Fast and Stable Nonconvex Constrained Distributed Optimization: The ELLADA Algorithm

Wentao Tang and Prodromos Daoutidis

***Abstract*—Distributed optimization, where the computations are performed in a localized and coordinated manner using multiple agents, is a promising approach for solving large-scale optimization problems, e.g., those arising in model predictive control (MPC) of large-scale plants. However, a distributed optimization algorithm that is computationally efficient, globally convergent, amenable to nonconvex constraints and general inter-subsystem interactions remains an open problem. In this paper, we combine three important modifications to the classical alternating direction method of multipliers (ADMM) for distributed optimization. Specifically, (i) an extra-layer architecture is adopted to accommodate nonconvexity and handle inequality constraints, (ii) equality-constrained nonlinear programming (NLP) problems are allowed to be solved approximately, and (iii) a modified Anderson acceleration is employed for reducing the number of iterations. Theoretical convergence towards stationary solutions and computational complexity of the proposed algorithm, named ELLADA, is established. Its application to distributed nonlinear MPC is also described and illustrated through a benchmark process system.**

***Index Terms*—Distributed optimization, nonconvex optimization, model predictive control, acceleration**

I. Introduction

DISTRIBUTED optimization [1]–[3] refers to methods of performing optimization using a distributed architecture – multiple networked agents are used for subsystems and necessary information among the agents is communicated to coordinate the distributed computation. An important desirable application of distributed optimization is in model predictive control (MPC), where control decisions are made through solving an optimal control problem minimizing the cost associated with the predicted trajectory in a future horizon subject to the system dynamics and operational constraints [4]. For large-scale systems, it is desirable to seek a decomposition (e.g., using community detection or network block structures [5]–[7]) and deploy distributed MPC strategies [8]–[10], which allows better performance than fully decentralized MPC by enabling coordination, while avoiding assembling and computing on a monolithic model in centralized MPC.

Despite efficient algorithms for solving monolithic nonlinear programming (NLP) problems in centralized MPC (e.g., [11]–[13]), extending them into distributed algorithms is nontrivial. A typical approach of distributed MPC is to iterate the control inputs among the subsystems (in sequence or in parallel) [14]–[16]. The input iteration routine is typically either semi-decentralized by implicitly assuming that the subsystems interact only through inputs and considering state coupling as disturbances, or semi-centralized by using moving-horizon predictions based on the entire system, which, however, contradicts the fact that the subsystem models should be usually packaged inside the local agents rather than shared over the entire system. Distributed MPC under truly localized model information is typically restricted to linear systems [17]–[19].

The authors are with the Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, U.S.A. (e-mails: tangx647@umn.edu, daout001@umn.edu).

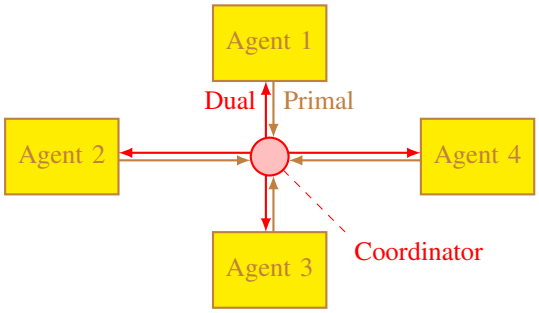

Fig. 1. Primal-dual distributed optimization.

We note that in general, distributed nonlinear MPC with subsystem interactions should be considered as a *distributed optimization problem under nonconvex constraints*. To solve such problems using distributed agents with local model knowledge, Lagrangian decomposition using dual relaxation of complicating interactions [20, Section 9] and the alternating direction method of multipliers (ADMM) algorithm using augmented Lagrangian [21], [22] were proposed as general frameworks. These are primal-dual iterative algorithms. As illustrated in Fig. 1, in each iteration, the distributed agents receive the dual information from the coordinator and execute subroutines to solve their own subproblems, and a coordinator collects information of their solutions to update the duals.

Convergence is the most basic requirement but also a major challenge in distributed optimization under nonconvex constraints. Although distributed optimization with nonconvex objective functions has been well discussed [23]–[26], the nonconvex constraints appear much more difficult to handle. To guarantee convergence, [27] suggested dualizing and penalizing all nonconvex constraints, making them undifferentiated and tractable by ADMM; however, this alteration of the problem structure eliminates the option for distributed agents to use any subroutine other than the method of multipliers (MM). [28] used a quadratic programming problem to decide the dual variables in the augmented Lagrangian as well as an extrapolation of primal updates; this algorithm, however, involves a central agent that extracts Hessian and gradient information of the subsystem models from the distributed agents

in every iteration, and is thus essentially semi-centralized. [29] adopted feasibility-preserving convex approximations to approach the solution, which is applicable to problems without nonconvex equality constraints. We note that several recent papers (e.g., [30]–[32]) proposed the idea of placing slack variables corresponding to the inter-subsystem constraints and forcing the decay to zero by tightening the penalty parameters of slack variables. This modification to the ADMM with slack variables and their penalties leads to a globally convergent *extra-layer* augmented Lagrangian-based algorithm with preserved agent-coordinator problem architecture.

Computational efficiency is also of critical importance for distributed optimization, especially in MPC. The slothfulness of primal-dual algorithms typically arises from two issues. First, the subgradient (first-order) update of dual variables restricts the number of iterations to be of linear complexity [33]–[35]. For convex problems, momentum methods [36], [37] can be adopted to obtain second-order dual updates. Such momentum acceleration can not be directly extended to nonconvex problems without a positive definite curvature, although our previous numerical study showed that a discounted momentum may allow limited improvement [38]. Another effort to accelerate dual updates in convex ADMM is based on Krylov subspace methods [39]. Under nonconvexity, it was only very recently realized that *Anderson acceleration*, a multi-secant technique for fixed-point problems, can be generally used to accelerate the dual variables [40]–[42].

The second cause for the high computational cost of distributed optimization is the instruction on the distributed agents to fully solve their subproblems to high precision in each iteration. Such exhaustive efforts may be unnecessary since the dual information to be received from the coordinator will keep changing. For convex problems, it is possible to linearize the augmented Lagrangian and replace the distributed subproblems with Arrow-Hurwicz-Uzawa gradient flows [43]. In the presence of nonconvexity of the objective functions, a dual perturbation technique to restore the convergence of the augmented Lagrangian was proposed in [44]. It is yet unknown how to accommodate such gradient flows to nonconvex constraints. A different approach is to allow inexact solution of the subproblems with adaptively tightening tolerances [45]. Such an *approximate ADMM algorithm* allows a better balance between the primal and dual updates, and avoids wasteful computational steps inside the subroutines.

The purpose of this work is to develop a convergent and computationally efficient algorithm for distributed optimization under nonconvex constraints. Although the algorithm is in principle not restricted to any specific problem, we consider the implementation of distributed nonlinear MPC as an important application. Based on the above discussion, we identify the following modifications to the classical ADMM algorithm as the key to mitigating the challenges in convergence and computational complexity: (i) additional slack variables are placed on the constraints relating the distributed agents and the coordinator, (ii) approximate optimization is performed in the distributed agents, and (iii) the Anderson acceleration technique is adopted by the coordinator. We therefore combine and extend as appropriate these techniques into a new algorithm with a two-layer augmented Lagrangian-based architecture, in which the outer layer handles the slack variables as well as inequality constraints by using a barrier technique, and the inner layer performs approximate ADMM under an acceleration scheme. With guaranteed stability and elevated speed, *to the best knowledge of the authors, the proposed algorithm is the first practical and generic algorithm of its kind for distributed nonlinear MPC with truly localized model information.* We name this algorithm as **ELLADA** (standing for **e**xtra-**l**ayer augmented **L**agrangian-based **a**ccelerated **d**istributed **a**pproximate optimization).

The paper discusses the movitation, develops the ELLADA algorithm and establishes its theoretical properties. An application to a benchmark quadruple tank process is also presented. The remainder of this paper is organized as follows. In Section II, we first review the classical ADMM and its modified versions. Then we derive our ELLADA algorithm in Section III with a trilogy pattern. First, a basic two-layer augmented Lagrangian-based algorithm (ELL) is introduced and its convergence is discussed. Then the approximate solution of equality-constrained NLP problems and the Anderson acceleration scheme are incorporated to form the ELLA and ELLADA algorithms. The implementation of the ELLADA algorithm on the distributed optimization problem involved in distributed nonlinear MPC is shown in Section IV, and the case study is examined in Section V. Conclusions and discussions are given in Section VI.

II. ADMM and Its Modifications

A. ADMM

The alterating direction method of multipliers is the most commonly used algorithm for distributed optimization under linear equality constraints [1]. Specifically, consider the following problem

$$\min_{x,\bar{x}} \;\; f(x)+g(\bar{x}) \quad \text{s.t.} \;\; Ax+B\bar{x}=0 \tag{1}$$

with two blocks of variables x and $\bar{x}$, where f and g are usually assumed to be convex. (The symbols in (1) are not related to the ones in Section IV.) The augmented Lagrangian for such a constrained optimization problem is

$$L(x,\bar{x};y)=f(x)+g(\bar{x})+y^\top(Ax+B\bar{x})+\frac{\rho}{2}\|Ax+B\bar{x}\|^2, \tag{2}$$

in which y stands for the vector of dual variables (Lagrangian multipliers) and $\rho > 0$ is called the penalty parameter. According to the duality theory, the optimal solution should be determined by a saddle point of the augmented Lagrangian:

$$\sup_{y}\min_{x,\bar{x}} \;\; L(x,\bar{x};y). \tag{3}$$

The classical method of multipliers (MM) deals with this saddle point problem with an iterative procedure, where the primal variables are optimized first and then the dual variables are updated with a subgradient ascent [46, Chapter 6]:

$$\begin{aligned}(x^{k+1},\bar{x}^{k+1}) &= \arg\min_{x,\bar{x}} L(x,\bar{x};y^k),\\ y^{k+1} &= y^k+\rho(Ax^{k+1}+B\bar{x}^{k+1}),\end{aligned} \tag{4}$$

in which the superscript stands for the count of iterations. In a distributed context, x and $\bar{x}$ usually can not be optimized simultaneously. ADMM is thus an approximation of MM that allows the optimization of x and $\bar{x}$ to be performed separately, i.e.,

$$\begin{aligned} x^{k+1} &= \arg\min_{x} L(x, \bar{x}^k; y^k), \\ \bar{x}^{k+1} &= \arg\min_{\bar{x}} L(x^k, \bar{x}; y^k), \\ y^{k+1} &= y^k + \rho(Ax^{k+1} + B\bar{x}^{k+1}). \end{aligned} \tag{5}$$

Since the appearance of ADMM in 1970s [47], [48], there have been many works regarding its theoretical properties, extensions and applications. As we have mentioned in the Introduction, ADMM is known to have a linear convergence rate for convex problems. This does not change when the variables are constrained in convex sets. For example, if $x \in \mathcal{X}$, it suffices to modify the corresponding term $f(x)$ in the objective function by adding an indicator function $\mathbb{I}_{\mathcal{X}}(x)$ (equal to 0 if $x \in \mathcal{X}$ and $+\infty$ otherwise), which is still a convex function.

B. ADMM with approximate updates

Unless the objective terms $f(x)$ and $g(\bar{x})$ are of simple forms such as quadratic functions, the optimization of x and $\bar{x}$ in (5) does not have an exact solution. Usually, iterative algorithms for nonlinear programming need to be called for the first two lines of (5), and always searching for a highly accurate solution in each ADMM iteration will result in an excessive computational cost. It is thus desirable to solve the optimization subproblems in ADMM inexactly when the dual variables are yet far from the optimum, i.e., to allow x^{k+1} and $\bar{x}^{k+1}$ to be chosen such that

$$d_x^{k+1} \in \partial_x L(x^{k+1}, \bar{x}^k; y^k), \;\; d_{\bar{x}}^{k+1} \in \partial_{\bar{x}} L(x^{k+1}, \bar{x}^{k+1}; y^k), \tag{6}$$

where ∂_x and $\partial_{\bar{x}}$ represent the subgradients with respect to x and $\bar{x}$, respectively, and d_x and $d_{\bar{x}}$ are not exactly 0 but only converging to 0 asymptotically. For example, one can assign externally a shrinking and summable sequence of absolute errors [49]:

$$\|d_x^k\| \le \epsilon_x^k, \;\; \|d_{\bar{x}}^k\| \le \epsilon_{\bar{x}}^k, \;\; \sum_{k=1}^{\infty} \epsilon_x^k < \infty, \;\; \sum_{k=1}^{\infty} \epsilon_{\bar{x}}^k < \infty, \tag{7}$$

or a sequence of relative errors to the errors proportional to other variations in the algorithm [45], [50].

It was shown in [45] that a relative error criterion for terminating the iterations in subproblems, compared to other approximation criteria such as a summable absolute error sequence, better reduces the total number of subroutine iterations throughout the ADMM algorithm. Such a relative error criterion is a *constructive* one, rendered to guarantee the decrease of a quadratic distance between the intermediate solutions $(x^k, \bar{x}^k, y^k)$ and the optimum $(x^*, \bar{x}^*, y^*)$. In the context of distributed optimization problems under nonconvex constraints, since the convergence proof is established on a different basis from the quadratic distance, the construction of such a criterion must be reconsidered. We will address this issue in Subsection III-B.

C. Anderson acceleration

Linear convergence of the classical ADMM is essentially the result of subgradient dual update, which uses the information of only the first-order derivatives with respect to the dual variables: $\partial_y L = Ax + B\bar{x}$. The idea of creating a *quadratically convergent* algorithm using only first-order derivatives originates back from Nesterov's approach of solving convex optimization problems, which performs iterations based on a linear extrapolation of the previous two iterations instead of the current solution alone [51]. Such a *momentum* method can be used to accelerate the ADMM algorithm, which can be seen as iterations over the second block of primal variables $\bar{x}$ and the dual variables y [36]. However, such a momentum is inappropriate for nonconvex problems, since the behavior of the extrapolated point can not be well controlled by a bound on the curvature of the objective function.

Therefore, we resort to a different type of technique – Anderson acceleration, which was proposed in [52] first and later "rediscovered" in the field of chemical physics [53]. Generally speaking, Anderson acceleration is used to solve the fixed-point iteration problem

$$w = h_0(w) \tag{8}$$

for some vector w and non-expansive mapping h_0 (satisfying $\|h_0(w) - h_0(w')\| \le \|w - w'\|$ for any w and w'). Different from the simple Krasnoselskii-Mann iteration $w^{k+1} = \kappa w^k + (1-\kappa)h_0(w^k)$ ($\kappa \in (0,1)$), Anderson acceleration takes a quasi-Newton approach, which aims at a nearly quadratic convergence rate [54]. Specifically[1], in each iteration k, the results from the previous m iterations are recalled from memory to form the matrix of secants in w and $h(w) = w - h_0(w)$:

$$\begin{aligned} &\Delta_w^k = \left[\delta_w^{k-m} \quad \dots \quad \delta_w^{k-1}\right], \\ &\delta_w^{k'} = w^{k'+1} - w^{k'}, \;\; k' = k-m, \dots, k-1; \\ &\Delta_h^k = \left[\delta_h^{k-m} \quad \dots \quad \delta_h^{k-1}\right], \\ &\delta_h^{k'} = h(w^{k'+1}) - h(w^{k'}), \;\; k' = k-m, \dots, k-1. \end{aligned} \tag{9}$$

An estimated Jacobian is given by

$$H_k = I + (\Delta_h^k - \Delta_w^k)(\Delta_w^{k\top}\Delta_w^k)^{-1}\Delta_w^{k\top}, \tag{10}$$

or

$$H_k^{-1} = I + (\Delta_w^k - \Delta_h^k)(\Delta_w^{k\top}\Delta_h^k)^{-1}\Delta_w^{k\top}, \tag{11}$$

which minimizes the Frobenius norm of $B_k - I$ subject to $B_k\Delta_w^k = \Delta_h^k$. Then the quasi-Newton iteration $w^{k+1} = w^k - H_k^{-1}h^k$ leads to a weighted sum of the previous m function values:

$$w^{k+1} = \sum_{m'=0}^{m} \alpha_{m'}^k h_0(x^{k-m+m'}) \tag{12}$$

where the weights $\{\alpha_{m'}^k\}_{m'=0}^m$ are specified by

$$\alpha_{m'}^k = \begin{cases} s_0^k, & m' = 0 \\ s_{m'}^k - s_{m'-1}^k, & m' = 1, \dots, m-1 \\ 1 - s_{m-1}^k, & m' = m \end{cases}, \tag{13}$$

[1]There are two different types of Anderson acceleration. Here we focus on Type I, which was found to have better performance [54] and was improved in [40].

with $s^k_{m'}$ being the m'-th component s^k:

$$s^k = (\Delta_w^{k\top}\Delta_h^k)^{-1}\Delta_w^{k\top}h^k. \tag{14}$$

Anderson acceleration (12) may not always be convergent, although local convergence was studied in some special cases [55]. Recently, a globally convergent modification of Anderson acceleration was proposed in [40], where regularization, restarting, and safeguarding measures are taken to ensure the well-conditioning of the Δ_w^k matrix, boundedness of the inverse Jacobian estimate (11), and acceleration only in a safety region, respectively.

