

Homogeneous Infeasible Interior Point Method for Convex Quadratic Programs

Raghunathan, Arvind; Jha, Devesh K.; Romeres, Diego

TR2022-157 December 09, 2022

Abstract

Optimization based control is widely used for stabilizing control of constrained linear dynamical systems. We present an Infeasible Interior Point Method (IIPM) for the solution of convex quadratic programs, such as those arising in Model Predictive Control (MPC) of constrained linear dynamical systems, using a novel homogeneous formulation [1]. The homogenization is applied on a slacked reformulation of the QP. We describe a tailored step computation in the IIPM that addresses the potential loss of sparsity resulting from the homogenization. We present arguments for the effectiveness of the slacked formulation in warm-start of IIPM. The algorithm is implemented in Julia. Numerical experiments on the formulation are provided comparing the proposed approach against existing IPM implementations on feasible and infeasible quadratic programs. We also demonstrate that the warm-starts of the proposed IIPM reduces the computational time by 50% on an MPC application.

IEEE Conference on Decision and Control (CDC) 2022

Homogeneous Infeasible Interior Point Method for Convex Quadratic Programs

Arvind U Raghunathan, Devesh Jha and Diego Romeres

Abstract—Optimization based control is widely used for stabilizing control of constrained linear dynamical systems. We present an Infeasible Interior Point Method (IIPM) for the solution of convex quadratic programs, such as those arising in Model Predictive Control (MPC) of constrained linear dynamical systems, using a novel homogeneous formulation [1]. The homogenization is applied on a slacked reformulation of the QP. We describe a tailored step computation in the IIPM that addresses the potential loss of sparsity resulting from the homogenization. We present arguments for the effectiveness of the slacked formulation in warm-start of IIPM. The algorithm is implemented in Julia. Numerical experiments on the formulation are provided comparing the proposed approach against existing IPM implementations on feasible and infeasible quadratic programs. We also demonstrate that the warm-starts of the proposed IIPM reduces the computational time by 50% on an MPC application.

I. INTRODUCTION

Convex Quadratic Programs (QPs) arise in a number of applications including financial portfolio optimization [2], control [3], [4], and as relaxations of mixed integer quadratic programs [5], [6]. Sequential Quadratic Programming (SQP) algorithms [7] also solve quadratic programs with appropriate inertia at each step of the algorithm [8]. There has been active interest in the development of QP algorithms for the model predictive control of linear dynamical systems including Interior Point Methods (IPMs) [9], active set methods [10], [11], IPMs for Second-Order Cone Programs (SOCPs) [12], gradient projection methods [13], dual gradient projection methods [14], [15], splitting methods [16], [17], [18], iterative approaches [19] and semismooth Newton methods [20]. More recently, there has been interest in the development of Mixed Integer QP (MIQP) solvers for the control of hybrid systems [21], [22], [23], [24].

In real-time control applications, it is critical for QP solvers to robustly certify optimality or infeasibility. Infeasibility detection takes increased relevance in the context of Branch & Bound (BB) methods for solution of MIQPs. Certification of infeasibility of the relaxation at a particular node in the BB tree allows to fathom the subtree and is critical to the overall efficiency of the solution process.

Infeasibility detection has also been addressed in recent years for first-order methods in [25], [26], [27]. Standard IPMs for QP (such as in [28]) are incapable of certifying infeasibility. IPMs based on Homogeneous Self-Dual (HSD) embedding [29], [30] are known to produce a certificate

of infeasibility for Linear Programs (LPs). This technique has been implemented in MOSEK [27] and has also been extended to handle SOCPs. To detect infeasible QPs with IPMs, the QPs must first be formulated as a SOCP to which the HSD embedding is then applied as is done in the ECOS [12] and SCS [31] solvers.

Raghunathan [1] presented a novel homogeneous formulation of QPs that allows robust detection of infeasibility in QPs. The homogenization formulates the problem as a QP as opposed to the SOCP that results when using the HSD embedding. In this paper, we have a two-fold objective. First, we want to investigate the computational efficiency of an Infeasible Interior Point Method (IIPM) when applied to the homogeneous formulation of QP introduced in [1]. Secondly, we want to explore the warm-start capabilities of our formulation. Motivated by the two objectives, we first present an equivalent slacked reformulation of the QP that increases the problem variables and constraints. The homogenization of [1] is applied to this slacked reformulation to obtain the Homogeneous Slacked QP (HSQP). We present a standard predictor-corrector IIPM [32] for the HSQP with a specialized step computation. The step computation ensures that the introduction of slacks does not significantly increase the computations in each iteration. We also show that the slacked formulation can be effectively used for *warm-start* of QPs in the presence of data perturbations. *Warm-start* refers to the idea of using the optimal solution for a particular instance as an initial guess for a slightly perturbed problem. Such a sequence of perturbed QPs arises in the context of MPC and in the solution of MIQPs. IPMs are not conducive for warm-start due to the need for initial iterates to lie in the interior of the bounds. The *slacked* formulation of our homogeneous formulation addresses this deficiency and is inspired by the slacked formulation of [33] for warm-starting Linear Programs (LPs). We implemented the IIPM in Julia [34] and use MKL Pardiso [35] for the linear algebra computations. Numerical experiments show that our implementation is faster than existing state-of-the-art IPMs that do not use the homogeneous formulation in [1]. Finally, we demonstrate that the warm-start capability of our approach can reduce the computation times by 50% when applied to MPC.

The paper is organized as follows. §II presents the QP formulation along with the assumptions. The slacked reformulation of the QP and the homogeneous formulation are presented in §III. §IV presents the predictor-corrector IIPM for the slacked formulation, and tailored linear algebra employed for the step computation. We present the effect

of data perturbations on the iteration complexity in §V and propose warm-start strategy for the HSQP. §VI presents the numerical results on more than 800 randomly generated problems of varying size and density. we also present the effect of warm-starts on an MPC application. We compare our approach to existing IPM implementations such as ECOS [12], Ipopt [8], and MOSEK [36].

Notation. The set of reals is denoted by \mathbb{R} and the set of vectors of dimension n by \mathbb{R}^n . The set of $n \times n$ symmetric matrices is denoted by \mathbb{S}^n . For a matrix $A \in \mathbb{S}^n$, the notation $A \succ (\succsim) 0$ denotes that A is positive semidefinite (definite). Given a vector $u \in \mathbb{R}^n$, U denotes a diagonal matrix with the elements of the vector on the diagonal. We denote by $\mathbf{1}_n$ the n -vector all ones, and I_n the identity matrix of size n .

II. PROBLEM FORMULATION

We consider QPs of the form

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + q^T x \quad (1a)$$

$$\text{s.t. } Ax = b \quad (1b)$$

$$l \leq x \leq u \quad (1c)$$

where $Q \in \mathbb{S}^n$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$. The vectors $l, u \in \mathbb{R}^n$ represent the lower and upper bounds on the variables. For sake of clarity in presentation, we assume that l, u are all finite. This is not a requirement for our implementation as described in §VI.

