A generalized distributed accelerated gradient method for distributed model predictive control with iteration complexity bounds

Pontus Giselsson
Department of Automatic Control LTH
Lund University
Box 118, SE-221 00 Lund, Sweden
pontusg@control.lth.se

***Abstract*— Most distributed optimization methods used for distributed model predictive control (DMPC) are gradient based. Gradient based optimization algorithms are known to have iterations of low complexity. However, the number of iterations needed to achieve satisfactory accuracy might be significant. This is not a desirable characteristic for distributed optimization in distributed model predictive control. Rather, the number of iterations should be kept low to reduce communication requirements, while the complexity within an iteration can be significant. By incorporating Hessian information in a distributed accelerated gradient method in a well-defined manner, we are able to significantly reduce the number of iterations needed to achieve satisfactory accuracy in the solutions, compared to distributed methods that are strictly gradient-based. Further, we provide convergence rate results and iteration complexity bounds for the developed algorithm.**

I. Introduction

Many distributed optimization algorithms are based on gradient methods, see [3] and the references therein. Gradient-based optimization methods have low computational complexity within each iteration. However, a limitation of gradient-based methods is the slow convergence rate. For functions with a Lipschitz continuous gradient, i.e., smooth functions, classical gradient-based methods converge at a rate of $O(\frac{1}{k})$ as shown in [2], [17], where k is the iteration number. This convergence rate is not optimal for gradient methods. It was in [14] shown that a lower bound on the convergence rate for gradient-based methods is $O(\frac{1}{k^2})$. The first method that achieves this accelerated convergence rate was presented by Nesterov in [15] for unconstrained problems. This result has been extended and generalized in several publications to handle constrained smooth problems and smooth problems with an additional non-smooth term [16], [18], [1], [21]. Recently the accelerated gradient methods has been generalized in [24] to allow for a step matrix instead of a scalar step length with preserved convergence rate guarantees.

In the DMPC literature, some distributed optimization methods have been used to control sparsely interacting dynamical systems. These include [13], [22], [6] in which different reformulations of the classical gradient method with suboptimal step sizes are used to solve the dual problem. In [9] an accelerated gradient method is used to solve the DMPC problem and the optimal step size is provided. Further, in [8] iteration bounds for the method presented in [9] are given. In [20] a quasi-Newton method is used to solve the DMPC problem in a water distribution network. The subproblems are solved in parallel, but a central coordinator is needed for this approach.

In this paper we extend the results in [9] and [8] using the generalized accelerated gradient algorithm presented in [24]. We present a distributed optimization algorithm applicable to DMPC that use not only gradient information, as is common in distributed optimization, but also Hessian information in each iteration. This significantly improves convergence rate compared to previous gradient-based distributed optimization methods for DMPC as is demonstrated by a numerical example. We also provide a bound on the number of iterations needed to guarantee a prespecified dual accuracy and indicate how an iteration bound for the primal variables can be computed. The latter bound is left out for space considerations.

II. Problem setup

The problem of controlling a linear dynamical system in distributed fashion to the origin is considered. We assume polytopic constraints and apply a distributed MPC controller in which the following optimal control problem with initial condition $\bar{x} \in \mathbb{R}^n$ is solved iteratively

$$\begin{aligned} V_N(\bar{x}) := \min_{\mathbf{x},\mathbf{u}} \quad & \frac{1}{2}\sum_{t=0}^{N-1}(x_t^T Q x_t + u_t^T R u_t) \\ \text{s.t.} \quad & (x_t, u_t) \in \mathcal{X} \times \mathcal{U}, \qquad t = 0, \dots, N-1 \\ & x_{t+1} = Ax_t + Bu_t, \quad t = 0, \dots, N-2 \\ & x_0 = \bar{x}. \end{aligned} \tag{1}$$

Note that no terminal constraint set or terminal cost is present in the problem formulation. Stability and feasibility results for distributed MPC without terminal constraint set and terminal cost is presented in [10]. We introduce the following state and control variable partitions

$$x_t = [(x_t^1)^T, \dots, (x_t^M)^T]^T, \quad u_t = [(u_t^1)^T, \dots, (u_t^M)^T]^T$$

where $x_t^i \in \mathbb{R}^{n_i}$ and $u_t^i \in \mathbb{R}^{m_i}$ are referred to as local variables and $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$ are referred to as global variables. The dynamics matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$

are partitioned accordingly,

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MM} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & \cdots & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & \cdots & B_{MM} \end{pmatrix}$$

where $A_{ij} = \mathbb{R}^{n_i \times n_j}$ and $B_{ij} = \mathbb{R}^{n_i \times m_j}$. These matrices are assumed to have a sparse structure, i.e., that some $A_{ij} = 0$ and $B_{ij} = 0$. The neighboring interaction is defined by the following sets

$$\mathcal{N}_i = \{j \in \{1, \dots, M\} \mid A_{ij} \neq 0 \text{ or } B_{ij} \neq 0\},$$
$$\mathcal{M}_i = \{j \in \{1, \dots, M\} \mid A_{ji} \neq 0 \text{ or } B_{ji} \neq 0\}$$

which gives the local dynamics

$$x_{t+1}^i = \sum_{j \in \mathcal{N}_i} \left(A_{ij} x_t^j + B_{ij} u_t^j \right), \qquad x_0^i = \bar{x}_i$$

for $i = 1, \dots, M$. The global constraint sets are assumed to be products of local sets, i.e.,

$$\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_M, \qquad \mathcal{U} = \mathcal{U}_1 \times \ldots \times \mathcal{U}_M$$

where the local constraint sets $\mathcal{X}_i$ and $\mathcal{U}_i$ for $i = 1, \dots, M$ are bounded polytopes containing zero in their respective interiors. The local constraint sets can be represented as

$$\mathcal{X}_i = \{x^i \in \mathbb{R}^{n_i} \mid C_x^i x^i \leq d_x^i\},$$
$$\mathcal{U}_i = \{u^i \in \mathbb{R}^{m_i} \mid C_u^i u^i \leq d_u^i\}$$

