Distributed Continuous-Time Algorithms for Time-Varying Constrained Convex Optimization

Shan Sun, Jie Xu, and Wei Ren, *Fellow, IEEE*

***Abstract*—This article is devoted to the distributed continuous-time optimization problems with time-varying objective functions and time-varying constraints. Different from most studied distributed optimization problems with time-invariant objective functions and constraints, the optimal solutions in this article are time varying and form a trajectory. First, for the case where there exist only time-varying nonlinear inequality constraints, we present a distributed control algorithm that consists of a sliding-mode consensus part and a Hessian-based optimization part coupled with the log-barrier penalty functions. The algorithm can guarantee the asymptotical tracking of the optimal solution with a zero tracking error. Second, we extend the previous result to the case where there exist not only time-varying nonlinear inequality constraints but also linear equality constraints. An extended algorithm is presented, where quadratic penalty functions are introduced to account for the equality constraints and an adaptive control gain is designed to remove the restriction on knowing the upper bounds on certain information. The asymptotical convergence of the extended algorithm to the vicinity of the optimal solution is studied under suitable assumptions. The effectiveness of the proposed algorithms is illustrated in simulation. In addition, one proposed algorithm is applied to a multirobot multitarget navigation problem with experimental demonstration on a multicrazyflie platform to validate the theoretical results.**

***Index Terms*—Continuous-time optimization, distributed time-varying optimization, multirobot multitarget navigation, time-varying constraints.**

I. Introduction

A. Background

Distributed optimization algorithms allow for decomposing certain optimization problems into smaller, more manageable subproblems that can be solved in parallel. Therefore, they are widely used to solve large-scale optimization problems such as optimization of network flows [1], big-data analysis [2], design of sensor networks [3], multirobot teams [4], and resource allocation [5]. There has been significant attention on distributed convex optimization problems, where the goal is to cooperatively seek the optimal solution that minimizes the sum of private convex objective functions available to each individual agent. In this context, discrete-time distributed optimization algorithms have been studied extensively (see, e.g., [6], [7], and references therein).

There exists another body of literature on distributed continuous-time optimization algorithms (see, e.g., [8]–[15]). The distributed continuous-time optimization algorithms have applications in coordinated control of multiagent teams. For example, multiple physical robots modeled by continuous-time dynamics might need to track a team optimal trajectory. Note that most studies in the literature focus on stationary optimization problems in which both the objective functions and constraints do not explicitly depend on time. However, in many applications, the local performance objectives or engineering constraints may evolve in time, reflecting the fact that the optimal solution could be changing over time and create a trajectory (see, e.g., [16]–[19]), which makes the design and analysis much more complex. Moreover, in practical optimization problems, constraints are sometimes inevitable. In this article, we are interested in the distributed continuous-time algorithms for time-varying constrained optimization problems.

B. Related Works

There are just a few works in the literature addressing the distributed continuous-time optimization problem with time-varying objective functions [20], [21], [22], [23], [24], [25], [26]. Specifically, [20] and [21] solve the distributed continuous-time time-varying optimization problems with convex set constraints. However, [20] and [21] are limited to solve, respectively, optimization problems with quadratic objective functions and linear programming optimization problems. Moreover, both [20] and [21] can only achieve bounded tracking errors to the optimal solutions. The work [22] addresses a Nash equilibrium seeking problem for noncooperative games where the Nash equilibrium under consideration can be time varying. However, [22] does not consider state constraints in the game problems. Distributed time-varying resource allocation problems are studied in [23] and [24], where time-varying objective functions or time-varying loads are considered. However, both [23] and [24] do not consider nonlinear inequality state

Manuscript received 5 April 2021; revised 19 December 2021 and 17 April 2022; accepted 27 July 2022. Date of publication 11 August 2022; date of current version 28 June 2023. This work was supported by National Science Foundation under Grant ECCS-2129949. Recommended by Associate Editor G. Hu. *(Corresponding author: Wei Ren.)*

The authors are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: ssun029@ucr.edu; jxu150@ucr.edu; ren@ee.ucr.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2022.3198113.

Digital Object Identifier 10.1109/TAC.2022.3198113

constraints. Recently, the second-order optimization methods are proven to work well in centralized time-varying optimization problems (see, e.g., [18], [19], and [27]). However, their use in distributed settings has been prohibited as they require global information of the network to compute the inverse of the global Hessian matrix. The works [25] and [26] solve the distributed time-varying optimization problems using second-order optimization methods. However, the algorithm in [26] and the consensus-based algorithm in [25] (Section III-B) are limited to the unconstrained problem with local objective functions that have identical Hessians. While the estimator-based algorithm in [25] (Section III-C) can deal with certain objective functions with nonidentical Hessians, it relies on the distributed average tracking techniques [28] and hence poses restrictive assumptions that the time derivatives of the Hessians and the time derivatives of the gradients of the local objective functions exist and be bounded. In addition, because the estimator-based algorithm has to estimate the Hessian inverse of the global objective function, it necessitates the communication of certain virtual variables between neighbors with increased computation costs. While it is possible to convert the constrained optimization problem to an unconstrained one using penalty methods, the resulting penalized objective functions would not have identical Hessians due to the involvement of the nonuniform local constraint functions (even if the original objective functions would), and they might not satisfy the restrictive assumptions mentioned above. As a result, the algorithms in [25] and [26] cannot be applied to address the distributed time-varying constrained optimization problem (see Remark 4 for a more detailed comparison). For distributed time-varying optimization algorithms in discrete-time settings, the readers are referred to [29] and [30]. It is worth mentioning that in the literature on discrete-time time-varying optimization algorithms, all the works can only achieve bounded tracking errors, which are usually related to the sampling rate or step size. The continuous-time and discrete-time algorithms serve in different application domains. In this article, we focus on the continuous-time algorithms, which have applications especially in motion coordination.

C. Contributions

This article aims to develop distributed algorithms to solve the continuous-time optimization problems with private time-varying objective functions and private time-varying constraints. In this article, the distributed time-varying optimization problems are deformed as a consensus subproblem and a minimization subproblem on the team objective function. First, for the case where there exist only time-varying inequality constraints, we develop a sliding-mode method with a Hessian-dependent gain for all the agents to achieve consensus on the states. Meanwhile, a Hessian-based (second-order) optimization method coupled with the log-barrier penalty functions is proposed to track the local time-varying optimal solution. Although [27] and [31] also use log-barrier penalty functions to address the inequality constraints, to the best of our knowledge, our article is the first to leverage the log-barrier penalty functions to the distributed time-varying optimization problems. To implement the algorithm, each agent just needs its own state and the relative states between itself and its neighbors. When the agents' states are their positions, the algorithm can be implemented based on purely local sensing (e.g., absolute and relative positions) without the need for communicating virtual variables. The asymptotical convergence to the optimal solution is established based on nonsmooth analysis, Lyapunov theory, and convex optimization theory. To the best of our knowledge, this is the first article in the literature on distributed continuous-time optimization with time-varying inequality constraints that guarantee zero tracking errors. Furthermore, we extend the previous result with the following improvements. We add quadratic penalty functions to account for equality constraints to make the algorithm be applicable to more general problems and we present an adaptive control gain design under which the restriction on knowing the upper bounds on certain prior information is removed. And the asymptotical convergence of the extended algorithm to the vicinity of the optimal solution is studied under suitable assumptions. Both numerical simulation and real experimental results are presented to illustrate the effectiveness of the theoretical results.

Some preliminary results of this article (Section III: Distributed time-varying optimization with nonlinear inequality constraints) are presented in [32]. This article extends [32] by considering not only time-varying nonlinear inequality constraints but also linear equality constraints. In addition, an adaptive control gain is designed to remove the restriction on knowing the upper bounds on certain information. It is worthwhile to mention that additional numerical examples and experimental results are also presented in the current article.

This article is organized as follows. In Section II, we present notation, preliminaries on graph theory, and nonsmooth analysis. We present the main results on the distributed continuous-time algorithms for the time-varying constrained optimization problems in Sections III and IV. Some numerical examples and experimental results are presented in Sections V and VI, respectively. Conclusions are drawn in Section VII.

II. Preliminaries

A. Notation

Let $\mathbb{R}, \mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ denote the sets of real numbers, real vectors of dimension n, and real matrices of size $n \times m$, respectively. Let $\mathbb{R}_{>0}$ represent the set of positive real numbers. The cardinality of a set S is denoted by $|S|$. Let $\mathbf{1}_n$ (respectively $\mathbf{0}_n$) denote the vector of n ones (respectively n zeros), and I_n denote the $n \times n$ identity matrix. For a matrix $A \in \mathbb{R}^{m\times n}$, $[A]_{k\bullet} \in \mathbb{R}^{1\times n}$ is the kth row of A, and A^T (respectively A^{-1}) is the transpose (respectively inverse) of A. For a square matrix $A \in \mathbb{R}^{n\times n}$, $\bar{\lambda}_{\min}(A)$ represents a positive value that is smaller than all the eigenvalues of A. For a vector $h = [h_1, \ldots, h_n]^T \in \mathbb{R}^n$, $\mathrm{diag}(h) \in \mathbb{R}^{n\times n}$ represents the diagonal matrix with the elements in the main diagonal being the elements of h, $\|h\|_p$ represents the p-norm of the vector h, $B(h, \delta)$ represents the open ball of radius δ centered at h, and $\mathrm{sgn}(h) = [\mathrm{sgn}(h_1), \ldots, \mathrm{sgn}(h_n)]^T$, where $\mathrm{sgn}(h_i)$ denotes the signum function defined as

$$\operatorname{sgn}(h_i) = \begin{cases} -1 & \text{if } h_i < 0, \\ 0 & \text{if } h_i = 0, \\ 1 & \text{if } h_i > 0. \end{cases}$$

For vectors $h \in \mathbb{R}^n$ and $c \in \mathbb{R}^n$, $h \preceq c$ (respectively $h \prec c$) means that $h_i \le c_i$ (respectively $h_i < c_i$) for all $i \in [1, n]$. The Lebesgue measure of N is denoted by $\mu(N)$. Let $B(h, \delta)$ be the open ball of radius δ centered at h. Let $\otimes$ denote the Kronecker product and $\overline{\text{co}}$ the convex closure. Let $\nabla f(h,t)$ and $\nabla^2 f(h,t)$ denote, respectively, gradient and Hessian of function $f(h,t)$ with respect to the vector h. Let $\partial f(h,t)/\partial t$ represent the partial derivative of function $f(h,t)$ with respect to t. Let $\dot{f}(h,t)$ be the time derivative of $f(h,t)$. That is, $\dot{f}(h,t) = \nabla f(h,t)\dot{h} + \partial f(h,t)/\partial t$.

B. Graph Theory

An undirected graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, and $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix with entries a_{ij}, $i, j \in \mathcal{V}$. For an undirected graph, an edge (j, i) implies that node i and node j are able to share data with each other, and $a_{ij} = 1$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Here, $a_{ij} = a_{ji}$. Let $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ denote the set of neighbors of node i. A path is a sequence of nodes connected by edges. An undirected graph is connected if for every pair of nodes there is a path connecting them. The Laplacian matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{n \times n}$ associated with $\mathcal{A}$ is defined as $l_{ii} = \sum_{j=1, j \ne i}^{n} a_{ij}$ and $l_{ij} = -a_{ij}$, where $i \ne j$. The incidence matrix $\mathcal{D} = [d_{ij}] \in \mathbb{R}^{n \times |\mathcal{E}|}$ associated with $\mathcal{G}$ is defined as $d_{ik} = -1$ if the kth edge leaves node i, $d_{ik} = 1$ if it enters node i, and $d_{ik} = 0$ otherwise. For the incidence matrix of an undirected graph, the orientation of the edges is assigned arbitrarily. Note that for an undirected graph, $\mathcal{L}\mathbf{1}_n = \mathbf{0}_n$, $\mathcal{L}^T = \mathcal{L}$, and $\mathcal{L} = \mathcal{D}\mathcal{D}^T$.

C. Nonsmooth Analysis

In this subsection, we recall some important definitions of the nonsmooth systems that will be exploited in our main result.

Definition 1: (Filippov Solution) [33] Consider the vector differential equation

$$\dot{x} = f(x, t) \tag{1}$$

where $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ is Lebesgue measurable and locally essentially bounded. A vector function $x(\cdot)$ is called a Filippov solution of (1) on $[t_0, t_1]$, if $x(\cdot)$ is absolutely continuous on $[t_0, t_1]$ and for almost all $t \in [t_0, t_1]$, $\dot{x}(t) \in K[f](x, t)$, where $K[f](x,t) := \bigcap_{\delta > 0} \bigcap_{\mu(N)=0} \overline{\text{co}} f(B(x, \delta) - N, t)$ is the Filippov set-valued map of $f(x, t)$ and $\bigcap_{\mu(N)=0}$ denotes the intersection over all sets N of Lebesgue measure zero.

Definition 2: (Clarke's Generalized Gradient) [33] Consider a locally Lipschitz continuous function $V(x) : \mathbb{R}^d \to \mathbb{R}$, the generalized gradient of the function V at x is given by $\partial V(x) := \overline{\text{co}}\{\lim \nabla V(x_i) | x_i \to x, x_i \notin \Omega_V\}$, where Ω_V is the set of Lebesgue measure zero where the gradient of V is not defined.

Definition 3: (Chain Rule) [33] Let $x(\cdot)$ be a Filippov solution of $\dot{x} = f(x, t)$ and $V(x) : \mathbb{R}^d \to \mathbb{R}$ be a locally Lipschitz continuous function. Then for almost all t

$$\frac{d}{dt} V[x(t)] \in \dot{\tilde{V}}$$

where $\dot{\tilde{V}}$ is the set-valued Lie derivative defined as $\dot{\tilde{V}} := \bigcap_{\xi \in \partial V} \xi^T K[f]$.

