

Consensus-Based Distributed Optimization Enhanced by Integral Feedback

Xuan Wang , Shaoshuai Mou , and Brian D. O. Anderson 

Abstract—Inspired and underpinned by the idea of integral feedback, a distributed constant gain algorithm is proposed for multi-agent networks to solve convex optimization problems with local linear constraints. Assuming agent interactions are modeled by an undirected graph, the algorithm is capable of achieving the optimum solution with an exponential convergence rate. Furthermore, inherited from the beneficial integral feedback, the proposed algorithm has attractive requirements on communication bandwidth and good robustness against disturbance. Both analytical proof and numerical simulations are provided to validate the effectiveness of the proposed distributed algorithms in solving constrained optimization problems.

Index Terms—Distributed optimization, integral feedback, multi-agent networks.

I. INTRODUCTION

Collective behaviors in nature have motivated rapidly expanding research efforts in the control of multiagent systems [1]. A multiagent system is composed of multiple interacting subsystems (agents), which makes them more challenging to control than single monolithic systems. Specifically, the network constraints, stemming from the relations between agents involving sensing, communication, or control, usually prohibit the application of traditional methods from controlling the multiagent systems in a centralized manner. In order to seek new control approaches that respect the network nature of multiagent systems, *distributed control* has recently received a significant amount of research attention, the goal of which is to allow multiagent systems to accomplish global objectives through only local coordination. Here, the word “local” connotes interaction between any given agent and a limited number of associated “neighbor” agents, often physically adjacent.

One of the key problems in multiagent control is distributed optimization, where each agent privately processes one objective function and one constraint, and the goal is to minimize the sum of local

objective functions [2]–[5] subject to all local constraints [6]–[11]. To solve this problem, one applicable approach is the alternating direction method of multipliers (ADMM). While the ADMM method originates from Lagrange duality, it usually needs a centralized state to coordinate across agents [12]. This limits the applicability of the method to fully distributed network systems. In order to remove the requirement for such a centralized state, many efforts have been made. For example, [13], [14] show the duality can instead be established by introducing slack states on the edges of the network; and in [14], [15], the centralized state can be further decomposed by the primal decomposition technique. Apart from the ADMM methods, another family of distributed coordination method arises from the idea of *consensus* [16]. The aim of consensus is to drive all agents in the network to reach an agreement regarding a certain quantity, which has served as a basis in deriving many distributed algorithms for multiagent systems such as motion synchronization [1], multirobot path planning/formation control [17], flocking of mobile robots [18], and cooperative sensing [19]. Fitting in the scope of distributed optimization, *consensus* is usually incorporated with *gradient descent* and *projection* operators, which handle objective functions and local constraints, respectively. However, note that in multiagent systems, different agents may have different local objective functions/constraints. This means that in general, the consensus, gradient descent, and projection operators will have different equilibria, so that the states in all agents can never converge to a same point. To circumvent this difficulty, the work in [20]–[23] applies a diminishing gain (i.e., $1/t$) to the gradient term in the update equations. As a side effect, this time-variant gain must be shared by all the agents in the network, and the convergence rate of the algorithm will be degraded most commonly to $\mathcal{O}(1/\sqrt{t})$. In order to improve the convergence rate, many recent works have shown that a possible approach is to double the dimension of the state vector. See, for example, the continuous-time update introduced in [9]–[11], [24], where the role of the extra vector is played by the Lagrangian dual vector for consensus errors; see also the discrete-time update introduced in [25], [26], where the extra vector performs gradient tracking; and the discrete-time update introduced in [27], [28], where the extra vectors combines proximal gradient and gradient tracking together. In these algorithms, the extra states have to be exchanged across the network, necessitating duplication of the network bandwidth requirement. Focusing on the convergence rate and the states to be exchanged across the network, a detailed comparison between this article and the related works is provided in Table I.

In this article, we propose a distributed algorithm for constrained optimization that is neither based on diminishing gains nor a doubled dimension of the vectors shared between agents. Actually, by comparing the very fundamental mechanisms of these algorithms, we notice that the common reason why the latter category of algorithms can achieve an improved convergence performance arises from elevating the type of the update to second order, and thereby effectively eliminating the accumulated consensus error. Inspired by this, in this article, we propose a continuous-time consensus-based algorithm for constrained distributed optimization based on integral feedback within each agent's

Manuscript received 5 November 2020; revised 7 November 2020, 28 June 2021, and 6 February 2022; accepted 15 April 2022. Date of publication 21 April 2022; date of current version 28 February 2023. This work was supported in part by NASA University Leadership Initiative (ULI) under Grant 80NSSC20M0161 and in part by Northrop Grumman Corporation. The work of B. D. O. Anderson was supported in part by Data61-CSIRO and in part by Australian Research Council under Grant DP190100887. Recommended by Associate Editor M. K. Camlibel. (Corresponding author: Xuan Wang.)

Xuan Wang is with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA (e-mail: xwang64@gmu.edu).

Shaoshuai Mou is with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906 USA (e-mail: shaoshuai.mou@aya.yale.edu).

Brian D. O. Anderson is with the Australian National University, Acton, ACT 2601, Australia, and also with the Hangzhou Dianzi University, Hangzhou 310005, China (e-mail: Brian.Anderson@anu.edu.au).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2022.3169179>.

Digital Object Identifier 10.1109/TAC.2022.3169179

TABLE I
COMPARISON WITH EXISTING ALGORITHMS

Algorithm	Key idea	State to share	Local constraints	Update form	Exponential Convergence
The proposed algorithm	Integral Feedback	Dimension n	Linear, closed	Continuous	Yes
Algorithms in [21]	Diminishing step-size/gain	Dimension n	Compact	Discrete	No, $\mathcal{O}(1/\sqrt{t})$
Algorithms in [29]	Integral Feedback [†]	Dimension n	Not applicable	Continuous	Yes
Algorithms in [26], [30], [31]	Gradient tracking	Dimension $2n^*$	Not applicable	Discrete	Yes
Algorithm in [24]	Saddle point dynamics	Dimension $2n$	Not applicable	Continuous	No theoretical guarantee
Algorithms in [9]–[11]	Saddle point dynamics	Dimension $2n$	Closed	Continuous	No theoretical guarantee

[†] Note that [29] requires an extra process for designing control gains. In this article, the process is not required.

* In [26], certain choices for its parameter design can lead to a discrete counterpart of the result in this article. In this case, its *state to share* is reduced to n .

controller, and with the integrated signal not being shared with other agents. The contributions of the article in more detail are as follows: 1) Without a time-variant gain that needs to be shared by agents, the algorithm is capable of achieving the optimum solution with an asymptotic convergence rate for general convex objective functions with nonunique minimizers, and a global exponential convergence rate for strongly convex functions. 2) To achieve exponential convergence, the proposed result only requires *the sum of all objective functions to be strongly convex at the optimum point*¹; this is a more relaxed condition compared with the existing results that require all agents' local objective functions to be strongly convex [26]–[31]. 3) Inherited from the benefit of integral feedback, it can be theoretically guaranteed that the proposed algorithm has good robustness against disturbances. 4) Apart from requiring the storing at each agent of the integral of the state vector in addition to the state vector itself, the algorithm does not introduce an extra state vector which has to be exchanged among the agents of the network. This further distinguishes the work from the existing results based on (primal–dual) saddle point dynamics in [9]–[11]. Note that the algorithm of this article is evidently related to the discrete-time algorithm for unconstrained optimization [26]–[28], [31], [32]. These authors increased the state dimension by using a form of gradient descent including the last two iterates. On the other hand, as noted already, our algorithm is motivated by the very old principle of using integral feedback to cancel steady state errors. Different from [26], [29]–[32], our algorithm can additionally handle local linear constraints, which commonly exist in many engineering application [33], [34]. Apart from the benefits, we are aware that the continuous-time nature of the proposed algorithm can implicitly raise one question: We cannot quantify straightforwardly the algorithm's convergence speed in terms of the number of iterations of a discretized version of the algorithm. As a further consequence, we are not able to measure the total amount of information exchange required by the algorithm to reach a certain error bound and compare it with other algorithms. These concerns will be addressed in our future work based on a discretization of the proposed algorithm.