where $C_x^i \in \mathbb{R}^{n_{c_{x^i}} \times n_i}$, $C_u^i \in \mathbb{R}^{n_{c_{u^i}} \times m_i}$, $d_x^i \in \mathbb{R}_{>0}^{n_{c_{x^i}}}$ and $d_u^i \in \mathbb{R}_{>0}^{n_{c_{u^i}}}$. We define the total number of inequalities in $\mathcal{X}$ and $\mathcal{U}$ by $n_c = \sum_i (n_{c_{x^i}} + n_{c_{u^i}})$. The quadratic cost function in (1) is assumed separable, i.e., $Q = \text{blkdiag}(Q_1, \dots, Q_M)$ and $R = \text{blkdiag}(R_1, \dots, R_M)$ where $Q_i \in \mathbb{R}^{n_i \times n_i}$ and $R_i \in \mathbb{R}^{m_i \times m_i}$ for $i = 1, \dots, M$ are symmetric positive definite matrices. We create the stacked vectors

$$\mathbf{y}_i = [(x_1^i)^T, \dots, (x_{N-1}^i)^T, (u_0^i)^T, \dots, (u_{N-1}^i)^T]^T$$

for $i = 1, \dots, M$ and $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_M^T]^T$. This implies that the optimization (1) problem can more compactly be written as

$$\begin{aligned} V_N(\bar{x}) := \quad & \min_{\mathbf{y}} \quad \tfrac{1}{2}\mathbf{y}^T \mathbf{H} \mathbf{y} \\ & \text{s.t.} \quad \mathbf{A}\mathbf{y} = \mathbf{b}\bar{x} \\ & \qquad \mathbf{C}\mathbf{y} \leq \mathbf{d} \end{aligned} \tag{2}$$

where

$$\mathbf{H} = \text{blkdiag}(\mathbf{H}_1, \dots, \mathbf{H}_M), \qquad \bar{x} = [\bar{x}_1^T, \dots, \bar{x}_M^T]^T,$$
$$\mathbf{A} = [\mathbf{A}_1^T, \dots, \mathbf{A}_M^T]^T, \qquad \mathbf{b} = [\mathbf{b}_1^T, \dots, \mathbf{b}_M^T]^T,$$
$$\mathbf{C} = \text{blkdiag}(\mathbf{C}_1, \dots, \mathbf{C}_M), \qquad \mathbf{d} = [\mathbf{d}_1^T, \dots, \mathbf{d}_M^T]^T$$

and

$$\mathbf{H}_i = \text{blkdiag}(Q_i, \dots, Q_i, R_i, \dots, R_i),$$
$$\mathbf{A}_i = [\mathbf{A}_{i1}, \dots, \mathbf{A}_{iM}],$$
$$\mathbf{A}_{ij} = \begin{bmatrix} 0 & & & & B_{ij} & & & \\ A_{ij} & \ddots & & & & \ddots & & \\ & \ddots & \ddots & & & & \ddots & \\ & & A_{ij} & 0 & & & & B_{ij} \end{bmatrix}, \quad j \neq i$$
$$\mathbf{A}_{ii} = \begin{bmatrix} -I & & & & B_{ii} & & & \\ A_{ii} & \ddots & & & & \ddots & & \\ & \ddots & \ddots & & & & \ddots & \\ & & A_{ii} & -I & & & & B_{ii} \end{bmatrix},$$
$$\mathbf{b}_i = [\mathbf{b}_{i1}, \dots, \mathbf{b}_{iM}],$$
$$\mathbf{b}_{ij} = [-A_{ij}^T, 0, \dots, 0]^T,$$
$$\mathbf{C}_i = \text{blkdiag}(C_x^i, \dots, C_x^i, C_u^i, \dots, C_u^i),$$
$$\mathbf{d}_i = [(d_x^i)^T, \dots, (d_x^i)^T, (d_u^i)^T, \dots, (d_u^i)^T]^T$$

where $\mathbf{A}_{ij} = 0$ and $\mathbf{b}_{ij} = 0$ if $j \notin \mathcal{N}_i$. We introduce dual variables $\boldsymbol{\lambda} \in \mathbb{R}^{n(N-1)}$ for the equality constraints and dual variables $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{N n_c}$ for the inequality constraints to get the following dual problem

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} \min_{\mathbf{y}} \frac{1}{2}\mathbf{y}^T \mathbf{H} \mathbf{y} + \boldsymbol{\lambda}^T (\mathbf{A}\mathbf{y} - \mathbf{b}\bar{x}) + \boldsymbol{\mu}^T (\mathbf{C}\mathbf{y} - \mathbf{d}). \tag{3}$$

As shown in [9], the inner minimization problem can be solved explicitly which gives the following dual problem

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} -\frac{1}{2}(\mathbf{A}^T\boldsymbol{\lambda} + \mathbf{C}^T\boldsymbol{\mu})^T \mathbf{H}^{-1} (\mathbf{A}^T\boldsymbol{\lambda} + \mathbf{C}^T\boldsymbol{\mu}) - \\ - \boldsymbol{\lambda}^T \mathbf{b}\bar{x} - \boldsymbol{\mu}^T \mathbf{d}. \tag{4}$$

We define the dual function for initial condition $\bar{x}$ as

$$D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := -\frac{1}{2}(\mathbf{A}^T\boldsymbol{\lambda} + \mathbf{C}^T\boldsymbol{\mu})^T \mathbf{H}^{-1} (\mathbf{A}^T\boldsymbol{\lambda} + \mathbf{C}^T\boldsymbol{\mu}) - \\ - \boldsymbol{\lambda}^T \mathbf{b}\bar{x} - \boldsymbol{\mu}^T \mathbf{d} \tag{5}$$

which is concave and differentiable with gradient

$$\nabla D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = -\begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \mathbf{H}^{-1} (\mathbf{A}^T\boldsymbol{\lambda} + \mathbf{C}^T\boldsymbol{\mu}) - \begin{bmatrix} \mathbf{b}\bar{x} \\ \mathbf{d} \end{bmatrix}. \tag{6}$$

A. Assumptions and definitions

We define by $\mathbb{X}_N$ the set of initial conditions for which (2) is feasible. We also define

$$P := \mathbf{b}^T (\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T)^{-1} \mathbf{b}$$

which characterizes the optimal solution without inequality constraints since

$$\frac{1}{2}\bar{x}^T P \bar{x} = \max_{\boldsymbol{\lambda}} D_N(\bar{x}, \boldsymbol{\lambda}, 0) \leq V_N(\bar{x}). \tag{7}$$

We also introduce the following definition.