The relevance of Anderson acceleration to ADMM lies in that the ADMM algorithm (5) can be seen as fixed-point iterations $(\bar{x}^k, y^k) \to (\bar{x}^{k+1}, y^{k+1})$, $k = 0, 1, 2, \dots$ [42], which is the same idea underlying the ADMM with Nesterov acceleration. For problems with nonconvex constraints, the iteration mapping h is not necessarily non-expansive, and hence one can not directly establish the convergence of Anderson acceleration with the original techniques used in [40]. We will address this issue in Subsection III-C.

D. ADMM under nonconvex constraints

The presence of nonconvexity largely increases the difficulty of distributed optimization. Most of the work in nonconvex ADMM considers problems with nonconvex objective function with bounded Hessian eigenvalues or the Kurdyka-Łojasiewicz property assumptions, under which a convergence rate of $\mathcal{O}(1/\sqrt{k})$ (slower than that of convex ADMM, $\mathcal{O}(1/k)$) was established [26], [56], [57]. However, for many distributed optimization problems, e.g., the distributed MPC of nonlinear processes, there exist nonconvex constraints on the variables, which is intrinsically non-equivalent to the problems with nonconvex objective functions. For our problem of interest, the relevant works are scarce.

Here we introduce the algorithm of [30] for (1) under nonconvex constraints $x \in \mathcal{X}$ and $\bar{x} \in \bar{\mathcal{X}}$, reformulated with slack variables z:

$$\begin{aligned} &\min_{x,\bar{x},z}\ f(x) + g(\bar{x}) \\ &\text{s.t. } Ax + B\bar{x} + z = 0,\ \ z = 0,\ \ x \in \mathcal{X},\ \ \bar{x} \in \bar{\mathcal{X}}. \end{aligned} \tag{15}$$

The augmented Lagrangian is now written as

$$\begin{aligned} &L(x, \bar{x}, z; y, \lambda, \rho, \beta) = f(x) + g(\bar{x}) + \mathbb{I}_{\mathcal{X}}(x) + \mathbb{I}_{\bar{\mathcal{X}}}(\bar{x}) \\ &+ y^\top(Ax + B\bar{x} + z) + \frac{\rho}{2}\|Ax + B\bar{x} + z\|^2 + \lambda^\top z + \frac{\beta}{2}\|z\|^2. \end{aligned} \tag{16}$$

The algorithm is a two-layer one, where each outer iteration (indexed by k) contains a series of inner iterations (indexed by r). In the inner iterations, the classical ADMM algorithm is used to update x, $\bar{x}$, z and y in sequence, while keeping λ and β unchanged:

$$\begin{aligned} x^{k,r+1} &= \arg\min_x L(x, \bar{x}^{k,r}, z^{k,r}; y^{k,r}, \lambda^k, \rho^k, \beta^k) \\ &= \arg\min_{x\in\mathcal{X}} f(x) + \frac{\rho^k}{2}\left\|Ax + B\bar{x}^{k,r} + z^{k,r} + \frac{y^{k,r}}{\rho^k}\right\|^2 \\ \bar{x}^{k,r+1} &= \arg\min_{\bar{x}} L(x^{k,r+1}, \bar{x}, z^{k,r}; y^{k,r}, \lambda^k, \rho^k, \beta^k) \\ &= \arg\min_{\bar{x}\in\bar{\mathcal{X}}} g(\bar{x}) + \frac{\rho^k}{2}\left\|Ax^{k,r+1} + B\bar{x} + z^{k,r} + \frac{y^{k,r}}{\rho^k}\right\|^2 \\ z^{k,r+1} &= \arg\min_z L(x^{k,r+1}, \bar{x}^{k,r+1}, z; y^{k,r}, \lambda^k, \rho^k, \beta^k) \\ &= -\frac{\rho^k}{\rho^k + \beta^k}\left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + \frac{y^{k,r}}{\rho^k}\right) - \frac{1}{\rho^k + \beta^k}\lambda^k \\ y^{k,r+1} &= y^{k,r} + \rho^k(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}) \end{aligned} \tag{17}$$

Under mild assumptions, in the presence of slack variables z, it was proved [30] that if one chooses $\rho^k = 2\beta^k$, then the inner iterations converge to the set of stationary points $(x^k, \bar{x}^k, z^k, y^k)$ of the relaxed problem

$$\begin{aligned} &\min_{x,\bar{x},z}\ f(x) + g(\bar{x}) + \lambda^{k\top}z + \frac{\beta^k}{2}\|z\|^2 \\ &\text{s.t. } Ax + B\bar{x} + z = 0,\ \ x \in \mathcal{X},\ \ \bar{x} \in \bar{\mathcal{X}}. \end{aligned} \tag{18}$$

Then in the outer iterations, the dual variables λ^k are updated. To enforce the convergence of the slack variables to zero, the corresponding penalty β^k is amplified by a ratio $\gamma > 1$ if the returned z^k from the inner iterations does not decay enough from the previous outer iteration z^{k-1} ($\|z^k\| > \omega\|z^{k-1}\|$, $\omega \in (0, 1)$). The outer iteration is written as

$$\begin{aligned} \lambda^{k+1} &= \Pi_{[\underline{\lambda}, \overline{\lambda}]}(\lambda^k + \beta^k z^k) \\ \beta^{k+1} &= \begin{cases} \gamma\beta^k, & \|z^k\| > \omega\|z^{k-1}\| \\ \beta^k, & \|z^k\| \le \omega\|z^{k-1}\| \end{cases} \end{aligned} \tag{19}$$

in which the projection Π onto a predefined compact hypercube $[\underline{\lambda}, \overline{\lambda}]$ is used to guarantee the boundedness of the dual variables and hence the augmented Lagrangian L. If the augmented Lagrangian L remains bounded despite the increase of the penalty parameters ρ^k and β^k, the algorithm converges to a stationary point of the original problem (1). The iterative complexity of such an algorithm to reach an ϵ-approximate stationary point is $\mathcal{O}(\epsilon^{-4}\ln(\epsilon^{-1}))$.

In the next section, building on the algorithm of [30] that guarantees the convergence of distributed optimization under nonconvex constraints, we propose a new algorithm that integrates into it the ideas of approximate ADMM and Anderson acceleration, aiming at improving the computational efficiency.

III. Proposed Algorithm

A. Basic algorithm and its convergence

Consider an optimization problem in the following form:

$$\begin{aligned} &\min_{x,\bar{x}}\ f(x) + g(\bar{x}) \\ &\text{s.t. } Ax + B\bar{x} = 0 \\ &\qquad x \in \mathcal{X} = \{x | \phi(x) \le 0, \psi(x) = 0\},\ \ \bar{x} \in \bar{\mathcal{X}} \end{aligned} \tag{20}$$

or equivalently with slack variables

$$\begin{aligned} &\min_{x,\bar{x},z}\ f(x) + g(\bar{x}) \\ &\text{s.t. } Ax + B\bar{x} + z = 0,\ \ z = 0, \\ &\qquad x \in \mathcal{X} = \{x | \phi(x) \le 0, \psi(x) = 0\},\ \ \bar{x} \in \bar{\mathcal{X}}. \end{aligned} \tag{21}$$

We make the following assumptions.

Assumption 1. *Assume that f is lower bounded, i.e., there exists $\underline{f}$ such that $f(x) \ge \underline{f}$ for any $x \in \mathcal{X}$.*

Assumption 2. *Function g is convex and is lower bounded.*

Set: *Bound of dual variables* $[\underline{\lambda}, \overline{\lambda}]$*, shrinking ratio of slack variables* $\omega \in [0,1)$*, amplifying ratio of penalty parameter* $\gamma > 1$*, diminishing outer iteration tolerances* $\{\epsilon_1^k, \epsilon_2^k, \epsilon_3^k\}_{k=1}^{\infty} \downarrow 0$*, terminating tolerances* $\epsilon_1,\ \epsilon_2,\ \epsilon_3 > 0$;
Initialization: *Starting points* x^0*,* $\bar{x}^0$*,* z^0*, dual variable and bounds* $\lambda^1 \in [\underline{\lambda}, \overline{\lambda}]$*, penalty parameter* $\beta^1 > 0$;
outer iteration count $k \leftarrow 0$;
while *stationarity criterion* (26) *is not met* **do**
  $\rho^k = 2\beta^k$;
  inner iteration count $r \leftarrow 0$;
  Initialization: $x^{k,0}$*,* $\bar{x}^{k,0}$*,* $z^{k,0}$*,* $y^{k,0}$ *satisfying* $\lambda^k + \beta^k z^{k,0} + y^{k,0} = 0$;
  while *stopping criterion* (22) *is not met* **do**
    $x^{k,r+1} = \arg\min_{x\in\mathcal{X}} f(x) + \frac{\rho^k}{2}\left\|Ax + B\bar{x}^{k,r} + z^{k,r} + \frac{y^{k,r}}{\rho^k}\right\|^2$;
    $\bar{x}^{k,r+1} = \arg\min_{\bar{x}\in\bar{\mathcal{X}}} g(\bar{x}) + \frac{\rho^k}{2}\left\|Ax^{k,r+1} + B\bar{x} + z^{k,r} + \frac{y^{k,r}}{\rho^k}\right\|^2$;
    $z^{k,r+1} = -\frac{\rho^k}{\rho^k+\beta^k}\left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + \frac{y^{k,r}}{\rho^k}\right) - \frac{1}{\rho^k+\beta^k}\lambda^k$;
    $y^{k,r+1} = y^{k,r} + \rho^k(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1})$;
    $r \leftarrow r+1$;
  end
  $(x^{k+1}, \bar{x}^{k+1}, z^{k+1}, y^{k+1}) \leftarrow (x^{k,r}, \bar{x}^{k,r}, z^{k,r}, y^{k,r})$;
  $\lambda^{k+1} = \Pi_{[\underline{\lambda},\overline{\lambda}]}(\lambda^k + \beta^k z^k)$;
  if $\|z^{k+1}\| > \omega\|z^k\|$ **then**
    $\beta^{k+1} \leftarrow \gamma\beta^k$;
  else
    $\beta^{k+1} \leftarrow \beta^k$;
  end
  $k \leftarrow k+1$;
end

Algorithm 1: Basic algorithm (ELL).

Our basic algorithm (Algorithm 1) for (21) is slightly modified from the procedure of [30], which considered the case where $g(x) = 0$ and $\bar{\mathcal{X}}$ is a hypercube. The algorithm uses an inner loop of ADMM iterations and an outer loop of MM with possibly amplifying penalty parameters. The inner iterations are terminated when the following criterion is met

$$\begin{aligned}\epsilon_1^k &\geq \epsilon_1^{k,r} := \|\rho^k A^\top(B\bar{x}^{k,r+1} + z^{k,r+1} - B\bar{x}^{k,r} - z^{k,r})\|,\\ \epsilon_2^k &\geq \epsilon_2^{k,r} := \|\rho^k B^\top(z^{k,r+1} - z^{k,r})\|,\\ \epsilon_3^k &\geq \epsilon_3^{k,r} := \|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}\|.\end{aligned} \tag{22}$$

The proof uses the augmented Lagrangian (16) as a decreasing Lyapunov function throughout the inner iterations [34], [57], which gives the convergence of the inner iterations.

Lemma 1 (Descent of the augmented Lagrangian)**.** *Suppose that Assumptions 1 and 2 hold. When* $\rho^k = 2\beta^k$*, it holds that*

$$\begin{aligned}&L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) \leq L(x^{k,r}, \bar{x}^{k,r}, z^{k,r}, y^{k,r})\\ &- \beta^k\|B\bar{x}^{k,r+1} - B\bar{x}^{k,r}\|^2 - \frac{\beta^k}{2}\|z^{k,r+1} - z^{k,r}\|^2\end{aligned} \tag{23}$$

for $r = 0, 1, 2, \ldots$[2]*, and hence the augmented Lagrangian nonincreasingly converges to a limit* $\underline{L}_k$*.*

Corollary 1 (Convergence of inner iterations)**.** *Suppose that Assumptions 1 and 2 hold. As* $r \to \infty$*,* $B\bar{x}^{k,r+1} - B\bar{x}^{k,r} \to 0$*,* $z^{k,r+1} - z^{k,r} \to 0$*, and* $Ax^{k,r} + B\bar{x}^{k,r} + z^{k,r} \to 0$*. Hence the inner iterations are terminated at a finite* r *when* (22) *is met and the point* $(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1})$ *the following conditions*

$$\begin{aligned}d_1^k &\in \partial f(x^{k,r+1}) + \mathcal{N}_{\mathcal{X}}(x^{k,r+1}) + A^\top y^{k,r+1}\\ d_2^k &\in \partial g(\bar{x}^{k,r+1}) + \mathcal{N}_{\bar{\mathcal{X}}}(\bar{x}^{k,r+1}) + B^\top y^{k,r+1}\\ 0 &= \lambda^k + \beta^k z^{k,r+1} + y^{k,r+1}\\ d_3^k &= Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}\end{aligned} \tag{24}$$

for some d_1^k*,* d_2^k *and* d_3^k *satisfying* $\|d_1^k\| \leq \epsilon_1^k$*,* $\|d_2^k\| \leq \epsilon_2^k$ *and* $\|d_3^k\| \leq \epsilon_3^k$*, respectively.* $\mathcal{N}_{\mathcal{X}}(x)$ *(*$\mathcal{N}_{\bar{\mathcal{X}}}(\bar{x})$*) refers to the normal cone to the set* $\mathcal{X}$ *(*$\bar{\mathcal{X}}$*) at point* x *(*$\bar{x}$*):*

$$\mathcal{N}_{\mathcal{X}}(x) = \{v | v^\top(x' - x) \leq 0,\ \ \forall x' \in \mathcal{X}\}. \tag{25}$$

The proofs of the above lemma and corollary are given in Appendix A and Appendix B, respectively. It is apparent that if $\epsilon_1^k, \epsilon_2^k, \epsilon_3^k$ are all equal to 0, (24) is the Karush-Kuhn-Tucker optimality condition of the relaxed problem (18) [58].

We note that although the augmented Lagrangian decreases throughout the inner iterations, the increase in the penalty parameters may cause an increase in the augmented Lagrangian across outer iterations, thus losing the guarantee of overall convergence. To establish the convergence of outer iterations, we need to make the following assumption to restrict the upper level of the augmented Lagrangian.

Assumption 3. *The augmented Lagrangians are uniformly upper bounded at initialization of all inner iterations, i.e., there exists* $\overline{L} \geq L(x^{k,0}, \bar{x}^{k,0}, z^{k,0}, y^{k,0}, \lambda^k, \rho^k, \beta^k)$ *for all* k*.*

The above assumption is actually a "warm start" requirement. Suppose that we have a feasible solution $(x^0, \bar{x}^0)$ to the original problem (20), then we can always choose $x^{k,0} = x^0$, $\bar{x}^{k,0} = \bar{x}^0$, $z^{k,0} = 0$, $y^{k,0} = -\lambda^k$ to guarantee an $\overline{L} = f(x^0) + g(\bar{x}^0)$.

Lemma 2 (Convergence of outer iterations)**.** *Suppose that Assumptions 1, 2 and 3 hold. Then for any* ϵ_1*,* ϵ_2*, and* $\epsilon_3 > 0$*, within a finite number of outer iterations* k*, Algorithm 1 finds an approximate stationary point* $(x^{k+1}, \bar{x}^{k+1}, z^{k+1}, y^{k+1})$ *of* (20)*, satisfying*

$$\begin{aligned}d_1 &\in \partial f(x^{k+1}) + \mathcal{N}_{\mathcal{X}}(x^{k+1}) + A^\top y^{k+1}\\ d_2 &\in \partial g(\bar{x}^{k+1}) + \mathcal{N}_{\bar{\mathcal{X}}}(\bar{x}^{k+1}) + B^\top y^{k+1}\\ d_3 &= Ax^{k+1} + B\bar{x}^{k+1}\end{aligned} \tag{26}$$

for some d_1*,* d_2*,* d_3 *satisfying* $\|d_j\| \leq \epsilon_j$*,* $j = 1, 2, 3$*.*

See Appendix C for a proof. In addition to the convergence, we can also establish a theoretical complexity. Previously in [30], it was shown that to reach an ϵ-approximate stationary point satisfying (26) with $\epsilon_1, \epsilon_2, \epsilon_3 = \epsilon > 0$, the total number of inner iterations needed is of the order $\mathcal{O}(\epsilon^{-4}\ln(1/\epsilon))$. Here, we show that by appropriately choosing the way that the tolerances $(\epsilon_1^k, \epsilon_2^k, \epsilon_3^k)$ shrink, the iteration complexity can be provably reduced anywhere in $(\mathcal{O}(\epsilon^{-2}), \mathcal{O}(\epsilon^{-4})]$, for which a proof is given in Appendix D.