III. Distributed Time-Varying Optimization With Nonlinear Inequality Constraints

Consider a network consisting of n agents. Each agent is regarded as a node in an undirected graph, and each agent can only interact with its local neighbors in the network. Suppose that each agent satisfies the following continuous-time dynamics:

$$\dot{x}_i(t) = u_i(t) \tag{2}$$

where $x_i(t) \in \mathbb{R}^m$ is the state of agent i, and $u_i(t) \in \mathbb{R}^m$ is the control input of agent i. In this section, we study the distributed time-varying optimization problem with time-varying nonlinear inequality constraints. The goal is to design $u_i(t)$ using only local information and interaction, such that all the agents work together to find the optimal trajectory $y^*(t) \in \mathbb{R}^m$ which is defined as

$$\begin{aligned} y^*(t) = \text{argmin} \quad & \sum_{i=1}^{n} f_i[y(t), t] \\ \text{s.t.} \quad & g_i[y(t), t] \preceq \mathbf{0}_{q_i}, \ i \in \mathcal{V} \end{aligned} \tag{3}$$

where $f_i[y(t), t] : \mathbb{R}^m \times \mathbb{R}_{>0} \to \mathbb{R}$ are the local objective functions, and $g_i[y(t), t] : \mathbb{R}^m \times \mathbb{R}_{>0} \to \mathbb{R}^{q_i}$ are the local inequality constraint functions. It is assumed that $f_i[y(t), t]$ and $g_i[y(t), t]$ are known only to agent i. We assume that the minimizer $y^*(t)$ is unique for each t (see Assumption 2).

If the underlying network is connected, the above problem (3) is equivalent to the problem that all the agents reach consensus while optimizing the team objective function $\sum_{i=1}^{n} f_i[x_i(t), t]$ under constraints, more formally,

$$\begin{aligned} x^*(t) \in \mathbb{R}^{m*n} = \text{argmin} \quad & \sum_{i=1}^{n} f_i[x_i(t), t] \\ \text{s.t.} \quad & g_i[x_i(t), t] \preceq \mathbf{0}_{q_i}, \quad x_i(t) = x_j(t), \quad \forall i, j \in \mathcal{V} \end{aligned} \tag{4}$$

where $x(t) \in \mathbb{R}^{m*n}$ is the stack of all the agents' states. Here, the goal is that each state $x_i(t)$, $\forall i \in \mathcal{V}$, converges to the optimal solution $y^*(t)$, i.e.,

$$\lim_{t \to \infty} [x_i(t) - y^*(t)] = \mathbf{0}_m. \tag{5}$$

Remark 1: (Examples of applications) This architecture of the distributed time-varying constrained optimization problem (3) with networked agents finds broad applications in distributed cooperative control problems, including multirobot navigation [16], [17] and resource allocation of power network [19]. For example, in a motion coordination case, knowing only their own and their neighbors' positions, multiple unmanned aerial vehicles (UAVs) might need to dock at a moving location without collision such that the total team performance is optimized. Here, the constraints can denote that the UAVs need to be located in safe areas.

For notational simplicity, we will remove the time index t from the variables $x_i(t)$ and $u_i(t)$ in most remaining parts of this article and only keep it in some places when necessary.

Lemma 1: [34] Let $f(r) : R^m \to R$ be a continuously differentiable convex function with respect to r. The function $f(r)$ is minimized at r^* if and only if $\nabla f(r^*) = 0$.

We make the following assumptions which are all standard in the literature and are used in recent works like [25], [26], and [27].

Assumption 1: (Graph connectivity) The graph $\mathcal{G}$ is fixed, undirected, and connected.

Assumption 2: (Convexity) Assume the following.

1) All the objective functions $f_i(x_i, t)$ and the inequality constraint functions $g_i(x_i, t)$ are twice continuously differentiable with respect to x_i and continuously differentiable with respect to t.
2) All the objective functions $f_i(x_i, t)$ are uniformly strongly convex in x_i, for all $t \geq 0$.
3) All the constraint functions $g_i(x_i, t)$ are uniformly convex in x_i, for all $t \geq 0$.
4) The optimal solution exists and such that is unique.

Assumption 3: (Slater's condition) For all $t \geq 0$, there exists at least one y such that $g_i(y, t) \prec \mathbf{0}_{q_i}$ for all $i \in \mathcal{V}$. Therefore, the Slater's condition holds for all time.

By Assumption 3, the interior of the feasible region is nonempty for all $t \geq 0$ and the optimal solution $y^*(t)$ in (3) at each $t \geq 0$ can be characterized using the Karush–Kuhn–Tucker (KKT) conditions.

A. Distributed Algorithm Design

In this subsection, we derive our distributed control algorithm for the time-varying constrained optimization problem in (4).

We design the following controller for agent i:

$$u_i = -\beta\left[\nabla^2 \tilde{L}_i(x_i, t)\right]^{-1} \sum_{j\in\mathcal{N}_i} \operatorname{sgn}(x_i - x_j) + \phi_i(t)$$
$$\phi_i(t) = -\left[\nabla^2 \tilde{L}_i(x_i, t)\right]^{-1}\left[\nabla \tilde{L}_i(x_i, t) + \frac{\partial}{\partial t}\nabla \tilde{L}_i(x_i, t)\right] \tag{6}$$

where $\beta \in \mathbb{R}_{>0}$ is a fixed control gain, and $\tilde{L}_i(x_i, t)$ is a penalized objective function of agent i, defined as follows:

$$\tilde{L}_i(x_i, t) = f_i(x_i, t) - \frac{1}{\rho_i(t)}\sum_{j=1}^{q_i} \log[\sigma_i(t) - g_{ij}(x_i, t)] \tag{7}$$

where $g_{ij}(x_i, t) : \mathbb{R}^m \times \mathbb{R}_{>0} \to \mathbb{R}$ denotes the jth component of function $g_i(x_i, t)$, $\rho_i(t) \in \mathbb{R}_{>0}$ is time-varying barrier parameter, and $\sigma_i(t) \in \mathbb{R}_{>0}$ is a time-varying slack function satisfying

$$\rho_i(t) = a_{i1}e^{a_{i2}t},\ \sigma_i(t) = a_{i3}e^{-a_{i4}t},\ a_{i1}, a_{i2}, a_{i3}, a_{i4} \in \mathbb{R}_{>0}. \tag{8}$$

Note that the domain of the penalized objective function $\tilde{L}_i(x_i, t)$ is $D_i = \{x_i \in \mathbb{R}^m \mid g_i(x_i, t) \prec \sigma_i(t)\mathbf{1}_{q_i}\}$. This would require that the dynamical system (2) with controller (6) is initialized at a point inside $D_i(0)$, i.e., $x_i(0) \in D_i(0)$. It is worthwhile to mention that the introduction of $\sigma_i(t)$ is to enlarge the initial feasible set. To make the algorithm (6) work, the initial states $x_i(0)$ need to satisfy

$$g_{ij}[x_i(0), 0] < \sigma_i(0), \qquad \forall i \in \mathcal{V},\ j = 1, \ldots, q_i. \tag{9}$$

We will prove that the dynamical system (2) is well defined under controller (6), initial condition (9), and certain other assumptions (see Lemma 2).

Remark 2 (Roles of terms in (6)): In this article, the time-varying optimization problem (4) is deformed as a consensus subproblem and a minimization subproblem on the team objective function. We develop a distributed sliding-mode control law to address the consensus part. That is, the role of term $-\beta\left[\nabla^2 \tilde{L}_i(x_i, t)\right]^{-1}\sum_{j\in\mathcal{N}_i}\operatorname{sgn}(x_i - x_j)$ in (6) is to drive all the agents to reach a consensus on states $\left(\lim_{t\to\infty}\|x_i(t) - \frac{1}{n}\sum_{j=1}^n x_j(t)\|_2 = 0\right)$. Here, the Hessian-dependent gain $\beta[\nabla^2 \tilde{L}_i(x_i, t)]^{-1}$ is introduced to guarantee the convergence of our algorithm under nonidentical $\nabla^2 \tilde{L}_i(x_i, t)$. While the second term, $\phi_i(t) \in \mathbb{R}^m$, is an auxiliary variable playing a role in minimizing the penalized objective function $\tilde{L}_i(x_i, t)$ given by (7). Note that we use the log-barrier penalty functions [see the second term in (7)] to incorporate the inequality constraints into the penalized objective function. As shown in (6), we use the second-order/Hessian information of the penalized objective function to achieve the optimization goal.

Remark 3 (Use of log-barrier function in (7)): In this article, we convert the considered constrained optimization problem into an unconstrained one using the penalty functions. Multiple penalty functions might be useful to address the inequality constraints, for example, $\{\max[0, g_{ij}(x_i, t)]\}^2$ and the log-barrier function used in (7). In this article, we aim to leverage the Hessian information to solve the time-varying optimization problem. Therefore, we need a smooth and differentiable penalty function. That is why we choose log-barrier penalty functions to address the inequality constraints. While log-barrier penalty functions are not novel in its use for optimization problems with inequality constraints [27], [31], [34], to the best of our knowledge, our article is the first to leverage its use to the distributed time-varying optimization settings.

In addition, we have

$$\nabla \tilde{L}_i(x_i, t) = \nabla f_i(x_i, t) + \sum_{j=1}^{q_i}\frac{\nabla g_{ij}(x_i, t)}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]} \tag{10}$$

$$\begin{aligned}\frac{\partial}{\partial t}\nabla \tilde{L}_i(x_i, t) = {}& \frac{\partial}{\partial t}\nabla f_i(x_i, t) + \sum_{j=1}^{q_i}\frac{\partial \nabla g_{ij}(x_i, t)/\partial t}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]}\\ &- \sum_{j=1}^{q_i}\frac{\dot{\rho}_i(t)\nabla g_{ij}(x_i, t)}{\rho_i^2(t)[\sigma_i(t) - g_{ij}(x_i, t)]},\\ &- \sum_{j=1}^{q_i}\frac{\dot{\sigma}_i(t)\nabla g_{ij}(x_i, t)}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]^2},\\ &+ \sum_{j=1}^{q_i}\frac{\nabla g_{ij}(x_i, t)\partial g_{ij}(x_i, t)/\partial t}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]^2}\end{aligned} \tag{11}$$

$$\nabla^2 \tilde{L}_i(x_i,t) = \nabla^2 f_i(x_i,t) + \sum_{j=1}^{q_i} \frac{\nabla^2 g_{ij}(x_i,t)}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i,t)]} + \sum_{j=1}^{q_i} \frac{\nabla g_{ij}(x_i,t)\nabla g_{ij}(x_i,t)^T}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i,t)]^2} \tag{12}$$

where $\frac{\partial}{\partial t}\nabla f_i(x_i,t)$, $\frac{\partial}{\partial t}\nabla g_{ij}(x_i,t)$, and $\frac{\partial}{\partial t} g_{ij}(x_i,t)$ are, respectively, the partial derivatives of $\nabla f_i(x_i,t)$, $\nabla g_{ij}(x_i,t)$, and $g_{ij}(x_i,t)$ with respect to t.

Also, for notational simplicity, we will remove the time index t from the auxiliary variable $\phi_i(t)$ in most remaining parts of this article and only keep it in some places when necessary.

Remark 4: (Heterogeneity in the Hessian and comparison with previous model) In this article, we convert the considered constrained optimization problem into an unconstrained one using the log-barrier penalty functions. It is worth noting that the proposed algorithm (6) is not a simple extension of the existing distributed time-varying unconstrained optimization algorithms in [25] and [26]. Especially, to apply the algorithm in [26] and the consensus-based algorithm in [25] (Section III-B), it is required that the Hessians of all the local objective functions be identical. In contrast, in our context with the penalized objective functions, the Hessians of them are nonuniform due to the involvement of the nonuniform local constraint functions even if the original objective functions have identical Hessians. The estimator-based algorithm in [25] (Section III-C) can deal with certain objective functions with nonidentical Hessians. However, it not only necessitates the communication of certain virtual variables between neighbors with increased computation costs but requires that the time derivatives of the Hessians and the time derivatives of the gradients of the objective functions exist and be bounded. Unfortunately, due to the complexity of the penalized objective functions in the considered constrained problem, such a requirement might be no longer guaranteed to hold, and, hence, the result therein might not be applicable to our problem. In this article, we introduce a novel algorithm with a Hessian-dependent gain to account for the complexity caused by the penalized objective functions, where only the partial derivatives of the gradients of the penalized objective functions with respect to time t are preassumed to be bounded (see Assumptions 4 and 5). Note that in [25] and [26], the partial derivatives of the gradients of the objective functions with respect to time t are also required to be bounded. In this article, we do not preassume that the Hessians and gradients of the penalized objective functions are bounded; however, we will prove that the Hessians and gradients of the penalized objective functions are bounded automatically under our proposed algorithms. The novel algorithm design, in turn, introduces new challenges in theoretical analysis, which will be addressed in the following.

Remark 5: (Relevance to robotic applications) In algorithm (6), each agent just needs its own information and the relative states between itself and its neighbors. In some robotic applications, the agents' states are their spatial positions. As a result, the relative positions can be obtained by local sensing and the communication necessity might be eliminated.

B. Convergence Analysis

In this subsection, the asymptotical convergence of the system (2) under the controller (6) to the optimal solution in (3) is established. To establish our results, we require the following assumptions.

Assumption 4: (Bounds about objective functions) If all local states x_i are bounded, then there exists a constant $\bar{\alpha}$ such that $\sup_{t\in[0,\infty)} \|\frac{\partial}{\partial t}\nabla f_i(x_i,t)\|_2 \leq \bar{\alpha}$ for all $i \in \mathcal{V}$ and $t \geq 0$.

