Asynchronous Distributed Optimization With Event-Driven Communication

Minyi Zhong, *Student Member, IEEE*, and Christos G. Cassandras, *Fellow, IEEE*

***Abstract*—We consider problems where multiple agents cooperate to control their individual state so as to optimize a common objective while communicating with each other to exchange state information. Since communication costs can be significant, especially when the agents are wireless devices with limited energy, we seek conditions under which communication of state information among nodes can be restricted while still ensuring that the optimization process converges. We propose an asynchronous (event-driven) optimization scheme that limits communication to instants when some state estimation error function at a node exceeds a threshold and prove that, under certain conditions, such convergence is guaranteed when communication delays are negligible. We subsequently extend the analysis to include communication delays as long as they are bounded. We apply this approach to a sensor network coverage control problem where the objective is to maximize the probability of detecting events occurring in a given region and show that the proposed asynchronous approach may significantly reduce communication costs, hence also prolonging the system's lifetime, without any performance degradation.**

***Index Terms*—Asynchronous optimization, cooperative control, distributed optimization, distributed systems, sensor networks.**

I. Introduction

The need for distributed optimization arises in settings which involve multiple controllable agents cooperating toward a common objective without a central controller to coordinate their actions. The cooperating agents define a dynamic system which may be thought of as a network with each agent corresponding to a node maintaining its own state s_i, $i = 1, \dots, N$. The goal of each node is to control its state so as to optimize some system-wide objective expressed as a function of $\mathbf{s} = [s_1, \dots, s_N]$ and possibly the state of the environment. Clearly, to achieve such a goal in a dynamic and uncertain environment, the nodes must share, at least partially, their state information. However, this may require a large amount of information flow and becomes a critical issue when the system consists of wirelessly communicating nodes which are often small, inexpensive devices with limited resources (e.g., a sensor network). Aside from energy required to move (if nodes are mobile), communication is known to be by far the largest consumer of the limited energy of a node [1], compared to other functions such as sensing and computation. Moreover, every communication among nodes offers an opportunity for corruption or loss of information due to random effects or adversarial action. Therefore, it is crucial to reduce communication between nodes to the minimum possible. This in turn imposes a constraint on the optimization task performed by each node, since it requires that actions be taken without full knowledge of other nodes' states. Standard synchronization schemes require that nodes exchange state information frequently, sometimes periodically, which can clearly be inefficient and, in fact, often unnecessary since it is possible that: (*i*) system inactivity makes the periodic (purely time-driven) exchange of information unnecessary, (*ii*) occasional state information is adequate for control and/or optimization mechanisms which do not require perfect accuracy at all times, (*iii*) the state information of other nodes can be estimated reasonably well and explicit communication is thus redundant. This motivates us to seek *asynchronous* optimization mechanisms in which a node communicates with others only when it considers it indispensable; in other words, each node tries to reduce the cost of communication by transmitting state information only under certain conditions and only as a last resort. This poses questions such as "what should the conditions be for a node to take such communication actions?" and "under what conditions, if any, can we guarantee that the resulting optimization scheme possesses desirable properties such as convergence to an optimum?"

The general setting described above applies to problems where the nodes may be vehicles controlling their locations and seeking to maintain some desirable formation [2], [3] while following a given trajectory. The system may also be a sensor network whose nodes must be placed so as to achieve objectives such as maximizing the probability of detecting events in a given region or maintaining a desired distance from data sources that ensures high-quality monitoring [4]–[11]; this is often referred to as a "coverage control" problem. In some cases, the state of a node may not be its location but rather its perception of the environment which changes based on data directly collected by that node or communicated to it by other nodes; consensus problems fall in this category [12]–[16].

In this paper, we consider a system viewed as a network of N cooperating nodes. The system's goal is to minimize an objective function $H(\mathbf{s})$ known to all nodes with every node controlling its individual state $s_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, N$. The state update scheme employed by the ith node is of the general form

$$s_i(k+1) = s_i(k) + \alpha_i d_i(\mathbf{s}(k)), \quad k = 0, 1, \dots \qquad (1)$$

Manuscript received January 14, 2009; revised June 17, 2009, January 07, 2010; accepted April 16, 2010. Date of publication May 03, 2010; date of current version December 02, 2010. This work was supported in part by the NSF under Grants DMI-0330171 and EFRI-0735974, by AFOSR under Grants FA9550-04-1-0133, FA9550-04-1-0208, FA9550-07-1-0361, and FA9550-09-1-0095, by DOE under Grant DE-FG52-06NA27490, and by ONR under Grant N00014-09-1-1051. Recommended by Senior Editor K. P. Wong.

The authors are with the Division of Systems Engineering, and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446 USA (e-mail: mzhong@bu.edu; cgc@bu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2010.2049518

where α_i is a constant positive step size and $d_i(\mathbf{s}(k))$ is an *update direction* evaluated at the kth *update event* (see also [17]). We often use

$$d_i(\mathbf{s}(k)) = -\nabla_i H(\mathbf{s}(k))$$

where $\nabla H(\mathbf{s}(k))$ is the gradient of $H(\mathbf{s}(k))$ and $\nabla_i H(\mathbf{s}(k)) \in \mathbb{R}^{n_i}$. In general, each state is characterized by dynamics of the form $\dot{s}_i(t) = f_i(s_i, u_i, t)$ where $u_i \in \mathbb{R}^l$ is a control vector; for our purposes, however, we treat s_i as a directly controllable vector. Thus, in (1) we view $s_i(k+1)$ as the *desired* state determined at the kth update event and assume that the control u_i is capable of reaching $s_i(k+1)$ from $s_i(k)$ within a time interval shorter than the time between update events.

A key difficulty in (1) is that $\mathbf{s}(k)$ is in fact not fully known to node i. Thus, $d_i(\mathbf{s}(k))$ has to be evaluated by synchronizing all nodes to provide their states to node i at the time its kth update event takes place. This is extremely costly in terms of communication and assumes no delays so that the state information is accurate. Alternatively, node i can evaluate $d_i(\mathbf{s}(k))$ using estimates of s_j for all $j \neq i$ relying on prior information from node j and possibly knowledge of its dynamics. Our concern is with determining instants when a node j may communicate its state to other nodes through what we term *communication events*. We note that such communication events occur at different times for each node, as do each node's state update events, so that the resulting mechanism is asynchronous. We also point out that a distributed optimization framework is necessitated by the fact that nodes operate in a dynamic and uncertain environment and each node generally has local information not available to others. An example is the appearance of a random obstacle in the coverage control setting mentioned earlier, detected only by a single node.

We propose a scheme through which a node j maintains an *error function* of its actual state relative to its state as estimated by other nodes (which node j can evaluate). The node then transmits its actual state at time t only if this error function at t exceeds a given *threshold* δ_j. In other words, a node does not incur any communication cost unless it detects that the deviation of its state from the other nodes' estimate of its state becomes too large; this may happen due to the normal state update (1) accumulating noise, imperfect state estimation or through unexpected state changes (e.g., if a mobile node encounters an obstacle). First, assuming negligible communication delays, we prove that by varying this threshold appropriately and under certain rather mild technical conditions the resulting optimization scheme converges and leads to a minimum of $H(\mathbf{s})$; this minimum may be local or global depending on the nature of the objective function. Our analysis is based on the distributed optimization framework in [17], but our emphasis is on controlling the asynchronous occurrence of communication events through the threshold-based scheme outlined above in a way that may drastically reduce the number of such events while still guaranteeing convergence. When an explicit noise term is included in (1), our analysis still leads to a similar convergence result under some additional conditions bounding this noise term. Subsequently, we allow communication delays to be non-negligible as long as there exists an upper bound in the number of state update events that occur between the time of a communication event initiated by a node and the time when all nodes receive the communicated message. This requires a modification in how communication events are generated. The resulting optimization mechanism is shown to still converge to a minimum of $H(\mathbf{s})$.

