcraft, based on a linear time-invariant model of the rotorcraft and on the minimization of a linear or convex quadratic cost function subject to bound constraints on the control and state variables and their derivatives. Thus, following much of the RHC literature, the optimization problem to be solved on-line at every time step is an LP or a CQP. Accordingly, it requires the online solution of LPs or CQPs. While this can be obtained by evaluating the associated explicit solution [6, 4, 2, 28, 5], we focus in this chapter on using an online LP-solver or CQP-solver.

The optimization problem in RHC as a practical example has two special properties. First, constraints far outnumber decision variables. Second, a difficulty is that constraint reduction requires the availability, for each optimization problem (to be solved on-line), of an initial strictly feasible point. But such points may not be readily available in the model-predictive control context. These two properties make model-predictive control particularly suitable for our algorithms. We solve this optimization problem by means of IrMPC in linear RHC and by IrQP in quadratic RHC. The novelty of our contribution lies in our speeding up the optimization by exploiting the structure of the problem in two ways: (i) we incorporate a “constraint reduction” scheme, which takes advantage of the much larger number of inequality constraints than optimization variables; and (ii) we use an exact penalty function technique to address the need of a strictly feasible “warm start” at each time step.

The remainder of this chapter is organized as follows. In section 4.2, we consider linear RHC, and report numerical results obtained by Iteration IrMPC for the altitude control of a rotorcraft. Section 4.3 is devoted to quadratic RHC, which is applied to both altitude control and trajectory following for rotorcrafts. Numerical results are shown that demonstrate the efficiency of IrQP. Some conclusions are given in section 4.4.

All algorithms were implemented in Matlab. Tests in section 4.2 and 4.3.1 were run on a laptop (Intel (R) / 1.83G Hz, 1GB of RAM, Windows XP Professional 2002) with Matlab R2009a. Tests in section 4.3.2 that involved a larger model were run on a faster desktop machine (Intel(R) Core(TM)i5-2400 CPU @ 3.10G Hz, 4GB of RAM, Windows 7 Enterprise 2009) with Matlab R2010b.

4.2 Linear RHC

Linear model predictive control, consisting of linear constraints and a linear objective function, is popular since the '70s. Solving linear optimization problems for RHC has been previously considered in [44, 49, 21]. In this section, we apply Iteration IrMPC to reduce constraints and hence speed up the LP optimization solving in RHC process.

4.2.1 Problem setup

Given the measured state of the system at every (discrete) time t, (ℓ_∞) linear RHC solves an optimization problem during time interval $(t-1, t)$ such as the following:

$$\min_{\boldsymbol{w},\boldsymbol{\theta}} \quad \sum_{k=0}^{M-1} \|Rw_k\|_\infty + \sum_{k=1}^{N} \|P\theta_k\|_\infty \tag{4.1}$$

s.t.

$$\theta_{k+1} = A_s\theta_k + B_s w_k, \tag{4.2}$$

$$\theta_0 = \theta(t-1), \tag{4.3}$$

$$\theta_{\min} \leq \theta_k \leq \theta_{\max}, \quad \text{for } k = 1, \cdots, N \tag{4.4}$$

$$w_{\min} \leq w_k \leq w_{\max}, \quad \text{for } k = 0, \cdots, M-1 \tag{4.5}$$

$$\delta w_{\min} \leq w_k - w_{k-1} \leq \delta w_{\max}, \quad \text{for } k = 0, \cdots, M-1 \tag{4.6}$$

$$w_k = 0, \quad \text{for } k = M, \cdots, N-1 \tag{4.7}$$

with $R \in \mathbb{R}^{r\times r}$, $P \in \mathbb{R}^{p\times p}$, $A_s \in \mathbb{R}^{p\times p}$ and $B_s \in \mathbb{R}^{p\times r}$; vectors $\theta_k \in \mathbb{R}^p, w_k \in \mathbb{R}^r$ respectively denote the state and the control input at time step k ahead of the current time. Positive integers M and N are respectively control horizons and prediction horizons, with usually $M < N$. Equation (4.2) is a simplified, LTI physical system being controlled, not accounting for the (unknown) future perturbations, such as wind gusts; $\theta(t-1)$, the measured state of the rotorcraft at time $t-1$, does of course reflect past perturbations, and is factored in by means of (4.3). Parameters $\theta_{\min}$, $\theta_{\max}$, $w_{\min}$, $w_{\max}$, $\delta w_{\min}$ and $\delta w_{\max}$ are given bounds,[2] and some of them can be set to $\pm\infty$. Constraints (4.6) restrict the rate of change of w. The optimization

[2] In general, $w_{\min}$ and $w_{\max}$ can change over time due to hardware failures or externally caused damage. Such a situation can be accounted for with no difficulties; this is one of the payoffs of model predictive control.

variables are the control sequence and state sequence, respectively denoted by

$$\boldsymbol{w} = [w_0; \cdots ; w_{M-1}] \in \mathbb{R}^{Mr}, \;\; \boldsymbol{\theta} = [\theta_1; \cdots ; \theta_N] \in \mathbb{R}^{Np}. \tag{4.8}$$

After problem (4.1)–(4.7) is solved, yielding the optimal control sequence $[w_0^*; \cdots ; w_{M-1}^*]$, only the first step $w_0^* =: w(t-1)$ of the sequence is applied as control input (at time t). The main computational task is to solve (4.1)–(4.7).

We can convert problem (4.1)–(4.7) into a standard dual linear program. First, introduce additional nonnegative optimization variables $[\epsilon_{w_0}, \cdots , \epsilon_{w_{M-1}}, \epsilon_{\theta_1}, \cdots , \epsilon_{\theta_N}]^T \in \mathbb{R}^{M+N}$ such that

$$Rw_k - \epsilon_{w_k} e \leq 0, \; -Rw_k - \epsilon_{w_k} e \leq 0, \; k = 0, \cdots , M-1, \tag{4.9}$$

$$P\theta_k - \epsilon_{\theta_k} e \leq 0, \; -P\theta_k - \epsilon_{\theta_k} e \leq 0, \; k = 1, \cdots , N. \tag{4.10}$$

Hence, minimizing the objective function of (4.1) is equivalent to minimizing $\epsilon_{w_0} + \cdots + \epsilon_{w_{M-1}} + \epsilon_{\theta_1} + \cdots + \epsilon_{\theta_N}$ with additional constraints (4.9)–(4.10). Second, states θ_k can be expressed explicitly in terms of w_k from constraints (4.2)–(4.3) as

$$\theta_k = A_s^k \theta_0 + \sum_{i=0}^{k-1} A_s^i B_s w_{k-1-i}, \;\; k = 1, \cdots , N,$$

or equivalently in the matrix form

$$\boldsymbol{\theta} = \Gamma \boldsymbol{w} + \Omega \theta_0, \tag{4.11}$$

where

$$\Gamma := \begin{bmatrix} B_s & 0 & \cdots & 0 & 0 \\ A_sB_s & B_s & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_s^{M-1}B_s & A_s^{M-2}B_s & \cdots & A_sB_s & B_s \\ A_s^MB_s & A_s^{M-1}B_s & \cdots & A_s^2B_s & A_sB_s \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_s^{N-1}B_s & A_s^{N-2}B_s & \cdots & \cdots & A_s^{N-M}B_s \end{bmatrix}, \quad \Omega := \begin{bmatrix} A_s \\ A_s^2 \\ \cdots \\ A_s^N \end{bmatrix}.$$

Hence, problem (4.1)-(4.7) can be rewritten into the following standard dual linear program

$$\max_{\boldsymbol{w},\, \boldsymbol{\epsilon_w},\, \boldsymbol{\epsilon_\theta}} \quad -(\epsilon_{w_0} + \cdots + \epsilon_{w_{M-1}} + \epsilon_{\theta_1} + \cdots + \epsilon_{\theta_N}) \tag{4.12}$$

$$s.t. \quad w_{\min}e \le \boldsymbol{w} \le w_{\max}e \tag{4.13}$$

$$\theta_{\min}e - \Omega\theta_0 \le \Gamma\boldsymbol{w} \le \theta_{\max}e - \Omega\theta_0 \tag{4.14}$$

$$\delta w_{\min} \le w_k - w_{k-1} \le \delta w_{\max} \quad \text{for } k = 0, \cdots, M-1 \tag{4.15}$$

$$\mathbf{R}\boldsymbol{w} - \boldsymbol{\epsilon_w} \le 0, \tag{4.16}$$

$$\mathbf{R}\boldsymbol{w} + \boldsymbol{\epsilon_w} \ge 0, \tag{4.17}$$

$$\mathbf{P}\Gamma\boldsymbol{w} - \boldsymbol{\epsilon_\theta} \le -\mathbf{P}\Omega\theta_0, \tag{4.18}$$

$$\mathbf{P}\Gamma\boldsymbol{w} + \boldsymbol{\epsilon_\theta} \ge -\mathbf{P}\Omega\theta_0 \tag{4.19}$$

where

$$\boldsymbol{\epsilon_w} := [\epsilon_{w_0}e; \cdots; \epsilon_{w_{M-1}}e] \in \mathbb{R}^{Mr}, \ \boldsymbol{\epsilon_\theta} := [\epsilon_{\theta_1}e; \cdots; \epsilon_{\theta_N}e] \in \mathbb{R}^{Np}$$

and

$$\mathbf{R} := \text{diag}\{R, R, \cdots, R\} \in \mathbb{R}^{Mr\times Mr}, \;\; \mathbf{P} := \text{diag}\{P, P, \cdots, P\} \in \mathbb{R}^{Np\times Np}.$$

When all states and control inputs are constrained by bounds, problem (4.12)–(4.19) has $Mr + M + N$ variables and $6Mr + 4Np$ constraints. Since usually $p > r$ and $N > M$, the number of constraints is many more than that of variables, and hence, constraint reduction is likely to be beneficial.

4.2.2 Physical model and data

We use the model described in [24], an eight-state LTI model of the vertical axis dynamics of a rotorcraft, including engine states, which allows for a reasonably accurate simulation, and includes fuel flow and rotor speed. The eight states (all in delta coordinates, i.e., difference from trim values) are the rotor speed (27 rad/sec), yaw rate (1.7 deg/sec), body-axis vertical velocity measured positively downwards (−1.7 ft/sec), inflow velocity (0 rad/sec), shaft torque (33000 ft×lb), engine fuel flow (0.11 lb/sec), integral of the delta-coordinate rotor speed, and altitude (difference from the target altitude) measured positively upwards.[3] The values listed within parentheses are the trim values. We use only one input, the collective control, with a trim value of 6.5 inches. For simulation purpose, we discretize the model with sample period $T_\text{s} = 0.01$ sec, yielding the discrete-time model

$$\theta(t) = A_s\theta(t-1) + B_s w(t-1), \tag{4.20}$$

[3] Hence the vertical velocity is the *negative* of the derivative of the altitude. This follows standard practice in the rotorcraft industry.

where A_s and B_s are given in Appendix A.2.

In our data setting, the weights R and Q in the RHC cost function (4.1) are taken to be $R = 0.1$ and $Q = \mathrm{diag}(0,0,1,0,0,0,0,0,1)$. We choose $M = 30$ and $N = 100$. Further, we set $w_{\max} = 3.5$ inches, $w_{\min} = -6.5$ inches (in delta coordinates, corresponding to a range of $[0, 10]$ inches in absolute coordinates), $\delta w_{\max} = -\delta w_{\min} = 0.02$ inches (corresponding to 2 inches per sec),
$\theta_{\max} = [\infty; \infty; 21.7 \text{ ft/sec}; \infty; 22000 \text{ ft}\times\text{lb}; \infty; \infty; \infty]$,
and $\theta_{\min} = -[\infty; \infty; 33.3 \text{ ft/sec}; \infty; 16000 \text{ ft}\times\text{lb}; \infty; \infty; \infty]$; the listed bounds on the body-axis vertical velocity and shaft torque, both in delta coordinates, correspond to the ranges $[-35, 20]$ ft/sec and $[17000, 55000]$ ft×lb, respectively, in absolute coordinates. The bounds on w are collective stick input limits, those on δw and on the body-axis vertical velocity reflect pilot preference, and those on shaft torque are notional gearbox limits.

4.2.3 Warm starts and exact penalty function

Because, clearly, the optimization problem to be solved at a given time step is typically closely related to that solved at the previous time step, use of a "warm start" is called for. We chose warm starts as follows. Given a vector $v = [v^1; \cdots; v^n]$, define $\bar{v} := [v^2; \cdots; v^n; v^n]$. Suppose at time $t-1$, the solution for program (4.12)–(4.19) is

$$[\boldsymbol{w}; \boldsymbol{\epsilon_w}; \boldsymbol{\epsilon_\theta}] := [w_*^0; \cdots; w_*^{M-1}; \epsilon_{w_*^0}; \cdots; \epsilon_{w_*^{M-1}}; \epsilon_{\theta_*^1}; \cdots; \epsilon_{\theta_*^N}],$$

then the initial points we used for the problem (4.12)–(4.19) at time t is

$$[\bar{\boldsymbol{w}}; \bar{\boldsymbol{\epsilon}}_{\boldsymbol{w}} + 0.01; \bar{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} + 0.01].$$

The initial penalty parameter ρ was set to $2\|x_*^t\|_\infty$, where x_*^t is the solution for the dual of (4.12)–(4.19). As for the next state $\theta(t)$, we generated it using the dynamics

$$\theta(t) = A_s\theta(t-1) + B_s w(t-1),$$

i.e., we assume for simplicity that the model used in the optimization is exact, and there are no perturbation. (Hence, the only possible cause for infeasibility of problem (4.12)–(4.19) is from the last element of $\bar{\boldsymbol{w}}$, $\bar{\boldsymbol{\epsilon}}_{\boldsymbol{w}}$ and $\bar{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$ in warm starts.)

Aside from the possible issue with setting the last block-component of the initial guess, in practice, strict feasibility is likely to fail due to modeling errors and exogenous perturbations (e.g., wind gusts), both of which will cause the new initial state at time t, to be different from that predicted at time $t-1$. Thus, the exact penalty function, allowing for infeasible warm starts, is necessary.

Even when the warm start given by the solution of the previous LP remains strictly feasible for the current problem (D), substituting problem (D$_\rho$) may speed up convergence. This is because the “old” solution may be close to the boundary of the feasible set and badly centered for the new problem (D). In such a situation, the interior-point method (e.g. [65]) may take many short steps, hindered by the proximity of constraint boundaries, before reaching the new solution. This difficulty is avoided with problem (D$_\rho$), which however, allows you to move outside the original feasible area, and thus increases the step-size. Hence, even with a strictly feasible initial point, the exact penalty problem can behave better than the original problem. See a closer discussion about this phenomenon in [10].

4.2.4 Simulation results

In the numerical tests reported below, we used the ℓ_1 version of IrMPC with parameters $\sigma = 10$, $\gamma_1 = 0.1$, $\gamma_2 = 100$ and $\gamma_4 = 100$ in (2.11)–(2.12). (The results obtained by the ℓ_∞ version are not much different, and hence omitted. In particular, the overhead difference in the CPU time between these two versions can be hardly observed for the problems solved here.)

We used a 10 second simulation window and assume that perturbations (in particular, wind gusts) occur only before the start of the simulation window and are reflected in the initial state $\theta(0) = [0; 0; 0; 0; 0; 0; 0; -40 \text{ ft}]$, which indicates a deviation from the target altitude. The control objective is to bring the rotorcraft's altitude back to nominal, under bound constraints on the collective control and its one-step variation as well as on the body-axis vertical velocity and shaft torque.

As the sample time is set to be $T_s = 0.01$ sec, there are 1000 time steps (the first one at 0.0 sec ($t - 1 = 0$), the last one at 9.99 sec ($t - 1 = 999$)), corresponding to 1000 LPs to solve. With $M = 30$ and $N = 100$, each LP has 160 variables and 1180 inequality constraints. Figure 4.9 shows the CPU time to solve the 10 sec (RHC) process (corresponding to 1000 LPs) with $|Q| = 300$ and $|Q| = n = 1180$ (corresponding to the case without constraint reduction). In order to keep figures readable, only every 10th time step is showed. Note that solving every LP with constraint reduction (in red circles) takes nearly or less than half of the time it takes without constraint reduction (in magenta triangles). Because not all constraints of (4.13)–(4.19) are dense, constraint reduction did not afford a full fourfold ($\frac{1180}{300}$) speedup factor.

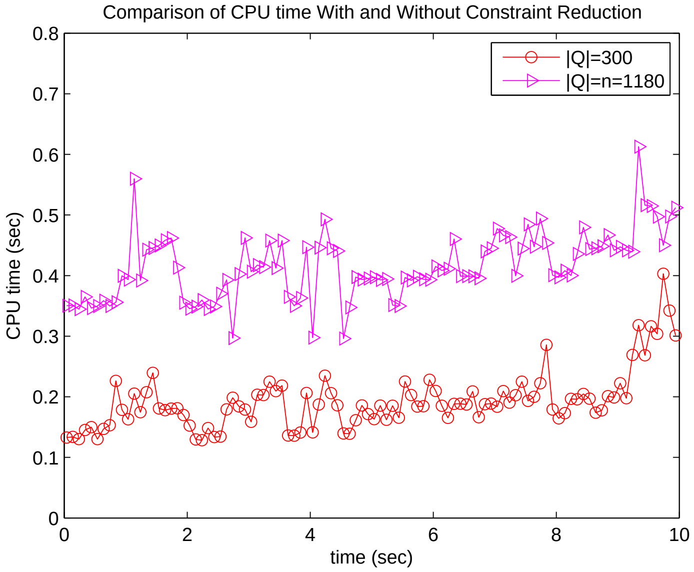


Figure 4.1: CPU time in seconds for $|Q| = 300$ and $|Q| = n = 1180$.

In particular, we show in Figure 4.2 the effect of constraint reduction on the single LP at time 5 sec ($t - 1 = 499$), which is a typical case. The CPU time to completely solve this problem is decreasing as the number of constraints is decreasing, all the way from 1180 till $|Q|$ is as small as 200, approximately 17% of the total constraints. For this LP, MPC takes much more time and iterations than Iteration IrMPC.

Table 4.1 shows that 516 of the 1000 LPs begin with warm starts that are not strictly feasible points (NFIPs), the remaining 484 start with strictly feasible initial points (FIPs). Because we used a warm start for the initial penalty parameter, only

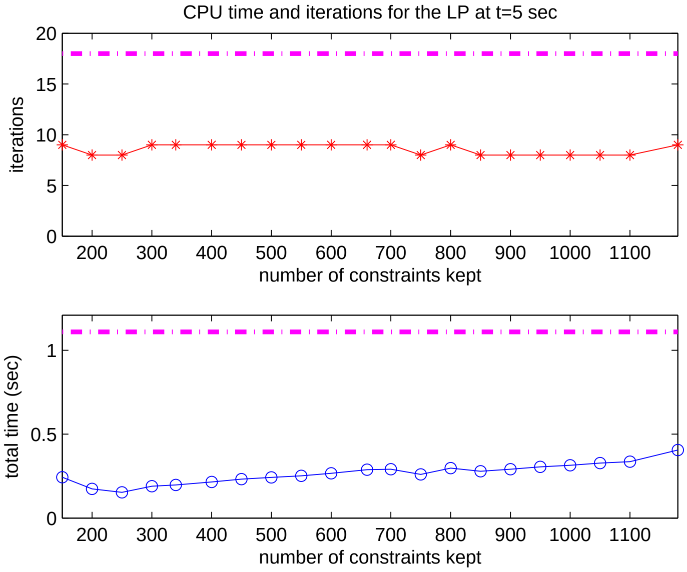


Figure 4.2: CPU time and the number of iterations to solve the problem at 5 sec by IrMPC with varying number $|Q|$ of kept constraints; see blue circles and red stars. MPC takes much larger to solve this problem; see the dashed magenta line.

5 problems start with too small initial penalty parameters (SIPPs), and we observed an increase of penalty parameter for those 5 problems only. For those 484 problems with feasible initial points, rMPC* in [65] is applicable, so we compared the time of algorithm IrMPC and rMPC* with $|Q| = 300$. It turned out that 95% of them (that is, 461 problems out of total 484 problems) take less time using IrMPC than using rMPC*, presumably due to the better ability of IrMPC to handle initial points close to the constraint boundaries.

4.3 Quadratic RHC

Quadratic model predictive control consists of linear constraints and a convex quadratic objective function. In this section, we apply Iteration IrQP in Chapter 3

Table 4.1: Number of problems with certain properties

NFIPs	FIPs	SIPPs	IrMPC < rMPC*
516	484	5	461 out of 484

to the quadratic RHC for rotorcraft altitude control and for rotorcraft trajectory following.