Assumption 5: (Bounds about the inequality constraint functions) If all local states x_i are bounded, then there exist constants $\bar{\beta}$ and $\bar{\gamma}$ such that $\sup_{t\in[0,\infty)} \|\frac{\partial}{\partial t}\nabla g_{ij}(x_i,t)\|_2 \leq \bar{\beta}$ and $\sup_{t\in[0,\infty)} \|\frac{\partial}{\partial t} g_{ij}(x_i,t)\|_2 \leq \bar{\gamma}$, for all $i \in \mathcal{V}$, $j = 1,\ldots,q_i$ and $t \geq 0$.

Remark 6: (Bounds rationality analysis) In Assumption 4, we assume that all $\|\frac{\partial}{\partial t}\nabla f_i(x_i,t)\|_2$ are bounded under bounded x_i. The assumption holds for an important class of situations. For example, consider the normal quadratic objective functions $\|c_i x_i + h_i(x_i,t)\|_2^2$. As long as $\frac{\partial}{\partial t} h_i(x_i,t)$ (e.g., $\sin(t), t$) are bounded under bounded x_i, $\|\frac{\partial}{\partial t}\nabla f_i(x_i,t)\|_2$ will be bounded. In Assumption 5, we assume that all $\|\frac{\partial}{\partial t}\nabla g_{ij}(x_i,t)\|_2$ and $\|\frac{\partial}{\partial t} g_{ij}(x_i,t)\|_2$ are bounded under bounded x_i. The assumption holds for an important class of situations. The boundedness of $\|\frac{\partial}{\partial t}\nabla g_{ij}(x_i,t)\|_2$ and $\|\frac{\partial}{\partial t} g_{ij}(x_i,t)\|_2$ holds for most commonly used boundary constraint functions, e.g., $x_i \leq b(t)$ or $x_i^2 \leq b(t)$ under bounded $\dot{b}(t)$.

Remark 7: (Nonsmooth analysis) With the piecewise-differentiable signum function involved in algorithm (6), the solution should be investigated in the sense of Filippov. However, since the signum function is measurable and locally essentially bounded, the Filippov solutions of the proposed system dynamics always exist [35]. To avoid symbol redundancy, we do not use the differential inclusions in the proofs when the Lyapunov candidates are continuously differentiable due to the following reason: if the Lyapunov function candidates are continuously differentiable, the set-valued Lie derivative of them is a singleton at the discontinuous points and the proof still holds without employing the nonsmooth analysis [36].

In this article, we convert the considered constrained optimization problem into an unconstrained optimization problem using the log-barrier penalty functions. That the log-barrier penalty function involved in (7) is always well defined under our proposed algorithm is important. This is described in the next lemma.

Lemma 2: Suppose that Assumptions 2 and 3 and the initial condition (9) hold. For the system (2) under the controller (6), each $x_i(t)$ belongs to the set $D_i = \{x_i \in \mathbb{R}^m \mid g_i(x_i,t) \prec \sigma_i(t)\mathbf{1}_{q_i}\}$ for all $t \geq 0$. That is, (7) is always well defined.

Proof: Assumption 3 ensures the existence of initial condition (9). Moreover, the time derivative of $\nabla\tilde{L}_i(x_i,t)$ is given by

$$\dot{\nabla}\tilde{L}_i(x_i,t) = \frac{\partial}{\partial x_i}\nabla\tilde{L}_i(x_i,t) \times \dot{x}_i + \frac{\partial}{\partial t}\nabla\tilde{L}_i(x_i,t) = \nabla^2\tilde{L}_i(x_i,t)\dot{x}_i + \frac{\partial}{\partial t}\nabla\tilde{L}_i(x_i,t). \tag{13}$$

Here, Assumption 2 ensures the existence of the Hessians of the penalized functions $\tilde{L}_i(x_i,t)$, i.e., $\nabla^2\tilde{L}_i(x_i,t)$. Substituting the solution of (2) with (6) into (13) leads to

$$\dot{\nabla\tilde{L}}_i(x_i,t) = -\beta\sum_{j\in\mathcal{N}_i}\text{sgn}(x_i-x_j) - \nabla\tilde{L}_i(x_i,t). \tag{14}$$

Then we can use the input-to-state stability [37] to analyze the system (14) by treating the term $-\beta\sum_{j\in\mathcal{N}_i}\text{sgn}(x_i - x_j)$ as the input and $\nabla\tilde{L}_i(x_i,t)$ as the state. Since the term $-\beta\sum_{j\in\mathcal{N}_i}\text{sgn}(x_i-x_j)$ is always smaller than $n\beta$, it is obvious that each $\nabla\tilde{L}_i(x_i,t)$ remains bounded for all $t\ge 0$. Note that (10) implies that $\nabla\tilde{L}_i(x_i,t)$ is unbounded at the boundary of D_i. Therefore, it follows from initial condition (9) that each x_i is in the set $D_i=\{x_i\in\mathbb{R}^m \mid g_i(x_i,t)\prec\sigma_i(t)\mathbf{1}_{q_i}\}$ for all $t\ge 0$. That is, (7) is always well defined. □

In the following, in Lemma 3, we prove that the eventual states of the agents satisfy the optimal requirement shown in Lemma 1, i.e., $\lim_{t\to\infty}\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)=\mathbf{0}_m$. The goal of problem (4) is that all the agents' states reach consensus on the optimal trajectory, and thus in Lemma 4, we prove that consensus can be achieved in finite time if all ϕ_i in the controller (6) are bounded, i.e., there exists a time T_2 such that $\|x_i(t)-\frac{1}{n}\sum_{j=1}^n x_j(t)\|=0$ for all $t>T_2$ if all ϕ_i are bounded. Then in Lemma 5, we prove that all ϕ_i associated with the system (2) under the controller (6) are indeed bounded. Finally, in Theorem 1, we present that the goal in (5) can be achieved, i.e., $\lim_{t\to\infty}\|x_i(t)-y^*(t)\|_2=0$ for all $i\in\mathcal{V}$.

Lemma 3: Suppose that Assumptions 1, 2, and 3 hold, the gain condition (8) and the initial condition (9) hold. For the system (2) under the controller (6), the summation of all $\nabla\tilde{L}_i(x_i,t)$ exponentially converges to $\mathbf{0}_m$.

Proof: It follows from Assumption 2 that all $f_i(x_i,t)$ are strongly convex in x_i. Also it follows from Assumption 2 that all $g_{ij}(x_i,t)$ are convex in x_i. From gain condition (8), we know that $\rho_i(t)$ and $\sigma_i(t)$ are always positive. Then it follows from initial condition (9) that $\tilde{L}_i(x_i,t)$ given by (7) must be continuously differentiable and strongly convex in x_i if x_i is in set $D_i=\{x_i\in\mathbb{R}^m\mid g_i(x_i,t)\prec\sigma_i(t)\mathbf{1}_{q_i}\}$. Note that Assumptions 2 and 3 and initial condition (9) hold. Lemma 2 has indicated that this is indeed the case. Therefore, each $\tilde{L}_i(x_i,t)$ must be continuously differentiable and strongly convex in x_i based on our algorithm. Consider the following Lyapunov function candidate:

$$W_1=\frac{1}{2}\left[\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)\right]^T\left[\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)\right]. \tag{15}$$

Note that the Lyapunov candidate W_1 is continuously differentiable. Based on the statements in Remark 7, we do not need to employ nonsmooth analysis in the stability analysis. Then we have

$$\dot{W}_1(t) = \left[\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)\right]^T \times\left[\sum_{i=1}^n\nabla^2\tilde{L}_i(x_i,t)\dot{x}_i+\frac{\partial}{\partial t}\nabla\tilde{L}_i(x_i,t)\right]. \tag{16}$$

Substituting the solution of (2) with (6) into (16) leads to

$$\dot{W}_1(t)=\left[\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)\right]^T\left(\sum_{i=1}^n\nabla^2\tilde{L}_i(x_i,t)\left\{\left[\nabla^2\tilde{L}_i(x_i,t)\right]^{-1}\times\beta\sum_{j\in\mathcal{N}_i}\text{sgn}(x_j-x_i)+\phi_i\right\}+\frac{\partial}{\partial t}\nabla\tilde{L}_i(x_i,t)\right).$$

Since the network is undirected (Assumption 1), we have $\sum_{i=1}^n\beta\sum_{j\in\mathcal{N}_i}\text{sgn}(x_j-x_i)=\mathbf{0}_m$ for all $t\ge 0$. It follows that

$$\dot{W}_1(t)=\left[\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)\right]^T\left[-\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)\right]=-2W_1(t)$$

which indicates that $W_1(t)=e^{-2t}W_1(0)$ for all $t\ge 0$. It can be concluded that $W_1(t)$ exponentially converges to zero, and, thus, $\sum_{i=1}^n\nabla\tilde{L}_i(x_i,t)$ exponentially converges to $\mathbf{0}_m$. □

Lemma 4: Suppose that Assumptions 1–3 hold, the gain condition (8) and the initial condition (9) hold. For the system (2) under the controller (6), if[1] there exists a constant $\bar{\phi}$ such that $\sup_{t\in[0,\infty)}\|\phi_i(t)\|_2\le\bar{\phi},\ \forall i\in\mathcal{V}$ and β satisfies that

$$\beta\ge\frac{2\bar{\phi}mn^2|\mathcal{E}|}{\min_{i\in\mathcal{V}}\left\{\bar{\lambda}_{\min}\left[\left(\nabla^2\tilde{L}_i\right)^{-1}\right]\right\}}+\epsilon^1 \tag{17}$$

where $\epsilon>0$ is a constant, all the states x_i will achieve consensus in finite time, i.e., there exists a time T_2 such that $\|x_i(t)-x_j(t)\|_2=0$, for all $i,j\in\mathcal{V}$ and for all $t>T_2$.

Proof: Define $\left[\nabla^2\tilde{L}(x,t)\right]^{-1}=\text{diag}\left\{\left[\nabla^2\tilde{L}_1(x_1,t)\right]^{-1},\ldots,\left[\nabla^2\tilde{L}_n(x_n,t)\right]^{-1}\right\}$, $x=\left[x_1^T,\ldots,x_n^T\right]^T$, and $\Phi=\left[\phi_1^T,\ldots,\phi_n^T\right]^T$. Consider the Lyapunov candidate

$$W_2(t)=\|\left(\mathcal{D}^T\otimes I_m\right)x\|_1. \tag{18}$$

The solution of (2) with (6) can be written in compact form as

$$\dot{x}=-\beta\left[\nabla^2\tilde{L}(x,t)\right]^{-1}(\mathcal{D}\otimes I_m)\text{sgn}\left[\left(\mathcal{D}^T\otimes I_m\right)x\right]+\Phi. \tag{19}$$

It is obvious that $W_2(t)$ is locally Lipschitz continuous but nonsmooth at some points. Then according to Definition 2, the generalized gradient of $W_2(t)$ is given by

$$\partial W_2(t)=\left(\mathcal{D}^T\otimes I_m\right)^T\left\{\text{SGN}\left[\left(\mathcal{D}^T\otimes I_m\right)x\right]\right\} \tag{20}$$

[1]Here $\min_{i\in\mathcal{V}}\{\bar{\lambda}_{\min}[(\nabla^2\tilde{L}_i)^{-1}]\}$ denotes the smallest value in the set $[\bar{\lambda}_{\min}[(\nabla^2\tilde{L}_1)^{-1}],\bar{\lambda}_{\min}[(\nabla^2\tilde{L}_2)^{-1}],\ldots,\bar{\lambda}_{\min}[(\nabla^2\tilde{L}_n)^{-1}]]$, where $\bar{\lambda}_{\min}[(\nabla^2\tilde{L}_i)^{-1}]$ is defined in Section II-A.

where SGN$(\cdot)$ [2] is the multivalued function defined as (see [36], (20)])

$$\mathrm{SGN}(z)=\begin{cases}1 & \text{if } z>0,\\ [-1,1] & \text{if } z=0,\\ -1 & \text{if } z<0.\end{cases} \quad (21)$$

Then based on Definition 3, the set-valued Lie derivative of $W_2(t)$ is given by

$$\dot{\tilde{W}}_2(t)=\bigcap_{\xi\in\mathrm{SGN}[(\mathcal{D}^T\otimes I_m)x]}\xi^T\left(\mathcal{D}^T\otimes I_m\right)K[f] \quad (22)$$

where $K[f]=\Phi-\beta\left[\nabla^2\tilde{L}(x,t)\right]^{-1}(\mathcal{D}\otimes I_m)\mathrm{SGN}\left[\left(\mathcal{D}^T\otimes I_m\right)x\right]$ is the set-valued Filippov map of the dynamical system (19).