In the second part of the paper, we apply this approach to a coverage control problem in which a distributed optimization scheme based on (1) is used in order to deploy sensor nodes in a region (possibly containing polygonal obstacles) so as to maximize the probability of detecting events (e.g., unknown data sources) in this region. In earlier work [9] it was assumed that all nodes have perfect state information by synchronizing update events with communication events. This imposed significant communication costs. Here, we relax this synchronization requirement and limit communication events to occur according to the new proposed event-driven policy leading to convergence to the optimum. Simulation results are included to show that the same performance is attained with only a fraction of the original communication costs.

Our work on asynchronous distributed optimization with event-driven communication was introduced in [18] with no communication delays in the convergence analysis. Such event-driven communication is also used in collaborative estimation (as opposed to optimization) [19] where a node transmits data to other nodes only when a computable estimation error exceeds some fixed threshold. In [20], an augmented Lagrangian method based on event-triggered message passing is developed to obtain an approximate solution of the network utility maximization problem.

The remainder of the paper is organized as follows. Section II describes our asynchronous distributed optimization framework and the proposed scheme for communication events. The convergence analysis under negligible communication delays is presented in Section III, including the case of explicit noise present in (1). We subsequently extend it in Section IV to the case where communication delays are present. In Section V we show how our approach applies to a coverage control problem for sensor networks and we conclude with Section VI.

II. Asynchronous Distributed Optimization Framework

In a setting where N cooperating nodes seek to optimize a common objective function, there are two processes associated with each node: a *state update process* and a *state communication process*. We begin with a discussion of the state update process.

Let t_k, $k = 1, 2, \ldots$, denote the time when any one node performs a state update, i.e., it takes an action based on (1). We impose no constraint on when precisely such an update event occurs at a node and allow it to be either periodic or according to some local node-based policy. However, we will assume that every node performs an update with sufficient frequency relative to the updates of other nodes (this assumption will be stated precisely later).

Let $\mathcal{C}^i$ be the set of indices in $\{t_k\}$ corresponding to update events at node i. As an example, in Fig. 1, where nodes 1 and 2 perform state updates at $\{t_1, t_4, t_5\}$ and $\{t_2, t_3, t_6\}$ respectively, we have $\mathcal{C}^1 = \{1, 4, 5\}$, and $\mathcal{C}^2 = \{2, 3, 6\}$. We will set $d_i(\mathbf{s}(k)) = 0$ in (1) for all $k \notin \mathcal{C}^i$, i.e.

$$s_i(k+1) = s_i(k) \quad \text{if } k \notin \mathcal{C}^i.$$

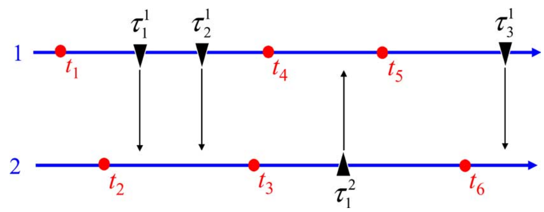


Fig. 1. State update and communication processes for two nodes. Dots represent state update events at a node and triangles represent state communication events.

We refer to any state update at such $k \notin \mathcal{C}^i$ as a *null step* at node i.

Next, let us discuss the state communication process. Let τ_n^j be the nth time when node j broadcasts its true state to all other nodes, $n = 1, 2, \ldots$ and $\tau_0^j = 0$ (a more general scheme which allows point-to-point communication will be introduced later in this section). Depending on the network connectivity at that time, it is possible that only a subset of nodes is reached. We assume that at all communication event times the state information broadcast by node j can reach any other node with negligible delay either directly or indirectly (in the latter case, through a sequence of transmissions), i.e., we assume that the underlying network is connected. The negligible communication delay assumption will be relaxed in Section IV. The network connectivity assumption is preserved throughout the paper, although it may be replaced by weaker conditions when the network has nodes that can be decoupled, i.e., the local decision of a node is not affected by the state information of some other nodes in the network. For the range of applications that motivate our work, there is usually a "base station" which maintains communication with all nodes in the network, thus ensuring full connectivity. However, the existence of such a base station is not required for the general optimization framework we propose. If connectivity is lost by some node, then a related problem is to ensure that the node (if mobile) can control its state so as to re-establish such connectivity. This problem is the subject of ongoing research and is not addressed in this paper.

Let us now consider a state update time t_k at node i, i.e., $k \in \mathcal{C}^i$. We are interested in the most recent communication event from a node $j \neq i$ and define

$$\tau^j(k) = \max\left\{\tau_n^j : \tau_n^j \le t_k,\ n = 0, 1, 2, \ldots\right\} \quad (2)$$

as the time of the most recent communication event at node j up to a state update event at t_k. As an example, in Fig. 1 node 1 communicates its state to node 2 twice in the interval (t_2, t_3); in this case, $\tau^1(3) = \tau_2^1$. However, no further communication event takes place from node 1 until after the next state update event at node 2 at time t_6, so that $\tau^1(6) = \tau^1(3) = \tau_2^1$. The policy used by node j to generate communication events is crucial and will be detailed later in this section, but we emphasize that it is in no way constrained to be synchronized with update events or with the communication events of any other node.

In order to differentiate between a node state at any time t and its value at the specific update times t_k, $k = 0, 1, \ldots$, we use $x_i(t)$ to denote the former and observe that

$$s_i(k) = x_i(t_k).$$

Thus, the state of node j communicated to other nodes at time $\tau^j(k)$ may be written as $x_j(\tau^j(k))$. Returning to the state update process, consider some t_k with $k \in \mathcal{C}^i$, and let $\mathbf{s}^i(k)$ be a vector with node i's estimates of all node states at that time, i.e., an estimate of $\mathbf{s}(k)$. There are various ways for node i to estimate the state of some $j \neq i$. The simplest is to use the most recent state information received at time $\tau^j(k)$ as defined in (2), i.e.

$$s_j^i(k) = x_j\left(\tau^j(k)\right). \quad (3)$$

Alternatively, node i may use a dynamic linear estimate of the form

$$s_j^i(k) = x_j\left(\tau^j(k)\right) + \frac{t_k - \tau^j(k)}{\Delta_j} \cdot \alpha_j \cdot d_{j,\tau^j(k)} \quad (4)$$

where Δ_j is an estimate of the average time between state updates at node j (e.g., a known constant if node j performs periodic updates) and $d_{j,\tau^j(k)}$ is the update direction communicated by node j at time $\tau^j(k)$ along with its state. Note that $[t_k - \tau^j(k)]/\Delta_j$ is an estimate of the number of state updates at j since its last communication event. More generally, if the precise local decision making process of j is known to i, then i can evaluate $s_j^i(k)$ using this information with initial condition $x_j(\tau^j(k))$. In this case, the estimate is error-free except for noise that may have affected the actual state evolution of node j in the interval $[\tau^j(k), t_k]$. In general, the value of an estimate $s_j^i(k)$ used by node i to estimate node j's state depends on t_k, the most recent communication event time $\tau^j(k)$, and the actual state $x_j(\tau^j(k))$ of node j at that time.

Now let us consider what criterion a node i might use to generate its communication events, recalling that we aim to reduce communication costs. If node i knows that node j uses a specific method to estimate its state, then node i can evaluate that estimate and hence the error in it at any time. If $x_i^j(t)$ is the estimate of $x_i(t)$ evaluated by node $j \neq i$ at time t, we can define an estimation *error function* $g(x_i(t), x_i^j(t))$, which measures the quality of the state estimate of node i with the requirement that

$$g\left(x_i(t), x_i^j(t)\right) = 0 \quad \text{if } x_i(t) = x_i^j(t). \quad (5)$$

Examples of $g(x_i(t), x_i^j(t))$ include $\|x_i(t) - x_i^j(t)\|_1$ and $\|x_i(t) - x_i^j(t)\|_2$. Let $\delta_i(k)$ be an error *threshold*, determined by node i after the kth state update event such that $k \in \mathcal{C}^i$. Thus, $\delta_i(k) = \delta_i(k-1)$ if $k \notin \mathcal{C}^i$. Let $\tilde{k}_t^i$ be the index of the most recent state update time of node i up to t, i.e.

