Complexity Certification of the Fast Alternating Minimization Algorithm for Linear Model Predictive Control

Ye Pu, Melanie N. Zeilinger and Colin N. Jones

Abstract

In this paper, the fast alternating minimization algorithm (FAMA) is proposed to solve model predictive control (MPC) problems with polytopic and second-order cone constraints. We extend previous theoretical results of FAMA to a more general case, where convex constraints are allowed to be imposed on the strongly convex objective and all convergence properties of FAMA are still preserved. Two splitting strategies for MPC problems are presented. Both of them satisfy the assumptions of FAMA and result in efficient implementations by reducing each iteration of FAMA to simple operations. We derive computational complexity certificates for both splitting strategies, by providing bounds on the number of iterations for both primal and dual variables, which are of particular relevance in the context of real-time MPC to bound the required online computation time. For MPC problems with polyhedral and ellipsoidal constraints, an off-line preconditioning method is presented to further improve the convergence speed of FAMA by reducing the complexity bound and enlarging the step-size of the algorithm. Finally, we demonstrate the performance of FAMA compared to other splitting methods using a quadrotor example.

I. Introduction

FAST NUMERICAL SOLVERS for model predictive control (MPC) have attracted significant research attention due to the increasing interest in applying MPC to problems with fast dynamics and the rapidly developing computational power of embedded systems.

Various on-line optimization methods for solving MPC problems have been proposed aiming at improving the computation time or at approximating the optimum with a sub-optimal but stabilizing solution. The fast gradient method introduced in [21] has been employed to solve MPC problems with box constraints on inputs in [24]. In [27] and [13], efficient implementations of interior-point methods have been studied. Accelerated gradient methods with dual decomposition are investigated in [19] and in [16] in the context of distributed MPC. In [14] and [2], efficient active set methods for MPC have been discussed.

In this paper, we focus on first-order methods, see e.g. [21], [3] and [17], because they offer simple iteration schemes that only require information of the function value and the gradient, and have shown good performance for solving medium and large size problems with moderate accuracy requirements. An efficient first-order method has the following two properties: 1. Each sub-problem, i.e. the computation of the gradient at each iteration, can be solved efficiently; 2. The optimization problem is well-posed, i.e. well conditioned, as the conditioning (geometry) of the optimization problem has a strong impact on the convergence speed of the algorithm. In this paper, we investigate a sub-group of first-order methods, called splitting methods, and apply them to MPC problems. We will show how the two properties above can be achieved by splitting the problem in a specific way and by off-line preconditioning the problem.

Splitting methods, which are also known as alternating direction methods, offer a powerful tool for general mathematical programming and optimization, see e.g. [17], [7] and [10]. Their efficiency results from splitting a complex convex minimization problem into simple sub-problems and solving them in an alternating manner. For a problem with multiple objectives, the main strategy is not to compute the descent direction of the sum of several objectives, but to take a combination of the descent directions of each objective. This can significantly reduce computation time, in particular when the objectives have different properties, for instance one being a quadratic function, one an l_1-norm and one involving indicator functions, which originate from constraints.

A variety of different spitting methods exist, requiring different assumptions on the problem setup, while exhibiting different properties, see e.g. [17] and [10] for an overview. In practice, splitting methods have shown good performance for solving complex problems in many fields, e.g. signal processing and machine learning. We focus on using these methods to solve control problems. The alternating direction method of multipliers (ADMM), as one of the most well-known splitting methods, was shown to solve optimal control problems both rapidly and robustly in [22]. However, ADMM also has its drawbacks.

Y. Pu and C.N. Jones are with the Automatic Control Lab, École Polytechnique Fédérale de Lausanne, EPFL-STI-IGM-LA Station 9 CH-1015 Lausanne, Switzerland, e-mail: {y.pu,colin.jones}@epfl.ch.

M.N. Zeilinger is with the Department of Electrical Engineering and Computer Sciences, UC Berkeley, 307 Cory Hall Berkeley, CA 94720-1770, USA and the Department of Empirical Inference, Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany, e-mail: mzeilinger@berkeley.edu.

This work has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ ERC Grant Agreement n. 307608. The research of M. N. Zeilinger has received funding from the EU FP7 under grant agreement no. PIOF-GA-2011-301436-"COGENT".

Firstly, ADMM only provides the convergence rate $O(\frac{1}{k})$ under the assumption that both objectives are convex, see [18]. An accelerated variant of ADMM, the fast alternating direction method of multipliers (FADMM) presented in [17], improves the practical convergence speed, but maintains the same theoretical convergence rate as ADMM. Secondly, ADMM and FADMM do not have efficient step-size tuning rules, whereas the step-size has been observed to be critical for their performance. In practice, the step-size is usually tuned by trial and error. The third aspect is that for ADMM and FADMM no theoretical complexity bound on the number of iterations is known. The complexity bound plays an important role in the context of real-time MPC, since it allows one to derive a certificate on the number of iterations to achieve a given accuracy, and thereby offers a prediction of the worst-case sub-optimality of the solution after running the algorithm for a fixed number of iterations.

In this paper, we propose the use of a slitting method called the fast (accelerated) alternating minimization algorithm (FAMA) introduced in [26] and [17] for solving MPC problems, offering superior theoretical properties while providing similar or even better performance. Compared to ADMM, FAMA requires stronger assumptions on the objectives of the optimization problem, which can, however, be satisfied by standard MPC problem formulations with polytopic and second-order cone constraints. In return, FAMA offers a faster convergence rate of $O(\frac{1}{k^2})$ and provides theoretical complexity bounds on the required number of iterations. The main contributions of this work are:

- Second-order cone constraints: In this work, we consider both polytopic and second-order cone constraints, which allows for solving a broad range of MPC problems, e.g. including ellipsoidal constraints and chance constraints.
- Extension of theoretical results of FAMA: We show that all convergence properties of FAMA are preserved when imposing convex constraints on the strongly convex objective.
- Splitting strategies: We propose two splitting strategies for MPC problems that satisfy the assumptions of FAMA and reduce each iteration of FAMA to simple operations.
- Complexity bound: Complexity upper-bounds on the number of iterations to achieve a certain solution accuracy for both primal and dual variables are derived for both splitting strategies.
- Preconditioning: For MPC problems with polytopic and ellipsoidal constraints, we propose an off-line preconditioning method to further improve the convergence speed of FAMA. The method reduces the complexity bounds and enlarges the step-size of the algorithm by scaling the polytopic constraints and reshaping the ellipsoidal constraints.

All properties above are demonstrated for the simulation example of a quadroter.

II. Preliminaries

A. Notation

Let f be a strongly convex function. σ_f denotes the convexity modulus of f, i.e. $\langle p-q, x-y\rangle \geq \sigma_f \|x-y\|^2$, where $p \in \partial f(x)$, $q \in \partial f(y)$ and $\partial(\cdot)$ denotes the set of sub-gradients of the function at a given point. The operators $\max$, $\leq$ and $\geq$ are defined to work on vectors as well as scalars. For vectors, the operators are defined to be element-wise. Let C be a matrix. $\rho(C)$ denotes the largest eigenvalue of C^TC. For a positive definite matrix H, $\lambda_{min}(H)$ denotes the smallest eigenvalue of H. Let $\mathbb{K}$ be a convex set. The indicator function $I_{\mathbb{K}}(\sigma)$ on $\mathbb{K}$ is defined to be zero if $\sigma \in \mathbb{K}$ and infinity otherwise. Let $\mathbb{C}$ be a convex cone. The set $\mathbb{C}^\circ = \{w \mid v^Tw \leq 0,\ \forall v \in \mathbb{C}\}$ denotes the polar cone of $\mathbb{C}$. The set $\mathbb{C}^\star = \{w \mid v^Tw \geq 0,\ \forall v \in \mathbb{C}\}$ denotes the dual cone of $\mathbb{C}$. A cone $\mathbb{C}$ is called self-dual, if $\mathbb{C} = \mathbb{C}^\star$. Let $f : \Theta \to \Omega$ be a function. $L(f)$ denotes a Lipschitz constant of f. The conjugate function of f is defined as $f^\star(y) = \sup_x(y^Tx - f(x))$. It holds that $p \in \partial f(q) \Leftrightarrow q \in \partial f^\star(p)$. If f is a strongly convex function with the convexity modulus σ_f, then the gradient of the conjugate function of f has a Lipschitz constant $L(\nabla f^\star) = \sigma_f^{-1}$. We refer to [5] and [8] for details on the definitions and properties above.

B. Fast alternating minimization algorithm (FAMA)

In [26], an alternating direction method called the alternating minimization algorithm (AMA) is proposed. In [17], an accelerated variant of AMA, called the fast alternating minimization algorithm (FAMA) is discussed. AMA and FAMA solve Problem 2.1 and require Assumption 2.2 for convergence. FAMA is presented in Algorithm 1.

Problem 2.1:

$$\begin{aligned} \min_{v\in\mathbb{R}^{n_v},\, w\in\mathbb{R}^{n_w}} \quad & f(v)+g(w)\ , \\ \text{s.t.} \quad & Av+Bw=c\ . \end{aligned}$$

Assumption 2.2:

- f is a strongly convex function. σ_f denotes the convexity modulus of f.
- g is a convex function, not necessarily smooth.

