



**HAL**  
open science

# Optimal self-adaptive parameter selection for ADMM: quadratically constrained quadratic program

Hoai Nam Nguyen

► **To cite this version:**

Hoai Nam Nguyen. Optimal self-adaptive parameter selection for ADMM: quadratically constrained quadratic program. 2023. hal-04308462

**HAL Id: hal-04308462**

**<https://hal.science/hal-04308462>**

Preprint submitted on 27 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Optimal Self-Adaptive Parameter Selection for ADMM: Quadratically Constrained Quadratic Program

H.-N. Nguyen<sup>†</sup>

**Abstract**—We propose two new algorithms for solving quadratically constrained quadratic programmings (QCQP) 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: i) its performance depends greatly on the efficiency of finding the orthogonal projection on the feasible set; ii) it is not trivial to find the correct penalty parameters. For the first challenge, we provide a way to reformulate the original QCQP problem such that there exists an explicit expression for the orthogonal projection. Hence, the computational cost per iteration is cheap. 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 simulation.

## 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., [2], [6], [23], [9], [24], [22] [4]. 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 orthogonal projection on the feasible set 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. In the context of convex constrained optimization to the best of the author’s knowledge, except the particular case of quadratic program (QP), where explicit expression for a scalar constant penalty parameter was proposed [7], [20], 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) [13], [18]. 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. The dual-mode MPC is one of the most well known MPC formulations. In the dual-mode MPC, recursive feasibility and asymptotic stability are guaranteed by using a terminal cost, and by

imposing the terminal state to lie in a terminal set. This set is constraint-admissible and invariant. For linear systems subject to linear and/or quadratic constraints, with a quadratic cost function and an ellipsoidal terminal constraint, the optimization problem is a QCQP. Note that ADMM was already discussed in the context of QCQP based MPC in [13], [18]. 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 addressed in [13]. In [18], the ellipsoid matrices are required to be positive definite.

Another potential application of ADMM is in robust prediction dynamics based MPC [3], [16], [17], or in interpolating control [15]. They are also real-time optimization based control techniques, where the set invariance concept is heavily exploited. In these techniques, if ellipsoid or the intersection of ellipsoids are used for the invariant set, then the resulting optimization problem is a QPQP.

In this paper, 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 orthogonal projection 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 [7] as a particular case, the first method provides a generalization of the results in [7]. In addition, we provide an improvement of the results in [7] by showing that the residuals sequence is  $Q$ -linearly convergent. In [7], 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 dual variables of the original QCQP for the penalty parameters, the ADMM algorithm can converge in one steps. Clearly, the dual variables of the original QCQP are unknown. We provide a way to update iteratively the penalty parameters such that they converge to the dual variables. The resulting adaptive ADMM algorithm is fully automated. A connection between our second procedure and the well known residual balancing method [11] is presented.

The paper is organized as follows. Section II covers the notation. Section III is concerned with the problem formulation and preliminaries. Section IV is dedicated to the QCQP problem with only one quadratic constraint. Then in Section V, the general case of QCQP is considered. In Section VI, we recall a brief theory of the robust prediction dynamics based MPC. Two simulated examples are evaluated in Section VII before drawing the conclusions in Section VIII.

<sup>†</sup> SAMOVAR - Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau France hoai-nam.nguyen@telecom-sudparis.eu

## II. NOTATION

**Notation:** A positive definite (semi-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 (semi-definite)  $n \times n$  matrices. For symmetric matrices, the symbol  $(*)$  denotes each of its symmetric block.  $\|x\|_P^2 = x^T P x$ ,  $\|x\|_2 = \sqrt{x^T x}$ . 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  $\{1, 2, \dots, n\}$  is denoted by  $\overline{1, n}$ .  $\mathbf{I}_n, \mathbf{0}_n$  are, respectively, the identity matrix and the zero matrix of dimension  $n \times n$ .  $\mathbf{1}_n$  is the vector of ones of dimension  $n \times 1$ . 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  $\|x(k+1) - x^*\| \leq \kappa(k)$ , where  $\kappa(k)$  is  $Q$ -linearly convergent.

## III. PROBLEM FORMULATION AND PRELIMINARIES

### A. Problem Formulation

We aim to solve the following QCQP

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

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

where  $x \in \mathbb{R}^n$ ,  $H \in \mathbb{S}_+^n$ ,  $Q_i \in \mathbb{S}^n, \dots, Q_m \in \mathbb{S}^n$ , and  $f, b_1, \dots, b_m \in \mathbb{R}^n$ .

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

$$\underline{u}_i \leq g_i^T x \leq \bar{u}_i \quad (3)$$

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

$$u_i = \frac{\bar{u}_i - \underline{u}_i}{2}, \tilde{g}_i = \frac{g_i}{u_i} \quad (4)$$

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

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

We can also recast the quadratic constraints

$$x^T \tilde{Q}_i x + 2 \tilde{b}_i^T x \leq \tilde{c}_i \quad (6)$$

as (2) with

$$Q_i = \frac{\tilde{Q}_i}{\tilde{c}_i + \tilde{b}_i^T \tilde{Q}_i^{-1} \tilde{b}_i}, b_i = \tilde{Q}_i^{-1} \tilde{b}_i \quad (7)$$

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

$$L(x, \theta) = \frac{1}{2} x^T H x + f^T x + \sum_{i=1}^m \frac{\theta_i}{2} \left( (x + b_i)^T Q_i (x + b_i) - 1 \right) \quad (8)$$

where  $\theta \in \mathbb{R}^m$  is the dual variable. Using (8), the optimal solution to (1), (2) is given by

$$x^* = - \left( H + \sum_{i=1}^m \theta_i^* Q_i \right)^{-1} \left( f + \sum_{i=1}^m \theta_i^* Q_i b_i \right) \quad (9)$$

where  $\theta^*$  is the solution of the following optimization problem

$$\begin{aligned} & \max_{\theta} \{V(\theta)\}, \\ & \text{s.t. } \theta_i \geq 0, \forall i = \overline{1, m} \end{aligned} \quad (10)$$

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

$$\begin{aligned} V(\theta) = & -\frac{1}{2} \left( f + \sum_{i=1}^m \theta_i Q_i b_i \right)^T \left( H + \sum_{i=1}^m \theta_i Q_i \right)^{-1} \left( f + \sum_{i=1}^m \theta_i Q_i b_i \right) \\ & + \frac{1}{2} \sum_{i=1}^m (\theta_i b_i^T Q_i b_i - \theta_i) \end{aligned} \quad (11)$$

Since strong duality is satisfied, we have the optimal value of (1), (2) is equal to the attained optimal value of the dual problem (10). Hence,  $V(\theta)$  is bounded from above by the optimal value of (1), (2). Using the KKT conditions, the optimal solution  $\theta^*$  of (10) satisfies

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

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

$$\begin{cases} \nabla V_i(\theta^*) \leq 0, & \text{if } \gamma_i^* = 0, \\ \nabla V_i(\theta^*) = 0 & \text{if } \gamma_i^* > 0 \end{cases} \quad (12)$$

In (12),  $\nabla V_i(\theta^*)$  is the  $i$ -th component of the vector  $\nabla V(\theta^*)$ . We will use (11), (12) to show the convergence of the new algorithm in Section IV-B.

### B. Alternating Direction Method of Multipliers

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

$$\min_{x, z} \{g(x)\} \quad (13)$$

$$\text{s.t. } \begin{cases} Ax + Bz = c, \\ z \in Z \end{cases} \quad (14)$$

where  $x \in \mathbb{R}^n, z \in \mathbb{R}^l$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  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 (13), (14) is

$$L_\rho(x, z, y) = g(x) + y^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \quad (15)$$

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

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

$$x(k+1) = \arg \min_x \left\{ g(x) + y(k)^T Ax + \frac{\rho}{2} \|Ax + Bz(k) - c\|_2^2 \right\} \quad (16)$$

$$z(k+1) = \arg \min_{z \in Z} \left\{ y(k)^T Bz + \frac{\rho}{2} \|Ax(k+1) + Bz - c\|_2^2 \right\} \quad (17)$$

$$y(k+1) = y(k) + \rho (Ax(k+1) + Bz(k+1) - c) \quad (18)$$

Under very mild assumptions, it is well-known [2], [5], [10] that the ADMM converges to a globally optimal solution of problem (13), (14) for any value of  $\rho > 0$ . However, the number of iteration to have small residuals depends strongly on the value of  $\rho$ . A poor choice of  $\rho$  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

$$\begin{cases} r_p(k) = Ax(k) + Bz(k) - c, \\ r_d(k) = \rho A^T B (z(k) - z(k-1)) \end{cases} \quad (19)$$

The justification for using  $r_d$  as a residual for dual optimality can be found in [2]. The ADMM algorithm (16), (17), (18) returns a sub-optimal solution point  $(\tilde{x}^*, \tilde{z}^*, \tilde{y}^*)$  of (13), (14) where the suboptimality is determined by the stopping condition  $\|r_p(k)\|_\infty \leq \epsilon_p, \|r_d(k)\|_\infty \leq \epsilon_d$  with given tolerances  $\epsilon_p > 0, \epsilon_d > 0$ . The parameters  $\epsilon_p, \epsilon_d$  are often chosen relative to the scaling of the algorithm iterates [2].