Definition 1: We define $\kappa \geq 1$ as the smallest scalar such that for every $\bar{x} \in \mathbb{X}_N$ the following holds

$$V_N(\bar{x}) \leq \frac{\kappa}{2}\bar{x}^T P \bar{x}.$$

Assumption 1: We assume that $\mathbf{A}$ has full row rank and that $\mathbf{A}^T\mathbf{A} + \mathbf{C}^T\mathbf{C}$ is invertible.

B. Notation

We use the following norm notation $\|x\|_L = \sqrt{x^T L x}$ and $\|x\| = \sqrt{x^T x}$ and inner product $\langle x, y\rangle = x^T y$. Also, $[\cdot]_i$ denotes the i:th element in a vector.

III. Distributed algorithm

In this section we show how the generalized accelerated gradient method presented in [24] can be used in distributed model predictive control. The generalized accelerated gradient method can be applied to problems of the form

$$\min_{x\in\mathcal{X}} f(x)$$

where $\mathcal{X}$ is a closed, convex, and non-empty set and $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable. Further, f should satisfy

$$f(x_1) \leq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle + \frac{1}{2}\|x_1 - x_2\|_L^2 \quad (8)$$

for every $x_1, x_2 \in \mathbb{R}^n$ where L is a positive definite matrix. The generalized accelerated gradient algorithm is defined by the iterations

$$v^k = x^k + \frac{k-1}{k+2}(x^k - x^{k-1})$$
$$x^{k+1} = \arg\min_{x\in\mathcal{X}} \left[f(v^k) + \langle \nabla f(v^k), x - v^k\rangle + \frac{1}{2}\|x - v^k\|_L^2 \right]$$

where k is the iteration number. Straightforward verification gives that these iterations can equivalently be written as

$$v^k = x^k + \frac{k-1}{k+2}(x^k - x^{k-1}) \quad (9)$$
$$x^{k+1} = \arg\min_{x\in\mathcal{X}} \left(\left\| x - v^k + L^{-1}\nabla f(v^k)\right\|_L^2 \right). \quad (10)$$

We see that L^{-1} serves as a step matrix for the gradient. The algorithm is a generalization of the algorithm in [1] with the difference is that in [1], L is restricted to being a multiple of the identity matrix.

Remark 1: The convergence of the algorithm depends on the quadratic upper bound (8) to f. The tighter this upper bound, the fewer iterations can be expected. For L being a multiple of the identity matrix, the quadratic part of the upper bound has the same curvature in every direction, which typically leads to bad convergence rate for ill-conditioned problems. For an appropriately chosen L-matrix the quadratic upper bound to f becomes tighter and a better convergence rate is expected.

In the following proposition we show how L should be chosen to satisfy (8) for $f = -D_N$. Before the result is stated, we introduce the matrix

$$T := [\mathbf{A}^T\mathbf{C}^T]^T\mathbf{H}^{-1}[\mathbf{A}^T\mathbf{C}^T]. \quad (11)$$

Proposition 1: Every positive definite matrix L that satisfies $L \succeq T$ satisfies (8) for $f = -D_N$ where D_N is defined in (5).

Proof. We introduce $g = [(\mathbf{b}\bar{x})^T\ \mathbf{d}]^T$, $z = [\boldsymbol{\lambda}^T\boldsymbol{\mu}^T]^T$ and $\widetilde{D}_N(\bar{x}, z) = -z^T T z - g^T z$. For every $z_1, z_2 \in \mathbb{R}^{N(n+n_c)-n}$ the following holds

$$\begin{aligned}
\frac{1}{2}\|z_1 - z_2\|_L^2 &\geq \frac{1}{2}\|z_1 - z_2\|_T^2 \\
&= \frac{1}{2}z_1^T T z_1 + \frac{1}{2}z_2^T T z_2 - z_1^T T z_2 \\
&= -\widetilde{D}_N(\bar{x}, z_1) - g^T z_1 - \frac{1}{2}z_2^T T z_2 - \\
&\quad - \langle T z_2, z_1 - z_2\rangle \\
&= -\widetilde{D}_N(\bar{x}, z_1) - g^T z_1 + \widetilde{D}_N(\bar{x}, z_2) + g^T z_2 + \\
&\quad + \left\langle \nabla\widetilde{D}_N(\bar{x}, z_2), z_1 - z_2\right\rangle + g^T(z_1 - z_2) \\
&= -\widetilde{D}_N(\bar{x}, z_1) + D_N(\bar{x}, z_2) + \\
&\quad + \left\langle \nabla\widetilde{D}_N(\bar{x}, z_2), z_1 - z_2\right\rangle.
\end{aligned}$$

Since $\widetilde{D}_N(\bar{x}, z) = D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ if $z = [\boldsymbol{\lambda}^T\boldsymbol{\mu}^T]^T$ and since D_N is concave we have that $f = -D_N$ is convex and satisfies (8). This concludes the proof. □

We have shown that the generalized accelerated gradient method can be applied to solve the dual problem provided that the matrix L satisfies $L \succeq T = [\mathbf{A}^T\mathbf{C}^T]^T\mathbf{H}^{-1}[\mathbf{A}^T\mathbf{C}^T]$. The following semidefinite program can be used to compute such an L-matrix:

$$\begin{aligned}
&\min_{L\in\mathcal{L}} \operatorname{tr}(L) \\
&\text{s.t. } L \succeq [\mathbf{A}^T\mathbf{C}^T]^T\mathbf{H}^{-1}[\mathbf{A}^T\mathbf{C}^T] \\
&\qquad L \succ 0
\end{aligned} \quad (12)$$

where $\mathcal{L}$ defines some structural constraint on the L-matrix.