4.3.1 Altitude control

4.3.1.1 Problem setup

Suppose we have the initial state $\theta(t-1)$, usually measured by on-line sensors. In RHC-based altitude control, during time interval $(t-1,t)$, model predictive control solves the minimization problem below

$$\min_{\boldsymbol{w},\boldsymbol{\theta}} \quad \sum_{k=0}^{M-1} w_k^T R w_k + \sum_{k=1}^{N} \theta_k^T P \theta_k \tag{4.21}$$

s.t.

$$\theta_{k+1} = A_s\theta_k + B_s w_k, \tag{4.22}$$

$$\theta_0 = \theta(t-1), \tag{4.23}$$

$$w_{min} \leq w_k \leq w_{max}, \quad \text{for } k = 0, \cdots, M-1 \tag{4.24}$$

$$\theta_{min} \leq \theta_k \leq \theta_{max}, \quad \text{for } k = 1, \cdots, N \tag{4.25}$$

$$\delta w_{min} \leq w_k - w_{k-1} \leq \delta w_{max}, \quad \text{for } k = 0, \cdots, M-1 \tag{4.26}$$

$$w_k = 0, \quad \text{for } k = M, \cdots, N-1. \tag{4.27}$$

It is the same to the linear RHC problem (4.1)–(4.7) except that a quadratic cost function is used.

Define the matrix

$$E_r := \begin{bmatrix} I_r & 0 & \cdots & 0 \\ -I_r & I_r & \cdots & 0 \\ & \ddots & \ddots & \\ 0 & \cdots & -I_r & I_r \end{bmatrix} \in \mathbb{R}^{Mr \times Mr},$$

and the vectors

$$\begin{aligned}
\boldsymbol{\theta}_{\max} &:= [\theta_{\max}; \ldots; \theta_{\max}] \in \mathbb{R}^{Np}, \\
\boldsymbol{\theta}_{\min} &:= [\theta_{\min}; \ldots; \theta_{\min}] \in \mathbb{R}^{Np}, \\
\boldsymbol{w}_{\max} &:= [w_{\max}; \ldots; w_{\max}] \in \mathbb{R}^{Mr}, \\
\boldsymbol{w}_{\min} &:= [w_{\min}; \ldots; w_{\min}] \in \mathbb{R}^{Mr}, \\
\delta\boldsymbol{w}_{\max} &:= [\delta w_{\max}; \ldots; \delta w_{\max}] \in \mathbb{R}^{Mr}, \\
\delta\boldsymbol{w}_{\min} &:= [\delta w_{\min}; \ldots; \delta w_{\min}] \in \mathbb{R}^{Mr}.
\end{aligned}$$

Variables $\boldsymbol{\theta}$ (see definition (4.8)) can be similarly eliminated by substituting (4.11) into (4.21) and (4.25), thus, the RHC problem can be transformed into a CQP problem in the standard form (Dq) with only $\boldsymbol{w}$ as variables where

$$b := -\Gamma^T \mathbf{P} \Omega \theta(t-1), \; H := \mathbf{R} + \Gamma^T \mathbf{P} \Gamma$$

and

$$A := -\begin{bmatrix} -I_{Mm} \\ I_{Mm} \\ -\Gamma \\ \Gamma \\ -E_r \\ E_r \end{bmatrix}^T, \quad c := -\begin{bmatrix} -\boldsymbol{w}_{\max} \\ \boldsymbol{w}_{\min} \\ -\boldsymbol{\theta}_{\max} + \Omega\theta(t-1) \\ \boldsymbol{\theta}_{\min} - \Omega\theta(t-1) \\ -(\delta\boldsymbol{w}_{\max} + E_r^1 w(t-1)) \\ \delta\boldsymbol{w}_{\min} + E_r^1 w(t-1) \end{bmatrix} \tag{4.28}$$

where $E_r^1 := [I_r; 0; \ldots; 0] \in \mathbb{R}^{Mr \times r}$.

The number of variables in the transformed CQP is Mr and, when all the state and input variables are constrained (all components of $\theta_{\min}, \theta_{\max}, w_{\min}, w_{\max}, \delta w_{\min}, \delta w_{\max}$ are finite), the total number of constraints is $4Mr + 2Np$; i.e., the size of A in (4.28) is $Mr \times (4Mr + 2Np)$. In our numerical example, many state variables are unconstrained; still, the number of constraints is significantly (at least four times) larger than that of variables.

4.3.1.2 Simulation results

We use a 10 second simulation window on the same model and data as in section 4.2.2. The initial state is $\theta(0) = [0; 0; 0; 0; 0; 0; 0; -80 \text{ ft}]$, indicating an 80-foot deviation from the target altitude. The control objective is the same: bring the rotorcraft's altitude back to nominal. As in linear RHC, we assume the model in theonline CQP (see (4.22)) is "exact", i.e., RHC uses the same matrices A_s, B_s as in the optimization problem.

With a single scalar control input ($r = 1$), the CQP problem solved at each iteration has M variables. Only two of eight state variables have finite prescribed upper and lower bounds, which gives us $4N$ constraints from (4.25), $2N$ for each state variable. Given that there are also 4 constraints at every time step on the control input from (4.24) and (4.26), the total number of constraints in the CQP problem is $4M + 4N$. With our current choice of $M = 30$ and $N = 100$, we thus have 30 optimization variables and 520 constraints.

As the sample time is set to be $T_s = 0.01$ sec, there are 1000 time steps, corresponding to 1000 QPs to solve. (Each node corresponds to one QP.) For the first CQP, which produces $w(1)$, the initial value $\boldsymbol{w}^0$ of $\boldsymbol{w}$ is set to zero, as is w_{-1}. For all other CQPs, $\boldsymbol{w}^0$ is set to be the warm start $\bar{\boldsymbol{w}}_* := [w_1^*; w_2^*; \cdots ; w_{M-1}^*; w_{M-1}^*]$ derived from the solution $\boldsymbol{w}_*$ of the previous CQP. The initial value of z of IrQP is always chosen to be $\max\{A^T\boldsymbol{w}^0 - c, 0\} + 0.001$. At each iteration, the working set Q consists of the indices of $|Q|$ smallest slack variables s. Parameters in (3.11)–(3.12) were set as $\sigma = 10$, $\gamma_1 = 0.1$, $\gamma_2 = 100$ and $\gamma_4 = 100$. Algorithm IrQP was terminated when stopping criterion (3.84) is satisfied with $tol = 10^{-4}$.

We next show the performance of control inputs and states in Figures 4.3 to 4.6. (In practice, it is more typical to use a quadratic objective in RHC-based control problems. Thus, we analyze the control behavior here but in the linear RHC.) Figures 4.3 and 4.4 show the evolution of the control input $w(t)$ and of the "input rate" $\delta w(t) = w(t) - w(t-1)$, respectively. Constraints on the input rate are seen to be active for the CQPs during the first few seconds; see Figure 4.4. In fact, the lower bound constraint on $\delta w(t)$ is violated at times 1.20 sec to 1.22 sec. This means that the original problem (Dq) was infeasible at those time steps and that, accordingly, the optimal z_* for problem (Dq$_\rho$) was not zero. If the lower bound on $\delta w(t)$ is deemed to be a hard constraint indeed, then one option would be to keep it as-is in (Dq$_\rho$), i.e., not incorporate it in the penalization scheme. (A variation on this idea would be to leave such hard constraints as-is only for $k = 0$, since only w_*^0 is actually applied as a control input $w(t)$.)

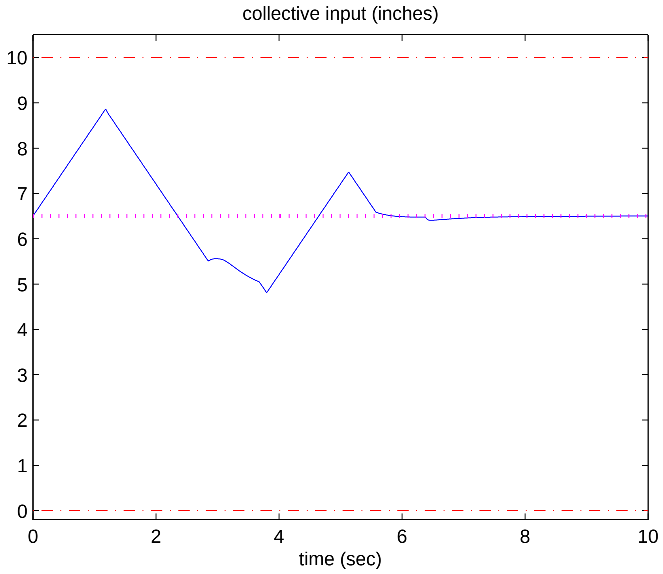


Figure 4.3: Simulation results: control input. The bounds $w_{\min}$ and $w_{\max}$ are marked by red dashed lines and the trim value by a magenta dotted line.

Figures 4.5 and 4.6 show the evolution of the states (in absolute coordinates). Feasibility of the CQPs (except for times 1.20 sec to 1.22 sec) insures that the prescribed bounds on the state variables are never overstepped. Still, as seen on the figures, some of these bounds are reached occasionally: (i) the vertical velocity (3rd state variable) meets its lower bound at times 2.17 sec to 2.21 sec, and (ii) a bound on the shaft torque (5th state variable) is attained during two time intervals, the upper bound at times 1.52 sec and 1.53 sec, and the lower bound at times 3.38 sec

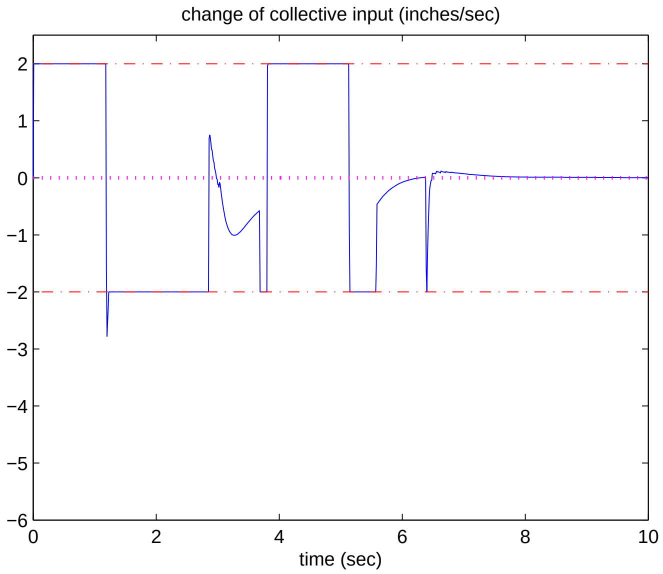


Figure 4.4: Simulation results: rate of change of control input. The bounds ($\frac{\delta w_{\mathrm{min}}}{T_{\mathrm{s}}}$ and $\frac{\delta w_{\mathrm{max}}}{T_{\mathrm{s}}}$ inches/sec) are marked by red dashed lines.

to 4.02 sec; see Figure 4.6.

Figures 4.7 and 4.8 and Tables 4.2 and 4.3 illustrate the performance of IrQP in terms of time to solution. Figure 4.7 shows the number of iterations and total time to solution of the 11th CQP (the one which produces $w(11)$), versus the cardinality of the working set Q (not counting the $z \geq 0$ constraints which, as we indicated, were always included in Q); the behavior is typical. The same information is displayed on Figure 4.8 for the 374th CQP, whose solution took the longest time.[4] Note that, interestingly, the total number of iterations to solution remains essentially constant as $|Q|$ is decreased, down to $|Q| = 80$ or even less. (Related observations were made in linear RHC (see section 4.2) and in [60], [65] and [32] on other classes of

[4] We discounted the very first CQP, solved during time interval $(0, 1)$, for which a warm start was not available.

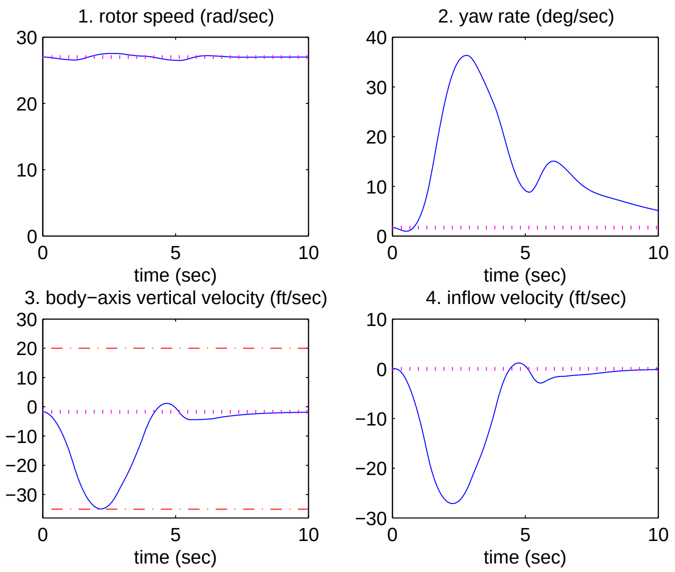


Figure 4.5: Simulation results: states 1 to 4. The bounds on the vertical velocity are marked by red dashed lines and the trim values by magenta dotted lines.

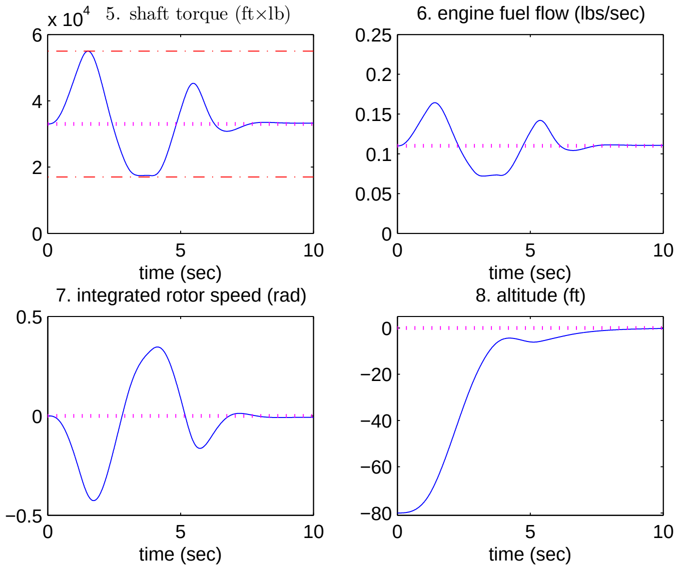


Figure 4.6: Simulation results: states 5 to 8. The bounds on the shaft torque are marked by red dashed lines and the trim values by magenta dotted lines.

applications.)

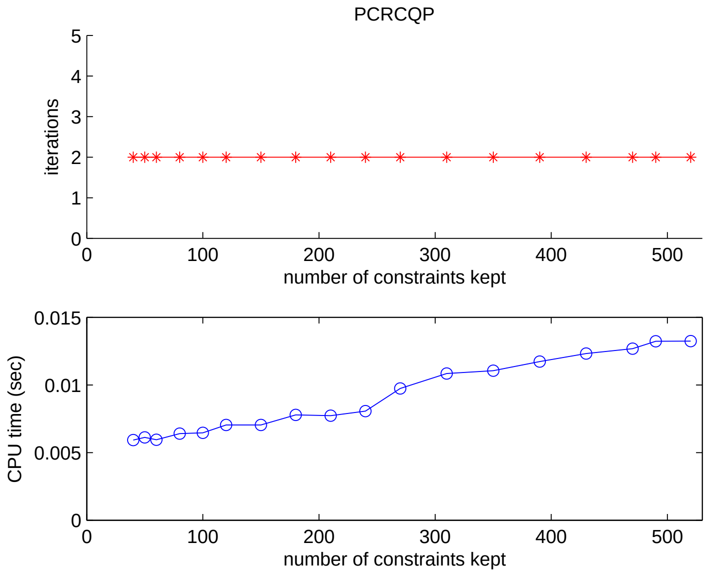


Figure 4.7: Effect of constraint reduction on the number of iterations and total CPU time needed for IrQP to solve the 11th CQP (typical). Each $*$ and o corresponds to a full optimization run, with the cardinality of the working set Q as specified. The rightmost $*$ and o correspond to no constraint reduction ($|Q| = 520$).

Table 4.2 reports the number of sample points at which solving the CQP takes more than a certain time, ranging from 0.01 sec to 0.05 sec, with constraint reduction at level $|Q| = 50$ and without constraint reduction (i.e., $|Q| = 520$).[5] It shows that most (more than 80%) of the CQPs are solved within 0.01 sec using constraint reduction, while all of them need more than 0.01 sec without constraint reduction. Finally, in Table 4.3, we compare the CPU time for three algorithms: Matlab commands qpdantz and quadprog; and IrQP with $|Q| = 50$ and $|Q| = 520$ (no constraint reduction). Default parameter values were used for qpdantz and

[5] See footnote 4.

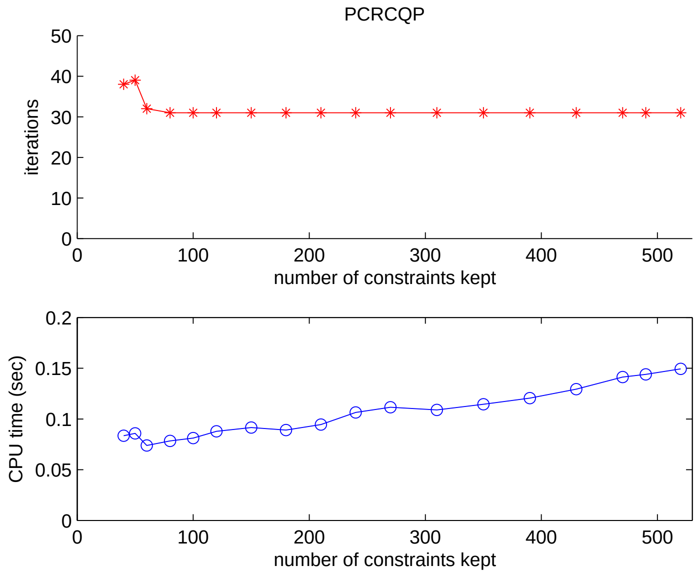


Figure 4.8: Effect of constraint reduction on the number of iterations and total CPU time needed for IrQP to solve the 374th (and most time-consuming) CQP.

Table 4.2: Number of time steps (out of 999) at which the total CPU time is larger than given thresholds

time (sec)	> 0.01	> 0.02	> 0.03	> 0.04	> 0.05
$\lvert Q\rvert = 50$	136	28	13	8	7
$\lvert Q\rvert = 520$	999	192	88	60	25

Table 4.3: CPU Time used by several algorithms to solve the 11th CQP (typical)

qpdantz	quadprog	IrQP ($\lvert Q\rvert = 50$)	IrQP ($\lvert Q\rvert = 520$)
0.3812	0.1383	0.0060	0.0132

quadprog.

We also compare IrQP (using the adaptive scheme for ρ) with Algorithm A of [32] (using the fixed scheme for ρ) which is the same as IrQP but without penalty parameter updates. We choose the fixed penalty parameter $\rho = 2 \times 10^7$ for Algorithm A of [32], because it is large enough that all 1000 CQPs are guaranteed to be solved. (The smallest such value is 2×10^5. In practice, a much larger value than the smallest value is used.) Figure 4.9 shows the CPU time to solve the 10 sec (RHC) process in four cases: with either the adaptive scheme for ρ or a fixed value $\rho = 2 \times 10^7$, and with either constraint reduction (CR) or no constraint reduction (NCR). Since we have 1000 CQPs to solve, to reduce clutter, only every 10th time step is showed. It is noticeable that the time to solve CQPs by IrQP takes less time than Algorithm A of [32] with the fixed value $\rho = 2 \times 10^7$, and the time by constraint reduction is much less than that by no constraint reduction. Thus, the combination of constraint reduction and the adaptive scheme, as we did in algorithm IrQP, is the best of four schemes.

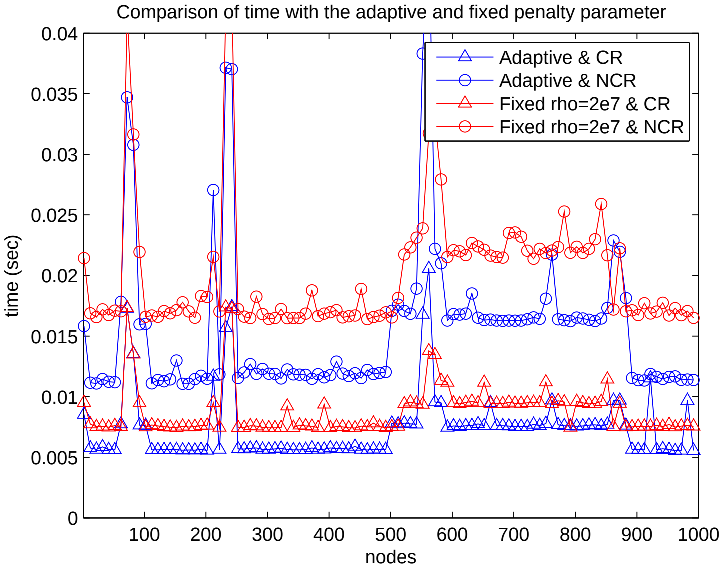


Figure 4.9: CPU time for four strategies: the adaptive scheme for ρ with constraint reduction (in blue triangles), the adaptive scheme for ρ without constraint reduction (in blue circles), the fixed scheme $\rho = 2 \times 10^7$ with constraint reduction (in red triangles) and the fixed scheme $\rho = 2 \times 10^7$ without constraint reduction (in red circles).