#### IV. 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

$$\min_x \left\{ \frac{1}{2} x^T H x + f^T x \right\} \quad (20)$$

$$\text{s.t. } (x+b)^T Q (x+b) \leq 1 \quad (21)$$

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

##### A. Optimal Fixed Penalty Parameter

The main aim of this section is to provide a way to optimize the fixed penalty parameter  $\rho$  for minimizing the rate of convergence of the ADMM algorithm (16), (17), (18). Define  $L$  as a "square root" matrix of  $Q$ , i.e.,  $L^T L = Q$ . Since  $Q$  is required to be only positive semi-definite, 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

$$z = Lx + Lb \quad (22)$$

**Remark 1:** The motivation of introducing the auxiliary variable  $z$  as in (22) is that in this case the  $z$ -update in (17) can be computed explicitly as will be shown later. If  $z$  is defined as  $z = x$  or  $z = x + b$ , then there is no closed form solution for the  $z$ -update in (17).  $\square$

We rewrite the problem (20), (21) as

$$\min_{x,z} \left\{ \frac{1}{2} x^T H x + f^T x \right\} \quad (23)$$

$$\text{s.t. } \begin{cases} Lx - z + Lb = 0, \\ z^T z \leq 1 \end{cases} \quad (24)$$

It is clear that  $x^*$  is the optimal solution of (23), (24) if and only if  $x^*$  is the optimal solution of (20), (21). For problem (23), (24), the associated augmented Lagrangian takes the following form

$$L_\rho(x, z, y) = \frac{1}{2} x^T H x + f^T x + y^T (Lx - z + Lb) + \frac{\rho}{2} \|Lx - z + Lb\|_2^2 \quad (25)$$

Using (16), the ADMM sub-problem for the  $x$ -update is an unconstrained quadratic program

$$\min_x \left\{ \frac{1}{2} x^T (H + \rho L^T L) x - (\rho L^T z(k) - L^T y(k) - \rho L^T Lb - f)^T x \right\} \quad (26)$$

Problem (26) has the unique solution, recall  $L^T L = Q$

$$\begin{aligned} x(k+1) &= (H + \rho Q)^{-1} (\rho L^T z(k) - L^T y(k) - f - \rho Qb) \\ &= (H + \rho Q)^{-1} (\rho L^T z(k) - L^T y(k)) \\ &\quad - (H + \rho Q)^{-1} (f + \rho Qb) \end{aligned} \quad (27)$$

Using (17), the  $z$ -update is the solution of the following optimization problem

$$\min_z \left\{ z^T z - 2 \left( Lx(k+1) + \frac{1}{\rho} y(k) + Lb \right)^T z \right\}, \quad (28)$$

$$\text{s.t. } z^T z \leq 1$$

Define

$$v(k+1) = Lx(k+1) + \frac{1}{\rho} y(k) + Lb \quad (29)$$

The following lemma is taken from [17], [18]. It enables us to find the closed-form solution for (28).

**Lemma 1.** [17], [18] The optimal solution of (28) is given by

$$z(k+1) = \begin{cases} v(k+1), & \text{if } v(k+1)^T v(k+1) \leq 1, \\ \frac{v(k+1)}{\sqrt{v(k+1)^T v(k+1)}}, & \text{otherwise} \end{cases} \quad (30)$$

Finally, using (18) the  $y$ -update is given as

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

thus, using (29)

$$y(k+1) = \rho (v(k+1) - z(k+1)) \quad (31)$$

Using (19), (22), the primal and dual residuals are given as

$$\begin{cases} r_p(k) = Lx(k) - z(k) + Lb, \\ r_d(k) = -\rho L^T (z(k) - z(k-1)) \end{cases} \quad (32)$$

Define  $u(k) = \frac{y(k)}{\rho}$ . Using (29), one has

$$r_p(k) = v(k) - u(k-1) - z(k)$$

thus, using (31)

$$r_p(k) = u(k) - u(k-1) \quad (33)$$

Combining (27), (29), (30), (31), (33), we obtain Algorithm 1, which shows the particularization of the ADMM method for solving (23), (24). Since problem (23), (24) is strictly convex, it is well known [2] that Algorithm 1 converges to the fixed point, and that the fixed point is the optimal solution of (23), (24).

---

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

---

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

---

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  $\rho$  that minimizes this rate of convergence.

Substituting (27) into (29), one gets

$v(k+1) = Rz(k) + (I - R)u(k) - L(H + \rho Q)^{-1} (f + \rho Q b) + Lb$   
or equivalently

$$v(k+1) = Rz(k) + (I - R)u(k) + L(H + \rho Q)^{-1} (Hb - f) \quad (34)$$

where

$$R = \rho L(H + \rho Q)^{-1} L^T \quad (35)$$

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 show that  $R \prec I$ .

**Proposition 1:** Consider  $R$  in (35). One has  $R \prec I, \forall \rho > 0$ .

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

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

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.  $\square$

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

$$\lambda = \max\{1 - \underline{\lambda}, \bar{\lambda}\} \quad (36)$$

Clearly,  $0 < \lambda < 1$ .

The following result shows that Algorithm 1 converges linearly to the fixed point for any  $\rho > 0$ .

**Theorem 1:** Consider Algorithm 1. The following relation holds

$$\begin{aligned} & \|z(k+1) - z^*\|_2^2 + \|u(k+1) - u^*\|_2^2 \\ & \leq \lambda (\|z(k) - z^*\|_2^2 + \|u(k) - u^*\|_2^2) \end{aligned} \quad (37)$$

**Proof:** See Appendix A.  $\square$

Theorem 1 states that the sequence  $\{(z(k), u(k))\}$  is  $Q$ -linearly convergent with the rate  $\lambda$ . The optimal penalty parameter  $\rho^*$  should be chosen to minimize  $\lambda$ . Note that minimizing  $\lambda$  is equivalent to minimize the maximal eigenvalue of the matrix  $\text{diag}(R, I - R)$ .

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

$$d_1 \geq d_2 \geq \dots \geq d_l > 0 \quad (38)$$

We have the following result.

**Theorem 2:** The optimal penalty parameter  $\rho$  is given by  $\rho^* = \frac{1}{\sqrt{d_1 d_l}}$ .

**Proof:** See Appendix B.  $\square$

**Remark 2:** With  $\rho = \rho^*$ , the corresponding convergence rate is  $\lambda^* = \frac{\sqrt{d_1}}{1 + \sqrt{d_1}}$ . 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  $\lambda^* \geq \frac{1}{2}$ .  $\square$

## B. Self-Adaptive Penalty Parameter

In this section we provide a new iteration scheme to find the solution of (20), (21). Here we are interested in optimizing the penalty parameter  $\rho$ . However unlike Section IV-A,  $\rho$  is a function of  $k$ . We will use  $\gamma(k)$  instead of  $\rho(k)$  in the ADMM scheme to distinguish the fixed penalty parameter case and the time-varying parameter case. For simplicity of discussion, the iteration counter  $k$  is also considered as a time index.

To facilitate the reading, we summarize here the  $z$ -,  $y$ -updates at iteration  $k-1$ , and the  $x$ -,  $v$ -updates at iteration  $k$ , using the ADMM scheme (27), (29), (30), (31) with a time-varying  $\gamma(k)$ .

$$z(k) = \begin{cases} v(k), & \text{if } v(k)^T v(k) \leq 1, \\ \frac{v(k)}{\sqrt{v(k)^T v(k)}}, & \text{otherwise} \end{cases}, \quad (39)$$

$$y(k) = \gamma(k-1) (v(k) - z(k)) \quad (40)$$

$$\begin{aligned} x(k+1) = & (H + \gamma(k)Q)^{-1} (\gamma(k)L^T z(k) - L^T y(k)) \\ & - (H + \gamma(k)Q)^{-1} (f + \gamma(k)Qb), \end{aligned} \quad (41)$$

$$v(k+1) = Lx(k+1) + \frac{1}{\gamma(k)} y(k) + Lb, \quad (42)$$

Using (39), one gets

$$z(k) = \alpha(k)v(k) \quad (43)$$

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

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

$\alpha(k)$  can be arbitrarily close to zero when  $v(k)^T v(k)$  is large. However,  $\alpha(k)$  is always different from zero. Substituting (43) into (40), one obtains

$$y(k) = \gamma(k-1) (1 - \alpha(k)) v(k) \quad (44)$$

With a slight abuse of notation, define

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

By substituting (41), (43), (44) into (42), one obtains

$$v(k+1) = A(k)v(k) + L(H + \gamma(k)Q)^{-1}(Hb - f) \quad (45)$$

with

$$A(k) = \alpha(k)R(k) + (1 - \alpha(k))\frac{\gamma(k-1)}{\gamma(k)}(I - R(k)) \quad (46)$$

Equation (45) describes a time-varying system with  $v(k)$  being 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  $A_1(k) = R(k)$ ,  $A_2(k) = \frac{\gamma(k-1)}{\gamma(k)}(I - R(k))$ . It is clear that the convergence rate of  $v(k)$  to the fixed point  $v^*$  depends greatly on  $A(k)$ . For example, if  $A(k)$  has an eigenvalue, that lies outside the unit circle, then  $v(k)$  diverges to infinity.

**Remark 3:** If  $\gamma$  is constant, then  $A_1(k) = R$ ,  $A_2(k) = I - R$ . The results in Section IV-A can be interpreted as to select  $\gamma$  to minimize the maximal eigenvalue of  $A_1$  and of  $A_2$ .  $\square$

Using (46), one has

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

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

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

or, equivalently

$$\gamma(k) = \left(\frac{1}{\alpha(k)} - 1\right)\gamma(k-1) = \left(\sqrt{v(k)^T v(k)} - 1\right)\gamma(k-1) \quad (48)$$

Substituting (48) into (47), one gets

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

Thus, using (45)

$$v(k+1) = \frac{v(k)}{\sqrt{v(k)^T v(k)}} + L(H + \gamma(k)Q)^{-1}(Hb - f) \quad (49)$$

Substituting (48) into (41), one obtains

$$\begin{aligned} x(k+1) &= -(H + \gamma(k)Q)^{-1}(f + \gamma(k)Qb) \\ &= (H + \gamma(k)Q)^{-1}(Hb - f) - b \end{aligned} \quad (50)$$

Using (49), (50), one gets

$$v(k+1) = \frac{v(k)}{\sqrt{v(k)^T v(k)}} + Lx(k+1) + Lb \quad (51)$$

Note that at iteration  $k$  only  $\gamma(k-1)$ ,  $v(k)$  are required in equations (48), (50), (51) to update  $\gamma(k)$ ,  $x(k+1)$ ,  $v(k+1)$ . Hence (48), (50) (51) can be used to find the fixed point  $\gamma^*$ ,  $x^*$ ,  $v^*$  of the original scheme (39), (40), (41), (42).