The rest of this article is organized as follows. In Section II, we describe the information flow of the multiagent network and formulate the problem of constrained distributed optimization. By introducing the idea of integral feedback, in Section III, we propose a continuous-time algorithm that can solve distributed optimization problems with linear constraints. The effectiveness, exponential convergence, and robustness of the algorithm are theoretically proved in Section IV. Section V provides the numerical validation for the convergence rate and the robustness against disturbance. Finally, Section VI concludes this article.

Notation: Let $\mathbf{1}_r$ denote the vector in $\mathbb{R}^r$ with all entries equal to 1. Let I_r denote the $r \times r$ identity matrix. We let $\text{col}\{A_1, A_2, \dots, A_r\}$ be a stack of matrices A_i possessing the same number of columns with the index in a top-down ascending order, $i = 1, 2, \dots, r$. By $M^\top$ is

meant the transpose of a matrix M . Let $\ker M$ and $\text{image } M$ denote the kernel and image of a matrix M , respectively. Let $\otimes$ denote the Kronecker product.

II. PROBLEM FORMULATION

Consider a network of m agents in which each agent i is able to communicate with certain other nearby agents called its neighbors, denoted by $\mathcal{N}_i$. The neighbor relations can be described by a graph $\mathbb{G}$ such that there is an edge from j to i if and only if $j \in \mathcal{N}_i$. We assume $\mathbb{G}$ is *connected* and *undirected*. Associated with each agent is a local state $x_i \in \mathbb{R}^n$; a convex function $f_i(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$; and a linear constraint $A_i x_i = b_i$, where $A_i \in \mathbb{R}^{n_i \times n}$ and $b_i \in \text{image } A_i \subset \mathbb{R}^{n_i}$. The problem of *interest* is to develop a *distributed* algorithm, which enables all nodes of $\mathbb{G}$ to reach a consensus value solving the problem

$$\text{minimize} \quad \sum_{i=1}^m f_i(x_i) \quad (1)$$

$$\text{subject to} \quad A_i x_i = b_i, \quad i = 1, \dots, m \quad (2)$$

$$x_1 = x_2 = \dots = x_m. \quad (3)$$

Remark 1: The linear constraint in (2) arises naturally from many engineering applications [33], [34]. Here, to avoid trivialities, we assume $b_i \in \text{image } A_i$, and $\text{rank}(A) < n$, where $A = \text{col}\{A_1, \dots, A_m\}$. This guarantees the optimization domain defined by (2)–(3) is nonempty and nonunique.

III. ALGORITHM AND MAIN RESULT

In order to solve the consensus-based constrained optimization problem characterized by (1)–(3), we first introduce some preliminary results.

A. Preliminary Results

Distributed Consensus: Consider a continuous-time update of the following form [35]:

$$\dot{x}_i = - \sum_{j \in \mathcal{N}_i} (x_i - x_j) \quad (4)$$

where each agent in the network tries to reduce the distances between itself and its neighbors. Update (4) will drive all states to a consensus value, i.e., there exists a certain $x^* = \mathbf{1}_m \otimes x^*$, such that $x_i - x^* \rightarrow 0$ for all i and the convergence is exponentially fast.

Consensus-Based Distributed Optimization: Leaving aside temporarily constraints (2), the attempt to find an algorithm also achieving the optimization objective (1) requires the introduction of a gradient term, so that the update of each agent becomes

$$\dot{x}_i = -\alpha(t) \nabla f_i(x_i) - \sum_{j \in \mathcal{N}_i} s_{ij} (x_i - x_j) \quad (5)$$

¹ Refer to Assumption 2 of the main text for details.

where $\alpha(t)$ is a positive gain shared by all the agents and $\nabla f_i(x_i)$ is the gradient.²

In order to solve the consensus-based optimization problem, a novel algorithm is presented in [20]–[22], where the authors apply a diminishing gain $\alpha(t)$ to the discrete-time version of (5) to eliminate the consensus error. Furthermore, with introduction of an additional projection operator, this algorithm is also able to handle local constraints. A continuous version of the algorithm developed in [20]–[22] is

$$\dot{x}_i = P_i \left[-\alpha(t) \nabla f_i(x_i) - \sum_{j \in \mathcal{N}_i} (x_i - x_j) \right] \quad (6)$$

where $P_i[\cdot]$ is a projection operator that projects any vector to the tangent space³ of the agent's local constraint at the point $x_i(t)$, and it guarantees that x_i always satisfies the local constraint of agent i . By letting all agents share a diminishing gain such that $\alpha(t) \rightarrow 0$ and $\int_0^\infty \alpha(t) \rightarrow \infty$, it has been theoretically proved that the states will asymptotically reach a consensus at the minimizer of $\sum_{i=1}^m f_i(x)$ subject to all agents' local constraints [20], [21]. Meanwhile, since the effect of the gradient term is discounted by the diminishing gain, the convergence rate of the algorithm is at most $\mathcal{O}(1/\sqrt{t})$. Here, to improve the convergence rate, the key idea is to get rid of the diminishing gain $\alpha(t)$ in (6). This has led to the main result of this article.

B. Proposed Update and Main Result

In this article, instead of using a diminishing gain or an additional state vector to each agent which has to be exchanged with neighbors, our key idea stems from introducing an additional integral term to effectively eliminate the steady-state error on consensus. We propose the following continuous-time distributed algorithm:

$$\dot{x}_i = -P_i \left(\nabla f_i(x_i) + \sum_{j \in \mathcal{N}_i} (x_i - x_j) + \int_0^t \sum_{j \in \mathcal{N}_i} (x_i - x_j) \right) \quad (7)$$

where $x_i(0)$ are initialized such that $A_i x_i(0) = b_i$ and $P_i \in \mathbb{R}^{n \times n}$ is a projection matrix to $\ker A_i$.

Remark 2: Obviously, the proposed algorithm is distributed, because the state update of each agent only relies on the information of itself and that of its neighbors. In update (7), the projection matrix P_i and the special initialization on $x_i(0)$ are used to handle the linear constraint (2). As a special case, if for one or more agents, the linear constraint does not exist, then one can correspondingly initialize $x_i(0)$ as an arbitrary value and replace the projection matrix by an identity matrix. Further note that similarly to [29], [36], the integral term requires extra historical state information; as will be seen later in (11), such information can be equivalently stored in a local state $y_i = \int_0^t \sum_{j \in \mathcal{N}_i} (x_i - x_j)$, and moreover, this extra state does not need to be shared across the network.

Ahead of studying the properties of Algorithm (7), we introduce the following assumptions.

Assumption 1: For all $i = 1, \dots, m$, suppose $f_i(\cdot)$ is convex, continuously differentiable, and its gradient is Lipschitz. Suppose

²Note that if the function is not differentiable, the gradient can usually be replaced by a subgradient of the function with the convergence property being unchanged. However, since subgradients are not unique, this can lead to nonunique trajectories and requires the tool of Filippov solution for analysis. $f_i(x_i)$, $s_{ij} > 0$ is a parameter, which is usually associated with the edge weight of the network. Note that in (5), if $\alpha(t)$ is chosen as a fixed positive constant, unless all f_i are minimized by a common vector, there may not exist a steady state x^* for (5), which also follows the consensus property (1).