We make the following assumptions on QP (1).

Assumption 1: The equality constraint matrix has full row rank.

Assumption 2: The Hessian is strictly convex on null-space of the equality constraints.

Assumption 1 states that the matrix A has full row rank of m . This is not restrictive and can be easily satisfied by removing dependent rows if necessary. Assumption 2 requires that the matrix $Z^T A Z \succ 0$ where $Z \in \mathbb{R}^{n \times (n-m)}$ is an orthonormal basis for the null-space of A , i.e. Z is a basis for $\{v \mid Av = 0\}$. This assumption readily holds for MPC formulations [25] and also for certain spectral relaxations of nonconvex MIQPs [5], [6]. Assumption 2 implies that the optimal solution to QP (1) is unique whenever QP (1) is feasible. Note that the assumptions do not preclude the infeasibility of QP (1). We end the section by stating the conditions for optimality and infeasibility of QP (1).

A point x^* minimizes QP (1) if $x^* \in [l, u]$ and there exist multipliers $\lambda_x^* \in \mathbb{R}^m$, $z_l^*, z_u^* \geq 0 \in \mathbb{R}^n$, satisfying [32]

$$Qx^* + A^T \lambda_x^* - z_l^* + z_u^* + q = 0 \quad (2a)$$

$$Ax^* - b = 0 \quad (2b)$$

$$(X^* - L)z_l^* = 0 \quad (2c)$$

$$(U - X^*)z_u^* = 0. \quad (2d)$$

The first-order optimality conditions (2) are necessary and sufficient for a minimizer of the QP (1) under Assumption 2.

The QP (1) is infeasible if there exist $\lambda_x^\circ \in \mathbb{R}^m$, and $z_l^\circ, z_u^\circ \in \mathbb{R}^n$ satisfying

$$A^T \lambda_x^\circ - z_l^\circ + z_u^\circ = 0 \quad (3a)$$

$$b^T \lambda_x^\circ - l^T z_l^\circ + u^T z_u^\circ = -1 \quad (3b)$$

$$z_l^\circ, z_u^\circ \geq 0. \quad (3c)$$

The conditions in (3) are obtained from applying Motzkin's Theorem of the Alternative [37, § 2.4] to (1b)-(1c).

III. HOMOGENEOUS SLACKED QP (HSQP)

We begin by presenting the equivalent *slacked* reformulation of the QP (1).

$$\min_{x, s_l, s_u \in \mathbb{R}^n} \frac{1}{2} x^T Q x + q^T x \quad (4a)$$

$$\text{s.t. } Ax = b \quad (4b)$$

$$x - s_l = l \quad (4c)$$

$$x + s_u = u \quad (4d)$$

$$s_l, s_u \geq 0. \quad (4e)$$

The variables s_l, s_u are the so-called *slack* variables and help to translate the bounds in (1c) into nonnegativity of s_l, s_u in (4e). The reformulation will be critical to the warm-start of our IIPM formulation as discussed in §V. It is easily shown that satisfaction of Assumptions 1 and 2 for QP (1) implies the full rank of equalities (4b)-(4d) and strict convexity in the null space of equalities in the slacked formulation in (4).

We present a new formulation obtained by homogenizing the equalities in (4) called the Homogeneous Slacked QP (HSQP) formulation. The HSQP is given by

$$\min_{x, s_l, s_u \in \mathbb{R}^n, \tau} \frac{1}{2} x^T Q x + \tau q^T x + \frac{\theta}{2} (\tau^2 - 2\tau) \quad (5a)$$

$$\text{s.t. } Ax - b\tau = 0 \quad (5b)$$

$$x - s_l - l\tau = 0 \quad (5c)$$

$$x + s_u - u\tau = 0 \quad (5d)$$

$$s_l, s_u, \tau \geq 0 \quad (5e)$$

where $\tau \in \mathbb{R}$ is an additional nonnegative variable and $\theta > 0$ is a parameter. Observe that the objective (5a) is obtained by multiplying the linear term $q^T x$ in (4a) with τ and appending with the term $(\theta/2)(\tau^2 - 2\tau)$. The equality constraints (5b)-(5d) are obtained by multiplying the right-hand side of (4b)-(4d) with τ . We present conditions on θ such that Assumption 2 also holds for HSQP in §III-A. Since HSQP is always feasible, the satisfaction of Assumption 2 ensures that HSQP has a unique solution. §III-B shows that solving HSQP allows to recover a solution to QP or declare infeasibility.

A. Conditions on θ

The parameter θ is chosen to satisfy two conditions:

- 1) $\theta > 2|\theta^*|$ where θ^* is valid lower bound on the optimal objective of slacked QP (4)

$$-\frac{1}{2}\theta < \theta^* \leq \frac{1}{2} x^T Q x + q^T x \quad (6)$$

$$\forall (x, s_l, s_u) \text{ satisfying (4b) - (4e).}$$

- 2) θ is chosen large enough so that

$$\widehat{Z}^T \begin{bmatrix} Q & q \\ q^T & \theta \end{bmatrix} \widehat{Z} \succ 0 \quad (7)$$

where \widehat{Z} is a basis for the null space of (5b).

The satisfaction of (7) implies that:

- 1) the Hessian of the objective in (5a) is positive definite on the null space of the equality constraints (5b)-(5d).
- 2) HSQP (5) is convex and the first-order optimality conditions are necessary and sufficient for a minimizer.

We refer the interested reader to [1, §III.B] for a discussion on the computation of θ .

B. Equivalence between QP and HSQP

We collect some simple observations on the HSQP (5).

- (O1) HSQP (5) is always feasible. It is easily verified that $(x, s_l, s_u, \tau) = 0$ satisfies (5b)-(5e).
- (O2) HSQP (5) has an optimal solution with optimal value less than or equal to 0. This follows directly from (O1).
- (O3) HSQP (5) has a finite optimum. This follows from (7).

We now state the first-order optimality conditions for HSQP (5).