In the following we provide an alternative way to find the fixed point  $\gamma^*$  that does not require  $v(k)$ . The idea is to use the Gauss-Seidel method [8]. As it will be shown later this method allows us to cover also the case  $\alpha(k) = 1$ . Using this method,  $v(k+1)$ ,  $\gamma(k+1)$  are updated in two steps

1) **Step 1:** Fix  $\gamma(k)$ , calculate  $v(k+1)$  as a solution of the following nonlinear equation

$$v = \frac{v}{\sqrt{v^T v}} + Lx(k+1) + Lb \quad (52)$$

where  $x(k+1)$  is given in (50).

2) **Step 2:** Update  $\gamma(k+1)$  using (48) and  $v(k+1)$

We have the following result.

**Proposition 2:** A solution  $v$  of (52) is given as

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

**Proof:** Using (53), one has

$$\frac{v}{\sqrt{v^T v}} = \frac{L(x(k+1) + b)}{\sqrt{(x(k+1) + b)^T Q(x(k+1) + b)}}$$

It follows that

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

The proof is complete.  $\square$

Using Proposition 2 after Step 1, one gets

$$v(k+1) = \left(1 + \frac{1}{\sqrt{(x(k+1)+b)^T Q(x(k+1)+b)}}\right) L(x(k+1) + b) \quad (54)$$

At Step 2, by substituting (54) into (48), one obtains

$$\begin{aligned} \gamma(k+1) &= (\sqrt{v(k+1)^T v(k+1)} - 1)\gamma(k) \\ &= \sqrt{(x(k+1) + b)^T Q(x(k+1) + b)}\gamma(k) \end{aligned} \quad (55)$$

Note that only  $\gamma(k)$  is required to update  $\gamma(k+1)$  in equation (55).

The following result holds.

**Theorem 3:** The iteration scheme (50), (55) converges to the optimal solution  $x^*$  of (20), (21) for any  $\gamma(0) > 0$ .

**Proof:** See Appendix C.  $\square$

**Remark 4:** As the iteration scheme (50), (55) has a root in the ADMM, it is possible to prove the convergence of (50), (55) to the optimal solution via the ADMM theory. The main advantage of the proof of Theorem 3 in the paper is that it shows that  $\gamma$  can be selected as the dual variable for the original problem (20), (21). Note that unlike Algorithm 1,  $\gamma$  in (50), (55) is allowed to be zero. If  $\gamma^* = 0$ , then the constraint (21) is inactive. Otherwise, (21) is active.  $\square$

The following result holds.

**Proposition 3:** Using (55),  $\gamma(k)$  converges to the fixed point  $\gamma^*$  monotonically for any  $\gamma(0) > 0$ .

**Proof:** See Appendix D.  $\square$

Concerning the convergence rate, it is well known [1] that using equation (55),  $\gamma(k)$  converges linearly to the fixed point  $\gamma^*$ . The convergence rate depends on  $\frac{d\gamma}{d\gamma}$  evaluated at the fixed point  $\gamma = \gamma^*$ . The parameter  $\gamma$  converges quickly to  $\gamma^*$  when  $\gamma$  is far from  $\gamma^*$ . The speed is slower when  $\gamma$  is close to  $\gamma^*$ . This behavior is very typical for first-order optimization methods.

In the following we will analyze the behavior of Algorithm 1 if we use  $\rho = \gamma^*$ . It can be verified that Algorithm 1 converges in one step with  $z(0) = 0, u(0) = 0$  if  $\gamma^* = 0$ , and with  $z(0) = L(x^* + b), u(0) = L(x^* + b)$  if  $\gamma^* > 0$ . This analysis suggests that  $\rho = \gamma^*$  is the best choice. However,  $\gamma^*$  is unknown. Our idea is to use the scheme (50), (55) to obtain a very rough approximation of  $\gamma^*$ . Because  $\gamma$  converges quickly to  $\gamma^*$  when  $\gamma$  is far from  $\gamma^*$ , we need only to make a few iterations for (50), (55). We then use the obtained estimate as the penalty parameter in Algorithm 1.

We will now analyze the behavior of Algorithm 1 with  $\rho = \tilde{\gamma}^*$  near the fixed point  $(x^*, v^*, z^*, u^*)$ , where  $\tilde{\gamma}^*$  is an approximation of  $\gamma^*$ . At step 3 of Algorithm 1, one has

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

where  $\tilde{x}^* = -(H + \tilde{\gamma}^*Q)^{-1}(f + \tilde{\gamma}^*Qb) \approx x^*$ . Hence one should have  $z(k) \approx u(k)$  at convergence. At step 9, one gets  $z(k) \approx u(k) \approx \frac{1}{2}v(k)$ . At step 3, one obtains

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

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

We summarize our development in Algorithm 2.

---

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

---

**Require:**  $H, f, Q, b, L, \gamma(0) > 0, T_{max} > 0, \epsilon_d > 0, \epsilon_p > 0$

1:  $t \leftarrow 0$

2: **repeat**

3:  $x(t+1) \leftarrow -(H + \gamma(t)Q)^{-1}(f + \gamma(t)Qb)$

4:  $\gamma(t+1) \leftarrow \sqrt{(x(t+1) + b)^T Q (x(t+1) + b)} \gamma(t)$

5:  $t \leftarrow t + 1$

6: **until**  $t = T_{max}$

7: Using Algorithm 1 with  $\rho \leftarrow \gamma(t), z(0) \leftarrow L(x(t) + b), u(0) \leftarrow L(x(t) + b)$

---

A good choice for  $\gamma(0)$  in Algorithm 2 is  $\gamma(0) = \rho^*$ . Step 6 implies that we run Algorithm 2 only for a few number of iterations  $T_{max}$ . Note that other criteria can be used, e.g., we stop to update  $\gamma$  when the speed of change or progress in  $\gamma$  is less than some tolerance value.

Results of numerical experiment given in Section VII 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  $L_h^T L_h = H, V_h^T V_h = I, D_h$  is a diagonal matrix, and

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

One has

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

The matrix  $I + \gamma(k)D$  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 [2], [11], [25]. The RB method is based on the observation that increasing  $\rho$  or equivalently  $\gamma$  strengthens the penalty term in the augmented Lagrangian (25), yielding smaller primal residual  $r_p$  but larger dual one  $r_d$ . Conversely, decreasing  $\gamma$  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  $\gamma$  to keep both  $r_p$  and  $r_d$  of similar magnitude. At iteration  $k$ , a simple way for this goal is

$$\gamma(k) = \begin{cases} \eta\gamma(k-1), & \text{if } \|r_p(k)\| \geq \tau\|r_d(k)\|, \\ \frac{\gamma(k-1)}{\eta}, & \text{if } \|r_d(k)\| \geq \tau\|r_p(k)\|, \\ \gamma(k-1), & \text{otherwise} \end{cases} \quad (56)$$

where  $\eta > 1$  and  $\tau > 0$ . In [11],  $\eta$  can also be time-varying. The adaptation (56) is generally turning off after a few number of iterations.

The scheme (56) has been found to be effective for a number of problems [12], [14], [21]. As far as we know, there is no systematic method to select  $\eta, \tau$ . The parameters  $\eta, \tau$  are chosen by the trial and error method. It is well known [25] 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  $\gamma$  in Algorithm 2 is to make  $z(k) \approx u(k)$ . Hence  $z(k+1) - z(k) \approx u(k+1) - u(k)$ , 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  $\eta$  is automatically and dynamically updated. This new ‘‘feedback’’ in Algorithm 2 makes it more robust to bad scaling in the data.

## V. 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  $\mathbf{z} \in \mathbb{R}^{l_1 + \dots + l_m}$  as

$$\begin{cases} z_i = L_i x + L_i b_i, \forall i = \overline{1, m}, \\ \mathbf{z} = [z_1^T \ z_2^T \ \dots \ z_m^T]^T \end{cases} \quad (57)$$

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

$$\begin{aligned} \min_{x, \mathbf{z}} \left\{ \frac{1}{2} x^T H x + f^T x \right\} \\ \text{s.t. } \begin{cases} L_i x + L_i b_i = z_i, \forall i = \overline{1, m}, \\ z_i^T z_i \leq 1, \forall i = \overline{1, m} \end{cases} \end{aligned} \quad (58)$$

Clearly, problem (1), (2) and problem (58) have the same optimal solution  $x^*$ . The augmented Lagrangian for (58) is

$$\begin{aligned} L_\rho(x, \mathbf{z}, \mathbf{y}) = \frac{1}{2} x^T H x + f^T x \\ + \sum_{i=1}^m y_i^T (L_i x + L_i b_i - z_i) + \sum_{i=1}^m \frac{\rho_i}{2} \|L_i x + L_i b_i - z_i\|_2^2 \end{aligned} \quad (59)$$

where  $y_i \in \mathbb{R}^{l_i}, \forall i = \overline{1, m}$  are the dual variables for the equality constraints,  $\rho_i > 0, \forall i = \overline{1, m}$  are penalty parameters.  $\rho_i$  can be different for different  $i, \forall i = \overline{1, m}$ .