$$\tilde{k}_t^i = \max\{n : n \in \mathcal{C}^i, t_n \le t\}. \quad (6)$$

If different nodes use different means to estimate i's state, then generally $x_i^j(t) \neq x_i^k(t)$ for nodes $j \neq k$ and communication may be limited to a node-to-node process. Let τ_n^{ij} be the nth time when node i sends its true state to node j. Let us also set $\tau_0^{ij} = 0$ for all i, j. Then, the communication event policy at node i with respect to node j is determined by

$$\tau_n^{ij} = \inf\left\{t : g\left(x_i(t), x_i^j(t)\right) \ge \delta_i\left(\tilde{k}_t^i\right),\ t > \tau_{n-1}^{ij}\right\}. \quad (7)$$

When a communication event is triggered by (7) at τ_n^{ij}, assuming negligible communication delay, x_i^j is instantaneously

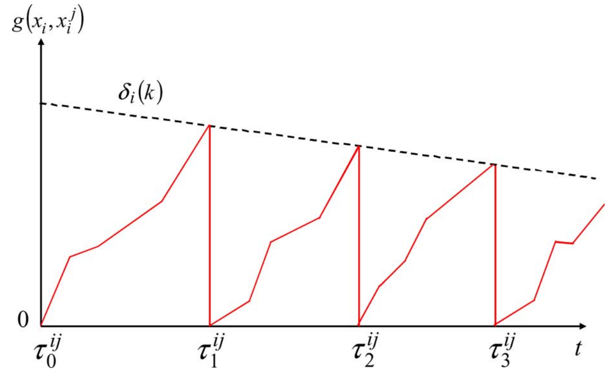


Fig. 2. Trajectory of the error function $g(x_i(t), x_i^j(t))$ when the communication delay is negligible.

set to $x_i(\tau_n^{ij})$, i.e., $x_i^j((\tau_n^{ij})^+) = x_i(\tau_n^{ij})$. Therefore, the error measure is reset to zero, i.e., $g(x_i((\tau_n^{ij})^+), x_i^j((\tau_n^{ij})^+)) = 0$.

If, on the other hand, all nodes use the exact same estimation method, then $x_i^j(t)$ will have the same value for all $j \neq i$ and we can replace τ_n^{ij} in (7) by τ_n^i. In other words, node i communicates its state to all other nodes only when it detects that its true state deviates from the other nodes' estimate of it by at least the threshold $\delta_i(\tilde{k}_t^i)$. Fig. 2 shows an example of a trajectory of $g(x_i(t), x_i^j(t))$ where the threshold is decreasing over time and the error function is continuous over all intervals $(\tau_{n-1}^{ij}, \tau_n^{ij})$, $n = 1, 2, \ldots$ Observe that the negligible communication delay assumption allows the value of $g(x_i(t), x_i^j(t))$ to be instantaneously reset to 0 at τ_n^{ij}, $n = 1, 2, \ldots$

Next, we discuss the way in which the threshold $\delta_i(k)$ should be selected. The basic idea is to use a large value at the initial stages of the optimization process and later reduce it to ultimately ensure convergence. One of the difficulties is in selecting an appropriate initial value for $\delta_i(k)$ which, if too large, may prevent any communication. The approach we follow is to control $\delta_i(k)$ in a manner which is proportional to $\|d_i(\mathbf{s}^i(k))\|_2$, the Euclidean norm of the update direction at the kth update event henceforth denoted by $\|\cdot\|$. Thus, let

$$\delta_i(k) = \begin{cases} K_\delta \left\|d_i\left(\mathbf{s}^i(k)\right)\right\| & \text{if } k \in \mathcal{C}^i \\ \delta_i(k-1) & \text{otherwise} \end{cases} \quad (8)$$

where K_δ is a positive constant. We also impose an initial condition such that

$$\delta_i(0) = K_\delta \left\|d_i\left(\mathbf{s}^i(0)\right)\right\|, \quad i = 1, \ldots, N \quad (9)$$

and

$$s_j^i(0) = x_j(0). \quad (10)$$

Note that (10) can be readily enforced by requiring all nodes to share their initial states at the beginning of the optimization process, i.e., $\tau_0^{ij} = 0$ for all i, j; since generally $g(x_i(0), x_i^j(0)) \geq \delta_i(0)$, this triggers (7) and results in $g(x_i((0)^+), x_i^j((0)^+)) = 0$ for all $i = 1, \ldots, N$. Also note that since $\mathbf{s}^i(k)$ is node i's local estimate of $\mathbf{s}(k)$ at t_k, the computation in (8) requires only local information.

Finally, we point out that the quality of the estimation method used may be critical for effective communication reduction in our event-driven optimization framework. Poor quality state estimation can potentially trigger communication events at a frequency that is higher than that of state updates, especially close to convergence when nodes have small $d_i(\mathbf{s}^i)$. Interestingly, when the simplest method, i.e., static estimation, is used this issue does not arise because triggering estimation errors coincide with state update events. Therefore, a node will communicate its real state at most as often as it updates its state.

III. Convergence Analysis

In this section, we study the convergence properties of the asynchronous distributed state update scheme, for $k = 0, 1, \ldots$

$$s_i(k+1) = s_i(k) + \alpha d_i\left(\mathbf{s}^i(k)\right) \quad (11)$$

used by nodes $i = 1, \ldots, N$, where $d_i(\mathbf{s}^i(k))$ is an update direction which satisfies $d_i(\mathbf{s}^i(k)) = 0$ for all $k \notin \mathcal{C}^i$. For simplicity, a common step size α is used, but each node may easily adjust its step size by incorporating a scaling factor into its own $d_i(\mathbf{s}^i(k))$. Recall that $\mathbf{s}^i(k)$ is the state estimate vector evaluated by node i at the kth update event using the most recent state updates from other nodes at times $\tau^j(k)$ defined by (2). We will follow the framework in [17]. The distinctive feature in our analysis is the presence of the controllable state communication process defined by (7), (8) and (9) which imposes a requirement on the constant K_δ in order to guarantee convergence. Further, our analysis gives us means to select this constant in conjunction with the step size parameter α in (11) in a way that may potentially drastically reduce the number of communication events while still guaranteeing convergence.

We begin with a number of assumptions, most of which are commonly used in the analysis of distributed asynchronous algorithms [17]. Recall that, for the time being, we assume that the communication delay is negligible whenever a node informs other nodes of its current states.

Assumption 1: There exists a positive integer B such that for every $i = 1, \ldots, N$ and $k \geq 0$ at least one of the elements of the set $\{k - B + 1, k - B + 2, \ldots, k\}$ belongs to $\mathcal{C}^i$.

This assumption imposes a bound on the state update frequency of every node in order to guarantee that the entire state vector will be iterated on. It does not specify a bound in time units but rather ensures that each node updates its state at least once during a period in which B state update events take place. We point out that an update event time t_k may correspond to more than one node performing updates.

Assumption 2: The objective function $H(\mathbf{s})$, where $\mathbf{s} \in \mathbb{R}^m$, $m = \sum_{i=1}^N n_i$, satisfies the following:

(a) $H(\mathbf{s}) \geq 0$ for all $\mathbf{s} \in \mathbb{R}^m$

(b) $H(\cdot)$ is continuously differentiable and $\nabla H(\cdot)$ is Lipschitz continuous, i.e., there exists a constant K_1 such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, $\|\nabla H(\mathbf{x}) - \nabla H(\mathbf{y})\| \leq K_1 \|\mathbf{x} - \mathbf{y}\|$.

In what follows, we shall take all vectors to be column vectors and use $'$ to denote a transpose. Let

$$d(k) = \left[d_1\left(\mathbf{s}^1(k)\right)', \ldots, d_N\left(\mathbf{s}^N(k)\right)'\right]'.$$

For simplicity, we will henceforth write $d_i(k)$ instead of $d_i(\mathbf{s}^i(k))$.