Lemma 3 (Complexity of the basic algorithm)**.** *Suppose that Assumptions 1, 2 and 3 hold. For some constant* $\vartheta \in (0, \omega]$*, choose* $\epsilon_1^k \sim \mathcal{O}(\vartheta^k)$*,* $\epsilon_2^k \sim \mathcal{O}(\vartheta^k)$*, and* $\epsilon_3^k \sim \mathcal{O}((\vartheta/\beta)^k)$*. Then each outer iteration* k *requires* $R^k \sim \mathcal{O}((\vartheta\omega)^{-2k})$

[2]For simplicity we did not write the last three entries λ^k, ρ^k, β^k that do not change during inner iterations in the augmented Lagrangian.

inner iterations. Hence, for the Algorithm 1 to reach an ϵ-approximate stationary point, the total iterations needed is $R \sim \mathcal{O}(\epsilon^{-2(1+\varsigma)})$, where $\varsigma = \log_\vartheta \omega \in (0, 1]$.

B. Approximate algorithm

We note that the basic algorithm requires complete minimization of x and $\bar{x}$ in each inner iteration (Lines 9–10, Algorithm 1). However, this is neither desirable due to the computational cost, nor practical since any nonlinear programming (NLP) solver finds only a point that approximately satisfies the KKT optimality conditions, except for very simple cases. For simplicity, we assume that such a minimization oracle[3], namely an explicit mapping G depending on matrix B, $Ax^{k,r+1} + z^{k,r} + y^{k,r}/\rho^k$, and ρ^k, exists for $\bar{x}$. For example, if $g(\bar{x}) = 0$ and $B^\top B = aI$ for some $a > 0$, then $G(B, v, \rho) = -\frac{1}{2a}B^\top v$. For the x-minimization, however, such an oracle usually does not exist. In this subsection, we will modify Algorithm 1 so as to allow approximate x-optimization on Line 9.

Assumption 4. *The minimization of the augmented Lagrangian with respect to $\bar{x}$ (Line 10, Algorithm 1) admits a unique explicit solution*

$$\bar{x}^{k,r+1} = G(B, Ax^{k,r+1} + z^{k,r} + y^{k,r}/\rho^k, \rho^k). \tag{27}$$

Let us also assume that the problem has a smoothness property as follows.

Assumption 5. *Functions f, ϕ and ψ are continuously differentiable, and $\mathcal{X}$ has a nonempty interior.*

Under this smoothness assumption, the KKT condition for x-minimization is written as the following equalities with $\mu \geq 0$ and ν representing the Lagrangian dual variables corresponding to the inequalities $\phi(x) \leq 0$ and $\psi(x) = 0$, respectively

$$\begin{aligned} 0 &= \nabla f(x^{k,r+1}) + \rho^k A^\top (Ax^{k,r+1} + B\bar{x}^{k,r} + z^{k,r} + y^{k,r}/\rho^k) \\ &\quad + \sum_{c=1}^{C_\phi} \mu_c \nabla \phi_c(x^{k,r+1}) + \sum_{c=1}^{C_\psi} \nu_c \nabla \psi_c(x^{k,r+1}) \\ 0 &= \mu_c \phi_c(x^{k,r+1}), \;\; c = 1, \ldots, C_\phi \\ 0 &= \psi_c(x^{k,r+1}), \;\; c = 1, \ldots, C_\psi. \end{aligned} \tag{28}$$

Line 9 of Algorithm 1 is thus to solve the above equations for $x^{k,r+1}$. This can be achieved through an interior point algorithm, which employs double-layer iterations to find the solution. In the outer iteration, a barrier technique is used to convert the inequality constraints into an additional term in the objective; the optima (or stationary points) of the resulting barrier problems converge to true optima (stationary points) as the barrier parameter converges to 0. In the inner iteration, a proper search method is used to obtain the optimum of the barrier problem. *Since both the interior point algorithm and the basic ADMM algorithm 1 have a double-layer structure, we consider matching these two layers.*

[3]We use the word "oracle" with its typical meaning in mathematics and computer science. An oracle refers to an ad hoc numerical or computational procedure, regarded as a black box mechanism, to generate the needed results as its outputs based on some input information.

Specifically, in the k-th outer iteration, the function $f(x)$ is appended with a barrier term $-b^k \sum_{c=1}^{C_\phi} \ln(-\phi_c(x))$ (b^k is the barrier parameter, converging to 0 as $k \to \infty$). Hence a "barrier augmented Lagrangian" can be specified as

$$L_b = L - b \sum_{c=1}^{C_\phi} \ln(-\phi_c(x)). \tag{29}$$

Based on the arguments in the previous subsection, if the x-optimization step returns a $x^{k,r+1}$ minimizing L_b with respect to x, then the inner iterations result in the descent of L_{b^k}, which implies the satisfaction of conditions (24), with f modified by the barrier function. Obviously, if Assumption 3 holds for L, then it also holds for L_{b^k} when $\mathcal{X}$ has a nonempty interior. It follows that the outer iterations can find an approximate stationary point of the original problem with the decay of barrier parameters b^k.

However, precisely finding the $x^{k,r+1}$ that minimizes L_b with respect to x, which is an equality-constrained NLP problem, still requires an iterative search procedure [59]. By matching the inner iterations of the interior point algorithm and the inner iterations of the ADMM, we propose to perform only *a proper amount of searching steps* instead of the entire equality-constrained NLP in each inner iteration, so that the solution to the equality-constrained NLP problem can be approached throughout the inner iterations. For this purpose, we assume that we have at hand a solver that can find any approximate solution of equality-constrained NLP.

Assumption 6. *Assume that for any equality-constrained smooth NLP problem*

$$\min_x \;\; \chi(x) \quad \text{s.t.} \;\; \psi(x) = 0 \tag{30}$$

a solver that guarantees the convergence to any approximate stationary point of the above problem with a lower objective function is available. That is, starting from any initial point x^0, for any tolerances $\epsilon_4, \epsilon_5 > 0$, within a finite number of searches the solver finds a point (x, ν) satisfying

$$\begin{aligned} d_4 &= \nabla \chi(x) + \sum_{c=1}^{C_\psi} \nu_c \nabla \psi_c(x) \\ d_{5c} &= \psi_c(x), \;\; c = 1, \ldots, C_\psi. \end{aligned} \tag{31}$$

for some $\|d_4\| \leq \epsilon_4$, $\|d_2\| \leq \epsilon_5$, and $f(x) \leq f(x^0)$. Such an approximate solution is denoted as $F(x^0; \chi, \psi, \epsilon_4, \epsilon_5)$.

The above approximate NLP solution oracle is realizable by NLP solvers where the tolerances of the KKT conditions are allowed to be specified by the user, e.g., the IPOPT solver [60]. Under Assumption 6, the x-update step on Line 9 of Algorithm 1 is replaced by an approximate NLP solution

$$x^{k,r+1} = F(x^{k,r}; \chi^{k,r}, \psi, \epsilon_4^{k,r}, \epsilon_5^{k,r}), \tag{32}$$

where the objective function in the current iteration is the part of barrier augmented Lagrangian L_{b^k} that is related to x with the indicator function $\mathbb{I}_{\mathcal{X}}(x)$ excluded:

$$\begin{aligned} \chi^{k,r}(x) = &f(x) - b_k \sum_{c=1}^{C_\phi} \ln(-\phi_c(x)) \\ &+ \frac{\rho^k}{2} \left\| Ax + B\bar{x}^{k,r} + z^{k,r} + y^{k,r}/\rho^k \right\|^2 \end{aligned} \tag{33}$$

Set: *Bound of dual variables* $[\underline{\lambda}, \overline{\lambda}]$*, shrinking ratio of slack variables* $\omega \in [0,1)$*, amplifying ratio of penalty parameter* $\gamma > 1$*, diminishing outer iteration tolerances* $\{\epsilon_1^k, \epsilon_2^k, \epsilon_3^k, \epsilon_4^k, \epsilon_5^k, \epsilon_6^k\}_{k=1}^{\infty} \downarrow 0$*, diminishing barrier parameters* $\{b^k\}_{k=1}^{\infty} \downarrow 0$*, terminating tolerances* ϵ_1*,* ϵ_2*,* ϵ_3*,* ϵ_4*,* ϵ_5*,* $\epsilon_6 > 0$*;*
Initialization: *Starting points* x^0*,* $\bar{x}^0$*,* z^0*, dual variable and bounds* $\lambda^1 \in [\underline{\lambda}, \overline{\lambda}]$*, penalty parameter* $\beta^1 > 0$*;*
outer iteration count $k \leftarrow 0$;
while $\epsilon_4^k \geq \epsilon_4$ *or* $\epsilon_5^k \geq \epsilon_5$ *or* $b^k \geq \epsilon_6$ *or stationarity criterion* (26) *is not met* **do**
  Set: *Diminishing tolerances* $\{\epsilon_4^{k,r}, \epsilon_5^{k,r}\}_{r=1}^{\infty} \downarrow 0$;
  let $\rho^k = 2\beta^k$;
  inner iteration count $r \leftarrow 0$;
  Initialization: $x^{k,0}$*,* $\bar{x}^{k,0}$*,* $z^{k,0}$*,* $y^{k,0}$ *satisfying* $\lambda^k + \beta^k z^{k,0} + y^{k,0} = 0$;
  while $\epsilon_4^{k,r} \geq \epsilon_4^k$ *or* $\epsilon_5^{k,r} \geq \epsilon_5^k$ *or stopping criterion* (22) *is not met* **do**
    $x^{k,r+1} = F(x^{k,r}; \chi^{k,r}, \psi, \epsilon_4^{k,r}, \epsilon_5^{k,r})$, where $\chi^{k,r}$ is given by (33);
    $\bar{x}^{r+1} = G(B, Ax^{r+1} + z^{k,r} + y^{k,r}/\rho^k, \rho^k)$, where G is given by (27);
    $z^{k,r+1} = -\frac{\rho^k}{\rho^k + \beta^k}\left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + \frac{y^{k,r}}{\rho^k}\right) - \frac{1}{\rho^k + \beta^k}\lambda^k$;
    $y^{k,r+1} = y^{k,r} + \rho^k(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1})$;
    $r \leftarrow r + 1$;
  end
  $(x^{k+1}, \bar{x}^{k+1}, z^{k+1}, y^{k+1}) \leftarrow (x^{k,r}, \bar{x}^{k,r}, z^{k,r}, y^{k,r})$;
  $\lambda^{k+1} = \Pi_{[\underline{\lambda}, \overline{\lambda}]}(\lambda^k + \beta^k z^k)$;
  if $\|z^{k+1}\| > \omega\|z^k\|$ **then**
    $\beta^{k+1} \leftarrow \gamma\beta^k$;
  else
    $\beta^{k+1} \leftarrow \beta^k$;
  end
  $k \leftarrow k + 1$;
end

Algorithm 2: Approximate algorithm (ELLA).

This approximate algorithm with inexact x-minimization is summarized as Algorithm 2. The inner iterations are performed until $\epsilon_4^{k,r}$ and $\epsilon_5^{k,r}$ are lower than ϵ_4^k and ϵ_5^k, respectively, and (22) holds. The outer iterations are terminated when $\epsilon_4^k \leq \epsilon_4$, $\epsilon_5^k \leq \epsilon_5$, the barrier parameter is sufficiently small $b^k \leq \epsilon_6$, and (26) holds.

Lemma 4 (Convergence of the approximate algorithm)**.** *Suppose that Assumptions 1–6 hold. For any outer iteration k, given any positive tolerances $\{\epsilon_1^k, \ldots, \epsilon_5^k\}$, within a finite number of inner iterations r, the obtained solution satisfies*

$$\begin{aligned} d_1^k + d_4^k &= \nabla f(x^{k,r+1}) + \sum_{c=1}^{C_\phi} \mu_c^{k,r+1} \nabla \phi_c(x^{k,r+1}) \\ &\quad + \sum_{c=1}^{C_\psi} \nu_c^{k,r+1} \nabla \psi_c(x^{k,r+1}) + A^\top y^{k,r+1} \\ d_2^k &\in \partial g(\bar{x}^{k,r+1}) + \mathcal{N}_{\bar{\mathcal{X}}}(\bar{x}^{k,r+1}) + B^\top y^{k,r+1} \\ 0 &= \lambda^k + \beta^k z^{k,r+1} + y^{k,r+1} \\ d_3^k &= Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}, \\ d_5^k &= \psi(x^{k,r+1}) \\ -b^k &= \mu_c^{k,r+1} \phi_c(x^{k,r+1}), \ \ c = 1, \ldots, C_\phi. \end{aligned} \tag{34}$$

for some $d_1^k, \ldots, d_5^k$ with $\|d_1^k\| \leq \epsilon_1^k, \ \ldots, \ \|d_5^k\| \leq \epsilon_5^k$. Then, suppose that the outer iteration tolerances $\{\epsilon_1^k, \ldots, \epsilon_5^k\}$ and barrier parameters b^k are diminishing with increasing k, given any terminating tolerances $\epsilon_1, \ldots, \epsilon_6 > 0$, within a finite number of outer iterations, Algorithm 2 finds a point $(x^{k+1}, \bar{x}^{k+1}, z^{k+1}, y^{k+1}, \mu^{k+1}, \nu^{k+1})$ satisfying

$$\begin{aligned} d_1 + d_4 &= \nabla f(x^{k+1}) + \sum_{c=1}^{C_\phi} \mu_c^{k+1} \nabla \phi_c(x^{k+1}) \\ &\quad + \sum_{c=1}^{C_\psi} \nu_c^{k+1} \nabla \psi_c(x^{k+1}) + A^\top y^{k+1} \\ d_2 &\in \partial g(\bar{x}^{k+1}) + \mathcal{N}_{\bar{\mathcal{X}}}(\bar{x}^{k+1}) + B^\top y^{k+1} \\ 0 &= \lambda^k + \beta^k z^{k+1} + y^{k+1} \\ d_3 &= Ax^{k+1} + B\bar{x}^{k+1}, \\ d_5 &= \psi(x^{k+1}) \\ -d_6 &= \mu_c^{k+1} \phi_c(x^{k+1}), \ \ c = 1, \ldots, C_\phi. \end{aligned} \tag{35}$$

for some $d_1, \ldots, d_6$ with $\|d_j\| \leq \epsilon_j$, $j = 1, \ldots, 5$, $d_6 \in (0, \epsilon_6]$.

The proof is self-evident following the techniques in the Proofs of Lemma 1, Corollary 1 and Lemma 2 given in Appendix A to Appendix C. The conditions (34) indicate an $(\epsilon_1^k, \ldots, \epsilon_5^k)$-approximate stationary point to the relaxed barrier problem

$$\begin{aligned} \min_{x, \bar{x}, z} \ & f(x) + g(\bar{x}) - b^k \sum_{c=1}^{C_\phi} \ln(-\phi_c(x)) \\ \text{s.t.} \ & Ax + B\bar{x} + z = 0, \ \ \psi(x) = 0, \ \ \bar{x} \in \bar{\mathcal{X}} \end{aligned} \tag{36}$$

and the condition (35) gives an $(\epsilon_1, \ldots, \epsilon_6)$-approximate stationary point to the original problem (20).

C. Accelerated algorithm

The key factor restricting the rate of convergence is the y-update, which is not a full or approximate maximization but only one step of subgradient ascent. As was proved in Lemma 3, such a subgradient ascent approach for nonconvex problems leads to a number of inner iterations proportional to the inverse squared error. Here, by modifying the Anderson acceleration scheme in [40], we propose an accelerated algorithm. Let us make the following assumption regarding our choice of tolerances $\epsilon_4^{k,r}$ and $\epsilon_5^{k,r}$.