4.3.2 Trajectory following

4.3.2.1 Problem setup

In RHC-based trajectory following, during the time interval between sample times $t-1$ and t, the convex quadratic program to be solved online is

$$\min \sum_{k=0}^{M-1} w_k^T R w_k + \sum_{k=1}^{N} (\theta_k - \theta_k^{\mathrm{r}})^T P (\theta_k - \theta_k^{\mathrm{r}}) \tag{4.29}$$

s.t.

$$\theta_{k+1} = A_s \theta_k + B_s w_k, \quad k = 0, \dots, N-1, \tag{4.30}$$

$$\theta_0 := A_s \theta(t-1) + B_s w(t-1), \tag{4.31}$$

$$w_k = w_{M-1}, \quad k = M, \dots, N-1, \tag{4.32}$$

$$w_{\min} \le w_k \le w_{\max}, \quad k = 0, \dots, M-1, \tag{4.33}$$

$$\delta w_{\min} \le w_k - w_{k-1} \le \delta w_{\max}, \quad k = 1, \dots, M-1 \tag{4.34}$$

$$\delta w_{\min} \le w_0 - w(t-1) \le \delta w_{\max}, \tag{4.35}$$

$$\theta_{\min} \le \theta_k \le \theta_{\max}, \quad k = 1, \dots, N, \tag{4.36}$$

$$\delta \theta_{\min} \le \theta_k - \theta_{k-1} \le \delta \theta_{\max}, \quad k = 1, \cdots, N, \tag{4.37}$$

where R is positive-definite and P is semi-positive definite. Path θ^{r} is a known reference trajectory; the objective is to find a control law w such that the rotorcraft closely tracks the specified trajectory subject to the dynamic system and constraints involving states and control inputs. Compared to the constraints of the RHC problem in section 4.3.1, constraints (4.37) on rates of change of the states are newly added and, following the Matlab Model-Predictive-Control Toolbox [42], control values beyond the control horizon are constrained to be the value at the control horizon (see (4.32)).

Define matrices

$$F := \begin{bmatrix} I_p & 0 & \cdots & 0 \\ -I_p & I_p & \cdots & 0 \\ & \ddots & \ddots & \\ 0 & \cdots & -I_p & I_p \end{bmatrix} \in \mathbb{R}^{Np \times Np},$$

and vectors

$$\begin{aligned} \delta\boldsymbol{\theta}_{\max} &:= [\delta\theta_{\max}; \ldots ; \delta\theta_{\max}] \in \mathbb{R}^{Np}, \\ \delta\boldsymbol{\theta}_{\min} &:= [\delta\theta_{\min}; \ldots ; \delta\theta_{\min}] \in \mathbb{R}^{Np}, \\ \boldsymbol{\theta}^{\mathrm{r}} &:= [\theta_1^{\mathrm{r}}; \cdots , \theta_N^{\mathrm{r}}] \in \mathbb{R}^{Np}. \end{aligned}$$

Then from (4.30)-(4.32), the state sequence $\boldsymbol{\theta}$ can be expressed in terms of the control input sequence $\boldsymbol{w}$:

$$\boldsymbol{\theta} = \Gamma_b \boldsymbol{w} + \Omega\theta_0, \tag{4.38}$$

where

$$\Gamma_b = \begin{bmatrix} B_s & 0 & \cdots & 0 & 0 \\ A_s B_s & B_s & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_s^{M-1} B_s & A_s^{M-2} B_s & \cdots & A_s B_s & B_s \\ A_s^M B_s & A_s^{M-1} B_s & \cdots & A_s^2 B_s & A_s B_s + B_s \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_s^{N-1} B_s & A_s^{N-2} B_s & \cdots & A_s^{N-M+1} B_s & \sum_{i=0}^{N-M} A_s^i B_s \end{bmatrix},$$

and Ω is defined as in equation (4.11). Substituting (4.38) into (4.29) and (4.36), the RHC problem can be transformed into a CQP problem of the standard form (Dq) with

$$b := -\Gamma_b^T \mathbf{P}(\Omega\theta_0 - \boldsymbol{\theta}^{\mathrm{r}}), \quad H := \mathbf{R} + \Gamma_b^T \mathbf{P} \Gamma_b$$

and

$$A := -\begin{bmatrix} -I_{Mm} \\ I_{Mm} \\ -\Gamma_b \\ \Gamma_b \\ -E_r \\ E_r \\ -F\Gamma_b \\ F\Gamma_b \end{bmatrix}^T, \quad c := -\begin{bmatrix} -\boldsymbol{w}_{\max} \\ \boldsymbol{w}_{\min} \\ -\boldsymbol{\theta}_{\max} + \Omega\theta_0 \\ \boldsymbol{\theta}_{\min} - \Omega\theta_0 \\ -(\delta\boldsymbol{w}_{\max} + E_r^1 w(t-1)) \\ \delta\boldsymbol{w}_{\min} + E_r^1 w(t-1) \\ -\delta\boldsymbol{\theta}_{\max} + (F\Omega - F_1)\theta_0 \\ \delta\boldsymbol{\theta}_{\min} - (F\Omega - F_1)\theta_0 \end{bmatrix}$$

where $\boldsymbol{w}_{\max}$, $\boldsymbol{w}_{\min}$, $\delta\boldsymbol{w}_{\max}$, $\delta\boldsymbol{w}_{\min}$, $\boldsymbol{\theta}_{\max}$, $\boldsymbol{\theta}_{\min}$ and E_1^r were defined in section 4.3.1, and $F_1 := [I_p; 0; \ldots; 0] \in \mathbb{R}^{Np\times p}$. Because H is positive definite, the transformed CQP has a unique solution $\boldsymbol{w}^* = [w_0^*; \ldots; w_{M-1}^*]$ whenever it is feasible. In the spirit of model predictive control, control input $w(t) = w_0^*$ is applied at sample time t. The other components of $\boldsymbol{w}^*$ are discarded but used as a "warm-start" towards solving the next CQP.

The number of variables in transformed standard CQP (Dq) is Mr and, when all the state and control input variables are constrained (all components of $w_{\min}, w_{\max}, \delta w_{\min}, \delta w_{\max}, \theta_{\min}, \theta_{\max}, \delta\theta_{\min}, \delta\theta_{\max}$ are finite), the total number of constraints is $4Mr + 4Np$. Thus, the number of constraints is significantly larger than that of variables, making constraint reduction beneficial.

4.3.2.2 Rotorcraft models

We started from model M2,[6] a linear time-invariant, continuous-time model of the UH60 Blackhawk, of the form

$$\dot{\theta} = A^{\mathrm{ct}}\theta + B^{\mathrm{ct}}w. \tag{4.39}$$

This model was provided to us by R. Celi, who generated it as described in [58]. It has 37 states and 10 control inputs; see Table 4.4. (Only the first four control inputs are shown; the others have a moderate effect and were set to zero throughout.) This model, rather than the classic model used in the previous sections 4.2.2 and 4.3.1, was used because it is larger and more complicated, thus, can further demonstrate the efficiency of our algorithm. For the purpose of the present section, model M2 was assumed to be exact and appropriate components $\theta(\cdot)$ of its states were substituted in (4.31) for the measured state of the rotorcraft.

The model (A_s, B_s) used in the RHC controller was obtained from model M2 as follows. First, a glance at the data shows that, with the initial state set to zero (as we did in our simulations: see section 4.3.2.3 below), the last six components (32 to 37) of the states remain at zero throughout, regardless of the value of the control inputs; in particular, the last six—indeed, eight— rows of B^{ct} are zero. Accordingly, the trajectories of the first 31 states remain unaffected if the last six

[6] The name reflects the fact that the model accounts for two flexible-blade modes (rigid body flap and drag).

Table 4.4: Description of the original model M2 from [58]. States and control inputs estimated/generated by the controller are marked in boldface.

States 1-3 $(\hat{u}, \hat{v}, \hat{w})$	**Velocity components along the body axes**
States 4-6 $(\hat{p}, \hat{q}, \hat{r})$	**Roll, pitch and yaw rates**
States 7-9 $(\hat{\phi}, \hat{\theta}, \hat{\psi})$	**Roll, pitch and yaw angles**
States 10-17	Flexible mode: Blade rigid body flap
States 18-25	Flexible mode: Blade drag
States 26, 27	Torsion, torsion rate
State 28	Constant portion of main rotor inflow
States 29-30	First harmonic (cos and sin) of main rotor inflow
State 31	Tail rotor inflow
State 32	Down wash on tail
State 33	Side wash on tail
States 34-37	Engine/rotor speed dynamics
Control inputs 1-4	**Lateral cyclic, longitudinal cyclic**, collective and **pedal**

rows and columns of A^{ct} and the last six rows of B^{ct} are deleted, which we did. Next, to keep the RHC controller reasonably simple and fast, we decided to only simulate the first nine states (the "classic nine") in the built-in simulator (4.30)-(4.31).

The reduction from 31 to nine states was effected as follows. Denote by $A^{\mathrm{ct}}_{\alpha,\beta}$ the submatrix of A^{ct} that consists of only those rows in set α and only those columns in set β. Let subscript r indicate the kept states in the reduced model, and d be the deleted states. By setting the derivatives of the deleted states to zero, system (4.39) becomes

$$\dot{\theta}_{\mathrm{r}} = A^{\mathrm{ct}}_{\mathrm{r,r}}\theta_{\mathrm{r}} + A^{\mathrm{ct}}_{\mathrm{r,d}}\theta_{\mathrm{d}} + B^{\mathrm{ct}}_{\mathrm{r}}w, \tag{4.40}$$

$$0 = \dot{\theta}_{\mathrm{d}} = A^{\mathrm{ct}}_{\mathrm{d,r}}\theta_{\mathrm{r}} + A^{\mathrm{ct}}_{\mathrm{d,d}}\theta_{\mathrm{d}} + B^{\mathrm{ct}}_{\mathrm{d}}w. \tag{4.41}$$

Noting that $A^{\mathrm{ct}}_{\mathrm{d,d}}$ in M2 is non-singular, we can solve (4.41) for θ_{d} and substitute into (4.40), yielding

$$\dot{\theta}_{\mathrm{r}} = \overline{A^{\mathrm{ct}}}\theta_{\mathrm{r}} + \overline{B^{\mathrm{ct}}}w \tag{4.42}$$

where

$$\overline{A^{\mathrm{ct}}} = A^{\mathrm{ct}}_{\mathrm{r,r}} - A^{\mathrm{ct}}_{\mathrm{r,d}}(A^{\mathrm{ct}}_{\mathrm{d,d}})^{-1}A^{\mathrm{ct}}_{\mathrm{d,r}}, \quad \overline{B^{\mathrm{ct}}} = B^{\mathrm{ct}}_{\mathrm{r}} - A^{\mathrm{ct}}_{\mathrm{r,d}}(A^{\mathrm{ct}}_{\mathrm{d,d}})^{-1}B^{\mathrm{ct}}_{\mathrm{d}}.$$

Our objective is to track a reference trajectory consisting of a nose-down followed by an aggressive nose-up maneuver (see Figure 4.13), subject to actuator and other limitations. One limitation is from the load factor associated to pitch, namely,

$$N_z := \frac{V}{g}\frac{Z_w}{s+Z_w}q \qquad (s \text{ is the Laplace variable})$$

which satisfies the differentiable equation (after taking the inverse Laplace transform)

$$\dot{N}_z = \frac{V}{g} Z_w q - Z_w N_z. \tag{4.43}$$

Here, g is the acceleration of gravity, V is the forward speed of the helicopter and Z_w is the negative of the $(3,3)$ entry of $\overline{A^{\mathrm{ct}}}$ (associated with state w) in the reduced model. According to the model M2 data, $V = 80$ kts $= 134.96$ ft/sec and $Z_w = 0.6920$ sec^{-1}.

We augmented both (4.39) and (4.42) with (4.43), yielding a 38-state model considered below as an exact representation of the rotorcraft, and a 10-state model to be used by the controller. Both models have 10 control inputs. Discrete-time models were then generated from both (using Matlab command `c2d`). The first one was fed the sequence $u(\cdot)$ generated by the controller, and appropriate components $x(\cdot)$ of its state were fed back (see (4.31)). The second one was used in controller's estimator (4.30)–(4.31).

4.3.2.3 Optimization details

The desired pitch trajectory was set as shown by the solid lines on Figure 4.12 (pitch rate) and Figure 4.13 (pitch attitude), the former being the derivative of the latter. All other components of the desired trajectories were set to zero. Because use of the collective is not essential for tracking such a trajectory, and for sake of controller simplicity, we decided to set to zero all but three control inputs: lateral cyclic, longitudinal cyclic and pedal. Values $R = \mathrm{diag}(1,1,1)$ and $Q = \mathrm{diag}(0,0,0,10^3,10^4,10,10,10^4,10,0)$ were found to be appropriate for the cost function (no tracking is attempted for $\hat{u}$, $\hat{v}$, $\hat{w}$ and N_z). Constraint bounds were set to $\hat{w}_{\max} = -\hat{w}_{\min} = [5,\ 5,\ 5]^T$ inches, $\delta\hat{w}_{\max} = -\delta\hat{w}_{\min} = [4,\,4,\,4]^T$ inches/sec, $\hat{\phi}_{\max} = -\hat{\phi}_{\min} = 5$ degrees, $\hat{\psi}_{\max} = -\hat{\psi}_{\min} = 4$ degrees, $N_z^{\max} = -N_z^{\min} = 1$ g, $\delta\hat{p}_{\max} = -\delta\hat{p}_{\min} = 1$ inch/sec^2 and $\delta\hat{q}_{\max} = -\delta\hat{q}_{\min} = 1$ inch/sec^2. Bound constraints on all control inputs and their rates of change were set to be "hard",[7] while all other constraints were made "soft".

With three control inputs ($r = 3$), the CQP problem in the form (Dq) has $3M$ variables. Only three of ten state variables have finite prescribed upper and lower bounds, which gives us $6N$ constraints from (4.36). Also, there are two of ten state variables have finite upper and lower bounds on their change rate, giving another $4N$ constraints from (4.37). In addition, all bounds on control inputs and their rates of change are finite, which are $4 \times 3M$. Hence, the total number of constraints in the CQP is $12M + 10N$. With our current choice of $M = 30$ and $N = 100$, we thus have 90 optimization variables and 1360 constraints.

Note that global convergence of IrQP is proved in Chapter 3 under general guidelines on how to select Q; much freedom is left to be exploited based on the application. In the present implementation, during the first few (10 in our experiments) iterations in the optimization process of current problem, we forced the set Q to always contain (as a subset) the active set at the solution of the *previous* problem. Indeed, active constraints at the solution of two successive problems do not change much, so this strategy is much more efficient from our experiments than that Q includes purely Q most active constraints.

[7] Hard constraints must be satisfied, even when there is no feasible point for all constraints.

For the first CQP (where $t = 0$), initial values of control inputs and initial states were all zeros. For all other CQPs, the way of setting warm starts and initial values is the same as in section 4.3.1.2. Additionally, the initial value of the penalty parameter was set to 2×10^7 for the first CQP, and for all others, it was set to $\min(2 \times 10^7, 10\|\lambda_*\|_\infty)$ where λ_* is the dual solution at previous time step. Parameters in (3.11)–(3.12) were set as $\sigma = 10$, $\gamma_1 = 0.1$, $\gamma_2 = 100$ and $\gamma_4 = 100$. The penalty parameter was allowed to increase at most five times because too large a value slows down the optimization process. The optimization runs were terminated when (3.84) was satisfied with $tol = 10^{-4}$ or when a maximum iterate count IterMax = 100 was reached. In one of our tests (see the end of section 4.3.2.4), they were terminated on elapsed time.

4.3.2.4 Simulation results

As already mentioned, our RHC controller was applied to the full 37-state model in lieu of the actual rotorcraft. The initial state was set to zero. We simulated the system for 10 sec with a sample time $T_s = 0.01$ sec, so that 1000 CQPs were solved.

Figures 4.10 and 4.11 show the evolution of three control inputs $\hat{w}(t)$ and their rates of change (defined by $(\hat{w}(t) - \hat{w}(t-1))/T_s$). It can be seen from Figure 4.11 that the upper or lower bounds on the rates of change were reached at most time steps. Figures 4.12 and 4.13 show the comparison of the actual and reference trajectories for the pitch rate and pitch, respectively. From about 5 sec to 7 sec, the reference trajectory of pitch was not followed well. This is due to the bound limitation on rates of change of control inputs and to the upper bound limitation of the load factor of pitch; see Figures 4.11 and 4.16. Figure 4.14 shows the rates of change of the roll and pitch rates, which are defined by $(\hat{p}(t) - \hat{p}(t-1))/T_s$ and $(\hat{q}(t) - \hat{q}(t-1))/T_s$, respectively. Evolution of other states is shown in Figures 4.15–4.18.[8]

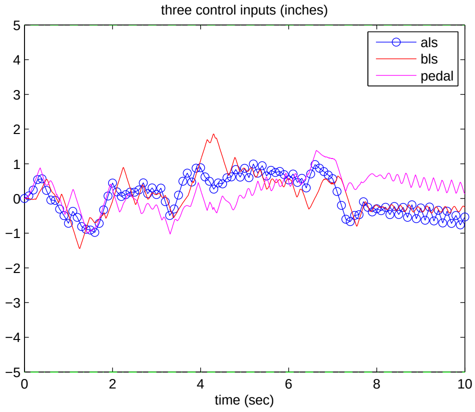


Figure 4.10: Three control inputs: lateral cyclic, longitudinal cyclic and pedal, marked by blue circles, a red and magenta line, respectively.

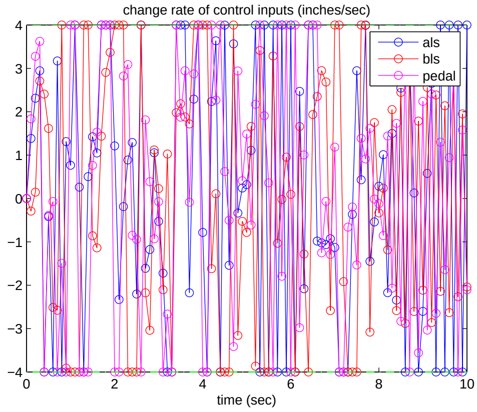
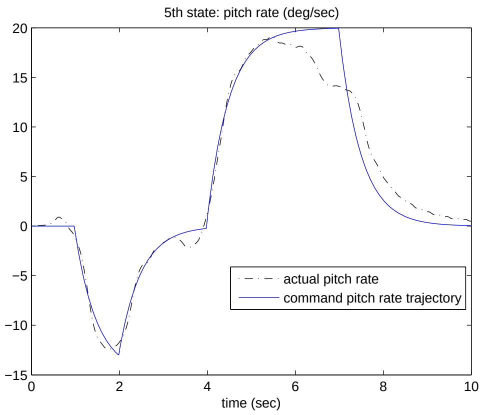


Figure 4.11: Rates of change of three control inputs. The upper bound 4 inches/sec and lower bound −4 inches/sec are marked by dashed green lines.

We compared in Figures 4.19 and 4.20 the CPU time and iterations to solve those 1000 CQPs with $|Q| = 120$ and with no constraint reduction (corresponding to $|Q| = 1360$). To reduce clutter, the results of only every tenth CQP are showed. It can be seen from Figure 4.19 that for each CQP, constraint reduction takes much less time than no constraint reduction does. We also observed from Figure 4.20 that

[8] The control inputs in Figure 4.10 would go into the actuator, which is as a second order filter, so a much smoother control input goes into to physical aircraft, yielding a much smoother state evolution than shown in the thesis.