Using (16) at iteration  $k$ , the  $x$ -update is the solution of the following quadratic program

$$\begin{aligned} \min_x \left\{ \frac{1}{2} x^T (H + \sum_{i=1}^m \rho_i L_i^T L_i) x \right. \\ \left. - \left( \sum_{i=1}^m (\rho_i L_i^T z_i(k) - L_i^T y_i(k)) - f - \sum_{i=1}^m \rho_i L_i^T L_i b_i \right)^T x \right\} \end{aligned} \quad (60)$$

Recall  $L_i^T L_i = Q_i, \forall i = \overline{1, m}$ . The solution to (60) can be computed analytically as

$$\begin{aligned} x(k+1) &= (H + \sum_{i=1}^m \rho_i Q_i)^{-1} \left( \sum_{i=1}^m (\rho_i L_i^T z_i(k) - L_i^T y_i(k)) \right. \\ &\quad \left. - f - \sum_{i=1}^m \rho_i Q_i b_i \right) \\ &= (H + \sum_{i=1}^m \rho_i Q_i)^{-1} \sum_{i=1}^m (\rho_i L_i^T z_i(k) - L_i^T y_i(k)) \\ &\quad - (H + \sum_{i=1}^m \rho_i Q_i)^{-1} \left( f + \sum_{i=1}^m \rho_i Q_i b_i \right) \end{aligned} \quad (61)$$

Using (17) at iteration  $k$ , the  $\mathbf{z}$ -update is the solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{z}} \left\{ \sum_{i=1}^m -y_i(k)^T z_i \right. \\ \left. + \frac{\rho_i}{2} \left( z_i^T z_i - 2 (L_i x(k+1) + L_i b_i)^T z_i \right) \right\} \\ \text{s.t. } z_i^T z_i \leq 1, \forall i = \overline{1, m} \end{aligned} \quad (62)$$

The cost function and the constraints of (62) 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,  $\forall i = \overline{1, m}$

$$\begin{aligned} \min_{z_i} \left\{ z_i^T z_i - 2 \left( L_i x(k+1) + \frac{y_i(k)}{\rho_i} + L_i b_i \right)^T z_i \right\} \\ \text{s.t. } z_i^T z_i \leq 1 \end{aligned} \quad (63)$$

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

$$v_i(k+1) = L_i x(k+1) + \frac{y_i(k)}{\rho_i} + L_i b_i \quad (64)$$

Using Lemma 1, the solution  $z_i(k+1)$  to the problem (63) is given by,  $\forall i = \overline{1, m}$

$$z_i(k+1) = \begin{cases} v_i(k+1), & \text{if } v_i(k+1)^T v_i(k+1) \leq 1 \\ \frac{v_i(k+1)}{\sqrt{v_i(k+1)^T v_i(k+1)}}, & \text{otherwise} \end{cases} \quad (65)$$

Finally using (18), the update of  $y_i$  is given by,  $\forall i = \overline{1, m}$

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

thus, using (64)

$$y_i(k+1) = \rho_i (v_i(k+1) - z_i(k+1)), \forall i = \overline{1, m} \quad (66)$$

Define  $u_i(k) = \frac{y_i(k)}{\rho_i}, \forall i = \overline{1, m}$ , and  $\mathbf{u} = [u_1^T \ u_2^T \ \dots \ u_m^T]^T$ . With a slight abuse of notation, we use Algorithm 1 to summarize the particularization of the ADMM algorithm (16), (17) (18) to the problem (1), (2). For any  $\rho_i > 0, \forall i = \overline{1, m}$ , as the problem (1), (2) is strictly convex, Algorithm 1 converges to the fixed point, which is the optimal solution to (1), (2).

---

Algorithm 1: Fixed Penalty Parameter - General Case

---

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

1:  $k \leftarrow 0$

2: **repeat**

3:  $x(k+1) \leftarrow$

$$\begin{aligned} (H + \sum_{i=1}^m \rho_i Q_i)^{-1} \sum_{i=1}^m \rho_i L_i^T (z_i(k) - u_i(k)) \\ - (H + \sum_{i=1}^m \rho_i Q_i)^{-1} \left( f + \sum_{i=1}^m \rho_i Q_i b_i \right) \end{aligned}$$

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

5:  $v_i(k+1) \leftarrow L_i x(k+1) + u_i(k) + L_i b_i$

6: **if**  $v_i(k+1)^T v_i(k+1) \leq 1$  **then**

7:  $z_i(k+1) \leftarrow v_i(k+1)$

8: **else**

9:  $z_i(k+1) \leftarrow \frac{v_i(k+1)}{\sqrt{v_i(k+1)^T v_i(k+1)}}$

10:  $u_i(k+1) \leftarrow v_i(k+1) - z_i(k+1)$

11:  $r_{p,i}(k+1) \leftarrow u_i(k+1) - u_i(k)$

12:  $r_{d,i}(k+1) \leftarrow -\rho_i L_i^T (z_i(k+1) - z_i(k))$

13:  $k \leftarrow k + 1$

14:  $r_p(k) \leftarrow [r_{p,1}(k)^T \ r_{p,2}(k)^T \ \dots \ r_{p,m}(k)^T]^T$

15:  $r_d(k) \leftarrow [r_{d,1}(k)^T \ r_{d,2}(k)^T \ \dots \ r_{d,m}(k)^T]^T$

16: **until**  $\|r_p(k)\|_\infty \leq \epsilon_p$  and  $\|r_d(k)\|_\infty \leq \epsilon_d$

---

In the rest of this section, we restrict ourselves to the case  $\rho_i = \rho, \forall i = \overline{1, m}$  to calculate the optimal  $\rho^*$  that minimizes the convergence rate. Define

$$\begin{aligned} \mathbf{L} = \begin{bmatrix} L_1 \\ \vdots \\ L_m \end{bmatrix}, \quad \mathbf{R} = \rho \mathbf{L} (H + \sum_{i=1}^m \rho Q_i)^{-1} \mathbf{L}^T \\ \mathbf{b} = -\mathbf{L} (H + \sum_{i=1}^m \rho Q_i)^{-1} \left( f + \sum_{i=1}^m \rho Q_i b_i \right), \end{aligned} \quad (67)$$

In this section we assume that  $\mathbf{L}$  is a full row rank matrix. In this case  $\mathbf{R} \succ 0, \forall \rho > 0$ . Using similar arguments as in the proof of Proposition 1, it can be shown that  $\mathbf{R} \prec I$ . 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

$$\lambda = \max\{1 - \bar{\lambda}, \underline{\lambda}\} \quad (68)$$



It is clear that  $0 < \lambda < 1$ . Define also  $\mathbf{v} = [v_1^T \ v_2^T \ \dots \ v_m^T]^T$ . Using (61), (64), (67), one obtains

$$\mathbf{v}(k+1) = \mathbf{R}\mathbf{z}(k) + (I - \mathbf{R})\mathbf{u}(k) + \mathbf{b} \quad (69)$$

We have the following result.

**Corollary 1:** Consider Algorithm 1. The following relation holds

$$\begin{aligned} & \|\mathbf{z}(k+1)^T - \mathbf{z}^*\|_2^2 + \|\mathbf{u}(k+1)^T - \mathbf{u}^*\|_2^2 \\ & \leq \lambda (\|\mathbf{z}(k)^T - \mathbf{z}^*\|_2^2 + \|\mathbf{u}(k)^T - \mathbf{u}^*\|_2^2) \end{aligned} \quad (70)$$

**Proof:** The proof is omitted here, as it follows the same lines as the proof of Theorem 1.  $\square$

**Remark 5:** 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. Corollary 1 states that the residual sequence  $\{(z(k), u(k))\}$  is Q-linear convergent. Using Proposition 2 in [7], it is only possible to establish that  $\{(z(k), u(k))\}$  is  $R$ -linearly convergent. Note also that the result in [7] is not valid for the constraints (5), that have a lower and upper bounds. Hence, Corollary 1 provides an improvement of the result in [7] for the QP case.  $\square$

Define  $\mathbf{W} = \mathbf{L}H^{-1}\mathbf{L}^T$ ,  $W \in \mathbb{R}^{1 \times 1}$ , with  $\mathbf{l} = l_1 + \dots + l_m$ . Since  $\mathbf{L}$  is a full row rank matrix, one has  $\mathbf{W} \succ 0$ . Denote  $\mathbf{d}_1, \dots, \mathbf{d}_l$  as the eigenvalues of  $\mathbf{W}$  with

$$\mathbf{d}_1 \geq \mathbf{d}_2 \geq \dots \geq \mathbf{d}_l > 0$$

We have the following result.

**Corollary 2:** The optimal penalty parameter  $\rho$  is given by

$$\rho^* = \frac{1}{\sqrt{\mathbf{d}_1 \mathbf{d}_l}} \quad (71)$$

**Proof:** It is omitted here, since it follows the same lines as the proof of Theorem 2.  $\square$

**Remark 6:** For the QP problem, the result in Corollary 2 is the same as in Theorem 4 of [7]. Hence for the QP problem, this paper provides an alternative proof to show that one has the optimal convergence rate with  $\rho^*$  given in (71). Similar to the proof of Theorem 2, it can be shown that the convergence rate is  $\lambda^* = \frac{\sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_l}}}{1 + \sqrt{\frac{\mathbf{d}_1}{\mathbf{d}_l}}}$ . Since  $\mathbf{d}_1 \geq \mathbf{d}_l$ , one has  $\lambda^* \geq \frac{1}{2}$ .  $\square$

**Remark 7:** If  $\mathbf{L}$  is not a full row rank matrix, following [7], a good heuristic choice for  $\rho$  is  $\rho = \frac{1}{\sqrt{\mathbf{d}_1 \mathbf{d}_l}}$ , where  $\mathbf{d}_l$  is the smallest nonzero eigenvalue of  $\mathbf{W}$ .  $\square$

### B. Self-Adaptive Penalty Parameters

In this section we consider the case when we have a vector of time-varying penalty parameters  $\rho_1(k), \rho_2(k), \dots, \rho_m(k)$ . As in Section IV-B, to distinguish the fixed penalty parameter case and the time-varying parameters case, we will employ  $\gamma_i(k)$  instead of  $\rho_i(k), \forall i = \overline{1, m}$ .

