Optimal Parameter Selection for ADMM: Quadratically Constrained Quadratic Program

Hoai-Nam Nguyen

***Abstract*—We propose two new algorithms for solving quadratically constrained quadratic programming (QCQP) problems arising from real-time optimization based control such as model predictive control or interpolating control. The proposed algorithms are based on the Alternating Direction Method of Multipliers (ADMM). ADMM is a powerful tool for solving a wide class of constrained optimization problems. There are two main challenges when applying ADMM: Its performance depends greatly on the efficiency of solving the suboptimization problems associated with the ADMM at each iteration; it is not trivial to find the correct penalty parameters. For the first challenge, we provide a way to reformulate the original QCQP problem into a form such that there exist analytical solutions for the suboptimization problems. Hence, the computational cost per iteration is low. For the second challenge, we provide two procedures to compute systematically the penalty parameters. In the first procedure, a closed-form expression for the optimal constant scalar parameter is derived in terms of the matrix condition number. In the second one, the penalty parameters are adaptively tuned to achieve fast convergence. The results are validated via numerical simulations.**

***Index Terms*—Mathematical programming, optimization methods, scalability.**

I. Introduction

The alternating direction method of multipliers (ADMM) provides a versatile tool for solving a broad variety of constrained optimization problems in several fields, see, e.g., [4], [7], [10], [14], [30], [31], [32]. The method is based on the augmented Lagrangian where additional variables, *the so-called penalty parameters*, are introduced. There are two main attractive features of ADMM. The first one is its simplicity, provided that the solution of the suboptimization problems associated with the ADMM at each iteration is easily computed. The second feature is that ADMM is guaranteed to converge for all positive values of its penalty parameters. However, it is well known that the rate of convergence of ADMM depends greatly on the proper choice of the penalty parameters. Except a particular case of quadratic programming (QP), where explicit expression for a scalar constant penalty parameter was proposed [12], [27], to the best of the author's knowledge, there is no systematic way to calculate the penalty parameters. The users need to tune them manually for their particular applications.

Our interest in ADMM is mainly motivated by its potential application to model predictive control (MPC) [17], [25]. MPC is a control technique that solves at each time instant a finite horizon optimal control problem formulated from the system dynamics, constraints, and a cost function. One of the most well known MPC formulations is the dual-mode MPC, where recursive feasibility and asymptotic stability are guaranteed. In the dual-mode MPC, the terminal state is imposed to lie in a terminal set, which is invariant and constraint-admissible. For linear systems subject to linear and/or quadratic constraints, with a quadratic cost function and an ellipsoidal terminal constraint, the online optimization problem is a QCQP.

We can also use the QCQP formulation in robust tube MPC [20]. In this technique, a minimal robustly invariant set (mRIS) is required to be built offline. The typical approach involves using a polyhedral set to describe the mRIS. However, constructing such a polyhedral set, especially for high-dimensional systems, is computationally challenging. An alternative way is to use ellipsoids to describe the mRIS. This approach simplifies the offline construction process, and leads to a QCQP formulation. The approach is advantageous for high-dimensional systems, where polyhedral based methods become impractical. Note that ADMM was already discussed in the context of QCQP-based MPC in [17] and [25]. However in these works, the penalty parameter is a scalar, and is found by the trial and error method. In addition, only box constraints are considered in [17]. In [25], the ellipsoid matrices are required to be positive definite.

Another potential application of ADMM is in robust prediction dynamics based MPC [6], [23], [24], or in interpolating control [22]. They are also real-time optimization-based control techniques, where the notion of set invariance is heavily exploited. In these techniques, the resulting online optimization problem is a QCQP if ellipsoid or the intersection of ellipsoids are used for the invariant set.

In this article, we propose two new ADMM-based algorithms to solve a generic QCQP. Our first objective is to reformulate the QCQP into an equivalent form such that ADMM can be applied, and that the solution of the suboptimization problems can be analytically computed. Our second and main contribution is to provide two new procedures to calculate the penalty parameters. In the first procedure, we are looking for a constant scalar penalty parameter. We show how to obtain a closed-form expression in this case. Since the considered QCQP includes the QP in [12] as a particular case, the first method provides a generalization of the results in [12]. In addition, we improve the results in [12] by showing that the residuals sequence is

Received 21 January 2025; accepted 18 April 2025. Date of publication 2 May 2025; date of current version 29 September 2025. This work was support by FMJH Program Gaspard Monge for optimization and operations research and their interactions with data science. Recommended by Associate Editor G. Notarstefano.

The author is with the SAMOVAR—Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France (e-mail: hoai-nam.nguyen@telecom-sudparis.eu).

Digital Object Identifier 10.1109/TAC.2025.3566632

$Q-$linearly convergent. In [12], it was only possible to establish the $R-$linear convergence for the residuals.

In the second procedure, we are looking for a vector of the time-varying penalty parameters. The keystone of the second procedure is based on the observation that if we employ the optimal dual variables of the original QCQP for the penalty parameters, the ADMM algorithm can converge in one step. Clearly, the optimal dual variables of the original QCQP are unknown. We provide a way to update iteratively the penalty parameters such that they converge to the optimal dual variables. The resulting adaptive ADMM algorithm is fully automated. A connection between our second procedure and the well known residual balancing method [15] is presented.

Note that as our QCQP is convex, the solution can be obtained by using, e.g., the interior point methods (IPMs) [21] or the projected gradient methods (PGMs) [3]. IPMs have well-defined convergence properties, and can solve convex QCQPs in polynomial time. However, each iteration requires sophisticated and heavy computational tasks to be performed, e.g., solving Newton's type systems. As a result, a single iteration of such a polynomial time algorithm is often too expensive to be of practical use in real-time. In contrast, PGMs are relatively simple. However, each iteration requires computing the orthogonal projection onto the feasible set, which can be challenging and computationally expensive for QCQPs.

The rest of this article is organized as follows. Section II is concerned with the problem formulation and preliminaries. Section III is dedicated to the QCQP problem with only one quadratic constraint. Then in Section IV, the general case of QCQP is considered. In Section V, we recall a brief theory of MPC for linear systems under linear and quadratic constraints. Two simulated examples are evaluated in Section VI. Finally, Section VII concludes this article.

Notations: A positive definite (sem-definite) matrix P is denoted by $P \succ 0$ ($P \succeq 0$). We denote by $\mathbb{R}$ the set of real numbers, by $\mathbb{R}^{n\times m}$ the set of real $n \times m$ matrices, and by $\mathbb{S}^n_+ (\mathbb{S}^n)$ the set of positive definite (semidefinite) $n \times n$ matrices. $||x||^2_P = x^T P x$, $||x||_2 = \sqrt{x^T x}$. For a given $P \in \mathbb{S}^n_+$, $\mathcal{E}(P)$ represents the following ellipsoid:

$$\mathcal{E}(P) = \{x \in \mathbb{R}^n : x^T P^{-1} x \leq 1\}.$$

The set $\{1, 2, \ldots, n\}$ is denoted by $\overline{1,n}$. $\mathbf{I}, \mathbf{0}, \mathbf{1}$ are, respectively, the identity matrix, the zero matrix, and the all-one matrix of appropriate dimension. A sequence $\{x(k)\}$ converging to x^* is said to converge at: i) $Q-$linear rate if $||x(k+1) - x^*|| \leq \kappa ||x(k) - x^*||$ with $0 < \kappa < 1$; ii) at $R-$linear rate if $||x(k) - x^*|| \leq \kappa(k)$, where $\kappa(k)$ is $Q-$linearly convergent.

II. Problem Formulation and Preliminaries

A. Problem Formulation

We aim to solve the following QCQP:

$$\min_x \left\{\frac{1}{2} x^T H x + f^T x\right\} \tag{1}$$

$$\text{s.t. } (x + b_i)^T Q_i (x + b_i) \leq 1, \forall i = \overline{1,m} \tag{2}$$

where $x \in \mathbb{R}^n$, $H \in \mathbb{S}^n_+$, $Q_1 \in \mathbb{S}^n, \ldots, Q_m \in \mathbb{S}^n$, and $f, b_1, \ldots, b_m \in \mathbb{R}^n$.

Throughout, we assume that (1), (2) is strictly feasible. This implies that Slater's condition is satisfied. Hence, strong duality holds. Note that the matrices $Q_i, \forall i = \overline{1,m}$ are required only to be positive semi-definite. As such, linear constraints such as

$$\underline{u}_i \leq g_i^T x \leq \bar{u}_i \tag{3}$$

with $g_i \in \mathbb{R}^n$, $g_i \neq 0$, and $\underline{u}_i < \bar{u}_i$ can be reformulated as (2) as follows. Define

$$u_i = \frac{\bar{u}_i - \underline{u}_i}{2}, \tilde{g}_i = \frac{g_i}{u_i}. \tag{4}$$

We can equivalently rewrite (3) as (2) with

$$Q_i = \tilde{g}_i \tilde{g}_i^T, b_i = -\left(\frac{\underline{u}_i + \bar{u}_i}{2\tilde{g}_i^T \tilde{g}_i}\right) \tilde{g}_i. \tag{5}$$

We can also recast the quadratic constraints

$$x^T \tilde{Q}_i x + 2\tilde{b}_i^T x \leq \tilde{c}_i$$

as (2) with b_i being any vector such that $\tilde{Q}_i b_i = \tilde{b}_i$ and

$$Q_i = \frac{\tilde{Q}_i}{\tilde{c}_i + b_i^T \tilde{b}_i}.$$

The associated Lagrangian for (1), (2) is

$$L(x, \theta) = \frac{1}{2} x^T H x + f^T x + \sum_{i=1}^m \frac{\theta_i}{2} \left((x + b_i)^T Q_i (x + b_i) - 1\right) \tag{6}$$

where $\theta \in \mathbb{R}^m$ is the dual variable. The $\frac{1}{2}$ factor in θ_i is primarily for scaling purposes. Using (6), the optimal solution to (1), (2) is given by

$$x^* = -\left(H + \sum_{i=1}^m \theta_i^* Q_i\right)^{-1} \left(f + \sum_{i=1}^m \theta_i^* Q_i b_i\right) \tag{7}$$

where θ^* is the solution of the following optimization problem:

$$\max_\theta \{V(\theta)\}$$
$$\text{s.t. } \theta_i \geq 0, \forall i = \overline{1,m} \tag{8}$$

with $V(\theta)$ being the Lagrange dual function [3], [5], i.e.,

$$V(\theta) = \frac{1}{2} \sum_{i=1}^m \left(\theta_i b_i^T Q_i b_i - \theta_i\right) - \frac{1}{2} \left(f + \sum_{i=1}^m \theta_i Q_i b_i\right)^T \left(H + \sum_{i=1}^m \theta_i Q_i\right)^{-1} \times \left(f + \sum_{i=1}^m \theta_i Q_i b_i\right). \tag{9}$$

Since strong duality is satisfied, we have the optimal value of (1), (2) is equal to the attained optimal value of the dual problem (8). Hence, $V(\theta)$ is bounded from above by the optimal value of (1), (2). Using the KKT conditions [3], [5], the optimal solution θ^* of (8) satisfies

$$\nabla V(\theta^*)^T (\theta - \theta^*) \leq 0, \forall \theta \geq 0$$

or equivalently, θ^* is optimal if and only if $\theta^* \geq 0$, and $\forall i = \overline{1,m}$

$$\begin{cases} \nabla V_i(\theta^*) \leq 0, & \text{if } \theta_i^* = 0, \\ \nabla V_i(\theta^*) = 0 & \text{if } \theta_i^* > 0. \end{cases} \tag{10}$$

In (10), $\nabla V_i(\theta^*)$ is the ith component of the vector $\nabla V(\theta^*)$. We will use (9), (10) to show the convergence of the new algorithm in Section III-B.

B. Alternating Direction Method of Multipliers

We can use the ADMM to solve convex optimization problems of the form

$$\min_{x,z} \{g(x)\} \tag{11}$$

$$\text{s.t. } \begin{cases} Ax + Bz = c, \\ z \in Z \end{cases} \tag{12}$$

where $x \in \mathbb{R}^n, z \in \mathbb{R}^l$, $g : \mathbb{R}^n \to \mathbb{R}$ is a convex function, $A \in \mathbb{R}^{d\times n}$, $B \in \mathbb{R}^{d\times l}$ and $c \in \mathbb{R}^d$. $Z \subseteq \mathbb{R}^l$ is a closed and convex set.

The augmented Lagrangian for (11), (12) is

$$\begin{aligned}\mathcal{L}_\rho(x, z, y) &= g(x) + y^T (Ax + Bz - c) \\ &\quad + \frac{\rho}{2}||Ax + Bz - c||_2^2\end{aligned} \tag{13}$$

where $y \in \mathbb{R}^d$ is the dual variable for the equality constraint. $\rho > 0$ is the penalty parameter. For simplicity, ρ is a scalar in this section. However, it is possible to consider a vector of the penalty parameters as in Section IV.

In the following, we use k as iteration counter of the ADMM. At each iteration of the ADMM, we perform alternating minimization of $\mathcal{L}_\rho(x, z, y)$ over x and z. At iteration k we carry out the following steps:

$$x(k+1) = \arg\min_x \left\{ g(x) + y(k)^T Ax + \frac{\rho}{2}||Ax + Bz(k) - c||_2^2 \right\}, \tag{14}$$

$$z(k+1) = \arg\min_{z\in Z} \left\{ y(k)^T Bz + \frac{\rho}{2}||Ax(k+1) + Bz - c||_2^2 \right\}, \tag{15}$$

$$y(k+1) = y(k) + \rho\,(Ax(k+1) + Bz(k+1) - c)\,. \tag{16}$$

Under very mild assumptions, it is well known [9] that the ADMM converges to a globally optimal solution of problem (11), (12) for any value of $\rho > 0$. However, the number of iterations to have small residuals depends strongly on the value of ρ. A poor choice of ρ might significantly slow down the convergence of the method.

The stopping condition of the ADMM algorithm is determined by the primal r_p and dual r_d residuals given by

$$\begin{cases} r_p(k) = Ax(k) + Bz(k) - c, \\ r_d(k) = \rho A^T B\,(z(k) - z(k-1))\,. \end{cases} \tag{17}$$

The justification for using r_d as a residual for dual optimality can be found in [4]. The ADMM algorithm (14), (15), (16) returns a suboptimal solution point $(\tilde{x}^*, \tilde{z}^*, \tilde{y}^*)$ of (11), (12) where the suboptimality is determined by the stopping condition $||r_p(k)||_\infty \le \epsilon_p, ||r_d(k)||_\infty \le \epsilon_d$ with given tolerances $\epsilon_p > 0, \epsilon_d > 0$. The parameters ϵ_p, ϵ_d are often chosen relative to the scaling of the algorithm iterates [4].