³For the linear constraints, one has $P_i[s] = P_i \cdot s$, where $P_i \in \mathbb{R}^{n \times n}$ is a projection matrix to $\ker A_i$.

rank $(A) < n$, where $A = \text{col}\{A_1, \dots, A_m\}$, such that the feasible set defined by constraints (2) and (3) is nonempty. Suppose a solution x^* to (1)–(3) exists, where x^* may be nonunique.⁴

Assumption 2: Suppose Assumption 1 holds. Further, suppose $F(x) = \sum_{i=1}^m f_i(x)$ is strongly convex⁵ at x^* , where x^* is the unique minimizer to problem (1)–(3).

Algorithm (7), along with Assumptions 1 and 2, allows us to propose the following theorem.

Theorem 1: Suppose the network $\mathbb{G}$ is connected and undirected; suppose Assumption 1 holds. Then, given any $x_i(0)$ such that $A_i x_i(0) = b_i$, update (7) drives the states $x_i(t)$ of all agents asymptotically to a point x^* , which is an optimum solution to (1)–(3). Furthermore, if Assumption 2 also holds, the convergence is globally exponentially fast.

Remark 3: Note immediately that in Assumption 2, we only require the sum $F(x) = \sum_{i=1}^m f_i(x)$ to be strongly convex (at x^*), which is a more relaxed condition compared with most of the existing results [26]–[28] that require each single $f_i(x_i)$ to be strongly convex. This allows the algorithm to handle the scenarios when linear objective functions or exponential penalty functions are associated with some of the agents.

Corollary 1: Suppose Assumption 2 holds. Consider the update under disturbance

$$\dot{x}_i = -P_i \left(\nabla f_i(x_i) + \sum_{j \in \mathcal{N}_i} (x_i - x_j) + \int_0^t \sum_{j \in \mathcal{N}_i} (x_i - x_j) \right) + v_i(t) \quad (8)$$

where $v_i(t)$ is a disturbance that is bounded for all $i = 1, \dots, m$ and $t \geq 0$. Then, the proposed algorithm (7) is robust against bounded disturbance, that is, as $t \rightarrow \infty$, one has that $\|x_i(t) - x^*\|$ remains bounded for all $i = 1, \dots, m$.

Remark 4: In Corollary 1, the disturbance $v_i(t)$ can frequently be caused by communication issues or sensor mismatches [37]. Also note that the robustness property proposed here is primarily a consequence of the exponential convergence of algorithm (7). On the contrary, for distributed algorithms that only have asymptotic convergence (i.e., $\mathcal{O}(1/t)$, $\mathcal{O}(1/\sqrt{t})$), the same disturbance will lead the output error to be unbounded. This statement will be validated later by simulations.

IV. ANALYSIS

This section proves the main results of the article.

A. Steady-State Analysis

We first propose the following lemma, which shows that the equilibrium point of (7) exists, and it is consistent with the solution to problems (1)–(3).

Lemma 1: Consider the updates (7) under Assumption 1, where $\mathbb{G}$ is connected and undirected. Then, an equilibrium point x_i^* of (7) exists. Furthermore, for all $i = 1, \dots, m$, there holds $x_i^* = x^*$, ensuring that the equilibrium point obeys the consensus property and optimizes the constrained optimization problem (1)–(3).

For simplicity in analyzing the proposed algorithm from a global perspective, define $\mathbf{x} = \text{col}\{x_1, \dots, x_m\} \in \mathbb{R}^{mn}$, $\bar{P} = \text{diag}\{P_1, \dots, P_m\} \in \mathbb{R}^{mn \times mn}$, $\nabla f(\mathbf{x}) = \text{col}\{\nabla f_1(x_1), \dots, \nabla f_m(x_m)\}$, and $\bar{L} = L \otimes I_n \in \mathbb{R}^{mn \times mn}$, where $L \in \mathbb{R}^{m \times m}$ is the Laplacian matrix

⁴We acknowledge the suggestion of an anonymous reviewer to include a nonuniqueness condition in this assumption.

⁵A function $F(\cdot)$ is called strongly convex at x^* with parameter $\omega > 0$ if $a^\top [\nabla F(a + x^*) - \nabla F(x^*)] \geq \omega \|a\|_2^2$ holds for all vectors a in its domain.

of the graph $\mathbb{G}^6$. Then, update (7) can be rewritten as

$$\dot{\mathbf{x}} = -\bar{P} \left(\nabla f(\mathbf{x}) + \bar{L}\mathbf{x} + \int_0^t \bar{L}\mathbf{x} \right). \quad (9)$$

This is further equivalent to

$$\dot{\mathbf{x}} = -\bar{P} (\nabla f(\mathbf{x}) + \bar{L}\mathbf{x} + \mathbf{y}) \quad (10)$$

$$\dot{\mathbf{y}} = \bar{L}\mathbf{x} \quad (11)$$

where $\mathbf{y} \in \mathbb{R}^{mn}$ and $\mathbf{y}(0) = 0$.

Remark 5: Evidently, the dynamics (10)–(11) is equivalent to (7), where the integral term implicitly introduces an extra state $\mathbf{y}$. Each component of this extra state can be obtained via local computations and stored by each agent, and does not have to be exchanged across the network. For existing algorithms characterized by saddle-point dynamics, the extra states must be exchanged across the network [24], [38].

Proof of Lemma 1: To prove Lemma 1, since (10)–(11) and (7) are equivalent, it is sufficient to show that there exist equilibrium points $(\mathbf{x}^*, \mathbf{y}^*)$ such that

$$0 = -\bar{P} (\nabla f(\mathbf{x}^*) + \bar{L}\mathbf{x}^* + \mathbf{y}^*) \quad (12)$$

$$0 = \bar{L}\mathbf{x}^* \quad (13)$$

where $\mathbf{x}^* = \mathbf{1}_m \otimes x^*$ and x^* is a minimizer of $F(x) = \sum_{i=1}^m f_i(x)$ subject to $A_i x^* = b_i$ for all i ; $\nabla f(\mathbf{x}^*)$ is the column stack of the vectors $\left. \frac{df_i(x_i)}{dx_i} \right|_{x^*}$.

Existence: Based on Assumption (1), let x^* be a minimizer of $\sum_i f_i(x)$ subject to $A_i x = b_i$, $i = 1, \dots, m$. Then, by standard Lagrange multiplier theory [39], there exist Lagrange multipliers z_i^* such that

$$\sum_{i=1}^m \nabla f_i(x^*) - \sum_{i=1}^m A_i^\top z_i^* = 0. \quad (14)$$

Using the z_i^* , we make the definition

$$\mathbf{y}_i^* = A_i^\top z_i^* - \nabla f_i(x^*). \quad (15)$$

Based on x^* and $\mathbf{y}_i^*$, let

$$\mathbf{x}^* = \mathbf{1}_m \otimes x^* \quad \text{and} \quad \mathbf{y}^* = \text{col} \{y_1^*, \dots, y_m^*\}. \quad (16)$$

In the following, we prove that the equilibrium to dynamics (10)–(11) exists, by showing that $\mathbf{x}^*$ and $\mathbf{y}^*$ satisfy (12)–(13). First, since $\ker \bar{L} = \text{image}(\mathbf{1} \otimes I_n)$ and $\mathbf{x}^* = \mathbf{1}_m \otimes x^*$, one has $\bar{L}\mathbf{x}^* = 0$, which is (13).