Using similar arguments as in Section V-A, it can be shown that the  $\mathbf{z}$ -,  $\mathbf{y}$ -updates at iteration  $k-1$ , and the  $x$ -,  $\mathbf{v}$ -updates at iteration  $k$  are given by

$$z_i(k) = \begin{cases} v_i(k), & \text{if } v_i(k)^T v_i(k) \leq 1, \\ \frac{v_i(k)}{\sqrt{v_i(k)^T v_i(k)}}, & \text{otherwise} \end{cases}, \forall i = \overline{1, m}, \quad (72)$$

$$y_i(k) = \gamma_i(k-1) (v_i(k) - z_i(k)), \forall i = \overline{1, m}, \quad (73)$$

$$\begin{aligned} x(k+1) = & (H + \sum_{i=1}^m \gamma_i(k) Q_i)^{-1} \sum_{i=1}^m (\gamma_i(k) L_i^T z_i(k) - L_i^T y_i(k)) \\ & - (H + \sum_{i=1}^m \gamma_i(k) Q_i)^{-1} (f + \sum_{i=1}^m \gamma_i(k) Q_i b_i) \end{aligned} \quad (74)$$

$$v_i(k+1) = L_i x(k+1) + \frac{1}{\gamma_i(k)} y_i(k) + L_i b_i, \forall i = \overline{1, m}, \quad (75)$$

Using (72), one obtains,  $\forall i = \overline{1, m}$

$$z_i(k) = \alpha_i(k) v_i(k) \quad (76)$$

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

$$\alpha_i(k) = \begin{cases} 1, & \text{if } v_i(k)^T v_i(k) \leq 1, \\ \frac{1}{\sqrt{v_i(k)^T v_i(k)}}, & \text{otherwise} \end{cases} \quad (77)$$

Substituting (77) into (74), one gets

$$y_i(k) = \gamma_i(k-1) (1 - \alpha_i(k)) v_i(k), \forall i = \overline{1, m} \quad (78)$$

By substituting (74), (76), (78) into (75), one obtains,  $\forall i = \overline{1, m}$

$$\begin{aligned} v_i(k+1) = & \sum_{j=1}^m \left( \alpha_j(k) - \frac{\gamma_j(k-1)}{\gamma_j(k)} (1 - \alpha_j(k)) \right) R_{ij} v_j(k) \\ & + \frac{\gamma_i(k-1)}{\gamma_i(k)} (1 - \alpha_i(k)) v_i(k) + L_i b_i \\ & - L_i (H + \sum_{j=1}^m \gamma_j(k) Q_j)^{-1} (f + \sum_{j=1}^m \gamma_j(k) Q_j b_j) \end{aligned} \quad (79)$$

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

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

Similar to Section IV-B, consider first the case  $\alpha_i(k) < 1, \forall i = \overline{1, m}$ . In this case  $\alpha_i(k) = \frac{1}{\sqrt{v_i(k)^T v_i(k)}}$ . The parameters  $\gamma_i(k)$  are chosen to satisfy the following equation,  $\forall i = \overline{1, m}$

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

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

$$\begin{aligned} \gamma_i(k) & = \frac{1 - \alpha_i(k)}{\alpha_i(k)} \gamma_i(k-1) \\ & = \left( \sqrt{v_i(k)^T v_i(k)} - 1 \right) \gamma_i(k-1) \end{aligned} \quad (80)$$

By substituting (80) into (74), one gets

$$x(k+1) = -(H + \sum_{j=1}^m \gamma_j(k) Q_j)^{-1} (f + \sum_{j=1}^m \gamma_j(k) Q_j b_j) \quad (81)$$

By substituting (80), (81) into (79), one obtains,  $\forall i = \overline{1, m}$

$$v_i(k+1) = \frac{v_i(k)}{\sqrt{v_i(k)^T v_i(k)}} + L_i(x(k+1) + b_i) \quad (82)$$

With a slight abuse of notation, denote

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

Similar to Section IV-B, we use the Gauss-Seidel method to find the fixed point  $\gamma^*, x^*, \mathbf{v}^*$  of the scheme (80), (81), (82). The main advantage of this method is that we also cover the case  $\alpha_i = 1$ . Using the Gauss-Seidel method,  $v_i(k+1)$  is updated as a solution of the following equation,  $\forall i = \overline{1, m}$

$$v_i = \frac{v_i}{\sqrt{v_i^T v_i}} + L_i(x(k+1) + b_i)$$

Using Proposition 2, one obtains,  $\forall i = \overline{1, m}$

$$v_i(k+1) = \left(1 + \frac{1}{\sqrt{(x(k+1)+b_i)^T Q_i (x(k+1)+b_i)}}\right) L_i(x(k+1) + b_i) \quad (83)$$

By substituting (83) into (80), one gets,  $\forall i = \overline{1, m}$

$$\gamma_i(k+1) = \sqrt{(x(k+1) + b_i)^T Q_i (x(k+1) + b_i)} \gamma_i(k) \quad (84)$$

The iteration scheme (81), (84) requires only  $\gamma(k)$ ,  $\forall i = \overline{1, m}$  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 (81), (84) converges linearly to the fixed point  $x^*, \gamma^*$ . The speed of convergence is fast when  $\gamma$  is far from  $\gamma^*$ . It becomes slower when  $\gamma$  and  $\gamma^*$  are close. Hence, similar to Section IV-B, we use the scheme (81), (84) with a few number of iterations to obtain a rough approximation of  $\gamma^*$ . We then use the obtained value as the penalty parameters in Algorithm 1. With a slight abuse of notation, we use Algorithm 2 to denote this two stages processes.

---

**Algorithm 2:** Time-Varying Penalty Parameters - General Case

---

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

1:  $t \leftarrow 0$

2: **repeat**

3:  $x(t+1) \leftarrow$

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

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

5:  $\gamma_i(t+1) \leftarrow$

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

6:  $t \leftarrow t + 1$

7: **until**  $t = T_{max}$

8: Using Algorithm 1 with  $\rho_i \leftarrow \gamma_i(t), z_i(0) \leftarrow L_i(x_i(t) + b_i), u_i(0) \leftarrow L_i(x_i(t) + b_i), \forall i = \overline{1, m}$

---

The number of iterations required for Algorithm 2 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  $\gamma_i^*, \forall i = \overline{1, m}$  helps 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 1 with  $\rho_i = \rho^*, \forall i = \overline{1, m}$  when  $d_1 = d_2 = \dots = d_l$ .

As written in Section IV-C, our scheme can be considered as a new adaptive RB balancing method. The most notable feature on Algorithm 2 is that we provide 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 tuning parameters.

The downside of Algorithm 2 is Step 3, which requires to inverse the matrix  $(H + \sum_{i=1}^m \gamma_i(k) Q_i)$ , or equivalently to find the solution of the following linear system

$$(H + \sum_{i=1}^m \gamma_i(t) Q_i) x(t+1) = -(f + \sum_{i=1}^m \gamma_i(t) Q_i b_i) \quad (85)$$

The eigenvalues/eigenvectors decomposition trick in Section IV-B works only with  $m = 1$ . Solving (85), e.g., via Gaussian elimination can be computationally demanding, especially for a large  $n$ . In this paper, we use the conjugate gradient (CG) method [19] to solve (85). The CG method is an iterative method, that is well known to be effective for (85) since  $(H + \sum_{i=1}^m \rho_i(k) Q_i)$  is a positive definite matrix. The interested reader is referred to [19] for more details on the CG method.

## VI. APPLICATION TO ROBUST MODEL PREDICTIVE CONTROL

We apply the results developed in this paper to the optimization problem of the prediction dynamics MPC method [3], [16], [17]. With a slight abuse of notation, we use  $t$  as the time step in this section.

Consider the following uncertain and/or time-varying linear discrete-time system

$$\chi(k+1) = \mathcal{A}(t)\chi(t) + \mathcal{B}(t)\nu(t) \quad (86)$$

where  $\chi \in \mathbb{R}^{n_x}$  is the measurable state,  $\nu \in \mathbb{R}^{n_\nu}$  is the control input. The matrices  $\mathcal{A}(t) \in \mathbb{R}^{n_x \times n_x}, \mathcal{B}(t) \in \mathbb{R}^{n_x \times n_\nu}$  satisfy

$$\mathcal{A}(t) = \sum_{i=1}^s \delta_i(t) \mathcal{A}_i, \mathcal{B}(t) = \sum_{i=1}^s \delta_i(t) \mathcal{B}_i \quad (87)$$

where  $\mathcal{A}_i \in \mathbb{R}^{n_x \times n_x}, \mathcal{B}_i \in \mathbb{R}^{n_x \times n_\nu}$  are known matrices.  $\delta(t) = [\delta_1(t) \ \dots \ \delta_s(t)]^T$  is a vector of parametric uncertainties such that

$$\sum_{i=1}^s \delta_i(t) = 1, \delta_i(t) \geq 0, \forall i = \overline{1, s} \quad (88)$$

The state  $\chi(t)$  and input  $\nu(t)$  are subject to the constraints

$$-\mathbf{1}_{n_c} \leq F_\chi \chi + F_\nu \nu \leq \mathbf{1}_{n_c} \quad (89)$$

where  $F_\chi \in \mathbb{R}^{n_c \times n_x}, F_\nu \in \mathbb{R}^{n_c \times n_\nu}$  are constant matrices.