III. ADMM Formulations for QCQP: The Case $m = 1$

In this section we study the case $m = 1$, i.e., we consider the following optimization problem:

$$\min_x \left\{ \frac{1}{2}x^T Hx + f^T x \right\} \tag{18}$$

$$\text{s.t. } (x + b)^T Q(x + b) \le 1 \tag{19}$$

with $b = b_1$, $Q = Q_1$. This case is considered first to introduce the main idea.

A. Optimal Fixed Penalty Parameter

The aims of this section are: 1) to provide a way to solve efficiently (18), (19) using ADMM; 2) to optimize the fixed penalty parameter ρ for minimizing the rate of convergence of the ADMM algorithm (14), (15), (16). Define L as a "square root" matrix of Q, i.e., $L^T L = Q$. Since Q is required to be positive semidefinite, one can always select L as a full row rank matrix with $L \in \mathbb{R}^{l\times n}$ where l is the rank of Q. For the ADMM implementation, define $z \in \mathbb{R}^l$ as

$$\frac{z}{\sqrt{\rho}} = Lx + Lb. \tag{20}$$

Remark 1: The motivation of introducing the auxiliary variable z as in (20) is that in this case the $z-$update in (15) can be computed explicitly as will be shown later. If z is defined as, e.g., $z = x$ or $z = x + b$, then there is no closed-form solution for the $z-$update in (15).

We rewrite the problem (18), (19) as

$$\min_{x,z} \left\{ \frac{1}{2}x^T Hx + f^T x \right\} \tag{21}$$

$$\text{s.t. } \begin{cases} Lx - \frac{z}{\sqrt{\rho}} + Lb = 0, \\ z^T z \le \rho. \end{cases} \tag{22}$$

It is clear that problem (18), (19) and problem (21), (22) have the same optimal solution x^*. For problem (21), (22), the associated augmented Lagrangian takes the following form:

$$\begin{aligned}\mathcal{L}_\rho(x, z, y) &= \frac{1}{2}x^T Hx + f^T x + y^T (Lx - \frac{z}{\sqrt{\rho}} + Lb) \\ &\quad + \frac{\rho}{2}||Lx - \frac{z}{\sqrt{\rho}} + Lb||_2^2.\end{aligned} \tag{23}$$

Using (14), the ADMM sub-problem for the $x-$update is an unconstrained quadratic program

$$\begin{aligned}\min_x &\left\{ \frac{1}{2}x^T (H + \rho L^T L)x \right.\\ &\left. - \left(\sqrt{\rho}L^T z(k) - L^T y(k) - \rho L^T Lb - f\right)^T x \right\}.\end{aligned} \tag{24}$$

Problem (24) has the unique solution, recall $L^T L = Q$

$$\begin{aligned} x(k+1) &= (H + \rho Q)^{-1} \left(\sqrt{\rho}L^T z(k) - L^T y(k) - f - \rho Qb\right) \\ &= (H + \rho Q)^{-1} \left(\sqrt{\rho}L^T z(k) - L^T y(k)\right) \\ &\quad - (H + \rho Q)^{-1} (f + \rho Qb)\,. \end{aligned} \tag{25}$$

Using (15), the $z-$update is the solution of the following optimization problem:

$$\begin{aligned}&\min_z \left\{ z^T z - 2v(k+1)^T z \right\} \\ &\text{s.t. } z^T z \le \rho\end{aligned} \tag{26}$$

where

$$v(k+1) = \sqrt{\rho}Lx(k+1) + \frac{1}{\sqrt{\rho}}y(k) + \sqrt{\rho}Lb. \tag{27}$$

The following lemma provides a slight generalization of a result in [24], [25]. It enables us to find the closed-form solution for (26).

Algorithm 1: Fixed Penalty Parameter - Case $m = 1$.

Require: $H, f, Q, b, L, z(0), u(0), \rho > 0, \epsilon_p > 0, \epsilon_d > 0$
- $k \leftarrow 0$
- **repeat**
  - $x(k+1) \leftarrow (H+\rho Q)^{-1}\sqrt{\rho}L^T(z(k)-u(k))$
    $\quad -(H+\rho Q)^{-1}(f+\rho Qb)$
  - $v(k+1) \leftarrow \sqrt{\rho}Lx(k+1)+u(k)+\sqrt{\rho}Lb$
  - **if** $v(k+1)^Tv(k+1) \leq \rho$ **then**
    - $z(k+1) \leftarrow v(k+1)$
  - **else**
    - $z(k+1) \leftarrow \frac{\sqrt{\rho}v(k+1)}{\sqrt{v(k+1)^Tv(k+1)}}$
  - $u(k+1) \leftarrow v(k+1) - z(k+1)$
  - $r_p(k+1) \leftarrow \frac{1}{\sqrt{\rho}}(u(k+1)-u(k))$
  - $r_d(k+1) \leftarrow -\sqrt{\rho}L^T(z(k+1)-z(k))$
  - $k \leftarrow k+1$
- **until** $||r_p(k)||_\infty \leq \epsilon_p$ and $||r_d(k)||_\infty \leq \epsilon_d$

Lemma 1: The optimal solution of (26) is given by

$$z(k+1) = \begin{cases} v(k+1), & \text{if } v(k+1)^Tv(k+1) \leq \rho \\ \frac{\sqrt{\rho}v(k+1)}{\sqrt{v(k+1)^Tv(k+1)}}, & \text{otherwise} \end{cases}. \tag{28}$$

Finally, using (16) the $y-$update is given as

$$\begin{aligned} y(k+1) &= y(k) + \rho\left(Lx(k+1) - \frac{z(k+1)}{\sqrt{\rho}} + Lb\right) \\ &= \sqrt{\rho}\left(\sqrt{\rho}Lx(k+1) + \frac{1}{\sqrt{\rho}}y(k) + \sqrt{\rho}Lb - z(k+1)\right) \end{aligned}$$

thus, using (27)

$$y(k+1) = \sqrt{\rho}\left(v(k+1) - z(k+1)\right). \tag{29}$$

Using (17), (20), the primal and dual residuals are given as

$$\begin{cases} r_p(k) = Lx(k) - \frac{z(k)}{\sqrt{\rho}} + Lb \\ r_d(k) = -\sqrt{\rho}L^T(z(k) - z(k-1)) \end{cases}. \tag{30}$$

Define $u(k) = \frac{y(k)}{\sqrt{\rho}}$. Using (27), one has

$$r_p(k) = \frac{1}{\sqrt{\rho}}\left(v(k) - u(k-1) - z(k)\right)$$

thus, using (29)

$$r_p(k) = \frac{1}{\sqrt{\rho}}\left(u(k) - u(k-1)\right). \tag{31}$$

Combining (25), (27), (28), (29), (31), we obtain Algorithm 1, which shows the particularization of the ADMM method for solving (21), (22). Since problem (21), (22) is strictly convex, it is well known [4] that Algorithm 1 converges to the fixed point (x^*, z^*, u^*), and that the fixed point x^* is the optimal solution of (21), (22).

In the following we aim to characterize the rate of convergence of Algorithm 1. We aim also to provide a way to compute the optimal ρ that minimizes this rate of convergence.

By substituting (25) into (27), one gets

$$\begin{aligned} v(k+1) &= Rz(k) + (I-R)u(k) \\ &\quad + \sqrt{\rho}L(H+\rho Q)^{-1}(Hb - f) \end{aligned} \tag{32}$$

where

$$R = \sqrt{\rho}L(H+\rho Q)^{-1}\sqrt{\rho}L^T. \tag{33}$$

Note that R is a symmetric matrix. Since $H \succ 0$ and L is full row rank, it follows that $R \succ 0$. The following result shows that $R \prec I$.

Proposition 1: Consider R in (33). One has $R \prec \mathbf{I}, \forall \rho > 0$.

Proof: By using the Schur complement, we can equivalently rewrite the condition $R \prec \mathbf{I}$ as

$$\begin{bmatrix} \mathbf{I} & \sqrt{\rho}L \\ \sqrt{\rho}L^T & H+\rho Q \end{bmatrix} \succ 0.$$

Using the Schur complement again, the condition is equivalent to

$$H + \rho Q - \rho L^T L \succ 0$$

or $H \succ 0$. The proof is complete. □

Define $\bar{\lambda}, \underline{\lambda}$ as the maximal and the minimal eigenvalues of R, respectively. $\bar{\lambda}, \underline{\lambda}$ are functions of ρ. One has $0 < \underline{\lambda} \leq \bar{\lambda} < 1$. Define

$$\lambda = \max\{1 - \underline{\lambda}, \bar{\lambda}\}. \tag{34}$$

It is clear that $0 < \lambda < 1$. The following result shows that Algorithm 1 converges *linearly* to the fixed point $\forall \rho > 0$.

Proposition 2: Consider Algorithm 1. The following relation holds:

$$\begin{aligned} &||z(k+1) - z^*||_2^2 + ||u(k+1) - u^*||_2^2 \\ &\leq \lambda\left(||z(k) - z^*||_2^2 + ||u(k) - u^*||_2^2\right). \end{aligned} \tag{35}$$

Proof: It is omitted here as we will provide the proof for the general case for any m. □

Proposition 2 states that the sequence $\{(z(k), u(k))\}$ is $Q-$linearly convergent with the rate $\lambda_s = \sqrt{\lambda}$.

We should select the optimal penalty parameter ρ^* that minimizes $\sqrt{\lambda}$, or equivalently λ. Note that minimizing λ is equivalent to minimize the condition number of the matrix R.

Define $W = LH^{-1}L^T$, $W \in \mathbb{R}^{l \times l}$. As L is a full row rank matrix and $H \succ 0$, one has $W \succ 0$. Denote $d_1, d_2, \ldots, d_l$ as the eigenvalues of W with

$$d_1 \geq d_2 \geq \ldots \geq d_l > 0. \tag{36}$$

We have the following result.

Proposition 3: The optimal penalty parameter ρ is given by $\rho^* = \frac{1}{\sqrt{d_1 d_l}}$.

Proof: It is omitted here as we will provide the proof for the general case for any m. □

Remark 2: With $\rho = \rho^*$, we will show in Section IV-A that the corresponding convergence rate is $\lambda_s^* = \sqrt{\frac{\sqrt{\frac{d_1}{d_l}}}{1+\sqrt{\frac{d_1}{d_l}}}}$. Note that $\frac{d_1}{d_l}$ is the condition number of W. Since $d_l \leq d_1$, it follows that $\frac{d_1}{d_l} \geq 1$ and $\lambda_s^* \geq \sqrt{\frac{1}{2}}$.

B. Self-Adaptive Penalty Parameter

In this section, we provide a new iteration scheme to find the solution of (18), (19). Here we are interested in optimizing the penalty parameter ρ. However unlike Section III-A, ρ is a function of k. To distinguish the fixed penalty parameter case and the time-varying parameter case, we employ $\gamma(k)$ instead of $\rho(k)$ in the ADMM scheme. The introduction of the auxiliary variable z in equation (20) involves ρ, which presents difficulty when $\rho(k)$ or equivalently $\gamma(k)$ is time-varying. To overcome this problem, we will use $\tilde{z}$ instead of z in the ADMM with

$$\tilde{z} = Lx + Lb. \tag{37}$$

We rewrite the problem (18), (19) as

$$\min_{x,\tilde{z}} \left\{ \frac{1}{2} x^T H x + f^T x \right\}$$
$$\text{s.t.} \begin{cases} Lx - \tilde{z} + Lb = 0 \\ \tilde{z}^T \tilde{z} \leq 1 \end{cases}. \tag{38}$$

Clearly, problem (18), (19) and (38) have the same optimal solution x^*. At iteration k, we have the following augmented Lagrangian:

$$\mathcal{L}_\gamma(x, \tilde{z}, \tilde{y}) = \frac{1}{2} x^T H x + f^T x + \tilde{y}^T (Lx - \tilde{z} + Lb) + \frac{\gamma(k)}{2} ||Lx - \tilde{z} + Lb||_2^2. \tag{39}$$

Using similar arguments to those to obtain (25), (27), (28), (29), we have the following scheme to update the $\tilde{z}-$, $\tilde{y}-$ updates at iteration $k-1$, and the $x-$, $\tilde{v}-$ updates at iteration k:

$$\tilde{z}(k) = \begin{cases} \tilde{v}(k), & \text{if } \tilde{v}(k)^T \tilde{v}(k) \leq 1 \\ \frac{\tilde{v}(k)}{\sqrt{\tilde{v}(k)^T \tilde{v}(k)}} & \text{otherwise} \end{cases}, \tag{40}$$

$$\tilde{y}(k) = \gamma(k-1) \left(\tilde{v}(k) - \tilde{z}(k) \right), \tag{41}$$

$$x(k+1) = (H + \gamma(k)Q)^{-1} L^T \left(\gamma(k)\tilde{z}(k) - \tilde{y}(k) \right) - (H + \gamma(k)Q)^{-1} (f + \gamma(k)Qb), \tag{42}$$

$$\tilde{v}(k+1) = Lx(k+1) + \frac{1}{\gamma(k)} \tilde{y}(k) + Lb. \tag{43}$$

Remark 3: Using (20) and (37), it is clear that the suboptimization problems (14), (15) can be solved efficiently by considering any auxiliary variable such as $\hat{z} = \eta Lx + \eta Lb$, $\eta \neq 0$. As ADMM is a first order optimization method, it is sensitive to the problem scaling. A way to improve convergence of the ADMM is to select a good η. However, this is beyond the scope of the article and will not be discussed further.

Using (40), one gets

$$\tilde{z}(k) = \alpha(k) \tilde{v}(k) \tag{44}$$

where $0 < \alpha(k) \leq 1$, with

$$\alpha(k) = \begin{cases} 1, & \text{if } \tilde{v}(k)^T \tilde{v}(k) \leq 1 \\ \frac{1}{\sqrt{\tilde{v}(k)^T \tilde{v}(k)}}, & \text{otherwise} \end{cases}.$$

When $\tilde{v}(k)^T \tilde{v}(k)$ is large, $\alpha(k)$ can be arbitrarily close to zero. However, $\alpha(k)$ is always different from zero. Substituting (44) into (41), one obtains

$$\tilde{y}(k) = \gamma(k-1) \left(1 - \alpha(k) \right) \tilde{v}(k). \tag{45}$$

With a slight abuse of notation, denote

$$R(k) = \gamma(k) L \left(H + \gamma(k)Q \right)^{-1} L^T.$$

By substituting (42), (44), (45) into (43), one gets

$$\tilde{v}(k+1) = A(k)\tilde{v}(k) + L(H + \gamma(k)Q)^{-1} (Hb - f) \tag{46}$$

with

$$A(k) = \alpha(k)R(k) + (1 - \alpha(k)) \frac{\gamma(k-1)}{\gamma(k)} \left(\mathbf{I} - R(k) \right). \tag{47}$$

Equation (46) describes a time-varying system, where the iteration counter k represents discrete-time steps, and $\tilde{v}(k)$ is the state. The system matrix $A(k)$ is the convex hull of two vertices, i.e.,

$$A(k) = \alpha(k)A_1(k) + (1 - \alpha(k))A_2(k)$$

with

$$A_1(k) = R(k), A_2(k) = \frac{\gamma(k-1)}{\gamma(k)} \left(\mathbf{I} - R(k) \right).$$

Clearly, the convergence rate of $\tilde{v}(k)$ to the fixed point $\tilde{v}^*$ depends greatly on $A(k)$. For example, $\tilde{v}(k)$ diverges to infinity if $A(k)$ has an eigenvalue outside the unit circle.