Since in the simulation section, we will compare the performance of FAMA with ADMM and FADMM, we provide a brief comparison between FAMA and the popular technique ADMM as well as its accelerated variant FADMM (see Algorithm 8 in [17]) from a theoretical perspective, highlighting the key differences. Firstly, they require different convergence assumptions and have different convergence rates. ADMM and FADMM require the objectives to be convex functions, and under this

Algorithm 1 Fast alternating minimization algorithm (FAMA)

Require: Initialize $\alpha^0 = 1$, $\lambda^0 = \hat{\lambda}^1 = \lambda^{start} \in \mathbb{R}^{N_b}$, and $\tau < \sigma_f/\rho(A)$

for $k = 1, 2, \cdots$ **do**

1: $v^k = \text{argmin}_v \quad f(v) + \left\langle \hat{\lambda}^k, -Av \right\rangle$

2: $w^k = \text{argmin}_w \quad g(w) + \left\langle \hat{\lambda}^k, -Bw \right\rangle + \frac{\tau}{2}\|c - Av^k - Bw\|^2$

3: $\lambda^k = \hat{\lambda}^k + \tau(c - Av^k - Bw^k)$

4: $\alpha^{k+1} = (1 + \sqrt{4\alpha^{k^2} + 1})/2$

5: $\hat{\lambda}^{k+1} = \lambda^k + (\alpha^k - 1)(\lambda^k - \lambda^{k-1})/\alpha^{k+1}$

end for

Methods	Assumptions	Convergence rate	Complexity bound	Condition on step-size
ADMM	both obj convex	$O(1/k)$	no	no
FADMM	both obj convex	$O(1/k)$	no	no
FAMA	one obj strongly convex	$O(1/k^2)$	yes	yes

Table I: Summary of assumptions and properties of ADMM, FADMM and FAMA.

assumption both algorithms guarantee the same theoretical convergence rate $O(\frac{1}{k})$. If both objectives are strongly convex (see [12]), ADMM and FADMM provide a linear convergence rate. However, this assumption can hardly be satisfied by MPC problems, since MPC problems have constraints, which can be considered as indicator functions and are not strongly convex. FAMA requires one objective to be strongly convex and the other to be convex, which is stronger than the basic assumption of ADMM and FADMM, but in return, it achieves a faster convergence rate $O(\frac{1}{k^2})$. The second difference is that FAMA provides a complexity upper-bound on the number of iterations for a given accuracy (Section III-A), which allows for a real-time solution guarantee, i.e. a certificate that the problem can be solved in the given fixed amount of time. For ADMM and FADMM, it is not clear how to derive such a complexity bound. The third difference is that for ADMM and FADMM, the question of how to best tune the step-size still remains largely unclear and is usually done by trial and error. Theoretically, any positive step-size guarantees convergence, however, not any positive step-size practically results in good convergence performance. FAMA, in contrast, has a clear step-size rule, i.e. it requires the step -size to be smaller than the reciprocal of the Lipschitz constant of the gradient of the dual objectives. This condition simplifies the selection of the step-size, and together with the complexity bound allows for preconditioning the problem to speed up the algorithm, which will be discussed in Section VII. The differences between ADMM, FADMM and FAMA are summarized in Table I.

C. Fast iterative shrinkage-thresholding algorithm

As shown in [17], FAMA corresponds to applying the fast iterative shrinkage-thresholding algorithm (FISTA) in [3] to the dual problem. We will therefore first introduce the convergence results for FISTA in the following, which will allow us to derive an upper bound on the number of iterations for FAMA. FISTA solves problems of the following form

Problem 2.3:

$$\min \quad F(z) + G(z) \ , \quad z \in \mathbb{R}^{n_z} \ . \tag{1}$$

Note that Problem 2.3 is a special case of Problem 2.1, setting the matrices A and B equal to the identity matrix and c equal to a zero vector.

Assumption 2.4:

- F is a continuous convex function with Lipschitz continuous gradient $L(\nabla F)$

$$|\nabla F(z_1) - \nabla F(z_2)| \leq L(\nabla F)|z_1 - z_2, | \quad \forall z_1, z_2 \in \mathbb{R}^n \ .$$

- G is a convex function.

The conceptual idea of FISTA is to build a linearization and regularization of the differentiable function part in the objective at each iteration and to apply Nesterov's acceleration step to achieve $O(\frac{1}{k^2})$ convergence rate. Lemma 2.5 states the complexity upper-bound on the number of iterations for FISTA.

Lemma 2.5 ([3], Theorem 4.4): Let $\{z^k\}$ be generated by FISTA. If Assumption 2.4 holds, then for any $k \geq 1$ we have

$$J(z^k) - J(z^\star) \leq \frac{2L(\nabla F)|z^{start} - z^\star|^2}{(k+1)^2} \ , \tag{2}$$

where $J(z) = F(z) + G(z)$, and z^{start} and $z^\star$ denote the starting point and the optimizer of Problem 2.3, respectively.

III. Theoretical properties of FAMA

In this section, we present the theoretical properties of FAMA. We first derive the complexity upper-bound for FAMA under Assumption 2.2. Then we weaken Assumption 2.2 to allow for convex constraints to be imposed on the strongly convex objective f and prove that all theoretical properties are maintained. Finally, we derive the KKT conditions of Problem 2.1, which will be important for the computation of complexity bounds in Section VI.

A. Computational complexity upper-bounds for the dual sequences $\{\lambda^k\}$ generated by FAMA

The dual problem of Problem 2.1 is defined in Problem 3.1.

Problem 3.1:

$$\max_{\lambda\in\mathbb{R}^{n_c}} \mathbf{D}(\lambda) = \underbrace{-f^\star(A^T\lambda)}_{-F(\lambda)} + \underbrace{c^T\lambda - g^\star(B^T\lambda)}_{-G(\lambda)} \ ,$$

where $\mathbf{D}(\cdot)$ denotes the dual function and λ denotes the Lagrange multiplier.

Theorem 3.2: Let $\{\lambda^k\}$ be generated by FAMA in Algorithm 1. If Assumption 2.2 is satisfied, then for any $k \geq 1$ we have

$$\mathbf{D}(\lambda^\star) - \mathbf{D}(\lambda^k) \leq \frac{2\rho(A)|\lambda^{start} - \lambda^\star|^2}{\sigma_f(k+1)^2} \ , \tag{3}$$

where $\mathbf{D}(\cdot)$ is the dual function in Problem 3.1, and λ^{start} and $\lambda^\star$ denote the starting point and the optimizer, respectively.

Proof: Theorem 1 in [17] shows that applying FAMA to Problem 2.1 is equivalent to applying FISTA to Problem 3.1. $F(\lambda)$ and $G(\lambda)$ are both convex, since the conjugate functions and linear functions as well as their weighted sum are always convex (conjugate function is the point-wise supremum of a set of affine functions). By Assumption 2.4, we know that $f(x)$ is strongly convex with modulus σ_f. By the property of the conjugate function, a Lipschitz constant of $\nabla f^\star$ is given by

$$L(\nabla f^\star) = \sigma_f^{-1} \ .$$

This provides a Lipschitz constant of ∇F,

$$L(\nabla F(\lambda)) = \sigma_f^{-1} \cdot \rho(A) \ .$$

By Lemma 2.5, it follows that the sequence $\{\lambda_k\}$ generated by FAMA satisfies the complexity bound (3). ■

Remark 3.3: The step-size constraint of FAMA $\tau < \sigma_f/\rho(A)$ originates from the step-size constraint of FISTA, i.e. $\tau < \frac{1}{L(\nabla F)}$. For the case that the Lipschitz constant is not known or very small, the backtracking step-size rule in [3] can be applied. The idea is to update L at every iteration k as $L_k = \eta^{i_k} L_{k-1}$, where η is a positive constant and i^k is the smallest non-negative integer satisfying

$$f^\star(A^T\bar{\lambda}^k) \leq \Gamma_{\bar{L}^k}(\bar{\lambda}^k, \lambda^{k-1}) \ ,$$

where $\bar{\lambda}^k$ denotes the one-iteration solution of FAMA based on λ^{k-1}, the step-size is $\bar{L}^k = \eta^{i^k} L^{k-1}$, and

$$\Gamma_{\bar{L}^k}(y_1, y_2) := f^\star(A^T y_2) + \langle y_1 - y_2, A\nabla f^\star(A^T y_2)\rangle + \frac{1}{\bar{L}^k}|y_1 - y_2|^2 + g^\star(B^T y_1) \ .$$

B. Computational complexity upper-bound for the dual sequence $\{\lambda^k\}$ generated by FAMA with weakened assumption

For many problems, it is desirable to choose a splitting where a convex constraint is additionally imposed on the strongly convex objective f, for instance the second splitting strategy, which will be introduced in Problem 4.7. Such a splitting violates the first condition in Assumption 2.2, because the constraints can be considered as indicator functions in the cost, which are only weakly convex. In this section, we present a weakened assumption in Assumption 3.4 that allows for convex constraints on the strongly convex objective in Problem 2.1 and show that the convergence properties of the algorithm are maintained.

Assumption 3.4:

- f is a strongly convex function defined on a convex set $\mathbb{K}$. σ_f denotes the convexity modulus of f.
- g is a convex function, not necessarily smooth.

We define the dual function $\bar{f}$ of f as (4). It is similar to the conjugate function $f^\star$, but the domain of the function is given by the convex constraint. Before showing the convergence proof of FAMA under Assumption 3.4, we state a lemma showing Lipschitz continuity of $\nabla\bar{f}(\cdot)$.

$$\bar{f}(y) := \sup_{x\in\mathbb{K}} \ y^T x - f(x) \ = - \inf_{x\in\mathbb{K}} \ f(x) - y^T x \ . \tag{4}$$

Lemma 3.5: If the function f is strongly convex with the convexity modulus σ_f and the constraint $\mathbb{K}$ is a convex set, then the function $\bar{f}$ has a Lipschitz continuous gradient

$$\nabla\bar{f}(y) \ = \ -x^\star(y) \ , \tag{5}$$

with

$$x^\star(y) \;=\; \text{argmin}_{x\in\mathbb{K}}\, f(x) - y^T x \;\; . \tag{6}$$

For any y_1 and y_2, we have

$$\|\nabla \bar{f}(y_1) - \nabla \bar{f}(y_2)\| \leq L(\nabla \bar{f})\|y_1 - y_2\| \;\; , \tag{7}$$

with a Lipschitz constant $L(\nabla \bar{f}) = \frac{1}{\sigma_f}$.

Proof: The proof follows directly from Theorem 1 in [25]. ■

Theorem 3.6: Let $\{\lambda^k\}$ be generated by FAMA with modified Step 1

$$v^k = \text{argmin}_{v\in\mathbb{K}} \quad f(v) + \left\langle \hat{\lambda}^k, -Av \right\rangle \;\; . \tag{8}$$

If Assumption 3.4 holds, then for any $k \geq 1$ the sequence $\{\lambda_k\}$ satisfies the complexity bound in (3).