To apply the generalized accelerated gradient method, defined by iterations (9)-(10), to solve the dual problem (4) we introduce the dual variable iterations $\boldsymbol{\lambda}^k$ and $\boldsymbol{\mu}^k$ where k is the iteration number and $\bar{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k + \frac{k-1}{k+2}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1})$ and $\bar{\boldsymbol{\mu}}^k = \boldsymbol{\mu}^k + \frac{k-1}{k+2}(\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1})$. We also define primal variable iterations as $\mathbf{y}^k = -\mathbf{H}^{-1}(\mathbf{A}^T\boldsymbol{\lambda}^k + \mathbf{C}^T\boldsymbol{\mu}^k)$ and $\bar{\mathbf{y}}^k = \mathbf{y}^k + \frac{k-1}{k+2}(\mathbf{y}^k - \mathbf{y}^{k-1})$. By insertion into (6), the dual function gradient becomes

$$\nabla D_N(\bar{x}, \bar{\boldsymbol{\lambda}}^k, \bar{\boldsymbol{\mu}}^k) = \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}\bar{\mathbf{y}}^k - \begin{bmatrix}\mathbf{b}\bar{x}\\ \mathbf{d}\end{bmatrix}.$$

By restricting the set of L matrices to be of the form $L = \operatorname{blkdiag}(L_\lambda, L_\mu)$ it can be verified that the iterations (9)-(10) when applied to the dual problem (4) becomes

$$\mathbf{y}^k = -\mathbf{H}^{-1}(\mathbf{A}^T\boldsymbol{\lambda}^k + \mathbf{C}^T\boldsymbol{\mu}^k) \quad (13)$$
$$\bar{\mathbf{y}}^k = \mathbf{y}^k + \frac{k-1}{k+2}(\mathbf{y}^k - \mathbf{y}^{k-1}) \quad (14)$$
$$\bar{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k + \frac{k-1}{k+2}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) \quad (15)$$
$$\boldsymbol{\lambda}^{k+1} = \bar{\boldsymbol{\lambda}}^k + L_\lambda^{-1}(\mathbf{A}\bar{\mathbf{y}}^k - \mathbf{b}\bar{x}) \quad (16)$$
$$\bar{\boldsymbol{\mu}}^k = \boldsymbol{\mu}^k + \frac{k-1}{k+2}(\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) \quad (17)$$
$$\boldsymbol{\mu}^{k+1} = \arg\min_{\mu\geq 0} \left(\left\|\mu - \bar{\boldsymbol{\mu}}^k - L_\mu^{-1}(\mathbf{C}\bar{\mathbf{y}}^k - \mathbf{d})\right\|_{L_\mu}^2 \right) \quad (18)$$

Remark 2: For diagonal L_μ the projection operation in (18) becomes very cheap, namely a max-operation for each element in $\boldsymbol{\mu}$. However, the number of iterations to achieve satisfactory accuracy might be significant. For, non-diagonal L_μ the projection operation is more computationally expensive but for appropriately chosen L a reduced number of iterations is expected. This is desirable in DMPC where the number of iterations, i.e., the amount of communication, should be kept as low as possible.

Remark 3: As opposed to classical gradient methods, fast gradient methods suffer from error accumulation [5]. Thus, the inner minimization problem must be solved accurately. Therefore, (12) should be formulated such that the condition number of the L-matrix does not become too large.

We introduce dual variable partitions $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^T, \dots, \boldsymbol{\lambda}_M^T]^T$ and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^T, \dots, \boldsymbol{\mu}_M^T]^T$ according to the division of the equality and inequality constraint matrices $\mathbf{A}$ and $\mathbf{C}$. By restricting the set of possible step matrices L_λ to $L_\lambda = \text{blkdiag}(L_\lambda^1, \dots, L_\lambda^M)$ and L_μ to $L_\mu = \text{blkdiag}(L_\mu^1, \dots, L_\mu^M)$, where the partitioning corresponds to the partitioning of $\mathbf{A}$ and $\mathbf{C}$, and by noting that

$$\mathbf{y}_i^k = -\mathbf{H}_i^{-1}\left(\left(\sum_{j\in\mathcal{M}_i} \mathbf{A}_{ji}^T \boldsymbol{\lambda}_j^k\right) + \mathbf{C}_i^T \boldsymbol{\mu}_i^k\right)$$

we get the following distributed algorithm.

Algorithm 1: **Distributed algorithm**

Initialize $\boldsymbol{\lambda}_i^0 = \boldsymbol{\lambda}_i^{-1}, \boldsymbol{\mu}_i^0 = \boldsymbol{\mu}_i^{-1}$ and $\mathbf{y}_i^0 = \mathbf{y}_i^{-1}$
In every node, i, the following computations are performed
For $k \geq 0$

1) Update primal variables according to

$$\mathbf{y}_i^k = -\mathbf{H}_i^{-1}\left(\left(\sum_{j\in\mathcal{M}_i} \mathbf{A}_{ji}^T \boldsymbol{\lambda}_j^k\right) + \mathbf{C}_i^T \boldsymbol{\mu}_i^k\right)$$
$$\bar{\mathbf{y}}_i^k = \mathbf{y}_i^k + \frac{k-1}{k+2}(\mathbf{y}_i^k - \mathbf{y}_i^{k-1})$$

2) Send $\bar{\mathbf{y}}_i^k$ to each $j \in \mathcal{M}_i$, receive $\bar{\mathbf{y}}_j^k$ from each $j \in \mathcal{N}_i$

3) Update dual variables according to

$$\bar{\boldsymbol{\lambda}}_i^k = \boldsymbol{\lambda}_i^k + \frac{k-1}{k+2}(\boldsymbol{\lambda}_i^k - \boldsymbol{\lambda}_i^{k-1})$$
$$\boldsymbol{\lambda}_i^{k+1} = \bar{\boldsymbol{\lambda}}_i^k + (L_\lambda^i)^{-1}\left(\sum_{j\in\mathcal{N}_i}(\mathbf{A}_{ij}\bar{\mathbf{y}}_j^k - \mathbf{b}_{ij}\bar{x}_j)\right)$$
$$\bar{\boldsymbol{\mu}}_i^k = \boldsymbol{\mu}_i^k + \frac{k-1}{k+2}(\boldsymbol{\mu}_i^k - \boldsymbol{\mu}_i^{k-1})$$
$$\boldsymbol{\mu}_i^{k+1} = \arg\min_{\mu\geq 0} \left\|\mu - \bar{\boldsymbol{\mu}}_i^k - (L_\mu^i)^{-1}(\mathbf{C}_i\bar{\mathbf{y}}_i^k - \mathbf{d}_i)\right\|_{L_\mu^i}^2$$

4) Send $\boldsymbol{\lambda}_i^{k+1}$ to each $j \in \mathcal{N}_i$, receive $\boldsymbol{\lambda}_j^{k+1}$ from each $j \in \mathcal{M}_i$

We introduce the set of optimal dual variables

$$M^*(\bar{x}) = \left\{\boldsymbol{\lambda} \in \mathbb{R}^{n_\lambda}, \boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{n_\mu} \;\middle|\; D_N(\bar{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \geq V_N(\bar{x})\right\}$$

where $n_\lambda = n(N-1)$ and $n_\mu = Nn_c$. The convergence rates for the dual function D_N and the primal variables when running Algorithm 1 are stated in the following theorem.