Remark 4: $A_1(k) = R$, $A_2(k) = \mathbf{I} - R$ if γ is constant. The results in Section III-A can be interpreted as to select γ to minimize the maximal eigenvalue of A_1 and of A_2. Using (47), one has

$$A(k) = (1 - \alpha(k)) \frac{\gamma(k-1)}{\gamma(k)} I + \left(\alpha(k) - (1 - \alpha(k)) \frac{\gamma(k-1)}{\gamma(k)} \right) R(k). \tag{48}$$

Consider first the case $\alpha(k) < 1$. Hence, $\alpha(k) = \frac{1}{\sqrt{\tilde{v}(k)^T \tilde{v}(k)}}$, and $\tilde{v}(k)^T \tilde{v}(k) > 1$. Similar to Section III-A, a natural idea is to select $\gamma(k+1)$ that minimizes the modulus of the maximal eigenvalue of $A(k)$ at each iteration k, for a given $\alpha(k)$ and $\gamma(k)$. Nevertheless, it is not trivial to extend this method for the case $m \geq 2$. In this article, we select $\gamma(k)$ by using the following equation:

$$\alpha(k) - (1 - \alpha(k)) \frac{\gamma(k-1)}{\gamma(k)} = 0$$

or, equivalently

$$\gamma(k) = \left(\frac{1}{\alpha(k)} - 1 \right) \gamma(k-1) = \left(\sqrt{\tilde{v}(k)^T \tilde{v}(k)} - 1 \right) \gamma(k-1). \tag{49}$$

Note that as $\tilde{v}(k)^T \tilde{v}(k) > 1$, one has $\gamma(k) > 0$ if $\gamma(k-1) > 0$. Substituting (49) into (48), one gets

$$A(k) = \alpha(k)\mathbf{I} = \frac{1}{\sqrt{\tilde{v}(k)^T \tilde{v}(k)}} \mathbf{I}.$$

Thus, using (46)

$$\tilde{v}(k+1) = \frac{\tilde{v}(k)}{\sqrt{\tilde{v}(k)^T \tilde{v}(k)}} + L(H + \gamma(k)Q)^{-1} (Hb - f). \tag{50}$$

Substituting (49) into (42), one obtains

$$x(k+1) = -(H + \gamma(k)Q)^{-1} (f + \gamma(k)Qb) = (H + \gamma(k)Q)^{-1} (Hb - f) - b. \tag{51}$$

Using (50) and (51), one gets

$$\tilde{v}(k+1) = \frac{\tilde{v}(k)}{\sqrt{\tilde{v}(k)^T \tilde{v}(k)}} + L \left(x(k+1) + b \right). \tag{52}$$

At iteration k only $\gamma(k-1), \tilde{v}(k)$ are required in (49), (51), (52) to update $\gamma(k), x(k+1), \tilde{v}(k+1)$. As a result, we can use (49), (51), (52) to find the fixed point $(\gamma^*, x^*, \tilde{v}^*)$ of the original scheme (40), (41), (42), (43).

In the following we provide an alternative way to compute γ^* that does not require $\tilde{v}(k)$. The idea is to use the Gauss-Seidel method [13]. As it will be shown later this method allows us to cover also the case $\alpha(k) = 1$. Using this method, we update $\gamma(k+1)$ in the following two steps.

1) **Step 1**: Given $x(k+1)$, calculate $\tilde{v}(k+1)$ of (52) as a solution of the following nonlinear equation

$$\tilde{v} = \frac{\tilde{v}}{\sqrt{\tilde{v}^T \tilde{v}}} + L\left(x(k+1) + b\right). \quad (53)$$

2) **Step 2**: Update $\gamma(k+1)$ using (49) and $\tilde{v}(k+1)$.

We have the following result.

Proposition 4: A solution $\tilde{v}$ of (53) is given as

$$\tilde{v} = \left(1 + \frac{1}{||x(k+1) + b||_Q}\right) L\left(x(k+1) + b\right). \quad (54)$$

Proof: Using (54), one has

$$\frac{\tilde{v}}{\sqrt{\tilde{v}^T \tilde{v}}} = \frac{L(x(k+1) + b)}{||x(k+1) + b)||_Q}.$$

It follows that:

$$\frac{\tilde{v}}{\sqrt{\tilde{v}^T \tilde{v}}} + L\left(x(k+1) + b\right) = \tilde{v}.$$

The proof is complete. □

Using Proposition 4 after Step 1, one gets

$$\tilde{v}(k+1) = \left(1 + \frac{1}{||x(k+1) + b||_Q}\right) L\left(x(k+1) + b\right). \quad (55)$$

At Step 2, by substituting (55) into (49), one obtains

$$\gamma(k+1) = \sqrt{(x(k+1) + b)^T Q (x(k+1) + b)}\gamma(k). \quad (56)$$

Note that only $\gamma(k)$ is required to update $\gamma(k+1)$ in (56).

The following result holds.

Theorem 1: The iteration scheme (51), (56) converges to the optimal solution x^* of (18), (19) for any $\gamma(0) > 0$.

Proof: See Appendix A. □

Remark 5: As the scheme (51), (56) has a root in the ADMM, it is possible to prove the convergence of (51), (56) to the optimal solution via the ADMM theory. The main advantage of the proof of Theorem 1 in the article is that it shows that γ can be selected as the optimal dual variable for the original problem (18), (19). Note that unlike Algorithm 1, γ in (51) and (56) is allowed to be zero. If $\gamma^* = 0$, then the constraint (19) is inactive. Otherwise, (19) is active.

The following result holds.

Proposition 5: Using (51) and (56), $\gamma(k)$ converges to the fixed point γ^* monotonically for any $\gamma(0) > 0$.

Proof: See Appendix B. □

By substituting (51) into (56), one obtains a nonlinear iteration scheme with only γ as variable. It is well known [2] that the convergence rate is linear for any nonlinear iteration scheme, if it converges. The convergence rate depends on $\frac{dh(\cdot)}{d\gamma}$, with $h = \sqrt{(x(\gamma) + b)^T Q (x(\gamma) + b)}\gamma$, evaluated at the fixed point $\gamma = \gamma^*$. The parameter γ converges quickly to γ^* when γ is far from γ^*. The speed is slower when γ is close to γ^*. This behavior is very typical for first-order optimization methods.

In the following we will analyze the behavior of the iteration scheme (40), (41), (42), (43) if we use $\gamma(k) = \gamma^*, \forall k$. Clearly, as the constraints (19) is inactive if $\gamma^* = 0$, one needs to analyze only the case $\gamma^* > 0$. It can be verified that the scheme (40), (41), (42), (43) converges in one step with $\tilde{z}(0) = L(x^* + b), \tilde{y}(0) = \gamma^* L(x^* + b)$. This analysis suggests that γ^* is the best choice for (40), (41), (42), and (43) if $\gamma(k) = \gamma^*, \forall k \geq 0$. However, γ^* is unknown. Our idea is to use the scheme (51), (56) to obtain a very rough approximation of γ^*. Because $\gamma(k)$ converges quickly to γ^* when $\gamma(k)$ is far from γ^*, we need only to make a few iterations, $k_{\max}$, for (51) and (56). We then use $\gamma(k) = \gamma(k_{\max}), \forall k \geq k_{\max}$ in (40), (41), (42), and (43).

We will now analyze the behavior of (40), (41), (42), and (43) with $\gamma(k) = \gamma(k_{\max}), \forall k \geq k_{\max}$ near the fixed point $(x^*, \tilde{v}^*, \tilde{z}^*, \tilde{y}^*)$, where $\gamma(k_{\max})$ is an approximation of γ^*. Denote $\tilde{\gamma}^* = \gamma(k_{\max})$, $\tilde{u} = \tilde{\gamma}^* \tilde{y}$. Using (42), one has

$$\begin{aligned} x(k+1) &= \tilde{\gamma}^*(H + \tilde{\gamma}^* Q)^{-1} L^T \left(\tilde{z}(k) - \tilde{u}(k)\right) \\ &\quad - \left(H + \tilde{\gamma}^* Q\right)^{-1} \left(f + \tilde{\gamma}^* Qb\right) \\ &= \tilde{\gamma}^*(H + \tilde{\gamma}^* Q)^{-1} L^T \left(\tilde{z}(k) - \tilde{u}(k)\right) + \tilde{x}^* \end{aligned}$$

where $\tilde{x}^* = -(H + \tilde{\gamma}^* Q)^{-1}(f + \tilde{\gamma}^* Qb) \approx x^*$. Hence one should have $\tilde{z}(k) \approx \tilde{u}(k)$ at convergence. Using (41), one gets $\tilde{z}(k) \approx \tilde{u}(k) \approx \frac{1}{2}\tilde{v}(k)$. Using (43), one obtains

$$\begin{aligned} \tilde{v}(k+1) &= Lx(k+1) + \tilde{u}(k) + Lb \\ &\approx \frac{1}{2}\tilde{v}(k) + L(x^* + b). \end{aligned}$$

It follows that $\tilde{v}(k)$ converges to $\tilde{v}^*$ with the convergence rate of $\frac{1}{2}$ near the fixed point. Note that at the fixed point one should have $\tilde{v}^* = 2L(x^* + b)$. Hence, $\tilde{z}^* = L(x^* + b)$, and $\tilde{u}^* = L(x^* + b)$.

Using similar arguments to those to obtain (30), (31), it can be shown that if γ is fixed, then the primal and dual residuals of the ADMM scheme (40), (41), (42), (43) are

$$\begin{cases} r_p(k) = \tilde{u}(k) - \tilde{u}(k-1) \\ r_d(k) = -\gamma L^T(\tilde{z}(k) - \tilde{z}(k-1)) \end{cases}. \quad (57)$$

We summarize our development in Algorithm 2.

Algorithm 2: Time-Varying Penalty Parameter - Case $m = 1$.

Require: $H, f, Q, b, L,$
$\gamma(0) > 0, k_{\max} > 0, \epsilon_d > 0, \epsilon_p > 0$
1: $k \leftarrow 0$
2: **repeat**
3: $x(k+1) \leftarrow -(H + \gamma(k)Q)^{-1}(f + \gamma(k)Qb)$
4: $\gamma(k+1) \leftarrow \sqrt{(x(k+1) + b)^T Q (x(k+1) + b)}\gamma(k)$
5: $k \leftarrow k + 1$
6: **until** $k = k_{\max}$
7: $\gamma \leftarrow \gamma(k_{\max})$, $\tilde{z}(k) \leftarrow L(x(k) + b), \tilde{u}(k) \leftarrow L(x(k) + b)$
8: **repeat**
9: $x(k+1) \leftarrow \gamma(H + \gamma Q)^{-1} L^T(\tilde{z}(k) - \tilde{u}(k))$
10: $-(H + \rho Q)^{-1}(f + \rho Qb)$
11: $\tilde{v}(k+1) \leftarrow Lx(k+1) + \tilde{u}(k) + Lb$
12: **if** $\tilde{v}(k+1)^T \tilde{v}(k+1) \leq 1$ **then**
13: $\tilde{z}(k+1) \leftarrow \tilde{v}(k+1)$
14: **else**
15: $\tilde{z}(k+1) \leftarrow \frac{\tilde{v}(k+1)}{\sqrt{\tilde{v}(k+1)^T \tilde{v}(k+1)}}$
16: $\tilde{u}(k+1) \leftarrow \tilde{v}(k+1) - \tilde{z}(k+1)$
17: $r_p(k+1) \leftarrow \tilde{u}(k+1) - \tilde{u}(k)$
18: $r_d(k+1) \leftarrow -\gamma L^T(\tilde{z}(k+1) - \tilde{z}(k))$
19: $k \leftarrow k + 1$
20: **until** $||r_p(k)||_\infty \leq \epsilon_p$ and $||r_d(k)||_\infty \leq \epsilon_d$

Step 6 implies that we update $\gamma(k)$ only for a few number of iterations $k_{\max}$. Note that other criteria can be used, e.g., we stop to update $\gamma(k)$ when the speed of change or progress in $\gamma(k)$ is less than some tolerance value.

Results of numerical experiment given in Section VI demonstrate that Algorithm 2 reduces substantially the number of iterations compared to Algorithm 1 with $\rho = \rho^*$. The drawback of Algorithm 2 is step 3, i.e., one needs to inverse $(H + \gamma(k)Q)$ in each iteration. In the following we provide a way to reduce this increased computational cost. Define $L_h \in \mathbb{R}^n, V_h \in \mathbb{R}^n, D_h \in \mathbb{R}^n$ such that $L_h^T L_h = H, V_h^T V_h = \mathbf{I}$, D_h is a diagonal matrix, and

$$V_h^T D_h V_h = (L_h^T)^{-1} Q L_h^{-1}.$$

One has

$$\begin{aligned}(H + \gamma(k)Q)^{-1} &= L_h^{-1}(\mathbf{I} + \gamma(k)V_h^T D V_h)^{-1}(L_h^T)^{-1} \\ &= L_h^{-1} V_h (I + \gamma(k)D)^{-1} V_h^T (L_h^T)^{-1}.\end{aligned}$$

The matrix $\mathbf{I} + \gamma(k)D$ is diagonal, its inverse can be calculated analytically for any $\gamma(k) \geq 0$.

C. Connection to Earlier Works

In this section we aim to show the connection between Algorithm 2 and the residual balancing (RB) method [4], [15], [33]. The RB method is based on the observation that increasing γ strengthens the penalty term in the augmented Lagrangian (82), yielding smaller primal residual r_p but larger dual one r_d. Conversely, decreasing γ leads to larger r_p and smaller r_d. Since both r_p and r_d must be small at convergence, it is reasonable to balance them, i.e., tune γ to keep both r_p and r_d of similar magnitude. At iteration k, a simple way for this goal is

$$\gamma(k) = \begin{cases} \eta\gamma(k-1), & \text{if } ||r_p(k)|| \geq \tau||r_d(k)|| \\ \frac{\gamma(k-1)}{\eta}, & \text{if } ||r_d(k)|| \geq \tau||r_p(k)|| \\ \gamma(k-1), & \text{otherwise} \end{cases} \tag{58}$$

where $\eta > 1$ and $\tau > 0$. In [15], η can also be time-varying. The adaptation (58) is generally turning OFF after a few number of iterations.