To continue, from (15), and the definitions of $\mathbf{x}^*$ and $\mathbf{y}^*$ in (16), one has

$$\nabla f(\mathbf{x}^*) + \mathbf{y}^* = \bar{A}^\top \mathbf{z}^* \quad (17)$$

where $\bar{A} = \text{diag} \{A_1, \dots, A_m\}$. Indeed,

$$\nabla f(\mathbf{x}^*) + \bar{L}\mathbf{x}^* + \mathbf{y}^* = \bar{A}^\top \mathbf{z}^* \quad (18)$$

From this, since $\bar{P}\bar{A}^\top = 0$, we obtain

$$0 = -\bar{P} (\nabla f(\mathbf{x}^*) + \bar{L}\mathbf{x}^* + \mathbf{y}^*). \quad (19)$$

This ensures the satisfaction of (12). Furthermore, since $\mathbf{y}(0) = 0$, the integration of $\mathbf{y}$ in (11) leads to another implicit condition, that is for all

⁶Since $\mathbb{G}$ is connected and undirected, L must be symmetric with kernel spanned by $\mathbf{1}_m$, the m -vector of all 1's.

$t, \mathbf{y}(t) \in \text{image } \bar{L}$. Hence, $\mathbf{y}^* \in \text{image } \bar{L}$. To validate this, from (14) and (15), one has

$$\begin{aligned} \sum_{i=1}^m \mathbf{y}_i^* &= \sum_{i=1}^m [A_i^\top z_i^* - \nabla f_i(x^*)] \\ &= - \left(\sum_{i=1}^m \nabla f_i(x^*) - \sum_{i=1}^m A_i^\top z_i^* \right) = 0. \end{aligned}$$

This ensures the satisfaction of $\mathbf{y}^* \in \text{image } \bar{L}$ and establishes the existence of the equilibrium points $(\mathbf{x}^*, \mathbf{y}^*)$.

Consistency: Based on the existence of the equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$, we now show that any $\mathbf{x}^*$ satisfying (12) and (13) takes the form set out in the lemma statement and is consistent with the solution to problem (1)–(3). Since $\ker \bar{L} = \text{image}(\mathbf{1} \otimes I_n)$ and $\bar{L}\mathbf{x}^* = 0$, one has $\mathbf{x}^* = \mathbf{1}_m \otimes u$, where $u \in \mathbb{R}^n$. This directly leads to the consensus property. To continue, we only need to show that $u = x^*$ is a solution to problem (1)–(3). To do this, recall that $\text{image } P_i = \ker A_i$, thus, $\ker P_i = \text{image } A_i^\top$. From (12) and $\bar{P} = \text{diag} \{P_1, \dots, P_m\}$, there exists a $\mathbf{z}^* = \text{col} \{z_1^*, \dots, z_m^*\}$, $z_i^* \in \mathbb{R}^{n_i}$ such that

$$\nabla f(\mathbf{x}^*) + \bar{L}\mathbf{x}^* + \mathbf{y}^* = \bar{A}^\top \mathbf{z}^*. \quad (20)$$

Further recall that $\nabla f(\mathbf{x}^*) = \nabla f(\mathbf{1}_m \otimes u)$ is the column stack of the vectors $\left. \frac{df_i(x_i)}{dx_i} \right|_u$, then multiplying (20) on the left by $(\mathbf{1}_m \otimes I_n)^\top$ yields

$$(\mathbf{1}_m \otimes I_n)^\top \nabla f(\mathbf{x}^*) = \sum_{i=1}^m \nabla f_i(u) = \sum_{i=1}^m A_i^\top z_i^*. \quad (21)$$

This, by standard Lagrange multiplier theory, tells us u is a critical point for $F(x) = \sum_{i=1}^m f_i(x)$ on the manifold defined by $A_i x = b_i$ for all i . Thus, $u = x^*$ is a minimizer to problem (1)–(3). This completes the proof. ■

As a side remark, we observe that under Assumption 1, a solution x^* can be nonunique, but under Assumption 2, x^* is clearly unique. Furthermore, note that the uniqueness, or otherwise, of y_i^* does not influence the result of Lemma 1, as it does not originally appear in update (7). Actually in (14), if $[A_1^\top, \dots, A_m^\top]$ does not have linearly independent columns, the value of z_i^* is nonunique [39]. Consequently, the value of y_i^* is nonunique.

B. Change of Coordinate Frame

In order to examine the transient behavior of (10)–(11), it is convenient to make a coordinate transformation which ensures that in the new coordinates, the equilibrium point corresponding to $\mathbf{x}^*$ is moved to zero. Here, we change the origin of updates (10)–(11), by defining vectors $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ as

$$\begin{aligned} \tilde{\mathbf{x}} &= \mathbf{x} - \mathbf{x}^* \\ \tilde{\mathbf{y}} &= \mathbf{y} - \mathbf{y}^* \end{aligned} \quad (22)$$

where, as above, $\mathbf{x}^* = \mathbf{1}_m \otimes x^*$ and $\mathbf{y}^*$ satisfies (19) with a certain Lagrange multiplier $\mathbf{z}^*$. Note that when $\mathbf{x}^*$ and $\mathbf{y}^*$ are nonunique, one can make an arbitrary choice consistent with (12)–(13). Evidently,

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= -\bar{P} (\nabla f(\tilde{\mathbf{x}} + \mathbf{x}^*) - \bar{L}(\tilde{\mathbf{x}} + \mathbf{x}^*) - (\tilde{\mathbf{y}} + \mathbf{y}^*)) \\ &= -\bar{P} ([\nabla f(\tilde{\mathbf{x}} + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] - \bar{L}\tilde{\mathbf{x}} - \tilde{\mathbf{y}}) \end{aligned} \quad (23)$$

$$\dot{\tilde{\mathbf{y}}} = \bar{L}(\tilde{\mathbf{x}} + \mathbf{x}^*) = \bar{L}\tilde{\mathbf{x}}. \quad (24)$$

To continue, we further modify updates (23) and (24) by a frame transformation. Since the linear equation set $A_i x = b_i$, $i = 1, \dots, m$ has multiple solutions, there exists at least one nonzero vector in the

kernel of every A_i , i.e., in the range of every P_i . Observe that if u is such a vector, then $P_i u = u$, $\forall i \in \{1, \dots, m\}$. Recalling that $\ker \bar{L} = \text{image}(\mathbf{1} \otimes I_n)$, then, $\bar{P}\bar{L}\bar{P}(\mathbf{1} \otimes u) = 0$, which means $\bar{P}\bar{L}\bar{P}$ is singular. Thus, there exists an orthogonal matrix $Q = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$, with $R_1 \in \mathbb{R}^{mn \times \bar{n}_1}$, $R_2 \in \mathbb{R}^{mn \times \bar{n}_2}$, $\bar{n}_1 + \bar{n}_2 = mn$, such that

$$Q^\top \bar{P}\bar{L}\bar{P}Q = \begin{bmatrix} 0 & 0 \\ 0 & R_2^\top \bar{P}\bar{L}\bar{P}R_2 \end{bmatrix} \quad (25)$$

where the matrix $R_2^\top \bar{P}\bar{L}\bar{P}R_2$ is nonsingular. Now define new vectors X, Y by the transformations

$$X = Q^\top \tilde{x}, \quad Y = Q^\top \tilde{y}. \quad (26)$$

Multiplying the differential (23) and (24) on the left, by $Q^\top$ and $Q^\top \bar{P}$, respectively, yields

$$\dot{X} = -Q^\top \bar{P}[\nabla f(QX + x^*) - \nabla f(x^*)] - Q^\top \bar{P}\bar{L}\bar{P}QX - Y \quad (27)$$

$$\dot{Y} = Q^\top \bar{P}\bar{L}\bar{P}QX. \quad (28)$$

Note that in the derivation of (27)–(28), we have replaced QX with $\bar{P}QX$. This equality holds because both $x_i(t)$ for all t and x^* are solutions to $A_i x = b_i$; then, $P_i(x_1 - x^*) = (x_1 - x^*)$, that is, $\bar{P}QX = \bar{P}\tilde{x} = \text{col}\{P_1(x_1 - x^*), \dots, P_m(x_m - x^*)\} = \tilde{x} = QX$. Based on (27)–(28), further partition the vectors X, Y as