A point $(\widehat{x}, \widehat{s}_l, \widehat{s}_u, \widehat{\tau})$ minimizes HSQP (5) if there exist multipliers $(\widehat{\lambda}_x, \widehat{\lambda}_l, \widehat{\lambda}_u) \in \mathbb{R}^{m+2n}$, $(\widehat{z}_l, \widehat{z}_u, \widehat{z}_\tau) \geq 0 \in \mathbb{R}^{2n+1}$ satisfying the first-order optimality conditions

$$Q\widehat{x} + A^T\widehat{\lambda}_x + \widehat{\lambda}_l + \widehat{\lambda}_u + q\widehat{\tau} = 0 \quad (8a)$$

$$-\widehat{\lambda}_l - \widehat{z}_l = 0 \quad (8b)$$

$$\widehat{\lambda}_u - \widehat{z}_u = 0 \quad (8c)$$

$$A\widehat{x} - b\widehat{\tau} = 0 \quad (8d)$$

$$\widehat{x} - \widehat{s}_l - l\widehat{\tau} = 0 \quad (8e)$$

$$\widehat{x} + \widehat{s}_u - u\widehat{\tau} = 0 \quad (8f)$$

$$\widehat{S}_l\widehat{z}_l = 0 \quad (8g)$$

$$\widehat{S}_u\widehat{z}_u = 0 \quad (8h)$$

$$\theta\widehat{\tau} - \theta + q^T\widehat{x} - b^T\widehat{\lambda}_x - l^T\widehat{\lambda}_l - u^T\widehat{\lambda}_u - \widehat{z}_\tau = 0 \quad (8i)$$

$$\widehat{\tau}\widehat{z}_\tau = 0. \quad (8j)$$

Theorem 1: Suppose θ is chosen to satisfy the conditions in §III-A. The QP (1) has an optimal solution x^* iff the HSQP (5) has an optimal solution $(\widehat{x}, \widehat{s}_l, \widehat{s}_u, \widehat{\tau})$ with $\widehat{\tau} > 0$.

Proof: Consider the *if* part. Let $(\widehat{x}, \widehat{s}_l, \widehat{s}_u, \widehat{\tau})$ be an optimal solution of HSQP (5) and let $(\widehat{\lambda}_x, \widehat{\lambda}_l, \widehat{\lambda}_u)$, $(\widehat{z}_l, \widehat{z}_u, \widehat{z}_\tau)$ be multipliers such that (8) holds. Then, it is easily verified that $x^* = \widehat{x}/\widehat{\tau}$, $(\lambda_x^*, z_l^*, z_u^*) = (\widehat{\lambda}_x/\widehat{\tau}, \widehat{z}_l/\widehat{\tau}, \widehat{z}_u/\widehat{\tau})$ satisfies the optimality conditions (2) for QP (1), proving the *if* part.

Consider the *only if* part. Let x^* be an optimal solution of QP (1) and let $(\lambda_x^*, z_l^*, z_u^*)$ be multipliers such that (2) holds. Define $\bar{\tau} = \theta/(\theta + q^T y^* - b^T \lambda_x^* + l^T z_l^* - u^T z_u^*)$. If $\bar{\tau} > 0$ then it can be easily verified that $(\widehat{x}, \widehat{s}_l, \widehat{s}_u) = \bar{\tau}(x^*, x^* - l, u - x^*)$, $\widehat{\tau} = \bar{\tau}$, $(\widehat{\lambda}_x, \widehat{\lambda}_l, \widehat{\lambda}_u) = \bar{\tau}(\lambda_x^*, -z_l^*, z_u^*)$ and $(\widehat{z}_l, \widehat{z}_u) = \bar{\tau}(z_l^*, z_u^*)$ satisfy the optimality conditions for HSQP (8). To show $\bar{\tau} > 0$ we need to show that $\theta + q^T x^* - b^T \lambda_x^* + l^T z_l^* - u^T z_u^* > 0$ since $\theta > 0$. Consider

$$\theta + q^T x^* - b^T \lambda_x^* + l^T z_l^* - u^T z_u^* \quad (9a)$$

$$= \theta + q^T x^* - (A^T \lambda_x^*)^T x^* + l^T z_l^* - u^T z_u^* \quad (9b)$$

$$= \theta + 2q^T x^* + (x^*)^T Q x^* + l^T z_l^* - u^T z_u^* + (x^*)^T (-z_l^* + z_u^*) \quad (9c)$$

$$= \theta + 2q^T x^* + (x^*)^T Q x^* > 0 \quad (9d)$$

where the equality in (9b) follows by multiplying (2b) by $(\nu^*)^T$ and substituting for $b^T \lambda_x^*$ with $(A^T \lambda_x^*)^T x^*$. Multiplying (2a) by $(x^*)^T$ and substituting for $-(A^T \nu^*)^T x^*$ as $q^T x^* + (x^*)^T Q x^* - (x^*)^T (z_l^* - z_u^*)$ yields (9c). Using the complementarity constraints (2c)-(2d) in (9c) yields the equality (9d). The final inequality follows from (6), completing the proof. ■

We now show that infeasibility of QP (1) is equivalent to the vanishing of the optimal solution to HSQP (5).

Theorem 2: Suppose θ satisfies the conditions in §III-A. The QP (1) is infeasible iff the HSQP (5) has optimal solution $(\widehat{x}, \widehat{s}_l, \widehat{s}_u, \widehat{\tau}) = 0$.

Proof: Consider the *only if* part of the claim. Suppose there exist $(\lambda_x^o, z_l^o, z_u^o)$ satisfying (3). It can be verified that $(\widehat{x}, \widehat{s}_l, \widehat{s}_u, \widehat{\tau}) = 0$, $(\widehat{\lambda}_x, \widehat{\lambda}_l, \widehat{\lambda}_u) = (\theta \lambda_x^o, -\theta z_l^o, \theta z_u^o)$, $(\widehat{z}_l, \widehat{z}_u, \widehat{z}_\tau) = (\theta z_l^o, \theta z_u^o, 0)$ satisfies (8). Hence $(\widehat{x}, \widehat{s}_l, \widehat{s}_u, \widehat{\tau}) = 0$ is an optimal solution to HSQP (5).

Consider the *if* part of the claim. Suppose $(\widehat{x}, \widehat{s}_l, \widehat{s}_u, \widehat{\tau}) = 0$ is the optimal solution to (5) and let $(\widehat{\lambda}_x, \widehat{\lambda}_l, \widehat{\lambda}_u)$, $(\widehat{z}_l, \widehat{z}_u, \widehat{z}_\tau)$ be the multipliers in (8). Then $(\lambda_x^o, z_l^o, z_u^o) = (\widehat{\lambda}_x, \widehat{z}_l, \widehat{z}_u)/(\theta + \widehat{z}_\tau)$ can be verified to satisfy (3). This proves the *if* part of the claim, completing the proof. ■

IV. INFEASIBLE INTERIOR POINT METHOD (IIPM)

We describe a predictor-corrector IIPM for solving HSQP in the following. We first present the step computation in §IV-A and then outline the predictor-corrector IIPM in §IV-B. The step computation shows that computational work scales with the size of QP (1) and preserves the sparsity in (1).