The scheme (58) has been found to be effective for a number of problems [16], [18], [29]. As far as we know, there is no systematic method to select η, τ. The parameters η, τ are chosen by the trial and error method. It is well known [33] that this is not a trivial problem. The performance of the RB method varies wildly with different problem scalings.

Recall that the main purpose of optimizing γ in Algorithm 2 is to make $\tilde{z}(k) \approx \tilde{u}(k)$. Hence $\tilde{z}(k+1) - \tilde{z}(k) \approx \tilde{u}(k+1) - \tilde{u}(k)$, or equivalently, r_p and r_d have the same order of magnitude. Consequently, Algorithm 2 can be considered as a new adaptive RB method. The main distinguished feature of Algorithm 2 compared to the standard RB method is that the parameter η is automatically and dynamically updated. This new "feedback" in Algorithm 2 makes it more robust to bad scaling in the data.

IV. General Case

A. Optimal Fixed Penalty Parameter

Consider the optimization problem (1), (2). Define $L_i \in \mathbb{R}^{l_i \times n}$ such that $L_i^T L_i = Q_i$, and l_i is the rank of $Q_i, \forall i = \overline{1,m}$. Define also $z_i \in \mathbb{R}^{l_i}$ and $\mathbf{z} \in \mathbb{R}^{l_1 + \ldots + l_m}$ as

$$\begin{cases} \frac{z_i}{\sqrt{\rho_i}} = L_i x + L_i b_i, \forall i = \overline{1,m}, \\ \mathbf{z} = \begin{bmatrix} z_1^T & z_2^T & \ldots & z_m^T \end{bmatrix}^T \end{cases} \tag{59}$$

where $\rho_i > 0$ are penalty parameters in the ADMM. ρ_i can be different for different index $i, \forall i = \overline{1,m}$

We rewrite the problem (1), (2) as

$$\begin{aligned} &\min_{x,\mathbf{z}} \left\{ \frac{1}{2} x^T H x + f^T x \right\} \\ &\text{s.t.} \begin{cases} L_i x + L_i b_i = \frac{z_i}{\sqrt{\rho_i}}, \forall i = \overline{1,m}, \\ z_i^T z_i \leq \rho_i, \forall i = \overline{1,m}. \end{cases} \end{aligned} \tag{60}$$

Clearly, problem (1), (2) and problem (60) have the same optimal solution x^*. The augmented Lagrangian for (60) is

$$\begin{aligned} &\mathcal{L}_\rho(x, \mathbf{z}, \mathbf{y}) \\ &= \frac{1}{2} x^T H x + f^T x + \sum_{i=1}^m y_i^T \left(L_i x + L_i b_i - \frac{z_i}{\sqrt{\rho_i}} \right) \\ &\quad + \sum_{i=1}^m \frac{\rho_i}{2} ||L_i x + L_i b_i - \frac{z_i}{\sqrt{\rho_i}}||_2^2 \end{aligned} \tag{61}$$

where $y_i \in \mathbb{R}^{l_i}, \forall i = \overline{1,m}$ are the dual variables for the equality constraints.

Recall $L_i^T L_i = Q_i, \forall i = \overline{1,m}$. Using (14) at iteration k, the $x-$update is the solution of the following quadratic program:

$$\begin{aligned} \min_x \Bigg\{ &\frac{1}{2} x^T (H + \sum_{i=1}^m \rho_i Q_i) x \\ &- \left(\sum_{i=1}^m L_i^T (\sqrt{\rho_i} z_i(k) - y_i(k)) - f - \sum_{i=1}^m \rho_i Q_i b_i \right)^T x \Bigg\}. \end{aligned} \tag{62}$$

The solution to (62) can be computed analytically as

$$\begin{aligned} &x(k+1) \\ &= (H + \sum_{i=1}^m \rho_i Q_i)^{-1} \left(\sum_{i=1}^m L_i^T (\sqrt{\rho_i} z_i(k) - y_i(k)) \right. \\ &\qquad \left. - f - \sum_{i=1}^m \rho_i Q_i b_i \right) \\ &= (H + \sum_{i=1}^m \rho_i Q_i)^{-1} \sum_{i=1}^m L_i^T (\sqrt{\rho_i} z_i(k) - y_i(k)) \\ &\quad - \left(H + \sum_{i=1}^m \rho_i Q_i \right)^{-1} \left(f + \sum_{i=1}^m \rho_i Q_i b_i \right). \end{aligned} \tag{63}$$

Using (15), the $\mathbf{z}-$update at iteration k is the solution of the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{z}} \left\{ \sum_{i=1}^m -y_i(k)^T \frac{z_i}{\sqrt{\rho_i}} \right. \\ &\qquad \left. + \left(z_i^T z_i - 2\sqrt{\rho_i} \left(L_i x(k+1) + L_i b_i \right)^T z_i \right) \right\} \\ &\text{s.t. } z_i^T z_i \leq \rho_i, \forall i = \overline{1,m}. \end{aligned} \tag{64}$$

The cost function and the constraints of (64) are separable in z_i. The updates of z_i can be carried out in parallel.

The update of z_i is the solution of the optimization problem, $\forall i = \overline{1,m}$

$$\min_{z_i} \left\{ z_i^T z_i - 2 v_i(k+1)^T z_i \right\}$$
$$\text{s.t. } z_i^T z_i \leq \rho_i \tag{65}$$

where $\forall i = \overline{1,m}$

$$v_i(k+1) = \sqrt{\rho_i} L_i x(k+1) + \frac{y_i(k)}{\sqrt{\rho_i}} + \sqrt{\rho_i} L_i b_i. \tag{66}$$

Using Lemma 1, the solution $z_i(k+1)$ to the problem (65) is given by, $\forall i = \overline{1,m}$

$$z_i(k+1) = \begin{cases} v_i(k+1), & \text{if } v_i(k+1)^T v_i(k+1) \leq \rho_i \\ \frac{\sqrt{\rho_i} v_i(k+1)}{\sqrt{v_i(k+1)^T v_i(k+1)}}, & \text{otherwise.} \end{cases} \tag{67}$$

Finally using (16), the update of y_i is given by, $\forall i = \overline{1,m}$

$$y_i(k+1) = y_i(k) + \rho_i \left(L_i x(k+1) + L_i b_i - \frac{z_i(k+1)}{\sqrt{\rho_i}} \right)$$
$$= \sqrt{\rho_i} \left(\sqrt{\rho_i} L_i x(k+1) + \frac{y_i(k)}{\sqrt{\rho_i}} + \sqrt{\rho_i} L_i b_i - z_i(k+1) \right)$$

thus, using (66)

$$y_i(k+1) = \sqrt{\rho_i} \left(v_i(k+1) - z_i(k+1) \right), \forall i = \overline{1,m}. \tag{68}$$

Using similar arguments to those to obtain (30), (31), it can be shown that the primal and dual residuals are given as

$$\begin{cases} r_p(k) = [r_{p,1}(k)^T \ \ldots \ r_{p,m}(k)^T]^T \\ r_d(k) = [r_{d,1}(k)^T \ \ldots \ r_{d,m}(k)^T]^T \end{cases} \tag{69}$$

with

$$\begin{cases} r_{p,i}(k) = \frac{1}{\sqrt{\rho_i}} (u_i(k) - u_i(k-1)) \\ r_{d,i}(k) = -\sqrt{\rho_i} L^T (z_i(k) - z_i(k-1)) \end{cases}$$

where $u_i(k) = \frac{y_i(k)}{\sqrt{\rho_i}}, \forall i = \overline{1,m}$.

Algorithm 3 shows the particularization of the ADMM algorithm (14), (15), (16) to the problem (1), (2).

As the problem (1), (2) is strictly convex, for any $\rho_i > 0, \forall i = \overline{1,m}$, Algorithm 3 converges to the fixed point (x^*, z^*, u^*), where x^* is the optimal solution of (1), (2). In the following, we aim to characterize the rate of convergence of Algorithm 3. Define

$$\mathbf{u} = [u_1^T \ u_2^T \ \ldots \ u_m^T]^T, \mathbf{L}_r = \begin{bmatrix} \sqrt{\rho_1} L_1 \\ \vdots \\ \sqrt{\rho_m} L_m \end{bmatrix}$$

$$H_q = H + \sum_{i=1}^m \rho_i Q_i, \mathbf{R} = \mathbf{L}_r H_q^{-1} \mathbf{L}_r^T$$

$$\mathbf{b} = \begin{bmatrix} \sqrt{\rho_1} L_1 b_1 \\ \vdots \\ \sqrt{\rho_m} L_m b_m \end{bmatrix} - \mathbf{L}_r H_q^{-1} \left(f + \sum_{i=1}^m \rho_i Q_i b_i \right). \tag{70}$$

In this section, we assume that $\mathbf{L}$ is a full row rank matrix. In this case $\mathbf{R} \succ 0, \forall \rho_i > 0, \forall i = \overline{1,m}$. Using similar arguments to those in the proof of Proposition 1, it can be shown that $\mathbf{R} \prec \mathbf{I}$. Define $\bar{\lambda}, \underline{\lambda}$, respectively, as the maximal and the minimal eigenvalues of $\mathbf{R}$. One has $0 < \underline{\lambda} \leq \bar{\lambda} < 1$. Define

$$\lambda = \max\{1 - \underline{\lambda}, \bar{\lambda}\}. \tag{71}$$

Algorithm 3: Fixed Penalty Parameter - General Case.

Require: $H, f, Q_i, b_i, L_i,$
$z_i(0), u_i(0), \rho_i > 0, \forall i = \overline{1,m}, \epsilon_p > 0, \epsilon_d > 0$

$k \leftarrow 0$
repeat
  $x(k+1) \leftarrow$
$$\left(H + \sum_{i=1}^m \sqrt{\rho_i} Q_i \right)^{-1} \sum_{i=1}^m \rho_i L_i^T (z_i(k) - u_i(k))$$
$$- \left(H + \sum_{i=1}^m \rho_i Q_i \right)^{-1} \left(f + \sum_{i=1}^m \rho_i Q_i b_i \right)$$
  for $i \leftarrow 1$ to m **do**
    $v_i(k+1) \leftarrow \sqrt{\rho_i} L_i x(k+1) + u_i(k) + \sqrt{\rho_i} L_i b_i$
    if $v_i(k+1)^T v_i(k+1) \leq \rho_i$ **then**
      $z_i(k+1) \leftarrow v_i(k+1)$
    else
      $z_i(k+1) \leftarrow \frac{\sqrt{\rho_i} v_i(k+1)}{\sqrt{v_i(k+1)^T v_i(k+1)}}$
    $u_i(k+1) \leftarrow v_i(k+1) - z_i(k+1)$
    $r_{p,i}(k+1) \leftarrow \frac{1}{\sqrt{\rho_i}} (u_i(k+1) - u_i(k))$
    $r_{d,i}(k+1) \leftarrow -\sqrt{\rho_i} L_i^T (z_i(k+1) - z_i(k))$
  $k \leftarrow k+1$
  $r_p(k) \leftarrow [r_{p,1}(k)^T \ r_{p,2}(k)^T \ \ldots r_{p,m}(k)^T]$
  $r_d(k) \leftarrow [r_{d,1}(k)^T \ r_{d,2}(k)^T \ \ldots r_{d,m}(k)^T]$
until $||r_p(k)||_\infty \leq \epsilon_p$ and $||r_d(k)||_\infty \leq \epsilon_d$

It is clear that $0 < \lambda < 1$. Define also $\mathbf{v} = [v_1^T \ v_2^T \ \ldots \ v_m^T]^T$. Using (63), (66), (70), one obtains

$$\mathbf{v}(k+1) = \mathbf{R}\mathbf{z}(k) + (\mathbf{I} - \mathbf{R})\mathbf{u}(k) + \mathbf{b}. \tag{72}$$

We have the following result.

Theorem 2: Consider Algorithm 3. The following relation holds:

$$||\mathbf{z}(k+1)^T - \mathbf{z}^*||_2^2 + ||\mathbf{u}(k+1)^T - \mathbf{u}^*||_2^2$$
$$\leq \lambda \left(||\mathbf{z}(k)^T - \mathbf{z}^*||_2^2 + ||\mathbf{u}(k)^T - \mathbf{u}^*||_2^2 \right). \tag{73}$$

Proof: See Appendix C. □

Remark 6: To the best of the author's knowledge, this is the first time the linear convergence rate of the ADMM with different penalty parameters is established.

Remark 7: Consider the case where $Q_i, \forall i = \overline{1,m}$ are given in (5). In this case, the QCQP problem (1), (2) is a QP problem. Theorem 2 states that the residual sequence $\{(z(k), u(k))\}$ is Q-linear convergent. Using Proposition 2 in [12], it is only possible to establish that $\{(z(k), u(k))\}$ is $R-$linearly convergent. Note also that the result in [12] is not applicable for the constraints (5), that have a lower and upper bounds. Hence, Corollary 1 provides an improvement of the result in [12] for the QP case.

We should select $\rho_i, \forall i = \overline{1,m}$ that minimize λ. This is equivalent to minimize the condition number of $\mathbf{R}$. However, the problem is non-convex. Fortunately, for a particular case where $\rho_i = \rho, \forall i = \overline{1,m}$ we can convexify the problem as in Section III-A.

Define $\mathbf{W} = \mathbf{L}H^{-1}\mathbf{L}^T$, where $\mathbf{L} = \begin{bmatrix} L_1 \\ \vdots \\ L_m \end{bmatrix} \in \mathbb{R}^{\mathbf{l}\times\mathbf{l}}$, with $\mathbf{l} = l_1 + \ldots + l_m$. One has $\mathbf{W} \succ 0$. Denote $\mathbf{d}_1, \ldots, \mathbf{d}_\mathbf{l}$ as the eigenvalues of $\mathbf{W}$ with

$$\mathbf{d}_1 \geq \mathbf{d}_2 \geq \ldots \geq \mathbf{d}_\mathbf{l} > 0.$$

We have the following result.

Theorem 3: The optimal penalty parameter ρ is given by

$$\rho^* = \frac{1}{\sqrt{\mathbf{d}_1 \mathbf{d}_\mathbf{l}}}. \quad (74)$$

Proof: See Appendix D. □

Remark 8: As shown in (3), (4), and (5), Algorithm 3 can cope with double-sided linear constraints. It can also cope with linear constraints such as

$$\bar{L}x \leq \bar{u} \quad (75)$$

where $\bar{L} \in \mathbb{R}^{1\times n}, \bar{u} \in \mathbb{R}$. For simplicity, we consider only one linear constraint, but the approach can be straightforwardly extended to any number of linear constraints.