Proof: Since the definition of the function $\bar{f}(\cdot)$ corresponds to the definition of the conjugate function of $f(\cdot)$ with the function domain given by $\mathbb{K}$, then the property $p \in \partial f(q) \Leftrightarrow q \in \partial \bar{f}(p)$ is preserved, which implies that Theorem 1 in [17] still holds. Hence, applying FAMA to Problem 2.1 with convex constraint on f is equivalent to applying FISTA on the dual problem defined in (9). What we need to show is that the dual problem in (9) satisfies the convergence Assumption 2.4.

$$\begin{aligned}\max_{\lambda\in\mathbb{R}^{n_c}} \; \mathbf{D}(\lambda) &= -\sup_{v\in\mathbb{K}} \left\{\lambda^T A v - f(v)\right\} - \sup_{w} \left\{\lambda^T B w - g(w)\right\} + \lambda^T c \\ &= \underbrace{-\bar{f}(A^T\lambda)}_{-\bar{F}(\lambda)} \underbrace{- g^\star(B^T\lambda) + c^T\lambda}_{-G(\lambda)} \;\; .\end{aligned} \tag{9}$$

From Lemma 3.5, it follows that $\bar{F}(\lambda)$ has a Lipschitz continuous gradient $\nabla \bar{F}(\lambda) = A\nabla \bar{f}(A^T\lambda)$ with a Lipschitz constant $L(\nabla F(\lambda)) = \sigma_f^{-1} \cdot \rho(A)$. Since $\bar{F}(\lambda)$ and $G(\lambda)$ are both convex by definition, then the dual problem in (9) satisfies Assumption 2.4 and Lemma 2.5 implies the complexity bound in (3). ■

C. KKT conditions for Problem 2.1

In this section, we show the KKT conditions of Problem 2.1, which will be used in Section VI for estimating the value of $|\lambda^{start} - \lambda^\star|$ in the complexity bound in (3). The KKT conditions of Problem 2.1 are given by

(i) $Ax^\star + Bz^\star = c$,
(ii) $0 \in \partial f(x^\star) + A^T\lambda^\star$,
(iii) $0 \in \partial g(z^\star) + B^T\lambda^\star$.

Condition (i) represents the feasibility condition, and (ii) and (iii) represent optimality conditions. In this paper, we will apply FAMA to MPC problems with polytopic and second-order cone constraints and propose two splitting strategies. In both splitting strategies, the second objective g is considered to be a set of indicator functions on convex constraints given by convex cones. In order to compute the explicit form of Condition (iii), in this case, we introduce the subdifferential of the indicator function on a convex cone, which is defined as

$$\partial I_{\mathbb{C}}(x) = \begin{cases} \mathbb{N}_{\mathbb{C}}(x) & \text{if } x \in \mathbb{C} \\ \emptyset & \text{if } x \notin \mathbb{C} \;\; , \end{cases} \tag{10}$$

where $\mathbb{N}_{\mathbb{C}}(x)$ denotes the normal cone of $\mathbb{C}$ at x. The following proposition states that being in the normal cone $\mathbb{N}_{\mathbb{C}}(x)$ is equivalent to two conditions: x is in the polar cone of $\mathbb{C}$ and the complementary slackness condition holds.

Proposition 3.7 ([1], Proposition 2.51): Let $\mathbb{C}$ be a non-empty closed convex cone. Then, for any point $\bar{x} \in \mathbb{C}$, $x \in \mathbb{N}_{\mathbb{C}}(\bar{x})$ is equivalent to $x^T\bar{x} = 0$ and $x \in \mathbb{C}^\circ$, where $\mathbb{C}^\circ$ denotes the polar cone of $\mathbb{C}$.

IV. Fast Alternating Minimization Algorithm (FAMA) for MPC

In the following, we show how FAMA can be applied to MPC problems to achieve an efficient online implementation. We consider an MPC problem with linear dynamical system, state and input constraints in the form of polytopic and/or second-order cone constraints and quadratic stage and terminal costs. We present two splitting strategies.

1) Splitting Strategy 1: By eliminating all state variables and moving the constraints to the cost in the form of indicator functions, MPC problems of this class can be reformulated in the following form suitable for the application of FAMA, with one strongly and one weakly convex objective.

Problem 4.1:

$$\begin{aligned}\min_{\mathbf{u}} \quad & \underbrace{\mathbf{u}^T H \mathbf{u} + h^T \mathbf{u}}_{f(\mathbf{u})} + \underbrace{\sum_{i=1}^{M} I_{\mathbb{C}_i}(\sigma_i)}_{g(\sigma)} \\ s.t. \quad & C_i\mathbf{u} - c_i = \sigma_i, \quad i = 1, \cdots, M \;\; ,\end{aligned}$$

where $\mathbf{u} = [u_0^T, u_1^T, \cdots, u_{N-1}^T]^T \in \mathbb{R}^{N \cdot m}$ denotes the sequence of inputs over the control horizon N and $\sigma = [\sigma_1^T, \cdots, \sigma_M^T]^T \in \mathbb{R}^{N_\sigma}$ are auxiliary variables. $\mathbb{C}_i$ denote the constraints on the states and inputs. In this paper, it is assumed that $\mathbb{C}_i$ are given either by the non-negative orthant, i.e., $\mathbb{C}_i := \{v \mid v \geq 0\}$, or simple second-order cone constraints, i.e., $\mathbb{C}_i := \{[v_1, v_2]| \, |v_1|_2 \geq v_2\}$. Note that these definitions cover all polytopic and second-order cone constraints on $\mathbf{u}$ by involving the affine coupling $C_i\mathbf{u} - c_i = \sigma_i$. Both the non-negative orthant and the second-order cone are self-dual cones, a fact that will be used in the proof of Theorem 5.3.

Assumption 4.2: The matrix H is positive definite, i.e., $\lambda_{min}(H) > 0$.

Remark 4.3: Assumption 4.2 is satisfied, if a positive definite quadratic cost on the inputs is chosen in the MPC problem and the linear dynamical system is controllable.

Remark 4.4: If Assumption 4.2 holds, the first objective function $f(\mathbf{u})$ is strongly convex and the convexity modulus σ_f is given by the minimum eigenvalue of the matrix H, i.e., $\sigma_f = \lambda_{min}(H)$. Since the second objective $g(\sigma)$ is an indicator function of a convex cone, which is a convex function, then Problem 4.1 satisfies Assumption 2.2 required by FAMA.

Remark 4.5: We denote the current measured state by $\bar{x}$. The matrix H is independent of $\bar{x}$ and the matrix h is a linear function of $\bar{x}$.

Algorithm 2 Fast alternating minimization algorithm (FAMA) for Problem 4.1

Require: Initialize $\alpha^0 = 1$, $\alpha^1 = (1+\sqrt{5})/2$, $\lambda_i^0 = \hat{\lambda}_i^1 = \lambda_i^{start} \in \mathbb{R}^{N_\sigma}$, $\mathbf{u}^0 = \frac{1}{2}H^{-1}(\sum_{i=1}^M C_i^T\lambda_i^0 - h)$ and $\tau < \sigma_f/\rho(C) = \lambda_{min}(H)/\rho(C)$.

for $k = 1, 2, \cdots$ **do**

1: $\mathbf{u}^k = \text{argmin}_{\mathbf{u}} \quad \mathbf{u}^T H\mathbf{u} + h^T\mathbf{u} - \sum_{i=1}^M {\lambda_i^{k-1}}^T C_i\mathbf{u}$

2: $\hat{\mathbf{u}}^k = \mathbf{u}^k(\alpha^{k-1} + \alpha^k - 1)/\alpha^k - \mathbf{u}^{k-1}(\alpha^{k-1} - 1)/\alpha^k$

3: $\alpha^{k+1} = (1 + \sqrt{4\alpha^{k^2} + 1})/2$

for $i = 1, \cdots, M$ **do**

4: $\sigma_i^k = \mathbf{Pr}_{\mathbb{C}_i}(C_i\hat{\mathbf{u}}^k - \frac{1}{\tau}\hat{\lambda}_i^k - c_i)$

5: $\lambda_i^k = \hat{\lambda}_i^k + \tau(c_i - C_i\hat{\mathbf{u}}^k + \sigma_i^k)$

6: $\hat{\lambda}_i^{k+1} = \lambda_i^k + (\alpha^k - 1)(\lambda_i^k - \lambda_i^{k-1})/\alpha^{k+1}$

end for

end for

We apply FAMA to MPC Problem 4.1 resulting in Algorithm 2, where $C := [C_1^T, \cdots, C_M^T]^T$. The advantage of the splitting strategy in Problem 4.1 is that the two objectives $f(\mathbf{u})$ and $g(\sigma)$ are very easy to minimize separately. The solution to the unconstrained minimization problem in Step 1 can be obtained analytically, i.e. $\mathbf{u}^k = \frac{1}{2}H^{-1}(\sum_{i=1}^M C_i^T\lambda_i^{k-1} - h)$, where the inverse H^{-1}, or an appropriate factorization, can be computed off-line. Step 4 involves basic projections onto the non-negative orthant and simplified second-order cone. We denote the projection operation as $\mathbf{Pr}_{\mathbb{C}}(\cdot)$. For the non-negative orthant, the projection is

$$\mathbf{Pr}_{\mathbb{C}}(v) = \max\{0, v\} \ . \tag{11}$$

For the second-order cone, the projection is

$$\mathbf{Pr}_{\mathbb{C}}([v_1, v_2]) = \begin{cases} [v_1, v_2] & \text{if } |v_1|_2 \leq v_2 \\ \frac{v_2 + |v_1|_2}{2|v_1|_2}[v_1, |v_1|_2] & \text{if } |v_1|_2 > v_2, \ v_1 \neq 0 \\ [0, 0] & \text{if } |v_1|_2 \leq -v_2 \ . \end{cases} \tag{12}$$

The projections in (11) and (12) are computationally cheap. They reduce to simply a clipping and a scaling operation.

Remark 4.6: Step 1 and 2 in Algorithm 2 are equivalent to $\hat{\mathbf{u}}^k = argmin \quad \mathbf{u}^T H\mathbf{u} + h^T\mathbf{u} - \sum_{i=1}^M {\hat{\lambda}_i^{k-1}}^T C_i\mathbf{u}$, which is the standard first step of FAMA. By splitting this step into two steps, we can represent $\mathbf{u}^k$ by a function of λ^k, i.e. $\mathbf{u}^k = \frac{1}{2}H^{-1}(C^T\lambda^k - h)$. This allows us to derive the primal complexity bound on $\mathbf{u}^k$ based on the dual complexity bound on the Lagrange multipliers in Section V.

2) Splitting Strategy 2: The second splitting strategy in Problem 4.7 maintains both the states and inputs as optimization variables and involves the dynamical constraint as a convex set on the first objective f. Note that this splitting violates the first condition in Assumption 2.2, since the convex constraint results in an indicator function that is convex but not strongly convex. Problem 4.7 however satisfies the weakened Assumption 3.4 by choosing a strictly convex cost on the states and inputs, i.e. $Q \succ 0$, on a convex domain. Convergence is then guaranteed by Theorem 3.6.