Assumption 7. *Suppose that we choose a continuous and strictly monotonically increasing function $\pi : [0, \infty) \to [0, \infty)$ with $\pi(0) = 0$ such that $\epsilon_5^{k,r} = \pi(\epsilon_4^{k,r})$, and choose $\epsilon_4^{k,r+1}$ proportional to $\|\rho^k A^\top (B\bar{x}^{k,r+1} - B\bar{x}^{k,r} + z^{k,r+1} - z^{k,r})\|$ when such a value is strictly smaller than the previous tolerance $\epsilon_4^{k,r}$ but not smaller the ultimate one ϵ_4^k.*

The choice of function π to relate the stationarity tolerance and equality tolerance in NLP subroutine is aimed at balancing the effort to reduce both errors. The choice of $\epsilon_4^{k,r+1}$ is based on the following rationale. After the r-th inner iteration, the obtained solution $x^{k,r+1}$ satisfies the approximate stationarity condition

$$\begin{aligned} d_4^{k,r+1} &= \nabla f(x^{k,r+1}) + \sum_{c=1}^{C_\phi} \mu_c^{k,r+1} \nabla \phi_c(x^{k,r+1}) + \sum_{c=1}^{C_\psi} \nu_c^{k,r+1} \cdot \\ &\nabla \psi_c(x^{k,r+1}) + \rho^k A^\top \left(Ax^{k,r+1} + B\bar{x}^{k,r} + z^{k,r} + y^{k,r}/\rho^k\right) \end{aligned} \tag{37}$$

for some $d_4^{k,r+1}$ with a modulus not exceeding $\epsilon_4^{k,r}$, $\mu^{k,r+1}$ satisfying $\mu_c^{k,r+1}\phi_c(x^{k,r+1}) = -b^k$, $c = 1, \ldots, C_\phi$. Using the formula for y-update (Line 13, Algorithm 2), we rearrange the above equation to obtain

$$d_4^{k,r+1} + \rho^k A^\top (B\bar{x}^{k,r+1} - B\bar{x}^{k,r} + z^{k,r+1} - z^{k,r}) = \nabla f(x^{k,r+1}) + \sum_{c=1}^{C_\phi} \mu_c^{k,r+1} \nabla \phi_c(x^{k,r+1}) + \sum_{c=1}^{C_\psi} \nu_c^{k,r+1} \nabla \psi_c(x^{k,r+1}) + A^\top y^{k,r+1} \tag{38}$$

Hence after the update of $\bar{x}$, z and y variables, the violation of the stationarity condition is bounded by $\epsilon_4^{k,r} + \|\rho^k A^\top (B\bar{x}^{k,r+1} - B\bar{x}^{k,r} + z^{k,r+1} - z^{k,r})\|$. Therefore, $\epsilon_4^{k,r}$ should be balanced with the second term, which, however, is realizable only after the $\bar{x}$- and z-updates after the x-update and hence assigned to $\epsilon_4^{k,r+1}$.

We note from Algorithm 2 that under Assumption 7, each inner iteration r is a mapping from $(x^{k,r}, \bar{x}^{k,r}, z^{k,r+1}, y^{k,r+1}, \epsilon_4^{k,r})$ to $(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}, \epsilon_4^{k,r+1})$. In fact, despite the dependence of the latter variables on $x^{k,r}$ and $\epsilon_4^{k,r}$, such dependence can be ignored in the sense that the descent of the barrier augmented Lagrangian L_{b^k} will always guide the sequence of intermediate solutions towards the set of ϵ_4^k-approximate stationary points of the relaxed barrier problem (36). Using Lemma 1 with the augmented Lagrangian substituted by the barrier augmented Lagrangian, it immediately follows that under the approximate algorithm, the sequence $\{(\bar{x}^{k,r}, z^{k,r})\}_{r=1}^\infty$ will converge to a fixed point, and the convergence of $\{y^{k,r}\}$ accompanies the convergence of $\{z^{k,r}\}$ due to (77). It is thus clear that we may resort to Anderson acceleration introduced in Subsection II-C by denoting $w = (\bar{x}, z)$, the iteration as a mapping h_0, and $h(w) = w - h_0(w)$, and collecting at the r-th inner iteration the following multi-secant information about the previous m inner iterations:

$$\Delta_w^{k,r} = [\delta_w^{k,r-m} \ \ldots \ \delta_w^{k,r-1}], \ \Delta_h^{k,r} = [\delta_h^{k,r-m} \ \ldots \ \delta_h^{k,r-1}], \tag{39}$$

where $\delta_h^{r-m'} = w^{k,r-m'+1} - w^{k,r-m'}$ and $\delta_h^{r-m'} = h(w^{k,r-m'+1}) - h(w^{k,r-m'})$, $m' = m-1, \ldots, 0$.

However, the possibility that Δ_w^k may not be of full rank and H_k may be singular requires certain modifications to the original accelration scheme. The following technique was used in [40]. First, it can be shown that the matrix H defined in (10) can be constructed in an inductive way, starting from $H_{k,r}^0 = I$, by rank-one updates

$$H_{k,r}^{m'+1} = H_{k,r}^{m'} + \frac{(\delta_h^{k,r-m+m'} - H_{k,r}^{m'} \delta_w^{k,r-m+m'})(\hat{\delta}_w^{k,r-m+m'})^\top}{(\hat{\delta}_w^{k,r-m+m'})^\top \delta_w^{k,r-m+m'}} \tag{40}$$

for $m' = 0, \ldots, m-1$ with $H_{k,r}^m = H_{k,r}$, where $\hat{\delta}_w^{k,r-m}, \ldots, \hat{\delta}_w^{k,r-1}$ are obtained from $\delta_w^{k,r-m}, \ldots, \delta_w^{k,r-1}$ through Gram-Schmidt orthogonalization. To ensure the invertibility of $H_{k,r}$, the δ_h vector in (40) is perturbed to

$$\tilde{\delta}_h^{k,r-m+m'} = (1 - \theta_{k,r}^{m'}) \delta_h^{k,r-m+m'} + \theta_{k,r}^{m'} \delta_w^{k,r-m+m'}, \tag{41}$$

where the perturbation magnitude $\theta_{k,r}^{m'}$ is determined by

$$\theta_{k,r}^{m'} = \varphi\left(\frac{(\hat{\delta}_w^{k,r-m+m'})^\top (H_{k,r}^{m'})^{-1} \delta_h^{k,r-m+m'}}{\|\hat{\delta}_w^{k,r-m+m'}\|^2}; \eta_\theta \right). \tag{42}$$

with regularization hyperparameter $\eta_\theta \in (0, 1)$. The function $\varphi(\theta; \eta)$ is defined by

$$\varphi(\theta; \eta) = \begin{cases} (\eta \text{sign}\theta - \theta)/(1-\theta) & , |\theta| \le \eta \\ 0 & , |\theta| > \eta \end{cases}. \tag{43}$$

Using the Sherman-Morrison formula for inverting the rank-one update, $H_{k,r}^{-1}$ can be induced from $(H_{k,r}^0)^{-1} = I$ according to

$$(H_{k,r}^{m'+1})^{-1} = (H_{k,r}^{m'})^{-1} + \frac{\left(\delta_w^{k,r-m+m'} - (H_{k,r}^{m'})^{-1} \tilde{\delta}_h^{k,r-m+m'}\right) (\hat{\delta}_w^{k,r-m+m'})^\top (H_{k,r}^{m'})^{-1}}{(\hat{\delta}_w^{k,r-m+m'})^\top (H_{k,r}^{m'})^{-1} \tilde{\delta}_h^{k,r-m+m'}}. \tag{44}$$

To avoid the rank deficiency Δ_w, a restart checking strategy is used, where the memory is cleared when the Gram-Schmidt orthogonalization becomes ill conditioned ($\|\hat{\delta}_w^{k,r}\| < \eta_w \|\delta_w^{k,r}\|$ for some $\eta_w \in (0, 1)$) or the memory exceeds a maximum M; otherwise the memory is allowed to grow. Hence the Anderson acceleration is well-conditioned.

Lemma 5 (Well-conditioning of Anderson acceleration, [40])**.** *Using the regularization and restart checking techniques, it is guaranteed that*

$$\|H_{k,r}^{-1}\|_2 \le \theta^{-M} \left[3(1 + \theta + \eta_w)^M \eta_w^{-N} - 2 \right]^{N-1} < +\infty \tag{45}$$

where M is the maximum number of steps in the memory and N is the dimension of w.

A well-conditioned Anderson acceleration is not yet sufficient to guarantee the convergence. Hence we employ a safeguarding technique modified from [40] which aims at suppressing a too large increase in the barrier augmented Lagrangian by rejecting such acceleration steps. When Anderson acceleration suggests an update from $w = (\bar{x}, z)$ to $\tilde{w} = (\tilde{\bar{x}}, \tilde{z})$ under the current value of Ax, the resulting Lagrangian increase is calculated as

$$\begin{aligned} \tilde{L}^k(w, \tilde{w}; Ax) =& g(\tilde{\bar{x}}) - g(\bar{x}) + \lambda^{k\top}(\tilde{z} - z) + \frac{\beta^k}{2}(\|\tilde{z}\|^2 - \|z\|^2) \\ &+ \tilde{y}^\top (Ax + B\tilde{\bar{x}} + \tilde{z}) - y^\top (Ax + B\bar{x} + z) \\ &+ \frac{\rho^k}{2} \left(\|Ax + B\tilde{\bar{x}} + \tilde{z}\|^2 - \|Ax + B\bar{x} + \tilde{z}\|^2 \right) \end{aligned} \tag{46}$$

where y and $\tilde{y}$ are calculated by

$$y = -\lambda^k - \beta^k z, \ \ \tilde{y} = -\lambda^k - \beta^k \tilde{z}, \tag{47}$$

which results from Line 12–13 of Algorithm 2. We require that such a change, if positive, must not exceed an upper bound:

$$\tilde{L}^k(w, \tilde{w}; Ax) = \tilde{L}_0 \eta_L (R_+ + 1)^{-(1+\sigma)} \tag{48}$$

where $\tilde{L}_0$ is the expected Lagrangian decrease after the first non-accelerated iteration after initialization according to Lemma 1, used as a scale for the change in the barrier augmented Lagrangian:

$$\tilde{L}_0 = \beta^k \|B\bar{x}^{k,1} - B\bar{x}^{k,0}\|^2 + \frac{\beta^k}{2} \|z^{k,1} - z^{k,0}\|^2, \tag{49}$$

where $\eta_L, \sigma > 0$ are hyperparameters, and R_+ is the number of already accepted acceleration steps. With safeguarding,

it can be guaranteed that the barrier augmented Lagrangian always stays bounded, since $\sum_{R_+=0}^{\infty}(R_+ + 1)^{-(1+\sigma)} < +\infty$. We also require that the acceleration should not lead to a drastic change in w:

$$\|\tilde{w} - w\|^2 \leq \frac{\tilde{L}_0}{\beta^k}\frac{\eta_{\tilde{w}}}{\sqrt{1+R_+}}, \tag{50}$$

where $\eta_{\tilde{w}} > 0$ is a hyperparameter. $1/\sqrt{1+R_+}$ reflects an expected change according to the plain ADMM iteration, which is used to suppress disproportionate large deviations due to Anderson acceleration.

Finally, the accelerated algorithm using the Anderson acceleration technique for fixed-point iteration of $(\bar{x}, z)$ is summarized as Algorithm 3. This is our final ELLADA algorithm, whose distributed implementation will be briefly discussed in the next subsection. With well-conditioned $H_{k,r}^{-1}$ matrix and a bounded barrier augmented Lagrangian, its convergence can now be guaranteed by the following lemma, the proof of which is given in Appendix E.

Lemma 6 (Convergence under Anderson acceleration)**.** *Suppose that Assumptions 1–7 hold. Under regulated and safeguarded Anderson acceleration, Algorithm 3 finds within a finite number of inner iterations r a point satisfying* (34)*. The convergence of outer iterations to an approximate stationary point satisfying* (35) *hence follows.*

Summarizing the conclusions of all the previous lemmas in this section, we have arrived at the following theorem.

Theorem 1. *Suppose that the following assumptions hold:*

1) *Function f is lower bounded on $\mathcal{X}$;*
2) *Function g is convex and lower bounded on $\bar{\mathcal{X}}$;*
3) *Initialization of outer iterations allows a uniform upper bound of the augmented Lagrangian, e.g., a feasible solution is known a priori;*
4) *Minimization of $g(\bar{x}) + \frac{\rho}{2}\|B\bar{x} + v\|^2$ with respect to $\bar{x}$ allows an oracle $G(B, v, \rho)$ returning a unique solution for any v of appropriate dimension and $\rho > 0$;*
5) *Functions f, ϕ, and ψ are continuously differentiable, and the constraints (ϕ, ψ) are strictly feasible;*
6) *There exists a solver for equality-constrained NLP to any specified tolerances of KKT conditions.*

Then given any tolerances $\epsilon_1, \ldots, \epsilon_6 > 0$, the ELLADA algorithm (Algorithm 3) gives an $(\epsilon_1, \ldots, \epsilon_6)$-approximate KKT point satisfying the conditions (35)*.*

If the problem itself has intrinsically better properties to guarantee that each KKT point is a local minimum, e.g., the second-order sufficient condition [46, §4.3.2], then the algorithm converges to a local minimum. Of course, it is well known that certifying a local minimum is itself a difficult problem.

IV. Implementation on Distributed Nonlinear MPC

Consider a nonlinear discrete-time dynamical system

$$x(t+1) = f(x(t), u(t)) \tag{51}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ are the vectors of states and inputs, respectively, for $t = 0, 1, 2, \ldots$, and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Suppose that at time t we have the current states $x = x(t)$, then in MPC, the control inputs are determined by the following optimal control problem:

$$\begin{aligned} \min\ & J = \sum_{\tau=t}^{t+T-1} \ell(\hat{x}(\tau), \hat{u}(\tau)) + \ell^{\mathrm{f}}(\hat{x}(t+T)) \\ \text{s.t. }& \hat{x}(\tau+1) = f(\hat{x}(\tau), \hat{u}(\tau)),\ \ \tau = t, \ldots, t+T-1 \\ & p(\hat{x}(\tau), \hat{u}(\tau), \tau) \leq 0,\ \ \tau = t, \ldots, t+T-1 \\ & q(\hat{x}(\tau), \hat{u}(\tau), \tau) = 0,\ \ \tau = t, \ldots, t+T-1 \\ & \hat{x}(t) = x. \end{aligned} \tag{52}$$

In the above formulation, the optimization variables $\hat{x}(\tau)$ and $\hat{u}(\tau)$ represent the predicted states and inputs in a future horizon $\{t, t+1, \ldots, t+T\}$ with length $T \in \mathbb{N}$. The predicted trajectory is constrained by the dynamics (51) as well as some additional path constraints p, q such as the bounds on the inputs and states or Lyapunov descent to enforce stability. Functions ℓ and ℓ^{f} are called the stage cost and terminal cost, respectively. By solving (52), one executes $u(t) = \hat{u}(t)$. For simplicity it is assumed here that the states are observable; otherwise, the states can be estimated using an optimization formulation such as moving horizon estimation (MHE). For continuous-time systems, collocation techniques can be used to discretize the resulting optimal control problem into a finite-dimensional one.

Now suppose that the system (51) is large-scale with its states and outputs decomposed into n subsystems: $x = [x_1^\top, x_2^\top, \ldots, x_n^\top]^\top$, $u = [u_1^\top, u_2^\top, \ldots, u_n^\top]^\top$, and that the optimal control problem should be solved by the corresponding n agents, each containing the model of its own subsystem:

$$x_i(\tau+1) = f_i(x_i(\tau), u_i(\tau), \{x_{ji}(\tau), u_{ji}(\tau)\}_{j \in \mathcal{P}(i)}). \tag{53}$$

where $\{x_{ji}, u_{ji}\}$ stands for the states and inputs in subsystem j (i.e., components of x_j and u_j) that appear in the arguments of f_i, which comprise of the components of f corresponding to the i-th subsystem. $\mathcal{P}_i$ is the collection of subsystems j that has some inputs and outputs influencing subsystem i. We assume that the cost functions and the path constraints are separable:

$$\begin{aligned} &\ell(\hat{x}, \hat{u}) = \sum_{i=1}^{n} \ell_i(\hat{x}_i, \hat{u}_i),\ \ \ell^{\mathrm{f}}(\hat{x}) = \sum_{i=1}^{n} \ell_i^{\mathrm{f}}(\hat{x}_i), \\ &p(\hat{x}, \hat{u}, \tau) = [p_1(\hat{x}_1, \hat{u}_1, \tau)^\top, \ldots, p_n(\hat{x}_n, \hat{u}_n, \tau)^\top]^\top, \\ &q(\hat{x}, \hat{u}, \tau) = [q_1(\hat{x}_1, \hat{u}_1, \tau)^\top, \ldots, q_n(\hat{x}_n, \hat{u}_n, \tau)^\top]^\top. \end{aligned} \tag{54}$$

A. Formulation on directed and bipartite graphs

To better visualize the problem structure and systematically reformulate the optimal control problem (52) into the distributed optimization problem in the form of (21) for the implementation of the ELLADA algorithm, we introduce some graph-theoretic descriptions of optimization problems [7]. For problem (52), we first define a directed graph (digraph), which is a straightforward characterization of the relation of mutual impact among the subsystem models.