Theorem 1: Suppose that $\bar{x} \in \mathbb{X}_N$ and let $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$. Then Algorithm 1 has the following convergence rate properties:

1) For $k \geq 1$ the convergence rate for the dual function is

$$D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \leq \frac{2\left\|\begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix} - \begin{bmatrix}\boldsymbol{\lambda}^0 \\ \boldsymbol{\mu}^0\end{bmatrix}\right\|_L^2}{(k+1)^2}. \tag{19}$$

2) Let $\mathbf{y}^*(\bar{x})$ be the unique optimal solution to (2) with initial condition $\bar{x}$. For $k \geq 1$ the convergence rate is

$$\|\mathbf{y}^k - \mathbf{y}^*(\bar{x})\|_2^2 \leq \frac{4\left\|\begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix} - \begin{bmatrix}\boldsymbol{\lambda}^0 \\ \boldsymbol{\mu}^0\end{bmatrix}\right\|_L^2}{\sigma_{\min}(\mathbf{H})(k+1)^2} \tag{20}$$

where $\sigma_{\min}(\mathbf{H})$ is the smallest eigenvalue to $\mathbf{H}$.

Proof. Argument 1 is proven in [24] while argument 2 is a straightforward generalization of [9, Theorem 1(2)]. □

IV. Lagrange multiplier norm bounds

From Theorem 1 we conclude that a bound on the norm of the optimal dual variables is needed to bound the number of iterations necessary to achieve a prespecified dual accuracy. First, we state a result from [8] in which a bound on the optimal dual variables is presented. Before the result is presented we define $d_{\min} := \min_i[\mathbf{d}]_i$, $\Phi := \mathbf{A}\mathbf{H}^{-1}\mathbf{A}^T$ and $\Psi := \Phi^{-1}\mathbf{A}\mathbf{H}^{-1}\mathbf{C}^T$. The matrix Φ is invertible since $\mathbf{H}^{-1}$ has full rank and $\mathbf{A}$ has full row rank due to Assumption 1.

Lemma 1: For every $\bar{x} \in \beta\mathbb{X}_N$ where $\beta \in (0,1)$ we have that

$$\max_{(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\in M^*(\bar{x})} \left\|\begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix}\right\| \leq h_\beta(\bar{x}) \tag{21}$$

where

$$h_\beta(\bar{x}) := \left\|\begin{bmatrix}\Psi \\ I\end{bmatrix}\right\| \frac{\kappa - 1}{2(1-\beta)d_{\min}} \bar{x}^T P \bar{x} + \|\Phi^{-1}\mathbf{b}\bar{x}\|$$

and κ is defined in Definition 1.

For the cold starting case, i.e., with $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$, the convergence rates (19) and (20) depend on $\|[(\boldsymbol{\lambda}^*)^T, (\boldsymbol{\mu}^*)^T]^T\|_L$. A bound on this can be obtained by noting that $\|[(\boldsymbol{\lambda}^*)^T, (\boldsymbol{\mu}^*)^T]^T\|_L \leq \|L\|\|[(\boldsymbol{\lambda}^*)^T, (\boldsymbol{\mu}^*)^T]^T\|$ and using Lemma 1. However, this bound becomes quite conservative and a tighter bound can be computed. To achieve this, we introduce the following decomposition of the dual variables, $\boldsymbol{\lambda} = \boldsymbol{\lambda}_p + \boldsymbol{\lambda}_n$ and $\boldsymbol{\mu} = \boldsymbol{\mu}_p + \boldsymbol{\mu}_n$, where

$$\begin{bmatrix}\boldsymbol{\lambda}_p \\ \boldsymbol{\mu}_p\end{bmatrix} \perp \mathcal{N}\left([\mathbf{A}^T \;\; \mathbf{C}^T]\right), \quad \begin{bmatrix}\boldsymbol{\lambda}_n \\ \boldsymbol{\mu}_n\end{bmatrix} \in \mathcal{N}\left([\mathbf{A}^T \;\; \mathbf{C}^T]\right) \tag{22}$$

and $\mathcal{N}$ denotes the null-space. We denote by Z an orthonormal basis to the null-space of $[\mathbf{A}^T \;\; \mathbf{C}^T]$, i.e., $[\mathbf{A}^T \;\; \mathbf{C}^T]Z = 0$ and $Z^TZ = I$. Since the null-space to $[\mathbf{A}^T \;\; \mathbf{C}^T]$ is perpendicular to the range of $[\mathbf{A}^T \;\; \mathbf{C}^T]^T$ the decomposed dual variables can be represented as

$$\begin{bmatrix}\boldsymbol{\lambda}_p \\ \boldsymbol{\mu}_p\end{bmatrix} = \begin{bmatrix}\mathbf{A} \\ \mathbf{C}\end{bmatrix} \bar{z}_p, \qquad \begin{bmatrix}\boldsymbol{\lambda}_n \\ \boldsymbol{\mu}_n\end{bmatrix} = Z\bar{z}_n \tag{23}$$

where $\bar{z}_p$ and $\bar{z}_n$ are new variables of smaller dimension. The KKT conditions for the dual problem described by the decomposed dual variables are presented next.

Proposition 2: The KKT conditions to (4) are

$$-\mathbf{A}\mathbf{H}^{-1}(\mathbf{A}^T\boldsymbol{\lambda}_p^* + \mathbf{C}^T\boldsymbol{\mu}_p^*) = \mathbf{b}\bar{x} \tag{24}$$

$$-\mathbf{C}\mathbf{H}^{-1}(\mathbf{A}^T\boldsymbol{\lambda}_p^* + \mathbf{C}^T\boldsymbol{\mu}_p^*) = \mathbf{d} + s \tag{25}$$

$$s \leq 0 \ , \ \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^* \geq 0 \tag{26}$$

$$[(\boldsymbol{\mu}_p^*) + (\boldsymbol{\mu}_n^*)]_i[s]_i = 0 \tag{27}$$

where $\boldsymbol{\lambda}_p^*$, $\boldsymbol{\lambda}_n^*$, $\boldsymbol{\mu}_p^*$ and $\boldsymbol{\mu}_n^*$ satisfy (22) and the optimal dual variables $\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*$ satisfy $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_p^* + \boldsymbol{\lambda}_n^*$ and $\boldsymbol{\mu}^* = \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^*$.