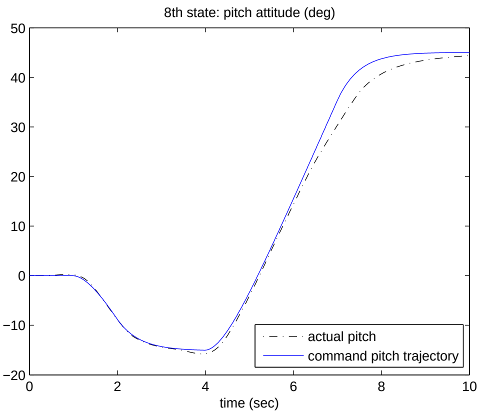


Figure 4.12: Trajectory following of the pitch rate

Figure 4.13: Trajectory following of the pitch

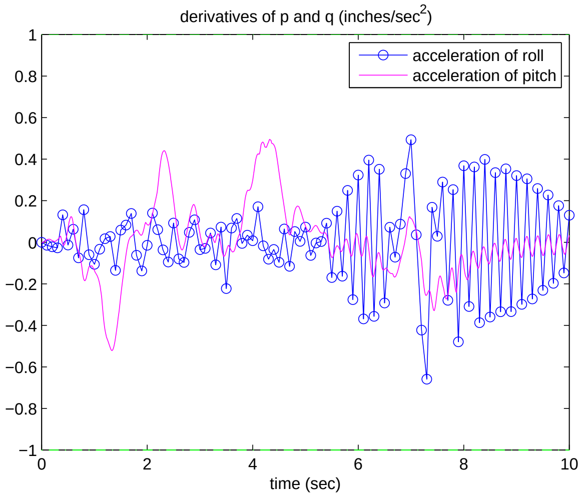


Figure 4.14: Acceleration of roll and pitch. Both have an upper bound 1 inch/sec^2 and a lower bound -1 inch/sec^2.

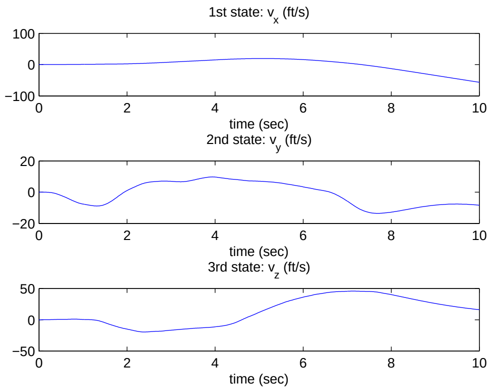


Figure 4.15: Velocity components along the body axes (without bounds)

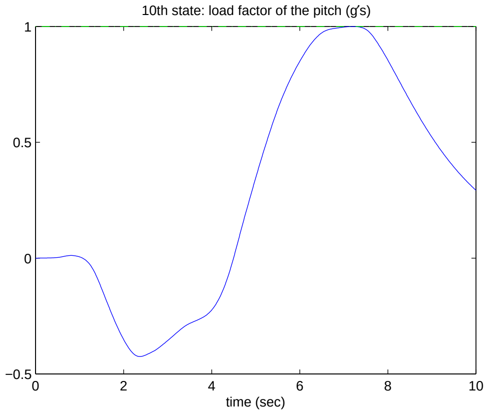


Figure 4.16: Load factor of the pitch.

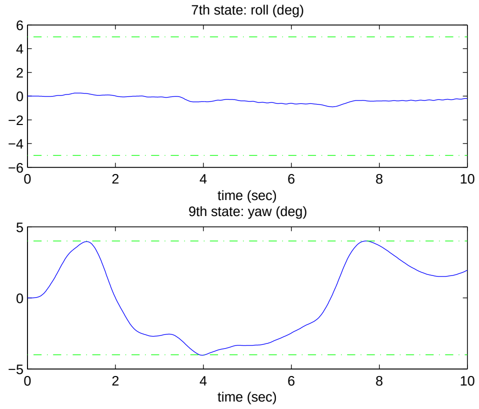


Figure 4.17: Roll and yaw angles. Bounds are marked by green dashed lines.

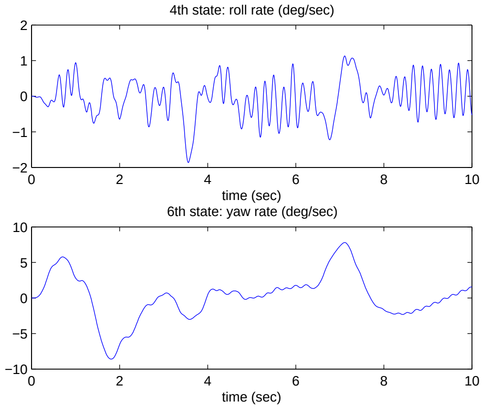


Figure 4.18: Roll and yaw rate (without bounds)

constraint reduction does not increase the number of iterations compared with no constraint reduction.

Table 4.5 shows a comparison of total time for solving all 1000 CQPs with four methods: function `qpdantz` and `quadprog` in Matlab, and IrQP with $|Q| = 120$ and $|Q| = 1360$. While function `qpdantz` (the solver used in the Matlab Model-Predictive-Control Toolbox) takes the longest time, constraint reduction takes the least time.

Figure 4.21 shows the total time and number of iterations to solve all 1000 CQPs with constraint reduction for various *fixed* values of the penalty parameter, i.e, with the adaptation scheme (3.11)–(3.12) turned off. (Algorithm IrQP then becomes Algorithm A of [32].) We recorded the total time and number of iterations for solving the 1000 problems using various values of ρ, from 2×10^6 to 10^9. (The value $\rho = 2 \times 10^6$ turns out to be the smallest value that is large enough for all 1000 CQPs, as was determined by trial and error. Of course, for other trajectories to be tracked, a larger value may be needed, so a robust fixed-ρ controller would have to use a value much larger than 2×10^6.) Magenta dashed lines denote the result by IrQP with the adaptive scheme where warm starts for the penalty parameter were used. It can been seen from the figure that the adaptive penalty scheme performs better than the scheme with fixed penalty values for most values, and much better for large penalty values.

When an RHC-based controller is implemented, the optimization process must be stopped when the next stopping time is reached, at which time the best control value obtained so far is applied to the system being controlled. We compared the

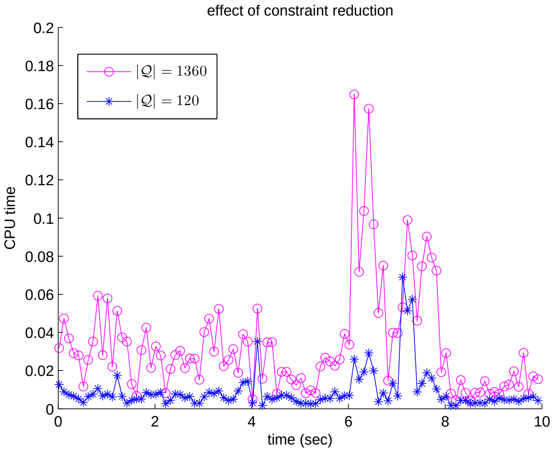


Figure 4.19: Comparison of CPU time with constraint reduction and no constraint reduction

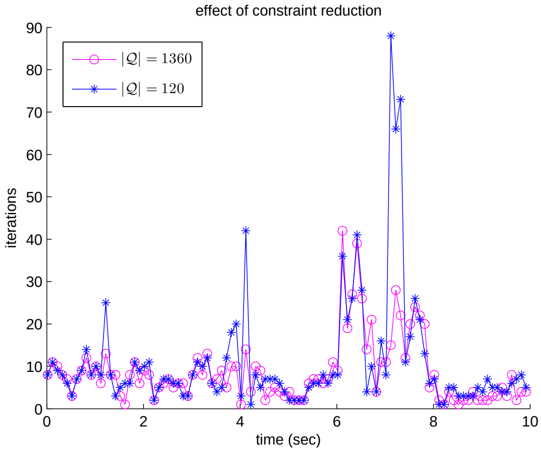


Figure 4.20: Comparison of iteration counts with constraint reduction and no constraint reduction

Table 4.5: The total RHC time with different QP solvers in a 10 sec simulation

Methods	`qpdantz`	`quadprog`	$\|Q\| = 1360$	$\|Q\| = 120$
time (sec)	5243.2	235.7	45.4	13.3

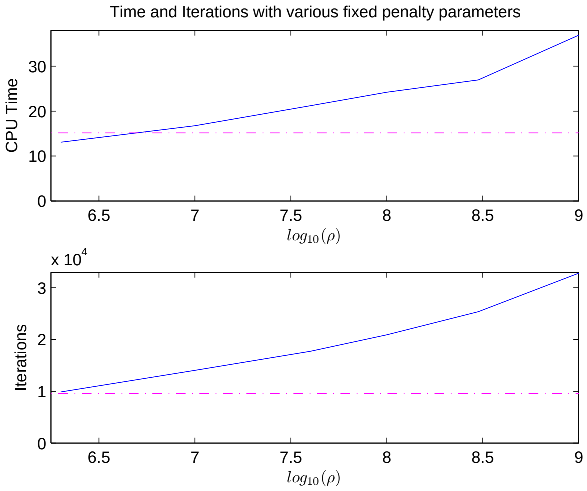


Figure 4.21: Total time and number of iterations with fixed penalty values ρ and $|Q|$=120. The magenta dash lines mark the total time and number of iterations with the adaptive scheme.

performances of IrQP on our trajectory following problem with $|Q| = 1360$ (in Figures 4.22–4.25) and $|Q| = 120$ (in Figures 4.26–4.29), with the optimization stopped after 0.012 sec (0.010 sec was a little too short a time for our controller with the computer we used for our runs). It can be seen that the trajectories were followed much better in the latter case than the former (Figures 4.22–4.23 vs. Figures 4.26–4.27), and the associated constraint violations were much smaller (Figures 4.24–4.25 vs. Figures 4.28–4.29).

4.4 Conclusion

We applied the algorithms IrMPC and IrQP to solve the optimization in RHC-based control problems. Numerical simulations, which were implemented for different control purposes (altitude control and trajectory following) and for different

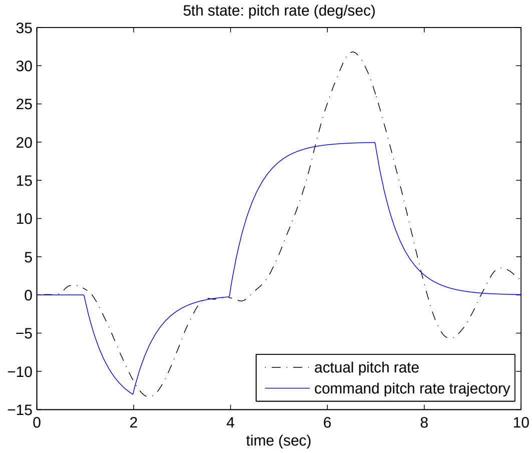


Figure 4.22: Trajectory following of the pitch rate with $|Q| = 1360$ and a time limit of 0.012 sec

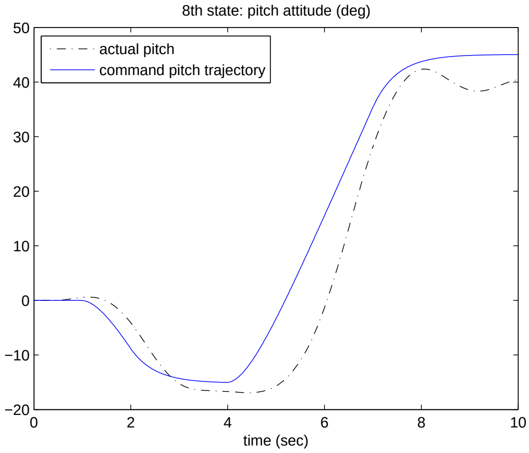


Figure 4.23: Trajectory following of the pitch with $|Q| = 1360$ and a time limit of 0.012 sec

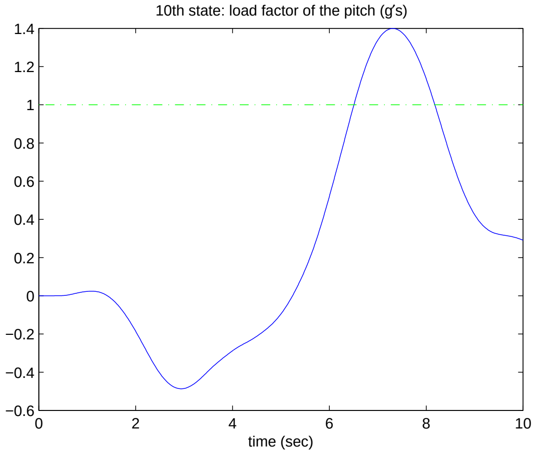


Figure 4.24: Load factor of the pitch with $|Q| = 1360$ and a time limit of 0.012 sec

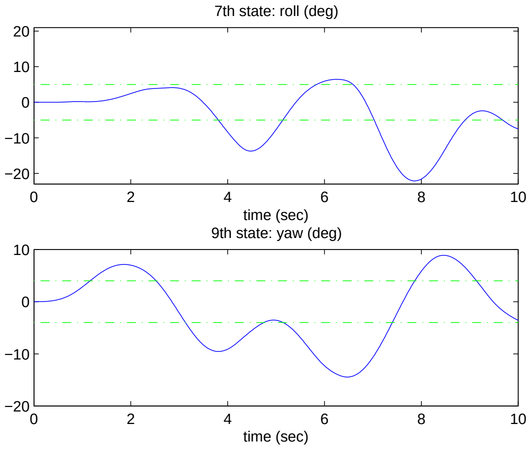


Figure 4.25: Roll and yaw with $|Q| = 1360$ and a time limit of 0.012 sec

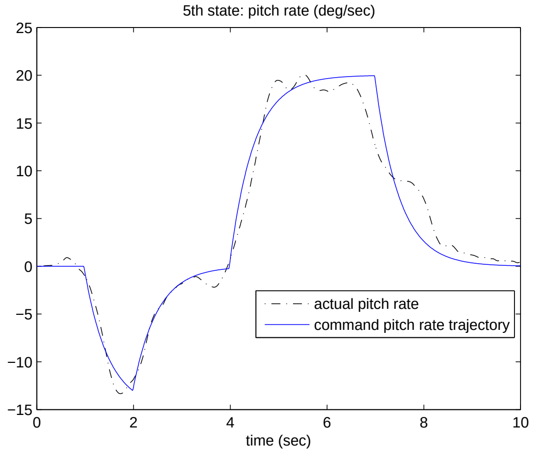


Figure 4.26: Trajectory following of the pitch rate with $|Q| = 120$ and a time limit of 0.012 sec

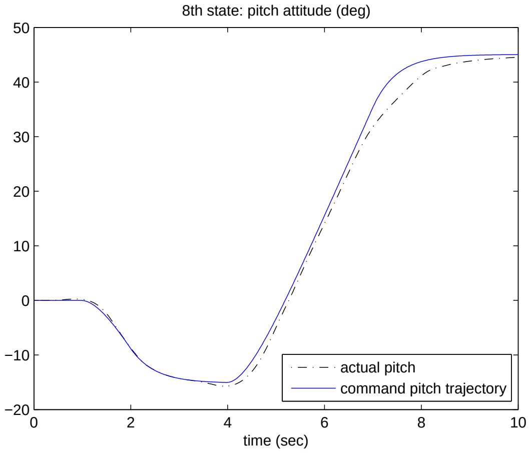


Figure 4.27: Trajectory following of the pitch with $|Q| = 120$ and a time limit of 0.012 sec

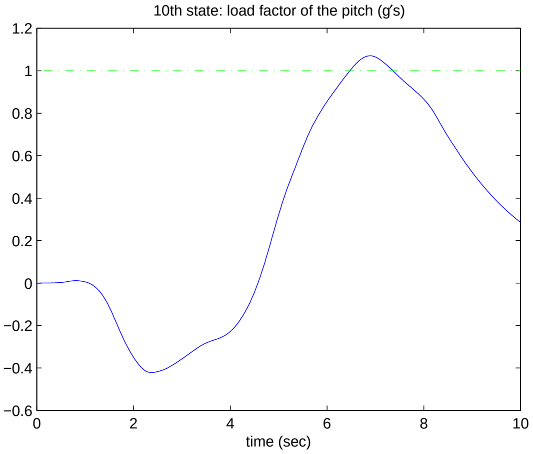


Figure 4.28: Load factor of the pitch with $|Q| = 120$ and a time limit of 0.012 sec

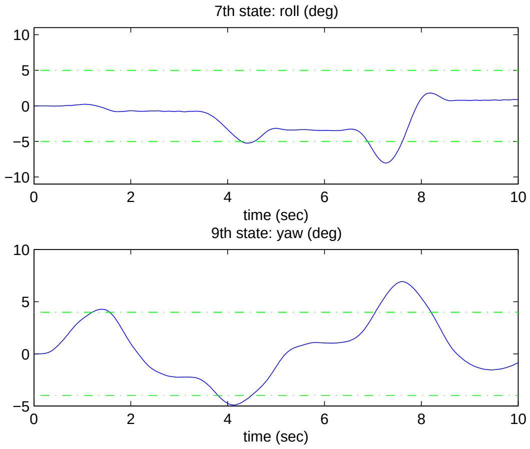


Figure 4.29: Roll and yaw with $|Q| = 120$ and a time limit of 0.012 sec

models (the classic eight-state model and 37-state model M2), suggest that our proposed algorithms are of practical interests.

While the "on-line" optimization problem was not always solved within the 0.01 second sample time interval, we expect that this should be overcome when a faster, multicore processor is used. Also, because algorithm IrQP is an affine-scaling version, a Mehrotra's Predictor-Corrector version can also be developed, which is likely to be faster than the former. Furthermore, we have the expectation that this infeasible constraint-reduced approach can be extended to solve the optimization problems in nonlinear model predictive control.

Chapter 5

Future Work

In this chapter, future work is discussed that includes the extensions of both feasible and infeasible constraint reduction algorithms to general convex optimization. For the former, the idea is to use constraint reduction schemes by extending the algorithms of linear optimization [60] and convex quadratic optimization [32]. For the latter, one would apply exact penalty functions to allow for arbitrary initial points and extend the penalty adaptation scheme in Chapter 2 and 3. In the end, several problems are discussed as potential applications of the proposed algorithms.

5.1 Extension to convex optimization (CO)

Consider the convex optimization (CO) in the form

$$\min f(y) \quad \text{s.t.} \quad h^i(y) \leq 0,\ i = 1, \cdots, n \tag{5.1}$$

where $f(y)$ and $h^i(y)$ for all $i = 1, \cdots, n$ are convex and their second derivatives are continuous. Convex optimization is special, because every local minimum is a global minimum, and because of the convexity of the cost function, its Newton direction is always descent.

Again, the case we consider is $n \gg m$; i.e., there are many more convex constraints than variables, which is the case when constraint reduction should be beneficial.

5.1.1 Feasible constraint reduction for CO

It is natural to extend the constraint-reduced algorithms of linear and quadratic optimization in [60, 65, 32] to the more general convex optimization. Such extension is based on the fact that the Newton direction Δy obtained in PDIPMs is a direction of descent [61]. This fact plays an important role in the analysis of global convergence and as is shown next, is preserved in the constraint-reduced versions of PDIPMs. Applying constraint reduction on (5.1) yields

$$\min \; f(y) \quad \text{s.t.} \quad h^Q(y) \leq 0. \tag{5.2}$$

The affine-scaling search direction $(\Delta x^Q, \Delta y, \Delta s^Q)$ for (5.2) can be obtained by solving

$$\begin{bmatrix} 0 & A^Q(y)^T & I \\ A^Q(y) & H^{(Q)}(x,y) & 0 \\ S^Q & 0 & X^Q \end{bmatrix} \begin{bmatrix} \Delta x^Q \\ \Delta y \\ \Delta s^Q \end{bmatrix} = \begin{bmatrix} 0 \\ -A^Q(y)x^Q - \nabla f(y) \\ -X^Q s^Q \end{bmatrix} \tag{5.3}$$

where $A(y)$ are the gradients of $h(y)$ at point y, and

$$H^{(Q)}(x,y) := \nabla^2 f(y) + \sum_{i \in Q} x_i \nabla^2 h_i(y).$$

Eliminating Δx^Q and Δs^Q in (5.3), we derive the normal equations

$$M^{(Q)} \Delta y = -\nabla f(y) \tag{5.4}$$

where the reduced normal matrix is

$$M^{(Q)} := A^Q(y) X^Q (S^Q)^{-1} (A^Q(y))^T + H^{(Q)}(x,y).$$

Notice that $[A^Q(y) \quad H^{(Q)}(x,y)]$, rather than $A^Q(y)$, can be chosen to be full rank, so $M^{(Q)}$ is positive-definite. (The implicit assumption is that $[A(y) \;\; H(x,y)]$ has full rank for any x and y.) Thus, (5.4) implies that

$$\nabla f(y)^T \Delta y = -\Delta y^T M^{(Q)} \Delta y < 0,$$

showing that direction Δy is a descent direction when constraint reduction is used.