A. IIPM Step Computation

Each iteration of the IIPM involves the solution of a linear system which represents the Newton step of the optimality conditions in (8). The linear system is of the form

$$Q\Delta x + A^T \Delta \lambda_x + \Delta \lambda_l + \Delta \lambda_u + q\Delta \tau = r_{dx} \quad (10a)$$

$$-\Delta \lambda_l - \Delta z_l = r_{dl} \quad (10b)$$

$$\Delta \lambda_u - \Delta z_u = r_{du} \quad (10c)$$

$$A\Delta x - b\Delta \tau = r_{px} \quad (10d)$$

$$\Delta x - \Delta s_l - l\Delta \tau = r_{pl} \quad (10e)$$

$$\Delta x + \Delta s_u - u\Delta \tau = r_{pu} \quad (10f)$$

$$Z_l \Delta s_l + S_l \Delta z_l = r_{cl} \quad (10g)$$

$$Z_u \Delta s_u + S_u \Delta z_u = r_{cu} \quad (10h)$$

$$\theta \Delta \tau - b^T \Delta \lambda_x - l^T \Delta \lambda_l - u^T \Delta \lambda_u - \Delta z_\tau = r_{d\tau} \quad (10i)$$

$$z_\tau \Delta \tau + \tau \Delta z_\tau = r_{c\tau} \quad (10j)$$

where the right hand quantities will be made precise in Algorithm 1. The linear system in (10) has $(7n+m+2)$ unknowns and constraints. Additionally, the equality in (10i) can be quite dense since the vector q, b, l, u can all be component-wise nonzero. As a result, the linear system in (10) can be dense even when the data matrices Q, A are sparse. A direct solution of such a system can result in computational inefficiency in the solution of the linear system. We show

in the following that the unknowns in (10) can be computed at expense of factorizing a linear system that is only of size $(n+m)$. This is precisely the dimensions of the QP (1). Thus, the introduction of the slacked formulation does not affect the computational efficiency of the IIPM step computation.

To begin with, we note from (10) that $(\Delta z_l, \Delta z_u, \Delta z_\tau)$ can be expressed using (10g),(10h),(10j) as

$$\Delta z_l = S_l^{-1}(r_{cl} - Z_l \Delta s_l) \quad (11a)$$

$$\Delta z_u = S_u^{-1}(r_{cu} - Z_u \Delta s_u) \quad (11b)$$

$$\Delta z_\tau = \tau^{-1}(r_{c\tau} - z_\tau \Delta \tau) \quad (11c)$$

We can eliminate $(\Delta z_l, \Delta z_u, \Delta z_\tau)$ from (10) by substituting (11) in (10b),(10c),(10i) to obtain the system

$$Q\Delta x + A^T \Delta \lambda_x + \Delta \lambda_l + \Delta \lambda_u + q\Delta \tau = r_{dx} \quad (12a)$$

$$\Sigma_l \Delta s_l - \Delta \lambda_l = r_{dl} + S_l^{-1} r_{cl} \quad (12b)$$

$$\Sigma_u \Delta s_u + \Delta \lambda_u = r_{du} + S_u^{-1} r_{cu} \quad (12c)$$

$$A\Delta x - b\Delta \tau = r_{px} \quad (12d)$$

$$\Delta x - \Delta s_l - l\Delta \tau = r_{pl} \quad (12e)$$

$$\Delta x + \Delta s_u - u\Delta \tau = r_{pu} \quad (12f)$$

$$q^T \Delta x - b^T \Delta \lambda_x - l^T \Delta \lambda_l - u^T \Delta \lambda_u + \tau^{-1} z_\tau \Delta \tau = r_{d\tau} + \tau^{-1} r_{c\tau} \quad (12g)$$

where $\Sigma_l = Z_l S_l^{-1}$, $\Sigma_u = Z_u S_u^{-1}$. Using (12) we express $(\Delta s_l, \Delta \lambda_l, \Delta s_u, \Delta \lambda_u)$ in terms of $(\Delta x, \Delta \lambda_x, \Delta \tau)$ as

$$\Delta s_l = -r_{pl} + \Delta x - l\Delta \tau \quad (13a)$$

$$\Delta \lambda_l = \Sigma_l(\Delta x - l\Delta \tau - r_{pl}) - r_{dl} - S_l^{-1} r_{cl} \quad (13b)$$

$$\Delta s_u = r_{pu} - \Delta x + u\Delta \tau \quad (13c)$$

$$\Delta \lambda_u = \Sigma_u(\Delta x - u\Delta \tau - r_{pu}) + r_{du} + S_u^{-1} r_{cu}. \quad (13d)$$

Substituting $(\Delta s_l, \Delta s_u)$ from (13) into (12) obtain

$$\begin{pmatrix} Q + \Sigma_l + \Sigma_u & A^T & \tilde{q} \\ A & 0 & -b \\ \tilde{q}^T & -b^T & \Sigma_\tau \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda_x \\ \Delta \tau \end{pmatrix} = \begin{pmatrix} \tilde{r}_{dx} \\ r_{px} \\ \tilde{r}_{d\tau} \end{pmatrix} \quad (14)$$

where

$$\Sigma_\tau = \theta + \tau^{-1} z_\tau + l^T \Sigma_l l + u^T \Sigma_u u$$

$$\tilde{q} = q - \Sigma_l l - \Sigma_u u.$$

$$\tilde{r}_{dx} = r_{dx} + \Sigma_l r_{pl} + r_{dl} + S_l^{-1} r_{cl}$$

$$+ \Sigma_u r_{pu} - r_{du} - S_u^{-1} r_{cu}.$$

$$\tilde{r}_{d\tau} = r_{d\tau} + \tau^{-1} r_{c\tau} - l^T (\Sigma_l r_{pl} + r_{dl} + S_l^{-1} r_{cl})$$

$$+ u^T (-\Sigma_u r_{pu} + r_{du} + S_u^{-1} r_{cu})$$

The system in (14) is solved by performing a symmetric indefinite factorization (defined in [38]) of the matrix

$$\begin{pmatrix} Q + \Sigma_l + \Sigma_u & A^T \\ A & 0 \end{pmatrix} = LDL^T \quad (15)$$

to obtain the lower triangular matrix L and D matrix with 1×1 or 2×2 factors. These factors are used to backsolve

two right hand sides

$$LDL^T \begin{pmatrix} w_1 & w_2 \end{pmatrix} = \underbrace{\begin{pmatrix} \tilde{q} & \tilde{r}_{dx} \\ -b & r_{px} \end{pmatrix}}_{\begin{pmatrix} r_1 & r_2 \end{pmatrix}} \quad (16)$$

Using w_1, w_2 obtained from (16) we express $(\Delta x, \Delta \lambda_x)$ as

$$\begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = w_2 - w_1 \Delta \tau. \quad (17)$$

Substituting (17) into the last equation in (14) yields the following linear system in the only unknown $\Delta \tau$ as

$$(-r_1^T w_1 + \Sigma_\tau) \Delta \tau = \tilde{r}_{d\tau} - r_1^T w_2. \quad (18)$$

The quantities $(\Delta x, \Delta \lambda_x)$ and other step quantities can be computed by back substitution in (17), (13) and (11).