Proof. The result is immediate from the KKT conditions [4, §5.5.3], the dual variable decomposition $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_p^* + \boldsymbol{\lambda}_n^*$, $\boldsymbol{\mu}^* = \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^*$, and due to (22) which implies that $\mathbf{A}^T\boldsymbol{\lambda}_n^* + \mathbf{C}^T\boldsymbol{\mu}_n^* = 0$. □

Remark 4: The variables $\boldsymbol{\lambda}_p^*$ and $\boldsymbol{\mu}_p^*$ satisfy the stationarity conditions while $\boldsymbol{\lambda}_n^*$ and $\boldsymbol{\mu}_n^*$ do not affect the stationarity conditions but instead ensure dual feasibility and complementarity.

Before we present bounds on the decomposed dual variables, we define ζ as the smallest positive scalar such that

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}^T L \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix} \preceq \zeta \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}^T T \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}. \tag{28}$$

where T is defined in (11). Such finite ζ exists since by Assumption 1 $\mathbf{A}^T\mathbf{A} + \mathbf{C}^T\mathbf{C}$ is invertible and

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}^T T \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix} = (\mathbf{A}^T\mathbf{A} + \mathbf{C}^T\mathbf{C})\mathbf{H}^{-1}(\mathbf{A}^T\mathbf{A} + \mathbf{C}^T\mathbf{C})$$

which is positive definite since $\mathbf{H}^{-1}$ is positive definite. In the following lemma, bounds for the decomposed optimal dual variables are presented.

Lemma 2: Suppose that $\bar{x} \in \beta\mathbb{X}_N$ and $\beta \in (0,1)$. Then

$$\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}^T L \begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix} \leq \kappa\zeta\bar{x}^T P\bar{x} \tag{29}$$

and

$$\begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix}^T L \begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix} \leq \|Z^T L Z\| \left((h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P\bar{x}}{\|L\|}\right) \tag{30}$$

hold for every $\boldsymbol{\lambda}_p^*$, $\boldsymbol{\lambda}_n^*$, $\boldsymbol{\mu}_p^*$ and $\boldsymbol{\mu}_n^*$ that satisfies (22) and the KKT conditions (24)-(27).

Proof. To show (29) we have

$$\begin{aligned}
\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}^T L \begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix} &= \bar{z}_p^T \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}^T L \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix} \bar{z}_p \\
&\leq \zeta \bar{z}_p^T \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}^T T \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix} \bar{z}_p \\
&= \zeta \begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}^T T \begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix} \\
&= \zeta \begin{bmatrix}\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}^* - \boldsymbol{\mu}_n^*\end{bmatrix}^T \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix} \mathbf{H}^{-1} \begin{bmatrix}\mathbf{A}\\ \mathbf{C}\end{bmatrix}^T \begin{bmatrix}\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}^* - \boldsymbol{\mu}_n^*\end{bmatrix} \\
&= \zeta \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix}^T T \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix}
\end{aligned} \tag{31}$$

where the first equality comes from (23), the first inequality from (28), the second equality from (23) the third equality holds since $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_p^* + \boldsymbol{\lambda}_n^*$ and $\boldsymbol{\mu}^* = \boldsymbol{\mu}_p^* + \boldsymbol{\mu}_n^*$ and due to (11) and the last equality is due to (22) which implies $\mathbf{A}^T\boldsymbol{\lambda}_n^* + \mathbf{C}^T\boldsymbol{\mu}_n^* = 0$.

Further, the KKT conditions for the dual problem (24)-(25) give that

$$0 = T \begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix} + \begin{bmatrix}\mathbf{b}\bar{x}\\ \mathbf{d} + s\end{bmatrix} = T \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix} + \begin{bmatrix}\mathbf{b}\bar{x}\\ \mathbf{d} + s\end{bmatrix}.$$

This implies that

$$\begin{aligned}
0 &= [(\boldsymbol{\lambda}^*)^T(\boldsymbol{\mu}^*)^T] \left(T \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix} + \begin{bmatrix}\mathbf{b}\bar{x}\\ \mathbf{d} + s\end{bmatrix}\right) \\
&= [(\boldsymbol{\lambda}^*)^T(\boldsymbol{\mu}^*)^T] T \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix} + \bar{x}^T\mathbf{b}^T\boldsymbol{\lambda}^* + (s + \mathbf{d})^T\boldsymbol{\mu}^* \\
&= -V_N(\bar{x}) + \frac{1}{2}[(\boldsymbol{\lambda}^*)^T(\boldsymbol{\mu}^*)^T] T \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix}
\end{aligned} \tag{32}$$

where $s^T\boldsymbol{\mu}^* = 0$ from (27) is used in the final equality. Using (31) and (32) we get

$$\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}^T L \begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix} \leq \zeta \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix}^T T \begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix} = \zeta 2V_N(\bar{x}) \leq \zeta\kappa\bar{x}^T P\bar{x}$$

where Definition 1 is used in the last inequality. This proves (29).

Next we show that (30) holds. From (22) we have that $[(\boldsymbol{\lambda}_p^*)^T \ (\boldsymbol{\mu}_p^*)^T][(\boldsymbol{\lambda}_n^*)^T \ (\boldsymbol{\mu}_n^*)^T]^T = 0$, hence Pythagoras' theorem implies that

$$\left\|\begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix}\right\|^2 = \left\|\begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix}\right\|^2 - \left\|\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}\right\|^2. \tag{33}$$

Further,

$$\|L\|\left\|\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}\right\|^2 \geq \left\|\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}\right\|_L^2 \geq \left\|\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}\right\|_T^2 = 2V_N(\bar{x}) \geq \bar{x}^T P\bar{x} \tag{34}$$

where the equality comes from (32) and the final inequality comes from (7). By applying Lemma 1 and (34) to (33), we get

$$\left\|\begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix}\right\|^2 = \left\|\begin{bmatrix}\boldsymbol{\lambda}^*\\ \boldsymbol{\mu}^*\end{bmatrix}\right\|^2 - \left\|\begin{bmatrix}\boldsymbol{\lambda}_p^*\\ \boldsymbol{\mu}_p^*\end{bmatrix}\right\|^2 \leq (h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P\bar{x}}{\|L\|}. \tag{35}$$