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \quad (29)$$

where $X_1, Y_1 \in \mathbb{R}^{\bar{n}_1}$ and $X_2, Y_2 \in \mathbb{R}^{\bar{n}_2}$. Consider now the equations for X_1, Y_1 alone. Using (25), there results

$$\dot{X}_1 = -R_1^\top \bar{P}[\nabla f(\tilde{x} + x^*) - \nabla f(x^*)] - Y_1 \quad (30)$$

$$\dot{Y}_1 = 0 \quad (31)$$

and

$$\dot{X}_2 = -R_2^\top \bar{P}[\nabla f(QX + x^*) - \nabla f(x^*)] - R_2^\top \bar{P}\bar{L}\bar{P}R_2 X_2 - Y_2 \quad (32)$$

$$\dot{Y}_2 = R_2^\top \bar{P}\bar{L}\bar{P}R_2 X_2. \quad (33)$$

Now observe that $Y_1(0) = 0$. The argument is as follows. Because $Q = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$, from (25), one has $R_1^\top \bar{P}\bar{L} = 0$. Recall that $y(0) = 0$ and $y^* \in \text{image } \bar{L}$, then $Y_1(0) = R_1^\top \bar{P}\tilde{y}(0) = R_1^\top \bar{P}(y(0) - y^*) = 0$. In light of (31), this means that $Y_1 = 0$ for all t ; furthermore, $X^\top Y = X_1^\top Y_1 + X_2^\top Y_2 = X_2^\top Y_2$.

Stability questions concerning (23) and (24) thus can be treated as stability questions concerning (30)–(33), leaving out (31).

C. Proof of Theorem 1

The proof comprises two main steps. In the first step, under Assumption 1, Lyapunov theory is used to establish asymptotic stability of (30)–(33). In the second step, the equilibrium is shown to be globally exponentially stable, provided Assumption 2 holds.

Asymptotic Stability: Noting that $R_2^\top \bar{P}\bar{L}\bar{P}R_2$ is symmetric positive definite, we can define a positive definite function V of X and Y_2 as follows:

$$V(X, Y_2) = \frac{1}{2} [X^\top X + Y_2^\top (R_2^\top \bar{P}\bar{L}\bar{P}R_2)^{-1} Y_2]. \quad (34)$$

Computing the derivative along motions of (27)–(28) gives us

$$\begin{aligned} \dot{V} &= -X^\top Q^\top \bar{P}[\nabla f(QX + x^*) - \nabla f(x^*)] - X^\top Q^\top \bar{P}\bar{L}\bar{P}QX \\ &\quad - X^\top Y + Y_2^\top (R_2^\top \bar{P}\bar{L}\bar{P}R_2)^{-1} (R_2^\top \bar{P}\bar{L}\bar{P}R_2) X_2 \\ &= -X^\top Q^\top \bar{P}[\nabla f(QX + x^*) - \nabla f(x^*)] - X^\top Q^\top \bar{P}\bar{L}\bar{P}QX \\ &= -X^\top Q^\top [\nabla f(QX + x^*) - \nabla f(x^*)] - X^\top Q^\top \bar{L}QX. \end{aligned} \quad (35)$$

Note that the last equality holds because $QX = \bar{P}QX$ and $X^\top Y = X_1^\top Y_1 + X_2^\top Y_2 = X_2^\top Y_2$.

Since $\nabla f(x) = \text{col}\{\nabla f_1(x_1), \dots, \nabla f_m(x_m)\}$ and each $f_i(\cdot)$ is convex, one has $f(\cdot)$ is also convex, that is $-X^\top Q^\top [\nabla f(QX + x^*) - \nabla f(x^*)] \leq 0$. Thus, $\dot{V}$ in (35) is nonpositive. By applying LaSalle's theorem [40], we know the system converges to $\dot{V} = 0$, i.e., $\bar{L}QX = X^\top Q^\top [\nabla f(QX + x^*) - \nabla f(x^*)] = 0$. Because $\ker \bar{L} = \text{image}(\mathbf{1}_m \otimes I_n)$, one has $QX = \mathbf{1}_m \otimes q$, $q \in \mathbb{R}^n$. Consequently,

$$\begin{aligned} &X^\top Q^\top [\nabla f(QX + x^*) - \nabla f(x^*)] \\ &= [\mathbf{1}_m \otimes q]^\top [\nabla f(\mathbf{1}_m \otimes (q + x^*)) - \nabla f(\mathbf{1}_m \otimes x^*)] \\ &= q^\top \sum_{i=1}^m [\nabla f_i(q + x^*) - \nabla f_i(x^*)] \\ &= q^\top [\nabla F(q + x^*) - \nabla F(x^*)] = 0. \end{aligned} \quad (36)$$

The last equality holds because $F(x) = \sum_{i=1}^m f_i(x)$. Recall that $x - x^* = QX = \mathbf{1}_m \otimes q$ and $y - y^* = \bar{L}QX = 0$; thus, the states of all agents converge asymptotically to a same vector $x^* + q$, which satisfies the consensus property (3). To complete the proof, given x^* is a minimizer to (1)–(3), we only need to show $x^* + q$ is also a minimizer. To do this, because $\bar{P}QX = QX$, then $q \in \cap_{i=1}^m \ker(A_i)$. Clearly $x_i = x^* + q$ satisfies constraint (2). Furthermore, since x^* is a minimizer, there holds $q^\top \sum_{i=1}^m \nabla f_i(x^*) = q^\top \nabla F(x^*) = 0$. This, together with (36) yields $q^\top \nabla F(q + x^*) = 0$. Finally, due to the convexity of $F(x)$, one has $F(x^*) \geq F(x^* + q) + (-q)^\top \nabla F(q + x^*) = F(x^* + q)$. Thus, $F(x^* + q) = F(x^*)$ and $x^* + q$ must be a minimizer to problem (1)–(3). This establishes the asymptotic convergence of the algorithm.