Assumption 3: There exist positive constants K_2 and K_3 such that for all $i = 1, \ldots, N$ and $k \in \mathcal{C}^i$, we have

(a) $d_i(k)'\nabla_i H(\mathbf{s}^i(k)) \leq -\|d_i(k)\|^2/K_3$

(b) $K_2\|\nabla_i H(\mathbf{s}^i(k))\| \leq \|d_i(k)\|$

Here $\nabla_i H(\mathbf{s}^i(k))$ denotes a vector with dimension n_i. Its jth component, denoted by $\nabla_{i,j} H(\mathbf{s}^i(k))$, is the partial derivative of $H(\cdot)$ with respect to the jth component of s_i. This assumption is very mild and is immediately satisfied with $K_2 = K_3 = 1$ when we use an update direction given by $d_i(k) = -\nabla_i H(\mathbf{s}^i(k))$.

Assumption 4: The error function $g(x_i(t), x_i^j(t))$ satisfies the following:

(a) There exists a positive constant K_4 such that $\|x_i(t) - x_i^j(t)\| \leq K_4 g(x_i(t), x_i^j(t))$ for all i, j, t

(b) $g(x_i(t), x_i^j(t)) \leq \delta_i(\tilde{k}_t^i)$ where $\tilde{k}_t^i$ was defined in (6).

In the common case where $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|$, part (a) of this assumption is obviously satisfied with $K_4 = 1$. Part (b) may be violated if there is a discontinuity when $g(x_i(t), x_i^j(t))$ exceeds $\delta_i(\tilde{k}_t^i)$, in which case we can simply redefine the error function to take the value $\delta_i(\tilde{k}_t^i)$ at all such points, i.e., we use instead an error function $\bar{g}(\cdot)$ such that $\bar{g}(x_i(t), x_i^j(t)) = \delta_i(\tilde{k}_t^i)$ if $g(x_i(t), x_i^j(t)) > \delta_i(\tilde{k}_t^i)$, otherwise $\bar{g}(\cdot) = g(\cdot)$. Alternatively, observe that if such a discontinuity occurs at t, then, by (7), node i generates a communication event and, under the negligible delay assumption, node i receives node j's state and sets $x_i^j(t) = x_i(t)$, hence $g(x_i(t), x_i^j(t)) = 0$ by (5); that is, the error is instantaneously reset to zero. Therefore, defining $g(\cdot)$ to be right-continuous guarantees that part (b) is always satisfied. With this understanding, we will continue to use the notation $g(x_i(t), x_i^j(t))$ and assume this assumption is satisfied throughout this section (Assumption 4(b) will be relaxed in the next section).

Theorem 1: Under Assumptions 1–4, the communication event policy (7), and the state update scheme (11), if the error threshold $\delta_i(k)$ controlling communication events is set by (8), (9), then there exist positive constants α and K_δ such that $\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$.

Proof: The main part of the proof consists of bounding $H(\mathbf{s}(k+1))$ by a sum of terms and showing that, at each step k, $H(\mathbf{s}(k))$ decreases by a quantity proportional to $\|d(k)\|$. By the nonnegativity of $H(\cdot)$, we can conclude that $\|d(k)\|$ eventually vanishes and so does $\nabla H(\mathbf{s}(k))$. We begin by using a result known as the "descent lemma" (see Prop. A.32 in [17]) stating that under Assumption 2 the following inequality holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$:

$$H(\mathbf{x}+\mathbf{y}) \leq H(\mathbf{x}) + \mathbf{y}'\nabla H(\mathbf{x}) + \frac{K_1}{2}\|\mathbf{y}\|^2. \quad (12)$$

Applying (12) to $H(\mathbf{s}(k+1)) = H(\mathbf{s}(k) + \alpha d(k))$ we get

$$\begin{aligned} H(\mathbf{s}(k+1)) &\leq H(\mathbf{s}(k)) + \alpha\sum_{i=1}^N d_i(k)'\nabla_i H(\mathbf{s}(k)) \\ &\quad + \frac{K_1}{2}\alpha^2\|d(k)\|^2 \\ &= H(\mathbf{s}(k)) + \alpha\sum_{i=1}^N d_i(k)'\nabla_i H\left(\mathbf{s}^i(k)\right) \\ &\quad + \alpha\sum_{i=1}^N d_i(k)'\left[\nabla_i H(\mathbf{s}(k)) - \nabla_i H\left(\mathbf{s}^i(k)\right)\right] \\ &\quad + \frac{K_1}{2}\alpha^2\|d(k)\|^2. \end{aligned}$$

Since, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ and all i, j, $\nabla_{i,j}H(\mathbf{x}) - \nabla_{i,j}H(\mathbf{y}) \leq \|\nabla H(\mathbf{x}) - \nabla H(\mathbf{y})\|$ and, by Assumption 2(b), $\|\nabla H(\mathbf{x}) - \nabla H(\mathbf{y})\| \leq K_1\|\mathbf{x} - \mathbf{y}\|$ applying this inequality to the third term above we get

$$\begin{aligned} H(\mathbf{s}(k+1)) &\leq H(\mathbf{s}(k)) + \alpha\sum_{i=1}^N d_i(k)'\nabla_i H\left(\mathbf{s}^i(k)\right) \\ &\quad + \alpha K_1\sum_{i=1}^N\sum_{j=1}^{n_i}|d_{i,j}(k)|\cdot\left\|\mathbf{s}(k) - \mathbf{s}^i(k)\right\| \\ &\quad + \frac{K_1}{2}\alpha^2\|d(k)\|^2 \end{aligned}$$

where $d_{i,j}(k)$ is the jth component of $d_i(k)$. Using Assumption 3(a) in the second term above

$$\begin{aligned} H(\mathbf{s}(k+1)) &\leq H(\mathbf{s}(k)) - \frac{\alpha}{K_3}\sum_{i=1}^N\|d_i(k)\|^2 \\ &\quad + \alpha K_1\sum_{i=1}^N\sum_{j=1}^{n_i}|d_{i,j}(k)|\cdot\left\|\mathbf{s}(k) - \mathbf{s}^i(k)\right\| \\ &\quad + \frac{K_1}{2}\alpha^2\|d(k)\|^2. \end{aligned} \quad (13)$$

Considering the third term in (13), Assumption 4(a) implies that for all i, j, k, $\|s_j(k) - s_j^i(k)\| \leq K_4 g(s_j(k), s_j^i(k))$ and Assumption 4(b) implies that $g(s_j(k), s_j^i(k)) \leq \delta_j(\tilde{k}_{t_k}^j)$. Recalling that $\delta_j(k) = \delta_j(k-1)$ if $k \notin \mathcal{C}^j$, we get $\delta_j(k) = \delta_j(\tilde{k}_{t_k}^j)$ and it follows that:

$$\begin{aligned} \left\|\mathbf{s}(k) - \mathbf{s}^i(k)\right\| &= \left[\sum_{j=1}^N\left\|s_j(k) - s_j^i(k)\right\|^2\right]^{1/2} \\ &\leq K_4\left[\sum_{j=1}^N\delta_j(k)^2\right]^{1/2}. \end{aligned} \quad (14)$$

In view of (8), (9), it follows that for all i:

$$\begin{aligned} \left\|\mathbf{s}(k) - \mathbf{s}^i(k)\right\| \leq K_4\Bigg[&\sum_{j:k\in\mathcal{C}^j} K_\delta^2\|d_j(k)\|^2 \\ &+ \sum_{j:k\notin\mathcal{C}^j} K_\delta^2\left\|d_j\left(\tilde{k}_{t_k}^j\right)\right\|^2\Bigg]^{1/2} \end{aligned} \quad (15)$$

and observe that, due to Assumption 1, we have $k - B + 1 \leq \tilde{k}_{t_k}^j$. Next, set

$$\left\|\tilde{d}(k)\right\| = \left[\sum_{j:k\in\mathcal{C}^j}\|d_j(k)\|^2 + \sum_{j:k\notin\mathcal{C}^j}\left\|d_j\left(\tilde{k}_{t_k}^j\right)\right\|^2\right]^{1/2}. \quad (16)$$

Then, combining (15) and (13), the latter becomes

$$H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha\right)\|d(k)\|^2 + \alpha K_1 K_4 K_\delta \left\|\tilde{d}(k)\right\| \sum_{i=1}^{N}\sum_{j=1}^{n_i}|d_{i,j}(k)|.$$