Further, from (23) we have

$$\begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix} = Z\bar{z}_n = Z(Z^TZ)^{-1}Z^T \begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix} = ZZ^T \begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix}$$

since $Z^TZ = I$. This implies

$$\begin{aligned}
\begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix}^T L \begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix} &= \begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix}^T ZZ^TLZZ^T \begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix} \\
&\leq \|ZZ^TLZZ^T\| \left\|\begin{bmatrix}\boldsymbol{\lambda}_n^*\\ \boldsymbol{\mu}_n^*\end{bmatrix}\right\|^2 \\
&\leq \|Z^TLZ\| \left((h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P\bar{x}}{\|L\|}\right)
\end{aligned}$$

where the last equality holds since $Z^TZ = I$ and due to (35). This concludes the proof. □

Using Lemma 2, we are now ready to state the following theorem on dual variable bounds.

Theorem 2: Suppose that $\bar{x} \in \beta\mathbb{X}_N$ and $\beta \in (0,1)$. Then for every $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in M^*(\bar{x})$ we have

$$\begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix}^T L \begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix} \leq \left(\sqrt{\|Z^T L Z\| \left[(h_\beta(\bar{x}))^2 - \frac{\bar{x}^T P \bar{x}}{\|L\|}\right]} + \right. \\ \left. + \sqrt{\kappa\zeta \bar{x}^T P \bar{x}}\right)^2.$$

Proof. Using the triangle inequality we get

$$\begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix}^T L \begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix} = \left\|\begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix}\right\|_L^2 \leq \left(\left\|\begin{bmatrix}\boldsymbol{\lambda}_p^* \\ \boldsymbol{\mu}_p^*\end{bmatrix}\right\|_L + \left\|\begin{bmatrix}\boldsymbol{\lambda}_n^* \\ \boldsymbol{\mu}_n^*\end{bmatrix}\right\|_L\right)^2.$$

Insertion of the corresponding bounds in Lemma 2 gives the result. □

Most conservatism in the dual variable bound comes from the function h_β, which originates from the estimate of the dual variable bound in Lemma 1. In Theorem 2 the function h_β is multiplied by $\|Z^T L Z\|$. If L approximates T well, it is anticipated that $\|Z^T L Z\|$ becomes small which gives improved bounds compared to using $\|L\| h_\beta(\bar{x})$.

V. Iteration bounds

The dual variable bounds presented in the previous section can be used to bound the number of iterations necessary to guarantee a prespecified accuracy of the dual function value and the primal variables. However, for space considerations we omit the primal variable iteration bound result, which is derived similarly to the dual function iteration bound. In the following theorem we present an iteration bound for the cold starting case. We have used a relative accuracy of the optimization problem to avoid that a scaling of the cost-matrices affects the iteration bound.

Theorem 3: Suppose that $\bar{x} \in \beta\mathbb{X}_N$ and $\beta \in (0,1)$ and that Algorithm 1 is cold-started, i.e., initialized with $\boldsymbol{\lambda}^0 = 0$, $\boldsymbol{\mu}^0 = 0$, and $\mathbf{y}^0 = 0$. Then the dual function satisfies

$$D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - D_N(\bar{x}, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) \leq \epsilon_v D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \quad (36)$$

for every $k \geq k_v(\bar{x})$ where

$$k_v(\bar{x}) = \frac{2}{\sqrt{\epsilon_v}} \left(\sqrt{\|Z^T L Z\| \left[\frac{h_\beta(\bar{x})^2}{\bar{x}^T P \bar{x}} - \frac{1}{\|L\|}\right]} + \sqrt{\kappa\zeta}\right) - 1. \quad (37)$$

Proof. For the cold starting case we have $\boldsymbol{\lambda}^0 = 0$ and $\boldsymbol{\mu}^0 = 0$. Due to Theorem 1 and since $\frac{1}{2}\bar{x}^T P \bar{x} \leq D_N(\bar{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ we conclude that if k is such that

$$\frac{2}{(k+1)^2} \left\|\begin{bmatrix}\boldsymbol{\lambda}^* \\ \boldsymbol{\mu}^*\end{bmatrix}\right\|_L^2 \leq \epsilon_v \frac{1}{2} \bar{x}^T P \bar{x} \quad (38)$$

then (36) holds. Insertion of the bound in Theorem 2 into (38) and rearranging the terms gives the result. □

To compute a bound that holds for all $\bar{x} \in \beta\mathbb{X}_N$, (37) is maximized subject to $\bar{x} \in \beta\mathbb{X}_N$. A more conservative bound is obtained by removing $1/\|L\|$ from (37) which gives the following maximization problem

$$\max_{\bar{x} \in \beta\mathbb{X}_N} \frac{2}{\sqrt{\epsilon_v}} \left(\sqrt{\|Z^T L Z\|} \left(\rho\sqrt{\bar{x}^T P \bar{x}} + \gamma\right) + \sqrt{\kappa\zeta}\right) \quad (39)$$

where

$$\gamma = \|\Phi^{-1} \mathbf{b} P^{-1/2}\|, \qquad \rho = \left\|\begin{bmatrix}\Psi \\ I\end{bmatrix}\right\| \frac{\kappa - 1}{2(1-\beta) d_{\min}}.$$

An over-estimator to (39) can be computed by optimizing over $\beta\mathcal{X}$, which satisfies $\beta\mathbb{X}_N \subseteq \beta\mathcal{X}$. This is beneficial since $\mathbb{X}_N$ might be difficult to express explicitly and $\mathcal{X}$ is of lower complexity. The resulting optimization problem depends affinely on $\sqrt{\bar{x}^T P \bar{x}}$. Hence, the maximizing $\bar{x}$ can be computed by maximizing $\bar{x}^T P \bar{x}$ over $\beta\mathcal{X}$ which is a quadratic maximization problem over a polytopic set. Such maximization problems are known to be NP-complete, but can be rewritten as a mixed integer linear program (MILP) as shown in [11, Lemma 2] for which efficient solvers exist. In every iteration, MILP-software produce upper and lower bounds to the optimal value. To compute an iteration bound, an upper bound to the objective is enough. This implies that the MILP optimization can be stopped when sufficient accuracy has been achieved.