Definition 1 (Digraph)**.** *The digraph of system* (51) *under the decomposition $x = [x_1^\top, x_2^\top, \ldots, x_n^\top]^\top$ and $u = [u_1^\top, u_2^\top, \ldots, u_n^\top]^\top$ is $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ with nodes $\mathcal{V}_1 =$*

Set: *Dual bounds* $[\underline{\lambda}, \overline{\lambda}]$, *outer iteration parameters* $\omega \in [0,1)$, $\gamma > 1$, $\{\epsilon_1^k, \epsilon_2^k, \epsilon_3^k, \epsilon_4^k, \epsilon_6^k\}_{k=1}^{\infty} \downarrow 0$, $\{b^k\}_{k=1}^{\infty} \downarrow 0$, *final tolerances* ϵ_1, ϵ_2, ϵ_3, ϵ_4, $\epsilon_6 > 0$, *function* π, *acceleration parameters* $\theta \in (0,1)$, $\sigma > 0$, $\eta_\epsilon > 0$, $\eta_w \in (0,1)$, $\eta_L > 0$, $\eta_{\tilde{w}} > 0$, $M \in \mathbb{N}$. Let $\epsilon_5 = \pi(\epsilon_4)$;
Initialization: *Starting points* x^0, $\bar{x}^0$, z^0, $\lambda^1 \in [\underline{\lambda}, \overline{\lambda}]$, *penalty parameter* $\beta^1 > 0$, $\epsilon_5^0 = \pi(\epsilon_4^0)$;
Outer iteration count $k \leftarrow 0$;
while $\epsilon_4^k \geq \epsilon_4$ *or* $\epsilon_5^k \geq \epsilon_5$ *or* $b^k \geq \epsilon_6$ *or stationarity criterion* (26) *is not met* **do**
  Set: *Initial tolerances* $\epsilon_4^{k,0}$, $\epsilon_5^{k,0} = \pi(\epsilon_4^{k,0})$, *penalty* $\rho^k = 2\beta^k$, *Jacobian estimate* $H_{k,0}^{-1} = I$;
  Inner iteration count $r \leftarrow 0$, count of accelerated steps $R_+^k = 0$, memory length $m \leftarrow 0$;
  Initialization: $x^{k,0}$, $\bar{x}^{k,0}$, $z^{k,0}$, $y^{k,0}$ *satisfying* $\lambda^k + \beta^k z^{k,0} + y^{k,0} = 0$;
  while $\epsilon_4^{k,r} \geq \epsilon_4^k$ *or* $\epsilon_5^{k,r} \geq \epsilon_5^k$ *or stopping criterion* (22) *is not met* **do**
    $x^{k,r+1} = F(x^{k,r}; \chi^{k,r}, \psi, \epsilon_4^{k,r}, \epsilon_5^{k,r})$, where $\chi^{k,r}$ is given by (33);
    $\bar{x}^{k,r+1} = G(B, Ax^{r+1} + z^{k,r} + y^{k,r}/\rho^k, \rho^k)$, where G is given by (27);
    $z^{k,r+1} = -\frac{\rho^k}{\rho^k+\beta^k}\left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + \frac{y^{k,r}}{\rho^k}\right) - \frac{1}{\rho^k+\beta^k}\lambda^k$;
    $y^{k,r+1} = y^{k,r} + \rho^k(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1})$;
    $\tilde{y}^{k,r} = -\lambda^k - \beta^k \tilde{z}^{k,r}$;
    $\tilde{x}^{k,r+1} = F(x^{k,r}; \tilde{\chi}^{k,r}, \psi, \epsilon_4^{k,r}, \epsilon_5^{k,r})$, with $\tilde{\chi}$ in (33) with $\bar{x}, z, y$ replaced by $\tilde{\bar{x}}, \tilde{z}, \tilde{y}$;
    $\tilde{\bar{x}}^{k,r+1} = G(B, A\tilde{x}^{r+1} + \tilde{z}^{k,r} + \tilde{y}^{k,r}/\rho^k, \rho^k)$;
    $\tilde{z}^{k,r+1} = -\frac{\rho^k}{\rho^k+\beta^k}\left(A\tilde{x}^{k,r+1} + B\tilde{\bar{x}}^{k,r+1} + \frac{\tilde{y}^{k,r}}{\rho^k}\right) - \frac{1}{\rho^k+\beta^k}\lambda^k$;
    if $r = 0$ **then**
      $\tilde{w}^{k,1} \leftarrow (\bar{x}^{k,1}, z^{k,1})$, and calculate $\tilde{L}_0$ by (49);
    else
      $\delta_w^{k,r-1} = \tilde{w}^{k,r} - w^{k,r-1}$, $\delta_h^{k,r} = \tilde{w}^{k,r} - \tilde{w}^{k,r+1} - w^{k,r-1} + w^{k,r}$, $m \leftarrow m+1$;
      $\hat{\delta}_w^{k,r-1} = \delta_w^{k,r-1} - \sum_{m'=2}^{m} \frac{(\hat{\delta}_w^{k,r-m'})^\top \delta_w^{k,r-1}}{\|\hat{\delta}_w^{k,r-m'}\|^2} \hat{\delta}_w^{k,r-m'}$;
      if $m = M+1$ *or* $\frac{\|\hat{\delta}_w^{k,r-1}\|}{\|\delta_w^{k,r-1}\|} < \eta_w$ **then** $m \leftarrow 0$, $\hat{\delta}_w^{k,r-1} \leftarrow \delta_w^{k,r-1}$, and $H_{k,r-1}^{-1} \leftarrow I$;
      Compute $\tilde{\delta}_h^{k,r-1}$ by (41) with $m' = m-1$ and $\theta_{k,r-1} = \varphi\left(\frac{(\hat{\delta}_w^{k,r-1})^\top H_{k,r-1}^{-1} \delta_h^{k,r-1}}{\|\hat{\delta}_w^{k,r-1}\|^2}; \theta\right)$;
      Update $H_{k,r}^{-1} = H_{k,r-1}^{-1} + \frac{(\delta_w^{k,r-1} - H_{k,r-1}^{-1}\tilde{\delta}_h^{k,r-1})(\hat{\delta}_w^{k,r-1})^\top H_{k,r-1}^{-1}}{(\hat{\delta}_w^{k,r-1})^\top H_{k,r-1}^{-1} \hat{\delta}_w^{k,r-1}}$, and suggest $\tilde{w}^{k,r+1} = w^{k,r} - H_{k,r}^{-1}(w^{k,r} - w^{k,r+1})$;
      if $\frac{\tilde{L}^k(w^{k,r}, \tilde{w}^{k,r+1}; Ax^{k,r})}{\tilde{L}_0 \eta_L (R_+ + 1)^{-(1+\sigma)}} \leq 1$ *and* $\|\tilde{w}^{k,r+1} - w^{k,r}\|^2 \leq \frac{\tilde{L}_0 \eta_{\tilde{w}}}{\beta^k \sqrt{R_+ + 1}}$ **then** accept the acceleration $w^{k,r+1} \leftarrow \tilde{w}^{k,r+1}$, and let $y^{k,r+1} \leftarrow -\lambda^k - \beta^k \tilde{z}^{k,r+1}$;
    end
    $\epsilon_4^{k,r+1} = \|\rho^k A^\top (B\bar{x}^{k,r+1} - B\bar{x}^{k,r} + z^{k,r+1} - z^{k,r})\|$, $\epsilon_5^{k,r+1} = \pi(\epsilon_4^{k,r+1})$;
    $r \leftarrow r+1$;
  end
  $(x^{k+1}, \bar{x}^{k+1}, z^{k+1}, y^{k+1}) \leftarrow (x^{k,r}, \bar{x}^{k,r}, z^{k,r}, y^{k,r})$;
  Update $\lambda^{k+1} = \Pi_{[\underline{\lambda}, \overline{\lambda}]}(\lambda^k + \beta^k z^k)$ and β^{k+1} according to (19);
  $k \leftarrow k+1$;
end

Algorithm 3: Accelerated algorithm (ELLADA).

$\{1, 2, \ldots, n\}$ *and edges* $\mathcal{E}_1 = \{(j,i) | j \in \mathcal{P}(i)\}$. *If* $(i,j) \in \mathcal{E}_1$, *i.e.,* $j \in \mathcal{P}(i)$, *we say that* j *is a parent of* i *and* i *is a child of* j *(denoted as* $i \in \mathcal{C}(j)$*).*

Then under the decomposition, (52) can be written as

$$\begin{aligned} \min \ & \sum_{i\in\mathcal{V}_1} J_i = \sum_{i\in\mathcal{V}} \sum_{\tau=t}^{t+T-1} \ell_i(\hat{x}_i(\tau), \hat{u}_i(\tau)) + \ell_i^{\mathrm{f}}(\hat{x}_i(t+T)) \\ \text{s.t. } & \hat{x}_i(\tau+1) = f_i(\hat{x}_i(\tau), \hat{u}_i(\tau), \{\hat{x}_{ji}(\tau), \hat{u}_{ji}(\tau)\}_{j\in\mathcal{P}(i)}), \\ & p_i(\hat{x}_i(\tau), \hat{u}_i(\tau), \tau) \leq 0, \ \ \tau = t, \ldots, t+T-1, \ \ i \in \mathcal{V}_1 \\ & q_i(\hat{x}_i(\tau), \hat{u}_i(\tau), \tau) = 0, \ \ \tau = t, \ldots, t+T-1, \ \ i \in \mathcal{V}_1 \\ & \hat{x}_i(t) = x_i, \ \ i \in \mathcal{V}_1, \end{aligned} \tag{55}$$

We denote the variables of the i-th agent as

$$\begin{aligned} \xi_i = [& \hat{x}_i(t)^\top, \hat{u}_i(t)^\top, \ldots, \hat{x}_i(t+T-1)^\top, \hat{u}_i(t+T-1)^\top, \\ & \hat{x}_i(t+T)^\top, \{\hat{x}_{ji}(t)^\top, \hat{u}_{ji}(t)^\top\}_{j\in\mathcal{P}(i)}, \\ & \ldots, \{\hat{x}_{ji}(t+T-1)^\top, \hat{u}_{ji}(t+T-1)^\top\}_{j\in\mathcal{P}(i)}]^\top, \end{aligned} \tag{56}$$

in which the variables related to the j-th subsystem are denoted as ξ_{ji}. Since ξ_{ji} is a part of the predicted states and inputs from subsystem j, i.e., some components of ξ_j, the interactions between the parent j and the child i be captured by a matrix $\overrightarrow{D}_{ji}$ with exactly one unit entry ("1") on every row: $\xi_{ji} = \overrightarrow{D}_{ji}\xi_j$, where the right arrow represents the impact of the parent subsystem j on the child subsystem i. By denoting the model and path constraints in agent i as $\xi_i \in \Xi_i$, the optimal control problem (51) is expressed in a compact way as follows:

$$\begin{aligned} \min \ & \sum_{i\in\mathcal{V}_1} J_i(\xi_i) \\ \text{s.t. } & \xi_i \in \Xi_i, \ \ i \in \mathcal{V}_1, \ \ \xi_{ji} = \overrightarrow{D}_{ji}\xi_j, \ \ (j,i) \in \mathcal{E}_1. \end{aligned} \tag{57}$$

This is an optimization problem defined on a *directed graph*. An illustration for a simple case when $\mathcal{E}_1 = \{(1,2), (2,3), (3,1)\}$ is shown in Fig. 2(a).

Although it is natural to represent the interactions among the subsystems in a digraph, performing distributed optimization on digraphs where the agents communicate among themselves without a coordinator can be challenging. For example, it is known that the ADMM algorithm, which behaves well for distributed optimization with 2 blocks of variables, can become divergent when directly extended to multi-block prob-

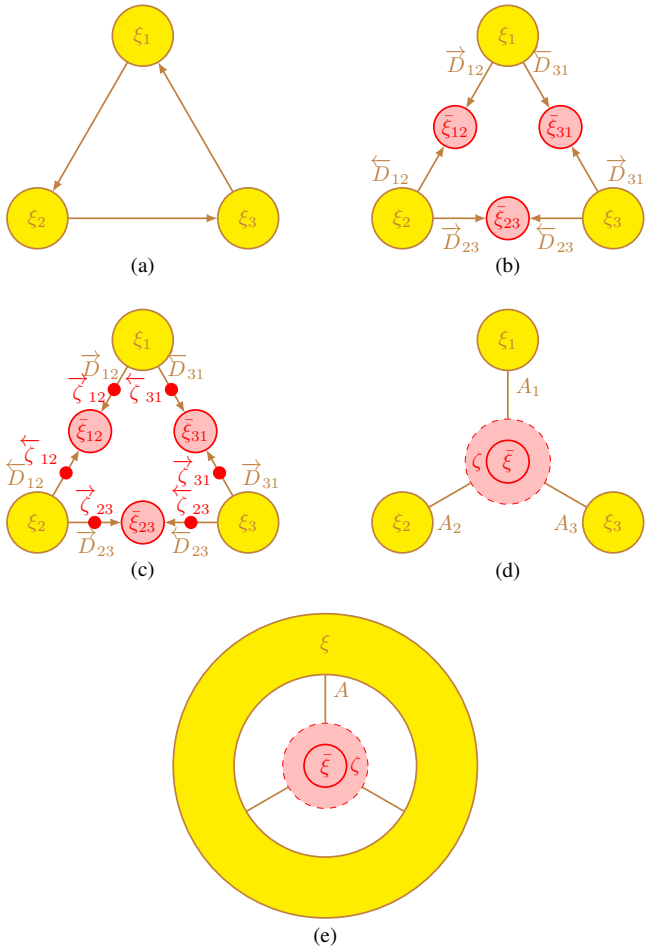

Fig. 2. Graphical illustrations of the problem structure of distributed MPC.

lems [61]. Hence we construct such a 2-block architecture by using a *bipartite graph*.

Definition 2 (Bipartite graph)**.** *The bipartite graph of system* (51) $\mathcal{G}_2$ *is constructed from the digraph* $\mathcal{G}_1$ *by taking both the nodes and edges as the new nodes, and adding an edge between* $i \in \mathcal{V}_1$ *and* $e \in \mathcal{E}_1$ *if* i *is the head or tail of* e *in the digraph, i.e.,* $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ *with* $\mathcal{V}_2 = \mathcal{V}_1 \cup \mathcal{E}_1$, $\mathcal{E}_2 = \{(i,e) | i \in \mathcal{V}_1, e \in \mathcal{E}_1, e = (i,j), j \in \mathcal{C}(i) \text{ or } e = (j,i), j \in \mathcal{P}(i)\}$.

Such a graph is bipartite since any edge is between a node of $\mathcal{V}_1$ and a node of $\mathcal{E}_1$.

We note that the last line of (57) corresponds to the digraph edges $\mathcal{E}_1$. In the bipartite graph, these edges should become nodes and hence new groups of variables should be associated with them. For this purpose, we simply need to pull out ξ_{ji} as overlapping variables $\bar{\xi}_{ji}$, and add the constraint that $\bar{\xi}_{ji}$ are some selected components of ξ_i: $\xi_{ji} = \overleftarrow{D}_{ji}\xi_i$:

$$\begin{aligned} \min \ & \sum_{i \in \mathcal{V}_1} J_i(\xi_i) \\ \text{s.t. } & \xi_i \in \Xi_i, \ \ i \in \mathcal{V}_1, \ \ \bar{\xi}_{ji} = \overrightarrow{D}_{ji}\xi_j = \overleftarrow{D}_{ji}\xi_j, \ \ (j,i) \in \mathcal{E}_1 \end{aligned} \tag{58}$$

In (58), variables ξ_i ($i \in \mathcal{V}_1$) and $\bar{\xi}_{ji}$ ($(j,i) \in \mathcal{E}_1$) are defined on the nodes of the bipartite graph, and the constraints captured by the matrices $\overrightarrow{D}_{ji}$ and $\overleftarrow{D}_{ji}$ correspond to the bipartite edges $(j,(j,i))$ and $(i,(j,i))$, respectively. We may also write the last line of (58) as

$$\bar{\xi}_e = D_{ie}\xi_i, \ \ (i,e) \in \mathcal{E}_2. \tag{59}$$

Therefore (58) is an optimization problem on the bipartite graph. An illustration is given in Fig. 2(b). Under this reformulation, the problem structure becomes a 2-block one – distributed agents $i = 1, \dots, N$ manage the decision variables ξ_i, $\mathcal{V}_1$ in parallel without interference, and the coordinator regulates the agents by using overlapping variables $\bar{\xi}_e$, $e \in \mathcal{E}_1$.