Since there is an intersection operation on the right side of (22), it follows that as long as $\dot{\tilde{W}}_2(t)$ is not empty and there exists $\xi\in\mathrm{SGN}[(\mathcal{D}^T\otimes I_m)x]$ such that $\xi^T(\mathcal{D}^T\otimes I_m)\tilde{f}<0,\ \forall\tilde{f}\in K[f]$, then the result of $\dot{\tilde{W}}_2(t)$ falls into the negative half-plane of the real axis. Arbitrarily choose $\eta\in\mathrm{SGN}[(\mathcal{D}^T\otimes I_m)x]$. Choose $\xi_k=\mathrm{sgn}[(\mathcal{D}^T\otimes I_m)_{k\bullet}x]$ if $\mathrm{sgn}[(\mathcal{D}^T\otimes I_m)_{k\bullet}x]\neq 0$ and $\xi_k=\eta_k$ if $\mathrm{sgn}[(\mathcal{D}^T\otimes I_m)_{k\bullet}x]=0$, where ξ_k and η_k denote the kth element in vectors ξ and η, respectively. If $\dot{\tilde{W}}_2(t)\neq\emptyset$, suppose that $\tilde{a}\in\dot{\tilde{W}}_2(t)$. It follows that

$$\begin{aligned}\tilde{a}=&-\beta\left\{\xi^T\left(\mathcal{D}^T\otimes I_m\right)\left[\nabla^2\tilde{L}(x,t)\right]^{-1}(\mathcal{D}\otimes I_m)\eta\right\}\\&+\xi^T\left(\mathcal{D}^T\otimes I_m\right)\Phi\\ \le&-\beta\left\{\xi^T\left(\mathcal{D}^T\otimes I_m\right)\left[\nabla^2\tilde{L}(x,t)\right]^{-1}(\mathcal{D}\otimes I_m)\xi\right\}\\&+\xi^T\left(\mathcal{D}^T\otimes I_m\right)\Phi\\ \le&-\beta\bar{\lambda}_{\min}\left[\left(\nabla^2\tilde{L}\right)^{-1}\right]\|(\mathcal{D}\otimes I_m)\xi\|_2^2+2\bar{\phi}mn^2|\mathcal{E}|.\end{aligned} \quad (23)$$

If there exists an edge $(i_2,j_2)\in\mathcal{E}$ such that $x_{i_2}\neq x_{j_2}$, then $\|(\mathcal{D}\otimes I_m)\xi\|\geq 1$. It follows that

$$\tilde{a}\le-\beta\bar{\lambda}_{\min}\left[\left(\nabla^2\tilde{L}\right)^{-1}\right]+2\bar{\phi}mn^2|\mathcal{E}|. \quad (24)$$

Since $\beta\geq\frac{2\bar{\phi}mn^2|\mathcal{E}|}{\min_{i\in\mathcal{V}}\{\bar{\lambda}_{\min}[(\nabla^2\tilde{L}_i)^{-1}]\}}+\epsilon=\frac{2\bar{\phi}mn^2|\mathcal{E}|}{\bar{\lambda}_{\min}[(\nabla^2\tilde{L})^{-1}]}+\epsilon$, it follows that if there exists an edge $(i_2,j_2)\in\mathcal{E}$ such that $x_{i_2}\neq x_{j_2}$, then $\tilde{a}\le-\epsilon$. Thus, we can conclude that $\dot{\tilde{W}}_2(t)\le-\epsilon$ if there exists an edge $(i_2,j_2)\in\mathcal{E}$ such that $x_{i_2}(t)\neq x_{j_2}(t)$. Based on the Lebesgue's theory for the Riemann integrability, a function on a compact interval is Riemann integrable if and only if it is bounded and the set of its discontinuous points has measure zero [38]. Therefore, although the time derivative $\dot{W}_2(t)$ here is discontinuous at some time points, it is Riemann integrable. Then, we have

$$W_2(t)-W_2(0)=\int_0^t\dot{W}_2(\tau)d\tau\le-\epsilon t$$

where $t>0$, if there exists an edge $(i_2,j_2)\in\mathcal{E}$ such that $x_{i_2}\neq x_{j_2}$. It follows that

$$W_2(t)\le W_2(0)-\epsilon t \quad (25)$$

if there exists an edge $(i_2,j_2)\in\mathcal{E}$ such that $x_{i_2}\neq x_{j_2}$. Based on the definition of $W_2(t)$ in (18), we have

$$\begin{aligned}W_2(t)&=\|(\mathcal{D}^T\otimes I_m)x\|_1\\&=\frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\|x_i-x_j\|_1.\end{aligned} \quad (26)$$

That is, only if $x_i(t)=x_j(t)$ holds for all edges $(i,j)\in\mathcal{E}$, we have $W_2(t)=0$. Then it follows from (25) that $W_2(t)$ converges to zero in finite time and the convergence time is smaller than or equal to $W_2(0)/\epsilon$. Also $W_2(t)\to 0$ implies that $\|x_i-x_j\|_1\to 0$ for all $i\in\mathcal{V}$ and $j\in\mathcal{N}_i$. Because the network is undirected and connected (see Assumption 1), it follows that all agents reach a consensus in finite time. That is, there exists a time T_2 such that $\|x_i(t)-\frac{1}{n}\sum_{j=1}^n x_j(t)\|_2=0$ for all $i\in\mathcal{V}$ and for all $t>T_2$. □

Lemma 5: Suppose that Assumptions 1–5 hold, the gain condition (8) and the initial condition (9) hold. For the system (2) under the controller (6), all ϕ_i remain bounded. That is, there exists a constant $\bar{\phi}$ such that $\sup_{t\in[0,\infty)}\|\phi_i(t)\|_2\le\bar{\phi}$, for all $i\in\mathcal{V}$.

Proof: To begin with, we prove that each x_{ik} associated with the system (2) under the controller (6) remains in a bounded region, which in turn guarantees that all ϕ_i are bounded. Here, $x_{ik}\in\mathbb{R}$ denotes the kth element in x_i. Note that Assumptions 2 and 3, the initial condition (9), and the gain condition (8) hold. Then using a similar analysis to that in Lemma 3, we have each $\tilde{L}_i(x_i,t)$ is continuously differentiable and strongly convex in x_{ik}. Assume that there exists x_{ik} such that $x_{ik}\to+\infty$ or $x_{ik}\to-\infty$ with the fastest speed among all the elements in x_i. Then due to the strong convexity and the continuous differentiability of $\tilde{L}_i(x_i,t)$, we have $\nabla_{x_{ik}}\tilde{L}_i(x_i,t)\to+\infty$ as $x_{ik}\to+\infty$ and $\nabla_{x_{ik}}\tilde{L}_i(x_i,t)\to-\infty$ as $x_{ik}\to-\infty$. Note that Assumptions 1–3, the initial condition (9), and the gain condition (8) hold. Then it follows from Lemma 3 that it is impossible that all x_{ik} go to infinity at the same time. Without loss of generality, let us assume that $x_{i_1k}\to+\infty$, where $i_1=\mathrm{argmax}_{j\in\mathcal{V}}(x_{jk})$. It follows that $-\beta\sum_{j\in\mathcal{N}_{i_1}}\mathrm{sgn}(x_{i_1k}-x_{jk})\le 0$ when $x_{i_1k}\to+\infty$. Therefore, from (14), it is clear that $\dot{\nabla}_{x_{i_1k}}\tilde{L}_{i_1}(x_{i_1},t)$ must be negative when $x_{i_1k}\to+\infty$. Similarly, assume that $x_{i_2k}\to-\infty$, where $i_2=\mathrm{argmin}_{j\in\mathcal{V}}(x_{jk})$. It follows that $-\beta\sum_{j\in\mathcal{N}_{i_2}}\mathrm{sgn}(x_{i_2k}-x_{jk})\geq 0$ when $x_{i_2k}\to-\infty$. Therefore, $\dot{\nabla}_{x_{i_2k}}\tilde{L}_{i_2}(x_{i_2},t)$ must be positive when $x_{i_2k}\to-\infty$. The decreasing $\nabla_{x_{ik}}\tilde{L}_i(x_i,t)$ when $x_{ik}\to+\infty$ and increasing $\nabla_{x_{ik}}\tilde{L}_i(x_i,t)$ when $x_{ik}\to-\infty$ will result in a bounded $\nabla_{x_{ik}}\tilde{L}_i(x_i,t)$ and, thus, a bounded x_{ik}, which contradicts with the unbounded x_i assumption. Hence, all x_i must be bounded.

[2]With the piecewise-differentiable signum function involved in algorithm (6), the solution of (2) with (6) should be replaced by inclusions at a point of discontinuity.

Then, we will prove that all $\nabla \tilde{L}_i(x_i, t)$ are bounded for all time. It follows from Lemma 3 that $\sum_{i=1}^n \nabla \tilde{L}_i(x_i, t)$ is always bounded. Since all x_i are bounded, we have all $\nabla f_i(x_i, t)$ and $\nabla g_{ij}(x_i, t)$ must be bounded. Then using an argument similar to [27], Lemma 2], all $\frac{1}{\sigma_i(t)-g_{ij}(x_i,t)}$ are bounded. Therefore, each $\nabla \tilde{L}_i(x_i, t)$ is always bounded for all $t \geq 0$ and for all $i \in \mathcal{V}$.

Next, we will prove that all $[\nabla^2 \tilde{L}_i(x_i, t)]^{-1}$ are bounded for all time. Since all $\tilde{L}_i(x_i, t)$ are continuous differentiable and strongly convex in its corresponding x_i, then based on the statements in [34, Sec. 9.1.2], we know that all $\nabla^2 \tilde{L}_i(x_i, t)$ satisfy

$$m(t) I_n \leq \nabla^2 \tilde{L}_i(x_i, t) \leq M(t) I_n$$

with $m(t), M(t) \in \mathbb{R}_{>0}$, which implies that all $[\nabla^2 \tilde{L}_i(x_i, t)]^{-1}$ are bounded and positive definite for all $t \geq 0$.

At last, given that all $\nabla \tilde{L}_i(x_i, t)$ and $\nabla^2 \tilde{L}_i(x_i, t)$ are bounded for all time, under Assumptions 4 and 5, it is easy to see that all $\frac{\partial}{\partial t} \nabla \tilde{L}_i(x_i, t)$ remain bounded for all $t \geq 0$.

Since $[\nabla^2 \tilde{L}_i(x_i, t)]^{-1}$, $\nabla \tilde{L}_i(x_i, t)$, and $\frac{\partial}{\partial t} \nabla \tilde{L}_i(x_i, t)$ are bounded for all $i \in \mathcal{V}$ and for all $t \geq 0$, we can get the conclusion that $\phi_i(t)$ is bounded for all $i \in \mathcal{V}$ and for all $t \geq 0$. □

Theorem 1: Suppose that Assumptions 1–3 hold, the initial condition (9) and the gain conditions (8) and (17) hold. For the system (2) under the controller (6), all the states x_i will converge to the optimal solution $y^*(t)$ in (3) eventually.

Proof: Define

$$\tilde{y}(t)^* \in \mathbb{R}^m = \text{argmin} \sum_{i=1}^n \tilde{L}_i[y(t), t] \tag{27}$$

where $\tilde{L}_i[y(t), t]$ is each agent's penalized objective function defined by (7). Note that Assumptions 1–5, initial condition (9), and gain condition (8) hold. It follows from Lemma 5 that all ϕ_i associated with the system (2) under the controller (6) are bounded for all $t \geq 0$, which in turn implies that $x_i(t) = x_j(t)$, $\forall i, j \in \mathcal{V}$ in finite time according to Lemma 4. Moreover, based on Lemma 3, we know that $\lim_{t\to\infty} \sum_{i=1}^n \nabla \tilde{L}_i(x_i, t) = \mathbf{0}_m$. Using a similar analysis to that in Lemma 3, we have that each $\tilde{L}_i(x_i, t)$ is continuously differentiable and strongly convex in x_i. Based on Lemma 1, we have

$$\sum_{i=1}^n \nabla \tilde{L}_i[\tilde{y}^*(t), t] = \mathbf{0}_m. \tag{28}$$

Then it follows that all x_i will converge to the optimal solution $\tilde{y}^*(t)$ in (27), i.e., $\lim_{t\to\infty} x_i(t) = \tilde{y}^*(t)$, $\forall i \in \mathcal{V}$.

Define

$$\hat{y}^*(t) \in \mathbb{R}^m = \text{argmin} \sum_{i=1}^n f_i[y(t), t]$$
$$\text{s.t.} \quad g_{ij}[y(t), t] \leq \sigma_i(t), \ \forall i \in \mathcal{V}, j = 1, \ldots, q_i. \tag{29}$$

The Lagrangian function of problem (29) can be written as

$$\text{Lag} = \sum_{i=1}^n f_i[y(t), t] + \sum_{i=1}^n \sum_{j=1}^{q_i} \lambda_{ij}(t) \left\{ g_{ij}[y(t), t] - \sigma_i(t) \right\} \tag{30}$$

where $\lambda_{ij}(t) > 0$ are the Lagrangian multipliers. The corresponding dual function of problem (29) is

$$g[\lambda_{ij}(t)] = \sup_{y(t)} \left(x \sum_{i=1}^n f_i[y(t), t] + \sum_{i=1}^n \sum_{j=1}^{q_i} \lambda_{ij}(t) \left\{ g_{ij}[y(t), t] - \sigma_i(t) \right\} \right). \tag{31}$$

It follows from (10) and (28) that

$$\sum_{i=1}^n \nabla \tilde{L}_i\left[\tilde{y}^*(t), t\right] = \sum_{i=1}^n \nabla f_i\left[\tilde{y}^*(t), t\right] + \sum_{i=1}^n \sum_{j=1}^{q_i} \frac{\nabla g_{ij}\left[\tilde{y}^*(t), t\right]}{\rho_i(t)\{\sigma_i(t) - g_{ij}\left[\tilde{y}^*(t), t\right]\}} = \mathbf{0}_m. \tag{32}$$

Define $\tilde{\lambda}_{ij}(t) = \frac{1}{\rho_i(t)\{\sigma_i(t)-g_{ij}[\tilde{y}^*(t),t]\}}$. We see that $\tilde{y}^*(t)$ minimizes the Lagrangian function defined in (30), for $\lambda_{ij}(t) = \tilde{\lambda}_{ij}(t)$. Therefore, the dual function (31) at point $\tilde{\lambda}_{ij}(t)$ is

$$\begin{aligned} g\left[\tilde{\lambda}_{ij}(t)\right] &= \sum_{i=1}^n f_i[\tilde{y}^*(t), t] + \sum_{i=1}^n \sum_{j=1}^{q_i} \lambda^*_{ij}(t)\{g_{ij}[\tilde{y}^*(t), t] - \sigma_i(t)\} \\ &= \sum_{i=1}^n f_i[\tilde{y}^*(t), t] - \sum_{i=1}^n \sum_{j=1}^{q_i} \frac{1}{\rho_i(t)} \\ &\leq \sum_{i=1}^n f_i[\hat{y}^*(t), t]. \end{aligned} \tag{33}$$

The last inequality in (33) holds since the dual function provides a lower bound to the solution of the primal problem (29).