The main effort in the step computation is the factorization of the matrix W which scales with $(n+m)$ as claimed.

B. Predictor-Corrector IIPM

In this section, we present a predictor-corrector IIPM [28] for the HSQP (5). In the following, we denote by v the iterate $(x, s_l, s_u, \tau, \lambda_x, \lambda_l, \lambda_u, z_l, z_u, z_\tau)$ and subvectors of v by $v_p = (x, s_l, s_u, \tau)$, $v_d = (\lambda_x, \lambda_l, \lambda_u)$ and $v_c = (z_l, z_u, z_\tau)$. The left hand sides of the equations in (8) are denoted respectively as $R_{dx}(v)$, $R_{dl}(v)$, $R_{du}(v)$, $R_{px}(v)$, $R_{pl}(v)$, $R_{pu}(v)$, $R_{cl}(v)$, $R_{cu}(v)$, $R_{d\tau}(v)$, and $R_{c\tau}(v)$. For sake of brevity, we will use $R_d(v)$, $R_p(v)$, $R_c(v)$ to denote

$$R_d(v) = (R_{dx}(v), R_{dl}(v), R_{du}(v), R_{d\tau}(v)),$$

$$R_p(v) = (R_{px}(v), R_{pl}(v), R_{pu}(v)),$$

$$R_c(v) = (R_{cl}(v), R_{cu}(v), R_{c\tau}(v)).$$

In a similar manner, r_d , r_p , and r_c will also denote corresponding collections of the right hand sides in (10).

At each iteration of the IIPM, starting from the iterate v^k the algorithm computes two steps called the *predictor or affine step* $(\Delta v^{k,a})$ and *corrector step* $(\Delta v^{k,c})$ which are then combined to produce the new point v^{k+1} . In the following, we will suppress the dependence on k for sake of clarity of presentation when the context is unambiguous.

The predictor or affine step Δv^a is computed by solving (10) with the right hand side set as

$$(r_d, r_p, r_c) = -(R_d(v), R_p(v), R_c(v)). \quad (19)$$

Using this computed step the algorithm determines the scalars $\alpha_p^a = \alpha_{\max,p}(v, \Delta v^a)$ and $\alpha_d^a = \alpha_{\max,d}(v, \Delta v^a)$ where the functions $\alpha_{\max,p}(v, \Delta v)$ and $\alpha_{\max,d}(v, \Delta v)$ are defined as

$$\alpha_{\max,p}(v, \Delta v) = \max \left\{ \alpha \left| \begin{array}{l} \alpha \in [0, 1] \\ s_l + \alpha \Delta s_l \geq 0 \\ s_u + \alpha \Delta s_u \geq 0 \end{array} \right. \right\} \quad (20a)$$

$$\alpha_{\max,d}(v, \Delta v) = \max \left\{ \alpha \left| \begin{array}{l} \alpha \in [0, 1] \\ z_l + \alpha \Delta z_l \geq 0 \\ z_u + \alpha \Delta z_u \geq 0 \end{array} \right. \right\}. \quad (20b)$$

The scalars α_p^a, α_d^a determine the largest step length so that the $(s_l^a, s_u^a, z_l^a, z_u^a)$ components of the iterate v^a (defined below) are nonnegative

$$\begin{aligned} v_p^a &= v_p + \alpha_p^a \Delta v_p^a, v_d^a = v_d + \alpha_d^a \Delta v_d^a, \\ v_c^a &= v_c + \alpha_d^a \Delta v_c^a. \end{aligned} \quad (21)$$

The corrector step Δv^c is also computed from (10) by setting the right sides as

$$\begin{aligned} (r_d, r_p) &= (0, 0) \\ r_c &= (\sigma \mu 1_n - \Delta S_l^a \Delta z_l^a, \sigma \mu 1_n - \Delta S_u^a \Delta z_u^a, \\ &\quad \sigma \mu - \Delta \tau^a \Delta z_\tau^a) \end{aligned} \quad (22)$$

where $\mu = \|R_c(v)\|_1 / (2n+1)$ and $\mu^a = \|(R_c(v^a))\|_1 / (2n+1)$ and $\sigma = (\mu^a / \mu)^3$. A combined step is defined as $\Delta v = \Delta v^a + \Delta v^c$ and the next iterate v^{k+1} is obtained as

$$\begin{aligned} (v_p^{k+1}, v_d^{k+1}, v_c^{k+1}) \\ = (v_p^k, v_d^k, v_c^k) + \kappa (\alpha_p^c \Delta v_p, \alpha_d^c \Delta v_d, \alpha_d^c \Delta v_c) \end{aligned} \quad (23)$$

where $\kappa \in (0, 1)$ is a damping factor and $\alpha_p^c = \alpha_{\max, p}(v, \Delta v)$, $\alpha_d^c = \alpha_{\max, d}(v, \Delta v)$.

The termination of the algorithm is based on the error in satisfying the dual, primal and complementarity conditions are respectively denoted as $e_d(v), e_p(v), e_c(v)$ given by

$$e_d(v) = \frac{\|R_d(v)\|_\infty}{1 + \max\{\|Q\|_\infty, \|q\|_\infty\}} \quad (24a)$$

$$e_p(v) = \frac{R_p(v)}{1 + \|b\|_\infty} \quad (24b)$$

$$e_c(v) = \max \left\{ \left\| \frac{R_{cl}(v)}{1 + |l|} \right\|_\infty, \left\| \frac{R_{cu}(v)}{1 + |u|} \right\|_\infty, R_{c\tau}(v) \right\}. \quad (24c)$$

Note that the division in the first two terms of (24c) are componentwise. Algorithm 1 summarizes the steps for computing an optimal point to HSQP satisfying the errors (24) to a tolerance of ϵ .

Algorithm 1: IIPM for HSQP (5)

Data: $\kappa \in (0, 1)$ and $\epsilon > 0$.