One way to apply Algorithm 3 is to recast (75) into the form (3) as

$$-M \leq \bar{L}x \leq \bar{u}$$

where $M > 0$ is a large number. In this case the constraint $-M \leq \bar{L}x$ inactive. An alternative way is to introduce directly an auxiliary variable $\bar{z}$ for the constraint (75) as

$$\frac{\bar{z}}{\sqrt{\bar{\rho}}} = \bar{L}x.$$

where $\bar{\rho} > 0$ is the penalty variable. We rewrite equivalently the problem (1), (2), (75) as

$$\min_{x,\mathbf{z}} \left\{ \frac{1}{2}x^T H x + f^T x \right\}$$

$$\text{s.t.} \begin{cases} L_i x + L_i b_i = \frac{z_i}{\sqrt{\rho_i}}, \forall i = \overline{1,m} \\ \bar{L}x = \frac{\bar{z}}{\sqrt{\bar{\rho}}} \\ z_i^T z_i \leq \rho_i, \forall i = \overline{1,m} \\ \bar{z} \leq \sqrt{\bar{\rho}}\bar{u} \end{cases}. \quad (76)$$

The augmented Lagrangian for (76) takes the form

$$\mathcal{L}_{\rho,\bar{\rho}}(x, \mathbf{z}, \bar{z}, \mathbf{y}, \bar{y}) = \frac{1}{2}x^T H x + f^T x + \sum_{i=1}^m y_i^T (L_i x + L_i b_i - \frac{z_i}{\sqrt{\rho_i}}) + \bar{y}\left(\bar{L}x - \frac{\bar{z}}{\sqrt{\bar{\rho}}}\right) + \sum_{i=1}^m \frac{\rho_i}{2} ||L_i x + L_i b_i - \frac{z_i}{\sqrt{\rho_i}}||_2^2 + \frac{\bar{\rho}}{2}\left(\bar{L}x - \frac{z}{\sqrt{\bar{\rho}}}\right)^2. \quad (77)$$

Using (77), it can be shown that the $x-$update is

$$x(k+1) = (H + \bar{\rho}\bar{L}^T\bar{L} + \sum_{i=1}^m \rho_i Q_i)^{-1} \left(\bar{L}^T(\sqrt{\bar{\rho}}\bar{z}(k) - \bar{y}(k)) + \sum_{i=1}^m L_i^T \left(\sqrt{\rho_i} z_i(k) - y_i(k)\right) - f - \sum_{i=1}^m \rho_i Q_i b_i \right). \quad (78)$$

The v_i-, z_i-, and y_i-updates, $\forall i = \overline{1,m}$ are, respectively, given in (66), (67), and (68) with $x(k+1)$ in (78). The $\bar{z}-$ and $\bar{y}-$updates are

$$\begin{cases} \bar{z}(k+1) = \min\left(\bar{v}(k+1), \sqrt{\bar{\rho}}\bar{u}\right) \\ \bar{y}(k+1) = \bar{y}(k+1) + \bar{\rho}\left(\bar{L}x(k+1) - \frac{z(k+1)}{\sqrt{\bar{\rho}}}\right) \end{cases} \quad (79)$$

where $\bar{v}(k+1) = \sqrt{\bar{\rho}}\bar{L}x(k+1) + \frac{\bar{y}(k)}{\sqrt{\bar{\rho}}}$.

It can be verified that the convergence results of Corollaries 1 and 2 are valid for the scheme (78), (66), (67), (68), (79). Following Remark 8, it is clear that we can treat the double-sided inequality constraints (3) explicitly in linear form without converting them into quadratic form.

Remark 9: For the QP problem, the result in Corollary 2 is the same as in Theorem 4 of [12]. Hence for the QP problem, this article provides an alternative proof to show that one has the optimal convergence rate with ρ^* given in (74). Similar to the proof of Theorem 2, it can be shown that the convergence rate is $\lambda_s^* = \sqrt{\frac{\sqrt{\frac{d_1}{d_\mathbf{l}}}}{1+\sqrt{\frac{d_1}{d_\mathbf{l}}}}}$. Since $d_1 \geq d_\mathbf{l}$, one has $\lambda_s^* \geq \sqrt{\frac{1}{2}}$.

Remark 10: If $\mathbf{L}$ is not a full row rank matrix, following [12], a good heuristic choice for ρ is $\rho = \frac{1}{\sqrt{\mathbf{d}_1\mathbf{d}_\mathbf{l}}}$, where $\mathbf{d}_\mathbf{l}$ is the smallest nonzero eigenvalue of $\mathbf{W}$.

B. Self-Adaptive Penalty Parameters

In this section we consider the case when we have a vector of time-varying penalty parameters $\rho_1(k), \rho_2(k), \ldots, \rho_m(k)$. As in Section III-B, to distinguish the fixed penalty parameters case and the time-varying parameters case, we will employ $\gamma_i(k)$ instead of $\rho_i(k)$, $\forall i = \overline{1,m}$. In addition, the introduction of the auxiliaries variables z_i in (59) involves ρ_i, which presents difficulty when $\rho_i(k)$ or equivalently $\gamma_i(k)$ are time-varying. To overcome this problem, we define

$$\begin{cases} \tilde{z}_i = L_i x + L_i b_i, \forall i = \overline{1,m} \\ \tilde{\mathbf{z}} = [\tilde{z}_1^T \; \tilde{z}_2^T \; \ldots \; \tilde{z}_m^T]^T \end{cases}. \quad (80)$$

We rewrite equivalently the problem (1), (2) as

$$\min_{x,\tilde{\mathbf{z}}} \left\{ \frac{1}{2}x^T H x + f^T x \right\},$$

$$\text{s.t.} \begin{cases} L_i x + L_i b_i = \tilde{z}_i, \forall i = \overline{1,m} \\ \tilde{z}_i^T \tilde{z}_i \leq 1, \forall i = \overline{1,m} \end{cases}. \quad (81)$$

We form the following augmented Lagrangian at iteration k:

$$\mathcal{L}_\gamma(x, \tilde{\mathbf{z}}, \tilde{\mathbf{y}}) = \frac{1}{2}x^T H x + f^T x + \sum_{i=1}^m \tilde{y}_i^T (L_i x - z_i + L_i b_i) + \sum_{i=1}^m \frac{\gamma_i(k)}{2} ||L_i x - z_i + L_i b_i||_2^2 \quad (82)$$

where $\tilde{\mathbf{y}} = [\tilde{y}_1^T \; \tilde{y}_2^T \; \ldots \; \tilde{y}_m^T]^T$.

Using similar arguments as in Section IV-A, it can be shown that the $\tilde{z}_i-$, $\tilde{y}_i-$updates at iteration $k-1$, and the $x-$, $\tilde{v}_i-$updates at iteration k are given by, $\forall i = \overline{1,m}$

$$\tilde{z}_i(k) = \begin{cases} \tilde{v}_i(k), & \text{if } \tilde{v}_i(k)^T\tilde{v}_i(k) \leq 1 \\ \frac{\tilde{v}_i(k)}{\sqrt{\tilde{v}_i(k)^T\tilde{v}_i(k)}} & \text{otherwise} \end{cases}, \quad (83)$$

$$\tilde{y}_i(k) = \gamma_i(k-1)\left(\tilde{v}_i(k) - \tilde{z}_i(k)\right), \quad (84)$$

$$x(k+1)$$

$$= \left(H + \sum_{i=1}^{m} \gamma_i(k) Q_i\right)^{-1} \sum_{i=1}^{m} \left(\gamma_i(k) L_i^T \tilde{z}_i(k) - L_i^T \tilde{y}_i(k)\right)$$
$$- \left(H + \sum_{i=1}^{m} \gamma_i(k) Q_i\right)^{-1} \left(f + \sum_{i=1}^{m} \gamma_i(k) Q_i b_i\right), \quad (85)$$

$$\tilde{v}_i(k+1) = L_i x(k+1) + \frac{1}{\gamma_i(k)} \tilde{y}_i(k) + L_i b_i. \quad (86)$$

Using (83), one obtains, $\forall i = \overline{1,m}$

$$\tilde{z}_i(k) = \alpha_i(k) \tilde{v}_i(k) \quad (87)$$

where $0 < \alpha_i(k) \leq 1, \forall i = \overline{1,m}$ with

$$\alpha_i(k) = \begin{cases} 1, & \text{if } \tilde{v}_i(k)^T \tilde{v}_i(k) \leq 1 \\ \frac{1}{\sqrt{\tilde{v}_i(k)^T \tilde{v}_i(k)}}, & \text{otherwise} \end{cases}. \quad (88)$$

Substituting (88) into (84), one gets, $\forall i = \overline{1,m}$

$$\tilde{y}_i(k) = \gamma_i(k-1)(1 - \alpha_i(k)) \tilde{v}_i(k). \quad (89)$$

By substituting (85), (87), (89) into (86), one obtains, $\forall i = \overline{1,m}$

$$\tilde{v}_i(k+1)$$
$$= \sum_{j=1}^{m} \left(\alpha_j(k) - \frac{\gamma_j(k-1)}{\gamma_j(k)}(1 - \alpha_j(k))\right) R_{ij}(k) \tilde{v}_j(k)$$
$$+ \frac{\gamma_i(k-1)}{\gamma_i(k)}(1 - \alpha_i(k)) \tilde{v}_i(k) + L_i b_i$$
$$- L_i(H + \sum_{j=1}^{m} \gamma_j(k) Q_j)^{-1}(f + \sum_{j=1}^{m} \gamma_j(k) Q_j b_j), \quad (90)$$

where $\forall i, j = \overline{1,m}$

$$R_{ij}(k) = \gamma_j(k) L_i (H + \sum_{i=1}^{m} \gamma_i(k) Q_i)^{-1} L_j^T.$$

Similar to Section III-B, we consider first the case $\alpha_i(k) < 1, \forall i = \overline{1,m}$. In this case $\alpha_i(k) = \frac{1}{\sqrt{\tilde{v}_i(k)^T \tilde{v}_i(k)}}$. We select the parameters $\gamma_i(k)$ to satisfy the following equation, $\forall i = \overline{1,m}$

$$\alpha_i(k) - \frac{\gamma_i(k-1)}{\gamma_i(k)}(1 - \alpha_i(k)) = 0.$$

As a result, $\forall i = \overline{1,m}$

$$\gamma_i(k) = \frac{1 - \alpha_i(k)}{\alpha_i(k)} \gamma_i(k-1)$$
$$= \left(\sqrt{\tilde{v}_i(k)^T \tilde{v}_i(k)} - 1\right) \gamma_i(k-1). \quad (91)$$

By substituting (91) into (85), one gets

$$x(k+1) = -\left(H + \sum_{j=1}^{m} \gamma_j(k) Q_j\right)^{-1} \left(f + \sum_{j=1}^{m} \gamma_j(k) Q_j b_j\right). \quad (92)$$

By substituting (91), (92) into (90), one obtains, $\forall i = \overline{1,m}$

$$\tilde{v}_i(k+1) = \frac{\tilde{v}_i(k)}{\sqrt{\tilde{v}_i(k)^T \tilde{v}_i(k)}} + L_i \left(x(k+1) + b_i\right). \quad (93)$$

With a slight abuse of notation, denote

$$\gamma(k) = [\gamma_1(k)\ \gamma_2(k)\ \ldots\ \gamma_m(k)]^T.$$

Similar to Section III-B, we use the Gauss-Seidel method to find the fixed point $(\gamma^*, x^*, \mathbf{v}^*)$ of the scheme (91), (92), (93). The main advantage of this method is that it also covers the case $\alpha_i(k) = 1$. Using the Gauss-Seidel method, $\tilde{v}_i(k+1)$ is updated as a solution of the following equation, $\forall i = \overline{1,m}$

$$\tilde{v}_i = \frac{\tilde{v}_i}{\sqrt{\tilde{v}_i^T \tilde{v}_i}} + L_i(x(k+1) + b_i).$$

Using Proposition 4, one obtains, $\forall i = \overline{1,m}$

$$\tilde{v}_i(k+1)$$
$$= \left(1 + \frac{1}{\sqrt{(x(k+1) + b_i)^T Q_i (x(k+1) + b_i)}}\right)$$
$$\times L_i \left(x(k+1) + b_i\right). \quad (94)$$

By substituting (94) into (91), one gets, $\forall i = \overline{1,m}$

$$\gamma_i(k+1) = \sqrt{(x(k+1) + b_i)^T Q_i (x(k+1) + b_i)} \gamma_i(k). \quad (95)$$

The iteration scheme (92), (95) requires only $\gamma_i(k), \forall i = \overline{1,m}$ to update $x(k+1), \gamma_i(k+1)$. Using similar arguments as the ones in Theorem 3, it can be shown that the scheme (92), (95) converges linearly to the fixed point (x^*, γ^*). The speed of convergence is fast when $\gamma(k)$ is far from γ^*. It becomes slower when $\gamma(k)$ and γ^* are close. Hence, similar to Section III-B, we use the scheme (92), (95) with a few number of iterations, $k_{\max}$, to obtain a rough approximation of γ^*. We then use $\gamma(k) = \gamma(k_{\max}), \forall k \geq k_{\max}$ as the penalty parameters in (83), (84), (85), and (86). We denote this two stage process by Algorithm 4.

Using similar arguments to those to obtain (30), (31), it can be shown that the primal and dual residuals are given as, for a fixed γ

$$\begin{cases} r_p(k) = [r_{p,1}(k)^T\ \ldots\ r_{p,m}(k)^T]^T \\ r_d(k) = [r_{d,1}(k)^T\ \ldots\ r_{d,m}(k)^T]^T \end{cases}, \quad (96)$$

with

$$\begin{cases} r_{p,i}(k) = \tilde{u}_i(k) - \tilde{u}_i(k-1) \\ r_{d,i}(k) = -\gamma_i L_i^T(\tilde{z}_i(k) - \tilde{z}_i(k-1)) \end{cases}$$

where $\tilde{u}_i(k) = \frac{\tilde{y}_i(k)}{\gamma_i}, \forall i = \overline{1,m}$.

The number of iterations required for Algorithm 4 to converge to the fixed point is significantly smaller than that of Algorithm 1 with $\rho_i = \rho^*$, $\forall i = \overline{1,m}$. This is because a "good" estimate of $\gamma_i^*, \forall i = \overline{1,m}$ allows us to have $z_i \approx u_i \approx \frac{v_i}{2}, \forall i = \overline{1,m}$ at each iteration. Hence the rate of convergence of v_i near the fixed point is close to $\frac{1}{2}$, which is the best possible rate for Algorithm 3 with $\rho_i = \rho^*, \forall i = \overline{1,m}$ when $d_1 = d_2 = \ldots = d_1$.

As written in Section III-C, our scheme can be considered as a new adaptive RB balancing method. The most notable feature on Algorithm 4 is that it provides a way to update the vector of penalty parameters. Note that the standard RB method was suggested only for the scalar penalty parameter. Although it is possible to extend the RB method to the case of several penalty parameters, the tuning will be problematic due to the large number of penalty parameters.