Constraint reduction in convex programming can save the computation cost per iteration in many respects. First, only partial gradients $A^Q(y)$ and partial Hessians of constraints are needed for the two summation terms in $M_{(Q)}$. Computing gradients and Hessians of all constraints, which might be expensive, will be reduced to those of only $|Q|$ constraints. Second, as in the LP and CQP cases, only a subset of constraints is involved in forming $M^{(Q)}$. This reduces the two n-term summations to two $|Q|$-term summations. Furthermore, old gradients and Hessians can be reused for direction $\Delta x^{\overline{Q}}$ or $\Delta s^{\overline{Q}}$ without losing global convergence: One idea is then to recompute only the gradients and Hessians of the constraints that are included in Q and fill the remaining with those computed in previous iterates. For instance, computing $\Delta s_k^{\overline{Q}}$ at iterate k can use the old gradients $A^{\overline{Q}}(y_j)$ at iterate j ahead of k, i.e.,

$$\Delta s_k^{\overline{Q}} = -A^{\overline{Q}}(y_j)^T \Delta y_k, \quad j < k.$$

5.1.2 Infeasible constraint reduction for CO

To allow for infeasible initial points in constraint reduction schemes, we can relax (5.1) with an ℓ_1 (or ℓ_∞) penalty function, and solve the penalized problem

$$\min f(y) + \rho e^T z \quad \text{s.t.} \qquad h^Q(y) \leq z^Q,\ z \geq 0 \tag{5.5}$$

with feasible algorithms discussed in the previous section. There are two main reasons for using the exact penalty function: Strictly feasible points are trivially available for problem (5.5), and according to Theorem 40 in [23], this ℓ_1 penalty function is "exact" as in the case of LPs and CQPs. An extra advantage of problem (5.5) over problem (5.1) is that the strong duality holds for problem (5.5), but not necessarily for (5.1). This is because in convex programming, strong duality does not always hold, even when the primal is feasible [12]. However, problem (5.5) is strictly feasible, satisfying the Slater condition, so it has both the property of strong duality and an optimal solution.

For (5.5), the challenge would be how to choose an appropriate penalty parameter, for which we need to design an automatic adjustment scheme. A scheme can be proposed as the same for LPs and QPs except for a significant modification in condition (i) of (2.12) and (3.12). Suppose there exists an integer k_o and a positive constant C such that for all k,

$$\max\{[x_k; u_k]\} \leq C\rho^{k_o},$$

where x and u are the dual variables associated with constraints $h(y) \leq z$ and $z \geq 0$, respectively. (This assumption is mild and satisfied in many cases. Indeed, $k_o = 1$

in the LP case of Chapter 2, and k_o is set to be zero for nonlinear programming [61] and for convex quadratic programming [32].) Accordingly, the iterates of primal variables x_k and u_k can be established to be bounded by $C\rho^{k_o}$, unlike the fixed bound χ in IrQP, and $w_{\max}$ in [32]. Then the adjustment scheme for ρ can be proposed as follows: increase ρ when either

$$\|z\| \geq \gamma_1 \rho$$

OR

$$\|[\Delta y; \Delta z]\| \leq \frac{\gamma_2}{\rho^{k_o}}, \text{ AND } \tilde{x}^Q \geq -\gamma_3 e, \text{ AND } \tilde{u}^Q \geq \gamma_4 e \tag{5.6}$$

is satisfied, where condition (i) of (5.6) has been modified to indicate the closeness to a stationary point. Indeed, condition (i) of (5.6), together with x and u bounded by $C\rho^{k_o}$, implies that $[y; z]$ is close to a stationary point of (5.5).

The main part of theoretical analysis is to show the boundedness of ρ_k, which can be proven similarly to the analysis in Chapter 3. A challenge is that in convex optimization, both the Hessian of objective function and the gradients of constraints depend on the variables, rather than being constants as for CQPs. This might lead to the unboundedness of the Hessian and gradients due to the unboundedness of the variable, causing theoretical difficulty. The effort would then be devoted to show the boundedness of the penalty parameter without any assumption on the sequence of $\{y_k\}$ and $\{z_k\}$, as in the LP and CQP cases. Since Lemma 3.2 (see also Corollary 20 in [23]) suggests that $\{z_k\}$ might be bounded if $\{y_k\}$ is bounded, it is likely to be sufficient to assume only the boundedness of $\{y_k\}$, which is still weaker than the assumption in nonlinear programming of [61].

5.2 Specific applications

The following are some potential application candidates of the infeasible constraint-reduced interior point method for convex optimization.

5.2.1 q-norm support vector machines

Consider the q-norm support vector machines (SVMs) [29, 59] as follows:

$$\min_{w,\gamma} \quad \frac{1}{q}\|w\|_q^q \quad \text{s.t.} \quad C(B^T w - e\gamma) \geq e$$

where $B \in \mathbb{R}^{m \times n}$ with $n \gg m$. The ith column b_i of B denotes the ith given point. C is a diagonal matrix whose diagonal entry $C_{ii} \in \{-1, 1\}$ labels the class the ith point belongs to. When $q = 1$ or $q = \infty$, it is a linear problem; When $q = 2$, it is classical 2-norm SVM [31, 32]. The objective of SVMs is to find the separating hyperplane

$$x^T w = \gamma$$

that maximizes the margin between the points tagged by 1 and those tagged by -1. Due to $n \gg m$ (more cases than attributes), this problem has many more constraints than variables; see the classification data sets at UCI Machine Learning Repository [43] from real examples.

However, such a hyperplane may not exist. "Soft-margin" methods choose the hyperplane that splits the points as cleanly as possible while maximizing the margin of those points that can be split. That is exactly the functionality of exact penalty functions. Specifically, using an ℓ_1 exact penalty function in (5.5) yields the "soft-margin" problem

$$\min_{w,\gamma} \frac{1}{q}\|w\|_q^q + \rho e^T z \quad \text{s.t. } C(B^T w - e\gamma) + z \geq e,\ z \geq 0.$$

5.2.2 Entropy optimization

Consider the problem to choose a probability distribution given the information about some moments of the distribution. Out of all possible distributions that satisfy the moments, we can choose the one that maximizes the entropy. Entropy is a measure of the uncertainty and maximizing the entropy tries to avoid any judgement about unknown information.

In particular, one entropy optimization problem with m variables and infinitely many linear constraints can be stated as

$$\begin{aligned} \inf \quad & f(x) = \sum_{j=1}^{m} x_j \ln x_j \\ \text{s.t.} \quad & \sum_{j=1}^{m} g_j(t) x_j \geq h(t),\ \forall t \in T, \\ & x_j \geq 0,\ j = 1, \cdots, m \end{aligned} \tag{5.7}$$

with a compact set T and real valued continuous functions $h(t)$ and $g_j(t)$, $j = 1, \cdots, m$. Note that the negative entropy function in the objective is convex on the set $x_j \geq 0$ for $j = 1, \cdots, m$.

This model has many potential applications, such as transportation planning, medication, image construction [22], especially with a continuous temporal or spatial domain. For example, in dosage distribution, many medications consist of a mixture of several drugs. Suppose there are m drugs and the concentration of drug j can be described by a function of time $g_j(t)$. Define $x_j \geq 0$ the proportion of drug j used in the mixture. To guarantee the effectiveness, the total concentration of the mixture must be above a level $h(t)$ all through the time interval T, yielding constraints (5.7). Meanwhile, each drug may have side effects. Hence, the mixing of these drugs must be as even as possible, resulting in the objective to maximize the entropy function $-\sum_{j=1}^{m} x_j \ln x_j$, or equivalently, to minimize $\sum_{j=1}^{m} x_j \ln x_j$.

Constraints (5.7) can be discretized at a finite set of points $\{t_1, \cdots, t_n\}$ on T. Hence, the problem can be approximated as a convex program with m variables and n linear inequality constraints:

$$\begin{aligned} \inf \quad & f(x) = \sum_{j=1}^{m} x_j \ln x_j & (5.8) \\ \text{s.t.} \quad & \sum_{j=1}^{m} g_j(t_i) x_j \geq h(t_i),\ i = 1, \cdots, n, & (5.9) \\ & x_j \geq 0,\ j = 1, \cdots, m. & (5.10) \end{aligned}$$

Obviously, the finer the discretization of T is, the more accurate the approximation is. Thus, to get a reasonably accurate approximation, usually $n \gg m$. Moreover, rather than beling linear in (5.7), constraints can be quadratic (e.g., [70]) and convex (e.g., [7]). These make a convex problem that is a potential candidate as a practical application for our constraint-reduced algorithms. It is worthwhile to mention that the solution of problem (5.8)–(5.10) has an exponential form which is strictly positive [22], implying that constraints (5.10) would not be active at the solution, and hence, can be excluded from the working set Q.

Appendix A

Complementary Material

A.1 A proposition

The following proposition was used in Lemma 3.4.

Proposition A.1. *Let* $g \in \mathrm{Range}(G)$ *where* $G \in \mathbb{R}^{p\times l}$ *and* $g \in \mathbb{R}^p$*, then there exists a constant* $C > 0$ *such that for every diagonal matrix* $D > 0$*, any solution of equations*

$$Gv = g, \tag{A.1}$$

$$D^{-1}v = G^T w \tag{A.2}$$

satisfies $\|v\| \le C\|g\|$.

Proof. With QR factorization, G can be written into the form

$$G = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \tag{A.3}$$

where $Q \in \mathbb{R}^{p\times p}$ is orthogonal, and $R_1 \in \mathbb{R}^{R_g \times l}$ with $R_g = \mathrm{rank}(G)$ has full row rank. Left-multiplying Q^T on both sides of (A.1), we get

$$R_1 v = g_1. \tag{A.4}$$

where g_1 is the first R_g elements of $Q^T g$. Denote w_1 the first l elements of $Q^T w$, then left-multiplying D on both sides of (A.2) and substituting (A.3) yields

$$v = D[R_1^T \ 0]Q^T w = DR_1^T w_1. \tag{A.5}$$

Substituting (A.5) into (A.4) and solving w_1, we obtain

$$w_1 = (R_1 D R_1^T)^{-1} g_1.$$

It follows from (A.5) that

$$v = D R_1^T (R_1 D R_1^T)^{-1} g_1.$$

Since $D > 0$, the claim follows from Theorem 1 of [55]. □

A.2 Description of the model used in RHC-based altitude control

In the rotorcraft altitude control problem (section 4.2 and section 4.3.1), we used the discretized linear model as follows:

$$A_{\mathrm{s}} = \begin{bmatrix} 0.9950 & -0.0044 & -0.0004 & 0.0002 & 0.0010 & 0.0156 & -0.0001 & 0 \\ -0.0000 & 0.9956 & -0.0001 & -0.0000 & 0.0002 & 0.0031 & -0.0000 & 0 \\ -0.0613 & 0.0229 & 0.9896 & 0.0103 & -0.0000 & -0.0003 & 0.0000 & 0 \\ 0.3322 & 0.0003 & 0.0872 & 0.8995 & 0.0002 & 0.0018 & -0.0000 & 0 \\ -0.1283 & 0.0002 & 0.0000 & -0.0000 & 0.9245 & 27.7228 & -0.1954 & 0 \\ -0.0083 & 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.8613 & -0.0126 & 0 \\ 0.0100 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0001 & 1.0000 & 0 \\ 0.0003 & -0.0001 & -0.0099 & -0.0001 & 0.0000 & 0.0000 & -0.0000 & 1.0 \end{bmatrix},$$

$$B_{\mathrm{s}} = \begin{bmatrix} -0.0108; & -0.0012; & -0.1626; & 0.0687; & 0.0483; & 0.0000; & -0.0001; & 0.0008 \end{bmatrix}.$$

Appendix B

An ℓ_∞ Version of Iteration IrPDIP

Iteration IrPDIP-ℓ_∞ is the same as Iteration IrPDIP but with an ℓ_∞ exact penalty function. Its analysis is similar to that of Iteration IrPDIP in section 2.2.

B.1 Description of Iteration IrPDIP-ℓ_∞

Applying constraint reduction to problem (2.63), we solve the pair

$$\min\,(c^Q)^T x^Q \quad \text{s.t.} \quad A^Q x^Q = b,\; e^T x^Q + u = \rho e,$$
$$\text{and} \qquad x^Q \geq 0,\; u \geq 0.$$

and

$$\max\; b^T y - \rho z \qquad \text{s.t.} \quad z \geq 0,$$
$$\text{and} \quad (A^Q)^T y - ze \leq c^Q.$$

Based on this pair, the affine-scaling search direction $\left(\Delta x^Q, \Delta u, \Delta y_a, \Delta z_a, \Delta s^Q\right)$ obtained by PDIPMs solves

$$\begin{bmatrix} 0 & 0 & (A^Q)^T & -e & I \\ A^Q & 0 & 0 & 0 & 0 \\ e^T & 1 & 0 & 0 & 0 \\ S^Q & 0 & 0 & 0 & X^Q \\ 0 & z & 0 & u & 0 \end{bmatrix} \begin{bmatrix} \Delta x^Q \\ \Delta u \\ \Delta y_a \\ \Delta z_a \\ \Delta s^Q \end{bmatrix} = \begin{bmatrix} 0 \\ b - A^Q x^Q \\ \rho - e^T x^Q - u \\ -X^Q s^Q \\ -zu \end{bmatrix}, \tag{B.1}$$

where (y, z) is assumed feasible. Eliminating Δx^Q, Δu and Δs^Q in system (B.1), we get normal equations

$$\begin{bmatrix} A^Q & 0 \\ -e^T & -1 \end{bmatrix} \begin{bmatrix} X^Q & 0 \\ 0 & u \end{bmatrix} \begin{bmatrix} S^Q & 0 \\ 0 & z \end{bmatrix}^{-1} \begin{bmatrix} A^Q & 0 \\ -e^T & -1 \end{bmatrix}^T \begin{bmatrix} \Delta y_a \\ \Delta z_a \end{bmatrix} = \begin{bmatrix} b \\ -\rho \end{bmatrix}. \tag{B.2}$$

$$\Delta s^Q = -(A^Q)^T \Delta y_a + \Delta z_a e, \tag{B.3}$$

$$\begin{bmatrix} \Delta x^Q \\ \Delta u \end{bmatrix} = - \begin{bmatrix} x^Q \\ u \end{bmatrix} - \begin{bmatrix} S^Q & 0 \\ 0 & z \end{bmatrix}^{-1} \begin{bmatrix} X^Q & 0 \\ 0 & u \end{bmatrix} \begin{bmatrix} \Delta s^Q \\ \Delta z_a \end{bmatrix} \tag{B.4}$$

The infeasible constraint-reduced IPM for LPs with an ℓ_∞ penalty function is stated below.

Iteration IrPDIP-ℓ_∞

Parameters: $\theta > 0, \tau > 0, \alpha > 0$, $\chi > 0$, $\sigma > 1$, $\gamma_i > 0$, for $i = 1, 2, 3, 4$.

Data: Any initial value y; pick an initial value $z > \max\{0, \max\{A^T y - c\}\}$; and initial values $x > 0$, $u > 0$ and $\rho > 0$; $Q \subseteq \mathbf{n}$ such that A^Q has full row rank; $s := c - A^T y + ze$.

Step 1: Computation of search direction:

(i) Obtain the primal-dual affine-scaling direction $\left(\Delta x^Q, \Delta u, \Delta y_a, \Delta z_a, \Delta s^Q\right)$ from (B.1).

(ii) Select $(\Delta y, \Delta z)$ to satisfy

$$b^T \Delta y - \rho \Delta z \geq \theta(b^T \Delta y_a - \rho \Delta z_a), \quad \|[\Delta y; \Delta z]\| \leq \tau \|[\Delta y_a; \Delta z_a]\|. \tag{B.5}$$

Step 2. Updates.

(i) Update dual variables by choosing a stepsize $t \in (0, 1]$ such that

$$s_+ = c - A^T y_+ + z_+ e > 0, \; z_+ > 0 \tag{B.6}$$

where

$$y_+ = y + t\Delta y,\ z_+ = z + t\Delta z. \tag{B.7}$$

(ii) Select $[x_+; u_+] > 0$ to satisfy

$$\|[x_+; u_+]\| \le \max\left(\|[x; u]\|, \alpha\|[\tilde{x}^Q; \tilde{u}]\|, \chi\right) \tag{B.8}$$

where

$$\begin{aligned} \tilde{x}^Q &:= x^Q + \Delta x^Q, && \text{(B.9)} \\ \tilde{u} &:= u + \Delta u. && \text{(B.10)} \end{aligned}$$

(iii) Pick $Q_+ \subseteq \mathbf{n}$ such that A^{Q_+} is full row rank.

(iv) The penalty parameter: check the two cases (2.65)–(2.66). If either case is satisfied, set

$$\rho_+ = \sigma\rho; \tag{B.11}$$

otherwise $\rho_+ = \rho$.

B.2 Convergence Analysis of Iteration IrPDIP-ℓ_∞

In this section, we show that the penalty parameter in algorithm IrPDIP-ℓ_∞ will be increased at a finite number of iterations under same assumptions as in ℓ_1 version.

Lemma B.1. *Step 1 (i) of IrPDIP-ℓ_∞ is well defined and $b^T\Delta y - \rho\Delta z > 0$.*

Proof. In view of the full-rankness of A^Q specified in IrPDIP-ℓ_∞, matrix $\begin{bmatrix} A^Q & 0 \\ -e^T & -1 \end{bmatrix}$ has full row rank. Therefore, we have from (B.2) that

$$b^T \Delta y_a - \rho \Delta z_a > 0.$$

The claim then follows immediately from (B.5).

□

Hence, Iterate IrPDIP-ℓ_∞ can be repeated indefinitely, generating an infinite sequence of iterates with the dual sequence feasible for problem (2.63).

Our goal is to show that ρ is increased finitely many times. First, Lemma B.2 shows that sequence $\{z_k\}$ is bounded.

Lemma B.2. *Suppose* (P) *is feasible, then the sequence* $\{z_k\}$ *is bounded, and* $\{b^T y_k\}$ *is bounded above.*

Proof. We first show that $\{z_k\}$ is bounded. If ρ_k is increased finitely many times to a finite value, say ρ_∞, then condition (2.65) must fail for k large enough, which implies that $z_k \leq \gamma_1 \frac{z_0}{\rho_0} \rho_\infty$ for k large enough, proving the claim. It remains to prove that $\{z_k\}$ is bounded when ρ_k is increased infinitely many times, i.e., when $\rho_k \to \infty$ as $k \to \infty$.

By assumption that (P) has a feasible point, say x^0, it follows that

$$Ax^0 = b, \quad x^0 \geq 0. \tag{B.12}$$

Since $\rho_k \to \infty$ as $k \to \infty$, there exists k_0 such that

$$\rho_k - \|x^0\|_1 > 0, \forall k \geq k_0. \tag{B.13}$$

Since $(y_k,\, z_k)$ is dual feasible for all k, we have

$$A^T y_k \;\leq\; z_k e + c, \tag{B.14}$$

$$z_k \;\geq\; 0. \tag{B.15}$$

Left-multiplying by $(x^0)^T \geq 0$ on both sides of (B.14) yields

$$(x^0)^T A^T y_k \leq z_k \|x^0\|_1 + c^T x^0.$$

It follows from (B.12) that

$$b^T y_k \leq z_k \|x^0\|_1 + c^T x^0. \tag{B.16}$$

Adding $\rho_k z_k$ to both sides of (B.16), we get

$$(\rho_k - \|x^0\|_1) z_k \leq \pi_k + \rho_k z_k, \tag{B.17}$$

where we have defined

$$\pi_k = c^T x^0 - b^T y_k. \tag{B.18}$$

In view of (B.13) and (B.15), z_k is upper bounded from (B.17) by

$$0 \leq z_k \leq \frac{\pi_k + \rho_k z_k}{\rho_k - \|x^0\|_1} =: \nu_k,\ \forall i,$$

Hence, in order to show that $\{z_k\}$ is bounded, it is sufficient to prove that $\{\nu_k\}$ is bounded above. We show next that $\nu_{k+1} \leq \nu_k$, $\forall k \geq k_0$, proving the boundedness of $\{\nu_k\}$.