B. Reformulation with slack variables

It is known that a key condition for distributed optimization in the context of the ADMM algorithm to converge is that one block of variables can always be made feasible given the other block [26]. Unfortunately this condition is not always met by the problem (58). For example, given ξ_1 and ξ_2, there may not be a $\bar{\xi}_{12}$ satisfying both $\bar{\xi}_{12} = \overrightarrow{D}_{12}\xi_1$ and $\bar{\xi}_{12} = \overleftarrow{D}_{12}\xi_2$. To deal with this issue, it was proposed to associate with each linear constraint in (58), namely each edge in the bipartite graph, a slack variable ζ_{ie} (e.g., [30]):

$$\begin{aligned} \min \ & \sum_{i \in \mathcal{V}_1} J_i(\xi_i) \\ \text{s.t. } & \xi_i \in \Xi_i, \ \ i \in \mathcal{V}_1 \\ & D_{ie}\xi_i - \bar{\xi}_e + \zeta_{ie} = 0, \ \ (i,e) \in \mathcal{E}_2 \\ & \zeta_{ie} = 0, \ \ (i,e) \in \mathcal{E}_2. \end{aligned} \tag{60}$$

Similar to the notation for D, we write ζ_{ie} as $\overrightarrow{\zeta}_{ij}$ if $e = (i,j)$ and $\overleftarrow{\zeta}_{ij}$ if $e = (j,i)$. Such a problem structure is graphically illustrated in Fig. 2(c).

Finally, we stack all the subscripted variables into ξ, $\bar{\xi}$, ζ in a proper ordering of $i \in \mathcal{V}_1$, $e \in \mathcal{E}_1$, and $(i,e) \in \mathcal{E}_2$. The matrices D_{ie} are stacked in a block diagonal pattern in the same ordering of $(i,e) \in \mathcal{E}_2$ into A. The appearance of $\bar{\xi}_e$ in the equality constraints is represented by a matrix B (satisfying $B^\top B = 2I$). We write the objective function as $J(\xi)$, and the set constraints Ξ_i are lumped into a Cartesian product $\Xi = \times_{i \in \mathcal{V}_1} \Xi_i$. Finally, we reach a compact formulation for (60):

$$\begin{aligned} \min \ & J(\xi) \\ \text{s.t. } & \xi \in \Xi, \ \ A\xi + B\bar{\xi} + \zeta = 0, \ \ \zeta = 0 \end{aligned} \tag{61}$$

Such an architecture is shown in Figs. 2(d) and 2(e). The variables $\bar{\xi}$ and ζ belong to the coordinator (marked in red), and ξ is in the distributed agents.

C. Implementation of ELLADA

Clearly, the optimal control problem formulated as (60) is a special form of (21) with ξ, $\bar{\xi}$ and ζ rewritten as x, $\bar{x}$ and z, respectively, and $g(\bar{x}) = 0$, $\bar{\mathcal{X}}$ equal to the entire Euclidean space. As long as the cost function J is lower bounded (e.g., a quadratic cost), Algorithm 3 is applicable to (60), where the operations on $\bar{x}$, z, y are performed by the coordinator, and the operations on x is handled by the distributed agents. Specifically,

- The update steps of $\bar{x}, z, y$ (Lines 10–13, 15, 16) and the entire Anderson acceleration (Lines 17–26) belong

to the coordinator. The updates of penalty parameters and outer-layer dual variables λ (Lines 31) should also be performed by the coordinator. The conditions for $\epsilon_1^k, \epsilon_2^k, \epsilon_3^k$ and $\epsilon_1, \epsilon_2, \epsilon_3$ are checked by the coordinator.

- The distributed agents are responsible for carrying out a trial x-update step for the Anderson acceleration (Line 9) as well as the plain x-update (Line 14). The conditions and updates for $\epsilon_4^{k,r}, \epsilon_5^{k,r}, \epsilon_4^k, \epsilon_5^k$, and $\epsilon_4, \epsilon_5, \epsilon_6$ are checked by the agents.

When executing the updates, the agents need the values of $B\bar{x}+z+y/\rho$ to add to Ax, and the coordinator needs the value of Ax from the agents. When the variables x are distributed into agents $x_1, \ldots, x_n$, and the equality constraints between the agents and the coordinator is expressed on a bipartite graph:

$$D_{ie}x_i - \bar{x}_e + z_{ie} = 0, \ \ (i,e) \in \mathcal{E}_2, \tag{62}$$

the communication of Ax and $B\bar{x} + z + y/\rho$ takes place in a distributed and parallel way, i.e., the i-th agent obtains the information of $-\bar{x}_e + z_{ie} + y_{ie}/\rho$ for all e such that $(i,e) \in \mathcal{E}_2$ from the coordinator. The coordinator, based on inter-subsystem edges e in the digraph, obtains the information of $D_{ie}x_i$ for all related agents i. When the objective function and $\mathcal{X}$ are separable $f(x) = \sum_{i=1}^n f_i(x_i)$, $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$, based on such distributed and parallel communication, the optimization problem

$$\begin{aligned} \min \ & f(x) - b\sum_{c=1}^{C_\phi} \ln(-\phi_c(x)) + \frac{\rho}{2}\left\| Ax + B\bar{x} + z + \frac{y}{\rho} \right\|^2 \\ \text{s.t.} \ & \psi(x) = 0 \end{aligned} \tag{63}$$

in an x-update step can be solved in a distributed and parallel manner:

$$\left.\begin{aligned} \min_{x_i} \ & f_i(x_i) - b\sum_{c=1}^{C_{\phi,i}} \ln(-\phi_{c,i}(x_i)) \\ & + \frac{\rho}{2} \sum_{\{e|(i,e)\in\mathcal{E}_2\}} \left\| D_{ie}x_i - \bar{x}_e + z_{ie} + \frac{y_{ie}}{\rho} \right\|^2 \\ \text{s.t.} \ & \psi_i(x_i) = 0 \end{aligned}\right\} \ /\!/ \text{ for } i. \tag{64}$$

Similarly, the $\bar{x}$-update with the G-mapping is in parallel for its components e, if $\bar{\mathcal{X}}$ is separable, i.e., if $\bar{\mathcal{X}}$ is a closed hypercube (whether bounded or unbounded), and if g is also separable. That is, $\bar{x}$-update can be expressed as

$$\left.\begin{aligned} \min_{\bar{x}_i} \ & g_i(\bar{x}_i) + \frac{\rho}{2} \sum_{\{i|(i,e)\in\mathcal{E}_2\}} \left\| D_{ie}x_i - \bar{x}_e + z_{ie} + \frac{y_{ie}}{\rho} \right\|^2 \\ \text{s.t.} \ & \bar{x}_i \in \bar{\mathcal{X}}_i \end{aligned}\right\} \ /\!/ \text{ for } e. \tag{65}$$

The z and y updates are in parallel for the edges (i,e) on the bipartite graph.

In Algorithm 3, the procedures are written such that in each iteration, the update steps are carried out in sequence. This requires a synchronization of all the agents i and the coordinating elements e and (i,e). For example, for the x-update, every distributed agent needs to create a "finish" signal after solving x_i in (64) and send it to the coordinator. Only after the coordinator receives the "finish" signals from all the distributed agents can the $\bar{x}$-update be carried out. Due to the possible computational imbalance among the agents and the coordinator, such synchronization implies that faster updates must idle for some time to wait for slower ones. In fact, the convergence properties of the ELLADA algorithm do not rely on the synchronization. Even when the inner iterations are asynchronous, the update steps still contribute to the convergence of the barrier augmented Lagrangian and hence result in convergence to KKT conditions. The only exception is that under Anderson acceleration, the steps for generating the candidate of accelerated updates are allocated to another coordinator and another set of distributed agents, and they should communicate to make the decision on executing the accelerations.

TABLE I
PARAMETERS AND NOMINAL STEADY STATE

Parameter	Value	Parameter	Value
A_1, A_3	28	a_1, a_3	3.145
A_2, A_4	32	a_2, a_4	2.525
γ_1	0.43	k_1	3.14
γ_2	0.34	k_2	3.29
Input	Value	Input	Value
v_1	3.15	v_2	3.15
State	Value	State	Value
h_1	12.44	h_2	13.17
h_3	4.73	h_4	4.99

V. APPLICATION TO A QUADRUPLE TANK PROCESS

The quadruple tank process is a simple benchmark process for distributed model predictive control [62] with 4 states (water heights in the 4 tanks) and 2 inputs (flow rates from the reservoir). The dynamic model is written as follows:

$$\begin{aligned} \dot{h}_1 &= -\frac{a_1}{A_1}\sqrt{h_1} + \frac{a_3}{A_1}\sqrt{h_3} + \frac{\gamma_1 k_1}{A_1}v_1 \\ \dot{h}_2 &= -\frac{a_2}{A_2}\sqrt{h_2} + \frac{a_4}{A_2}\sqrt{h_4} + \frac{\gamma_2 k_2}{A_2}v_2 \\ \dot{h}_3 &= -\frac{a_3}{A_3}\sqrt{h_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2 \\ \dot{h}_4 &= -\frac{a_4}{A_4}\sqrt{h_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1. \end{aligned} \tag{66}$$

Other parameter values and the nominal steady state are given in Table I. The process is considered to have 2 subsystems, one containing tanks 1 and 4 and the other containing tanks 2 and 3. Each subsystem has 2 states, 1 input and 1 upstream state. We first design a centralized MPC with quadratic objective function for each tank, and bounds on the inputs $2.5 \le v_1, v_2 \le 3.5$. We first decide through the simulation of centralized MPC that a receding horizon of $T = 400$ with sampling time $\delta t = 10$ is appropriate. (The computations are performed using the Python module `pyomo.dae` with an IPOPT solver [63].)

The closed-loop trajectories under the traditional MPC controllers, including a centralized MPC (black), a semi-centralized MPC where the inputs are iteratively updated based on predictions over the entire process (green), a decentralized MPC (blue), and a distributed MPC with only state feedforwarding among the agents (purple), are shown in Fig. 3. It was observed that a semi-centralized MPC based on system-wide prediction maintains the control performance, yielding

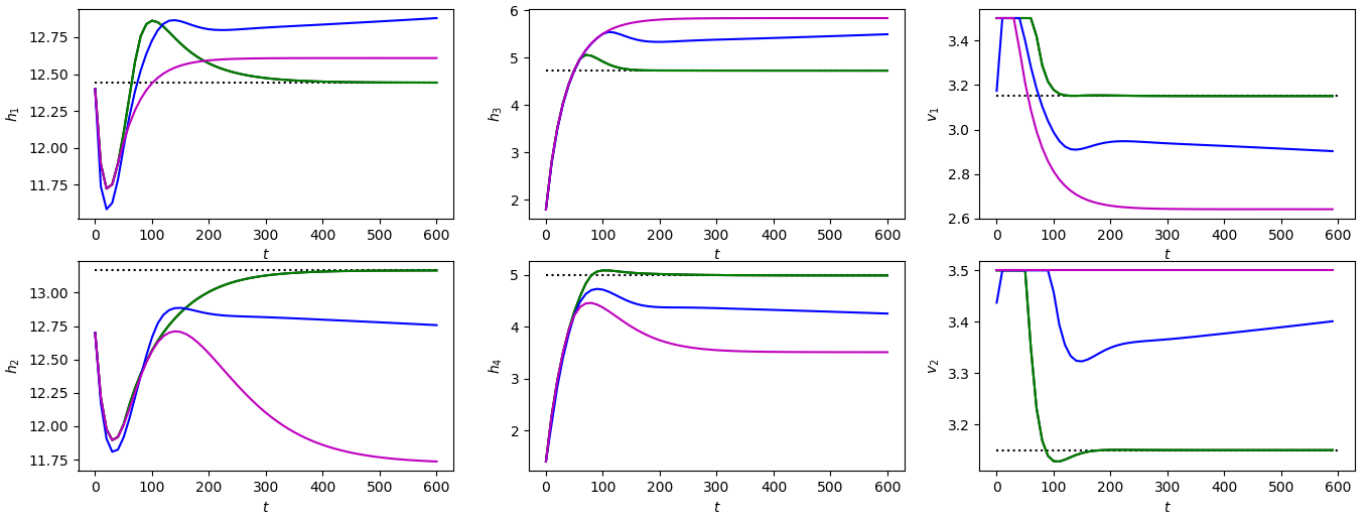

Fig. 3. Closed-loop trajectories under traditional MPC controllers.

trajectories overlapping with those of the centralized MPC. However, the state-feedforward distributed MPC without sufficient coordination accounting for the state interactions results in unsatisfactory control performance, whose ultimate deviation from the steady state is even larger than the decentralized MPC without any communication between the controllers.

Next we use the proposed ELLADA algorithm for distributed nonlinear MPC of the process. We first examine the basic ELL algorithm (Algorithm 1) by solving the corresponding distributed MPC problem at a state with $h_1 = 12.6$, $h_2 = 12.4$, $h_3 = 5.0$, $h_4 = 4.5$, where we set $\omega = 0.75$, $\gamma = 2$, $\epsilon_1^k = \epsilon_2^k = 10^{-2}/2^{k-1}$, $\epsilon_3^k = 10^{-1}/2^{k-1}$, $\epsilon_1 = \epsilon_2 = 10^{-4}$, $\epsilon_3 = 10^{-3}$ and $\overline{\lambda} = -\underline{\lambda} = 10$ (in an element-wise sense) through empirical tuning. The solution results in terms of the variation of the augmented Lagrangian $L^{k,r}$, the violations to the KKT conditions $\epsilon_{1,2,3}^{k,r}$, and penalty parameters ρ^k throughout the inner and outer iterations are presented in Fig. 4, where the rainbow colormap from blue to red colors stand for increasing outer iteration number. In accordance to the conclusion of Lemma 1, the augmented Lagrangian is monotonically decreasing in each outer iterations and remains upper bounded, which guarantees the convergence of the algorithm. Using the ELL algorithm for the afore-mentioned closed-loop MPC simulation, the resulting trajectories are found identical to those of the centralized control, which corroborates the theoretical property of the algorithm of converging to the set of stationary solutions.

With the preserved control performance of the ELL algorithm, we seek to improve its computational efficiency with the ELLA and ELLADA algorithms (Algorithms 2 and 3). In ELLA, the tolerances for approximate NLP solution are set as $\epsilon_1 = \epsilon_2 = \epsilon_4 = 10^3\epsilon_3 = 1$, $\epsilon_1^k = \epsilon_2^k = 10^3\epsilon_3^k = \epsilon_4^k = 100/2^{k-1}$, $\epsilon_4^{k,r} = 10^3\epsilon_5^{k,r} = \max(\epsilon_4^k, 40(\epsilon_1^{k,r})^2)$. The barrier constants are updated throughout outer iterations according to $\|z\|$ according to $b^{k+1} = \min(10^{-1}, \max(10^{-4}, 25(\epsilon_3^k)^2))$. Compared to ELL, the accumulated number of iterations and computational time of ELLA are reduced by over an order of magnitude. To seek for better computational performance, we apply the ELLADA algorithm, where we set $M = 10$, $\sigma = 1$, $\eta_L = \eta_{\tilde{w}} = 0.01$, $\eta_\theta = 0.5$, $\eta_w = 0.05$. This further reduces the number of iterations and computational time. These results are shown in Fig. 5.

Compared to the basic ELL algorithm, ELLADA achieves acceleration by approximately 18 times in terms of iterations and 19 times in computational time for the entire simulation time span. These improvements are more significant when the states are far from the target steady state (43 and 45 times, respectively, for the first 1/6 of the simulation). We note that the improvement from ELLA to ELLADA by using the Anderson scheme is not an order-of-magnitude one mainly because each outer iteration needs only a few number of inner iterations, leaving little space for further acceleration (e.g., for the first sampling time, 12 outer iterations including only 102 inner iterations are needed in ELLA, and in ELLADA, 61 inner iterations are needed). Under the accelerations, ELLADA returns the identical solution to the centralized optimization, thus preserving the control performance of the centralized MPC.

VI. Conclusions and Discussions

We have proposed a new algorithm for distributed optimization allowing nonconvex constraints, which simultaneously guarantees convergence under mild assumptions and achieves fast computation. Specifically, convergence is established by adopting a two-layer architecture. In the outer layer, the slack variables are tightened using the method of multipliers, and the inequalities are handled using a barrier technique. In the inner layer, ADMM iterations are performed in a distributed and coordinated manner. Approximate NLP solution and Anderson acceleration techniques are integrated into inner iterations for computational acceleration.

Such an algorithm is generically suitable for distributed nonlinear MPC. The advantages include:

- Arbitrary input and state couplings among subsystems are allowed. No specific pattern is required a priori.