Exponential Stability: Here, we establish the exponential convergence rate of the proposed update under Assumption 2. Note that in this case, x^* must be unique. Define

$$\Theta_1 = \bar{P}Q \begin{bmatrix} X_1 \\ 0_{\bar{n}_2} \end{bmatrix} \quad \text{and} \quad \Theta_2 = \bar{P}Q \begin{bmatrix} 0_{\bar{n}_1} \\ X_2 \end{bmatrix} \quad (37)$$

such that $\Theta_1, \Theta_2 \in \mathbb{R}^{mn}$ and $\Theta_1 + \Theta_2 = \bar{P}Q \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \bar{P}QX$. To continue, recall the second last line of (35) and the fact that $QX = \bar{P}QX$, then $\Theta_1 + \Theta_2 = QX$ and

$$\begin{aligned} \dot{V} &= -X^\top Q^\top \bar{P}[\nabla f(\bar{P}QX + x^*) - \nabla f(x^*)] - X^\top Q^\top \bar{P}\bar{L}\bar{P}QX \\ &= -(\Theta_1 + \Theta_2)^\top [\nabla f(\Theta_1 + \Theta_2 + x^*) - \nabla f(x^*)] \\ &\quad - X^\top Q^\top \bar{P}\bar{L}\bar{P}QX. \end{aligned} \quad (38)$$

Since $f(\cdot)$ is convex, $-(\Theta_1 + \Theta_2)^\top [\nabla f(\Theta_1 + \Theta_2 + x^*) - \nabla f(x^*)] \leq 0$. Then, given any scalar $0 < \gamma < 1$, there holds

$$\begin{aligned} \dot{V} &\leq -\gamma(\Theta_1 + \Theta_2)^\top [\nabla f(\Theta_1 + \Theta_2 + x^*) - \nabla f(x^*)] \\ &\quad - X^\top Q^\top \bar{P}\bar{L}\bar{P}QX \end{aligned}$$

$$\begin{aligned}
&= -\gamma\Theta_1^\top [\nabla f(\Theta_1 + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] \\
&\quad -\gamma\Theta_1^\top [\nabla f(\Theta_1 + \Theta_2 + \mathbf{x}^*) - \nabla f(\Theta_1 + \mathbf{x}^*)] \\
&\quad -\gamma\Theta_2^\top [\nabla f(\Theta_1 + \Theta_2 + \mathbf{x}^*) - \nabla f(\Theta_2 + \mathbf{x}^*)] \\
&\quad -\gamma\Theta_2^\top [\nabla f(\Theta_2 + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] \\
&\quad -X^\top Q^\top \bar{P} \bar{L} \bar{P} Q X.
\end{aligned} \tag{39}$$

Now, from definition (37) and the upper left 0 entry of (35), it can be observed that $\bar{L} \bar{P} Q X = \bar{L} \bar{P} Q \begin{bmatrix} 0_{\bar{n}_1} \\ X_2 \end{bmatrix} = \bar{L} \Theta_2$. Consequently, one has $\bar{L} \Theta_1 = \bar{L} \bar{P} Q X - \bar{L} \Theta_2 = 0$. Because $\ker \bar{L} = \text{image}(\mathbf{1}_m \otimes I_n)$, one has $\Theta_1 = \mathbf{1}_m \otimes q_2$, $q_2 \in \mathbb{R}^n$. Recall also that $F(x) = \sum_{i=1}^m f_i(x)$ is strongly convex at $\mathbf{x}^*$; thus, there exists a positive ω such that for any $q_2 \in \mathbb{R}^n$,

$$\begin{aligned}
&\sum_{i=1}^m [q_2^\top (\nabla f_i(q_2 + \mathbf{x}^*) - \nabla f_i(\mathbf{x}^*))] \\
&= (\mathbf{1}_m \otimes q_2)^\top (\nabla F(q_2 + \mathbf{x}^*) - \nabla F(\mathbf{x}^*)) \geq \omega \|q_2\|_2^2.
\end{aligned} \tag{40}$$

Thus, for $\Theta_1 = \mathbf{1}_m \otimes q_2$, $q_2 \neq 0$,

$$\begin{aligned}
\Theta_1^\top [\nabla f(\Theta_1 + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] &= \sum_{i=1}^m [q_2^\top (\nabla f_i(q_2 + \mathbf{x}^*) - \nabla f_i(\mathbf{x}^*))] \\
&\geq \omega \|q_2\|_2^2 = \frac{\omega}{m} \|\Theta_1\|_2^2.
\end{aligned} \tag{41}$$

In addition, from (35) and the fact that $R_2^\top \bar{P} \bar{L} \bar{P} R_2$ is positive definite, there exists a positive β_0 such that

$$X^\top Q^\top \bar{P} \bar{L} \bar{P} Q X = X_2^\top R_2^\top \bar{P} \bar{L} \bar{P} R_2 X_2 \geq \beta_0 \|X_2\|_2^2.$$

From definition (37) and the fact that the 2-matrix norms of $\bar{P}$, Q are no greater than 1, there holds

$$X^\top Q^\top \bar{P} \bar{L} \bar{P} Q X \geq \beta_0 \|\Theta_2\|_2^2. \tag{42}$$

To continue, recall that all $f_i(\cdot)$ are convex, then, they are also locally Lipschitz continuous [39]. From the convexity, one has

$$\Theta_2^\top [\nabla f(\Theta_2 + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] \geq 0. \tag{43}$$

From the locally Lipschitz continuity and the fact that X is bounded (due to its asymptotic convergence), then there must exist a certain constant $\ell_p > 0$ such that

$$\|\nabla f(\xi_1) - \nabla f(\xi_2)\| \leq \ell_p \|\xi_1 - \xi_2\|. \tag{44}$$

From (44), by setting $\xi_1 = \Theta_1 + \Theta_2 + \mathbf{x}^*$, $\xi_2 = \Theta_1 + \mathbf{x}^*$ and $\xi_1 = \Theta_1 + \Theta_2 + \mathbf{x}^*$, $\xi_2 = \Theta_2 + \mathbf{x}^*$, one has

$$\begin{aligned}
-\gamma\Theta_1^\top [\nabla f(\Theta_1 + \Theta_2 + \mathbf{x}^*) - \nabla f(\Theta_1 + \mathbf{x}^*)] &\leq \gamma\ell_p \|\Theta_1\| \|\Theta_2\| \\
-\gamma\Theta_2^\top [\nabla f(\Theta_1 + \Theta_2 + \mathbf{x}^*) - \nabla f(\Theta_2 + \mathbf{x}^*)] &\leq \gamma\ell_p \|\Theta_1\| \|\Theta_2\|.
\end{aligned}$$

Bringing the above equations and (41)–(43) into (39) yields

$$\begin{aligned}
\dot{V} &\leq -\frac{\gamma\omega}{m} \|\Theta_1\|_2^2 + 2\gamma\ell_p \|\Theta_1\| \|\Theta_2\| - \beta_0 \|\Theta_2\|_2^2 \\
&= -\left(\sqrt{\frac{\gamma\omega}{2m}} \|\Theta_1\|_2 - \sqrt{\frac{2\gamma m \ell_p^2}{\omega}} \|\Theta_2\|_2 \right)^2 \\
&\quad -\frac{\gamma\omega}{2m} \|\Theta_1\|_2^2 - \left(\beta_0 - \frac{2\gamma m \ell_p^2}{\omega} \right) \|\Theta_2\|_2^2 \\
&\leq -\frac{\gamma\omega}{2m} \|\Theta_1\|_2^2 - \beta_1 \|\Theta_2\|_2^2
\end{aligned} \tag{45}$$

where $\beta_1 = \beta_0 - \frac{2\gamma m \ell_p^2}{\omega}$. Note that β_0, ℓ_p, ω are positive constants and, as in (39), γ can be selected arbitrarily from $(0,1)$; hence, by choosing sufficiently small γ , the constant β_1 can be made strictly positive. Consider the triangle inequality

$$\|\Theta_1\|_2^2 + \|\Theta_2\|_2^2 \geq \frac{(\|\Theta_1\|_2 + \|\Theta_2\|_2)^2}{2} \geq \frac{\|\Theta_1 + \Theta_2\|_2^2}{2}$$

and recall the fact that $\Theta_1 + \Theta_2 = QX$, by choosing $\beta_2 = \min(\frac{\gamma\omega}{2m}, \beta_1)$, there holds

$$\dot{V} \leq -\frac{\beta_2 \|\Theta_1 + \Theta_2\|_2^2}{2} = -\frac{\beta_2 \|QX\|_2^2}{2} = -\frac{\beta_2 \|X\|_2^2}{2}. \tag{46}$$