The objective is to design a control law  $\nu(t) = \mathcal{U}(\chi(t))$  such that the controlled system

$$\chi(t+1) = \mathcal{A}(t)\chi(t) + \mathcal{B}(t)\mathcal{U}(\chi(t))$$

fulfills the state and input constraints (89) despite the uncertainties. Furthermore,  $\mathcal{U}(\chi(t))$  should solve the following minimax problem

$$\min_{\nu(t), \nu(t+1), \dots} \max_{\delta(t)} \left\{ \sum_{j=0}^{\infty} \|\chi(t+j)\|_{\mathcal{Q}}^2 + \|\nu(t+j)\|_{\mathcal{R}}^2 \right\} \quad (90)$$

where  $\chi(t+j), \nu(t+j), \forall j = 0, 1, \dots$  are the predicted states and the predicted inputs from time  $t$ .  $\mathcal{Q} \in \mathbb{S}^{n_x}, \mathcal{R} \in \mathbb{S}_+^{n_\nu}$  are weighting matrices.

Without the constraints (89), it is well known that (90) is a linear quadratic regulator problem. The solution is the linear state feedback controller

$$\nu(t) = K_0\chi(t) \quad (91)$$

where the gain  $K_0 \in \mathbb{R}^{n_\nu \times n_x}$  can be found by solving a semi-definite program (SDP).

In the presence of (89), the problem (90) is intractable due to the need of guaranteeing (89) for the infinite number of constraints. A way to overcome this problem is to employ the following dynamic control law

$$\begin{cases} \nu(t) = K_0\chi(t) + (K - K_0)r(t) \\ r(t+1) = \sum_{i=1}^s \delta_i(t) (\mathcal{A}_i + \mathcal{B}_i K) r(t) \end{cases} \quad (92)$$

where  $r(t) \in \mathbb{R}^{n_x}$  is the controller state.  $K \in \mathbb{R}^{n_\nu \times n_x}$  is any gain, that robustly stabilizes (86). In general,  $K$  is selected as a low matrix gain to enlarge the domain of attraction.  $\delta_i, \forall i = \overline{1, s}$  in (92) are the same as in (88). As it will be shown later in this section,  $\delta_i$  are not required for the implementation for the prediction dynamics based MPC control law.

Using (86), (92), one obtains

$$\tilde{\chi}(t+1) = \tilde{\mathcal{A}}(t)\tilde{\chi}(t) \quad (93)$$

where  $\tilde{\chi}(t) = [\chi(t)^T \ r(t)^T]^T$ , and

$$\tilde{\mathcal{A}}(t) = \begin{bmatrix} \mathcal{A}(t) + \mathcal{B}(t)K_0 & \mathcal{B}(t)(K - K_0) \\ \mathbf{0}_{n_x \times n_x} & \mathcal{A}(t) + \mathcal{B}(t)K \end{bmatrix}$$

It is clear that  $\tilde{\mathcal{A}}(t) = \sum_{i=1}^s \delta_i(t)\tilde{\mathcal{A}}_i$  with

$$\tilde{\mathcal{A}}_i = \begin{bmatrix} \mathcal{A}_i + \mathcal{B}_i K_0 & \mathcal{B}_i(K - K_0) \\ \mathbf{0}_{n_x \times n_x} & \mathcal{A}_i + \mathcal{B}_i K \end{bmatrix}$$

Using (89), (92), one obtains

$$-\mathbf{1}_{n_c} \leq \tilde{F}_\chi \tilde{\chi} \leq \mathbf{1}_{n_c} \quad (94)$$

where  $\tilde{F}_\chi = [F_\chi + F_\nu K_0 \ F_\nu(K - K_0)]$ .

The prediction dynamics based MPC method consists of two stages: offline stage and online stage. The offline stage includes two steps: i) Estimate the domain of attraction of (93), (94); ii) Calculate the upper bound of the cost function (90).

In [17], it was show that we can employ the intersection of ellipsoids  $\mathcal{E}(P_i), i = \overline{1, s}$  for the domain of attraction of (93), (94). The matrices  $P_i \in \mathbb{S}_+^{2n_x}$  satisfy the following conditions

$$\begin{cases} \begin{bmatrix} P_j & \tilde{\mathcal{A}}_i P_i \\ P_i \tilde{\mathcal{A}}_i^T & P_i \end{bmatrix} \succeq 0, \forall i, \forall j = \overline{1, s} \\ 1 - \tilde{F}_{j,\chi} P_i \tilde{F}_{j,\chi}^T \geq 0, \forall j = \overline{1, n_c}, \exists i = \overline{1, s} \end{cases} \quad (95)$$

where  $\tilde{F}_{j,\chi}$  is the  $j$ -row of  $\tilde{F}_\chi, j = \overline{1, n_x}$ .

The matrices  $P_i, \forall i = \overline{1, s}$  are optimized to enlarge the estimation of the domain of attraction of (93), (94). The interested reader can find more details in [17]. For the upper bound of the cost function, it was shown in [16], [17] that

$$\max_{\delta(t)} \left\{ \sum_{j=0}^{\infty} \|\chi(t+j)\|_{\mathcal{Q}}^2 + \|\nu(t+j)\|_{\mathcal{R}}^2 \right\} \leq \tilde{\chi}(t)^T \mathcal{H} \tilde{\chi}(t)$$

where the matrix  $\mathcal{H} = \text{diag}(H_\chi, H_r)$  satisfies

$$\begin{bmatrix} \mathcal{H} - \tilde{\mathcal{Q}} - \tilde{\mathcal{R}} & \tilde{\mathcal{A}}_i^T \mathcal{H} \\ \mathcal{H} \tilde{\mathcal{A}}_i & \mathcal{H} \end{bmatrix} \succeq 0, \forall i = \overline{1, s} \quad (96)$$

with

$$\tilde{\mathcal{Q}} = \text{diag}(\mathcal{Q}, \mathbf{0}_{n_x}), \tilde{\mathcal{R}} = [K_0 \ K - K_0].$$

Once the matrices  $P_i, i = \overline{1, s}, H_r$  are computed, at the online stage for a given current state  $\chi(t)$ , consider the following optimization problem

$$\begin{cases} \min_r \{r^T H_r r\} \\ \text{s.t. } [\chi(t)^T \ r^T]^T P_i \begin{bmatrix} \chi(t) \\ r \end{bmatrix} \leq 1, \forall i = \overline{1, s} \end{cases} \quad (97)$$

It is clear that (97) is a QCQP problem, that can be reformulated as (1), (2). Denote  $r^*$  as the optimal solution of (97). The control signal at time  $t$  is computed as

$$\nu(t) = K_0\chi(t) + (K - K_0)r^* \quad (98)$$

Assuming feasibility of (97) at initial state  $\chi(0)$ , it was proved in [17] that the control law (97), (98) guarantees recursive feasibility and robust asymptotic stability.

## VII. NUMERICAL EXAMPLE

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

$$\begin{aligned} H &= \mathbf{I}_2, f = \begin{bmatrix} 17 \\ 15 \end{bmatrix}, \\ Q &= \begin{bmatrix} 0.5485 & -0.2492 \\ -0.2492 & 0.1441 \end{bmatrix}, b = \mathbf{0}_{2 \times 1} \end{aligned}$$

Using Theorem 2, one obtains  $\rho^* = 7.6737$ , Figure 1 presents the number of iterations of Algorithm 1 as a function of  $\rho$  with the following parameters  $z(0) = \mathbf{0}_{2 \times 1}, u(0) = \mathbf{0}_{2 \times 1}, \epsilon_p = 10^{-5}, \epsilon_\rho = 10^{-5}$ . Using figure 1, it can be observed that the worst-case optimal penalty parameter performs reasonably well.

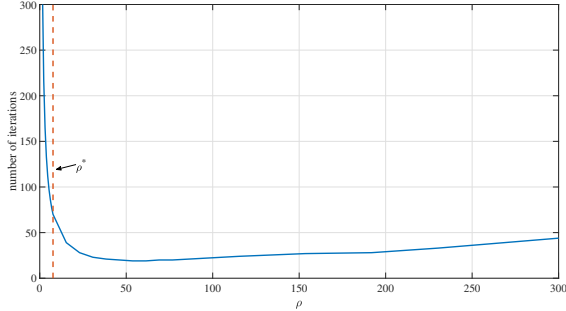


Fig. 1: Number of iterations for Algorithm 1 as a function of  $\rho$  for example 1.

Using Algorithm 2, Figure 2 shows the number of iterations (solid blue line) and  $\gamma$  (dashed red line) as functions of  $T_{max}$ . Using this figure, it can be observed that the number of iterations required for Algorithm 2 is drastically reduced compared to that for Algorithm 1.

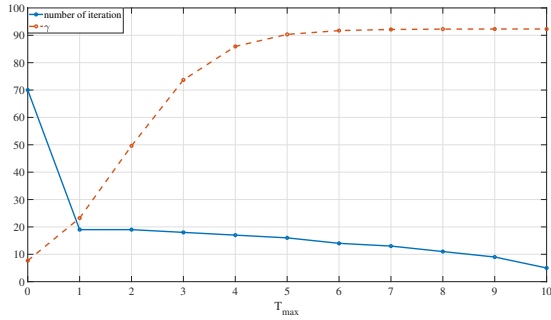


Fig. 2: Number of iterations for Algorithm 2 (solid blue), and  $\gamma$  (dashed red line) as functions of  $T_{max}$  for example 1.