Note that for each k, Lemma B.1 implies from (B.7) that

$$b^T y_{k+1} - \rho_k z_{k+1} = b^T y_k - \rho_k z_k + t_k (b^T \Delta y_k - \rho_k \Delta z_k) \geq b^T y_k - \rho_k z_k.$$

It follows from (B.13) that

$$\nu_k = \frac{\pi_k + \rho_k z_k}{\rho_k - \|x^0\|_1} \geq \frac{\pi_{k+1} + \rho_k z_{k+1}}{\rho_k - \|x^0\|_1}, \ \forall k \geq k_0. \tag{B.19}$$

Since

$$\nu_{k+1} = \frac{\pi_{k+1} + \rho_{k+1} z_{k+1}}{\rho_{k+1} - \|x^0\|_1}, \ \forall k \geq k_0,$$

and $\rho_{k+1} > \rho_k$, in order to prove that ν_k is decreasing for $k \geq k_0$, it is sufficient to show that the function g given by

$$g(\rho) := \frac{\pi_{k+1} + \rho z_{k+1}}{\rho - \|x^0\|_1}$$

has a nonpositive derivative $g'(\rho)$. Indeed, since

$$\begin{aligned}
\pi_{k+1} + z_{k+1}\|x^0\|_1 &= c^T x^0 - b^T y_{k+1} + z_{k+1}\|x^0\|_1 && \text{(using (B.18))} \\
&= (x^0)^T c - (x^0)^T A^T y_{k+1} + z_{k+1}\|x^0\|_1 && \text{(using (B.12))} \\
&\geq -(x^0)^T z_{k+1} e + z_{k+1}\|x^0\|_1 && \text{(using (B.12) and (B.14))} \\
&= 0,
\end{aligned}$$

it is readily checked using (B.13) that

$$g'(\rho) = -\frac{\pi_{k+1} + z_{k+1}\|x^0\|_1}{(\rho - \|x^0\|_1)^2} \leq 0,$$

proving the first claim. The second claim follows immediately from (B.16). □

Lemma B.3. *There exists a constant $C > 0$ such that*

$$\left\| [\tilde{x}_k^{Q_k}; \tilde{u}_k; x_k; u_k] \right\| \ \leq \ C\rho_k. \tag{B.20}$$

Proof. It suffices to show the divided two parts

$$\|[\tilde{x}_k^{Q_k}; \tilde{u}_k]\| \leq \frac{C}{2}\rho_k, \ \|[x_k; u_k]\| \leq \frac{C}{2}\rho_k. \tag{B.21}$$

With the definition in (B.9) and (B.10), substituting (B.3) into (B.4), we have

$$\begin{bmatrix} \tilde{x}_k^{Q_k} \\ \\ \tilde{u}_k \end{bmatrix} = \begin{bmatrix} X_k^{Q_k}\left(S_k^{Q_k}\right)^{-1} & 0 \\ \\ 0 & \frac{u_k}{z_k} \end{bmatrix} \begin{bmatrix} A^{Q_k} & 0 \\ \\ -e^T & -1 \end{bmatrix}^T \begin{bmatrix} \Delta y_{a,k} \\ \\ \Delta z_{a,k} \end{bmatrix}. \tag{B.22}$$

Solving (B.2) and substituting its solution $[\Delta y_{a,k}; \Delta z_{a,k}]$ into (B.22), we get

$$\begin{bmatrix} \tilde{x}_k^{Q_k} \\ \\ \tilde{u}_k \end{bmatrix} = H_k \begin{bmatrix} b \\ \\ -\rho_k \end{bmatrix} \tag{B.23}$$

where

$$H_k = \begin{bmatrix} X_k^{Q_k}\left(S_k^{Q_k}\right)^{-1} & 0 \\ \\ 0 & \frac{u_k}{z_k} \end{bmatrix} \begin{bmatrix} A^{Q_k} & 0 \\ \\ -e^T & -1 \end{bmatrix}^T \left(\begin{bmatrix} A^{Q_k} & 0 \\ \\ -e^T & -1 \end{bmatrix} \begin{bmatrix} X_k^{Q_k}\left(S_k^{Q_k}\right)^{-1} & 0 \\ \\ 0 & \frac{u_k}{z_k} \end{bmatrix} \begin{bmatrix} A^{Q_k} & 0 \\ \\ -e^T & -1 \end{bmatrix}^T \right)^{-1}.$$

Because diagonal matrices $X_k^{Q_k}$ and $S_k^{Q_k}$ are positive definite, and u_k and z_k are positive for all k, it follows from [55] that the sequence H_k is bounded. Therefore, (B.23) implies that there must exist $C' > 0$, $C'' > 0$ and $C > 0$, all of which are independent of k, such that

$$\left\| \begin{bmatrix} \tilde{x}_k^{Q_k} \\ \\ \tilde{u}_k \end{bmatrix} \right\| \leq C' \left\| \begin{bmatrix} b \\ \\ -\rho_k \end{bmatrix} \right\| \leq C''\rho_k \leq \frac{C}{2}\rho_k, \ \forall k, \tag{B.24}$$

proving part of (B.21). Now without loss of generality, we suppose $\frac{C}{2} \geq \max\{1, \alpha\}C'' \geq \frac{\max(\|[x_0;u_0]\|, \chi)}{\rho_0}$, where α is a parameter in iteration IrPDIP-ℓ_∞. To show that $\|[x_k; u_k]\| \leq \frac{C}{2}\rho_k$, it suffices to show $\|[x_k; u_k]\| \leq \alpha C''\rho_k$, which follows by induction. Clearly, it holds at $k = 0$. Suppose $\|[x_k; u_k]\| \leq \alpha C''\rho_k$ at some iterate k. With $\{\rho_k\}$ nondecreasing, we have from (B.8) that

$$\|[x_{k+1}; u_{k+1}]\| \leq \max\{\alpha C''\rho_k, \alpha C''\rho_k, \chi\} \leq \alpha C'' \max\{\rho_k, \rho_0\} \leq \alpha C''\rho_{k+1},$$

finishing the induction. □

If ρ_k goes to infinity as k goes to infinity and (P) is feasible, then Lemma B.2 rules out the possibility that condition (2.65) is satisfied infinitely many times. Therefore, condition (2.66) must happen infinitely many times. The following lemma studies the boundedness property on K_ρ in this case.

Lemma B.4. *Suppose $\rho_k \to \infty$ and* (P) *is feasible, then $\{z_k \tilde{u}_k\}$ and $\{S_k^{Q_k} \tilde{x}_k^{Q_k}\}$ are bounded on K_ρ. If in addition,* (P)–(D) *is primal-dual feasible, then $z_k \to 0$ as $k \to \infty$, $k \in K_\rho$, and if furthermore,* (P) *is strictly feasible, then $\{y_k\}$ is bounded on K_ρ.*

Proof. Since ρ_k goes to infinity on K_ρ, and since Lemma B.2 implies that condition (2.65) is eventually violated, condition (2.66) must be satisfied for $k \in K_\rho$ large enough. In particular, there exists k_0 such that for all $k \geq k_0$, $k \in K_\rho$,

$$\begin{aligned} \|[\Delta y_{a,k}; \Delta z_{a,k}]\| &\leq \frac{\gamma_2}{\rho_k}, && \text{(B.25)} \\ \tilde{x}_k^{Q_k} &\geq -\gamma_3 e. && \text{(B.26)} \end{aligned}$$

Note from the first block row of (B.1) that

$$\Delta s_k^{Q_k} = -(A^{Q_k})^T \Delta y_{a,k} + \Delta z_{a,k}^{Q_k} e.$$

Since A^{Q_k} can only take finitely many values, it follows from (B.25) that there exists certain $\delta > 0$ such that

$$\|\Delta s_k^{Q_k}\| \leq \frac{\delta}{\rho_k}, \; k \geq k_0, k \in K_\rho. \tag{B.27}$$

In view of (B.20) in Lemma B.3, (B.25) and (B.27), we have from the last two block rows in (B.1) that

$$\|z_k\tilde{u}_k\| = \|u_k \Delta z_{a,k}\| \le C\rho_k \cdot \frac{\gamma_2}{\rho_k} = C\gamma_2,\ k \ge k_0,\ k \in K_\rho, \tag{B.28}$$

$$\text{and} \qquad \left\| S_k^{Q_k} \tilde{x}_k^{Q_k} \right\| = \left\| X_k^{Q_k} \Delta s_k^{Q_k} \right\| \le C\rho_k \cdot \frac{\delta}{\rho_k} = C\delta,\ k \ge k_0,\ k \in K_\rho \tag{B.29}$$

proving the first claim. Next, without loss of generality, assuming $\rho_{k_0} > \|x^0\|_1$ where x^0 is a feasible point of (P), we have

$$u_k^0 := \rho_k - e^T x^0 > 0, \quad \text{for } k \ge k_0. \tag{B.30}$$

Since by assumption of that (P)–(D) is feasible, there exist y^0 and $s^0 \ge 0$ associated with $x^0 \ge 0$ and $u_k^0 \ge 0$ such that, for all $k \ge k_0$,

$$\begin{aligned} A^{Q^k}(x^0)^{Q_k} + A^{\overline{Q}_k}(x^0)^{\overline{Q}_k} = Ax^0 &= b, \\ e^T x^0 + u_k^0 &= \rho_k, \\ A^T y^0 + s^0 &= c, \end{aligned}$$

On the hand other, in view of the second and third rows of (B.1), definitions (B.9) and (B.10), and (B.6), we get

$$\begin{aligned} A^{Q_k}\tilde{x}_k^{Q_k} &= b, \\ e^T \tilde{x}_k^{Q_k} + \tilde{u}_k &= \rho_k, \\ A^T y_k + s_k - z_k e &= c. \end{aligned} \tag{B.31}$$

The two groups of equations above yield that

$$\begin{bmatrix} A^{Q_k} & A^{\overline{Q}_k} & 0 \\ e^T & e^T & 1 \end{bmatrix} \begin{bmatrix} (\tilde{x}_k - x^0)^{Q_k} \\ -(x^0)^{\overline{Q}_k} \\ \tilde{u}_k - u_k^0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{B.32}$$

$$\begin{bmatrix} A^{Q_k} & A^{\overline{Q}_k} & 0 \\ e^T & e^T & 1 \end{bmatrix}^T \begin{bmatrix} y^0 - y_k \\ z_k \end{bmatrix} = \begin{bmatrix} (s_k - s^0)^{Q_k} \\ (s_k - s^0)^{\overline{Q}_k} \\ z_k \end{bmatrix}. \tag{B.33}$$

This implies that $[(\tilde{x}_k - x^0)^{Q_k}; -(x^0)^{\overline{Q}_k}; (\tilde{u}_k - u_k^0)]$ is orthogonal to $[(s_k - s^0)^{Q_k}; (s_k - s^0)^{\overline{Q}_k}; z_k]$, i.e.,

$$(\tilde{x}_k^{Q_k})^T (s_k - s^0)^{Q_k} - (x^0)^T (s_k - s^0) + (\tilde{u}_k - u_k^0) z_k = 0. \tag{B.34}$$

Hence, we have for C' large enough

$$\begin{aligned} u_k^0 z_k + (x^0)^T s_k &= (x^0)^T s^0 + (\tilde{x}_k^{Q_k})^T s_k^{Q_k} - (\tilde{x}_k^{Q_k})^T (s^0)^{Q_k} + \tilde{u}_k z_k & \text{(B.35)} \\ &\le (x^0)^T s^0 + C'\delta + \gamma_3 e^T s^0 + C'\gamma_2 & \text{(B.36)} \end{aligned}$$

where equality (B.35) comes from the expansion of (B.34), and the inequality uses (B.29), (B.26), and (B.28). Noting that u_k^0, z_k, x^0 and s_k are all nonnegative for $k \ge k_0$, we get

$$z_k \le \left((x^0)^T s^0 + C'\delta + \gamma_3 e^T s^0 + C'\gamma_2 \right) / u_k^0, \ \forall i, \ k \ge k_0, \ k \in K_\rho. \tag{B.37}$$

Since $u_k^0 \to \infty$ as $k \to \infty$ on K_ρ from definition (B.30), this proves that

$$\lim_{k \to \infty, k \in K_\rho} z_k \to 0.$$

Furthermore, if (P) is strictly feasible, then we can select $x^0 > 0$ and

$$s_k^i \leq \left((x^0)^T s^0 + C'\delta + \gamma_3 e^T s^0 + C'\gamma_2\right)/(x^0)^i, \ \forall i,\ k \geq k_0,\ k \in K_\rho, \tag{B.38}$$

proving that $\{s_k\}$ is bounded on K_ρ. Boundednesses of $\{s_k\}$ and $\{z_k\}$, together with (B.31) and full-rankness of A, imply that $\{y_k\}$ is bounded on K_ρ.

□

The following proposition proves that ρ_k is increased at most finitely many times.

Proposition B.1. *Suppose* (P)–(D) *is strictly feasible, then ρ_k is increased at most finitely many times, i.e., K_ρ is finite.*

Proof. By contradiction, suppose K_ρ is infinite. Then in view of Lemma B.2, there must exist $k_0 > 0$ and an infinite index set $K \subseteq K_\rho$ such that condition (2.66) is satisfied for $k \geq k_0$, $k \in K$; in particular,

$$\begin{aligned} \tilde{x}_k^Q &\geq -\gamma_3 e, \quad k \geq k_0,\ k \in K. && \text{(B.39)} \\ \tilde{u}_k &\leq \gamma_4, \quad k \geq k_0,\ k \in K, && \text{(B.40)} \end{aligned}$$

where without loss of generality, we assume $Q_k = Q$, $\forall k \in K$ for some Q. Moreover, we have from Lemma B.4, for certain constant $C > 0$,

$$\begin{aligned} \|S_k^Q \tilde{x}_k^Q\| &\leq C, \quad \forall k \in K, && \text{(B.41)} \\ \|y_k\| &\leq C, \quad \forall k \in K, && \text{(B.42)} \\ \lim_{k\to\infty} z_k &= 0, \quad k \in K. && \text{(B.43)} \end{aligned}$$

Now, since $\lim_{k\to\infty} \rho_k = \infty$, we have from the third block row of (B.1) and inequality (B.40) that

$$\lambda_k := e^T \tilde{x}_k^Q = \rho_k - \tilde{u}_k \to \infty, \quad \text{as} \quad k \to \infty,\, k \in K.$$

It follows that

$$e^T \bar{x}_k^Q = 1, \quad k \geq k_0,\, k \in K \tag{B.44}$$

where we define

$$\bar{x}_k^Q = \frac{\tilde{x}_k^Q}{\lambda_k}, \quad \forall k \in K. \tag{B.45}$$

Thus, (B.44) and (B.39) imply that $\bar{x}_k^Q$ is bounded below on K. This further implies from (B.44) that $\bar{x}_k^Q$ is bounded on K. So in view of (B.44), (B.42) and (B.43), there exists an infinite index set $K' \subseteq K$ such that

$$\bar{x}_k^Q \to \bar{x}_*^Q \neq 0,\, y_k \to y_*,\, z_k \to z_* = 0, \quad \text{as} \quad k \to \infty, k \in K' \tag{B.46}$$

Next, dividing by λ_k and taking the limit on both sides of (B.41), we have from (B.45)

$$S_k^Q \bar{x}_k^Q \to 0, \text{ as } k \to \infty,\, k \in K'$$

which implies that

$$\bar{x}_*^i = 0, \ \forall i \in Q \backslash I(y_*).^1 \tag{B.47}$$

On the other hand, the second block equation in (B.1) and (B.9) give

$$A^Q \tilde{x}_k^Q = b, \forall k.$$

Dividing by λ_k and taking the limit of both sides, using (B.47), we have

$$\sum_{i \in I(y_*) \cap Q} \bar{x}_*^i a^i = 0. \tag{B.48}$$

[1] Since $z_* = 0$, $I(y_*, z_*) = I(y_*)$ and $y_* \in \mathcal{F}$.

Now note from (B.39) and (B.45) that

$$\bar{x}_*^Q = \lim_{k\to\infty, k\in K'} \frac{\tilde{x}_k^Q}{\lambda_k} \geq \lim_{k\to\infty, k\in K'} \frac{-\gamma_3 e}{\lambda_k} = 0. \tag{B.49}$$

Since the strict feasibility of (D) implies positively linear independence of vectors $\{a^i : i \in I(y_*) \cap Q,\ y_* \in \mathcal{F}\}$, it follows from (B.48) and (B.49) that

$$\bar{x}_*^i = 0,\ \forall i \in I(y_*) \cap Q.$$

Together with (B.47), we therefore have

$$\bar{x}_*^Q = 0,$$

which is a contradiction to (B.46). □

B.3 Infeasible constraint-reduced MPC: IrMPC-ℓ_∞

To give a simple example of IrPDIP-ℓ_∞, we do the following specifications of [65] for algorithm IrPDIP-ℓ_∞ and name the resulting full algorithm by IrMPC-ℓ_∞.

$\boxed{\text{Iteration IrMPC-}\ell_\infty}$

- Perform Step 1 (i) in algorithm IrPDIP-ℓ_∞.
- Computation of search direction (corresponding to Step 1 (ii) in IrPDIP-ℓ_∞)

$$(\Delta y, \Delta z) = (\Delta y_a, \Delta z_a).$$

- Updates (corresponding to Step 2 (i) and (ii) in IrPDIP-ℓ_∞)

Compute

$$\begin{aligned}
\Delta s^{\overline{Q}} &= -(A^{\overline{Q}})^T \Delta y + \Delta z e, \\
\hat{t}^{\mathrm{d}} &= \arg\max\{t \in [0,1] \,|\, s + t\Delta s \geq 0, z + t\Delta z \geq 0\}; \\
\hat{t}^{\mathrm{p}} &= \arg\max\{t \in [0,1] \,|\, x^Q + t\Delta x^Q \geq 0, u + t\Delta u \geq 0\}.
\end{aligned}$$

Set step sizes

$$\begin{aligned}
t^{\mathrm{d}} &= \max\{\beta \hat{t}^{\mathrm{d}}, \hat{t}^{\mathrm{d}} - \|[\Delta y_a; \Delta z_a]\|\}; \\
t^{\mathrm{p}} &= \max\{\beta \hat{t}^{\mathrm{p}}, \hat{t}^{\mathrm{p}} - \|[\Delta y_a; \Delta z_a]\|\}.
\end{aligned}$$

(i) Dual variables: set

$$\begin{aligned}
(\hat{x}^Q, \hat{u}) &= (x^Q, u) + t^{\mathrm{p}}(\Delta x^Q, \Delta u); \\
(y_+, s_+, z_+) &= (y, s, z) + t^{\mathrm{d}}(\Delta y, \Delta s, \Delta z).
\end{aligned}$$

(ii) Primal variables: set

$$u_+ = \max\{\min\{\| [\Delta y_a; \Delta z_a] \|^2 + \| [\tilde{u}_-; \tilde{x}_-] \|^2, w_{\min}\}, \hat{u}\};$$

$$x_+^i = \max\{\min\{\| [\Delta y_a; \Delta z_a] \|^2 + \| [\tilde{u}_-; \tilde{x}_-] \|^2, w_{\min}\}, \hat{x}^i\},\ i \in Q,$$

and

$$x_+^i = \min\left\{ \frac{\mu_+^Q}{s_+^i}, \chi \right\},\ i \notin Q$$

where

$$\mu_+^Q = \frac{(x_+^Q)^T s_+^Q + u_+ z_+}{|Q| + 1}.$$

- Select Q as in section 2.2 of [65].

- Perform (iv) in algorithm IrPDIP-ℓ_∞.

With those specifications above, after finitely many iterations, IrMPC-ℓ_∞ reduced to rMPC* of [65] with $\psi = 0$ applied to problem (2.63) with $\rho = \bar{\rho}$. Thus, we can have the results directly from [65] under necessary assumptions.

Proposition B.2. *Suppose* (P)-(D) *is strictly feasible. Then* $\{(y_k, z_k)\}$ *converges to a stationary point* (y_*, z_*) *of problem* (2.63) *with* $\rho = \bar{\rho}$.

Proof. Proposition B.1 implies that $\rho_k = \bar{\rho}$ for sufficiently large k. When ρ_k is constant, IrMPC-ℓ_∞ reduces to algorithm rMPC* of [65] with $\psi = 0$. Hence, it follows from Theorem 3.8 in [65] that $\{(y_k, z_k)\}$ converges to a stationary point of problem (2.63) if and only if the penalized dual objective function is bounded. We next show the dual penalized function is bounded indeed. Since Lemma B.1 implies that $\{b^T y_k - \rho_k z_k\}$ is increasing for k large enough such that $\rho_k = \bar{\rho}$, it suffices to show that $\{b^T y_k - \rho_k z_k\}$ is bounded above. Indeed, since $\{b^T y_k\}$ is bounded above by Lemma B.2, this claim follows from the boundedness of $\{z_k\}$ and $\{\rho_k\}$ from Lemma B.2 and Proposition B.1 respectively.

□

We next establish that $\{z_k\}$ converges to zero, and thus that $\{y_k\}$ converges to an optimal solution of (2.63). To that end, we need the following lemma whose proof is trivial and hence omitted.

Lemma B.5. *Suppose for all* $y \in \mathbb{R}^m$, $\{a^i : (a^i)^T y = c^i\}$ *is a linearly independent set of vectors. Then for all* $(y, z) \in \mathcal{F}_\rho$, *the gradients of the active constraints of problem* (2.63) *are linearly independent vectors.*

Theorem B.1. *Suppose* (P)-(D) *is strictly feasible and, for all* $y \in \mathbb{R}^m$, $\{a^i : (a^i)^T y = c^i\}$ *is a linearly independent set of vectors. Then* K_ρ *is finite,* $z_k \to 0$ *and* $\{y_k\}$ *converges to an optimal solution of problem* (D).