It follows that

$$\left| \sum_{i=1}^n f_i[\hat{y}^*(t), t] - \sum_{i=1}^n f_i[\tilde{y}^*(t), t] \right| \leq \sum_{j=1}^n \sum_{k=1}^{q_j} \rho_j^{-1}(t). \tag{34}$$

Note that $y^*(t) \in \mathbb{R}^m$ is the optimal solution of problem (3). Then we can use the perturbation and sensitivity analysis in [34, Sec. 5.9] to analyze problem (29) by treating (29) as a perturbed version of the problem (3) after including the slack variables $\sigma_i(t)$ in the constraints. Under Assumption 3, the optimal solution $y^*(t)$ can be characterized using the KKT conditions for all $t \geq 0$. Then we have

$$\left| \sum_{i=1}^n f_i[\hat{y}^*(t), t] - \sum_{i=1}^n f_i[y^*(t), t] \right| \leq \sum_{j=1}^n \sum_{k=1}^{q_j} \lambda^*_{jk}(t) \sigma_j(t). \tag{35}$$

Hence, because $\lim_{t\to\infty} \rho_i(t) = \infty$ and $\lim_{t\to\infty} \sigma_i(t) = 0$ for all $i \in \mathcal{V}$, we have

$$\lim_{t\to\infty} \left| \sum_{i=1}^n f_i[y^*(t), t] - \sum_{i=1}^n f_i[\tilde{y}^*(t), t] \right| = 0. \tag{36}$$

Since we assume that the optimal solution $y^*(t)$ is unique, it follows that $\lim_{t\to\infty} x_i(t) = y^*(t),\ \forall i \in \mathcal{V}$. □

Remark 8: (Application in unconstrained optimization problems) As a byproduct, the algorithm (6) can also be used for distributed unconstrained optimization problems with much more relaxed assumptions on the objective functions (e.g., those with nonidentical Hessians) than those in [25]. In particular, it is applicable to objective functions that are strongly convex and twice continuously differentiable with respect to x_i and whose partial gradients with respect to time, i.e., $\frac{\partial \nabla f_i(x_i,t)}{\partial t}$, are bounded.

IV. Distributed Time-Varying Optimization With Nonlinear Inequality and Linear Equality Constraints

In this section, we extend the results in Section III-A to take into account both time-varying nonlinear inequality and linear equality constraints. The goal is to design $u_i(t)$ using only local information and local interaction for system (2), such that all the agents work together to find the optimal trajectory $r^*(t) \in \mathbb{R}^m$ defined as

$$r^*(t) = \operatorname{argmin} \sum_{i=1}^{n} f_i[r(t), t],$$
$$\text{s.t.} \quad g_i[r(t), t] \preceq \mathbf{0}_{q_i}, \quad A_i(t) r(t) = b_i(t), \quad \forall i \in \mathcal{V} \tag{37}$$

where $A_i(t) \in \mathbb{R}^{p_i \times m}$ and $b_i(t) \in \mathbb{R}^{p_i}$ are the local equality constraint functions. It is assumed that $A_i(t)$ and $b_i(t)$ are known only to agent i and are continuously differentiable with respect to t. Here the goal is that each state $x_i(t)$ converges to the optimal solution $r^*(t)$, i.e.,

$$\lim_{t\to\infty} [x_i(t) - r^*(t)] = \mathbf{0}_m. \tag{38}$$

We need an additional assumption.

Assumption 6: (Full rank condition) The number of the equality constraints is less than the dimension of the agents' states, i.e., $p_i < m$, and $\operatorname{rank}(A_i) = p_i$, for all $i \in \mathcal{V}$. And for all $t \geq 0$, there exists at least one r such that $A_i(t) r = b_i(t)$ for all $i \in \mathcal{V}$.

Assumption 6 ensures that the system of equations $A_i(t) x_i(t) = b_i(t)$ is consistent and has infinitely many solutions at each $t \geq 0$. We assume that the optimal solution $r^*(t)$ in (37) is unique for all $t \geq 0$. For notational simplicity, we will remove the time index t from the variables $A_i(t)$ and $b_i(t)$ in most remaining parts of this article and only keep it in some places when necessary.

A. Distributed Algorithm Design

In this subsection, we derive a distributed control algorithm such that (38) holds. In addition, in algorithm (6), it is required that the upper bounds on auxiliary variable ϕ_i be known in advance such that the control gain β can be chosen to satisfy (17). To remove this restriction, we introduce an adaptive gain design in the algorithm.

We design the following controller for agent i:

$$u_i = -w_i(t) \left[\nabla^2 \hat{L}_i(x_i, t)\right]^{-1} \sum_{j\in\mathcal{N}_i} \operatorname{sgn}(x_i - x_j) + \phi_i \tag{39a}$$

$$\phi_i = -\left[\nabla^2 \hat{L}_i(x_i, t)\right]^{-1} \left[\nabla \hat{L}_i(x_i, t) + \frac{\partial}{\partial t} \nabla \hat{L}_i(x_i, t)\right] \tag{39b}$$

$$\dot{s}_i(t) = \sum_{j\in\mathcal{N}_i} \operatorname{sgn}(\|x_i - x_j\|_1) \tag{39c}$$

$$w_i(t) = z_i(t) + s_i(t) \tag{39d}$$

$$\dot{z}_i(t) = -\alpha \sum_{j\in\mathcal{N}_i} \operatorname{sgn}[w_i(t) - w_j(t)]. \tag{39e}$$

In (39a) and (39b), $w_i(t) \in \mathbb{R}$ is a dynamic gain, and $\hat{L}_i(x_i, t)$ is the penalized objective function of agent i given by

$$\hat{L}_i(x_i, t) = f_i(x_i, t) - \frac{1}{\rho_i(t)} \sum_{j=1}^{q_i} \log[\sigma_i(t) - g_{ij}(x_i, t)] + \frac{\kappa_i}{2} \|A_i x_i - b_i\|_2^2 \tag{40}$$

where $\kappa_i \in \mathbb{R}_{>0}$ is a constant gain. Note that (40) includes the local log-barrier penalty functions and the local quadratic penalty functions to account for, respectively, the inequality and equality constraints in (37). In (39c), the gain $s_i(t) \in \mathbb{R}$ is adapted according to the state differences between agent i and its neighbors. The dynamic gain $w_i(t)$ is the output of a distributed average tracking estimator given by (39d) and (39e), where $\alpha \in \mathbb{R}_{>0}$ is a constant gain, $z_i(t) \in \mathbb{R}$ is the internal state, and $s_i(t)$ is the reference signal associated with agent i. In the next subsection, we will show that the dynamic gain $w_i(t)$ can help all the agents achieve consensus without knowing certain prior information. In addition, we have

$$\nabla \hat{L}_i(x_i, t) = \nabla f_i(x_i, t) + \sum_{j=1}^{q_i} \frac{\nabla g_{ij}(x_i, t)}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]} + \kappa_i A_i^T (A_i x_i - b_i) \tag{41}$$

$$\begin{aligned} \frac{\partial}{\partial t} \nabla \hat{L}_i(x_i, t) = {} & \frac{\partial}{\partial t} \nabla f_i(x_i, t) + \sum_{j=1}^{q_i} \frac{\partial \nabla g_{ij}(x_i, t)/\partial t}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]} \\ & - \sum_{j=1}^{q_i} \frac{\dot{\rho}_i(t) \nabla g_{ij}(x_i, t)}{\rho_i^2(t)[\sigma_i(t) - g_{ij}(x_i, t)]} \\ & - \sum_{j=1}^{q_i} \frac{\dot{\sigma}_i(t) \nabla g_{ij}(x_i, t)}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]^2} \\ & + \sum_{j=1}^{q_i} \frac{\nabla g_{ij}(x_i, t) \partial g_{ij}(x_i, t)/\partial t}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i, t)]^2} \\ & + 2\kappa_i A_i^T \dot{A}_i x_i - \kappa_i \dot{A}_i b_i - \kappa_i A_i \dot{b}_i \end{aligned} \tag{42}$$

$$\nabla^2 \hat{L}_i(x_i,t) = \nabla^2 f_i(x_i,t) + \sum_{j=1}^{q_i} \frac{\nabla^2 g_{ij}(x_i,t)}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i,t)]} + \sum_{j=1}^{q_i} \frac{\nabla g_{ij}(x_i,t)\nabla g_{ij}(x_i,t)^T}{\rho_i(t)[\sigma_i(t) - g_{ij}(x_i,t)]^2} + \kappa_i A_i^T A_i. \quad (43)$$

To make algorithm (39) work, the gain α and the initial internal states $s_i(0)$ and $z_i(0)$ need to satisfy

$$\alpha > n \quad (44)$$

$$s_i(0) > 0, \quad \forall i \in \mathcal{V} \quad (45)$$

$$z_i(0) = 0, \quad \forall i \in \mathcal{V}. \quad (46)$$

Remark 9: (Reason for adaptive control gains) In algorithm (6), it is required that the bounds on the auxiliary variables ϕ_i be known in advance such that the control gain β can be chosen to satisfy (17). However, these bounds might not be obtained or estimated accurately in certain circumstances. Therefore, in algorithm (39), adaptive control gains are designed to remove the need for using the information of these bounds. The tradeoff is that the virtual variable $w_i(t)$ need to be communicated between neighbors to implement algorithm (39).

Remark 10: (Distributed algorithms) Algorithm (39) is distributed since each agent only needs its own information and information received from its neighbors. Take agent i as an example. Agent i uses it own information: $x_i(t)$, $w_i(t)$, $s_i(t)$, $z_i(t)$, and Hessian and gradient information of its penalized objective function. It is worthwhile to mention that agent i only needs to know its own penalty parameters: $\rho_i(t)$, $\sigma_i(t)$, and $\kappa_i(t)$. Moreover, agent i needs to know information received from it neighbors: $x_j(t)$ and $w_j(t)$, $j \in \mathcal{N}_i$. While a common control gain α is needed for all the agents, α is a constant and only required to be larger than n. The work [39] provides an answer about how to estimate n in a distributed way.

The estimator given by (39d) and (39e) guarantees that the gains $w_i(t)$ for all agents become uniform after a finite time as shown in the following lemma.

Lemma 6: Suppose that Assumption 1, the gain condition (44), and the initial condition (46) hold. For system (2) under the controller (39), all $w_i(t)$ will converge to $\frac{1}{n}\sum_{j=1}^n s_j(t)$ in finite time. That is, there exists a time T_0 such that $w_i(t) = \frac{1}{n}\sum_{i=j}^n s_j(t)$ for all $i \in \mathcal{V}$, and $t \geq T_0$.

Proof: The proof is evident based on [28], Th. 1]. □

In the following, for notational simplicity, we will remove the time index t from $s_i(t)$, $z_i(t)$, and $w_i(t)$ in most remaining parts of this article and only keep it in some places when necessary.

B. Convergence Analysis

In this subsection, the asymptotical convergence of the system (2) under the controller (39) to the vicinity of the optimal solution in (37) is established. Note that in the case where there only exist nonlinear inequality constraints, algorithm (6) is capable of tracking the optimal solution in (3) with a zero tracking error. Since we use the quadratic penalty functions to account for the equality constraints, the larger the penalty weight κ, the better the approximation x_i to a solution of the original problem (37). We need an additional assumption.

Assumption 7: (Bounds about equality constraint functions) The time derivatives of the local constraint parameters are bounded. That is, there exist constants $\bar{a}$ and $\bar{b}$ such that $\sup_{t\in[0,\infty)} \|\dot{A}_i(t)\|_2 \leq \bar{a}$ and $\sup_{t\in[0,\infty)} \|\dot{b}_i(t)\|_2 \leq \bar{b}$, for all $i \in \mathcal{V}$, and for all $t \geq 0$.

In the following, we give the main results on the distributed continuous-time optimization with time-varying nonlinear inequality and linear equality constraints.

Theorem 2: Suppose that Assumptions 1 to 7 hold, the initial conditions (9), (45), and (46) hold, and the gain conditions (8) and (44) hold. For system (2) under controller (39), all the states x_i will converge to the vicinity of the optimal solution $r^*(t)$ in (37) eventually, i.e., $\lim_{t\to\infty} x_i(t) = \bar{\bar{r}}^*(t)$, for all $i \in \mathcal{V}$, where $\sup_t \|\bar{\bar{r}}^*(t) - r^*(t)\| \leq \varepsilon$ and ε is a constant.

Proof: We first show that (40) is always well defined under our proposed algorithm (39). The time derivative of $\nabla\hat{L}_i(x_i,t)$ is given by

$$\dot{\nabla}\hat{L}_i(x_i,t) = \nabla^2\hat{L}_i(x_i,t)\dot{x}_i + \frac{\partial}{\partial t}\nabla\hat{L}_i(x_i,t). \quad (47)$$

Substituting the solution of (2) with (39) into (47) leads to

$$\dot{\nabla}\hat{L}_i(x_i,t) = -w_i \sum_{j\in\mathcal{N}_i} \mathrm{sgn}(x_i - x_j) - \nabla\hat{L}_i(x_i,t). \quad (48)$$

Then using a similar analysis to that in Lemma 2, we have that each $\nabla\hat{L}_i(x_i,t)$ must remain bounded for all time, and, thus, each x_i is in the set $D_i = \{x_i \in \mathbb{R}^m \mid g_i(x_i,t) \prec \sigma_i(t)\mathbf{1}_{q_i}\}$ for all $t \geq 0$. That is, (41) is well defined for all $t \geq 0$.