Problem 4.7:

$$\min_{\mathbf{z}} \quad \underbrace{\{\mathbf{z}^T Q\mathbf{z} + q^T\mathbf{z} \mid T\mathbf{z} = t\}}_{f(\mathbf{z})} + \underbrace{\sum_{i=1}^{M} I_{\mathbb{C}_i}(\sigma_i)}_{g(\sigma)}$$
$$s.t. \quad D_i\mathbf{z} - d_i = \sigma_i, \quad i = 1, \cdots, M \ .$$

where $\mathbf{z} = [\mathbf{x}; \mathbf{u}]$ contains the state and input sequences over the control horizon. The matrices T and t represent the dynamical constraint. We apply FAMA to MPC Problem 4.7 resulting in Algorithm 3, where $D := [D_1^T, \cdots, D_M^T]^T$.

Assumption 4.8: The matrix Q is positive definite, i.e., $\lambda_{min}(Q) > 0$.

Remark 4.9: Assumption 4.8 is satisfied, if the cost on the states and the inputs in the MPC problem are chosen to be positive definite quadratic functions.

Algorithm 3 Fast alternating minimization algorithm (FAMA) for Problem 4.7

Require: Initialize $\alpha^0 = 1$, $\alpha^1 = (1+\sqrt{5})/2$, $\lambda_i^0 = \hat{\lambda}_i^1 = \lambda_i^{start} \in \mathbb{R}^{N_\sigma}$, $\mathbf{z}^0 = \text{argmin}_{T\mathbf{z}=t} \quad \mathbf{z}^T Q\mathbf{z} + q^T\mathbf{z} - \sum_{i=1}^{M} \lambda_i^{0^T} D_i\mathbf{z}$ and $\tau < \sigma_f/\rho(D) = \lambda_{min}(Q)/\rho(D)$.

for $k = 1, 2, \cdots$ **do**

1: $\mathbf{z}^k = \text{argmin}_{T\mathbf{z}=t} \quad \mathbf{z}^T Q\mathbf{z} + q^T\mathbf{z} - \sum_{i=1}^{M} \lambda_i^{k-1^T} D_i\mathbf{z}$

2: $\hat{\mathbf{z}}^k = \mathbf{z}^k(\alpha^{k-1} + \alpha^k - 1)/\alpha^k - \mathbf{z}^{k-1}(\alpha^{k-1} - 1)/\alpha^k$

3: $\alpha^{k+1} = (1 + \sqrt{4\alpha^{k^2} + 1})/2$

for $i = 1, \cdots, M$ **do**

4: $\sigma_i^k = \mathbf{Pr}_{\mathbb{C}_i}(D_i\hat{\mathbf{z}}^k - \frac{1}{\tau}\hat{\lambda}_i^k - d_i)$

5: $\lambda_i^k = \hat{\lambda}_i^k + \tau(d_i - D_i\hat{\mathbf{z}}^k + \sigma_i^k)$

6: $\hat{\lambda}_i^{k+1} = \lambda_i^k + (\alpha^k - 1)(\lambda_i^k - \lambda_i^{k-1})/\alpha^{k+1}$

end for

end for

Remark 4.10: The projection loop, i.e. Step 4-6 in Algorithm 2 and Algorithm 3 can be computed in parallel.

Remark 4.11: Similar to Algorithm 2, Step 2 and 3 in Algorithm 3 are equivalent to the first step of FAMA in Algorithm 1, i.e. $\hat{\mathbf{z}}^k = \text{argmin}_{T\mathbf{z}=t} \quad \mathbf{z}^T Q\mathbf{z} + q^T\mathbf{z} - \sum_{i=1}^{M} \hat{\lambda}_i^{k-1^T} D_i\mathbf{z}$. By splitting this step into two steps, we can represent $\mathbf{z}^k$ as a function of λ^k, which allows us to derive the complexity bound on $|\mathbf{z}^k - \mathbf{z}^\star|$ in Section V.

V. Compexity Bounds of FAMA for MPC

Complexity upper-bounds of optimization algorithms are important for real-time MPC, since they provide a certificate that a solution of pre-specified sub-optimality can be obtained within the available computation time. In this section, we will derive the complexity upper-bounds on the number of iterations to achieve a certain solution accuracy for the sequences generated by Algorithm 2 and Algorithm 3.

A. Complexity upper-bounds for both the primal and dual sequences $\{\mathbf{u}^k\}$ and $\{\lambda^k\}$ generated by Algorithm 2

Before we present the complexity upper-bounds on the primal and dual sequences in Theorem 5.3, we state two lemmas showing properties of convex cones, which will be used in the proof of Theorem 5.3.

Lemma 5.1: Let $\mathbb{C}$ be a convex cone. The conjugate function of the indicator function of the set $\mathbb{S} := \{v | -v \in \mathbb{C}\}$ is equal to the indicator function of the dual cone of $\mathbb{C}$, i.e., $I_{\mathbb{S}}^\star(v) = I_{\mathbb{C}^\star}(v)$.

Proof: By the definition of a convex cone, the set $\mathbb{S}$ is still a convex cone. Example 7.3.5 in [5] shows $I_{\mathbb{S}}^\star(v) = I_{\mathbb{S}^\circ}(v)$, where $\mathbb{S}^\circ$ denotes the polar cone of $\mathbb{S}$. By the definitions of the polar cone and the dual cone, we know $\mathbb{S}^\circ = \{w \mid w'v \leq 0, \ -v \in \mathbb{C}\} = \{w \mid w'v \geq 0, \ v \in \mathbb{C}\} = \mathbb{C}^\star$, and the result $I_{\mathbb{S}}^\star(v) = I_{\mathbb{C}^\star}(v)$ follows. ■

Lemma 5.2: Let $\mathbb{C}$ be the non-negative orthant $\mathbb{C} := \{v \mid v \geq 0\}$ or a second order cone $\mathbb{C} := \{[v_1, v_2] | \ |v_1| \leq v_2\}$. For any $v \in \mathbb{R}^{N_\mathbb{C}}$, the point $z = \mathbf{Pr}_{\mathbb{C}}(v) - v$ satisfies $z \in \mathbb{C}$.

Proof: For the non-negative orthant, it is easy to show that $z = \mathbf{Pr}_{\mathbb{C}}(v) - v = \max\{0, v\} - v \geq 0$. For a second-order cone, we denote $z = [z_1, z_2]$ and show that $|z_1| \leq z_2$ holds for the three cases in equation (12). For the first case, it can easily be verified that $|z_1| \leq z_2$. For the second case, we have

$$|z_1|_2 = \left| \frac{v_2 + |v_1|_2}{2|v_1|_2} v_1 - v_1 \right|_2 = \left| \frac{v_2 - |v_1|_2}{2} \right| \ ,$$

$$z_2 = \frac{v_2 + |v_1|_2}{2|v_1|_2}|v_1|_2 - v_2 = \frac{|v_1|_2 - v_2}{2} \ .$$

Since in this case it holds that $|v_1|_2 > v_2$, we get $|z_1|_2 \leq z_2$. For the third case, we have $|z_1|_2 = |v_1|_2$ and $z_2 = -v_2$. Since $|v_1|_2 \leq -v_2$, we prove $|z_1|_2 \leq z_2$. ■

Theorem 5.3: Consider Problem 4.1. Let $\{\mathbf{u}^k\}$ and $\{\lambda^k\}$ be generated by Algorithm 2, where $\lambda^k = [\lambda_1^{k^T}, \cdots, \lambda_M^{k^T}]^T$ and λ_i are the Lagrange multipliers associated with the constraint $C_i\mathbf{u} - c_i = \sigma_i$ at iteration k. If Assumption 4.2 is satisfied, then for any $k \geq 1$

$$\mathbf{D}(\lambda^\star) - \mathbf{D}(\lambda^k) \leq \frac{2\rho(C)|\lambda^{start} - \lambda^\star|^2}{\lambda_{min}(H)(k+1)^2} \ , \tag{13}$$

where $\lambda^{start} = [\lambda_1^{start^T}, \cdots, \lambda_M^{start^T}]^T$ and $\lambda^\star$ denote the starting point and the optimizer, respectively. If $\lambda_i^{start} \in \mathbb{C}_i$ for all $i = 1, \cdots, M$, then $\lambda_i^k \in \mathbb{C}_i$ for all $k \geq 1$ and $i = 1, \cdots, M$ and

$$|\mathbf{u}^k - \mathbf{u}^\star|^2 \leq \frac{4\rho(C)|\lambda^{start} - \lambda^\star|^2}{\lambda_{min}^2(H)k^2} \ . \tag{14}$$

Proof: The first objective in Problem 4.1 is equal to $f(\mathbf{u}) = \mathbf{u}^T H\mathbf{u} + h^T\mathbf{u}$, and therefore $\sigma_f = \lambda_{min}(H)$. Since Assumption 4.2 holds, Problem 4.1 satisfies Assumption 2.2, and the complexity upper-bound in (3) holds by Theorem 3.2. This directly implies the dual bound in (13) by considering the matrix A in Problem 2.1 to be the matrix $C = [C_1^T, \cdots, C_M^T]^T$ in Problem 4.1.

The second step is to prove that if $\lambda_i^{start} \in \mathbb{C}_i$ for all $i = 1, \cdots, M$, then $\lambda_i^k \in \mathbb{C}_i$ for all $k \in \mathbb{N}$ and $i = 1, \ldots, M$. From Step 4 and 5 in Algorithm 2, we know that $\lambda_i^k = \hat{\lambda}_i^k + \tau(c_i - C_i\mathbf{u}^k + \sigma_i^k) = \tau(\mathbf{Pr}_{\mathbb{C}_i}(C_i\mathbf{u}^k - \frac{1}{\tau}\hat{\lambda}_i^k - c_i) - (C_i\mathbf{u}^k - \frac{1}{\tau}\hat{\lambda}_i^k - c_i))$. By the fact that $\tau > 0$ and Lemma 5.2, we can conclude $\lambda_i^k \in \mathbb{C}_i$ for all $k \geq 1$ and $i = 1, \cdots, M$. The last step is to prove inequality (14). From Step 1 in Algorithm 2 we have

$$\mathbf{u}^k = \frac{1}{2}H^{-1}(C^T\lambda^{k-1} - h) \ ,$$

which implies

$$\begin{aligned}
|\mathbf{u}^k - \mathbf{u}^\star|^2 &= |\frac{1}{2}H^{-1}C^T(\lambda^{k-1} - \lambda^\star)|^2 \\
&\leq \frac{1}{2\lambda_{min}(H)}(\lambda^{k-1} - \lambda^\star)^T CH^{-1}C^T(\lambda^{k-1} - \lambda^\star) \\
&= \frac{2}{\lambda_{min}(H)}[\frac{1}{4}\lambda^{k-1^T}CH^{-1}C^T\lambda^{k-1} - (\frac{1}{2}h^TH^{-1}C^T + d^T)\lambda_{k-1} \\
&\quad - \frac{1}{4}\lambda^{\star^T}CH^{-1}C^T\lambda^\star + (\frac{1}{2}h^TH^{-1}C^T + d^T)\lambda^\star \\
&\quad + \frac{1}{2}((\lambda^\star)^TCH^{-1}C^T\lambda^\star - \lambda^{k^T}CH^{-1}C^T\lambda^\star) \\
&\quad + (\frac{1}{2}h^TH^{-1}C^T + d^T)(\lambda^\star - \lambda^{k-1})] \ .
\end{aligned}$$