Proof. Proposition B.1 proved ρ_k is increased finitely many time to $\bar{\rho}$, so K_ρ is finite. Lemma B.5 implies the gradients of the active constraints of problem (2.63) are linearly independent. With these in hand, applying the second part of Theorem 3.8 in [65], we conclude that (y_k, z_k) converges to an optimal value (y_*, z_*) of problem (2.63) where $\rho = \bar{\rho}$. Next, Proposition 3.9 of [65] states that there exists an infinite subsequence K on which $[\tilde{x}_k; \tilde{u}_k]$ converges to an optimal solution $[\tilde{x}_*; \tilde{u}_*]$ of problem (2.64) with $\rho = \bar{\rho}$ and on which

$$[\Delta y_{a,k}; \Delta z_{a,k}] \to 0, \text{ as } k \to \infty,\ k \in K.$$

Thus the conditions (i) and (ii) in condition (2.66) are satisfied on K. On the other hand, since $\rho_k = \bar{\rho}$ for $k \in K$ large enough, one condition in condition (2.66) must fail, and thus we have $\tilde{u}_k \geq \gamma_4$ for $k \in K$ large enough. Hence

$$\tilde{u}_* \geq \gamma_4.$$

Complementary slackness implies that $z_* = 0$. It follows that y_* is an optimal solution of problem (D). □

Appendix C

An ℓ_∞ Version of IrQP

For problem (Pq)–(Dq), Iteration IrQP-ℓ_∞ is the same as Iteration IrQP but with an ℓ_∞ exact penalty function. Its analysis is similar to that of Iteration IrQP in chapter 3.

C.1 Description of Iteration IrQP-ℓ_∞

Instead of the ℓ_1 exact penalty function used in (Pq$_\rho$)–(Dq$_\rho$), we can use an ℓ_∞ exact penalty function and consider the relaxed problem

$$\left.\begin{array}{ll} \max\limits_{y,z} & f(y) - \rho z \\ \text{s.t.} & A^T y - ze \le c,\ z \ge 0 \end{array}\right\} \qquad (\mathrm{Dq}_\rho oo)$$

with its associated primal problem

$$\left.\begin{array}{ll} \max\limits_{y,x,u} & c^T x + \frac{1}{2} y^T H y \\ \text{s.t.} & Hy + Ax - b = 0, \\ & e^T x + u = \rho, \\ & x \ge 0,\ u \ge 0, \end{array}\right\} \qquad (\mathrm{Pq}_\rho oo)$$

where z and u are both scalars. Strictly feasible points are readily available for penalized problem (Dq$_\rho oo$) by selecting z large enough with any given y.

The scheme of choosing the penalty parameter for this ℓ_∞ version is akin to scheme (3.11)–(3.12) in the ℓ_1 version. As z and u are scalar variables, the adjustment scheme can be slightly simplified as: Increase ρ at each iteration when either

$$z \geq \gamma_1 \frac{z_0}{\rho_0} \rho \tag{C.1}$$

OR

$$\text{(i) } \|[\Delta y; \Delta z]\| \leq \gamma_2, \text{ AND (ii) } \tilde{x}^Q \geq -\gamma_3 e, \text{ AND (iii) } \tilde{u}^Q < \gamma_4 \tag{C.2}$$

is satisfied, where again, γ_i, $i = 1, 2, 3, 4$ are positive parameters.

Applying constraint reduction schemes to problem (Dq$_\rho oo$), we solve the reduced problem

$$\max_{y,z} \quad f(y) - \rho z \tag{C.3}$$

$$\text{s.t.} \quad (A^Q)^T y - ze \leq b^Q, \; z \geq 0. \tag{C.4}$$

The affine-scaling direction $(\Delta x^Q, \Delta u, \Delta y, \Delta z, \Delta s^Q)$ can be obtained by solving the linear system

$$\begin{bmatrix} 0 & 0 & (A^Q)^T & -e & 0 & I \\ A^Q & 0 & H & 0 & 0 & 0 \\ e^T & 1 & 0 & 0 & 0 & 0 \\ S^Q & 0 & 0 & 0 & 0 & X^Q \\ 0 & z & 0 & u & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x^Q \\ \Delta u \\ \Delta y \\ \Delta z \\ \Delta s^Q \end{bmatrix} = \begin{bmatrix} 0 \\ b - A^Q x^Q - Hy \\ \rho - e^T x^Q - u \\ -X^Q s^Q \\ -zu \end{bmatrix}. \tag{C.5}$$

After block Gaussian elimination, this system can be written to normal equations

$$M^{(Q)} \begin{bmatrix} \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} b - Hy \\ -\rho \end{bmatrix}, \tag{C.6}$$

$$\Delta s^Q = -(A^Q)^T \Delta y + \Delta z e, \tag{C.7}$$

$$\begin{bmatrix} \Delta x^Q \\ \Delta u \end{bmatrix} = - \begin{bmatrix} x^Q \\ u \end{bmatrix} - \begin{bmatrix} (S^Q)^{-1} X^Q & 0 \\ 0 & z^{-1} u \end{bmatrix} \begin{bmatrix} \Delta s^Q \\ \Delta z \end{bmatrix} \tag{C.8}$$

where

$$M^{(Q)} = \begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} A^Q & 0 \\ -e^T & -1 \end{bmatrix} \begin{bmatrix} (S^Q)^{-1} \Lambda^Q & 0 \\ 0 & z^{-1} u \end{bmatrix} \begin{bmatrix} A^Q & 0 \\ -e^T & -1 \end{bmatrix}^T . \tag{C.9}$$

The dominant cost of computing $\left(\Delta x^Q, \Delta u, \Delta y, \Delta z, \Delta s^Q\right)$ is to solve (C.6), which is dominated by forming matrix $M^{(Q)}$, taking $|Q|(m+1)^2$ flops, enjoying the same speed up in *feasible* constraint-reduced IPMs.

We are now ready to state Iteration IrQP-ℓ_∞, an ℓ_∞ version of Iteration IrQP.

Iteration IrQP-ℓ_∞

Parameters: $\beta \in (0, 1)$, $\sigma > 1$, $\gamma_i > 0$, for $i = 1, 2, 3, 4$; $w_{\min} > 0$ and $\chi > 0$.

Data: $y \in \mathbb{R}^m$ and $z \in \mathbb{R}$ such that $z > \max\{\max\{0, A^T y - c\}\}$; $s := c - A^T + ze$; $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ and $\rho \in \mathbb{R}$ such that $x > 0$, $u > 0$ and $\rho > 0$; $Q \subseteq \mathbf{n}$ such that $[H \;\; A^Q]$ is full rank.

Step 1: Computation of the search direction.

(i). Obtain $\left(\Delta x^Q, \Delta u, \Delta y, \Delta z, \Delta s^Q\right)$ by solving (C.6)–(C.8). Compute

$$\Delta s^{\overline{Q}} = -(A^{\overline{Q}})^T \Delta y + e \Delta z. \tag{C.10}$$

Set

$$\tilde{x}^i = \begin{cases} x^i + \Delta x^i, & i \in Q, \\ 0, & i \notin Q. \end{cases} \tag{C.11}$$

$$\tilde{u} = u + \Delta u, \tag{C.12}$$

$$\tilde{y} = y + \Delta y. \tag{C.13}$$

and set

$$\tilde{x}_- := \min\{\tilde{x}, 0\}, \quad \tilde{u}_- := \min\{\tilde{u}, 0\}. \tag{C.14}$$

(ii). Compute

$$\hat{t} = \arg\max\{\bar{t} \in [0,1] \mid s + \bar{t}\Delta s \geq 0, z + \bar{t}\Delta z \geq 0\}; \tag{C.15}$$

Set step sizes

$$t = \max\{\beta\hat{t}, \hat{t} - \|[\Delta y; \Delta z]\|\}; \tag{C.16}$$

Step 2. Updates.

(i). Dual variables: set

$$(y_+, s_+, z_+) = (y, s, z) + t(\Delta y, \Delta s, \Delta z). \tag{C.17}$$

(ii). Primal variables: set

$$u_+^i = \max\{\max\{\min\{\|[\Delta y; \Delta z]\|^2 + \|\,[\tilde{u}_-; \tilde{x}_-]\,\|^2, w_{\min}\}, \tilde{u}^i\}, \chi\} \quad i \in \mathbf{n}; \tag{C.18}$$

$$x_+^i = \max\{\max\{\min\{\|[\Delta y; \Delta z]\|^2 + \|\,[\tilde{u}_-; \tilde{x}_-]\,\|^2, w_{\min}\}, \tilde{x}^i\}, \chi\} \quad i \in \mathbf{n}. \tag{C.19}$$

(iii). The penalty parameter: If either (C.1) or (C.2) is satisfied, set

$$\rho_+ = \sigma\rho \tag{C.20}$$

and

$$\mu = \frac{x^T s + u^T z}{2n}, \quad x^i = \frac{\mu}{s^i}, \quad u^i = \frac{\mu}{z^i} \quad \forall i \in \mathbf{n};$$

otherwise set $\rho_+ = \rho$.

(iv) The working set: select Q_+ such that $[H \; A^{Q_+}]$ has full rank.

□

It is clear from Iteration IrQP-ℓ_∞ that $x_+ > 0$, $u_+ > 0$, $z_+ > \max\{\max\{0, Ay_+ - c\}\}$, and (see (C.22) below) $s_+ = c - A^T y_+ + z_+ e$. Further, since $[H \; A^Q]$ has full rank and $(x, u, s, z) > 0$, $M^{(Q)}$ (see (C.9)) is positive definite, so the search direction $(\Delta y, \Delta z)$ in step 1 (i) of IrQP-ℓ_∞ is well defined. Hence, Iteration IrQP can be repeated indefinitely, generating an infinite sequence of iterates.

Also, note that equations (C.7) and (C.10) imply

$$A^T \Delta y_k - e\Delta z_k + \Delta s_k = 0 \quad \forall k. \tag{C.21}$$

Since $s_0 = c - A^T y_0 + z_0 e$ (see Data section of Iteration IrQP-ℓ_∞), in view of (C.17), it follows that

$$A^T y_k - z_k e + s_k = c, \quad s_k > 0, \quad z_k > 0, \quad \forall k, \tag{C.22}$$

i.e., the primal sequence $\{(y_k, z_k)\}$ is strictly feasible for (Dq$_\rho oo$).

C.2 Global convergence

In this section, we show that ρ_k generated by Iteration IrQP-ℓ_∞ is increased only finitely many times under mild assumptions. The following lemma states that the objective function of (Dq$_\rho$$oo$) is decreasing when parameter ρ is not updated.

Lemma C.1. *For all k, $f(y_{k+1}) - \rho_k z_{k+1} \geq f(y_k) - \rho_k z_k$.*

Proof. At each iteration k, the update of primal iterates can be viewed as applying on $(Dq_{\rho_k}oo)$ one iteration of Algorithm A of [32]. Thus, the claim follows from Proposition A. 4 (i) of [32] where $\alpha \in (0, 2)$ is substituted by $t \in [0, 1]$ here (see (C.15) and (C.16)). □

Lemma C.2. *Suppose* (Pq) *is feasible, then sequence $\{z_k\}$ is bounded.*

Proof. If ρ_k is increased finitely many times to a finite value, say ρ_∞, then condition (C.1) must fail for k large enough, i.e., $z_k \leq \gamma_1 \frac{z_0}{\rho_0} \rho_\infty$ for k large enough, proving the claim. It remains to prove that $\{z_k\}$ is bounded when ρ_k is increased infinitely many times, i.e., when $\rho_k \to \infty$ as $k \to \infty$.

By assumption that (Pq) has a feasible point, say (x^0, y^0), we have

$$Hy^0 + Ax^0 = b, \;\; x^0 \geq 0. \tag{C.23}$$

Since $\rho_k \to \infty$ as $k \to \infty$, there exists k_0 such that

$$\rho_k > \|x^0\|_1, \, \forall k \geq k_0. \tag{C.24}$$

Since $(y_k,\, z_k)$ is dual feasible for $(Pq_{\rho_k}oo)$ for all k, we have

$$A^T y_k \;\; \leq \;\; c + e z_k, \tag{C.25}$$

$$z_k \;\; \geq \;\; 0. \tag{C.26}$$

Left-multiplying by $(x^0)^T \geq 0$ on both sides of (C.25), using (C.23), yields

$$(b - Hy^0)^T y_k \leq c^T x^0 + z_k \|x^0\|_1. \tag{C.27}$$

Adding $\rho_k z_k$ to both sides of (C.27), after simple reorganization, we get

$$(\rho_k - \|x^0\|_1) z_k \leq c^T x^0 - (b - Hy^0)^T y_k + \rho_k z_k. \tag{C.28}$$

Next, inequality (3.27) implies that

$$-(\frac{1}{2} y_k^T H y_k - (y^0)^T H y_k - c^T x^0) \leq \frac{1}{2} (y^0)^T H y^0 + c^T x^0 := M, \quad \forall k. \tag{C.29}$$

In view of (C.24) and (C.26), inequality (C.28) implies

$$\begin{aligned} 0 \leq z_k &\leq \frac{c^T x_0 - (b - Hy_0)^T y_k + \rho_k z_k}{\rho_k - \|x^0\|_1} \\ &= \frac{(c^T x^0 + (y^0)^T H y_k - \frac{1}{2} y_k^T H y_k) - (b^T y_k - \frac{1}{2} y_k^T H y_k) + \rho_k z_k}{\rho_k - \|x^0\|_1} \\ &\leq \frac{M - f(y_k) + \rho_k z_k}{\rho_k - \|x^0\|_1} =: \nu_k, \end{aligned} \tag{C.30}$$

Hence, in order to show that $\{z_k\}$ is bounded, it suffices to prove that $\{\nu_k\}$ is bounded. We show next that $\nu_{k+1} \leq \nu_k$, $\forall k \geq k_0$. Since ν_k is nonnegative for all k, this proves the boundedness of $\{\nu_k\}$.

In view of (C.24) and (C.30), it is implied from Lemma C.1 that

$$\nu_k = \frac{M - f(y_k) + \rho_k z_k}{\rho_k - \|x^0\|_1} \geq \frac{M - f(y_{k+1}) + \rho_k z_{k+1}}{\rho_k - \|x^0\|_1}, \forall k \geq k_0. \tag{C.31}$$

On the other hand,

$$\nu_{k+1} = \frac{M - f(y_{k+1}) + \rho_{k+1} z_{k+1}}{\rho_{k+1} - \|x^0\|_1}.$$

Since $\rho_{k+1} \geq \rho_k$, in order to conclude that $\nu_k \geq \nu_{k+1}$ for $k \geq k_0$, it is sufficient to verify that the function g given by

$$g(\rho) := \frac{M - f(y_{k+1}) + \rho z_{k+1}}{\rho - \|x^0\|_1}$$

has a nonpositive derivative $g'(\rho)$ for all ρ satisfying (C.24). Note that

$$\begin{aligned}
M - f(y_{k+1}) + \|x^0\|_1 z_{k+1} &\geq c^T x^0 + (y^0)^T H y_{k+1} - \frac{1}{2} y_{k+1}^T H y_{k+1} - f(y_{k+1}) + \|x^0\|_1 z_{k+1} \\
&= c^T x^0 + (y^0)^T H y_{k+1} - b^T y_{k+1} + \|x^0\|_1 z_{k+1} \\
&= (x^0)^T c - (x^0)^T A^T y_{k+1} + \|x^0\|_1 z_{k+1} \\
&\geq -(x^0)^T z_{k+1} + \|x^0\|_1 z_{k+1} \\
&\geq 0,
\end{aligned}$$

where the first inequality comes from (C.29) with k replaced by $k+1$, the first equality from the substitution of $f(y_{k+1})$ by $b^T y_{k+1} - \frac{1}{2} y_{k+1}^T H y_{k+1}$, the second one from (C.23), the second inequality from (C.25) and the non-negativeness of x^0, and the last one from (C.26). It follows from (C.24) that

$$g'(\rho) = -\frac{M - f(y_{k+1}) + \|x^0\|_1 z_{k+1}}{(\rho - \|x^0\|_1)^2} \leq 0,$$

proving $\nu_k \geq \nu_{k+1}$ and hence the boundedness of $\{z_k\}$.

□

The following lemma studies the boundedness property of iterates sequences on K_ρ.

Lemma C.3. *Suppose* (Pq) *is feasible. If $\rho_k \to \infty$, then $\{z_k \tilde{u}_k\}$ and $\{S_k \tilde{x}_k\}$ are bounded on K_ρ. If additionally,* (Dq) *is feasible, then $z_k \to 0$ as $k \to \infty$, $k \in K_\rho$, and if, moreover,* (Pq) *is strictly feasible, then $\{y_k\}$ is bounded on K_ρ.*

Proof. Since (Pq) is feasible, and since ρ_k goes to infinity on K_ρ, Lemma C.2 implies that condition (C.2) must be satisfied for all $k \in K_\rho$ large enough. In particular, there exists an integer k_0 such that for all $k \geq k_0$, $k \in K_\rho$,

$$\|[\Delta y_k; \Delta z_k]\| \leq \gamma_2, \tag{C.32}$$

and

$$\tilde{x}_k^{Q_k} \geq -\gamma_3 e. \tag{C.33}$$

In view of (C.21), inequality (C.32) implies that $\{\Delta s_k\}$ is bounded on K_ρ. Thus, with boundedness of x_k and u_k (enforced by Iteration IrQP-ℓ_∞) and boundedness of Δz_k and Δs_k on K_ρ, it follows from (C.8) and definitions (C.11)–(C.12) that for C large enough,

$$|z_k \tilde{u}_k| = |u_k \Delta z_k| \leq C, \quad k \geq k_0,\, k \in K_\rho, \tag{C.34}$$

and

$$\left\| S_k^{Q_k} \tilde{x}_k^{Q_k} \right\| = \left\| X_k^{Q_k} \Delta s_k^{Q_k} \right\| \leq C, \quad k \geq k_0,\, k \in K_\rho, \tag{C.35}$$

which proves the first claim that $\{z_k \tilde{u}_k\}$ on K_ρ and from (C.11) that $\{S_k \tilde{x}_k\}$ are bounded on K_ρ.

Now, by assumption that (Pq)–(Dq) is primal-dual feasible, there exist (x^0, y^0, s^0) such that

$$A^T y^0 + s^0 = c, \quad Ax^0 = b - Hy^0, \quad [x^0; s^0] \geq 0. \tag{C.36}$$

Without loss of generality, we assume that $\rho_{k_0} > \|x^0\|_1$, so that

$$u_k^0 := \rho_k - e^T x^0 > 0, \text{ for } k \geq k_0. \tag{C.37}$$

On the other hand, in view of definitions (C.11), (C.12) and (C.13), equation (C.21) and the first four block equations (C.5) imply that, for all k,

$$A^T y_k - z_k e + s_k = c,\ A\tilde{x}_k = b - H\tilde{y}_k,\ e^T \tilde{x}_k + \tilde{u}_k = \rho_k. \tag{C.38}$$

Equations (C.36)–(C.38) yield that

$$\begin{aligned} A^T(y^0 - y_k) + z_k e + s^0 - s_k &= 0, && \text{(C.39)} \\ A(x^0 - \tilde{x}_k) &= -H(y^0 - \tilde{y}_k), && \text{(C.40)} \\ e^T(x^0 - \tilde{x}_k) + (u_k^0 - \tilde{u}_k) &= 0. && \text{(C.41)} \end{aligned}$$

It follows that, for all k,

$$\begin{aligned} (s^0 - s_k)^T(x^0 - \tilde{x}_k) - z_k(u_k^0 - \tilde{u}_k) &= (s^0 - s_k + z_k e)^T(x^0 - \tilde{x}_k) \\ &= -(y^0 - y_k)^T A(x^0 - \tilde{x}_k) \end{aligned}$$

where the first equality comes from (C.41) and the second one from (C.39). Hence, from (C.40), we obtain

$$(s^0 - s_k)^T(x^0 - \tilde{x}_k) - z_k(u_k^0 - \tilde{u}_k) = (y^0 - y_k)^T H(y^0 - \tilde{y}_k). \tag{C.42}$$

It follows that

$$\begin{aligned} & (y^0 - y_k)^T H(y^0 - \tilde{y}_k) + s_k^T x^0 + z_k u_k^0 \\ = \ & s_k^T \tilde{x}_k + z_k \tilde{u}_k - \tilde{x}_k^T s^0 + (x^0)^T s^0 \\ \leq \ & 2nC + \gamma_3 e^T s^0 + (x^0)^T s^0, \quad \forall k \in K_\rho, && \text{(C.43)} \end{aligned}$$

where the equality comes from the expansion of (C.42), and the inequality from (C.34), (C.35) and (C.33). Now, inequality (3.27) with $y := y^0 - y_k$ and $p := \frac{1}{2}\Delta y_k$ implies that

$$\psi_k := (y^0 - y_k)^T H(y^0 - \tilde{y}_k) = (y^0 - y_k)^T H(y^0 - y_k) - (y^0 - y_k)^T H\Delta y_k \geq -\frac{1}{4}\Delta y_k^T H \Delta y_k$$

where the equality is from (C.13). Note from (C.32) that $\Delta y_k^T H \Delta y_k$ is bounded, so ψ_k is bounded from below. Hence, it follows from (C.43) that there exists $\pi > 0$ large enough such that

$$s_k^T x^0 \leq \pi, \quad \forall k \in K_\rho, \tag{C.44}$$

and

$$z_k u_k^0 \leq \pi.$$

Using (C.37) and the fact that $\rho_k \to \infty$ as $k \to \infty$, it follows from positiveness of z_k and u_k^0 that

$$\lim_{k\to\infty} \sup_{k\in K_\rho} z_k \leq \frac{\pi}{\lim_{k\to\infty} \inf_{k\in K_\rho} u_k^0} = 0 \quad \forall i,\, k \geq k_0,$$

proving that z_k converges to zero on K_ρ.