Then we show that the agents' states under controller (39) satisfy the optimal requirement shown in Lemma 1 eventually. It follows from Assumption 2 that all $f_i(x_i,t)$ are strongly convex in x_i. Also it follows from Assumption 2 that all $g_{ij}(x_i,t)$ are convex in x_i. From gain condition (8), we know that $\rho_i(t)$ and $\sigma_i(t)$ are always positive. In addition, from Assumption 6, we know that $A_i^T A_i$ must be positive semidefinite. Note that κ_i is a positive constant. Therefore, from initial condition (9), we know that $\hat{L}_i(x_i,t)$ given by (40) must be continuously differentiable and strongly convex in x_i if x_i is in the set $D_i = \{x_i \in \mathbb{R}^m \mid g_i(x_i,t) \prec \sigma_i(t)\mathbf{1}_{q_i}\}$. The above analysis has indicated that this is indeed the case. Therefore, each $\hat{L}_i(x_i,t)$ must be continuously differentiable and strongly convex in x_i based on our algorithm (39). Consider the Lyapunov function candidate

$$W_3 = \frac{1}{2}\left[\sum_{i=1}^n \nabla\hat{L}_i(x_i,t)\right]^T \left[\sum_{i=1}^n \nabla\hat{L}_i(x_i,t)\right]. \quad (49)$$

Similarly, based on the statements in Remark 7, there is no need to employ the nonsmooth analysis. Then we have

$$\dot{W}_3(t) = \left[\sum_{i=1}^n \nabla\hat{L}_i(x_i,t)\right]^T \times \left[\sum_{i=1}^n \nabla^2\hat{L}_i(x_i,t)\dot{x}_i + \frac{\partial}{\partial t}\nabla\hat{L}_i(x_i,t)\right]. \quad (50)$$

Substituting the solution of (2) with (39) into (50) leads to

$$\dot{W}_3(t)=\left[\sum_{i=1}^{n}\nabla\hat{L}_i(x_i,t)\right]^T\left(\sum_{i=1}^{n}\nabla^2\hat{L}_i(x_i,t)\left\{\left[\nabla^2\hat{L}_i(x_i,t)\right]^{-1}\times w_i\sum_{j\in\mathcal{N}_i}\text{sgn}(x_j-x_i)+\phi_i\right\}+\frac{\partial}{\partial t}\nabla\hat{L}_i(x_i,t)\right).$$

Notice that Assumption 1, the initial condition (46), and the gain condition (44) hold, it follows from Lemma 6 that $w_i=w_j$, for all $i,j\in\mathcal{V}$ and for all $t\geq T_0$. Since the network is undirected (Assumption 1), we have $\sum_{i=1}^{n}w_i\sum_{j\in\mathcal{N}_i}\text{sgn}(x_j-x_i)=\mathbf{0}_m$ for all $t\geq T_0$. Then for all $t\geq T_0$, we have

$$\dot{W}_3(t)=\left[\sum_{i=1}^{n}\nabla\hat{L}_i(x_i,t)\right]^T\left[-\sum_{i=1}^{n}\nabla\hat{L}_i(x_i,t)\right]=-2W_3(t)$$

indicating that $W_3(t)=e^{-2(t-T_0)}W_3(T_0)$ for all $t\geq T_0$. The time derivative of $\nabla\hat{L}_i(x_i,t)$ is given by (47). It follows that each $\nabla\hat{L}_i(x_i,t)$ is bounded at all time. Therefore, $W_3(T_0)$ is bounded. Then it can be concluded that $W_3(t)$ exponentially converges to zero, and, thus, $\sum_{i=1}^{n}\nabla\hat{L}_i(x_i,t)$ exponentially converges to $\mathbf{0}_m$.

Next, we show that all x_i remain bounded under algorithm (39). Based on (39c)–(39e), the time derivative of $w_i(t)$ is given by

$$\begin{aligned}\dot{w}_i(t)&=\dot{z}_i(t)+\dot{s}_i(t)\\&=-\alpha\sum_{j\in\mathcal{N}_i}\text{sgn}[w_i(t)-w_j(t)]\\&\quad+\sum_{j\in\mathcal{N}_i}\text{sgn}(\|x_i(t)-x_j(t)\|_1).\end{aligned}$$

It follows that $\dot{w}_{\min}(t)$ must be nonnegative, where $w_{\min}(t)$ is defined as $\min_i w_i(t)$. The reason is that $\text{sgn}[w_{\min}(t)-w_j(t)]$ must be nonpositive. Note that $w_i(0)=z_i(0)+s_i(0)$. It follows from (45) and (46) that $w_i(0)>0, \forall i\in\mathcal{V}$, which in turn guarantees that $w_{\min}(t)$ and thus all $w_i(t)$ are positive for all $t\geq 0$. Then similar to the proof in Lemma 5, we have that all x_i remain bounded for all $t\geq 0$. Given the above results and Assumption 7, using similar analysis to that in Lemma 5, it is easy to prove that each $\phi_i(t)$ is bounded for all the time.

Next, we show that all the agents reach a consensus in finite time. Consider any edge $(i,j)\in\mathcal{E}$. Let $0<t_{11}^{ij}<t_{12}^{ij}<t_{21}^{ij}<t_{22}^{ij}<\cdots$ denote the contiguous switching times such that $x_i\neq x_j$ during the time interval $\left[t_{k1}^{ij},t_{k2}^{ij}\right]$ and $x_i=x_j$ during the time interval $\left[t_{k2}^{ij},t_{k+1,1}^{ij}\right)$, $k=1,2,\ldots$. From the dynamics of s_i in (39c), it is easy to see that $s_i(\infty)=\sum_{j\in\mathcal{N}_i}\sum_{k=1}^{\infty}\left(t_{k2}^{ij}-t_{k1}^{ij}\right)+s_i(0)$. It follows from Lemma 6 that $w_i(t)=w_j(t)=\frac{1}{n}\sum_{k=1}^{n}s_k(t)$ for all $i,j\in\mathcal{V}$ and for all $t\geq T_0$. If for all edges, $\sum_{k=1}^{\infty}\left(t_{k2}^{ij}-t_{k1}^{ij}\right)<\infty,\ \forall(i,j)\in\mathcal{E}$, it is clear that $t_{k2}^{ij}-t_{k1}^{ij}\to 0$ as $k\to\infty$. Since the graph is connected (Assumption 1), it follows that consensus can be achieved eventually. If there exists an edge (i,j) such that $\sum_{k=1}^{\infty}\left(t_{k2}^{ij}-t_{k1}^{ij}\right)=\infty$, then we have $s_i(\infty)=\infty$ and $w_i(\infty)=w_j(\infty)=\infty$ for all $i,j\in\mathcal{V}$. Then there must exist a time $T_1>T_0$ such that $w_i(T_1)=w_j(T_1)>\frac{2\bar{\phi}mn^2|\mathcal{E}|}{\min_{i\in\mathcal{V}}\{\bar{\lambda}_{\min}[(\nabla^2\hat{L}_i)^{-1}]\}}$ for all $i,j\in\mathcal{V}$ and all $t\geq T_1$. Then similar to the proof of Lemma 4, we have that all agents reach a consensus in finite time, i.e., there exists a time T_2 such that $\|x_i(t)-x_j(t)\|_2=0$ for all $t>T_2$.

Now, we show that all the agents with the system (2) under the controller (39) converge to the vicinity of the optimal solution $r^*(t)$ in (37). Define

$$\tilde{r}^*(t)\in\mathbb{R}^m=\text{argmin}\sum_{i=1}^{n}\hat{L}_i[r(t),t]$$

where $\hat{L}_i[r(t),t]$ is each agent's penalized objective function defined by (40). Summarizing the above analysis and similar to the analysis in Theorem 1, it follows from Lemma 1 that all x_i converge to the optimal solution $\tilde{r}^*(t)$, i.e., $\lim_{t\to\infty}x_i(t)=\tilde{r}^*(t),\ \forall i\in\mathcal{V}$.

Define

$$\begin{aligned}\hat{r}^*(t)\in\mathbb{R}^m&=\text{argmin}\sum_{i=1}^{n}f_i[r(t),t]+\frac{\kappa_i}{2}\|A_ir(t)-b_i\|_2^2\\\text{s.t.}\quad& g_{ij}[r(t),t]\leq\sigma_i(t),\ \forall i\in\mathcal{V},j=1,\ldots,q_i.\end{aligned}$$

Similar to the analysis in Theorem 1, we know that

$$\begin{aligned}&\left|\sum_{i=1}^{n}\left\{f_i[\hat{r}^*(t),t]+\frac{\kappa_i}{2}\|A_i\hat{r}^*(t)-b_i\|_2^2\right\}\right.\\&\left.\quad-\sum_{i=1}^{n}\left\{f_i[\tilde{r}^*(t),t]+\frac{\kappa_i}{2}\|A_i\tilde{r}^*(t)-b_i\|_2^2\right\}\right|\\&\leq\sum_{j=1}^{n}\sum_{k=1}^{q_j}\rho_j^{-1}(t).\end{aligned}$$

Define

$$\begin{aligned}\bar{\bar{r}}(t)^*\in\mathbb{R}^m&=\text{argmin}\sum_{i=1}^{n}f_i[r(t),t]+\frac{\kappa_i}{2}\|A_ir(t)-b_i\|_2^2\\&\quad\text{s.t. } g_{ij}(x_i,t)\leq 0,\ \forall i\in\mathcal{V},j=1,\ldots,q_i.\end{aligned}\tag{51}$$

Then based on [34, Sec. 5.9], we have

$$\begin{aligned}&\left|\sum_{i=1}^{n}\left\{f_i[\hat{r}^*(t),t]+\frac{\kappa_i}{2}\|A_i\hat{r}^*(t)-b_i\|_2^2\right\}\right.\\&\left.\quad-\sum_{i=1}^{n}\left\{f_i[\bar{\bar{r}}^*(t),t]+\frac{\kappa_i}{2}\|A_i\bar{\bar{r}}^*(t)-b_i\|_2^2\right\}\right|\\&\leq\sum_{j=1}^{n}\sum_{k=1}^{q_j}\lambda_{jk}^*(t)\sigma_j(t)\end{aligned}$$

where $\lambda_{jk}(t)$ are the Lagrangian multipliers corresponding to the inequality constraint defined in (51), and $\lambda_{jk}^*(t)$ are the optimal Lagrangian multipliers. Hence, because $\lim_{t\to\infty}\rho_i(t)=\infty$ and

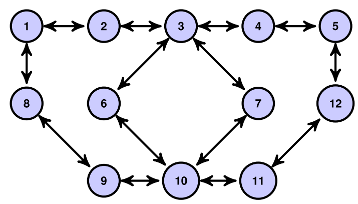

Fig. 1. Undirected graph.

$\lim_{t\to\infty} \sigma_i(t) = 0$ for all $i \in \mathcal{V}$, we have

$$\lim_{t\to\infty} \left| \sum_{i=1}^{n} \left\{ f_i[\bar{\bar{r}}^*(t), t] + \frac{\kappa_i}{2} \|A_i \bar{\bar{r}}^*(t) - b_i\|_2^2 \right\} - \sum_{i=1}^{n} \left\{ f_i[\tilde{r}^*(t), t] + \frac{\kappa_i}{2} \|A_i \tilde{r}^*(t) - b_i\|_2^2 \right\} \right| = 0$$

which indicates that $\lim_{t\to\infty} x_i(t) = \bar{\bar{r}}^*(t)$, $\forall i \in \mathcal{V}$. Then based on the standard quadratic penalty theory [34], $\bar{\bar{r}}^*(t)$ is in the neighborhood of the optimal solution $r^*(t) \in \mathbb{R}^m$ in (37), i.e., $\lim_{t\to\infty} x_i(t) = \bar{\bar{r}}^*(t)$, for all $i \in \mathcal{V}$, where $\sup_t \|\bar{\bar{r}}^*(t) - r^*(t)\| \le \varepsilon$ with ε being a constant. And the larger the penalty parameters κ_i, the smaller of ε. The conclusion of the theorem then follows by combining the above statements. □

V. Numerical Simulation Results

In this section, the proposed distributed time-varying constrained optimization algorithms are illustrated through two simulation cases. In both cases, we consider a network with $n = 12$ and $m = 2$. The network topology is shown by the undirected graph in Fig. 1. Let $x_i = [x_i^p, y_i^p]^T \in \mathbb{R}^2$ denote the states of each agent. Agent i is assigned a local objective function $f_i = \frac{1}{2}\left[x_i^p(t) + i\sin(t)\right]^2 + \frac{3}{2}\left[y_i^p(t) - i\cos(t)\right]^2$, $i \in \mathcal{V}$.

First, we show the simulation result using algorithm (6). Assume that agent j is assigned a constraint function $y_j^p(t) - x_j^p(t) - \cos(t) \le 0$, for all $j \in [1, 2, \ldots, 6]$, and agent k is assigned a constraint function $x_k^p(t)y_k^p(t) - 5t \le 0$, for all $k \in [7, 8, \ldots, 12]$. All the initial states $x_i^p(0)$ and $y_i^p(0)$ are generated randomly from the range $[-5, 5]$. We choose $\beta = 15$, $\rho_i(t) = 10\exp(0.05t)$ and $\sigma_i(t) = 30\exp(-t)$ for all $i \in \mathcal{V}$. Therefore, the initial condition (9) and the gain condition (8) are satisfied. The state trajectories of the agents are shown in Fig. 2. We can see that all the agents track the optimal trajectory eventually which is consistent with Theorem 1. The constraint result is shown in Fig. 3. In our simulation, agents 1–6 are assigned the constraint function $y_i^p(t) - x_i^p(t) - \cos(t) \le 0$, $i \in [1, \ldots, 6]$; so all $y_i^p(t) - x_i^p(t) - \cos(t) - \sigma_i(t)$, $i \in [1, \ldots, 6]$ always remain negative. Agents 7–12 are assigned the constraint function $x_i^p(t)y_i^p(t) - 5t \le 0$, $i \in [7, \ldots, 12]$, and, thus, all $x_i^p(t)y_i^p(t) - 5t - \sigma_i(t)$, $i \in [7, \ldots, 12]$ always remain negative.