The dual function of Problem 4.1 is

$$\begin{aligned}
\mathbf{D}(\lambda) &= -f^\star(C^T\lambda) + d^T\lambda - g^\star(-\lambda) \\
&= -\frac{1}{4}\lambda^TCH^{-1}C^T\lambda + \frac{1}{2}h^TH^{-1}C^T\lambda - \frac{1}{4}h^TH^{-1}h + d^T\lambda - \sum_{i=1}^M I^\star_{-\lambda_i \in \mathbb{C}_i} \ .
\end{aligned}$$

Since we know that all $\mathbb{C}_i$ are self-dual cones, i.e. $\mathbb{C}_i^\star = \mathbb{C}_i$, Lemma 5.1 implies that the dual function can be simplified as

$$\mathbf{D}(\lambda) = -\frac{1}{4}\lambda^TCH^{-1}C^T\lambda + (\frac{1}{2}h^TH^{-1}C^T + d^T)\lambda - \frac{1}{4}h^TH^{-1}h - \sum_{i=1}^M I_{\lambda_i \in \mathbb{C}_i} \ ,$$

and the gradient of the dual function for $\lambda_i \in \mathbb{C}_i$ is

$$\nabla\mathbf{D}(\lambda) = -\frac{1}{2}C(H^{-1})^TC^T\lambda + (\frac{1}{2}C(H^{-1})^Th + d) \ . \tag{15}$$

Since we have shown that $\lambda_i^k \in \mathbb{C}_i$ for all $k \geq 0$ and $i = 1, \cdots, M$, we obtain

$$|\mathbf{u}^k - \mathbf{u}^\star|^2 \leq \frac{2}{\lambda_{min}(H)}(\mathbf{D}(\lambda^\star) - \mathbf{D}(\lambda^{k-1}) - \nabla\mathbf{D}^T(\lambda^\star)(\lambda^\star - \lambda^{k-1})) \ .$$

Since the dual function $\mathbf{D}$ is concave, by optimality, we conclude

$$\begin{aligned}\frac{2}{\lambda_{min}(H)}(\mathbf{D}(\lambda^\star)-\mathbf{D}(\lambda^{k-1})-\nabla\mathbf{D}(\lambda^\star)(\lambda^{k-1}-\lambda^\star)) &\leq \frac{2}{\lambda_{min}(H)}(\mathbf{D}(\lambda^\star)-\mathbf{D}(\lambda^{k-1}))\\ &\leq \frac{4\rho(C)|\lambda^{start}-\lambda^\star|^2}{\lambda_{min}^2(H)k^2}\ .\end{aligned}$$

■

Remark 5.4: Theorem 5.3 can be directly extended to MPC problem with positive semi-definite cone constraints, since a positive semi-definite cone is also a self-dual cone.

Remark 5.5: The proof of inequality (14) is an extension of the proof in [16] for the case of polytopic constraints in the context of distributed MPC problems.

B. Complexity upper-bounds for both the primal and dual sequences $\{\mathbf{z}^k\}$ and $\{\lambda^k\}$ generated by Algorithm 3

Theorem 5.6: If Assumption 4.8 holds, the sequence $\{\lambda_k\}$ generated by Algorithm 3 satisfies

$$\mathbf{D}(\lambda^\star)-\mathbf{D}(\lambda^k)\leq\frac{2\rho(D)|\lambda^{start}-\lambda^\star|^2}{\lambda_{min}(Q)(k+1)^2}\ , \tag{16}$$

where $\lambda^{start}=[\lambda_1^{start^T},\cdots,\lambda_M^{start^T}]^T$ and $\lambda^\star$ denote the starting point and the optimizer, respectively. If $\lambda_i^{start}\in\mathbb{C}_i$ for all $i=1,\cdots,M$, then $\lambda_i^k\in\mathbb{C}_i$ for all $k\geq 1$ and $i=1,\cdots,M$ and

$$|\mathbf{z}^k-\mathbf{z}^\star|^2\leq\frac{2\lambda_{max}(M_1)\rho(D)|\lambda^{start}-\lambda^\star|^2}{\lambda_{max}(I-M_1^TQ)\lambda_{min}(Q)k^2}\ , \tag{17}$$

where $M_1=\frac{1}{2}(Q^{-1}-Q^{-1}T^T(TQ^{-1}T^T)^{-1}TQ^{-1})$, and I is an identity matrix.

Proof: Since Assumption 4.8 is satisfied, Problem 4.7 satisfies Assumption 3.4. It follows from Theorem 3.6 that the sequence $\{\lambda^k\}$ generated by Algorithm 3 satisfies the complexity bound in (16). Following the same argument as in the proof of Theorem 5.3, we obtain that if $\lambda_i^{start}\in\mathbb{C}_i$ for all $i=1,\cdots,M$, then $\lambda_i^k\in\mathbb{C}_i$ for all $k\geq 1$ and $i=1,\cdots,M$. In the following, we prove inequality (17). From Step 1 in Algorithm 3, we know

$$\mathbf{z}^k=M_1(D^T\lambda^{k-1}-q)+M_2t\ ,$$

where $M_1\in\mathbb{R}^{n_z\times n_z}$, $M_2\in\mathbb{R}^{n_z\times n_t}$ and

$$\begin{bmatrix}M_1 & M_2\\ M_3 & M_4\end{bmatrix}=\begin{bmatrix}2Q & T^T\\ T & 0\end{bmatrix}^{-1}=\begin{bmatrix}\frac{1}{2}(Q^{-1}-Q^{-1}T^T(TQ^{-1}T^T)^{-1}TQ^{-1}) & Q^{-1}T^T(TQ^{-1}TT)^{-1}\\ (TQ^{-1}T^T)^{-1}TQ^{-1} & -\frac{1}{2}(TQ^{-1}T^T)^{-1}\end{bmatrix}. \tag{18}$$

Then we have

$$\begin{aligned}|\mathbf{z}^k-\mathbf{z}^\star|^2=&|M_1D^T(\lambda^{k-1}-\lambda^\star)|^2\\ \leq&\frac{\lambda_{max}(M_1)}{\lambda_{max}(I-M_1^TQ)}(\lambda^{k-1}-\lambda^\star)^TD(I-M_1^TQ)M_1D^T(\lambda^{k-1}-\lambda^\star)\\ =&\frac{\lambda_{max}(M_1)}{\lambda_{max}(I-M_1^TQ)}\Big[-\lambda^{k-1^T}D(M_1^TQ-I)M_1D^T\lambda^{k-1}+\lambda^{\star^T}D(M_1^TQ-I)M_1D^T\lambda^\star\\ &-(2DM_1q-DM_2t+2DM_1^TQM_2t-2DM_1^TQM_1q+d)\lambda^{k-1}\\ &+(2DM_1q-DM_2t+2DM_1^TQM_2t-2DM_1^TQM_1q+d)\lambda^\star\\ &-2\lambda^{\star^T}D(M_1^TQ-I)M_1D^T\lambda^\star+2\lambda^{\star^T}D(M_1^TQ-I)M_1D^T\lambda^{k-1}\\ &-(2DM_1q-DM_2t+2DM_1^TQM_2t-2DM_1^TQM_1q+d)(\lambda^\star-\lambda^{k-1})\Big]\ .\end{aligned}$$

The dual function of Problem 4.7 is equal to

$$\begin{aligned}\mathbf{D}(\lambda)=&-\bar{f}(D^T\lambda)+d^T\lambda-g^\star(-\lambda)\\ =&\lambda^TD(M_1^TQM_1-M_1)D^T\lambda+(M_1q-M_2t)^TQ(M_1q-M_2t)-q^T(M_1q-M_2t)\\ &+(2DM_1q-DM_2t+2DM_1^TQM_2t-2DM_1^TQM_1q+d)^T\lambda-\sum_{i=1}^M I_{\lambda_i\in\mathbb{C}_i}\ .\end{aligned}$$

Note that the matrix $(I-M_1^TQ)M_1$ is positive semi-definite, which can be easily seen as the dual function $\mathbf{D}(\lambda)$, which contains the quadratic function with weight matrix $(M_1^TQ-I)M_1$, is always concave. Hence, the function $-\mathbf{D}(\lambda)$ is convex,

which implies $(I - M_1^T Q)M_1$ is a positive semi-definite matrix. Since we have shown that $\lambda_i^k \in \mathbb{C}_i$ for all $k \geq 0$ and $i = 1, \cdots, M$, we obtain

$$|\mathbf{z}^k - \mathbf{z}^\star|^2 \leq \frac{\lambda_{max}(M_1)}{\lambda_{max}(I - M_1^T Q)}(\mathbf{D}(\lambda^\star) - \mathbf{D}(\lambda^\star) - \nabla \mathbf{D}^T(\lambda^{k-1})(\lambda^\star - \lambda^{k-1})) \ .$$

By optimality, we conclude

$$|\mathbf{z}^k - \mathbf{z}^\star|^2 \leq \frac{\lambda_{max}(M_1)}{\lambda_{max}(I - M_1^T Q)}(\mathbf{D}(\lambda^\star) - \mathbf{D}(\lambda^{k-1})) \leq \frac{2\lambda_{max}(M_1)\rho(D)|\lambda^{start} - \lambda^\star|^2}{\lambda_{max}(I - M_1^T Q)\lambda_{min}(Q)k^2} \ .$$