- 1 Choose v^0 with $s_l^0, s_u^0, \tau^0, z_l^0, z_u^0, z_\tau^0 > 0$.
 - 2 Set $k = 0$.
 - 3 **while** $\max\{e_d(v^k), e_p(v^k), e_c(v^k)\} > \epsilon$ **do**
 - 4 Solve (10) by setting (r_d, r_p, r_c) as defined in (19) to obtain $\Delta v^{k,a}$.
 - 5 Determine $v^{k,a}$ using (21).
 - 6 Solve (10) by setting (r_d, r_p, r_c) as defined in (22) to obtain $\Delta v^{k,c}$.
 - 7 Determine the new iterate v^{k+1} using (23).
 - 8 Set $k = k + 1$.
-

V. WARM-START OF IIPM

MPC and global optimization of MIQPs involve the solution of a sequence of QPs that are closely related through perturbations in the problem data. By problem data, we refer

to $\mathcal{D} = \{Q, q, A, b, l, u\}$. The nominal QP that has been solved using \mathcal{D} is referred to simply as \mathcal{D} , and the optimal solution is denoted by $v^*(\mathcal{D})$ where v^* is a collection as defined in §IV-B. The perturbed QP, for which a solution is desired, is referred to by $\mathcal{D} + \Delta\mathcal{D} = \{Q + \Delta Q, A + \Delta A, b + \Delta b, l + \Delta l, u + \Delta u\}$. In the case of MPC for linear dynamical systems, only $\Delta b \neq 0$ for the QPs solved at successive time instants. In the global optimization of MIQPs, $\Delta l \neq 0$ or $\Delta u \neq 0$ for the QPs at parent and child nodes. *Warm-start* refers to the idea of using $v^*(\mathcal{D})$ to determine the solution $v^*(\mathcal{D} + \Delta\mathcal{D})$. For small $\Delta\mathcal{D}$, it is typically expected that the algorithm converges in fewer iterations as compared to not using $v^*(\mathcal{D})$.

IIPMs are in general not conducive for warm-starting due to the strict positivity requirement for certain components of the initial iterate v^0 (refer to Line 2 in Algorithm 1). We refer the interested to [33] for an excellent survey on the attempts at warm-starts for IIPM. Engau et al [33] applied an IIPM on a slacked reformulation of Linear Programs (LPs). The slacked formulation allowed to warm-start the algorithm for $\mathcal{D} + \Delta\mathcal{D}$ in an effective manner. Our slacked formulation is inspired by [33]. To the best of our knowledge, the slacked approach has not been considered for warm-start of QPs.

As shown in § IV-A, the addition of slacks does not increase the computational effort in the IIPM. We show in the following how the slacks help to reduce initial infeasibility for $\mathcal{D} + \Delta\mathcal{D}$. We consider MPC i.e. only $\Delta b \neq 0$.

The iteration complexity of IIPMs scales as [28], [33]

$$\log \left(\max \left\{ \frac{\|R_d(v^0)\|_\infty}{\epsilon}, \frac{\|R_p(v^0)\|_\infty}{\epsilon}, \frac{n\zeta^2}{\epsilon} \right\} \right)$$

where the initial iterate v^0 is chosen as

$$\begin{aligned} v_p^0 &= (\zeta 1_n, \zeta 1_n, \zeta 1_n), v_d^0 = (0, 0, 0), \\ v_c^0 &= (\zeta 1_n, \zeta 1_n), \end{aligned} \quad (25)$$

and $\zeta \geq \|(x^*, s_l^*, s_u^*, z_l^*, z_u^*)(\mathcal{D})\|_\infty$. If the perturbation $\Delta\mathcal{D}$ is sufficiently small then ζ is likely to be an upper bound for the solution of the $\mathcal{D} + \Delta\mathcal{D}$ as well, i.e. $\zeta \geq \|(x^*, s_l^*, s_u^*, z_l^*, z_u^*)(\mathcal{D} + \Delta\mathcal{D})\|_\infty$. We propose to choose the initial iterate for $\mathcal{D} + \Delta\mathcal{D}$ for some $\beta \in (0, 1)$ as

$$v_p^{ws} = (x^*, \beta s_l^* + (1 - \beta)\zeta 1_n, \beta s_u^* + (1 - \beta)\zeta 1_n, \beta \tau^* + (1 - \beta)) \quad (26a)$$

$$v_d^{ws} = (\lambda_d^*, \lambda_l^*, \lambda_u^*), \quad (26b)$$

$$v_c^{ws} = (\beta z_l^* + (1 - \beta)\zeta 1_n, \beta z_u^* + (1 - \beta)\zeta 1_n). \quad (26c)$$

Since only $\Delta b \neq 0$ in the MPC context we have by substituting v^{ws} into (8) that

$$\|R_d(v^{ws})\|_\infty \leq (1 - \beta) \max(|\tau^* - \zeta|, \|z_l^* - \zeta 1_n\|_\infty, \|z_u^* - \zeta 1_n\|_\infty)$$

$$\|R_p(v^{ws})\|_\infty \leq \max(\|\Delta b\|_\infty, (1 - \beta)\|s_l^* - \zeta 1_n\|_\infty, (1 - \beta)\|s_u^* - \zeta 1_n\|_\infty)$$

$$\|R_c(v^{ws})\|_\infty \leq \zeta^2.$$

For all Δb sufficiently small, the initial residuals using the warm start iterate v^{ws} will be smaller than that using the

v^0 in (25). As a result the worst-case iteration complexity when using the warm-start iterate v^{ws} is never worse than using v^0 in (25). This is the rationale for using the iterate v^{ws} . However, the typical savings that can be obtained in computational time depend on the magnitude of the perturbation $\Delta\mathcal{D}$.

Another aspect in the above analysis is that ζ is never available for \mathcal{D} . IIPMs typically resort to setting $\zeta = 1$ in (25) for lack of a better estimate. For example, the authors of [39] set $\zeta = 1$ in (25) and use a convex combination of v^0 and $v^*(\mathcal{D})$ when performing a warm-start. They do not employ a slacked formulation. Our approach of setting components to be a convex combination is inspired by [39].

VI. NUMERICAL EXPERIMENTS

We implemented the predictor-corrector IIPM described in Algorithm 1 using Julia [34]. The matrix factorization required for the step computations in §IV-A is performed using MKL Pardiso [35]. The goal of the numerical experiments is to demonstrate: (i) the computational efficiency of the homogeneous formulation based IIPMs on feasible QPs; (ii) the robustness of the homogeneous formulation for detecting infeasible QPs; and (iii) effectiveness of the slacked formulation in warm-start of QPs arising in MPC. In this paper, we compare our performance on (i)-(iii) with other IPM implementations such as ECOS [12], Ipopt [8] and MOSEK [36]. ECOS and MOSEK both employ the HSD formulation for IPMs to convert QPs to SOCPs whereas Ipopt uses the standard IPM formulation for QPs [28]. We executed ECOS, Ipopt and MOSEK through their JuMP interfaces [40]. We refer to our approach as HSQP in the following. All tests were performed on a MacBook with 2.6 GHz Intel Core i7 processor and 16 GB of memory.

A. Strictly Convex and Feasible QPs

We generated 8 different test sets of strictly convex QPs to benchmark HSQP against other popular IPM solvers. In particular, we generate QPs with two different variable sizes $n \in \{100, 1000\}$, constraint size $m \in \{100, 1000\}$. We also used two different densities for the Q and the A matrix, $d \in \{0.1, 0.5\}$. The Q matrix is modified as $Q + \max(-1.1 * \sigma_{\min}(Q), 0)$, with $\sigma_{\min}(Q)$ the smallest eigenvalue of Q , in order to make it positive definite. All variables are imposed bounds of 0 and 1. Thus, we created 8 different problem sets for each combination of (n, m, d) . For each set we created 100 random QPs and present the statistics of times taken by each of the solver.