The downside of Algorithm 4 is Step 3, which requires to inverse the matrix $(H + \sum_{i=1}^{m} \gamma_i(k) Q_i)$, or equivalently to find

Algorithm 4: Time-Varying Penalty Parameters - General Case.

Require: $H, f, Q_i, b_i, L_i, \gamma_i(0) > 0, \forall i = \overline{1,m}$, $k_{\max} > 0, \epsilon_p > 0, \epsilon_d > 0$

$k \leftarrow 0$

repeat

  $x(k+1) \leftarrow$
  $$-\left(H + \sum_{i=1}^{m} \gamma_i(k) Q_i\right)^{-1} \left(f + \sum_{i=1}^{m} \gamma_i(k) Q_i b_i\right)$$

  for $i \leftarrow 1$ to m **do**

    $\gamma_i(k+1) \leftarrow$
    $$\sqrt{(x(k+1)+b_i)^T Q_i (x(k+1)+b_i)}\gamma_i(k)$$

  end for

  $k \leftarrow k+1$

until $k = k_{\max}$

$\gamma_i \leftarrow \gamma_i(k_{\max}), \tilde{z}_i(k) \leftarrow L_i(x_i(k_{\max}) + b_i), \tilde{u}_i(k) \leftarrow L_i(x_i(k_{\max}) + b_i), \forall i = \overline{1,m}$

  repeat

    $x(k+1) \leftarrow$
    $(H + \sum_{i=1}^{m} \gamma_i Q_i)^{-1} \sum_{i=1}^{m} \gamma_i L_i^T (\tilde{z}_i(k) - \tilde{u}_i(k))$
    $- (H + \sum_{i=1}^{m} \gamma_i Q_i)^{-1} (f + \sum_{i=1}^{m} \gamma_i Q_i b_i)$

  for $i \leftarrow 1$ to m **do**

    $\tilde{v}_i(k+1) \leftarrow L_i x(k+1) + \tilde{u}_i(k) + L_i b_i$

    if $\tilde{v}_i(k+1)^T \tilde{v}_i(k+1) \le 1$ **then**

      $\tilde{z}_i(k+1) \leftarrow \tilde{v}_i(k+1)$

    else

      $\tilde{z}_i(k+1) \leftarrow \frac{\tilde{v}_i(k+1)}{\sqrt{\tilde{v}_i(k+1)^T \tilde{v}_i(k+1)}}$

    $\tilde{u}_i(k+1) \leftarrow \tilde{v}_i(k+1) - \tilde{z}_(k+1)$

    $r_{p,i}(k+1) \leftarrow \tilde{u}_i(k+1) - \tilde{u}_i(k)$

    $r_{d,i}(k+1) \leftarrow -\gamma_i L_i^T (\tilde{z}_i(k+1) - \tilde{z}_i(k))$

  $k \leftarrow k+1$

  $r_p(k) \leftarrow [r_{p,1}(k)^T \; r_{p,2}(k)^T \; \ldots r_{p,m}(k)^T]$

  $r_d(k) \leftarrow [r_{d,1}(k)^T \; r_{d,2}(k)^T \; \ldots r_{d,m}(k)^T]$

until $||r_p(k)||_\infty \le \epsilon_p$ and $||r_d(k)||_\infty \le \epsilon_d$

the solution of the following linear system:

$$\left(H + \sum_{i=1}^{m} \gamma_i(k) Q_i\right) x(k+1) = -\left(f + \sum_{i=1}^{m} \gamma_i(k) Q_i b_i\right). \tag{97}$$

The eigenvalues/eigenvectors decomposition trick in Section II-I-B works only with $m = 1$. Solving (97), e.g., via Gaussian elimination can be computationally demanding, especially for a large n. In this article, we use the conjugate gradient (CG) method [26] to solve (97). The CG method is an iterative method, that is well known to be effective for (97) since $(H + \sum_{i=1}^{m} \rho_i(k) Q_i)$ is a positive definite matrix. The interested reader is referred to [26] for more details on the CG method.

V. Application to Model Predictive Control

We apply the results developed in this article to the optimization problem of the MPC method with both linear and quadratic inequality constraints. It is well known that such constraints occur in many applications, e.g., in MPC for electric drives [11], power electronics [1], ellipsoidal corridors in robotics [28], or closed-loop min-max MPC [8].

We consider the following linear discrete-time system:

$$\chi(t+1) = \mathcal{A}\chi(t) + \mathcal{B}\nu(t) \tag{98}$$

where $\chi \in \mathbb{R}^{n_\chi}$ is the state, $\nu \in \mathbb{R}^{n_\nu}$ is the control input, t is the time step, and $\mathcal{A} \in \mathbb{R}^{n_\chi \times n_\chi}, \mathcal{B} \in \mathbb{R}^{n_\chi \times n_\nu}$.

Consider the following cost function:

$$\mathcal{J} = \sum_{j=0}^{N-1} \left(||\chi(t+j)||_{\mathcal{Q}}^2 + ||\nu(t+j)||_{\mathcal{R}}^2\right) + ||\chi(t+N)||_{\mathcal{P}}^2. \tag{99}$$

At each discrete-time step $t \in \mathcal{Z}, t \ge 0$, MPC solves the following optimization problem:

$$\min_{\nu(t),\ldots,\nu(t+N-1)} \{\mathcal{J}\}$$
$$\text{s.t.} \begin{cases} \chi(t+j+1) = \mathcal{A}\chi(t+j) + \mathcal{B}\nu(t+j), j = \overline{1, N-1} \\ \chi(t+j) \in \mathcal{C}_\chi, j = \overline{1, N} \\ \nu(t+j) \in \mathcal{C}_\nu, j = \overline{0, N-1} \end{cases} \tag{100}$$

where $\chi(t+j), \nu(t+j)$ are, respectively, the predicted state and the predicted input at time $t+j$ given the current state $\chi(t)$ at time t, $\mathcal{Q} \in \mathbb{S}^{n_\chi}$, $\mathcal{P} \in \mathbb{S}_+^{n_\chi}$ are stage and terminal weighting matrices on the state, $\mathcal{R} \in \mathbb{S}_+^{n_\nu}$ is the input weighting matrix. The sets $\mathcal{C}_\chi, \mathcal{C}_\nu$ are

$$\begin{cases} \mathcal{C}_\chi = \{\chi \in \mathbb{R}^{n_\chi} | -\mathbf{1} \le \mathcal{G}\chi \le \mathbf{1}\}, \\ \mathcal{C}_\nu = \{\nu \in \mathbb{R}^{n_\nu} | \nu^T \mathbf{Q}_i \nu \le 1, \forall i = \overline{1, m_\nu}\} \end{cases} \tag{101}$$

where $\mathcal{G}$ is a matrix of appropriate dimension, and $\mathbf{Q}_i \in \mathbb{S}^{n_\nu}, i = \overline{1, m_\nu}$. This implies that $\mathcal{C}_\chi$ is the intersection of linear constraints, and $\mathcal{C}_\nu$ is the intersection of ellipsoids. The sets $\mathcal{C}_\chi, \mathcal{C}_\nu$ are considered to demonstrate that our algorithms can cope efficiently with both linear and quadratic constraints. $\mathcal{C}_\chi, \mathcal{C}_\nu$ define feasible regions for the state and input, respectively.

Denote $\overline{\nu}^*$ with $\overline{\nu} = [\nu(t)^T \; \nu(t+1)^T \; \ldots \; \nu(t+N-1)^T]^T$ as the optimal solution of (100). The control signal at time t is $\nu(t) = \nu(t)^*$.

In the following, we provide a way to reformulate (100) as a QCQP problem. Using (98), one has

$$\overline{\chi} = \overline{\mathcal{A}}\chi(t) + \overline{\mathcal{B}}\overline{\nu} \tag{102}$$

where

$$\overline{\chi} = [\chi(t+1)^T \; \chi(t+2)^T \; \ldots \; \chi(t+N)^T]^T,$$

$$\overline{\mathcal{A}} = \begin{bmatrix} \mathcal{A} \\ \mathcal{A}^2 \\ \vdots \\ \mathcal{A}^N \end{bmatrix}, \overline{\mathcal{B}} = \begin{bmatrix} \mathcal{B} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathcal{A}\mathcal{B} & \mathcal{B} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}^{N-1}\mathcal{B} & \mathcal{A}^{N-2}\mathcal{B} & \vdots & \mathcal{B} \end{bmatrix}.$$

Using (102), we rewrite the cost function (99) as

$$\mathcal{J} = \overline{\nu}^T \mathcal{H} \overline{\nu} + 2\chi(t)^T \overline{f}_0^T \overline{\nu} + \chi(t)^T (\overline{\mathcal{A}}^T \overline{\mathcal{Q}} \overline{\mathcal{A}} + \mathcal{Q})\chi(t) \tag{103}$$

where $\mathcal{H} = \overline{\mathcal{R}} + \overline{\mathcal{B}}^T \overline{\mathcal{Q}} \overline{\mathcal{B}}, \overline{f}_0 = \overline{\mathcal{B}}^T \overline{\mathcal{Q}} \overline{\mathcal{A}}$, and

$$\overline{\mathcal{R}} = \text{diag}(\mathcal{R}, \ldots, \mathcal{R}, \mathcal{R}), \overline{\mathcal{Q}} = \text{diag}(\mathcal{Q}, \ldots, \mathcal{Q}, \mathcal{P}).$$

Note that we can omit the last term $\chi(t)^T (\overline{\mathcal{A}}^T \overline{\mathcal{Q}} \overline{\mathcal{A}} + \mathcal{Q})\chi(t)$ of the cost (103) in the online optimization problem as it does not affect the optimal solution.

Using (102), we rewrite the state constraint (100) as

$$-\mathbf{1} - \overline{\mathcal{G}}\overline{\mathcal{A}}\chi(t) \le \overline{\mathcal{G}}\overline{\mathcal{B}}\overline{\nu} \le \mathbf{1} - \overline{\mathcal{G}}\overline{\mathcal{A}}\chi(t) \tag{104}$$

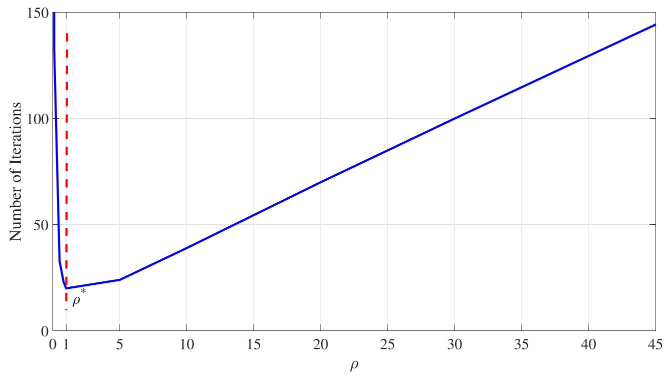

Fig. 1. Number of iterations for Algorithm 1 as a function of ρ for example 1.

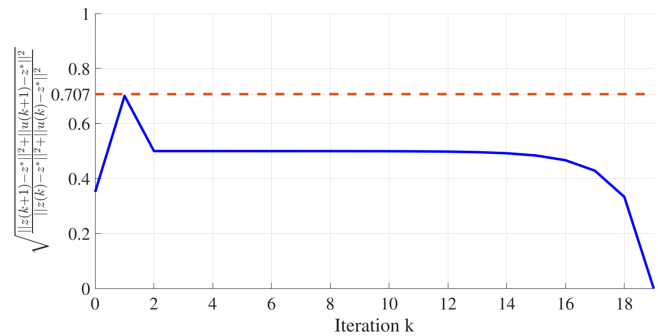

Fig. 2. Observed (solid blue) and theoretical (dashed red) convergence rates of Algorithm 1 as a function of iterations with $\rho = \rho^*$ for example 1.

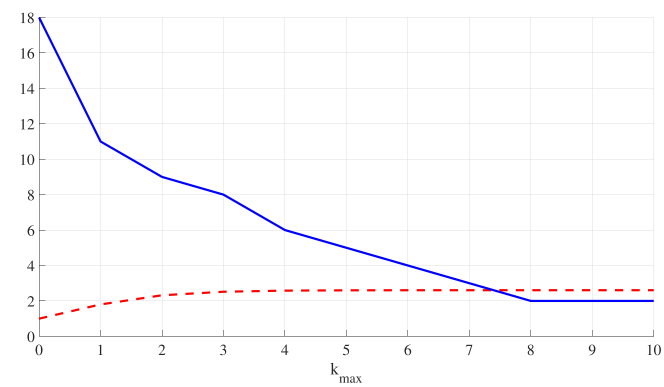

Fig. 3. Number of iterations (solid blue line) and γ (dashed red line) for $\gamma(0) = \rho^*$ as functions of $k_{\max}$ for Algorithm 2 for example 1.

where $\overline{\mathcal{G}} = \mathrm{diag}(\mathcal{G}, \ldots, \mathcal{G}, \mathcal{G})$.

Using (103), (104), we rewrite the online MPC optimization problem as

$$\min_{\overline{\nu}} \{\overline{\nu}^T \mathcal{H} \overline{\nu} + 2\chi(t)^T \overline{f}_0^T \overline{\nu}\},$$
$$\text{s.t.} \begin{cases} -\mathbf{1} - \overline{\mathcal{G}\mathcal{A}}\chi(t) \leq \overline{\mathcal{G}\mathcal{B}}\overline{\nu} \leq \mathbf{1} - \overline{\mathcal{G}\mathcal{A}}\chi(t), \\ \nu(t+j)^T \mathbf{Q}_i \nu(t+j) \leq 1, \forall j = \overline{0, N-1}, \forall i = \overline{1, m_\nu}. \end{cases} \tag{105}$$

It is clear that (105) is a QCQP problem, that can be reformulated as (1), (2). As a result, we can use Algorithm 3 or Algorithm 4 to calculate the solution $\overline{\nu}^*$ of (105).

VI. Numerical Examples

This section illustrates the potential benefit of the new methods by simulations of two examples.

A. Example 1

We first consider a very simple QCQP problem (1), (2) with

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, f = \begin{bmatrix} 3 \\ 2 \end{bmatrix},$$
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Using Proposition 3, one obtains $\rho^* = 1$. Fig. 1 presents the number of iterations of Algorithm 1 as a function of ρ with the following parameters $\epsilon_p = 10^{-5}, \epsilon_p = 10^{-5}$ and $u(0) = [5.1025 \ \ 4.3086]^T, z(0) = [-0.5 \ \ -1.2]^T$. Using Fig. 1, it can be observed that the worst-case optimal penalty parameter performs well.