Recalling that $m = \sum_{i=1}^{N} n_i$ and using the inequality $\sum_{i=1}^{N}\sum_{j=1}^{n_i}|d_{i,j}(k)| \leq \sqrt{m}\|d(k)\|$ (an immediate consequence of the Schwarz inequality), we get

$$H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha\right)\|d(k)\|^2 + \alpha K_1 K_4 K_\delta \sqrt{m}\left\|\tilde{d}(k)\right\| \cdot \|d(k)\|.$$

Using $\|\tilde{d}(k)\| \cdot \|d(k)\| \leq (1/2)(\|\tilde{d}(k)\|^2 + \|d(k)\|^2)$, we further obtain

$$H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha\right)\|d(k)\|^2 + \frac{\alpha K_1 K_4 K_\delta \sqrt{m}}{2}\left(\left\|\tilde{d}(k)\right\|^2 + \|d(k)\|^2\right).$$

We now make use of Assumption 1 to bound $\|\tilde{d}(k)\|^2$. Based on this assumption, from time t_{k-B+1} to t_k each node performs at least one non-null state update step, i.e., $k-B+1 \leq \tilde{k}^j_{t_k} \leq k-1$ for all j such that $k \notin \mathcal{C}^j$. Thus, the second sum in (16) can be written as

$$\sum_{j:k\notin\mathcal{C}^j}\left\|d_j\left(\tilde{k}^j_{t_k}\right)\right\|^2 = \sum_{r=k-B+1}^{k-1}\;\sum_{\left\{j:k\notin\mathcal{C}^j,\tilde{k}^j_{t_k}=r\right\}}\|d_j(r)\|^2 \leq \sum_{r=k-B+1}^{k-1}\|d(r)\|^2.$$

In addition, recalling that $d_j(k) = 0$ for all j such that $k \notin \mathcal{C}^j$, we have $\sum_{j:k\in\mathcal{C}^j}\|d_j(k)\|^2 = \|d(k)\|^2$ and it follows that:

$$\left\|\tilde{d}(k)\right\|^2 \leq \sum_{r=k-B+1}^{k}\|d(r)\|^2.$$

Since it is possible that $k - B + 1 < 0$, let us for convenience set $\|d(r)\| = 0$, for all $r < 0$. Then

$$H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) + \frac{\alpha K_1 K_4 K_\delta \sqrt{m}}{2}\sum_{r=k-B+1}^{k}\|d(r)\|^2 - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha - \frac{K_1 K_4 K_\delta\sqrt{m}}{2}\right)\|d(k)\|^2. \quad (17)$$

Adding all inequalities of the form (17) for $k = 0, 1, \ldots$ through recursive substitution, we can see that all terms $H(\mathbf{s}(k))$ cancel each other except for $H(\mathbf{s}(k+1))$ and $H(\mathbf{s}(0))$. Moreover, all sums $\sum_{r=k-B+1}^{k}\|d(r)\|^2$ added together result in $B\sum_{r=0}^{k}\|d(r)\|^2$ (recalling that $\|d(r)\| = 0$, for all $r < 0$). Therefore, we finally get

$$\begin{aligned}H(\mathbf{s}(k+1)) &\leq H(\mathbf{s}(0)) + \frac{\alpha K_1 K_4 K_\delta \sqrt{m} B}{2}\sum_{r=0}^{k}\|d(r)\|^2 - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha - \frac{K_1 K_4 K_\delta\sqrt{m}}{2}\right)\sum_{r=0}^{k}\|d(r)\|^2\\ &= H(\mathbf{s}(0)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha - \frac{(1+B)}{2}K_1 K_4 K_\delta\sqrt{m}\right) \times \sum_{r=0}^{k}\|d(r)\|^2.\end{aligned}$$

By Assumption 2(a), we have $H(\mathbf{s}(k+1)) \geq 0$, so that the inequality above implies

$$0 \leq H(\mathbf{s}(0)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha - \frac{(1+B)}{2}K_1 K_4 K_\delta\sqrt{m}\right) \times \sum_{r=0}^{k}\|d(r)\|^2. \quad (18)$$

We can always select positive α and K_δ such that

$$1/K_3 - K_1\alpha/2 - (1+B)K_1 K_4 K_\delta\sqrt{m}/2 > 0.$$

For example, the inequality is satisfied by selecting $K_\delta < 2/[(1+B)K_1 K_3 K_4\sqrt{m}]$ and then $\alpha < 2/K_1 K_3 - (1+B)K_4 K_\delta\sqrt{m}$. Thus, we obtain the inequality

$$\sum_{r=0}^{k}\|d(r)\|^2 \leq \frac{H(\mathbf{s}(0))}{\alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha - \frac{(1+B)}{2}K_1 K_4 K_\delta\sqrt{m}\right)}.$$

Observe that the right-hand side is a finite positive constant independent of k and the inequality holds for all k, therefore

$$\sum_{r=0}^{\infty}\|d(r)\|^2 \leq \frac{H(\mathbf{s}(0))}{\alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha - \frac{(1+B)}{2}K_1 K_4 K_\delta\sqrt{m}\right)} < \infty. \quad (19)$$

Since $\|d(r)\|^2 > 0$, the tail of the sequence must vanish, i.e.

$$\lim_{k\to\infty}\|d(k)\| = 0. \quad (20)$$

It follows from (11) that:

$$\lim_{k\to\infty}\|\mathbf{s}(k+1) - \mathbf{s}(k)\| = 0. \quad (21)$$

Since $\tilde{k}^j_{t_k} \to \infty$ when $k \to \infty$ for all j, we have $\lim_{k\to\infty}\|\tilde{d}(k)\| = 0$. Due to (15), we have for all j

$$\lim_{k\to\infty}\left\|\mathbf{s}^j(k) - \mathbf{s}(k)\right\| = 0. \quad (22)$$

Assumption 3(b) and (20) imply that for all j, $\lim_{k\to\infty,k\in\mathcal{C}^j}\|\nabla_j H(\mathbf{s}^j(k))\| = 0$. Recalling the definition of $\tilde{k}^j_{t_k}$, we have $\lim_{k\to\infty}\|\nabla_j H(\mathbf{s}^j(\tilde{k}^j_{t_k}))\| = 0$ and due to (22) we conclude that $\lim_{k\to\infty}\|\nabla_j H(\mathbf{s}(\tilde{k}^j_{t_k}))\| = 0$.

Finally, by (21) we obtain $\lim_{k\to\infty}\|\mathbf{s}(\tilde{k}_{t_k}^j)-\mathbf{s}(k)\|=0$ and Assumption 2(b) implies that $\lim_{k\to\infty}\|\nabla_j H(\mathbf{s}(k))\|=0$ for all j, which completes the proof. ■

Corollary 1: If $0<\alpha<2/K_1K_3$, then

$$K_\delta < \frac{1}{(1+B)K_4\sqrt{m}}\left(\frac{2}{K_1K_3}-\alpha\right) \tag{23}$$

guarantees that $\lim_{k\to\infty}\nabla H(\mathbf{s}(k))=0$.

Proof: This follows directly from (17) by ensuring that the term $1/K_3-K_1\alpha/2-(1+B)K_1K_4K_\delta\sqrt{m}/2$ is positive. ■

Note that (23) provides an upper bound for K_δ that guarantees convergence under the conditions of Theorem 1. Obviously, there may be larger values of K_δ under which convergence is still possible.

As already mentioned, we often set $d_i(\mathbf{s}^i(k)) = -\nabla_i H(\mathbf{s}^i(k))$ in (11) and use $g(x_i(t),x_i^j(t)) = \|x_i(t)-x_i^j(t)\|$, in which case Assumption 3 is satisfied with $K_2=K_3=1$ and Assumption 4(a) with $K_4=1$. It follows from Corollary 1 that we can choose $\alpha<2/K_1$ and (23) leads to a K_δ arbitrarily close from below to $(2/K_1-\alpha)[(1+B)\sqrt{m}]^{-1}$. Observe that, if node states are scalar, this value is inversely proportional to $\sqrt{m}=\sqrt{N}$. Thus, large networks require a smaller value of K_δ, implying that convergence is less tolerant to a node's state estimates evaluated by other nodes and communication needs to be more frequent. For vector node states the same is true since $m=\sum_{i=1}^N n_i$. Along the same lines, note that K_δ is inversely proportional to B, which means that when there is a larger difference in the state update frequency between the fastest node and the slowest node (larger B), more communication is necessary in order to preserve convergence. Finally, a smaller step size α (slower change of states) allows us to choose a larger K_δ, which means greater tolerance to estimation errors.