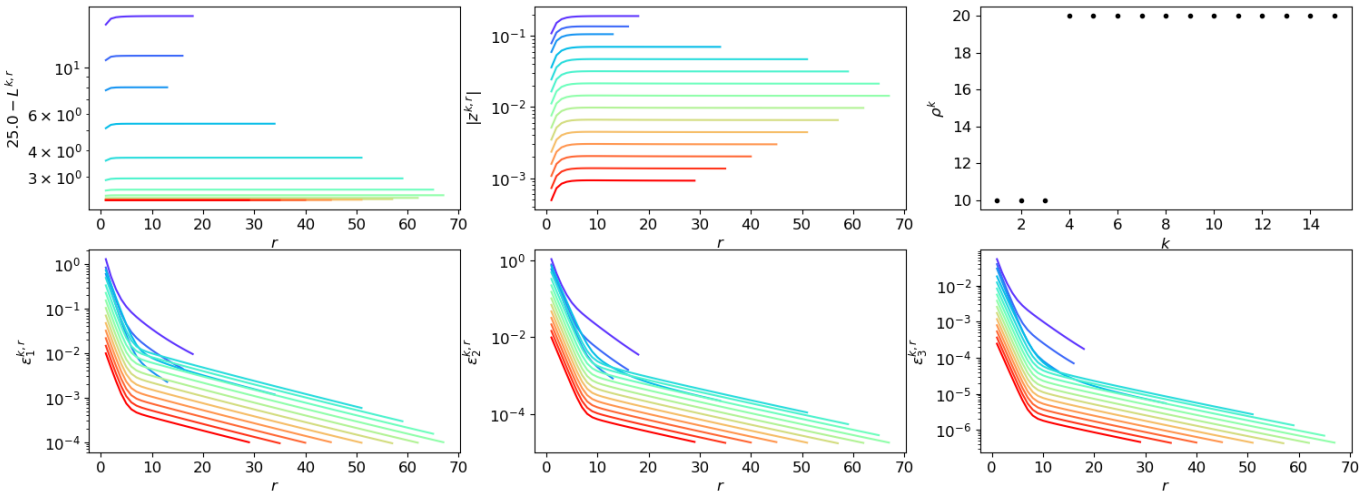

Fig. 4. Solution results of the ELL algorithm.

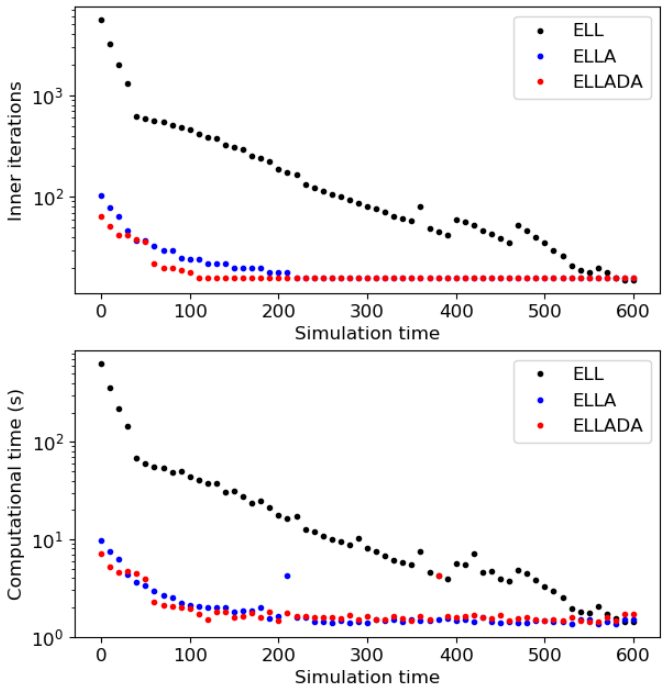

Fig. 5. Iteration and computational time under ELL, ELLA and ELLADA algorithms.

- The convergence property of the algorithm towards a stationary point is theoretically guaranteed, and its performance can be monitored throughout iterations.
- Equality-constrained NLP solvers can be used only as a subroutine. No internal modification of solvers is needed, and the choice of any appropriate solver is flexible.
- Asynchronous updates are allowed without affecting the convergence properties.
- Although motivated with a nominal optimal control problem, the algorithm could be suitable for more intricate MPC formulations such as stochastic/robust MPC or sensitivity-based advance-step MPC.

The application of the ELLADA algorithm on the distributed nonlinear MPC of a quadruple tank process has already shown its improved computational performance compared to the basic convergent Algorithms 1 and 2, and improved control performance compared to the decentralized MPC and distributed MPC without accounting for state interactions. Of course, due to the small size of the specific benchmark process, the control can be realized easily with a centralized MPC. A truly large-scale control problem is more suitable to demonstrate the effectiveness of our algorithm, and this shall be presented in an upcoming separate paper.

Appendix A
Proof of Lemma 1

First, since $x^{k,r+1}$ is chosen as the minimizer of the augmented Lagrangian with respect to x (Line 9, Algorithm 1), the update of x leads to a decrease in L:

$$L(x^{k,r+1}, \bar{x}^{k,r}, z^{k,r}, y^{k,r}) \leq L(x^{k,r}, \bar{x}^{k,r}, z^{k,r}, y^{k,r}). \tag{67}$$

Second, we consider the decrease resulted from $\bar{x}$-update:

$$\begin{aligned}
&L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r}, y^{k,r}) - L(x^{k,r+1}, \bar{x}^{k,r}, z^{k,r}, y^{k,r}) \\
&= g(\bar{x}^{k,r+1}) - g(\bar{x}^{k,r}) + y^{k,r\top}(B\bar{x}^{k,r+1} - B\bar{x}^{k,r}) \\
&\quad + \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r}\|^2 \\
&\quad - \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r} + z^{k,r}\|^2 \\
&= g(\bar{x}^{k,r+1}) - g(\bar{x}^{k,r}) - \frac{\rho^k}{2}\|B\bar{x}^{k,r+1} - B\bar{x}^{k,r}\|^2 \\
&\quad - \rho^k(\bar{x}^{k,r} - \bar{x}^{k,r+1})^\top B^\top \left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r} + \frac{y^{k,r}}{\rho^k}\right).
\end{aligned} \tag{68}$$

The minimization of $\bar{x}$ (Line 10, Algorithm 1) should satisfy the optimality condition

$$\begin{aligned}
0 \in &\rho^k B^\top \left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r} + \frac{y^{k,r}}{\rho^k}\right) \\
&+ \partial g(\bar{x}^{k,r+1}) + \mathcal{N}_{\bar{\mathcal{X}}}(\bar{x}^{k,r+1}),
\end{aligned} \tag{69}$$

i.e., there exist vectors $v_1 \in \partial g(\bar{x}^{k,r+1})$ and $v_2 \in \mathcal{N}_{\bar{\mathcal{X}}}(\bar{x}^{k,r+1})$ with

$$\rho^k B^\top \left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r} + \frac{y^{k,r}}{\rho^k} \right) = -v_1 - v_2. \quad (70)$$

Since $v_1 \in \partial g(\bar{x}^{k,r+1})$ and g is convex, $v_1^\top(\bar{x}^{k,r} - \bar{x}^{k,r+1}) \leq g(\bar{x}^{k,r}) - g(\bar{x}^{k,r+1})$. And $v_2 \in \mathcal{N}_{\bar{\mathcal{X}}}(\bar{x}^{k,r+1})$ implies $v_2^\top(\bar{x}^{k,r} - \bar{x}^{k,r+1}) \leq 0$. Hence

$$\begin{aligned} &\rho^k(\bar{x}^{k,r} - \bar{x}^{k,r+1})^\top B^\top \left(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r} + \frac{y^{k,r}}{\rho^k} \right) \\ &= -v_1^\top(\bar{x}^{k,r} - \bar{x}^{k,r+1}) - v_2^\top(\bar{x}^{k,r} - \bar{x}^{k,r+1}) \\ &\geq -(g(\bar{x}^{k,r}) - g(\bar{x}^{k,r+1})). \end{aligned} \quad (71)$$

Substituting the above inequality in (68), we obtain

$$\begin{aligned} L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r}, y^{k,r}) \leq L(x^{k,r+1}, \bar{x}^{k,r}, z^{k,r}, y^{k,r}) \\ -\frac{\rho^k}{2}\|B\bar{x}^{k,r+1} - B\bar{x}^{k,r}\|^2. \end{aligned} \quad (72)$$

Third, we consider the decrease resulted from z- and y-updates:

$$\begin{aligned} &L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) - L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r}, y^{k,r}) \\ &= \lambda^{k\top}(z^{k,r+1} - z^{k,r}) + \frac{\beta^k}{2}(\|z^{k,r+1}\|^2 - \|z^{k,r}\|^2) \\ &\quad + y^{k,r+1\top}(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}) \\ &\quad - y^{k,r\top}(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r}) \\ &\quad + \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}\|^2 \\ &\quad - \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r}\|^2. \end{aligned} \quad (73)$$

Since $\upsilon(z;\lambda,\beta) = \lambda^\top z + \frac{\beta}{2}\|z\|^2$ is a convex function, whose gradient is $\nabla\upsilon(z;\lambda,\beta) = \lambda + \beta z$,

$$\begin{aligned} &\upsilon(z^{k,r+1};\lambda^k,\beta^k) - \upsilon(z^{k,r};\lambda^k,\beta^k) \\ &\leq (\lambda^k + \beta^k z^{k,r+1})^\top(z^{k,r+1} - z^{k,r}), \end{aligned} \quad (74)$$

From Line 11 of Algorithm 1 it can be obtained

$$\lambda^k + \beta z^{k,r+1} = -y^{k,r+1}. \quad (75)$$

Substituting into (73), we obtain

$$\begin{aligned} &L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) - L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r}, y^{k,r}) \\ &\leq (y^{k,r+1} - y^{k,r})^\top(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r}) \\ &\quad + \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}\|^2 \\ &\quad - \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r}\|^2 \\ &= \frac{\rho^k}{2}(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1})^\top(Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r}) \\ &\quad + \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}\|^2 \\ &\quad - \frac{\rho^k}{2}\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r}\|^2 \\ &= -\frac{\rho^k}{2}\|z^{k,r+1} - z^{k,r}\|^2 + \rho^k\|Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1}\|^2 \end{aligned} \quad (76)$$

From (75),

$$\begin{aligned} Ax^{k,r+1} + B\bar{x}^{k,r+1} + z^{k,r+1} &= \frac{1}{\rho_k}(y^{k,r+1} - y^{k,r}) \\ &= -\frac{\beta^k}{\rho^k}(z^{k,r+1} - z^{k,r}). \end{aligned} \quad (77)$$

Then (76) becomes

$$\begin{aligned} &L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) - L(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r}, y^{k,r}) \\ &\leq -\left(\frac{\rho^k}{2} - \frac{(\beta^k)^2}{\rho^k}\right)\|z^{k,r+1} - z^{k,r}\|^2 = -\frac{\beta^k}{2}\|z^{k,r+1} - z^{k,r}\|^2. \end{aligned} \quad (78)$$

Summing up the inequalities (67), (72) and (78), we have proved the inequality (23). Next, we show that the augmented Lagrangian is lower bounded, and hence is convergent towards some $\underline{L}^k \in \mathbb{R}$. We note that $\upsilon(z;\lambda,\beta)$ is a convex function of modulus β, it can be easily verified that

$$\begin{aligned} \upsilon(z^{k,r};\lambda^k,\beta^k) + (\lambda^k + \beta^k z^{k,r})^\top(z' - z^{k,r}) \\ + \frac{\rho^k}{2}\|z' - z^{k,r}\|^2 \geq \upsilon(z';\lambda^k,\beta^k) \end{aligned} \quad (79)$$

for any z', i.e.,

$$\begin{aligned} &\upsilon(z^{k,r};\lambda^k,\beta^k) + y^{k,r\top}(z^{k,r} - z') \\ &\geq \upsilon(z';\lambda^k,\beta^k) - \frac{\rho^k}{2}\|z' - z^{k,r}\|^2. \end{aligned} \quad (80)$$

Let $z' = -(Ax^{k,r} + B\bar{x}^{k,r})$ and remove the last term on the right-hand side. Then

$$\begin{aligned} \upsilon(z^{k,r};\lambda^k,\beta^k) + y^{k,r\top}(Ax^{k,r} + B\bar{x}^{k,r} + z^{k,r}) \\ \geq \upsilon(-(Ax^{k,r} + B\bar{x}^{k,r});\lambda^k,\beta^k). \end{aligned} \quad (81)$$

Hence

$$\begin{aligned} &L(x^{k,r}, \bar{x}^{k,r+1}, z^{k,r}, y^{k,r}) = f(x^{k,r}) + g(\bar{x}^{k,r}) + \upsilon(z^{k,r};\lambda^k,\beta^k) \\ &+ y^{k,r\top}(Ax^{k,r} + B\bar{x}^{k,r} + z^{k,r}) + \frac{\rho^k}{2}\|Ax^{k,r} + B\bar{x}^{k,r} + z^{k,r}\|^2 \\ &\geq f(x^{k,r}) + g(\bar{x}^{k,r}) + \upsilon(-(Ax^{k,r} + B\bar{x}^{k,r});\lambda^k,\beta^k). \end{aligned} \quad (82)$$

Since $\upsilon(z) = \lambda^\top z + \frac{\beta}{2}\|z\|^2 \geq -\|\lambda\|^2/(2\beta)$, λ is bounded in $[\underline{\lambda}, \overline{\lambda}]$, $\beta^k \geq \beta^1$, and f and g are bounded below, L has a lower bound. Lemma 1 is proved.

Appendix B
Proof of Corollary 1

Taking the limit $r \to \infty$ on the both sides of inequality (23), it becomes obvious that $B\bar{x}^{k,r+1} - B\bar{x}^{k,r}$ and $z^{k,r+1} - z^{k,r}$ converge to 0. Due to (77), we have $Ax^{k,r} + B\bar{x}^{k,r} + z^{k,r} \to 0$. Hence there must exist a r such that (22) is met. At this time, the optimality conditions for $x^{k,r+1}$ is written as

$$\begin{aligned} 0 \in &\partial f(x^{k,r+1}) + \mathcal{N}_{\mathcal{X}}(x^{k,r+1}) + A^\top y^{k,r} \\ &+ \rho^k A^\top(Ax^{k,r+1} + B\bar{x}^{k,r} + z^{k,r}). \end{aligned} \quad (83)$$

According to the update rule of $y^{k,r}$, the above expression is equivalent to

$$\begin{aligned} 0 \in &\partial f(x^{k,r+1}) + \mathcal{N}_{\mathcal{X}}(x^{k,r+1}) + A^\top y^{k,r+1} \\ &- \rho^k A^\top(B\bar{x}^{k,r+1} + z^{k,r+1} - B\bar{x}^{k,r} - z^{k,r}), \end{aligned} \quad (84)$$

i.e.,

$$\begin{aligned} &\rho^k A^\top(B\bar{x}^{k,r+1} + z^{k,r+1} - B\bar{x}^{k,r} - z^{k,r}) \\ &\in \partial f(x^{k,r+1}) + \mathcal{N}_{\mathcal{X}}(x^{k,r+1}) + A^\top y^{k,r+1}. \end{aligned} \quad (85)$$

According to the first inequality of (22), the norm of the left hand side above is not larger than ϵ_1^k, which directly implies the first condition in (24). In a similar manner, the second condition in (24) can be established. The third one follows from (75) and the fourth condition is obvious.

APPENDIX C
PROOF OF LEMMA 2

We first consider the situation when β^k is unbounded. From (82), we have

$$\begin{aligned}\overline{L} \geq & f(x^{k+1}) + g(x^{k+1}) \\ & - \lambda^{k\top}(Ax^{k+1} + B\bar{x}^{k+1}) + \frac{\beta^k}{2}\|Ax^{k+1} + B\bar{x}^{k+1}\|^2.\end{aligned} \tag{86}$$

Since f and g are both lower bounded, as $\beta^k \to \infty$, we have $Ax^{k+1} + B\bar{x}^{k+1} \to 0$. Combined with the first two conditions of (24) in the limit of ϵ_1^k, ϵ_2^k, $\epsilon_3^k \downarrow 0$, we have reached (26).

Then we suppose that β^k is bounded, i.e., the amplification step $\beta^{k+1} = \gamma\beta^k$ is executed for only a finite number of outer iterations. According to Lines 17–21 of Algorithm 1, expect for some finite choices of k, $\|z^{k+1}\| \leq \omega\|z^k\|$ always hold. Therefore $z^{k+1} \to 0$. Apparently, (26) follows from the limit of (24).