Figures 1 and 2 show box plots for each solver over the 100 instances in the test set. Figure 1 consists of QPs of smaller dimensions $(n, m) = (100, 25), (100, 50)$. On these instances, MOSEK and HSQP perform comparably on both densities of 0.1, 0.25. ECOS and Ipopt are slower but the overall computational times are small. The computational times of all solvers increase with problem density as expected. On the larger instances of $(n, m) = (1000, 250), (1000, 500)$ we see that HSQP performs better

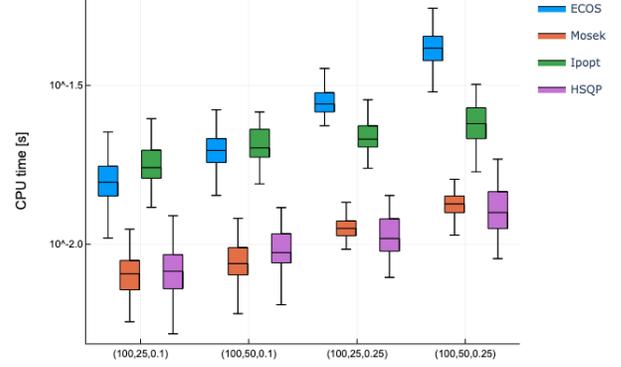


Fig. 1. Comparison of the CPU times for solving the smaller feasible instances. The horizontal axis shows the instances set as a combination of (n, m, d) .

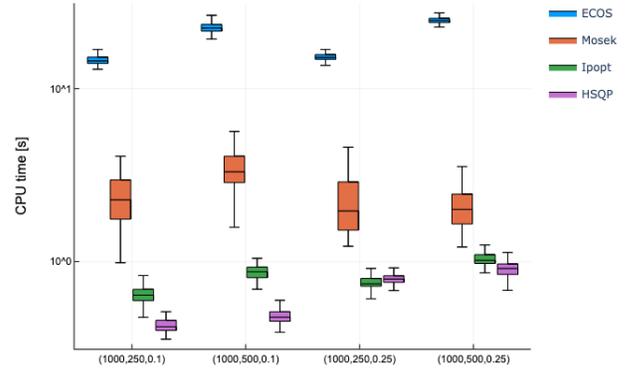


Fig. 2. Comparison of the CPU times for solving the larger feasible instances. The horizontal axis shows the instances set as a combination of (n, m, d) .

than all other approaches with MOSEK being quite comparable.

B. Strictly Convex and Infeasible QPs

We use the same test set generated in §VI-A and create the infeasible instances by changing the right hand side of the first constraints to $\sum_j \max(0, A_{ij}) + 1$ which is inconsistent with the variable bounds of $[0, 1]$. For the infeasible instances, we exclude Ipopt from the comparisons since Ipopt cannot detect infeasibility. Our goal is to compare the IIPM performance on such instances. To this end we deactivated the presolve option in MOSEK. Figure 3 shows the box plots for the solvers on the larger instances. All solvers are able to correctly detect infeasibility of the instances. The computational times of HSQP are better than those of ECOS and MOSEK.

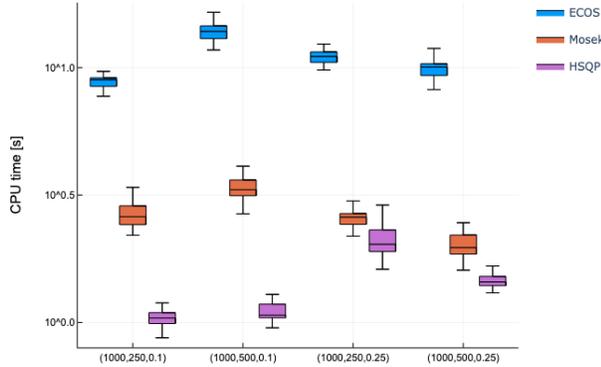


Fig. 3. CPU time comparison in solving the larger infeasible instances. The horizontal axis shows the instances set as a combination of (n, m, d) .

C. MPC Problems & Warm-start

We consider the problem of driving a spacecraft system to the origin [20]. The system has 6 states, 3 inputs and starts from the initial state of $-[2.8, 0.01, 1][km]$ at rest. We consider 4 different horizon lengths, N , for MPC. The length of the horizon determines the number of variables $n = 9N + 6$ and constraints $m = 6N + 6$ of the QP that is solved at each time-step. For each N we solve the MPC problems 100 times. Figure 4 shows the box plots for all four solvers and also HSQP with the warm-start (HSQP-ws) for all solves of QPs after the first one. We used a value $\beta = 0.99$ to obtain v^{ws} in (26) to initialize the IIPM. The Q, A matrices in the MPC instances have densities that scale as $1/N$ which makes them sparser than the test sets we generated. On these problem, ECOS and HSQP are comparable for $N = 10, 20$ while ECOS is better for larger N . The HSQP-ws reduces the computational times by almost 50% when compared to HSQP. This clearly demonstrates the benefit of using the slacked QP formulation. Further, HSQP-ws resulted to be the fastest among all the solver as it does better than ECOS for all the horizon lengths.

VII. CONCLUSIONS & FUTURE WORK

We presented a predictor-corrector IIPM using a novel homogeneous formulation of convex QPs. The homogeneous formulation is obtained on slacked reformulation of QPs. We showed that the increase in dimensions due to the slacks does not affect the complexity of the step computation. Further, we also presented arguments for why the slacked formulation is conducive to warm-starts. Numerical experiments have shown that the formulation compares favorably against the existing implementations. Our approach is also shown to reduce the computational time per iteration by 50% when warm-starts are used in the MPC applications.

We also outline a number of directions for future work.

- MPC problems have a block-diagonal structure in the Hessian and an almost block-diagonal structure in the

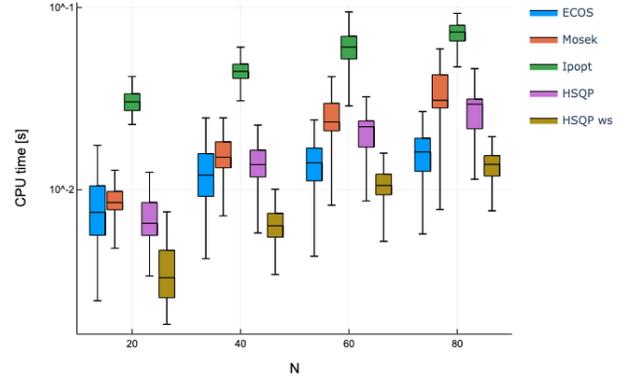


Fig. 4. CPU time comparison in solving the MPC problem to control the spacecraft system

constraints that can be exploited in the step computation [9]. This can reduce the computational work involved for the matrix factorization in §IV-A. The decomposition in combination with the warm-starts can allow our approach to solve MPC at higher frequencies which is critical in robotic applications.