A. Optimization With State Update Noise

As already mentioned, we have assumed that each node executing (1) is capable of reaching a desired new state $s_i(k+1)$ from $s_i(k)$ within a time interval shorter than the time between update events. Thus, any noise present in this update step affects only the time required for $s_i(k+1)$ to be attained. If, however, we explicitly include a noise term $w_i(k)$ in (1), then we must consider a new state update process

$$s_i(k+1)=s_i(k)+\alpha d_i\left(\mathbf{s}^i(k)\right)+w_i(k). \tag{24}$$

We will make the following assumption regarding this added noise process:

Assumption 5: There exists a positive constant $K_w<2/K_1K_3$ such that $w_{i,j}(k)\le K_w d_{i,j}(k)$, where $d_{i,j}(k)$ and $w_{i,j}(k)$ are the jth scalar component of $d_i(k)$ and $w_i(k)$ respectively.

This assumption requires the noise incurred during a state change to generally decrease with the magnitude of the state changes in the sense that it is bounded by $K_w d_{i,j}(k)$, which decreases with k.

Theorem 2: Under Assumptions 1–5, the communication event policy (7), and the state update scheme (24), if the error threshold $\delta_i(k)$ controlling communication events is set by (8), (9), then there exist positive constants α and K_δ such that $\lim_{k\to\infty}\nabla H(\mathbf{s}(k))=0$.

Proof: The proof is almost identical to that of Theorem 1. We apply (12) to $H(\mathbf{s}(k+1))=H(\mathbf{s}(k)+\alpha d(k)+w(k))$ and get

$$H(\mathbf{s}(k+1))\le H(\mathbf{s}(k))+\sum_{i=1}^N(\alpha d_i(k)+w_i(k))'\nabla_i H(\mathbf{s}(k)) + \frac{K_1}{2}\|\alpha d(k)+w(k)\|^2.$$

By Assumption 5, we have $\|\alpha d(k)+w(k)\|^2 \le (\alpha+K_w)^2\|d(k)\|^2$ and $(\alpha d_i(k)+w_i(k))'\nabla_i H(\mathbf{s}(k)) \le (\alpha+K_w)d_i(k)'\nabla_i H(\mathbf{s}(k))$. Thus

$$\begin{aligned} H(\mathbf{s}(k+1)) &\le H(\mathbf{s}(k))+\sum_{i=1}^N(\alpha+K_w)d_i(k)'\nabla_i H(\mathbf{s}(k)) + \frac{K_1}{2}(\alpha+K_w)^2\|d(k)\|^2 \\ &= H(\mathbf{s}(k))+\sum_{i=1}^N(\alpha+K_w)d_i(k)'\nabla_i H\left(\mathbf{s}^i(k)\right) \\ &\quad+\sum_{i=1}^N(\alpha+K_w)d_i(k)' \times\left[\nabla_i H(\mathbf{s}(k))-\nabla_i H\left(\mathbf{s}^i(k)\right)\right] \\ &\quad+\frac{K_1}{2}(\alpha+K_w)^2\|d(k)\|^2. \end{aligned}$$

Omitting all steps that are the same as those in the proof of Theorem 1, we obtain the analog of (18)

$$0\le H(\mathbf{s}(0))-(\alpha+K_w)\left[\frac{1}{K_3}-\frac{K_1}{2}(\alpha+K_w) - \frac{(1+B)}{2}K_1K_4K_\delta\sqrt{m}\right]\sum_{r=0}^k\|d(r)\|^2.$$

We can always select positive α and K_δ such that

$$1/K_3-K_1\alpha/2-K_1K_w/2-(1+B)K_1K_4K_\delta\sqrt{m}/2>0$$

because first, by Assumption 5, $1/K_3>K_1K_w/2$, and second, $K_1\alpha/2$ and $(1+B)K_1K_4K_\delta\sqrt{m}/2$ can be arbitrarily close to 0 by choosing small α and K_δ. The rest of the proof is identical to that of Theorem 1 and is thus omitted. ■

IV. Optimization With Non-Negligible Communication Delays

Thus far, we have assumed negligible communication delays, i.e., the transmission and reception of a message in a communication event has been treated as instantaneous. Under this assumption, when a communication event is triggered by node i at t according to (7), a copy of the true state of node i is sent to node j and the error function $g(x_i(t),x_i^j(t))$ is reset to zero immediately. This was illustrated by the trajectory of $g(x_i(t),x_i^j(t))$ in Fig. 2. As a result, we also have at our disposal the upper bound $\delta_i(\tilde{k}_t^i)$ for $g(x_i(t),x_i^j(t))$, which, due to Assumption 4(a), leads to a bound for $\|x_i(t)-x_i^j(t)\|$. This upper bound was instrumental in the proof of Theorem 1.

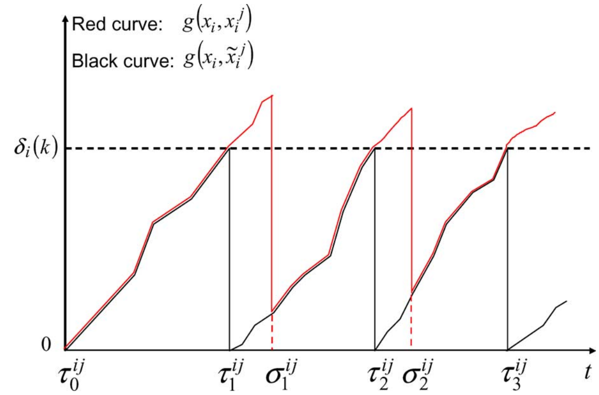


Fig. 3. Trajectories of error functions with communication delays. In this case, every message sent by node i is received before a new communication event is triggered at node i.

In this section, we discuss the case where communication delays are not negligible, modify the communication event policy accordingly, and derive a convergence result similar to Theorem 1. The key to achieving this is to find another bound for $g(x_i(t), x_i^j(t))$ and $\|x_i(t) - x_i^j(t)\|$ without resorting to a negligible delay assumption. We begin by observing that, as a consequence of non-negligible communication delays, after a communication event originates from node i to node j at τ_n^{ij}, node j cannot set x_i^j to $x_i(\tau_n^{ij})$ immediately. Therefore, the error function $g(x_i(t), x_i^j(t))$ continues to grow until the message containing $x_i(\tau_n^{ij})$ is received by node j. This point is illustrated in Fig. 3 where the red curve is an example of the trajectory of $g(x_i(t), x_i^j(t))$ and σ_n^{ij} denotes the time when a message sent by node i at τ_n^{ij} is received at node j. Obviously, if messages sent by a node can take infinitely long to reach other nodes, there can be no cooperation among these nodes and a distributed algorithm may never reach an optimal solution. If this happens, then the underlying communication network used to control the system is likely to be unstable, a case we must exclude. In particular, if $\sigma_n^{ij} - \tau_n^{ij}$ for some $n = 1, 2, \ldots$ in Fig. 3 is unbounded, then $g(x_i(t), x_i^j(t))$ may be unbounded, possibly leading to an infinitely large estimation error (for simplicity, the error bound $\delta_i(k)$ is treated as constant in all figures in this section). Thus, as a first step, we add the following assumption:

Assumption 6: There exists a non-negative integer D such that if a message is sent before t_{k-D} from any node i to any other node j, it will be received before t_k.

In other words, we assume that at most D state update events can occur between a node sending a message and all destination nodes receiving this message.