To continue, define the following function: $V_2(X, Y) = \frac{1}{2}(X + Y)^\top (X + Y)$. Then,

$$\begin{aligned}
\dot{V}_2 &= (X + Y)^\top (\dot{X} + \dot{Y}) \\
&= -(X + Y)^\top [Q^\top \bar{P} [\nabla f(QX + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] + Y] \\
&= -X^\top Q^\top [\nabla f(QX + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] - X^\top Y \\
&\quad - [\nabla f(QX + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)]^\top Y - Y^\top Y \\
&= -X^\top Q^\top [\nabla f(QX + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] - \frac{1}{2} Y^\top Y \\
&\quad - \left\| X + \frac{1}{2} Y \right\|_2^2 - \left\| [\nabla f(QX + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)] + \frac{1}{2} Y \right\|_2^2 \\
&\quad + \|X\|_2^2 + \|\nabla f(QX + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)\|_2^2 \\
&\leq -\frac{1}{2} Y^\top Y + \|X\|_2^2 + \|\nabla f(QX + \mathbf{x}^*) - \nabla f(\mathbf{x}^*)\|_2^2 \\
&\leq -\frac{1}{2} \|Y\|_2^2 + (1 + \ell_p^2) \|X\|_2^2.
\end{aligned} \tag{47}$$

The last inequality holds because of the inequality associated with Lipschitz continuity given in (44). Now consider

$$\tilde{V} = \frac{2(2 + \ell_p^2)}{\beta_2} V + V_2. \tag{48}$$

From (46) and (47), there holds

$$\dot{\tilde{V}} \leq -\|X\|_2^2 - \frac{1}{2} \|Y\|_2^2. \tag{49}$$

By the definitions of $\tilde{V}$, V , and V_2 , obviously, there exists a positive constant β such that $\|X\|_2^2 + \frac{1}{2} \|Y\|_2^2 \geq \beta \tilde{V}$. Therefore,

$$\dot{\tilde{V}} \leq -\beta \tilde{V}. \tag{50}$$

This established the global exponential convergence of the system and completes the proof the theorem. ■

D. Proof of Corollary 1

Consider the proof of Theorem 1 with replacement of (7) by (8), and define $\mathbf{v} = \text{col}\{v_1, \dots, v_m\}$. From the definition of $\tilde{V}$, and the obtained $\dot{\tilde{V}}$ in (50), there must exist positive constants η_1, η_2, η , such that

$$\begin{aligned}
\dot{\tilde{V}} &\leq -\beta \tilde{V} + \eta_1 \|\mathbf{v}\|_2 \|X\|_2 + \eta_2 \|\mathbf{v}\|_2 \|Y\|_2 \\
&\leq -\beta \tilde{V} + \eta \|\mathbf{v}\|_2 \sqrt{\tilde{V}}.
\end{aligned} \tag{51}$$

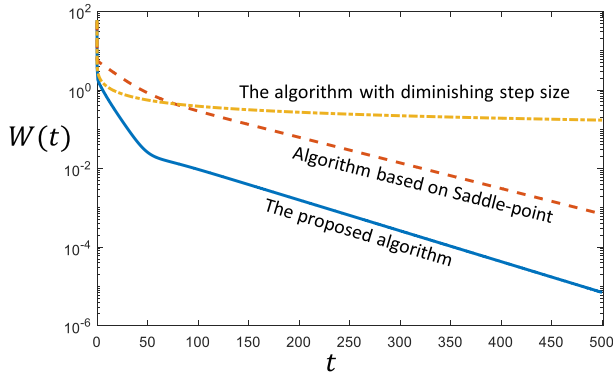


Fig. 1. 5-agent case, exponential convergence of the algorithm, with simulation execution time 0.13 s.

Further, since $\|v\|_2$ is bounded, i.e., $\|v\|_2 \leq \epsilon$ for some positive ϵ , then as $t \rightarrow \infty$, one has

$$\tilde{V} \leq \frac{\eta^2 \epsilon^2}{\beta^2} \quad (52)$$

is bounded. It follows that $\|x_i(t) - x^*\|$ is bounded for all $i = 1, \dots, m$. This completes the proof. ■

V. SIMULATION

A. Exponential Convergence Rate

First, consider an example with $m = 5$, $n = 20$, and $n_i = 3$, for $i = 1, \dots, 5$. Suppose the agents in the network have the following neighbor relations: $\mathcal{N}_1 = \{1, 2, 3, 4\}$, $\mathcal{N}_2 = \{1, 2, 3\}$, $\mathcal{N}_3 = \{1, 2, 3, 4\}$, $\mathcal{N}_4 = \{1, 3, 4, 5\}$, $\mathcal{N}_5 = \{4, 5\}$, which ensures the network is undirected and connected. We let each agent i know a local objective function $f_i(x)$ and a local constraint $A_i x = b_i$ such that

$$\begin{aligned} f_1(x) &= \|x\|_2^2 & f_2(x) &= \|x - c_2\|_2^2 \\ f_3(x) &= \sum_{k=1}^{20} e^{x[k]} & f_4(x) &= \sum_{k=1}^{20} e^{-2x[k]} \\ f_5(x) &= \|x - c_5\|_2^4 \end{aligned} \quad (53)$$

where $x[k]$ denotes the k th entry of vector x ; $A_i \in \mathbb{R}^{3 \times 20}$, $b_i \in \text{image } A_i \subset \mathbb{R}^3$, and $c_2, c_5 \in \mathbb{R}^{20}$ are constant matrices/vectors. We let the simulation configuration satisfy Assumption 2.

In order to validate Theorem 1, we let each agent initialize its local state $x_i \in \mathbb{R}^{20}$ as $A_i x_i(0) = b_i$ and then update its state by (7). Define the following function:

$$W(t) = \sum_{i=1}^5 \|x_i(t) - x^*\|_2^2 \quad (54)$$

for which $W(t) = 0$ if and only if all $x_i(t) = x^*$ for all $i = 1, \dots, 5$, where x^* is the unique minimizer of $F(x) = \sum_{i=1}^m f_i(x)$ subject to $A_i x = b_i$, $i = 1 \dots, 5$. The simulation result is obtained using the Ode45 solver of MATLAB, with a computer equipped with Intel 6700 CPU. The result is presented in Fig. 1, where the Y-axis is scaled by $\log(\cdot)$. The constant slope of the curve $W(t)$ [with $W(t)$ converging to 0] indicates the exponential convergence of the algorithm, which validates Theorem 1. In contrast, algorithms with diminishing gain ($\alpha(t) = 1/t$) only achieve an asymptotic convergence rate. Note that the continuous algorithms based on saddle-point dynamics (primal-dual) [24] can also achieve an exponential convergence rate, but as

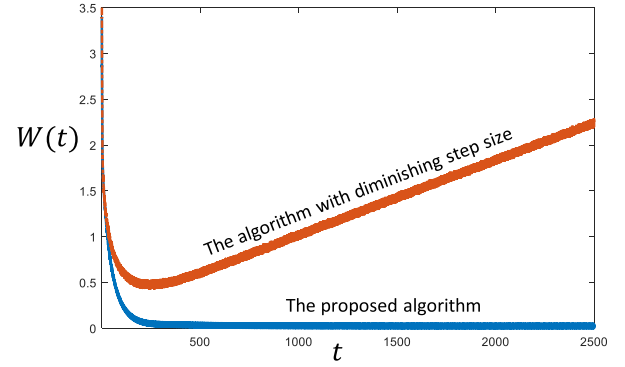


Fig. 2. 30-agent case, under bounded disturbance, with simulation execution time 1.05 s.

discussed earlier, require extra states that are transferred across the network.