Fig. 2 shows the observed convergence rate (solid blue line) and the worst-case theoretical convergence rate (dashed red line) of Algorithm 1 as a function of iterations with $\rho = \rho^*$ and $u(0) = [5.1025 \ \ 4.3086]^T, z(0) = [-0.5 \ \ -1.2]^T$. The observed convergence rate is computed as

$$\sqrt{\frac{||z(k+1) - z^*||^2 + ||u(k+1) - z^*||^2}{||z(k) - z^*||^2 + ||u(k) - z^*||^2}}.$$

It can be seen that the peak of the observed convergence rate is 0.7, which is close to the worst-case theoretical convergence rate.

Using Algorithm 2, Fig. 3 shows the number of iterations (solid blue line) and γ (dashed red line) for $\gamma(0) = \rho^*$ as functions of $k_{\max}$. Note that to demonstrate the efficacy of using the adaptive algorithm to select γ, Fig. 3 shows the number of iterations excluding the first $k_{\max}$ iterations where the selection of γ is being established. Using Fig. 3, it can be observed that the number of iterations required for Algorithm 2 is drastically reduced compared to that for Algorithm 1.

B. Example 2

Consider the following two-state, two-input system

$$\chi(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \chi(t) + \begin{bmatrix} 0.084 & 0.180 \\ 0.076 & 0.134 \end{bmatrix} \nu(t). \tag{106}$$

The input and state constraints are

$$-10 \leq \chi_1 \leq 10, \ \ \nu_1^2 + \nu_2^2 \leq 1. \tag{107}$$

The weighting matrices are

$$\mathcal{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathcal{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The matrix $\mathcal{P}$ is obtained from the discrete-time algebraic Riccati equation leading to

$$\mathcal{P} = \begin{bmatrix} 3.5734 & 4.1012 \\ 4.1012 & 17.1309 \end{bmatrix}.$$

The prediction horizon N is $N = 40$.

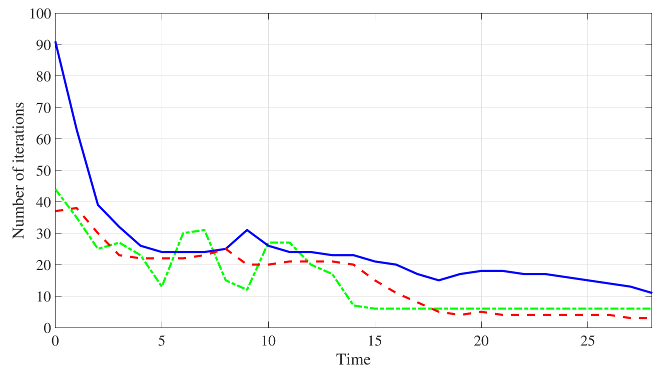

Fig. 4. Number of iterations for Algorithm 3 (solid blue), for Algorithm 4 (dashed red), for the IPM (dash-dot green) as functions of time for example 2.

TABLE I
AVERAGE AND MAXIMUM ONLINE COMPUTATION TIMES FOR ONE SAMPLING INTERVAL OF ALGORITHM 3, OF ALGORITHM 4, AND OF THE IPM FOR EXAMPLE 2

	Algorithm 3	Algorithm 4	IPM
Average	13.9637×10^{-4}[s]	8.2139×10^{-4}[s]	0.0425[s]
Maximum	0.0077[s]	0.0085[s]	0.3494[s]

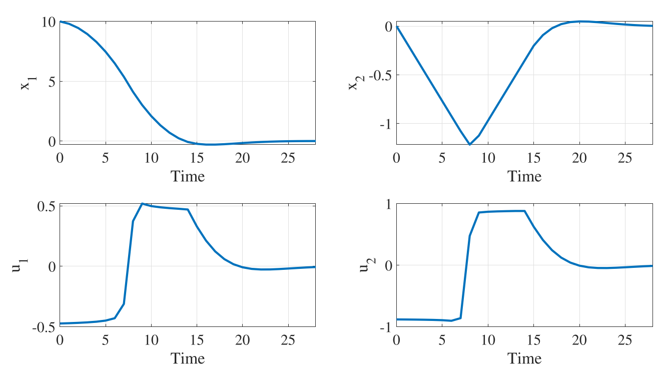

Fig. 5. State and input trajectories as functions of time for example 2.

Since $\mathbf{L}$ in (70) is not a full row rank matrix, the penalty parameter ρ^* for Algorithm 3 is chosen heuristically as in Remark 10. As a result, one gets $\rho^* = 2$.

For the initial condition $\chi(0) = [10 \;\; 0]^T$, Fig. 4 presents the number of iterations for Algorithm 3 (solid blue), and for Algorithm 4 with $k_{\max} = 1$ (dashed red) as functions of time. The tolerances are $\epsilon_d = 10^{-3}$, $\epsilon_p = 10^{-3}$. For comparison purpose, Fig. 4 also presents the number of iterations using the interior point method (IPM) via the CONEPROG function (dash-dot green) of MATLAB 2022b. To apply CONEPROG, we recast the quadratic constraints as second order cone constraints using the method in [19]. It can be seen that the numbers of iterations of Algorithm 4 and of the IPM are generally smaller than that of Algorithm 3.

Using the TIC/TOC function of MATLAB 2022b, Table I shows the average and the maximum online computation times for one sampling interval of Algorithm 3, of Algorithm 4, and of the IPM. It can be observed that the online computation times of the IPM are larger than that of Algorithm 3 and Algorithm 4. A reason for this is that the IPM has a high computation cost per iteration.

Finally, Fig. 5 presents the state and the input trajectories as functions of time.

VII. CONCLUSION

In this article, we proposed two new ADMM-based algorithms for solving quadratically constrained quadratic programming problems. In the first algorithm, we provided a closed form expression for the optimal penalty parameter that guarantees the smallest convergence factor. We also showed that the residual sequence converges at linear rate. In the second algorithm, we proposed a way to tune adaptively the vector of penalty parameters to achieve fast convergence. We showed that the second algorithm can be considered as a new residual balancing method, where the parameters are dynamically updated using the problem data. We validated the effectiveness of the proposed methods via two numerical examples. A toolbox for the proposed method is currently under development.

APPENDIX

A. Proof of Theorem 1

Using (56), it is clear that if $\gamma(0) > 0$, then $\gamma(k) \geq 0, \forall k \geq 1$. Consider the following function:

$$V(\gamma(k)) = -\frac{1}{2}(f + \gamma(k)Qb)^T (H + \gamma(k)Q)^{-1}(f + \gamma(k)Qb) - \frac{1}{2}\gamma(k) + \frac{1}{2}\gamma(k)b^T Qb. \quad (108)$$

Using (9), $V(\gamma(k))$ is the Lagrange dual function of (18) and (19), where γ is the dual variable, i.e., $\gamma = \theta$. Recall that $V(\gamma)$ is concave and bounded from above. One has

$$\frac{dV(\gamma(k))}{d\gamma(k)} = \frac{1}{2}(f + \gamma(k)Qb)^T Q_\gamma (f + \gamma(k)Qb) - b^T Q(H + \gamma(k)Q)^{-1}(f + \gamma(k)Qb) - \frac{1}{2} + \frac{1}{2}b^T Qb \quad (109)$$

with $Q_\gamma = (H + \gamma(k)Q)^{-1} Q (H + \gamma(k)Q)^{-1}$. Thus, using (51)

$$\frac{dV(\gamma(k))}{d\gamma(k)} = \frac{1}{2}\left((x(k+1) + b)^T Q (x(k+1) + b) - 1\right). \quad (110)$$

Using (56) and (110), one has

1) If $(x(k+1) + b)^T Q(x(k+1) + b) > 1$, then $\frac{dV(\gamma(k))}{d\gamma(k)} > 0$, or equivalently $V(\gamma(k))$ is a strictly increasing function. Using (56) one has $\gamma(k+1) > \gamma(k)$. As a consequence, $V(\gamma(k+1)) > V(\gamma(k))$.
2) If $(x(k+1) + b)^T Q(x(k+1) + b) < 1$, then $\frac{dV(\gamma(k))}{d\gamma(k)} < 0$, or equivalently $V(\gamma(k))$ is a strictly decreasing function. Using (56) one has $\gamma(k+1) < \gamma(k)$. As a consequence, $V(\gamma(k+1)) > V(\gamma(k))$.

It follows that $V(\gamma(k))$ is strictly increasing along the trajectories of (56) until $\gamma(k)$ reaches to the fixed point γ^*. Using (56), γ^* satisfies the following equation:

$$\gamma^* = h(\gamma^*) \quad (111)$$

where $h(\gamma) = \sqrt{(x(\gamma) + b)^T Q(x(\gamma) + b)}\gamma$, and $x(\gamma) = -(H + \gamma Q)^{-1}(f + \gamma Qb)$.

Equation (111) can have at most two solutions: 1) γ^* is such that $(x^*+b)^TQ(x^*+b)=1$; 2) $\gamma^*=0$. It is clear that one needs only to discuss the case $\gamma^*=0$, since in the other case $\frac{dV}{d\gamma}=0$. One has

$$\frac{dh}{d\gamma}\bigg|_{\gamma=0}=\sqrt{(-H^{-1}f+b)^TQ(-H^{-1}f+b)}. \tag{112}$$

For any small enough $\gamma(0)$, $\lim_{k\to\infty}\gamma(k)\to 0$, if and only if $\frac{dh}{d\gamma}\Big|_{\gamma=0}<1$. It follows that:

$$(-H^{-1}f+b)^TQ(-H^{-1}f+b)<1. \tag{113}$$

Using (110), one has

$$\frac{dV(\gamma)}{d\gamma}\bigg|_{\gamma=0}=\frac{1}{2}\left(1-(-H^{-1}f+b)^TQ(-H^{-1}f+b)\right).$$

Hence, $\frac{dV(\gamma)}{d\gamma}\Big|_{\gamma=0}>0$. Consequently the KKT condition (10) is satisfied. In other words, $\gamma^*=0$ is the optimal solution. The proof is complete. □

B. Proof of Proposition 5

One has

$$\begin{aligned}\frac{dh}{d\gamma}&=\sqrt{(x(\gamma)+b)^TQ(x(\gamma)+b)}\\&\quad-\frac{\gamma x(\gamma)+b)^TQ(H+\gamma Q)^{-1}Q(x(\gamma)+b)}{\sqrt{(x(\gamma)+b)^TQ(x(\gamma)+b)}}\\&=\frac{(x(\gamma)+b)^T\left(Q-\gamma Q(H+\gamma Q)^{-1}Q\right)(x(\gamma)+b)}{\sqrt{(x(\gamma)+b)^TQ(x(\gamma)+b)}}.\end{aligned} \tag{114}$$

Recall that $Q=L^TL$. Using proposition 1, one has $Q-\gamma Q(H+\gamma Q)^{-1}Q\succ 0, \forall\gamma\geq 0$. Hence $\frac{dh}{d\gamma}>0$, and $h(\gamma)$ is an increasing function. Consequently, for any $\gamma(0)>0$

1) If $\gamma(1)>\gamma(0)$, then

$$\gamma(2)=h(\gamma(1))>h(\gamma(0))=\gamma(1),$$
$$\vdots$$
$$\gamma(k+1)=h(\gamma(k))>h(\gamma(k-1))=\gamma(k)$$

2) If $\gamma(1)<\gamma(0)$, then

$$\gamma(2)=h(\gamma(1))<h(\gamma(0))=\gamma(1),$$
$$\vdots$$
$$\gamma(k+1)=h(\gamma(k))<h(\gamma(k-1))=\gamma(k).$$

It follows that $\gamma(k)$ converges monotonically to γ^*. □

C. Proof of Theorem 2

The main idea of the proof is to show that

$$||\mathbf{z}(k+1)^T-\mathbf{z}^*||_2^2+||\mathbf{u}(k+1)^T-\mathbf{u}^*||_2^2\leq||\mathbf{v}(k+1)^T-\mathbf{v}^*||_2^2 \tag{115}$$

$$||\mathbf{v}(k+1)^T-\mathbf{v}^*||_2^2\leq\lambda\left(||\mathbf{z}(k)^T-\mathbf{z}^*||_2^2+||\mathbf{u}(k)^T-\mathbf{u}^*||_2^2\right) \tag{116}$$

where $\mathbf{v}=[v_1^T\ v_2^T\ \dots\ v_m^T]^T$.

Using (68), and recall that $y_i(k)=\sqrt{\rho}u_i(k), i=\overline{1,m}$, one obtains

$$\mathbf{v}(k+1)=\mathbf{z}(k+1)+\mathbf{u}(k+1).$$

For the fixed point, one has $\mathbf{v}^*=\mathbf{z}^*+\mathbf{u}^*$. Hence

$$(\mathbf{v}(k+1)-\mathbf{v}^*)=(\mathbf{z}(k+1)-\mathbf{z}^*)+(\mathbf{u}(k+1)-\mathbf{u}^*).$$

It follows that:

$$\begin{aligned}||\mathbf{v}(k+1)-\mathbf{v}^*||_2^2&=||\mathbf{z}(k+1)-\mathbf{z}^*+\mathbf{u}(k+1)-\mathbf{u}^*||_2^2\\&=||\mathbf{z}(k+1)-\mathbf{z}^*||_2^2+||\mathbf{u}(k+1)-\mathbf{z}^*||_2^2\\&\quad+2(\mathbf{u}(k+1)-\mathbf{u}^*)^T(\mathbf{z}(k+1)-\mathbf{z}^*).\end{aligned}$$

Hence to prove (115), it suffices to show that

$$(\mathbf{u}(k+1)-\mathbf{u}^*)^T(\mathbf{z}(k+1)-\mathbf{z}^*)\geq 0. \tag{117}$$

Recall that $z_i(k+1)$ is the optimal solution of (65), $i=\overline{1,m}$. Using the KKT optimality condition for (65), one has, $\forall z_i: z_i^Tz_i\leq\rho_i$

$$\left(z_i(k+1)-\sqrt{\rho_i}L_ix(k+1)-\frac{1}{\sqrt{\rho_i}}y_i(k)-\sqrt{\rho_i}L_ib_i\right)^T$$
$$(z_i(k+1)-z_i)\leq 0$$

thus, using (66)

$$\left(z_i(k+1)-v_i(k+1)\right)^T\left(z_i(k+1)-z_i\right)\leq 0.$$

Using (68), one gets

$$u_i(k+1)^T(z_i(k+1)-z_i)\geq 0. \tag{118}$$

Equation (118) holds $\forall z_i: z_i^Tz_i\leq\rho_i$. Since $(z_i^*)^Tz_i^*\leq\rho_i$, one obtains

$$u_i(k+1)^T(z_i(k+1)-z_i^*)\geq 0, i=\overline{1,m}.$$

Consequently

$$\mathbf{u}(k+1)^T(\mathbf{z}(k+1)-\mathbf{u}^*)\geq 0. \tag{119}$$

Using similar arguments, and since z_i^* is the optimal solution of (26) for x^*,y_i^*,v_i^*, one gets

$$(\mathbf{u}^*)^T(\mathbf{z}^*-\mathbf{z}(k+1))\geq 0. \tag{120}$$

Combining (119) and (120), one obtains

$$\mathbf{u}(k+1)^T(\mathbf{z}(k+1)-\mathbf{u}^*)+(\mathbf{u}^*)^T(\mathbf{z}^*-\mathbf{z}(k+1))\geq 0.$$

It follows that (117) holds. Consequently, (115) holds.