VI. Numerical example

We evaluate the efficiency of the proposed distributed optimization algorithm and the conservatism of the iteration bound by applying it to a dynamical system with sparse structure that is randomly generated. The largest eigenvalue of the dynamics matrix is 1.1, i.e., the system is unstable. The system has 3 subsystems with 5 states and 1 input each, i.e., 15 states and 3 inputs in all. The state and input variables are bounded from above and below by random numbers in the interval $[0.5\ 1.5]$ and $[-0.15\ -0.05]$ respectively. The cost matrices Q and R are diagonal and each diagonal element is randomly chosen from the interval $[1\ 100]$ and the control horizon is chosen to $N = 6$. A full specification of the problem is found in [7, Supplement A.1].

The problem is solved using three different methods: Algorithm 1 with a block-diagonal L-matrix, the fast gradient method (FGM) in [9] which is optimal w.r.t. what can be achieved using gradient methods, and the classical gradient method (GM) that is traditionally used with dual decomposition. All simulations are performed in MATLAB and the semidefinite programs are solved through YALMIP [12] using SeDuMi [19] for the preconditioning and SDPNAL [23] (which is more memory-efficient than SeDuMi) for the L-matrix.

In Table I the number of iterations needed to achieve a prespecified dual accuracy using Algorithm 1, the fast gradient method (FGM), and a gradient method (GM) are compared. We also compare the iteration complexity bounds presented in Theorem 3 and the one presented [8] which was developed for the case where L is a multiple of the identity

TABLE I

EVALUATION OF ALGORITHM 1 BY COMPARISON TO THE FAST GRADIENT METHOD (FGM) AND THE GRADIENT METHOD (GM). THE NUMBER OF ALGORITHM ITERATIONS AND ITERATION COMPLEXITY BOUNDS FROM THEOREM 3 AND [8] ARE PRESENTED.

Alg.	ϵ_v	β	# iters		iter bound	
			avg.	max.	Thm 3	[8]
Alg. 1	0.005	0.25	6.15	11	574	5096
Alg. 1	0.005	0.50	7.02	13	1352	12483
Alg. 1	0.005	0.75	7.35	13	3686	34647
Alg. 1	0.005	0.90	7.48	13	10688	101136
FGM	0.005	0.25	149.51	252	-	-
FGM	0.005	0.50	155.78	262	-	-
FGM	0.005	0.75	159.84	330	-	-
FGM	0.005	0.90	164.84	370	-	-
GM	0.005	0.25	3235.10	4741	-	-
GM	0.005	0.50	3637.64	5107	-	-
GM	0.005	0.75	3783.02	6051	-	-
GM	0.005	0.90	3848.55	6540	-	-

matrix. The first column in Table I specifies the algorithm used. The second column specifies the duality tolerance and the third column specifies the set from which the initial conditions are chosen where β is the scaling factor, i.e., initial conditions are chosen from $\beta\mathbb{X}_N$. The fourth and fifth columns present average and max number of iterations while the sixth and seventh columns specify the iteration bounds. The comparison is obtained by solving the optimization problem for 10000 randomly generated initial conditions.

From Table I we conclude two things. The first is that Algorithm 1 significantly outperforms previous methods to solve DMPC optimization problems in distributed fashion, i.e., fast gradient methods and gradient methods. This is accomplished because second order information is incorporated into the distributed algorithm. The second conclusion is that the iteration bound presented in Theorem 3 is quite conservative. However, it is much less conservative than the bound in [8] for the presented algorithm.

VII. CONCLUSIONS AND FUTURE WORK

We have presented a distributed optimization algorithm for distributed MPC that reduces significantly the number of iterations compared to distributed optimization algorithms where only gradient information is used. The reason for this improved iteration complexity is that we have shown how to incorporate Hessian information into the distributed algorithm. Further, we have presented an iteration complexity bound for the proposed algorithm. A future work direction is to use the iteration complexity bound to optimally precondition the problem data, where optimally refers to the preconditioning that minimizes the provided iteration bound.

REFERENCES

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, October 2009.

[2] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

[3] 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):1–122, 2011.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[5] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. Submitted to *Mathematical Programming*, available `http://www.optimization-online.org/DB_FILE/2010/12/2865.pdf`, 2010.

[6] M.D. Doan, T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter. A distributed version of Han's method for DMPC using local communications only. *Control Engineering and Applied Informatics*, 11(3):6–15, 2009.

[7] P. Giselsson. *Gradient-Based Distributed Model Predictive Control*. PhD thesis, Department of Automatic Control, Lund University, Sweden, November 2012.

[8] P. Giselsson. Optimal preconditioning and iteration complexity bounds for gradient-based optimization in model predictive control. In *2013 American Control Conference*, Washington, D.C., 2013. Accepted for publication.

[9] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829–833, 2013.

[10] P. Giselsson and A. Rantzer. On feasibility, stability and performance in distributed model predictive control. *IEEE Transactions on Automatic Control*, 2013. Accepted for publication.

[11] C.N. Jones and M. Morari. Approximate Explicit MPC using Bilevel Optimization. In *European Control Conference*, Budapest, Hungary, August 2009.

[12] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[13] R.R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, April 2008.

[14] A. Nemirovsky and D. Yudin. *Informational complexity and efficient methods for solution of convex extremal problems*. Wiley, New York, 1983.

[15] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[16] Y. Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ékonom. i. Mat. Metody*, 24:509–517, 1988.

[17] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*. Springer Netherlands, 1 edition, 2003.

[18] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005.

[19] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[20] P. Trnka, J. Pekar, and V. Havlena. Application of distributed MPC to Barcelona water distribution network. In *Proceedings of the 18th IFAC World Congress*, Milan, Italy, 2011.

[21] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, Department of Mathematics, University of Washington, 2008.

[22] Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi. Decentralized model predictive control via dual decomposition. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 381 – 386, Cancun, Mexico, December 2008.

[23] X-Y. Zhao, D. Sun, and K-C. Toh. A Newton-cg augmented Lagrangian method for semidefinite programming. *SIAM J. on Optimization*, 20(4):1737–1765, January 2010.

[24] W. Zuo and Z. Lin. A generalized accelerated proximal gradient approach for total-variation-based image restoration. *IEEE Transactions on Image Processing*, 20(10):2748 –2759, October 2011.