■

VI. Computation of complexity bounds

Theorem 5.3 and Theorem 5.6 provide complexity bounds on the number of iterations for Algorithm 2 and Algorithm 3 to reach a sub-optimal solution. For a given MPC problem, all quantities in the complexity bounds are known except for $|\lambda^{start} - \lambda^\star|$. This section is devoted to computing or approximating the value of $|\lambda^{start} - \lambda^\star|$. Early work includes [4] and [23], where the authors consider quadratic programming (QP) problems and estimate the unknown term by solving an off-line mixed-integer linear programming problem. In this section, we provide two methods to compute $|\lambda^{start} - \lambda^\star|$. We consider the cold-starting strategy, i.e. $\lambda^{start} = 0$. Then, the problem reduces to computing the largest $|\lambda^\star_{min}(\bar{x})|$ for all initial states $\bar{x} \in \mathbb{X}_0$, where $\mathbb{X}_0$ denotes the set of feasible states for Problem 4.1. The optimal value for a given $\mathbb{X}_0$ is denoted by $|\lambda^\star_{min}|$ and is obtained by solving the optimization problem

$$|\lambda^\star_{min}| := \max_{\bar{x} \in \mathbb{X}_0} \quad |\lambda^\star_{min}(\bar{x})| \ , \tag{19}$$

where $\lambda^\star_{min}(\bar{x})$ is the minimal l_2-norm solution

$$\lambda^\star_{min}(\bar{x}) := \text{argmin}_{\lambda \in \Lambda^\star(\bar{x})} \quad |\lambda| \ , \tag{20}$$

where $\Lambda^\star(\bar{x})$ denotes the set of optimal Lagrange multipliers of Problem 4.1 or Problem 4.7 for a given initial state $\bar{x}$. The constraint $\lambda \in \Lambda^\star(\bar{x})$ is equivalent to that λ satisfies the KKT conditions of Problem 4.1 or Problem 4.7. In the following, we focus on the computation of $|\lambda^\star_{min}|$ for a given $\mathbb{X}_0$ for Problem 4.1, and present two methods to approximate the optimal solution $|\lambda^\star_{min}|$. The results can be easily extended to Problem 4.7.

A. Compute upper-bound of $|\lambda^\star_{min}|$ using sum of squares (SOS) relaxations.

From the description above, we know that $|\lambda^\star_{min}|$ is the optimal solution of a three-level optimization problem, i.e. Problem 4.1 and the two problems in (19) and (20). In this section, we propose a method, which can avoid solving the three-level problem and provides an upper bound of $|\lambda^\star_{min}|$. Note that replacing $|\lambda^\star_{min}|$ by its upper bound, the complexity upper bounds in Theorem 5.3 still hold. The idea of the method is the following. We first rewrite the three-level problem as one optimization problem by involving the KKT conditions of the inner problems as constraints, i.e. we solve problem (19) subject to the KKT conditions of Problem 4.1 and those of problem (20). The KKT conditions include conic constraints originating from the primal and dual feasibility and polynomial constraints originating from the optimality and the complementary slackness conditions. Due to the fact that polynomial constraints are nonconvex, we use sum of squares relaxations to approximate the solution of the problem, resulting in an SDP problem. See, e.g. [11] for previous work on using SOS relaxations for optimization problems with polynomial constraints.

The KKT conditions of Problem 4.1 are obtained from the KKT conditions derived for Problem 2.1 in Section III-C and Proposition 3.7. Since the constraint $\lambda \in \Lambda^\star(\bar{x})$ is equivalent to the KKT conditions of Problem 4.1, problem (20) can be represented in (21), where $\mathbb{C}_i^\circ$ denotes the polar cone of $\mathbb{C}_i$. Note that problem (21) corresponds to one initial state $\bar{x}$. The vectors h and c_i are affine functions of $\bar{x}$. Then we derive the KKT conditions of problem (21), which can be easily obtained by following the standard rules in [5] and [8], and put these KKT conditions into problem (19). Due to the non-convexity of the polynomial constraints originating from the optimality and the complementary slackness conditions, we apply SOS relaxations in [11] to the problem to compute an upper bound of $|\lambda^\star_{min}|$. It is important to point out that according to the theorems in [11], SOS relaxations always provide us an upper-bound of the optimal solution $|\bar{\lambda}^\star_{min}| > |\lambda^\star_{min}|$, and there exists a sufficiently high-order SOS relaxation such that $|\bar{\lambda}^\star_{min}| = |\lambda^\star_{min}|$.

$$\begin{aligned} \lambda^\star_{min}(\bar{x}) = \text{argmin}_{\lambda, \mathbf{u}, \sigma_i} \quad & |\lambda| \\ s.t. \quad & 2H\mathbf{u} + h + \sum_{i=1}^{M} C_i^T \lambda_i = 0, \\ & C_i \mathbf{u} - d_i = \sigma_i, \ \sigma_i \in \mathbb{C}_i, \ \lambda_i \in \mathbb{C}_i^\circ, \ \sigma_i^T \lambda_i = 0, \ i = 1, \cdots, M \ , \end{aligned} \tag{21}$$

B. Sample-based estimation of $|\lambda^\star_{min}|$

We present a sample-based approach to estimate $|\lambda^\star_{min}|$, which can be easily applied for all problem dimensions. The optimization problem in (19) can be reformulated as

Problem 6.1:

$$\begin{aligned} |\lambda^\star_{min}| = \min \quad & \gamma \\ s.t. \quad & |\lambda^\star_{min}(\bar{x})| - \gamma \leq 0, \ \forall \bar{x} \in \mathbb{X}_0 \ , \end{aligned}$$

where $\lambda^\star(\bar{x})$ is defined in (20). We consider $\bar{x}$ as an uncertainty parameter, and apply the scenario approach in [9] to Problem 6.1. We first introduce some required definitions.

Definition 6.2 (Definition 1 in [9]): Let $\tilde{\gamma} \in \mathbb{R}$ be a candidate solution for Problem 6.1. The probability of violation of $\tilde{\gamma}$ is defined as

$$V(\tilde{\gamma}) := \mathbf{P}\{\bar{x} \in \mathbb{X}_0 : \ \tilde{\gamma} \leq |\lambda^\star(\bar{x})|\} \ .$$

Definition 6.3 (Definition 2 in [9]): Let $\epsilon \in [0,1]$. We say that $\tilde{\gamma} \in \mathbb{R}$ is an ϵ-level robustly feasible solution if $V(\tilde{\gamma}) \leq \epsilon$. We collect N_s random samples $\{\bar{x}_1, \cdots, \bar{x}_{N_s}\}$ in $\mathbb{X}_0$, and construct the sample-based optimization problem

Problem 6.4:

$$\begin{aligned} |\lambda^{N_s}_{min}| := \min \quad & \gamma \\ s.t. \quad & |\lambda^\star_{min}(\bar{x}_i)| - \gamma \leq 0, \forall \ \bar{x}_i, \quad i = 1, \cdots, N_s \ . \end{aligned}$$

The following corollary states the probabilistic meaning of the optimal solution $|\lambda^{N_s}_{min}|$ returned by Problem 6.4.

Corollary 6.5: (Corollary 1 in [9]) Fix two real numbers $\epsilon \in [0,1]$ (level parameter) and $\beta \in [0,1]$ (confidence parameter) and let $N_s \geq \frac{1}{\epsilon\beta} - 1$. Then, with probability no smaller than $1-\beta$, $|\lambda^{N_s}_{min}|$ returned by Problem 6.4 is an optimal solution with ϵ-level robust feasibility for Problem 6.1.

The remaining question is how to solve Problem 6.4. In the following, we provide a method to compute an upper-bound of $|\lambda^{N_s}_{min}|$. For a sample $\bar{x}_i$, an optimal solution $\lambda^\star(\bar{x}_i)$ can be computed by running Algorithm 2. Since the solution satisfies $|\lambda^\star(\bar{x}_i)| \geq |\lambda^\star_{min}(\bar{x}_i)|$, then, we can easily compute $|\tilde{\lambda}^{N_s}_{min}| := \max_{1 \leq i \leq N_s}\{|\lambda^\star(\bar{x}_i)|\}$, which satisfies $|\tilde{\lambda}^{N_s}_{min}| \geq |\lambda^{N_s}_{min}|$.

The procedure of using the scenario approach to estimate a solution for Problem 6.1 is summarized as the follows: Choose ϵ and β, and take N_s satisfying $N_s \geq \frac{1}{\epsilon\beta} - 1$. Randomly draw N_s samples $\{\bar{x}_1, \cdots, \bar{x}_{N_s}\}$ in $\mathbb{X}_0$ and run Algorithm 2 to calculate the corresponding $\{|\lambda^\star(\bar{x}_1)|, \cdots, |\lambda^\star(\bar{x}_{N_s})|\}$. Then compute $|\tilde{\lambda}^{N_s}_{min}| = \max_{1 \leq i \leq N_s}\{|\lambda^\star(\bar{x}_i)|\}$, which is an upper-bound of an optimal solution with ϵ-level robust feasibility for Problem 6.1. Note that all computation is done off-line, and therefore a large number of samples can be potentially considered to compute a good approximation of $|\lambda^\star_{min}|$ with high probabilistic confidence.

VII. Preconditioning

Preconditioning has been observed to offer significant compuational speedups in gradient based methods in [24] and [15]. In [24] and [15], preconditioning methods for solving linear MPC problem with polytopic constraints were presented. In this section, we present a preconditioning technique to improve the performance of FAMA, when applied to linear MPC problems with polytopic and ellipsoidal (a special case of second-order cone) constraints. The goal of the method is to enlarge the step-size and decrease the complexity bounds of the algorithms by minimizing the condition number of the constraint matrices.

We again focus on the first splitting in Problem 4.1. The preconditioning method can be easily extended to the second splitting in Problem 4.7. Recall the condition on the step-size $\tau < \lambda_{min}(H)/\rho(C)$ and the complexity bound in Theorem 5.3. The value $\rho(C)$ affects them in the way that the smaller $\rho(C)$ is, the larger the step-size can be and the smaller the complexity bound will be. Therefore, we minimize the condition number of the matrix C by imposing preconditioning matrices to C to enlarge the step-size and decrease the complexity bounds. In order to simplify the notation, we assume that Problem 4.1 has only two constraints, a polytopic constraint $C_1\mathbf{u} - c_1 \geq 0$ and an ellipsoidal constraint $|C_2\mathbf{u} - c_2| \leq 1$. We introduce a positive-definite diagonal matrix P_1 and a positive-definite matrix P_2 to precondition the constraints: P_1 to scale the polytopic constraints $P_1C_1\mathbf{u} - P_1c_1 \geq 0$, and $P_2 \in \mathbb{R}^{n_{C_2} \times n_{C_2}}$ to reshape the ellipsoidal constraint $|P_2C_2\mathbf{u} - P_2c_2| \leq 1$, where n_{C_1} and n_{C_2} denote the number of rows of the matrices C_1 and C_2, respectively.