It remains to prove (116). One has, for the fixed point $(\mathbf{v}^*,\mathbf{z}^*,\mathbf{u}^*)$ of (72)

$$\mathbf{v}^*=\mathbf{R}\mathbf{z}^*+(\mathbf{I}-\mathbf{R})\mathbf{u}^*+\mathbf{b}. \tag{121}$$

Combining (72) and (121), one obtains

$$\begin{aligned}&||\mathbf{v}(k+1)-\mathbf{v}^*||_2^2\\&\quad=||\mathbf{R}\left(\mathbf{z}(k)-\mathbf{z}^*\right)+(\mathbf{I}-\mathbf{R})\left(\mathbf{u}(k)-\mathbf{u}^*\right)||_2^2\end{aligned}$$

or, equivalently

$$\begin{aligned}||\mathbf{v}(k+1)-\mathbf{v}^*||_2^2&=(\mathbf{z}(k)-\mathbf{z}^*)^T\mathbf{R}^2(\mathbf{z}(k)-\mathbf{z}^*)\\&\quad+(\mathbf{u}(k)-\mathbf{u}^*)^T(\mathbf{I}-\mathbf{R})^2(\mathbf{u}(k)-\mathbf{u}^*)\\&\quad+2(\mathbf{z}(k)-\mathbf{z}^*)^T(\mathbf{R}-\mathbf{R}^2)(\mathbf{u}(k)-\mathbf{u}^*).\end{aligned} \tag{122}$$

Because $0\prec\mathbf{R}\prec\mathbf{I}$, one has $\mathbf{R}-\mathbf{R}^2\succ 0$. Hence

$$\|(\mathbf{z}(k)-\mathbf{z}^*)-(\mathbf{u}(k)-\mathbf{u}^*)\|_{(\mathbf{R}-\mathbf{R}^2)}^2\geq 0.$$

It follows that:

$$\begin{aligned}&(\mathbf{z}(k)-\mathbf{z}^*)^T(\mathbf{R}-\mathbf{R}^2)(\mathbf{z}(k)-\mathbf{z}^*)\\&\quad+(\mathbf{u}(k)-\mathbf{u}^*)^T(\mathbf{R}-\mathbf{R}^2)(\mathbf{u}(k)-\mathbf{u}^*)\\&\geq 2(\mathbf{z}(k)-\mathbf{z}^*)^T(\mathbf{R}-\mathbf{R}^2)(\mathbf{u}(k)-\mathbf{u}^*),\end{aligned}\tag{123}$$

Combining (122) and (123), one gets

$$\begin{aligned}||\mathbf{v}(k+1)-\mathbf{v}^*||_2^2&\leq(\mathbf{z}(k)-\mathbf{z}^*)^T\mathbf{R}^2(\mathbf{z}(k)-\mathbf{z}^*)\\&+(\mathbf{u}(k)-\mathbf{u}^*)^T(\mathbf{I}-\mathbf{R})^2(\mathbf{u}(k)-\mathbf{u}^*)\\&+(\mathbf{z}(k)-\mathbf{z}^*)^T(\mathbf{R}-\mathbf{R}^2)(\mathbf{z}(k)-\mathbf{z}^*)\\&+(\mathbf{u}(k)-\mathbf{u}^*)^T(\mathbf{R}-\mathbf{R}^2)(\mathbf{u}(k)-\mathbf{u}^*)\end{aligned}$$

or, equivalently

$$\begin{aligned}||\mathbf{v}(k+1)-\mathbf{v}^*||_2^2&\leq(\mathbf{z}(k)-\mathbf{z}^*)^T\mathbf{R}(\mathbf{z}(k)-\mathbf{z}^*)\\&+(\mathbf{u}(k)-\mathbf{u}^*)^T(\mathbf{I}-\mathbf{R})(\mathbf{u}(k)-\mathbf{u}^*).\end{aligned}\tag{124}$$

Using (34), one has

$$\begin{cases}(\mathbf{z}(k)-\mathbf{z}^*)^T\mathbf{R}(\mathbf{z}(k)-\mathbf{z}^*)\\\quad\leq\bar{\lambda}||\mathbf{z}(k)-\mathbf{z}^*||_2^2\leq\lambda||\mathbf{z}(k)-\mathbf{z}^*||_2^2\\(\mathbf{u}(k)-\mathbf{u}^*)^T(\mathbf{I}-\mathbf{R})(\mathbf{u}(k)-\mathbf{u}^*)\\\quad\leq(1-\underline{\lambda})||\mathbf{u}(k)-\mathbf{u}^*||_2^2\leq\lambda||\mathbf{u}(k)-\mathbf{u}^*||_2^2\end{cases}.$$

Therefore, using (124), one obtains (116). □

D. Proof of Theorem 3

We decompose $\mathbf{W}$ as $\mathbf{W}=\mathbf{E}^T\mathbf{D}\mathbf{E}$, where $\mathbf{E}\in\mathbb{R}^{\mathbf{l}\times\mathbf{l}}$ with $\mathbf{E}^T\mathbf{E}=\mathbf{I}$, and $\mathbf{D}=\text{diag}(\mathbf{d}_1,\ldots,\mathbf{d}_l)$. By using the matrix inversion lemma for $\mathbf{R}$ with

$$H_q=H+\rho\sum_{i=1}^m Q_i=H+\rho\mathbf{L}^T\mathbf{L},$$

one has

$$\begin{aligned}\mathbf{R}&=\rho\mathbf{L}\left(H^{-1}-H^{-1}\mathbf{L}^T\left(\frac{1}{\rho}I+\mathbf{L}H^{-1}\mathbf{L}^T\right)^{-1}\mathbf{L}H^{-1}\right)\mathbf{L}^T\\&=\rho\mathbf{W}-\rho\mathbf{W}\left(\frac{1}{\rho}\mathbf{I}+\mathbf{W}\right)^{-1}\mathbf{W}\\&=\mathbf{E}^T\left(\rho\mathbf{D}-\rho\mathbf{D}\left(\frac{1}{\rho}\mathbf{I}+\mathbf{D}\right)^{-1}\mathbf{D}\right)\mathbf{E}\end{aligned}$$

or equivalently

$$\mathbf{R}=\mathbf{E}^T\text{diag}\left(\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1},\ldots,\frac{\rho\mathbf{d}_\mathbf{l}}{\rho\mathbf{d}_\mathbf{l}+1}\right)\mathbf{E}.\tag{125}$$

Hence, $\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1},\ldots,\frac{\rho\mathbf{d}_\mathbf{l}}{\rho\mathbf{d}_\mathbf{l}+1}$ are the eigenvalues of $\mathbf{R}$. Consequently, $1-\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1},\ldots,1-\frac{\rho\mathbf{d}_\mathbf{l}}{\rho\mathbf{d}_\mathbf{l}+1}$, or equivalently $\frac{1}{\rho\mathbf{d}_1+1},\ldots,\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}$ are the eigenvalues of $(\mathbf{I}-\mathbf{R})$. As

$$\mathbf{d}_1\geq\mathbf{d}_2\geq\ldots\mathbf{d}_\mathbf{l}$$

one has

$$\begin{cases}\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1}\geq\frac{\rho\mathbf{d}_2}{\rho\mathbf{d}_2+1}\geq\ldots\geq\frac{\rho\mathbf{d}_\mathbf{l}}{\rho\mathbf{d}_\mathbf{l}+1}\\\frac{1}{\rho\mathbf{d}_1+1}\leq\frac{1}{\rho\mathbf{d}_2+1}\leq\ldots\leq\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}\end{cases}.$$

Therefore, $\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1}$ and $\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}$ are, respectively, the maximal eigenvalue of $\mathbf{R}$, and of $(\mathbf{I}-\mathbf{R})$. So our problem becomes the problem of selecting ρ to minimize the maximum between $\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1}$ and $\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}$.

Note that the function $\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1}=\frac{\mathbf{d}_1}{\mathbf{d}_1+\frac{1}{\rho}}$ is monotonically increasing $\forall\rho>0$. On the other hand, the function $\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}$ is monotonically decreasing. With $\rho=\rho^*=\frac{1}{\sqrt{\mathbf{d}_1\mathbf{d}_\mathbf{l}}}$, one has

$$\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1}=\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}=\frac{\sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_\mathbf{k}}}}{1+\sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_\mathbf{l}}}}.\tag{126}$$

If $\rho>\rho^*$, then

$$\max\left\{\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1},\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}\right\}=\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1}>\frac{\sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_\mathbf{l}}}}{1+\sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_\mathbf{l}}}}.$$

If $\rho<\rho^*$, then

$$\max\left\{\frac{\rho\mathbf{d}_1}{\rho\mathbf{d}_1+1},\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}\right\}=\frac{1}{\rho\mathbf{d}_\mathbf{l}+1}>\frac{\sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_\mathbf{l}}}}{1+\sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_\mathbf{l}}}}.$$

If follows that $\rho=\rho^*$ is the optimal value. □

References

[1] S. Almér, D. Frick, G. Torrisi, and S. Mariéthoz, "Predictive converter control: Hidden convexity and real-time quadratically constrained optimization," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 403–410, Jan. 2021.

[2] K. E. Atkinson and W. Han, *Elementary Numerical Analysis*. Wiley New York, USA, 1993.

[3] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.

[6] M. Cannon and B. Kouvaritakis, "Optimizing prediction dynamics for robust MPC," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1892–1897, Nov. 2005.

[7] R. Carli and M. Dotoli, "Distributed alternating direction method of multipliers for linearly constrained optimization over a network," *IEEE Control Syst. Lett.*, vol. 4, no. 1, pp. 247–252, Jan. 2020.

[8] M. Diehl, "Formulation of closed-loop min–max mpc as a quadratically constrained quadratic program," *IEEE Trans. Autom. Control*, vol. 52, no. 2, pp. 339–343, Feb. 2007.

[9] J. Eckstein and W. Yao, "Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives," *Pac. J. Optim.*, vol. 11, no. 4, pp. 619–644, 2015.

[10] P. A Forero, A. Cano, and G. B Giannakis, "Consensus-based distributed linear support vector machines," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 35–46.

[11] D. Geweth, A. Zanelli, G. Frison, U. Vollmer, and M. Diehl, "Field oriented economic model predictive control for permanent magnet synchronous motors," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9093–9099, 2020.

[12] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, Mar. 2015.

[13] G. H. Golub and C.F.V. Loan. *Matrix Computations*. Baltimore, MD, USA: JHU press, 2013.

[14] H.E. Güven, A. Güngör, and M. Cetin, "An augmented lagrangian method for complex-valued compressed sar imaging," *IEEE Trans. Comput. Imag.*, vol. 2, no. 3, pp. 235–250, Sep. 2016.

[15] B.-S. He, H. Yang, and S.L. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *J. Optim. Theory Appl.*, vol. 106, pp. 337–356, 2000.

[16] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Collaborative sparse regression for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.

[17] P. Krupa, R. Jaouani, D. Limon, and T. Alamo, “A sparse ADMM-based solver for linear MPC subject to terminal quadratic constraint,” *IEEE Trans. Control Syst. Technol.*, vol. 32, no. 6, pp. 2376–2384, Nov. 2024.

[18] Z. Liu, A. Hansson, and L. Vandenberghe, “Nuclear norm system identification with missing inputs and outputs,” *Syst. Control Lett.*, vol. 62, no. 8, pp. 605–612, 2013.

[19] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, “Applications of second-order cone programming,” *Linear algebra Appl.*, vol. 284, no. 1–3, pp. 193–228, 1998.

[20] D. Q. Mayne, M.M. Seron, and S.V. Raković, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[21] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, USA: SIAM, 1994.

[22] H.-N. Nguyen, “Constrained control of uncertain, time-varying, discrete-time systems,” in *Lecture Notes in Control and Information Sciences*, vol. 451. Springer, 2014.

[23] H.-N. Nguyen, “Optimizing prediction dynamics with saturated inputs for robust model predictive control,” *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 383–390, Jan. 2021.

[24] H.-N. Nguyen, “Improved prediction dynamics for robust MPC,” *IEEE Trans. Autom. Control*, vol. 68, no. 9, pp. 5445–5460, Sep. 2023.

[25] H.-N. Nguyen, “Efficient min–max MPC: Achieving a large domain of attraction with short horizon,” *Int. J. Robust Nonlinear Control*, vol. 33, pp. 7020–7044, 2023.

[26] J. Nocedal and S. J Wright, *Numerical Optimization*. Berlin, Germany: Springer, 1999.

[27] A. U. Raghunathan and S.C. Di, “Alternating direction method of multipliers for strictly convex quadratic programs: Optimal parameter selection,” in *Proc. Amer. Control Conf.*, 2014, pp. 4324–4329.

[28] N. van Duijkeren, “Online motion control in virtual corridors-for fast robotic systems,” Ph.D. dissertation, KU Leuven, 2019.

[29] V. Q. Vu, J. Cho, J. Lei, and K. Rohe, “Fantope projection and selection: A near-optimal convex relaxation of sparse pca,” in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2013, vol. 2, pp. 2670–2678.

[30] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, “An ADMM algorithm for a class of total variation regularized estimation problems,” *IFAC Proc. Volumes*, vol. 45, no. 16, pp. 83–88, 2012.

[31] Y. Wang, J. Yang, W. Yin, and Y. Zhang, “A new alternating minimization algorithm for total variation image reconstruction,” *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248–272, 2008.

[32] Z. Wen, D. Goldfarb, and W. Yin, “Alternating direction augmented lagrangian methods for semidefinite programming,” *Math. Program. Computation*, vol. 2, no. 3–4, pp. 203–230, 2010.

[33] B. Wohlberg, “ADMM penalty parameter selection by residual balancing,” 2017, *arXiv:1704.06209*.

Hoai-Nam Nguyen received the engineering degree in automatic control from Bauman, Moscow State Technical University, Moscow, Russia, in 2009, and the Ph.D. degree in automatic control from École Supérieure d’Électricité, Supelec, Gif-sur-Yvette, France, in 2012.

After two years of postdoc, he spent seven years in industry with IFP Energies Nouvelles as a Research Engineer. Since 2021, he works with Telecom-SudParis and Institut Polytechnique de Paris, Palaiseau, France, as an Associate Professor. His main research interests include optimization based control, sensor fusion, embedded systems, and autonomous systems.