APPENDIX D
PROOF OF LEMMA 3

From Lemma 1 one knows that within R inner iterations

$$\frac{\overline{L} - \underline{L}^k}{\beta^k} \geq \sum_{r=1}^{R}\left(\|B\bar{x}^{k,r+1} - B\bar{x}^{k,r}\|^2 + \frac{1}{2}\|\tilde{z}^{k,r+1} - z^{k,r}\|^2\right). \tag{87}$$

Then

$$\|B\bar{x}^{k,R+1} - B\bar{x}^{k,R}\|, \ \ \|z^{k,R+1} - z^{k,R}\| \sim \mathcal{O}(1/\sqrt{\beta^k R}). \tag{88}$$

For the k-th outer iteration, its inner iterations are terminated when (22) is met, which is translated into the following relations:

$$\begin{aligned}&\mathcal{O}(\rho^k/\sqrt{\beta^k R^k}) \leq \epsilon_1^k \sim \mathcal{O}(\vartheta^k), \\ &\mathcal{O}(\rho^k/\sqrt{\beta^k R^k}) \leq \epsilon_2^k \sim \mathcal{O}(\vartheta^k), \\ &\mathcal{O}(1/\sqrt{\beta^k R^k}) \leq \epsilon_3^k \sim \mathcal{O}(\vartheta^k/\beta^k).\end{aligned} \tag{89}$$

where the last relation uses (77) with $\rho^k = 2\beta^k$. Therefore

$$R^k \sim \mathcal{O}(\beta^k/\vartheta^{2k}). \tag{90}$$

At the end of the k-th iteration, suppose that Lines 19–20 and Lines 17–18 of Algorithm 1 have been executed for k_1 and k_2 times, respectively ($k_1 + k_2 = k$). Then the obtained z^{k+1} satisfies $\|z^{k+1}\| \sim \mathcal{O}(\omega^{k_1})$, and $\|Ax^{k+1} + B\bar{x}^{k+1} + z^{k+1}\| \leq \epsilon_3^k \sim \mathcal{O}(\vartheta^k/\beta^k)$, which imply

$$\|Ax^{k+1} + B\bar{x}^{k+1}\| \leq \mathcal{O}(\vartheta^k/\beta^k) + \mathcal{O}(\omega^{k_1}). \tag{91}$$

From (86),

$$\beta^k\|Ax^{k+1} + B\bar{x}^{k+1}\|^2 \sim \beta^k(\mathcal{O}(\vartheta^k/\beta^k) + \mathcal{O}(\omega^{k_1}))^2 \sim \mathcal{O}(1). \tag{92}$$

Substituting (92) into (90), we obtain

$$R^k \sim \mathcal{O}\left(\frac{1}{\vartheta^{2k}}\frac{1}{(\mathcal{O}(\vartheta^k/\beta^k) + \mathcal{O}(\omega^{k_1}))^2}\right). \tag{93}$$

When $\vartheta \leq \omega$, $\vartheta^k \leq \omega^k \leq \omega^{k_1}\gamma^{k_2}$, and hence $\gamma^{k_2}\vartheta^k \leq \omega^{k_1}$, i.e., ω^{k_1} dominates over ϑ^k/β^k, leading to

$$R^k \sim \mathcal{O}(1/\vartheta^{2k}\omega^{2k_1}) \sim \mathcal{O}(1/\vartheta^{2k}\omega^{2k}). \tag{94}$$

For K outer iterations, the total number of inner iterations is

$$R = \sum_{k=1}^{K} R^k \sim \mathcal{O}\left(\sum_{k=1}^{K}\frac{1}{\vartheta^{2k}\omega^{2k}}\right) \sim \mathcal{O}\left(\frac{1}{\vartheta^{2K}\omega^{2K}}\right). \tag{95}$$

The number of outer iterations needed to reach an ϵ-approximate stationary point is obviously $K \sim \mathcal{O}(\log_\vartheta \epsilon)$. Then

$$R \sim \mathcal{O}(\epsilon^{-2(1+\varsigma)}). \tag{96}$$

APPENDIX E
PROOF OF LEMMA 6

Through the inner iterations, only Anderson acceleration might lead to an increase in the barrier augmented Lagrangian. Combining Assumption 3, Assumption 5, and the safeguarding criterion (48), we obtain

$$\begin{aligned}&L_{b^k}(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) \\ &\leq \overline{L} + \tilde{L}_0\eta_L\sum_{r=0}^{\infty}\frac{1}{r^{1+\sigma}} < +\infty,\end{aligned} \tag{97}$$

Together with Assumptions 1 and 2, L_{b^k} is also bounded below. Therefore L_{b^k} is bounded in a closed interval and must have converging subsequences. Therefore we can choose a subsequence converging to the lower limit $\underline{L}$. For any $\varepsilon > 0$ there exists an index R of inner iteration in this subsequence, such that $\tilde{L}_0\eta_L\sum_{r=R}^{\infty} r^{-(1+\sigma)} < \varepsilon/2$ and $L_{b^k}(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) < \underline{L} + \varepsilon/2$ for any $r \geq R$ on this subsequence. It then follows that for any $r \geq R$, whether on the subsequence or not, it holds that

$$L_{b^k}(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) < \underline{L} + \varepsilon. \tag{98}$$

Hence the upper limit is not larger than $\underline{L} + \varepsilon$. Due to the arbitrariness of $\varepsilon > 0$, the lower limit coincides with the upper limit, and hence the sequence of barrier augmented Lagrangian is convergent.

The convergence of the barrier augmented Lagrangian implies that as $r \to \infty$, $L_{b^k}(x^{k,r+1}, \bar{x}^{k,r+1}, z^{k,r+1}, y^{k,r+1}) - L_{b^k}(x^{k,r}, \bar{x}^{k,r}, z^{k,r}, y^{k,r}) \to 0$. Suppose that r is not an accelerated iteration, then since this quantity does not exceed $-\beta^k\|B\bar{x}^{k,r+1} - B\bar{x}^{k,r}\|^2 - (\beta^k/2)\|z^{k,r+1} - z^{k,r}\|^2$, we must have $B\bar{x}^{k,r+1} - B\bar{x}^{k,r} \to 0$ and $z^{k,r+1} - z^{k,r} \to 0$. Otherwise if inner iteration r is accelerated, the convergence of $B\bar{x}^{k,r+1} - B\bar{x}^{k,r}$ and $z^{k,r+1} - z^{k,r}$ are automatically guaranteed by the second criterion (50) of accepting Anderson acceleration. The convergence properties of these two sequences naturally fall into the paradigm of Lemma 1 for establishing the convergence to approximate KKT conditions of the relaxed problem.

ACKNOWLEDGMENT

This work was supported by National Science Foundation (NSF-CBET). The authors would also like to thank Prof. Qi Zhang for his constructive opinions.

REFERENCES

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

[2] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, 2014.

[3] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[4] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model predictive control: theory, computation, and design*, 2nd ed. Nob Hill Publishing, 2017.

[5] P. Daoutidis, W. Tang, and S. S. Jogwar, "Decomposing complex plants for distributed control: perspectives from network theory," *Comput. Chem. Eng.*, vol. 114, pp. 43–51, 2018.

[6] W. Tang, A. Allman, D. B. Pourkargar, and P. Daoutidis, "Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection," *Comput. Chem. Eng.*, vol. 111, pp. 43–54, 2018.

[7] P. Daoutidis, W. Tang, and A. Allman, "Decomposition of control and optimization problems by network structure: concepts, methods and inspirations from biology," *AIChE J.*, vol. 65, no. 10, p. e16708, 2019.

[8] R. Scattolini, "Architectures for distributed and hierarchical model predictive control – a review," *J. Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[9] P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Comput. Chem. Eng.*, vol. 51, pp. 21–41, 2013.

[10] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Syst. Mag.*, vol. 34, no. 4, pp. 87–97, 2014.

[11] M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using h_p-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw. (TOMS)*, vol. 41, no. 1, pp. 1–37, 2014.

[12] Y. Mao, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems and its convergence properties," in *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3636–3641.

[13] L. T. Biegler and D. M. Thierry, "Large-scale optimization formulations and strategies for nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 1–15, 2018, the 6th IFAC Conference on Nonlinear Model Predictive Control (NMPC).

[14] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Syst. Control Lett.*, vol. 59, no. 8, pp. 460–469, 2010.

[15] J. Liu, X. Chen, D. Muñoz de la Peña, and P. D. Christofides, "Sequential and iterative architectures for distributed model predictive control of nonlinear process systems," *AIChE J.*, vol. 56, no. 8, pp. 2137–2149, 2010.

[16] X. Chen, M. Heidarinejad, J. Liu, and P. D. Christofides, "Distributed economic MPC: Application to a nonlinear chemical process network," *J. Process Control*, vol. 22, no. 4, pp. 689–699, 2012.

[17] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proc. 44th Conf. Decis. Control (CDC)*. IEEE, 2005, pp. 6680–6685.

[18] M. Farina and R. Scattolini, "Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems," *Automatica*, vol. 48, no. 6, pp. 1088–1096, 2012.

[19] 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*, vol. 49, no. 3, pp. 829–833, 2013.

[20] L. Grüne and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.

[21] F. Farokhi, I. Shames, and K. H. Johansson, "Distributed MPC via dual decomposition and alternative direction method of multipliers," in *Distributed model predictive control made easy*. Springer, 2014, pp. 115–131.

[22] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "Distributed optimization with local domains: applications in MPC and network flows," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 2004–2009, 2014.

[23] S. Magnússon, P. C. Weeraddana, M. G. Rabbat, and C. Fischione, "On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 3, pp. 296–309, 2015.

[24] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[25] N. Chatzipanagiotis and M. M. Zavlanos, "On the convergence of a distributed augmented Lagrangian method for nonconvex optimization," *IEEE Trans. Automatic Control*, vol. 62, no. 9, pp. 4405–4420, 2017.

[26] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, 2019.

[27] J.-H. Hours and C. N. Jones, "A parametric nonconvex decomposition algorithm for real-time and distributed NMPC," *IEEE Trans. Autom. Control*, vol. 61, no. 2, pp. 287–302, 2015.

[28] B. Houska, J. Frasch, and M. Diehl, "An augmented Lagrangian based algorithm for distributed nonconvex optimization," *SIAM J. Optim.*, vol. 26, no. 2, pp. 1101–1127, 2016.

[29] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization – part I: theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, 2016.

[30] K. Sun and X. A. Sun, "A two-level distributed algorithm for general constrained non-convex optimization with global convergence," *arXiv preprint arXiv:1902.07654*, 2019.

[31] B. Jiang, T. Lin, S. Ma, and S. Zhang, "Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis," *Comput. Optim. Appl.*, vol. 72, no. 1, pp. 115–157, 2019.

[32] Y. Yang, G. Hu, and C. J. Spanos, "A proximal linearization-based fecentralized mthod for nonconvex problems with nonlinear constraints," *arXiv preprint arXiv:2001.00767*, 2020.

[33] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the ADMM with multiblock variables," *SIAM J. Optim.*, vol. 25, no. 3, pp. 1478–1497, 2015.

[34] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Math. Prog.*, vol. 162, no. 1-2, pp. 165–199, 2017.

[35] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5082–5095, 2017.

[36] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM J. Imaging Sci.*, vol. 7, no. 3, pp. 1588–1623, 2014.

[37] Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao Jr, "An accelerated linearized alternating direction method of multipliers," *SIAM J. Imaging Sci.*, vol. 8, no. 1, pp. 644–681, 2015.

[38] W. Tang and P. Daoutidis, "Distributed nonlinear model predictive control through accelerated parallel ADMM," in *Am. Control Conf.* IEEE, 2019, pp. 1406–1411.

[39] R. Y. Zhang and J. K. White, "GMRES-accelerated ADMM for quadratic objectives," *SIAM J. Optim.*, vol. 28, no. 4, pp. 3025–3056, 2018.

[40] J. Zhang, B. O'Donoghue, and S. Boyd, "Globally convergent type-I Anderson acceleration for non-smooth fixed-point iterations," *arXiv preprint arXiv:1808.03971*, 2018.

[41] A. Fu, J. Zhang, and S. Boyd, "Anderson accelerated Douglas-Rachford splitting," *arXiv preprint arXiv:1908.11482*, 2019.

[42] J. Zhang, Y. Peng, W. Ouyang, and B. Deng, "Accelerating ADMM for efficient simulation and optimization," *ACM Trans. Graph.*, vol. 38, no. 6, p. 163, 2019.

[43] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović, "The proximal augmented Lagrangian method for nonsmooth composite optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2861–2868, 2019.

[44] D. Hajinezhad and M. Hong, "Perturbed proximal primal-dual algorithm for nonconvex nonsmooth optimization," *Math. Prog.*, vol. 176, no. 1-2, pp. 207–245, 2019.

[45] J. Eckstein and W. Yao, "Approximate ADMM algorithms derived from Lagrangian splitting," *Comput. Optim. Appl.*, vol. 68, no. 2, pp. 363–405, 2017.

[46] D. P. Bertsekas, *Nonlinear programming*, 3rd ed. Athena Scientific, 2016.

[47] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *Rev. Fr. Autom. Inform. Rech. Opér., Anal. Numr.*, vol. 9, no. R2, pp. 41–76, 1975.

[48] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.

[49] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Prog.*, vol. 55, no. 1-3, pp. 293–318, 1992.

[50] J. Xie, A. Liao, and X. Yang, "An inexact alternating direction method of multipliers with relative error criteria," *Optim. Lett.*, vol. 11, no. 3, pp. 583–596, 2017.

[51] Yu. E. Nesterov, "A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$," *Dokl. Akad. Nauk SSSR*, vol. 269, no. 3, pp. 543–547, 1983.

[52] D. G. Anderson, "Iterative procedures for nonlinear integral equations," *J. ACM*, vol. 12, no. 4, pp. 547–560, 1965.

[53] P. Pulay, "Convergence acceleration of iterative sequences. the case of SCF iteration," *Chem. Phys. Lett.*, vol. 73, no. 2, pp. 393–398, 1980.

[54] H.-r. Fang and Y. Saad, "Two classes of multisecant methods for nonlinear acceleration," *Numer. Linear Algebra Appl.*, vol. 16, no. 3, pp. 197–221, 2009.

[55] A. Toth and C. Kelley, "Convergence analysis for Anderson acceleration," *SIAM J. Numer. Anal.*, vol. 53, no. 2, pp. 805–819, 2015.

[56] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.

[57] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[58] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer, 1998.

[59] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: Motivation and global convergence," *SIAM J. Optim.*, vol. 16, no. 1, pp. 1–31, 2005.

[60] ——, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Prog.*, vol. 106, no. 1, pp. 25–57, 2006.

[61] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Prog.*, vol. 155, no. 1-2, pp. 57–79, 2016.

[62] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 456–465, 2000.

[63] B. Nicholson, J. D. Siirola, J.-P. Watson, V. M. Zavala, and L. T. Biegler, "`pyomo.dae`: a modeling and automatic discretization framework for optimization with differential and algebraic equations," *Math. Prog. Comput.*, vol. 10, no. 2, pp. 187–223, 2018.

Wentao Tang was born in Yongzhou, Hunan Province, P. R. China. He received a Bachelor of Science degree in Chemical Engineering and a secondary degree in Mathematics from Tsinghua University, Beijing, China, in 2015. He is now pursuing a Ph.D. degree in Chemical Engineering at University of Minnesota. He is the recipient of the Doctoral Dissertation Fellowship of University of Minnesota for 2018–2019, and the 1st place in CAST Directors' Student Presentation Award of the 2019 AIChE Annual Meeting. His current research interests include the architecture design and algorithm of distributed and hierarchical control and optimization problems, nonlinear system identification, and data-driven control of nonlinear processes.

Prodromos Daoutidis is a College of Science and Engineering Distinguished Professor and Executive Officer in the Department of Chemical Engineering and Materials Science at the University of Minnesota. He received a Diploma degree in Chemical Engineering (1987) from the Aristotle University of Thessaloniki, M.S.E. degrees in Chemical Engineering (1988) and Electrical Engineering: Systems (1991) from the University of Michigan, and a Ph.D. degree in Chemical Engineering (1991) from the University of Michigan. He has been on the faculty at Minnesota since 1992, while he has also held a position as Professor at the Aristotle University of Thessaloniki (2004–06). He is the recipient of several awards and recognitions, including the AIChE Computing in Chemical Engineering Award, the PSE Model Based Innovation Prize, the Best Paper Prize from the *Journal of Process Control*, an NSF Career Award, and the AIChE Ted Peterson Award. He has also been a Humphrey Institute Policy Fellow. He is the Associate Editor for Process Systems Engineering in the *AIChE Journal*, and an Associate Editor in the *Journal of Process Control*. He has co-authored 5 books, 290 refereed papers, and has supervised to completion 35 Ph.D. students and post-docs. His current research is on control and optimization of complex and networked systems, and the design and operation of distributed renewable systems for power generation and production of fuels and chemicals.