We first compute optimal preconditioning matrices P_1 and P_2 minimizing the condition number of C^TC by means of the following optimization problem (see Chapter 3.1 in [6]). Let $W_1 = P_1^TP_1$ and $W_2 = P_2^TP_2$.

Problem 7.1:

$$\begin{aligned} \min_{\alpha, W_1, W_2} \quad & \alpha \\ s.t. \quad & \mu I \preceq \begin{bmatrix} C_1^T & C_2^T \end{bmatrix} \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \preceq \alpha I \ , \\ & W_1 = blkdiag(w_1, \cdots, w_{n_{C_1}}) \ , \ w_i > 0 \ , i = 1, \cdots, n_{C_1} \ , \ W_2 \succ 0 \ , \end{aligned}$$

where μ is equal to the minimum eigenvalue of C^TC. It is important to point out that by setting the preconditioning matrix P_1 to be a diagonal matrix, the preconditioned polytopic constraint is equivalent to the original one. However, the preconditioning matrix P_2 changes the ellipsoidal constraint, and the preconditioned ellipsoidal constraint is not equivalent to the original one, which means that the preconditioning of the ellipsoidal constraint modifies the optimal solution. In order to guarantee that the solution given by the preconditioned problem is still a sub-optimal and feasible solution to Problem 4.1, one extra step is required to compute the maximal reshaped ellipsoidal constraint contained in the original ellipsoidal constraint. Problem 7.2 computes the maximal inner approximation.

Problem 7.2:

$$\begin{aligned} &\min_{\beta,\omega} \quad \beta \\ &s.t. \quad \begin{bmatrix} -\omega+1 & 0 & 0 \\ 0 & \omega I & \beta(P_2C_2)^{-1} \\ 0 & \beta(P_2C_2)^{-1} & (C_2^TC_2)^{-1} \end{bmatrix} \succeq 0 \ , \ \omega \geq 0 \ , \ \beta \geq 0 \ . \end{aligned}$$

The LMI constraint in Problem 7.2 guarantees that the new scaled ellipsoidal constraint $|P_2C_2\mathbf{u} - P_2c_2| \leq \beta$ is contained in the original constraint $|C_2\mathbf{u} - c_2| \leq 1$ (see Chapter 8.4.2 in [8]). Note that the matrix P_2 in Problem 7.2 is not an optimization variable but a constant computed by Problem 7.1. We summarize the properties of the proposed preconditioning method in the following theorem.

Theorem 7.3: Let $\mathbf{u}^\star$ be an optimal solution of the original problem: $\min_{\mathbf{u}} \ \mathbf{u}^TH\mathbf{u} + h^T\mathbf{u}$, s.t. $C_1\mathbf{u} - c_1 \geq 0$ and $|C_2\mathbf{u} - c_2| \leq 1$, and let $\mathbf{u}_p^\star$ be an optimal solution of the preconditioned problem: $\min_{\mathbf{u}_p} \ \mathbf{u}_p^TH\mathbf{u}_p + h^T\mathbf{u}_p$, s.t $P_1C_1\mathbf{u}_p - P_1c_1 \geq 0$ and $|P_2C_2\mathbf{u}_p - P_2c_2| \leq \beta$, where the matrices P_1 and P_2 and the parameter β are computed by Problem 7.1 and Problem 7.2. The following holds:

- The optimal solution $\mathbf{u}_p^\star$ is a feasible solution of the original problem, i.e. $C_1\mathbf{u}_p^\star - c_1 \geq 0$ and $|C_2\mathbf{u}_p^\star - c_2| \leq 1$.
- The preconditioned constraint matrix $C_p = [C_1^TP_1^T, C_2^TP_2^T]^T$ satisfies $\rho(C_p) \leq \rho(C)$, where $C = [C_1^T, C_2^T]^T$.

Proof: Since the matrix W_1 is set to be a positive definite diagonal matrix, the preconditioned polytopic constraint $P_1C_1\mathbf{u}_p - P_1c_1 \geq 0$ is equivalent to the original one $C_1\mathbf{u} - c_1 \geq 0$. Problem 7.2 guarantees that the preconditioned ellipsoidal constraint $|P_2C_2\mathbf{u}_p - P_2c_2| \leq \beta$ is an inner-approximation of the original ellipsoidal constraint. Hence, the optimal solution $\mathbf{u}_p^\star$ is a feasible solution of the original problem. By optimality of P_1 and P_2 for Problem 7.1, it follows immediately that $\rho(C_p) \leq \rho(C)$. ∎

In the following, we consider the special case that the ellipsoidal constraint $|C_2\mathbf{u} - c_2| \leq 1$ originates from a terminal state constraint, which is a common problem type in MPC. In this case, the preconditioning of the ellipsoidal constraint has to not only maintain feasibility, but also stability properties. To address this problem, we modify the preconditioning method in the following way. Some notations and preliminaries about stability of an MPC controller are first presented. Consider the discrete-time linear time-invariant system $x_{t+1} = A_dx_t + B_du_t$, where x_t and u_t denote the state and input at time t, and A_d and B_d denote the dynamical matrices. Let $\mathbb{X}$ and $\mathbb{U}$ be the state and input constraints and K be a linear control law, such that $A_d + B_dK$ is stable.

Definition 7.4: (Positive invariant (PI) set): A set $\mathbb{P} \subseteq \mathbb{R}^n$ is a positively invariant set of system $x_{t+1} = A_dx_t + B_dKx_t$, if $A_dx_t + B_dKx_t \in \mathbb{P}$ and $Kx_t \in \mathbb{U}$ for all $x_t \in \mathbb{P}$.

The original ellipsoidal terminal constraint on the state is denoted by $|Ex_N - e| \leq 1$, where $E \succ 0$ and N is the horizon of the MPC problem. Since x_N can be represented by a linear combination of the control sequence $\mathbf{u}$ and the initial state $\bar{x}$, i.e., $x_N = S_1\mathbf{u} + S_2\bar{x}$, it follows that $C_2 = ES_1$ and $c_2 = e - ES_2\bar{x}$.

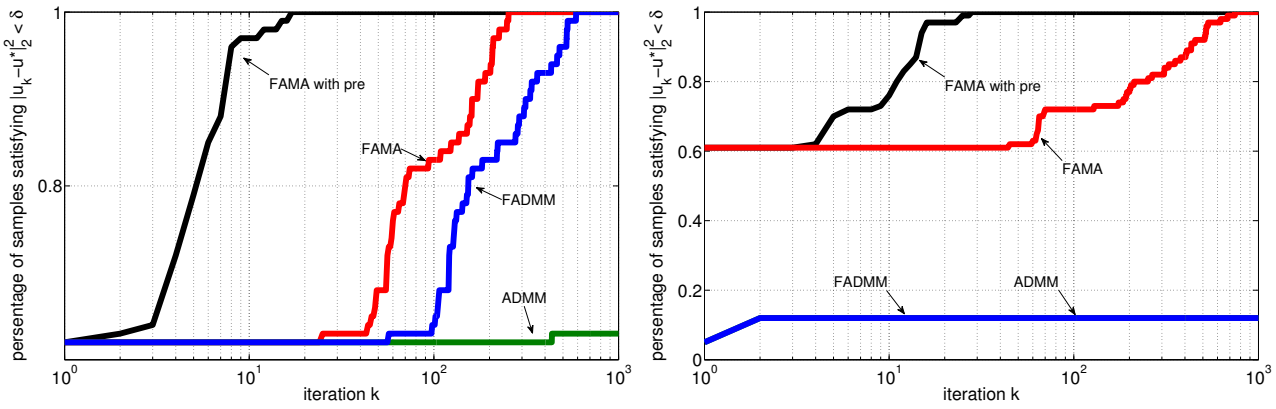
According to the standard MPC theory, the original terminal constraint $|Ex_N - e| \leq 1$ has to be a positively invariant set to guarantee stability of the closed-loop system. In order to maintain this property for the preconditioned ellipsoidal constraint, an additional invariance condition is imposed on W_2 in Problem 7.1. We exemplify this procedure for an ellipsoidal terminal constraint of the form $\mathbb{X}_f = \{x_N | x_N^T\Gamma x_N < 1\}$, where $\Gamma \succ 0$, i.e. $\Gamma = E^TE$ and $e = 0$. Invariance of the preconditioned ellipsoid can be ensured by enforcing the constraint in (22) in Problem 7.1, which can be written as an LMI by using Schur complements. Again, by solving Problem 7.2, we can compute the maximal inner approximation in $|Ex_t - e| \leq 1$. If the state and input constraints $\mathbb{X}$ and $\mathbb{U}$ are polytopic sets, then the inner approximation in Problem 7.2 can be relaxed by only requiring the scaled new ellipsoid to be contained in $\mathbb{X} \cap K\mathbb{U}$.

$$(A_d + B_dK)^T\Gamma^{\frac{1}{2}^T}W_2\Gamma^{\frac{1}{2}}(A_d + B_dK) - \Gamma^{\frac{1}{2}^T}W_2\Gamma^{\frac{1}{2}} \preceq 0 \tag{22}$$

We summarize that if the matrices P_1 and P_2 are computed by Problem 7.1 with the extra constraint in (22), and the parameter β is computed by Problem 7.2, then the preconditioned ellipsoidal constraint $|P_2C_2\mathbf{u}_p - P_2c_2| \leq \beta$ is a positive invariant set. All invariance and stability properties of the original MPC problem will be maintained by the preconditioning.

Remark 7.5: Since the matrices C_1, E and S_1 are independent of the initial state $\bar{x}$, the computation of the preconditioning matrices P_1 and P_2 and the parameter β can be computed off-line.

Remark 7.6: The preconditioning method introduced in this section can be easily extended to the case with more than two constraints and with ellipsoidal input constraints.



(a) $\delta = 10^{-4}$ (b) $\delta = 10^{-6}$

Figure 1: Performance of ADMM, FADMM, FAMA and FAMA with preconditioning applied to Problem 4.1 for the quadroter example

Remark 7.7: The preconditioning is enabled by the existence of the step-size rule and the complexity bounds. Since ADMM and FADMM do not provide these properties, it is unclear how to provide a similar preconditioning method for them. The derived theoretical properties of FAMA therefore also have practical implications by allowing for tuning the algorithm and improving its performance for the particular problem at hand.