An additional complication caused by non-negligible delays is in the triggering of communication events. Specifically, if we still use the policy in (7), then between τ_n^{ij} and σ_n^{ij} the value of $g(x_i(t), x_i^j(t))$ generally remains above δ_i (as shown in Fig. 3). As a result, node i will continuously trigger communication events until σ_n^{ij}. To suppress this redundancy over the interval $[\tau_n^{ij}, \sigma_n^{ij}]$, we need to modify (7). Doing so requires carefully differentiating between the way in which node j estimates the state of node i in the presence of communication delays as opposed to no delays (as assumed in the previous section). Thus, let us express such a general-purpose estimate as follows:

$$x_i^j(t) = \begin{cases} h_i^j\left(t; \tau_{n-1}^{ij}\right) & \rho_{n-1} < t < \rho_n, \quad n = 1, 2, \ldots \\ x_n & t = \rho_n, \quad n = 1, 2, \ldots \end{cases} \tag{25}$$

where $h_i^j(\cdot)$ denotes the specific functional form of the estimator used by node j to estimate the state of node i and τ_n^{ij}, $n = 1, 2, \ldots$, is the time when the nth communication event from i to j is generated and viewed as a parameter of the estimator. The estimate is defined over an interval (ρ_{n-1}, ρ_n) and ρ_n is the nth time when the estimate is reset to a given value x_n. This provides a formal representation of the estimation process in our optimization framework: At instants ρ_n, $n = 1, 2, \ldots$, node j resets its estimate of node i's state to some value x_n and then repeats the estimation process with this new initial condition over the next interval (ρ_n, ρ_{n+1}). In the case of negligible communication delays (considered in the previous section), we have

$$\rho_n = \tau_n^{ij}, \quad x_n = x_i\left(\tau_n^{ij}\right) \tag{26}$$

that is, the nth reset time is the instant when a communication event is generated from node i to node j, and the reset value is the corresponding actual state of i at that time. However, in the presence of communication delays, we must set

$$\rho_n = \sigma_n^{ij}, \quad x_n = h_i^j\left(\sigma_n^{ij}; \tau_n^{ij}\right) \tag{27}$$

where $\sigma_n^{ij} > \tau_n^{ij}$ is the time the nth message sent from node i is received by node j and the reset value depends on the estimation method used by node j.

In view of this discussion, we define the new communication event policy

$$\tau_n^{ij} = \inf\left\{t : g\left(x_i(t), h_i^j\left(t; \tau_{n-1}^{ij}\right)\right) \geq \delta_i\left(\tilde{k}_t^i\right), \quad t > \tau_{n-1}^{ij}\right\} \tag{28}$$

with the initial condition $\tau_0^{ij} = 0$. Next, we introduce a new variable $\tilde{x}_i^j(t)$, defined as

$$\tilde{x}_i^j(t) = \begin{cases} h_i^j\left(t; \tau_{n-1}^{ij}\right) & \tau_{n-1}^{ij} < t < \tau_n^{ij}, \quad n = 1, 2, \ldots \\ x_i(\tau_n^{ij}) & t = \tau_n^{ij}, \quad n = 1, 2, \ldots \end{cases} \tag{29}$$

with an initial condition $\tilde{x}_i^j(0) = x_i(0)$. In other words, $\tilde{x}_i^j(t)$ is the estimate of node i's state used by node j as if there were no communication delays, consistent with (26). This variable is maintained by node i. If the estimation process is designed to have the property that $h_i^j(\tau_n^{ij}; \tau_n^{ij}) = x_i(\tau_n^{ij})$, (29) can be further simplified to

$$\tilde{x}_i^j(t) = h_i^j\left(t; \tau_{n-1}^{ij}\right), \quad \tau_{n-1}^{ij} \leq t < \tau_n^{ij}, \quad n = 1, 2, \ldots$$

Fig. 3 illustrates the difference between a trajectory of $g(x_i(t), \tilde{x}_i^j(t))$ (black curve) and a trajectory of $g(x_i(t), x_i^j(t))$ (red curve) under (28). Notice that if the communication delay is zero, then $g(x_i(t), \tilde{x}_i^j(t))$ and $g(x_i(t), x_i^j(t))$ will obviously follow the exact same trajectory. There is an additional reason for introducing $\tilde{x}_i^j(t)$ at node i: If communication delays are

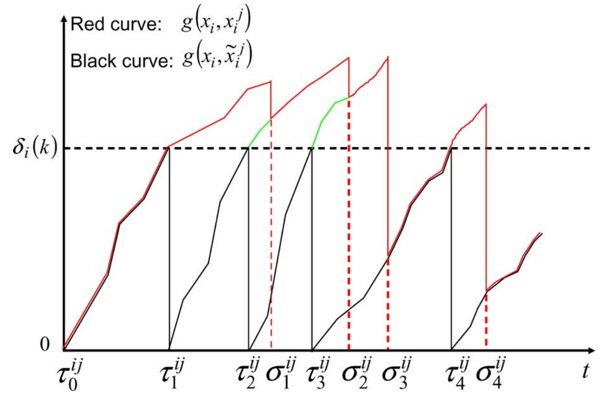


Fig. 4. Trajectories of error functions with communication delays. In this case, additional communication events occur before a message is received.

not negligible and are typically unknown, then node i does not know the exact value of $x_i^j(t)$ and hence cannot execute (7).

Regarding the possible forms that $h_i^j(t;\tau_n^{ij})$ can take, we will discuss two cases. First, if node j uses a static state estimation method, i.e., $h_i^j(t;\tau_n^{ij}) = x_i(\tau_n^{ij})$, it simply sets its estimate of i to the value contained in the received message

$$x_i^j\left(\sigma_n^{ij}\right) = x_i\left(\tau_n^{ij}\right). \qquad (30)$$

In this case, $\tilde{x}_i^j(t) = x_i(\tau_n^{ij})$ for all $t \in [\tau_n^{ij}, \tau_{n+1}^{ij})$, in accordance with (29). On the other hand, if node j uses a dynamic state estimation method, the value of $x_i^j(\sigma_n^{ij})$ depends on τ_n^{ij} as well. For example, if node j uses (4), then

$$x_i^j\left(\sigma_n^{ij}\right) = x_i\left(\tau_n^{ij}\right) + \left(\sigma_n^{ij} - \tau_n^{ij}\right) \cdot D_i \qquad (31)$$

where $D_i = \alpha_i \cdot d_{i,\tau_n^{ij}}/\Delta_i$. Note that in this case evaluating the state estimate in (31) requires knowledge of τ_n^{ij}, i.e., the message sent by node i must be time-stamped (otherwise, in the presence of random delays, node j cannot infer the value of τ_n^{ij} when it receives a message at σ_n^{ij}). This, in turn, requires a clock synchronization mechanism across nodes (which is often not simple to implement).

Finally, we discuss how the new communication event policy (28) and Assumption 6 help us bound $g(x_i(t), x_i^j(t))$ and $\|x_i(t) - x_i^j(t)\|$. This will be formalized within the proof of Theorem 3. For simplicity, we first consider the case where every message from a node reaches its destination before the next communication event is triggered, i.e., $\sigma_n^{ij} < \tau_{n+1}^{ij}$ for all $n = 1, 2, \ldots$, as shown in Fig. 3. At each σ_n^{ij}, the state estimate $x_i^j(\sigma_n^{ij})$ is synchronized with $\tilde{x}_j^j(\sigma_n^{ij})$ defined in (29) and we have $g(x_i(\sigma_n^{ij}), x_i^j(\sigma_n^{ij})) \le \delta_i(k^i_{\sigma_n^{ij}})$. Due to Assumption 6, at most D state update events occur between τ_n^{ij} and σ_n^{ij} for all $n = 1, 2, \ldots$, which guarantees that $g(x_i(t), x_i^j(t))$ will not exceed $\delta_i(k)$, for some k, augmented by the estimation error accumulated over these D state update steps.

If, on the other hand, $g(x_i(t), \tilde{x}_i^j(t))$ reaches $\delta_i(\tilde{k}_t^i)$ during $[\tau_n^{ij}, \sigma_n^{ij}]$, then a new communication message will be sent out by node i before the previous one is received. For example, in Fig. 4, $\tau_2^{ij} < \sigma_1^{ij}$ and $\tau_3^{ij} < \sigma_2^{ij}$. Although the trajectory of $g(x_i(t), x_i^j(t))$ becomes more complicated in this case, $g(x_i(t), x_i^j(t))$ (the red curve) still has an upper bound, which will be given within the proof of Theorem 3.