- More extensive comparisons on other problem classes and other algorithms.
- We will also explore the implementation of algorithms such as ADMM or active-set on the homogeneous formulation. These algorithms are more amenable to warm-starts than the IIPM. However, they have not been studied in the context of the homogeneous formulation.
- We also intend to make the code publicly available for the wider community by the time of submission of the paper.

REFERENCES

- [1] A. U. Raghunathan, “Homogeneous formulation of convex quadratic programs for infeasibility detection,” in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 968–973.
- [2] H. Markowitz, “Portfolio selection,” *Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [3] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.
- [4] D. Mayne, J. Rawlings, and M. Diehl, *Model predictive control: Theory, Computation and Design*, 2nd ed. Santa Barbara: Nob Hill, LLC, 2020.
- [5] C. J. Nohra, A. U. Raghunathan, and N. Sahinidis, “Spectral relaxations and branching strategies for global optimization of mixed-integer quadratic programs,” *SIAM Journal on Optimization*, vol. 31, no. 1, pp. 142–171, 2021.
- [6] C. J. Nohra, A. U. Raghunathan, and N. V. Sahinidis, “SDP-quality bounds via convex quadratic relaxations for global optimization of mixed-integer quadratic programs,” *Mathematical Programming*, 2021.
- [7] P. Boggs and J. Tolle, “Sequential Quadratic Programming,” *Acta Numerica*, vol. 4, pp. 1–51, 1995.
- [8] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [9] C. Rao, S. Wright, and J. Rawlings, “Application of interior-point methods to model predictive control,” *Journal of Optimization Theory and Applications*, p. 723–757, 1998.

- [10] R. Bartlett and L. Biegler, "QPSchur: A dual, active-set, schur-complement method for large-scale and structured convex quadratic programming," *Optimization and Engineering*, vol. 7, p. 5–32, 2006.
- [11] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit mpc," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, p. 816–830, 2008.
- [12] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An sopc solver for embedded systems," in *2013 European Control Conference (ECC)*, 2013, pp. 3071–3076.
- [13] S. Richter, C. Jones, and M. Morari, "Computational complexity certification for real-time mpc with input constraints based on the fast gradient method," *IEEE Trans. Autom. Control*, vol. 57, no. 6, p. 1391–1403, 2012.
- [14] P. Giselsson, "Optimal preconditioning and iteration complexity bounds for gradient-based optimization in model predictive control," in *2013 American Control Conference*, 2013, pp. 358–364.
- [15] P. Patrinos, L. Stella, and A. Bemporad, "Douglas-Rachford splitting: Complexity estimates and accelerated variants," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 4234–4239.
- [16] A. U. Raghunathan and S. Di Cairano, "Alternating direction method of multipliers for strictly convex quadratic programs: Optimal parameter selection," in *American Control Conference*, 2014, pp. 4324–4329.
- [17] P. Giselsson and S. Boyd, "Diagonal scaling in Douglas-Rachford splitting and admm," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 5033–5039.
- [18] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, p. 18–33, 2014.
- [19] R. Quirynen, A. Knyazev, and S. D. Cairano, "Block structured preconditioning within an active-set method for real-time optimal control," in *2018 European Control Conference (ECC)*, 2018, pp. 1154–1159.
- [20] D. Liao-McPherson and I. Kolmanovsky, "Fbstab: A proximally stabilized semismooth algorithm for convex quadratic programming," *Automatica*, vol. 113, p. 108801, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109819306648>
- [21] D. Frick, A. Domahidi, and M. Morari, "Embedded optimization for mixed logical dynamical systems," *Computers & Chemical Engineering*, vol. 72, pp. 21–33, 2015.
- [22] V. Naik and A. Bemporad, "Embedded mixed-integer quadratic optimization using accelerated dual gradient projection," in *IFAC-PapersOnLine*, vol. 50, no. 1, 2017, pp. 10 723–10 728.
- [23] B. Stellato, V. Naik, A. Bemporad, P. Goulart, and S. Boyd, "Embedded mixed-integer quadratic optimization using the osqp solver," in *2018 European Control Conference (ECC)*, 2018, pp. 1536–1541.
- [24] P. Hespanhol, R. Quirynen, and S. Di Cairano, "A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control," in *2019 18th European Control Conference (ECC)*, 2019, pp. 2763–2768.
- [25] A. U. Raghunathan and S. Di Cairano, "Infeasibility detection in alternating direction method of multipliers for convex quadratic programs," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 5819–5824.
- [26] G. Banjac, P. Goulart, B. Stellato, and S. Boyd, "Infeasibility detection in the alternating direction method of multipliers for convex optimization," *Journal of Optimization Theory and Applications*, vol. 183, no. 2, pp. 490–519, 2019. [Online]. Available: <https://doi.org/10.1007/s10957-019-01575-y>
- [27] D. Applegate, M. Díaz, H. Lu, and M. Lubin, "Infeasibility detection with primal-dual hybrid gradient for large-scale linear programming," 2021. [Online]. Available: <https://arxiv.org/abs/2102.04592>
- [28] S. Wright, *Primal-Dual Interior-Point Methods*. SIAM, 1997.
- [29] Y. Ye, M. J. Todd, and S. Mizuno, "An $o(\sqrt{n})$ -iteration homogeneous and self-dual linear programming algorithm," *Mathematics of Operations Research*, vol. 19, pp. 53–67, 1994.
- [30] X. Xu, P. Hung, and Y. Ye, "A simplified homogeneous and self-dual linear programming algorithm and its implementation," *Annals of Operations Research*, vol. 62, pp. 151–172, 1996.
- [31] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *J Optim Theory Appl*, vol. 169, p. 1042–1068, 2016.
- [32] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
- [33] A. Engau, M. F. Anjos, and A. Vannelli, "On interior-point warmstarts for linear and combinatorial optimization," *SIAM J Optimization*, vol. 20, no. 4, pp. 1828–1861, 2010.
- [34] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: <https://doi.org/10.1137/141000671>
- [35] O. Schenk and K. Gartner, "Solving unsymmetric sparse systems of linear equations with pardiso," *J. of Future Generation Computer Systems*, vol. 20, no. 3, pp. 475–487, 2004.
- [36] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. [Online]. Available: <http://docs.mosek.com/9.0/toolbox/index.html>
- [37] O. Mangasarian, *Nonlinear Programming*, ser. Classics in Applied Mathematics. SIAM, 1994.
- [38] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.
- [39] A. Skajaa, E. D. Andersen, and Y. Ye, "Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems," *Mathematical Programming Computation*, vol. 5, no. 1, pp. 1–25, 2013.
- [40] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.