### B. Example 2

This example is taken from [17] and is concerned with a classical angular positioning system. The system consists of a rotating antenna at the origin of the plane, driven by an electric motor. The control problem is to rotate the antenna by applying the input voltage to the motor so that it always points in the direction of a moving object in the plane. The motion of the antenna can be described by the discrete-time system (86) with

$$\mathcal{A}_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \mathcal{A}_2 = \begin{bmatrix} 1 & 0.1 \\ 0 & 0 \end{bmatrix}, \mathcal{B}_1 = \mathcal{B}_2 = \begin{bmatrix} 0 \\ 0.1574 \end{bmatrix} \quad (99)$$

The input and state constraint matrices (89) are:

$$F_\chi = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & 0 \end{bmatrix}, F_\nu = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (100)$$

The weighting matrices (90) are

$$\mathcal{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathcal{R} = 2 \times 10^{-5}$$

The corresponding LQ gain matrix is  $K_0 = [-46.0650 \quad -7.7831]$ . The matrix  $K$  is chosen an LQ gain with the same  $Q$  and with  $R = 10$  leading to  $K = [-0.1479 \quad -1.4582]$ . The matrices  $P_1, P_2$  are

$$P_1 = \begin{bmatrix} 5.80 & -0.16 & 5.80 & -0.16 \\ -0.16 & 0.86 & -0.17 & 0.95 \\ 5.80 & -0.17 & 5.81 & -0.18 \\ -0.16 & 0.95 & -0.18 & 1.11 \\ 5.83 & -0.27 & 5.83 & -0.28 \\ -0.27 & 1.00 & -0.25 & 0.98 \\ 5.83 & -0.25 & 5.83 & -0.26 \\ -0.28 & 0.98 & -0.26 & 1.03 \end{bmatrix},$$

$$\text{and } H_r = 10^3 \begin{bmatrix} 2.52 & 0.32 \\ 0.32 & 4.19 \end{bmatrix}.$$

Since  $\mathbf{L}$  in (67) is not a full row rank matrix, the penalty parameter  $\rho^*$  for Algorithm 1 is chosen heuristically as in remark 7. As the result, one gets  $\rho^* = 7.5267$ .

For the initial condition  $\chi(0) = [-1 \quad 0.5]^T$ , Figure 3 presents the number of iterations for Algorithm 1 (dashed red), and for Algorithm 2 with  $T_{max} = 1$  (solid blue), and with  $T_{max} = 2$  (dash-dot yellow) as functions of time. The tolerances are  $\epsilon_d = 10^{-2}, \epsilon_p = 10^{-2}$ . It can be observed that for this example, comparing to Algorithm 1, the most significant reduction of the number of iterations for Algorithm 2 is with  $T_{max} = 1$ .

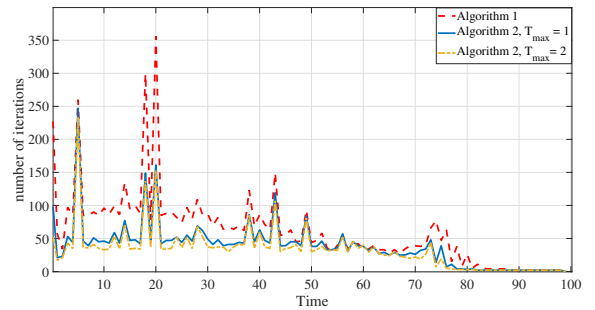


Fig. 3: Number of iterations for Algorithm 2 (dashed red), for Algorithm 2 with  $T_{max} = 1$  (solid blue), with  $T_{max} = 2$  (dash-dot yellow) as functions of time for example 2.

Finally, Figure 4 shows the state trajectory, the input trajectory and the  $\delta$  realization as functions of time.

## VIII. CONCLUSION

In this paper 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 shown 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 shown 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

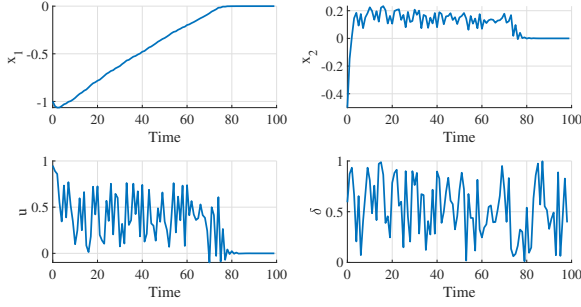


Fig. 4: State trajectory, input trajectory and  $\delta$  realization as functions of time for example 2.

the effectiveness of the proposed methods via two numerical examples.

## REFERENCES

- [1] Kendall E Atkinson and Weimin Han. *Elementary numerical analysis*. Wiley New York, 1993.
- [2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [3] Mark Cannon and Basil Kouvaritakis. Optimizing prediction dynamics for robust mpc. *IEEE Transactions on Automatic Control*, 50(11):1892–1897, 2005.
- [4] Raffaele Carli and Mariagrazia Dotoli. Distributed alternating direction method of multipliers for linearly constrained optimization over a network. *IEEE Control Systems Letters*, 4(1):247–252, 2019.
- [5] Jonathan Eckstein and Wang Yao. Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. *Pac. J. Optim.*, 11(4):619–644, 2015.
- [6] Pedro A Forero, Alfonso Cano, and Georgios B Giannakis. Consensus-based distributed linear support vector machines. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 35–46, 2010.
- [7] Euhanna Ghadimi, André Teixeira, Iman Shames, and Mikael Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2014.
- [8] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- [9] H Emre Güven, Alper Güngör, and Müjdat Cetin. An augmented lagrangian method for complex-valued compressed sar imaging. *IEEE Transactions on Computational Imaging*, 2(3):235–250, 2016.
- [10] Deren Han, Defeng Sun, and Liwei Zhang. Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Mathematics of Operations Research*, 43(2):622–637, 2018.
- [11] Bing-Sheng He, Hai Yang, and SL Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and applications*, 106:337–356, 2000.
- [12] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Collaborative sparse regression for hyperspectral unmixing. *IEEE Transactions on geoscience and remote sensing*, 52(1):341–354, 2013.
- [13] Pablo Krupa, Rim Jaouani, Daniel Limon, and Teodoro Alamo. A sparse admm-based solver for linear mpc subject to terminal quadratic constraint. *arXiv preprint arXiv:2105.08419*, 2021.
- [14] Zhang Liu, Anders Hansson, and Lieven Vandenbergh. Nuclear norm system identification with missing inputs and outputs. *Systems & Control Letters*, 62(8):605–612, 2013.
- [15] Hoai-Nam Nguyen. Constrained control of uncertain, time-varying, discrete-time systems. *Lecture Notes in Control and Information Sciences*, 451:17, 2014.
- [16] Hoai-Nam Nguyen. Optimizing prediction dynamics with saturated inputs for robust model predictive control. *IEEE Transactions on Automatic Control*, 66(1):383–390, 2020.
- [17] Hoai-Nam Nguyen. Improved prediction dynamics for robust mpc. *IEEE Transactions on Automatic Control*, 2022.
- [18] Hoai-Nam Nguyen. Efficient min–max mpc: Achieving a large domain of attraction with short horizon. *International Journal of Robust and Nonlinear Control*, 2023.
- [19] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.
- [20] Arvind U Raghunathan and Stefano Di Cairano. Alternating direction method of multipliers for strictly convex quadratic programs: Optimal parameter selection. In *2014 American Control Conference*, pages 4324–4329. IEEE, 2014.
- [21] Vincent Q Vu, Juhee Cho, Jing Lei, and Karl Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. *Advances in neural information processing systems*, 26, 2013.
- [22] Bo Wahlberg, Stephen Boyd, Mariette Annergren, and Yang Wang. An admm algorithm for a class of total variation regularized estimation problems. *IFAC Proceedings Volumes*, 45(16):83–88, 2012.
- [23] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.
- [24] Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3–4):203–230, 2010.
- [25] Brendt Wohlberg. Admm penalty parameter selection by residual balancing. *arXiv preprint arXiv:1704.06209*, 2017.

## APPENDIX

**A. Proof of Theorem 1:** To prove (37) we will show that

$$\|z(k+1) - z^*\|_2^2 + \|u(k+1) - u^*\|_2^2 \leq \|v(k+1) - v^*\|_2^2 \quad (101)$$

$$\|v(k+1) - v^*\|_2^2 \leq \lambda (\|z(k) - z^*\|_2^2 + \|u(k) - u^*\|_2^2) \quad (102)$$

Using (31), and recall that  $y(k) = \rho u(k)$ , one obtains

$$v(k+1) = z(k+1) + u(k+1) \quad (103)$$

For the fixed point, one has

$$v^* = z^* + u^* \quad (104)$$

Combining (103), (104), one obtains

$$(v(k+1) - v^*) = (z(k+1) - z^*) + (u(k+1) - u^*)$$

It follows that

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

Hence to prove (101), it suffices to show that

$$(u(k+1) - u^*)^T (z(k+1) - z^*) \geq 0 \quad (105)$$

Recall that  $z(k+1)$  is the optimal solution of (28). Using the KKT optimality condition for (28), one has,  $\forall z : z^T z \leq 1$

$$\left( z(k+1) - Lx(k+1) - \frac{1}{\rho} y(k) - Lb \right)^T (z(k+1) - z) \leq 0$$

thus, using (29)

$$(z(k+1) - v(k+1))^T (z(k+1) - z) \leq 0$$

Using (31), one gets

$$u(k+1)^T (z(k+1) - z) \leq 0 \quad (106)$$

Equation (106) is valid  $\forall z : z^T z \leq 1$ . Since  $(z^*)^T z^* \leq 1$ , one obtains

$$u(k+1)^T(z(k+1) - z^*) \geq 0 \quad (107)$$

Using similar arguments, and since  $z^*$  is the optimal solution of (28) for  $x^*, y^*, v^*$ , one gets

$$(u^*)^T(z^* - z(k+1)) \geq 0 \quad (108)$$

Combining (107), (108), one obtains

$$u(k+1)^T(z(k+1) - z^*) + (u^*)^T(z^* - z(k+1)) \geq 0$$

It follows that (105) holds. Consequently, (101) holds. It remains to prove (102). One has, for the fixed point  $(v^*, z^*, u^*)$  of (34)

$$v^* = Rz^* + (I - R)u^* + L(H + \rho Q)^{-1}(Hb - f) \quad (109)$$

Combining (34), (109), one obtains,

$$\|v(k+1) - v^*\|_2^2 = \|R(z(k) - z^*) + (I - R)(u(k) - u^*)\|_2^2$$

or, equivalently

$$\begin{aligned} \|v(k+1) - v^*\|_2^2 = & (z(k) - z^*)^T R^2 (z(k) - z^*) \\ & + (u(k) - u^*)^T (I - R)^2 (u(k) - u^*) \\ & + 2(z(k) - z^*)^T (R - R^2)(u(k) - u^*) \end{aligned} \quad (110)$$