VIII. Numerical Example

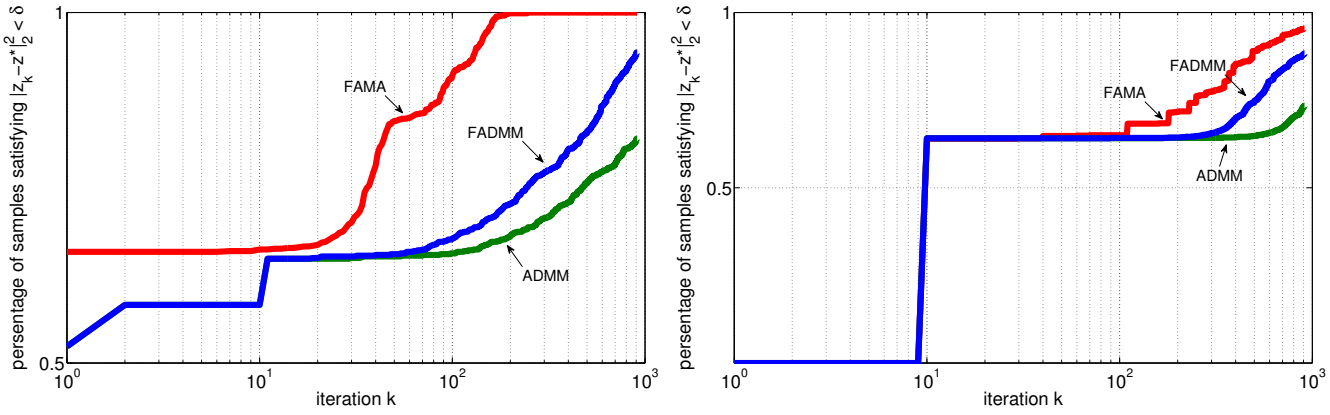
This section illustrates the theoretical findings of the paper and demonstrates the performance of FAMA for solving MPC problems. We consider a quadroter model, see [20], which is driven by four independently controlled rotors. In this experiment, we use a cascaded control structure and design an MPC controller to control the inner-loop, which is in charge of the derivative of the height of the quadroter, the roll, pitch and yaw angles and the derivative of these angles, i.e $x = [\dot{z}, \alpha, \beta, \gamma, \dot{\alpha}, \dot{\beta}, \dot{\gamma}]^T$. The state of the system is subject to constraints on the maximum angle, maximum angle velocity as well as maximum velocity in the z direction - these constraints are mainly chosen to ensure validity of the linearized model and have been specified as: $|\dot{z}| \leq 1m/s$, $|\alpha| \leq 10°$, $|\beta| \leq 10°$, $|\dot{\alpha}| \leq 15°$, $|\dot{\beta}| \leq 15°$ and $|\dot{\gamma}| \leq 60°$. The input constraint is $0 \leq u \leq 1$. The horizon of the MPC controller is set to $N = 25$. The terminal state x_N is subject to a positively invariant ellipsoidal terminal constraint.

In the simulations shown in Fig. 1 and Fig. 2, we collect 1000 randomly sampled initial states in the set $\{\bar{x}|[-0.5m/s, -5°, -5°, -60°, -5°/s, -5°/s, -30°/s]^T \leq \bar{x} \leq [0.5m/s, 5°, 5°, 60°, 5°/s, 5°/s, 30°/s]^T\}$ and compare the proposed FAMA algorithms with ADMM and FADMM for the two splitting strategies in Problem 4.1 and Problem 4.7. For FAMA, the step-size is set to $0.99 * \lambda_{min}(H)/\rho(C)$ and $0.99 * \lambda_{min}(Q)/\rho(D)$, respectively, while for ADMM and FADMM the step-size is set to the best value obtained by manual tuning. Performance is measured by the percentage of samples, for which $|\mathbf{u}^k - \mathbf{u}^\star|/|\mathbf{u}^\star| < \delta$ or $|\mathbf{z}^k - \mathbf{z}^\star|/|\mathbf{z}^\star| < \delta$ after k iterations. For the first splitting strategy, FAMA with preconditioning shows the best performance, fastest convergence speed and good accuracy after few iterations. FAMA without preconditioning converges more slowly but still faster than ADMM and FADMM. Fig. 1b shows that the solution accuracy given by ADMM and FADMM is inferior to FAMA. They achieve a solution accuracy of $\delta = 10^{-6}$ in 1000 iterations for only 10% of the samples. In Fig. 2, it is shown that for the second splitting strategy, FAMA similarly performs better than ADMM and FADMM, fast convergence speed and good accuracy. The preconditioning is not shown for the second splitting strategy, as it will not provide significant improvements due to the fact that for this example the constraint matrix D is a block-diagonal matrix. Fig. 1 and Fig. 2 also show that for this example FAMA on the first splitting strategy in Problem 4.1 requires less iterations and provide better accuracy than FAMA on the second one in Problem 4.7.

Fig. 3 illustrates the primal sequences generated by Algorithm 2 and Algorithm 3 and the corresponding complexity bounds in Theorem 5.3 and Theorem 5.6, respectively, for the initial state $\bar{x} = [0.5m/s, 5°, 5°, 60°, 5°/s, 5°/s, 20°/s]^T$. The complexity bounds are computed by setting $\lambda^{start} = 0$ and assessing $|\lambda^*|$ by the sample-based method in Section VI-B using 3000 samples. According to Corollary 6.5, β and ϵ are set to be 1.6×10^{-2}. In Fig. 3a, it can be clearly seen that the preconditioning method improves the convergence speed of the algorithm and reduces the complexity upper-bound. As the number of iterations k increases, the complexity upper-bound of Algorithm 2 with preconditioning appears to be less tight, and the practical convergence is faster than the bound. The complexity bound for the second splitting in Fig. 3b shows similar behavior.

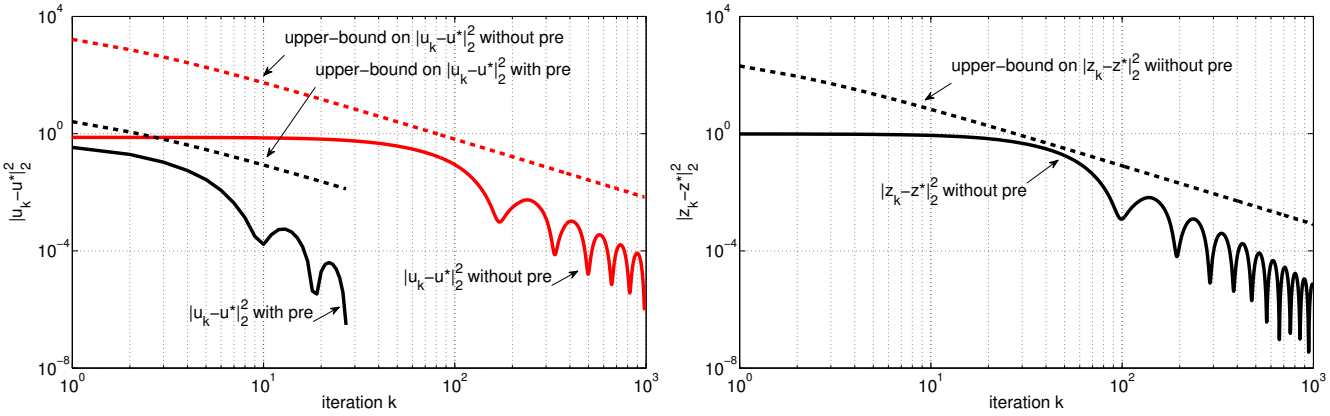
Acknowledgment

The authors would like to thank Jean-Hubert Hours, Milan Korda and Georgios Stathopoulos for helpful discussions.



(a) $\delta = 10^{-4}$ (b) $\delta = 10^{-6}$

Figure 2: Performance of ADMM, FADMM and FAMA applied to Problem 4.7 for the quadroter example



(a) (b)

Figure 3: Illustration of $|\mathbf{u}^k - \mathbf{u}^\star|^2$ and $|\mathbf{z}^k - \mathbf{z}^\star|^2$ generated by (a) Algorithm 2, (b) Algorithm 3 and the corresponding complexity upper-bounds in (a) Theorem 5.3 and (b) Theorem 5.6 for $\bar{x} = [-0.5m/s, 10^\circ, -10^\circ, 60^\circ, 10^\circ/s, 10^\circ/s, 20^\circ/s]$.

REFERENCES

[1] V. Acary, O. Bonnefon, and B. Brogliato. *Nonsmooth Modeling and Simulation for Switched Circuits*, volume 69 of *Lecture Notes in Electrical Engineering*. Springer Netherlands, 2011.

[2] R. A. Bartlett and L. T. Biegler. QPSchur: a dual, active-set, schur-complement method for large-scale and structured convex quadratic programming. *Optimization and Engineering*, 7:5–32, 2006.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, pages 183–202, 2009.

[4] A. Bemporad and M. Panaglotis. Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. In *4th IFAC Nonlinear Model Predictive Control Conference*, pages 14–20, 2012.

[5] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex analysis and optimization*. Athena Scientific Belmont, 2003.

[6] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[9] G. Calafiore and M. C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, pages 25–46, 2005.

[10] P. L. Combettes and J-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pages 185–212. Springer New York, 2011.

[11] J. Demmel, J. Nie, and V. Powers. Representations of positive polynomials on noncompact semialgebraic sets via KKT ideals. *Journal of Pure and Applied Algebra*, 209(1):189–200, 2007.

[12] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Technical report, DTIC Document, 2012.

[13] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *Proc. of the 51st IEEE Conf. on Decision and Control*, pages 668–674, 2012.

[14] H. I. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[15] P. Giselsson. Optimal preconditioning and iteration complexity bounds for gradient-based optimization in model predictive control. In *American Control Conference*, pages 358–364, 2013.

[16] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49:829–833, 2013.
[17] T. Goldstein, B. O'Donoghue, and S. Setzer. Fast alternating direction optimization methods. *CAM report*, pages 12–35, 2012.
[18] B. He and X. Yuan. On the $o(1/n)$ convergence rate of the douglasrachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.
[19] M. Kogel and R. Findeisen. Fast predictive control of linear systems combining nesterov's gradient method and the method of multipliers. In *Proc. of the 50th IEEE Conf. on Decision and Control*, pages 501–506, 2011.
[20] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE International Conference on Robotics and Automation*, pages 2520–2525, 2011.
[21] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.
[22] B. O'Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, page 1, 2013.
[23] P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.
[24] R. Richter, C. N. Jones, and M. Morari. Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control*, 57(6):1391–1403, 2012.
[25] S. Richter, C. N. Jones, and M. Morari. Certification aspects of the fast gradient method for solving the dual of parametric convex programs. *Mathematical Methods of Operations Research*, 77(3):305–321, 2012.
[26] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29:119–138, 1991.
[27] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18:267–278, 2010.