We then show the simulation result using algorithm (39). Assume that agent j is assigned a constraint function $y_j^p(t) - x_j^p(t) - \cos(t) \le 0$, for all $j \in [1, \ldots, 6]$, and agent k is assigned a constraint function $y_k^p(t) + x_k^p(t) - t - 3 = 0$, for all $k \in [7, \ldots, 12]$. The initial states $x_i^p(0)$ and $y_i^p(0)$, $i \in \mathcal{V}$ are generated randomly from the range $[-5, 5]$. We choose $\kappa = 12, \alpha = 15$, $\rho_i(t) = 10\exp(0.05t)$, $\sigma_i(t) = 30\exp(-t)$, and $z_i(0) = 0$, $s_i(0) = 5$ for all $i \in \mathcal{V}$. Therefore, initial condition (9), the initial conditions (45), and (46) and the gain conditions (8) and (44) are satisfied. The state trajectories of the agents are shown in Fig. 4. We can see that all the agents converge to the vicinity of the optimal trajectory eventually which is consistent with Theorem 2. The constraint results are shown in Fig. 5. In our simulation, agents 1–6 are assigned the constraint function $y_i^p(t) - x_i^p(t) - \cos(t) \le 0$, $i \in [1, \ldots, 6]$, and thus all $y_i^p(t) - x_i^p(t) - \cos(t) - \sigma_i(t)$, $i \in [1, \ldots, 6]$ always remain negative. Moreover, all the equality constraint functions $y_i^p(t) + x_i^p(t) - t - 3$, $i \in [7, \ldots, 12]$ converge to the neighborhood of the zero line eventually.

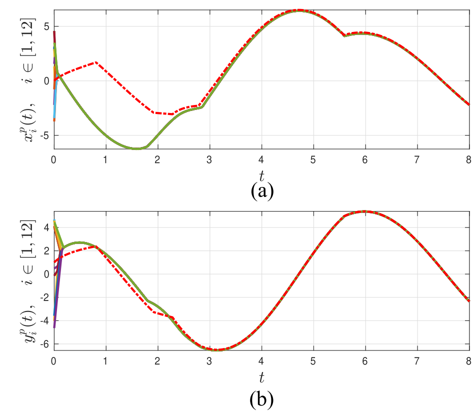

Fig. 2. State trajectories of all the agents with system (2) under controller (6). The red dashed line is the optimal solution and the other solid lines are the trajectories of all agents' states.

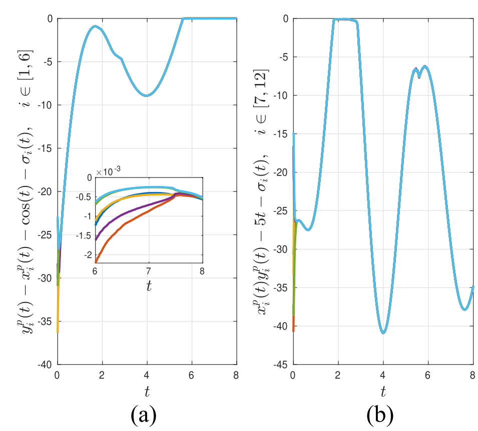

Fig. 3. Plots of the constraint results with system (2) under controller (6).

VI. Application to Multirobot Multitarget Navigation Problem in Clustered Environment

The introduced framework, distributed continuous-time time-varying constrained optimization, is of great significance in motion coordination. In this section, we apply the proposed optimization algorithm (6) to a class of the motion coordination

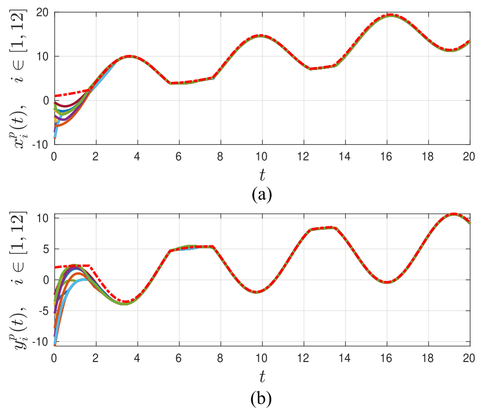

Fig. 4. State trajectories of all the agents with system (2) under controller (39). The red dashed line is the optimal solution and the other solid lines are the trajectories of all agents' states.

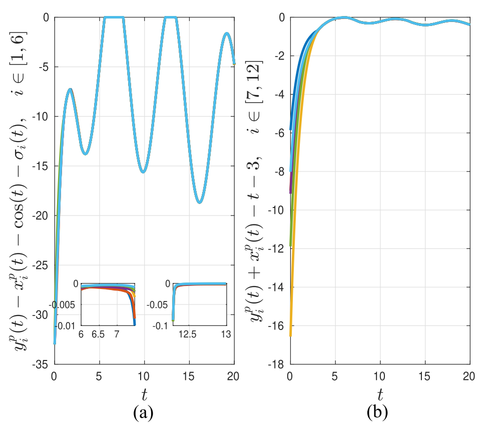

Fig. 5. Plots of the constraint results with system (2) under controller (39).

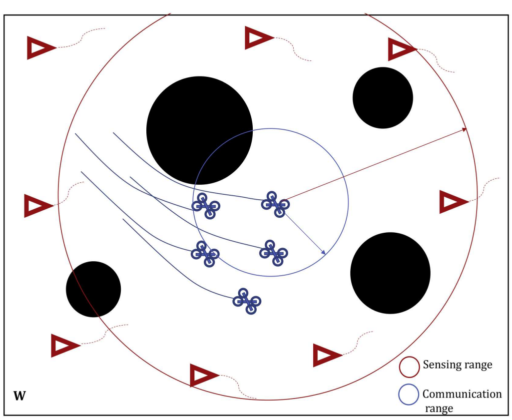

Fig. 6. Multirobot multitarget navigation problem.

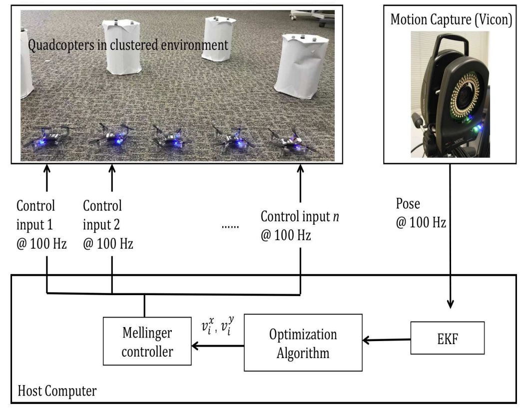

Fig. 7. Information flow in our multirobot multitarget navigation experiments.

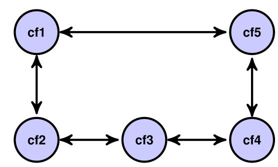

Fig. 8. Communication topology between crazyflies.

problems: the multirobot multitarget navigation problem. As shown in Fig. 6, let us consider a closed and convex workspace $W \in \mathbb{R}^2$. Consider the scenario where there are n disk-shaped robots (blue quadrotors) with center positions $x_i, i \in [1, \ldots, n]$ and radius $r_i > 0, i \in [1, \ldots, n]$ and k moving targets (red triangles) in an unknown space having obstacles inside. The objective here is to have the robots stay close while simultaneously ensuring that each independent moving target stays in the detection range of at least one robot. Assume that the workspace is populated with Q nonintersecting spherical obstacles (black circles), where the center and radius of the ith obstacle are denoted by $o_i \in W$ and $r_i^o > 0$, respectively. Since there are unknown obstacles in the environment, we have to guarantee no collisions during the tracking process.

We define the so-called collision-free local workspace around x_i as [27]

$$LF(x_i) = \left\{p \in W : a_j(x_i)^T p - b_j(x_i) \leq 0, j = 1, \ldots, Q\right\} \tag{52}$$

where

$$a_j(x_i) = o_j - x_i, \quad \theta_j(x_i) = \frac{1}{2} - \frac{r_j^{o2} - r_i^2}{2\|o_j - x_i\|^2},$$
$$b_j(x_i) = (o_j - x_i)^T \left[\theta_j o_j + (1 - \theta_j) x_i + r_i \frac{x_i - o_j}{\|x_i - o_j\|}\right]. \tag{53}$$

In order to have the robots stay close while simultaneously ensuring that each target stays in the sensing range of at least one robot, one method is to let all the robots assemble in the geometric center of all the targets with deviation vectors introduced to each robot. We tackle the navigation task by solving the following optimization problem with nonlinear inequality constraints:

$$\begin{aligned} \min \quad & \sum_i^n f_i = \|x_i - T_i(t)\|_2^2 \\ \text{s.t.} \quad & x_i = x_k \quad \forall i, k \in \mathcal{V}, \\ & a_j(x_i)^T x_i - b_j(x_i) \leq 0, \quad \forall i \in \mathcal{V}, \ j \in [1, \ldots, q_i] \end{aligned} \tag{54}$$

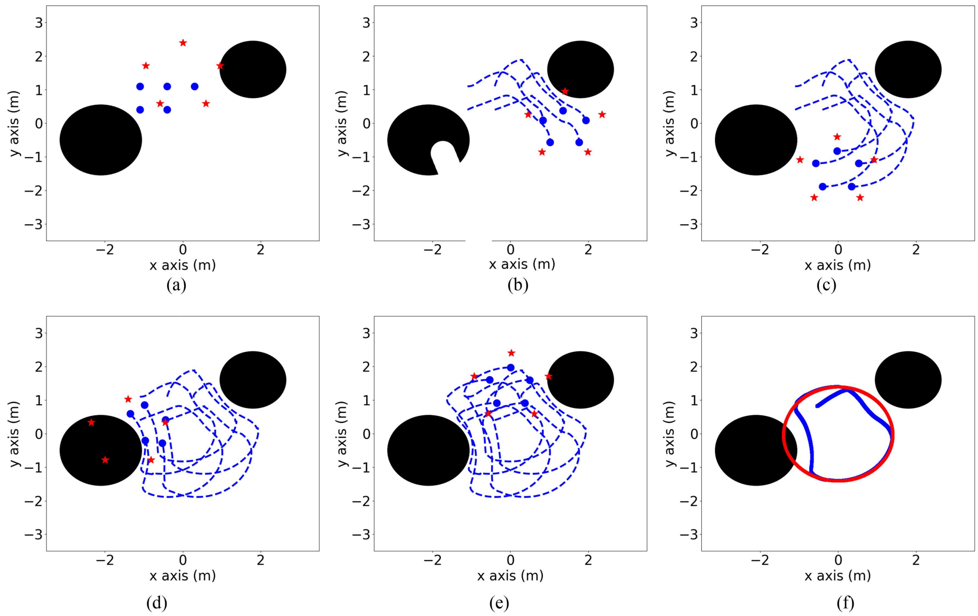

Fig. 9. Simulation result with Crazyswarm simulator. (a) Initial positions of all the crazyflies (blue circles) and all the targets (red stars). (b)–(e) Trajectories of all the crazyflies up to time instances 25, 50, 75, and 100 s. The positions of all the crazyflies and all the targets at each time instance are represented by blue circles (crazyflies) and red stars (targets). (f) Geometric center trajectory of all the crazyflies (blue line) and the geometric center trajectory of all the targets (red line).

where $T_i(t)$ is the geometric center of all the moving targets that robot i can sense and q_i is the number of obstacles that robot i can sense. Note that a robot might not be able to sense all the Q obstacles in the workspace, but it is safe enough to stay in the collision-free area determined by the nearby obstacles. Since $a_j(x_i)$ and $b_j(x_i)$ depend on the position of robot i, the above optimization problem has an implicit dependence on time through x_i. However, it is very hard to directly address the inequality constraints in (54) due to the complexity of $a_j(x_i)$ and $b_j(x_i)$ given by (53). Therefore, here, we treat $a_j(x_i)$ and $b_j(x_i)$ as $a_j(t)$ and $b_j(t)$. Based on (7), the corresponding penalized objective function is defined as $\tilde{L}_i = f_i(x_i, t) - \frac{1}{\rho_i(t)} \sum_{j=1}^{q_i} \log\{\sigma_i(t) - a_j(t)^T x_i + b_j(t)\}$. If the communication topology between the robots is undirected and connected, problem (54) satisfies all the Assumptions 1–5 in Theorem 1. Therefore, for robots with single-integrator dynamics defined by (2), the proposed constrained optimization algorithm (6) can be applied to reach on agreement at the geometric center of the targets and spread the robots in a desired formation about this center. Therefore, we introduce an offset vector δ_i for each robot i and replace x_i in algorithms (6) with $x_i - \delta_i$. Here, $\delta_i - \delta_j$ defines the desired relative position from robot j to robot i in the formation.

Our proposed algorithm is tested in the experiment with five Crazyflies 2.1 quadrotors [41] in an indoor environment. The experimental setup is shown in Fig. 7. We consider five quadrotors moving in 2-D space controlled by velocity commands $[v_i^x, v_i^y]$. Therefore, all the Crazyflies follow the single-integrator dynamics given by (2). We use the Vicon positioning system [42] coupled with the extended Kalman filter to estimate their positions $[x_i^p, y_i^p]$. Here, the control system is divided into two parts, namely, high level and low level. The high-level control involves the setup of the network topology, calculation of the targets' positions and velocities, capture of the obstacles' positions, implementation of the distributed constrained optimization algorithm, and generation of the velocity commands $[v_i^x, v_i^y]$. The low-level control is responsible for achieving the velocity commands (by using the Mellinger controller [40]).