B. Robustness Towards Disturbance

Here, we investigate the impact of bounded disturbance to the proposed algorithm. Consider a connected, undirected network with $m = 30$ agents. We let each agent i know a local state $x_i \in \mathbb{R}^5$, a local objective function $f_i(x_i)$, and a local constraint $A_i x = b_i$. Note that the local objective functions basically follow one of the linear, norm, or exponential forms, which are similar to the examples provided in (53). The coefficients of the objective functions are randomly chosen, and we make sure Assumptions 2 holds. In addition, for each agent, we use MATLAB Random Source tool to introduce a bounded disturbance v_i , such that each entry of v_i is chosen from $[0, 0.01]$ with sampling time 0.1 s. We compare the performance of update (8) and the update

$$\dot{x}_i = -P_i \left(\alpha(t) \nabla f_i(x_i) + \sum_{j \in \mathcal{N}_i} (x_i - x_j) \right) + v_i \quad (55)$$

equipped with diminishing gain $\alpha(t) = 1/t$ (the discrete-time versions of this update are proposed in [20], [21]).

For both algorithms, we use the same initial state and the Ode45 solver of MATLAB to perform simulation. The obtained error curves $W(t)$ are shown in Fig. 2, which is a representative example we choose from the many random simulations we performed. It can be observed that for the update with diminishing gain, the non-zero disturbance will accumulate with time t and finally lead the curve $W(t)$ to blow up. This means the agents' states are not able to converge to the optimum point x^* . For the proposed update, the curve of $W(t)$ does not grow with time t ; instead, it converges to a bounded neighborhood of x^* , which validates Corollary 1.

VI. CONCLUSION

In this article, by incorporating the idea of integral feedback, we proposed a continuous-time distributed algorithm which is able to solve a constrained distributed optimization problem with exponential convergence rate. To sum up, the proposed algorithm 1) does represent an increase in the state dimension at each agent over an algorithm with diminishing gain; 2) in comparison with other algorithms which use an increase of dimension to avoid diminishing gain, does not impose an additional burden on the communication bandwidth; 3) presents relaxed condition on objective function to guarantee exponentially fast convergence; and 4) offers robustness against disturbance. Our

future works include the generalization of the proposed algorithm to a discrete-time update, in the first instance with a combination of forward and backward Euler discretization, which is known to preserve exponential convergence [5]. We also aim to treat time-varying directed networks, and the application of general local constraints other than those expressed by a local linear equation. Building on the discrete-time update, we plan to rigorously characterize its convergence speed with respect to the number of iterations, by considering the properties of the underlying network and the local optimization functions in each agent.

REFERENCES

- [1] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
- [2] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.
- [3] T. Yang *et al.*, "A survey of distributed optimization," *Annu. Rev. Control*, vol. 47, pp. 278–305, 2019.
- [4] K. Scaman, F. Bach, S. Bubeck, L. Massoulié, and Y. T. Lee, "Optimal algorithms for non-smooth distributed optimization in networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2740–2749.
- [5] X. Wang, J. Zhou, S. Mou, and M. J. Corless, "A distributed algorithm for least squares solutions," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4217–4222, Oct. 2019.
- [6] P. Wang, S. Mou, J. Lian, and W. Ren, "Solving a system of linear equations: From centralized to distributed algorithms," *Annu. Rev. Control*, vol. 47, pp. 306–322, 2019.
- [7] K. Lu, G. Jing, and L. Wang, "A distributed algorithm for solving mixed equilibrium problems," *Automatica*, vol. 105, pp. 246–253, 2019.
- [8] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 2863–2878, Nov. 2015.
- [9] Z. Qiu, S. Liu, and L. Xie, "Distributed constrained optimal consensus of multi-agent systems," *Automatica*, vol. 68, pp. 209–215, 2016.
- [10] X. Zeng, P. Yi, and Y. Hong, "Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5227–5233, Oct. 2017.
- [11] Q. Liu and J. Wang, "A second-order multi-agent network for bound-constrained distributed optimization," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3310–3315, Dec. 2015.
- [12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [13] 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, Apr. 2017.
- [14] T.-H. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3719–3734, Jul. 2016.
- [15] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, "Distributed linearized alternating direction method of multipliers for composite convex consensus optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 5–20, Jan. 2018.
- [16] M. Cao, A. S. Morse, and B. D. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM J. Control Optim.*, vol. 47, no. 2, pp. 575–600, 2008.
- [17] X. Chen, M.-A. Belabbas, and T. Başar, "Controllability of formations over directed time-varying graphs," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 407–416, Sep. 2017.
- [18] C. Yan and H. Fang, "A new encounter between leader-follower tracking and observer-based control: Towards enhancing robustness against disturbances," *Syst. Control Lett.*, vol. 129, pp. 1–9, 2019.
- [19] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multi-Agent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
- [20] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.
- [21] P. Lin, W. Ren, and Y. Song, "Distributed multi-agent optimization subject to nonidentical constraints and communication delays," *Automatica*, vol. 65, pp. 120–131, 2016.
- [22] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.
- [23] G. Shi, B. D. O. Anderson, and U. Helmke, "Network flows that solve linear equations," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2659–2674, Jun. 2017.
- [24] B. Ghareisifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 781–786, Mar. 2014.
- [25] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation," *SIAM J. Optimiz.*, vol. 32, no. 2, pp. 354–385, 2022.
- [26] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6013–6023, Nov. 2015.
- [27] S. Alghunaim, K. Yuan, and A. H. Sayed, "A linearly convergent proximal gradient algorithm for decentralized optimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 2848–2858.
- [28] S. A. Alghunaim, E. Ryu, K. Yuan, and A. H. Sayed, "Decentralized proximal gradient algorithms with linear convergence rates," *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2787–2794, Jun. 2021.
- [29] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.
- [30] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4494–4506, Sep. 2019.
- [31] D. Jakovetić, D. Bajović, J. Xavier, and J. M. F. Moura, "Primal-dual methods for large-scale and distributed convex optimization and data analytics," *Proc. IEEE*, vol. 108, no. 11, pp. 1923–1938, Nov. 2020.
- [32] H. Li, Z. Lin, and Y. Fang, "Variance reduced EXTRA and DIGing and their optimal acceleration for strongly convex decentralized optimization," 2020, *arXiv:2009.04373*.
- [33] J. B. Rosen, "The gradient projection method for nonlinear programming. Part I. Linear constraints," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 1, pp. 181–217, 1960.
- [34] J. Luo and K. Nonami, "Approach for transforming linear constraints on Petri nets," *IEEE Trans. Autom. Control*, vol. 56, no. 12, pp. 2751–2765, Dec. 2011.
- [35] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *Proc. 43rd IEEE Conf. Decis. Control*, 2004, pp. 3998–4003.
- [36] A. Olshevsky, "Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control," *SIAM J. Control Optim.*, vol. 48, no. 22, 2017.
- [37] T. Yucelen and M. Egerstedt, "Control of multiagent systems under persistent disturbances," in *Proc. Amer. Control Conf.*, 2012, pp. 5264–5269.
- [38] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.
- [39] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, pp. 334–334, 1997.
- [40] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Pearson, 2002.