Next, since H is a symmetric, semi-positive definite matrix, there exists a matrix L such that

$$H = L^T L,$$

so ψ_k can be written as

$$\psi_k = \|L(y_k + \frac{\Delta y_k}{2} - y^0)\|^2 - \frac{1}{4}\Delta y_k^T \Delta y_k. \tag{C.45}$$

Since $[u_k^0; x^0; z_k; s_k] \geq 0$ for $k \geq k_0$, we obtain from (C.43) that ψ_k is bounded from above. It follows from (C.45) and (C.32) that $\{Ly_k\}$ is bounded on K_ρ, and hence so is $\{Hy_k\}$. If (Pq) is strictly feasible, we can select $x^0 > 0$, and thus boundedness of $\{s_k\}$ on K_ρ follows from (C.44). In view of (C.22), together with boundedness of $\{z_k\}$ and $\{s_k\}$ on K_ρ, $\{A^T y_k\}$ is bounded on K_ρ. With boundedness of $\{Hy_k\}$ and

$\{A^T y_k\}$ on K_ρ in hand, full-rankness of $[H \ A]$ implies that $\{y_k\}$ is bounded on K_ρ, proving the last claim.

$\square$

Lemma C.4. *Suppose $\rho_k \to \infty$ as $k \to \infty$. If $\{y_k\}$ has a limit point on K_ρ and $\{z_k\}$ is bounded on K_ρ, then for any limit point $\{[y_*; z_*]\}$ of $\{[y_k; z_k]\}$ on K_ρ, there exists $\bar{x}_* \neq 0$ with $s_* = c - A^T y_* + z_* e \geq 0$ and $z_* \geq 0$ such that*

$$A\bar{x}_* = 0, \tag{C.46}$$

$$z_*(1 - e^T \bar{x}_*) = 0, \tag{C.47}$$

$$S_* \bar{x}_* = 0, \tag{C.48}$$

$$\bar{x}_* \geq 0. \tag{C.49}$$

Proof. Since $\{z_k\}$ is bounded on K_ρ, condition (C.1) will be violated eventually, and conditions (C.2) must be satisfied for all $k \in K_\rho$ large enough, i.e.,

$$\|[\Delta y_k; \Delta z_k]\| \leq \gamma_2, \quad k \in K_\rho \tag{C.50}$$

$$\tilde{x}_k^{Q_k} \geq -\gamma_3 e, \quad k \in K_\rho, \tag{C.51}$$

$$\tilde{u}_k^{Q_k} < \gamma_4 e, \quad k \in K_\rho. \tag{C.52}$$

Moreover, we have from (C.5) and (C.11)–(C.12) that for all k,

$$A\tilde{x}_k = A^{Q_k}\tilde{x}_k^{Q_k} = b - H(y_k + \Delta y_k), \tag{C.53}$$

$$e^T\tilde{x}_k^{Q_k} + \tilde{u}_k = \rho_k, \tag{C.54}$$

$$\tilde{x}_k^i = 0, \quad \forall i \notin Q_k \tag{C.55}$$

$$A^T\Delta y_k - \Delta z_k e + \Delta s_k = 0, \tag{C.56}$$

$$S_k^{Q_k}\tilde{x}_k^{Q_k} = -X_k^{Q_k}\Delta s_k^{Q_k}, \tag{C.57}$$

$$z_k\tilde{u}_k = -u_k\Delta z_k. \tag{C.58}$$

Because Q_k can take only finitely many values, it follows from Proposition A.1 (see Appendix A) with

$$G := \begin{bmatrix} A^{Q_k} & 0 \\ e^T & 1 \end{bmatrix}, \; D := \begin{bmatrix} S_k^{Q_k}(X_k^{Q_k})^{-1} & 0 \\ 0 & zu^{-1} \end{bmatrix},$$

and

$$v := \begin{bmatrix} \tilde{x}_k^{Q_k} \\ \tilde{u}_k \end{bmatrix}, \; g := \begin{bmatrix} b - H(y_k + \Delta y_k) \\ \rho_k \end{bmatrix}, \; w := \begin{bmatrix} \Delta s_k^{Q_k} \\ \Delta z_k \end{bmatrix}$$

that there exists $C > 0$ such that

$$\|[\tilde{x}_k^{Q_k}; \tilde{u}_k]\| \leq C\|[b - H(y_k + \Delta y_k); \rho_k]\| \quad \forall k.$$

Since by assumption, there exists an infinite sequence $K \subseteq K_\rho$ such that $\{y_k\}$ is bounded on K, and since $\{\Delta y_k\}$ is bounded on K_ρ in view of (C.50), we have for some C' large enough

$$\|[\tilde{x}_k^{Q_k}; \tilde{u}_k]\| \leq C'\rho_k, \quad k \in K.$$

Together with (C.55), we get that $\{[\bar{x}_k; \bar{u}_k]\}$ is bounded on K, where we have defined

$$\bar{x}_k = \frac{\tilde{x}_k}{\rho_k}, \quad \bar{u}_k = \frac{\tilde{u}_k}{\rho_k}. \tag{C.59}$$

Since both $\{y_k\}$ and $\{z_k\}$ are bounded on K, let $\{(y_*, z_*, s_*, \bar{x}_*, \bar{u}_*)\}$ be any limit point of $\{(y_k, z_k, s_k, \bar{x}_k, \bar{u}_k)\}$ on K with $z_* \geq 0$, $s_* = c - A^T y_* + z_* e \geq 0$, and

$$e^T \bar{x}_* = 1 - \bar{u}_* = 1, \tag{C.60}$$

where the equality comes from (C.54)–(C.55) and (C.59), and the inequality from (C.52). Next, since $\{x_k\}$ and $\{u_k\}$ are bounded by construction (Step 2 (ii) of IrQP-ℓ_∞), equations (C.56) and (C.50) imply from (C.55)–(C.58) that there exists $C'' > 0$ such that

$$\begin{aligned} s_k^i \tilde{x}_k^i &= 0, \quad i \notin Q & \text{(C.61)} \\ \|S_k^{Q_k} \tilde{x}_k^{Q_k}\| &= \|X_k^{Q_k} \Delta s_k^{Q_k}\| \leq C'' \quad k \in K, & \text{(C.62)} \\ \|z_k \tilde{u}_k\| &= \|u_k \Delta z_k\| \leq C'' \quad k \in K. & \text{(C.63)} \end{aligned}$$

Dividing both sides of (C.53)–(C.55), (C.61)–(C.63) and (C.51) by ρ_k and taking limits on K, we conclude that $\bar{x}_* \neq 0$ (as for (C.60)) satisfies

$$\begin{aligned} A\bar{x}_* &= 0, \\ e^T \bar{x}_* + \bar{u}_* &= 1, \\ z_* \bar{u}_* &= 0, \\ S_* \bar{x}_* &= 0, \\ \bar{x}_* &\geq 0, \end{aligned}$$

proving the claim.

□

The following theorem establishes that ρ_k is increased at most finitely many times.

Theorem C.1. *Suppose* (Pq)–(Dq) *is strictly feasible,*[1] *then ρ_k is increased at most finitely many times, i.e., K_ρ is finite.*

Proof. By contradiction, suppose K_ρ is infinite, i.e., $\rho_k \to \infty$ as $k \to \infty$. In view of Lemma C.3, $\{y_k\}$ is bounded, and $\{z_k\} \to 0$ as $k \to \infty, k \in K_\rho$. Let y_* and z_* be the limit points of $\{y_k\}$ and $\{z_k\}$ on K_ρ, so $z_* = 0$, i.e., $y_* \in \mathcal{F}$. It follows from Lemma C.4 that there exists $\bar{x}_* \neq 0$ that satisfies (C.46)–(C.49). In view of (C.48),

$$\bar{x}_* = 0, \quad \forall i \notin I(y_*). \tag{C.64}$$

Together with (C.46), we get

$$\sum_{i \in I(y_*)} \bar{x}_*^i a^i = 0.$$

Since the strict feasibility of (Dq) implies positive linear independence of vectors $\{a^i : i \in I(y_*), y_* \in \mathcal{F}\}$, it follows from (C.49) that

$$\bar{x}_* = 0, \quad \forall i \in I(y_*).$$

Together with (C.64), we have

$$\bar{x}_* = 0,$$

contradicting to that $\bar{x}_*$ is nonzero. □

Finally, if positive linear independence of $\{a^i : i \in I(y_*)\}$ at feasible limit points y_* of $\{y_k\}$ is replaced with the much stronger assumption of linear independence of $\{a_i : i \in I(y)\}$ at *all* $y \in \mathbb{R}^m$, then boundedness of ρ_k follows without any feasibility assumption, as we state next.

[1] That (Pq) is strictly feasible is equivalent to that the solution set of problem (Dq) is nonempty and bounded (see Theorem 2.1 in [20]). So our assumptions are the same as that in [32].

Proposition C.1. *Suppose that, at every point $y \in \mathbb{R}^m$, $\{a^i : i \in I(y)\}$ is a linearly independent set. If $\{y_k\}$ has a limit point on K_ρ and $\{z_k\}$ is bounded on K_ρ, then ρ_k is increased at most finitely many times.*

Proof. By contradiction, suppose $\rho_k \to \infty$ as $k \to \infty$. Lemma 3.4 then implies that for any limit point $z_* \geq 0$, $s_* \geq 0$ and y_* of sequences $\{z_k\}$, $\{s_k\}$ and $\{y_k\}$ with $s_* = c - A^T y_* + z_* e$, there exists $\bar{x}_* \neq 0$ that satisfies (C.46)–(3.55). We can then conclude the proof with a contradiction argument, exactly as is in the proof of Theorem C.1, except that the requirement of positive linear independence of $\{a^i : i \in I(y_*),\, y_* \in \mathcal{F}\}$ is replaced by the assumption of linear independence of $\{a^i : i \in I(y_*),\, y \in \mathbb{R}^m\}$. □

Bibliography

[1] M. Achache. A new primal-dual path-following method for convex quadratic programming. *Computational and Applied Mathematics*, 25(1):97–110, 2006.

[2] A. Alessio and A. Bemporad. A survey on explicit model predictive control. *In Proc. int. workshop on assessment and future directions of nonlinear model predictive control.*

[3] P. Armand. A quasi-Newton penalty barrier method for convex minimization problems. *Computational Optimization and Applications*, 26(1):5–34, 2003.

[4] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming-the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[5] A. Bemporad, F. Borrelli, and M. Morari. Using interpolation to improve efficiency of multiparametric predictive control. *Automatica*, 41:637–643, 2005.

[6] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit solution of model predictive control via multiparametric quadratic programming. *American Control Conference*, pages 872–876, 2000.

[7] A. Ben-Tal and A. Charnes. A dual optimization framework for some problems of information theory and statistics. *Problems of Control and Information Theory*, (8):387–401, 1979.

[8] H. Y. Benson, A. Sen, and D. F. Shanno. interior-point methods for nonconvex nonlinear programming: Convergence analysis and computational performance. *http://rutcor.rutgers.edu/˜shanno/converge5.pdf*, 2009.

[9] H. Y. Benson, A. Sen, D. F. Shanno, and R. J. Vanderbei. Interior-point algorithms, penalty methods and equilibrium problems. *Computational Optimization and Applications*, 34(2):155–182, 2006.

[10] H. Y. Benson and D. F. Shanno. An exact primal–dual penalty method approach to warmstarting interior-point methods for linear programming. *Computational Optimization and Applications*, 38(3):371–399, 2007.

[11] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization.* Athena Scientific, 2003.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[13] E. F. Camacho and C. Bordons. *Model Predictive Control.* 2nd Edition, Springer, 2004.

[14] C. Cartis and N. I. M. Gould. Finding a point in the relative interior of a polyhedron. *Technical report*, 2006.

[15] G. Dantzig and Y. Ye. A build-up interior-point method for linear programming: Affine scaling form. *Working paper, Department of Management Science, University of Iowa*, 1991.

[16] D. den Hertog, J. Kaliski, C. Roos, and T. Terlaky. A logarithmic barrier cutting plane method for convex programming. *the Faculty of Technical Mathematics and Informatics, Delft, The Netherlands*, 1993.

[17] D. den Hertog, C. Roos, and T. Terlaky. A polynomial method of weighted centers for convex quadratic programming. *Journal of Information and Optimization Science*, 12:187–205, 1991.

[18] D. den Hertog, C. Roos, and T. Terlaky. Adding and deleting constraints in logarithmic barrier method for linear programming problems. *Shell report, AMER 92-001*, 1992.

[19] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Mathematics*, 8:674–675, 1967.

[20] L. M. Grana Drummond and B. F. Svaiter. On well definedness of the central path. *Journal of Optimization Theory and Application*, 102(2):223–237, 1999.

[21] K. Edlund, L. E. Sokoler, and J. B. Jorgensen. A primal-dual interior-point linear programming algorithm for MPC. In *Conference on Decision and Control*, pages 351–356. IEEE, 2009.

[22] S. Fang, J. R. Rajasekera, and H. J. Tsao. *Entropy optimization and mathematical programming.* Springer, 1997.

[23] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Society for Industrial and Applied Mathematics, 1990.

[24] J. W. Fletcher. Identification of UH-60 stability derivative models in hover from flight test data. *Journal of the American Helicopter Society*, 40(1):32–46, 1995.

[25] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, (3):95–110, 1956.

[26] G. C. Goodwin, M. M. Seron, and J. A. De Dona. *Constrained Control and Estimation: an Optimisation Approach.* Springer, London, 2005.

[27] M. Y. He and A. L. Tits. An infeasible constraint-reduced interior-point method for linear programming. *Optimization Methods and Software*, 2010.

[28] S. Hovl, J. T. Gravdahl, and K. E. Willcox. MPC for large-scale systems via model reduction and multiparametric quadratic programming. *Proc. 45th IEEE conference on decision and control*, 2006.

[29] K. Ikeda and N. Murata. Learning properties of support vector machines with p-norm. *Circuits and Systems*, 3:69–72, 2004.

[30] B. Jansen, B. Jansen, C. Roos, C. Roos, T. Terlaky, and T. Terlaky. A polynomial primal-dual dikin-type algorithm for linear programming. Technical report, Faculty of Technical Mathematics and Computer Science, Delft University of Technology, 1993.

[31] J. H. Jung, D. P. O'Leary, and A. L. Tits. Adaptive constraint reduction for training support vector machines. *Electronic Transactions on Numerical Analysis*, 31:156–177, 2008.

[32] J. H. Jung, D. P. O'Leary, and A. L. Tits. Adaptive constraint reduction for convex quadratic programming. *Computational Optimization and Applications*, pages DOI: 10.1007/s10589–010–9324–8, published on-line as of March 2010.

[33] T. Keviczky and G. Balas. Software-enabled receding horizon control for autonomous unmanned aerial vehicle guidance. *Journal of Guidance*, 29(3), 2006.

[34] R. Kulhavy, J. Lu, and T. Samad. Emerging technologies for enterprise optimization in the process industries. *Aiche Symposium Series*, 98(326):352–363, 2001.

[35] S. Leyffer, G. L. Calva, and J. Nocedal. Interior methods for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 17(1):52–77, 2006.

[36] J. M. Maciejowski. *Predictive Control with Constraints.* Prentice Hall, 2002.

[37] O. L. Mangasarian. *Nonlinear Programming.* McGraw-Hill, Inc, 1969.

[38] V. Manikonda, P. O. Arambel, M. Gopinathan, R. K. Mehra, and F. Y. Hadaegh. A model predictive control-based approach for spacecraft formation keeping and attitude control. In *Proceedings of the 1999 American Control Conference*, pages 4258–4262, 1999.

[39] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[40] R. Monteiro and I. Adler. Interior path following primal-dual algorithms. Part I: Linear programming. *Mathematical Programming*, 44:27–41, 1989.

[41] R. D. C. Monteiro, I. Adler, and M. G. C. Resende. A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, 15(2):191–214, 1990.

[42] M. Morari and N. L. Ricker. *Model Predictive Control Toolbox*. Natick, MA: The MathWorks Inc., 1994.

[43] P. M. Murphy and D. W. Aha. UCI machine learning repository. *http://archive.ics.uci.edu/ml/index.html*, 1992.

[44] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 36, 1993.

[45] S. O. Nicholls. Column generation in infeasible predictor-corrector methods for solving linear programs. *Thesis, University of Maryland*, 2009.

[46] F. A. Potra. A quadratically convergent predictor-corrector method for solving linear programs from infeasible starting points. *Mathematical Programming: Series A and B*, 67(3):383–406, 1994.

[47] F. A. Potra. An infeasible-interior-point predictor-corrector algorithm for linear programming. *SIAM Journal on Optimization*, 6(1):19–32, 1996.

[48] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[49] C. V. Rao and J. B. Rawlings. linear programming and model predictive control. *Journal of Process Control*, 10, 2000.

[50] C. Roos, T. Terlaky, and J. P. Vial. *Interior point methods for linear optimization*. Springer, 2005.

[51] R. Saigal. On the primal-dual affine scaling method. *Technical report, Department of Industrial and Operational Engineering, The University of Michigan*, 1994.

[52] R. Saigal. A simple proof of a primal affine scaling method. *Annals of Operations Research*, 62:303–324, 1996.

[53] M. Salahi and T. Terlaky. On Mehrotra-type predictor-corrector algorithms. *SIAM J. Optim*, 18(4):1377–1397, 2007.

[54] A. Sen and D. F. Shanno. Convergence analysis of an interior-point method for mathematical programs with equilibrium constraints. *http://rutcor.rutgers.edu/˜shanno/IPMPEC2.pdf*, 2006.

[55] G. W. Stewart. On scaled projections and pseudo-inverses. *Linear Algebra and its Applications*, 112:189–194, 1989.

[56] The Netlib LP test problem set. http://www.numerical.rl.ac.uk/cute/netlib.html.

[57] COAP test problems. http://www.math.ufl.edu/˜hager/coap/format.html.

[58] C. Theodore and R. Celi. Helicopter flight dynamic simulation with refined aerodynamic and flexible blade modeling. *Journal of Aircraft*, 39(4):577–586, 2002.

[59] Y. Tian, J. Yu, and W. Chen. L_p-norm support vector machine with CCCP. *Fuzzy Systems and Knowledge Discovery*, pages 1560–1564, 2010.

[60] A. L. Tits, P. A. Absil, and W. P. Woessner. Constraint reduction for linear programs with many inequality constraints. *SIAM Journal on Optimization*, 17(1):119–146, 2006.

[61] A. L. Tits, A. Wächter, S. Bakhtiari, T. J. Urban, and C. T. Lawrence. A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. *SIAM Journal on Optimization*, 14(1):173–199, 2003.

[62] K. Tone. An active-set strategy in an interior point method for linear programming. *Mathematical Programming*, 59:345–360, 1993.

[63] E. A. Wan, A. A. Bogdanov, and E. A. Bogdanov. Model predictive neural control with applications to a 6 DoF helicopter model. In *Proceedings of the 2001 American Control Conference*, pages 488–493, 2001.

[64] L. B. Winternitz. Primal-dual interior-point algorithms for linear programs with many inequality constraints. *Thesis, University of Maryland*, 2010.

[65] L. B. Winternitz, S. O. Nicholls, A. L. Tits, and D. P. O'Leary. A constraint-reduced variant of Mehrotra's predictor-corrector algorithm. *Computational Optimization and Applications*, DOI: 10.1007/s10589-010-9389-4, published online as of January 2011.

[66] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, 1997.

[67] Y. Ye. Eliminating columns in the simplex method for linear programming. *Journal of Optimization Theory and Applications*, 63(1):69–77, 1989.

[68] Y. Ye. A "build-down" scheme for linear programming. *Mathematical Programming*, 46(1-3):61–72, 1990.

[69] Y. Ye. *Interior Point Algorithms: Theory and Analysis.* Wiley-Interscience, 1997.

[70] J. Zhang and P. L. Brockett. Quadratically constrained information theoretic analysis. *SIAM Journal on Applied Mathematics*, 47(4):871–885, 1987.

[71] Y. Zhang. Solving large-scale linear programs by interior-point methods under the MATLAB environment. Technical report, 1998.