The host computer is used to run the high-level controller because the crazyflies used in the experiments do not have sufficient computation capability to run the controller in real time. However, it should be noted that the restrictions of a distributed environment are fully considered and the defined distributed network topology is emulated. Five nodes under the robotics operating system are established to control the five crazyflies in parallel. The communication topology between the crazyflies is shown in Fig. 8.

In our experiment, a 5×5 m^2 area is used to implement the experiment. To simplify the experiment, we assume that each crazyflies is only assigned one target moving in the environment. Note that our algorithm still works for multiple targets since we only care about the geometric center of all the targets that the crazyflie can sense. The obstacles are located at $o_1 = [-2.1 \text{ m}, -0.5 \text{ m}]$ and $o_2 = [1.8 \text{ m}, 1.6 \text{ m}]$ with radius $r_1^0 = 0.9$ m, and $r_2^0 = 0.7$ m. Each crazyflie is able to sense an obstacle if any point of the obstacle falls into the circle with the center being the crazyflie position and the radius being 1.0 m. The offset vectors are chosen as

$$\delta_1 = [0.2\sin(0.2\pi) \text{ m}, -0.2\cos(0.2\pi) \text{ m}]^T,$$

$$\delta_2 = [-0.2\sin(0.2\pi) \text{ m}, -0.2\cos(0.2\pi) \text{ m}]^T,$$

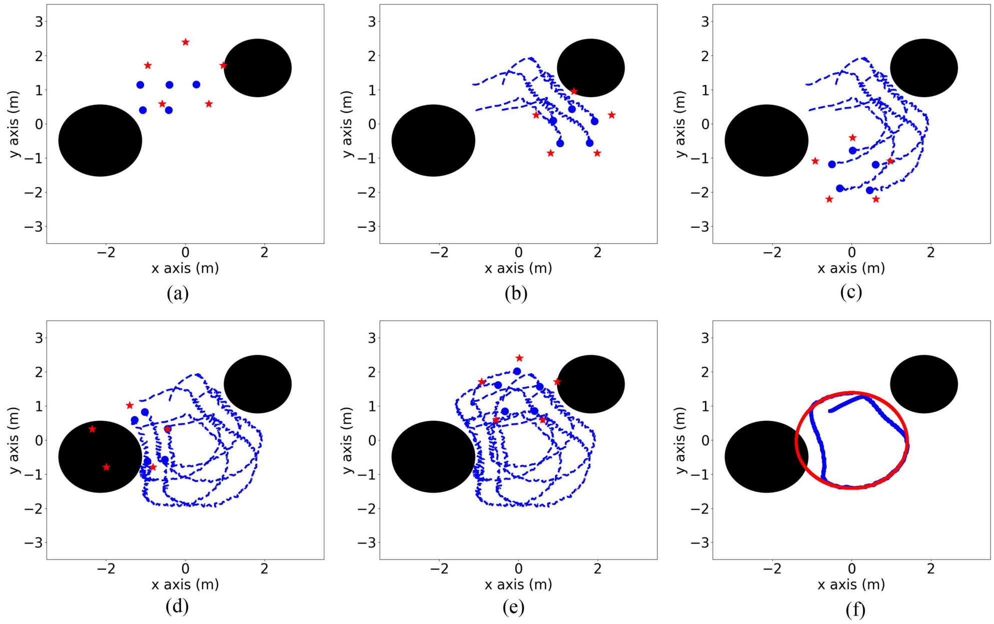

Fig. 10. Experimental result with crazyflies. (a) Initial positions of all the crazyflies (blue circles) and all the targets (red stars). (b)–(e) Trajectories of all the crazyflies up to time instances 25, 50, 75, and 100 s. The positions of all the crazyflies and all the targets at each time instance are represented by blue circles (crazyflies) and red stars (targets). (f) Geometric center trajectory of all the crazyflies (blue line) and the geometric center trajectory of all the targets (red line).

$$\delta_3 = [-0.2\cos(0.1\pi)\text{ m}, 0.2\sin(0.1\pi)\text{ m}]^T,$$
$$\delta_4 = [0\text{ m}, 0.2\text{ m}]^T,$$
$$\delta_5 = [0.2\cos(0.1\pi)\text{ m}, 0.2\sin(0.1\pi)\text{ m}]^T.$$

The initial positions of the five crazyflies are chosen as $x_1(0) = [-0.4\text{ m}, 0.4\text{ m}]$, $x_2(0) = [-1.1\text{ m}, 0.4\text{ m}]$, $x_3(0) = [-1.1\text{ m}, 1.1\text{ m}]$, $x_4(0) = [-0.4\text{ m}, 1.1\text{ m}]$, and $x_5(0) = [0.3\text{ m}, 1.1\text{ m}]$. We choose $\rho_i(t) = 125\exp(0.01t)$, $\sigma_i(t) = \exp(-t)$, and $\beta = 5$. The trajectories of the crazyflies in the Crazyswarm simulator [41] and in the experiment are, respectively, shown in Figs. 9 and 10. In both figures, the black circles are obstacles and the blue lines are the trajectories of the crazyflies. Subplots (a)–(e) show the trajectories of all the crazyflies up to time instances 0, 25, 50, 75, and 100 s. In addition, five snapshots at 0, 25, 50 s, 75, and 100 s denoted by the red stars (targets) and blue circles (crazyflies) are shown in subplots (a)–(e). It is obvious that all the crazyflies assemble together and avoid obstacles successfully both in the Crazyswarm simulator and real experiment. Subplot (f) shows the trajectories of the geometric center of all the crazyflies (blue line) and all the targets (red line). In the Crazyswarm simulator, the geometric center of all crazyflies are able to track the geometric center of all the targets with zero tracking error which are consistent with Theorem 1. In our experimental result, the crazyflies tremble slightly in flight and the geometric center of all crazyflies are able to track the geometric center of all the targets with small tracking error (about 0.001 m). It is worthwhile to mention that the trembling phenomena and tracking error in the experiment might stem from the time-delay of communication with the Vicon system and failure of achieving the velocity commands accurately.

VII. Conclusion

In this article, we have studied the distributed continuous-time constrained optimization problem with time-varying objective functions and time-varying constraints. The goal is for a set of networked agents to cooperatively track the time-varying optimal solution that minimizes the summation of all the local time-varying objective functions subject to all the local time-varying constraints, where each agent has only local information and local interaction. We have proposed distributed sliding-mode algorithms built on the Hessian-based optimization methodology. We have shown that asymptotical convergence to the optimal solution or its vicinity is guaranteed under some reasonable assumptions. Both numerical simulation results and experimental results are given to illustrate the theoretical algorithm.

References

[1] D. K. Molzahn et al., “A survey of distributed optimization and control algorithms for electric power systems,” *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.
[2] R. Zhang and J. Kwok, “Asynchronous distributed ADMM for consensus optimization,” in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 1701–1709.
[3] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Proc. Int. Symp. Inf. Process. Sensor Netw.*, Berkeley, CA, USA, 2004, pp. 20–27.
[4] M. T. Shahab and M. Elshafei, “Distributed optimization of multi-robot motion with time-energy criterion,” *Path Plan. Auton. Veh.*, IntechOpen, 2019.
[5] H. Lakshmanan and D. P. De Farias, “Decentralized resource allocation in dynamic networks of agents,” *SIAM J. Optim.*, vol. 19, no. 2, pp. 911–940, Sep. 2008.
[6] T. Yang et al., “A survey of distributed optimization,” *Annu. Rev. Control*, vol. 47, pp. 278–305, 2019.
[7] 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.

[8] P. Lin, W. Ren, and J. A. Farrell, "Distributed continuous-time optimization: Nonuniform gradient gains, finite-time convergence, and convex constraint set," *IEEE Trans. Autom. Control*, vol. 62, no. 5, pp. 2239–2253, May 2017.

[9] Z. Li, Z. Wu, Z. Li, and Z. Ding, "Distributed optimal coordination for heterogeneous linear multi-agent systems with event-triggered mechanisms," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1763–1770, Apr. 2020.

[10] Z. Qiu, S. Liu, and L. Xie, "Distributed constrained optimal consensus of multi-agent systems," *Automatica*, vol. 68, pp. 209–215, Jun. 2016.

[11] J. Lu and C. Tang, "Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2348–2354, Sep. 2012.

[12] Z. Feng and G. Hu, "Finite-time distributed optimization with quadratic objective functions under uncertain information," in *Proc. IEEE Conf. Decis. Control*, Melbourne, Australia, 2017, pp. 208–213.

[13] S. Liang, X. Zeng, and Y. Hong, "Distributed nonsmooth optimization with coupled inequality constraints via modified Lagrangian function," *IEEE Trans. Autom. Control*, vol. 63, no. 6, pp. 1753–1759, Jun. 2018.

[14] Y. Zhu, W. Yu, G. Wen, G. Chen, and W. Ren, "Continuous-time distributed subgradient algorithm for convex optimization with general constraints," *IEEE Trans. Autom. Control*, vol. 64, no. 4, pp. 1694–1701, Apr. 2019.

[15] S. Yang, Q. Liu, and J. Wang, "A multi-agent system with a proportional-integral protocol for distributed constrained optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3461–3467, Jul. 2017.

[16] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 489–493, Apr. 2015.

[17] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.

[18] W. Su, "Traffic engineering and time-varying convex optimization," Ph.D dissertation, Dept. Electrical Eng., Pennsylvania State Univ., State College, PA, USA 2009.

[19] Y. Tang, "Time-varying optimization and its application to power system operation," Ph.D dissertation, Dept. Electrical Eng., California Inst. Technol., Pasadena, CA, USA, 2019.

[20] C. Sun, M. Ye, and G. Hu, "Distributed time-varying quadratic optimization for multiple agents under undirected graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3687–3694, Jul. 2017.

[21] H. Mehdi, E. Garone, and L. Schenato, "A distributed method for linear programming problems with box constraints and time-varying inequalities," *IEEE Control Syst. Lett.*, vol. 3, no. 2, pp. 404–409, Apr. 2019.

[22] M. Ye and G. Hu, "Distributed seeking of time-varying Nash equilibrium for non-cooperative games," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 3000–3005, Nov. 2015.

[23] C. Ashish and J. Cortes, "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment," *Automatica*, vol. 74, pp. 183–193, Dec. 2016.

[24] B. Wang, S. Sun, and W. Ren, "Distributed continuous-time algorithms for optimal resource allocation with time-varying quadratic cost functions," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 4, pp. 1974–1984, Dec. 2020.

[25] S. Rahili and W. Ren, "Distributed continuous-time convex optimization with time-varying cost functions," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1590–1605, Apr. 2017.

[26] B. Huang, Y. Zou, Z. Meng,, and W. Ren, "Distributed time-varying convex optimization for a class of nonlinear multi-agent systems," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 801–808, Feb. 2020.

[27] M. Fazlyab, S. Paternain, V. M. Preciado,, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1973–1986, Jul. 2018.

[28] F. Chen, Y. Cao, and W. Ren, "Distributed average tracking of multiple time-varying reference signals with bounded derivatives," *IEEE Trans. Autom. Control*, vol. 57, no. 12, pp. 3169–3174, Dec. 2012.

[29] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *IEEE Trans. Signal Process.*, vol. 68, pp, 731–746, 2020.

[30] A. Simonetto, A. Koppel, A. Mokhtari, G. Leus, and A. Ribeiro, "Decentralized prediction-correction methods for networked time-varying convex optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5724–5738, Nov. 2017.

[31] J. Wang and E. Nicola, "A control perspective for centralized and distributed convex optimization," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf.*, Orlando, FL, USA, 2011, pp. 3800–3805.

[32] S. Sun and W. Ren, "Distributed continuous-time optimization with time-varying objective functions and inequality constraints," in *Proc. IEEE Conf. Decis. Control*, Jeju Island, Republic of Korea, 2020, pp. 5622–5627.

[33] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39, no. 9, pp. 1910–1914, Sep. 1994.

[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[35] A. F. Filippov, *Differential Equations With Discontinuous Righthand Sides*. Dordrecht, The Netherlands: Kluwer Academic, 1988.

[36] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Syst. Mag.*, vol. 28, no. 3, pp. 36–73, Jun. 2008.

[37] H. K. Khalil and W. G. Jessy, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[38] T. M. Apostol, *Mathematical Analysis*. Boston, MA, USA: Addison-Wesley, 1974.

[39] F. Chen, W. Ren, W. Lan, and G. Chen, "Distributed average tracking for reference signals with bounded accelerations," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 863–869, Mar. 2015.

[40] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Automat.*, Shanghai, China, 2011, pp. 2520–2525.

[41] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *Proc. IEEE Int. Conf. Robot. Automat.*, Singapore, Singapore, 2017, pp. 3299–3304.

[42] Vicon motion capture system, Accessed: Jun. 2017. [Online]. Available: https://www.vicon.com

Shan Sun received the B.S. and M.S. degrees in navigation, guidance and control from the School of Astronautics, Beihang University, Beijing, China, in 2014 and 2017, respectively. She is currently working toward the Ph.D. degree in electrical engineering with the University of California, Riverside, CA, USA.

Her research focuses on distributed optimization and multiagent systems.

Jie Xu received the B.S. degree in mechanical engineering and automation from Huazhong University of Science and Technology, Wuhan, China, in 2018, and the M.S. degree in mechanical engineering from the University of California, Riverside, CA, USA, in 2020. He is currently working toward the Ph.D. degree in electrical engineering with the University of California, Riverside, CA, USA.

His research interests include autonomous control on robotics and cooperative control on multiagent systems.

Wei Ren (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Brigham Young University, Provo, UT, USA, in 2004.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA.

Dr. Ren was a recipient of the IEEE Control Systems Society Antonio Ruberti Young Researcher Prize in 2017 and the National Science Foundation CAREER Award in 2008. He is an IEEE Control Systems Society Distinguished Lecturer.