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

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

It follows that

$$\begin{aligned} & (z(k) - z^*)^T (R - R^2)(z(k) - z^*) + \\ & (u(k) - u^*)^T (R - R^2)(u(k) - u^*) \\ & \geq 2(z(k) - z^*)^T (R - R^2)(u(k) - u^*) \end{aligned} \quad (111)$$

Combining (110), (111), one gets

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

or, equivalently

$$\begin{aligned} \|v(k+1) - v^*\|_2^2 \leq & (z(k) - z^*)^T R (z(k) - z^*) \\ & + (u(k) - u^*)^T (I - R) (u(k) - u^*) \end{aligned} \quad (112)$$

Using (36), one has

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

Therefore, using (112), one obtains (102). The proof is complete.  $\square$

**B. Proof of Theorem 2:** Using (38), we decompose  $W$  as  $W = E^T D E$ , where  $E \in \mathbb{R}^{l \times l}$  with  $E^T E = I$ , and  $D =$

$\text{diag}(d_1, \dots, d_l)$ . By using the matrix inversion lemma for  $R$  in (35) with  $Q = L^T L$ , one has

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

or equivalently

$$R = E^T \text{diag} \left( \frac{\rho d_1}{\rho d_1 + 1}, \dots, \frac{\rho d_l}{\rho d_l + 1} \right) E \quad (113)$$

Hence  $\frac{\rho d_1}{\rho d_1 + 1}, \dots, \frac{\rho d_l}{\rho d_l + 1}$  are the eigenvalues of  $R$ . It follows that  $1 - \frac{\rho d_1}{\rho d_1 + 1}, \dots, 1 - \frac{\rho d_l}{\rho d_l + 1}$ , or equivalently  $\frac{1}{\rho d_1 + 1}, \dots, \frac{1}{\rho d_l + 1}$  are the eigenvalues of  $(I - R)$ . Using (38), one has

$$\begin{cases} \frac{\rho d_1}{\rho d_1 + 1} \geq \frac{\rho d_2}{\rho d_2 + 1} \geq \dots \geq \frac{\rho d_l}{\rho d_l + 1} \\ \frac{1}{\rho d_1 + 1} \leq \frac{1}{\rho d_2 + 1} \leq \dots \leq \frac{1}{\rho d_l + 1} \end{cases}$$

Therefore  $\frac{\rho d_1}{\rho d_1 + 1}$  and  $\frac{1}{\rho d_1 + 1}$  are, respectively, the maximal eigenvalue of  $R$ , and of  $(I - R)$ . So our problem becomes the problem of selecting  $\rho$  to minimize the maximum between  $\frac{\rho d_1}{\rho d_1 + 1}$  and  $\frac{1}{\rho d_1 + 1}$ .

Note that the function  $\frac{\rho d_1}{\rho d_1 + 1} = \frac{d_1}{d_1 + \frac{1}{\rho}}$  is monotonically increasing  $\forall \rho > 0$ . On the other hand, the function  $\frac{1}{\rho d_1 + 1}$  is monotonically decreasing. With  $\rho = \rho^* = \frac{1}{\sqrt{d_1 d_l}}$ , one has

$$\frac{\rho d_1}{\rho d_1 + 1} = \frac{1}{\rho d_1 + 1} = \frac{\sqrt{\frac{d_1}{d_l}}}{1 + \sqrt{\frac{d_1}{d_l}}} \quad (114)$$

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

$$\max \left\{ \frac{\rho d_1}{\rho d_1 + 1}, \frac{1}{\rho d_1 + 1} \right\} = \frac{\rho d_1}{\rho d_1 + 1} > \frac{\sqrt{\frac{d_1}{d_l}}}{1 + \sqrt{\frac{d_1}{d_l}}}$$

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

$$\max \left\{ \frac{\rho d_1}{\rho d_1 + 1}, \frac{1}{\rho d_1 + 1} \right\} = \frac{1}{\rho d_1 + 1} > \frac{\sqrt{\frac{d_1}{d_l}}}{1 + \sqrt{\frac{d_1}{d_l}}}$$

It follows that  $\rho = \rho^*$  is the optimal value. The proof is complete.  $\square$

**C. Proof of Theorem 3:** Using (55), it is clear that if  $\gamma(0) > 0$ , then  $\gamma(k) \geq 0, \forall k \geq 1$ . Consider the following function

$$\begin{aligned} V(\gamma(k)) = & -\frac{1}{2}(f + \gamma(k)Qb)^T (H + \gamma(k)Q)^{-1} (f + \gamma(k)Qb) \\ & - \frac{1}{2}\gamma(k) + \frac{1}{2}\gamma(k)b^T Qb \end{aligned} \quad (115)$$

Using (11),  $V(\gamma(k))$  is the Lagrange dual function of (20), (21), where  $\gamma$  is the dual variable, i.e.,  $\gamma = \theta$ . Recall that  $V(\gamma)$  is concave and bounded from above. One has

$$\begin{aligned} & \frac{dV(\gamma(k))}{d\gamma(k)} \\ &= \frac{1}{2}(f + \gamma(k)Qb)^T Q \gamma (f + \gamma(k)Qb) \\ & \quad - b^T Q (H + \gamma(k)Q)^{-1} (f + \gamma(k)Qb) - \frac{1}{2} + \frac{1}{2}b^T Qb \end{aligned} \quad (116)$$

with  $Q_\gamma = (H + \gamma(k)Q)^{-1}Q(H + \gamma(k)Q)^{-1}$ . Thus, using (50)

$$\frac{dV(\gamma(k))}{d\gamma(k)} = \frac{1}{2} ((x(k+1) + b)^T Q(x(k+1) + b) - 1) \quad (117)$$

Using (55), (117), one has

- 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 (55) one has  $\gamma(k+1) > \gamma(k)$ . As a consequence,  $V(\gamma(k+1)) > V(\gamma(k))$ .
- 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 (55) 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 (55) until  $\gamma(k)$  reaches to the fixed point  $\gamma^*$ . Using (55),  $\gamma^*$  satisfies the following equation

$$\gamma^* = h(\gamma^*) \quad (118)$$

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

Equation (118) can have at most two solutions: i)  $\gamma^*$  is such that  $(x^* + b)^T Q(x^* + b) = 1$ ; ii)  $\gamma^* = 0$ . 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

$$\left. \frac{dh}{d\gamma} \right|_{\gamma=0} = \sqrt{(-H^{-1}f + b)^T Q(-H^{-1}f + b)} \quad (119)$$

For any small enough  $\gamma(0)$ ,  $\lim_{k \rightarrow \infty} \gamma(k) \rightarrow 0$  if and only if

$\left. \frac{dh}{d\gamma} \right|_{\gamma=0} < 1$ . It follows that

$$(-H^{-1}f + b)^T Q(-H^{-1}f + b) < 1 \quad (120)$$

Using (117), one has

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

Hence  $\left. \frac{dV(\gamma)}{d\gamma} \right|_{\gamma=0} > 0$ . Consequently the KKT condition (12) is satisfied. In other words,  $\gamma^* = 0$  is the optimal solution. The proof is complete.  $\square$

**D. Proof of Proposition 3:** One has

$$\begin{aligned} \frac{dh}{d\gamma} &= \frac{\sqrt{(x(\gamma) + b)^T Q(x(\gamma) + b)}}{\frac{\gamma x(\gamma) + b)^T Q(H + \gamma Q)^{-1} Q(x(\gamma) + b)}{\sqrt{(x(\gamma) + b)^T Q(x(\gamma) + b)}} \\ &= \frac{(x(\gamma) + b)^T (Q - \gamma Q(H + \gamma Q)^{-1} Q)(x(\gamma) + b)}{\sqrt{(x(\gamma) + b)^T Q(x(\gamma) + b)}} \end{aligned} \quad (121)$$

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

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

$$\begin{aligned} \gamma(2) &= h(\gamma(1)) > h(\gamma(0)) = \gamma(1), \\ &\vdots \\ \gamma(k+1) &= h(\gamma(k)) > h(\gamma(k-1)) = \gamma(k) \end{aligned}$$

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

$$\begin{aligned} \gamma(2) &= h(\gamma(1)) < h(\gamma(0)) = \gamma(1), \\ &\vdots \\ \gamma(k+1) &= h(\gamma(k)) < h(\gamma(k-1)) = \gamma(k) \end{aligned}$$

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



**Hoai-Nam Nguyen** received the engineering degree from Bauman, Moscow State Technical University, Russia in 2009, and the Ph.D. degree from École Supérieure d'Électricité, Supelec, Gif-sur-Yvette, France in 2012. After two years of postdoc, he spent 7 years in industry at IFP Energies Nouvelles as a research engineer. Since September, 2021, he works at Telecom-SudParis and Institut Polytechnique de Paris as an associate professor. His main research interests are optimization based control, sensor fusion, embedded systems, autonomous systems.