In light of this discussion, if we choose an error function such that $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|/K_4$, it is obvious that $\|x_i(t) - x_i^j(t)\|$ is bounded. Thus, if we choose any error function such that $\|x_i(t) - x_i^j(t)\| \le K_4 \cdot g(x_i(t), x_i^j(t))$ [according to Assumption 4(a)], $\|x_i(t) - x_i^j(t)\|$ is also bounded, which is key to the proof of Theorem 3. In Theorem 3, we consider the static estimation case (30) which is the least costly in terms of estimation complexity. Moreover, static estimation, as already pointed out, does not require a synchronization mechanism which (31), for example, does. We must also point out that in Theorem 3 we no longer use Assumption 4(b); instead, Assumption 6 will allow us to derive the bounds required to establish the desired convergence result.

Theorem 3: Under Assumptions 1–4(a) and 6, the communication event policy (28), the state update scheme (11), and the static estimation method (30), if the error threshold $\delta_i(k)$ controlling communication events is set by (8), (9), then there exist positive constants α and K_δ such that $\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$.

Proof: The first few steps of the proof are identical to those of Theorem 1. In particular, we obtain (13) which we rewrite below for convenience

$$H(\mathbf{s}(k+1)) \le H(\mathbf{s}(k)) - \frac{\alpha}{K_3} \sum_{i=1}^{N} \|d_i(k)\|^2 + \alpha K_1 \times \sum_{i=1}^{N} \sum_{j=1}^{n_i} |d_{i,j}(k)| \cdot \left\|\mathbf{s}(k) - \mathbf{s}^i(k)\right\| + \frac{K_1}{2}\alpha^2 \|d(k)\|^2. \qquad (32)$$

Next, we seek a bound for $\|\mathbf{s}(k) - \mathbf{s}^i(k)\|$ in the third term of (32). Under the assumption of negligible communication delay, we were able to use Assumption 4 and establish the bound $\|s_j(k) - s_j^i(k)\| \le K_4\delta_j(k)$. In the presence of communication delays, however, this no longer holds (as seen in the examples of Figs. 3 and 4). Instead, we make use of Assumption 6, based on which if node j sends a message using (28) in the worst case it can take a period of time which contains D state update events to reach all destination nodes. During this period, x_j might be updated up to D times and each update changes the value of x_j by $\alpha\|d_j(k)\|$ through (11) (recall that $d_j(k)$ is the shorthand for $d_j(\mathbf{s}^j(k))$). Since, by assumption, static estimation is used, we can ignore changes in x_j^i. We can then establish the following bound for the jth component of $\|s(k) - s^i(k)\|$:

$$\left\|s_j(k) - s_j^i(k)\right\| \le \max_{l=k-D,\ldots,k+1} \left(K_4\delta_j(l-1) + \alpha \sum_{r=l}^{k} \|d_j(r)\|\right). \qquad (33)$$

To justify this bound, first observe that the value of $s_j^i(k)$, which node i uses in updating its state at t_k, is based on a message arrival from node j at some time σ_L^{ji} such that $\sigma_L^{ji} \le t_k$ and there is no message from j in $(\sigma_L^{ji}, t_k]$. Since static estimation is assumed, $s_j^i(k) = x_j(\tau_L^{ji})$, where $\tau_L^{ji} < \sigma_L^{ji}$ is the associated time of the communication event at node j. On the other hand, immediately after sending this message at τ_L^{ji}, node j sets $\tilde{x}_j^i(t) = x_j(\tau_L^{ji})$ and evaluates $g(x_j(t), \tilde{x}_j^i(t))$ for $t > \tau_L^{ji}$. There are now two cases to consider. First, if no new communication event occurs at j in the interval $(\tau_L^{ji}, t_k]$, then the error function at j remains below its threshold throughout this period and we have $\|s_j(k)-s_j^i(k)\| = \|x_j(t_k)-x_j(\tau_L^{ji})\| \leq K_4\delta_j(k)$ by Assumption 4(a), which makes (33) valid.

The second possibility is that node j sends at least one new message to i in $(\tau_L^{ji}, t_k]$. If $\tau_L^{ji} < t_{k-D}$, due to Assumption 5, $g(x_j(t), \tilde{x}_j^i(t))$ could not have exceeded its threshold before t_{k-D} (if that were to happen, a new message would have arrived at i). So, we only need to consider $[\max(t_{k-D}, \tau_L^{ji}), t_k]$ or, since $t_{k-D} \leq \max(t_{k-D}, \tau_L^{ji})$, just $[t_{k-D}, t_k]$. Suppose $g(x_j(t), \tilde{x}_j^i(t))$ exceeds the current error threshold $\delta_j(\tilde{k}_t^j)$ at some $t \in [t_{k-D}, t_k]$. Since a new communication event is triggered, we have $t = \tau_{L+1}^{ji}$. Moreover, since static estimation is assumed, $\tilde{x}_j^i(t) = x_j(\tau_L^{ji})$ is fixed in $[\tau_L^{ji}, \tau_{L+1}^{ji})$ and this event must be due to $x_j(t)$ changing at τ_{L+1}^{ji} through (11). Thus, we set $t_l = \tau_{L+1}^{ji}$ for some $l \leq k$. Right before this state update, we have $\|x_j(t_l^-) - \tilde{x}_j^i(t_l^-)\| < K_4\delta_j(\tilde{k}_{t_l^-}^j)$. Right after the state update, based on (11), $x_j(t_l) = s_j(l)$ is changed by $\alpha\|d_j(l)\|$, hence

$$\left\|x_j(t_l) - \tilde{x}_j^i\left(t_l^-\right)\right\| \leq K_4\delta_j\left(\tilde{k}_{t_l^-}^j\right) + \alpha\left\|d_j(l)\right\| = K_4\delta_j(l-1) + \alpha\left\|d_j(l)\right\|. \quad (34)$$

Observe that τ_{L+1}^{ji} may occur before or after σ_L^{ji}, but either way, at t_k we have $s_j^i(k) = x_j(\tau_L^{ji})$, thus $s_j^i(k) = \tilde{x}_j^i(t_l^-)$. Therefore, the difference between $\|x_j(t_l) - \tilde{x}_j^i(t_l^-)\|$ and $\|s_j(k) - s_j^i(k)\|$ can only be caused by any change in $x_j(t), t \in (\tau_{L+1}^{ji}, t_k]$. Since there are at most $k-l$ such state updates in this interval, we need the sum $\alpha\sum_{r=l+1}^{k}\|d_j(r)\|$ in (33) to bound $\|s_j(k) - s_j^i(k)\|$.

Finally, since t_l in (34) is such that $t_l \in [t_{k-D}, t_k]$ and $\delta_j(l-1)$ varies over this interval, the max operator in (33) is needed to ensure the worst case is accounted for.

It follows from (33) that:

$$\left\|s_j(k) - s_j^i(k)\right\| \leq \max_{l=k-D,\ldots,k+1}\left(K_4\delta_j(l-1)\right) + \alpha\sum_{r=k-D}^{k}\left\|d_j(r)\right\| = K_4\delta_j\left(l_{jk}^*\right) + \alpha\sum_{r=k-D}^{k}\left\|d_j(r)\right\|$$

where

$$l_{jk}^* = \arg\max_{l=k-D-1,\ldots,k}\delta_j(l) \quad (35)$$

and in the case of a tie for l_{jk}^* it is broken arbitrarily. Thus

$$\left\|\mathbf{s}(k)-\mathbf{s}^i(k)\right\| = \left[\sum_{j=1}^{N}\left\|s_j(k)-s_j^i(k)\right\|^2\right]^{1/2} \leq \left[\sum_{j=1}^{N}\left(K_4\delta_j\left(l_{jk}^*\right)+\alpha\sum_{r=k-D}^{k}\left\|d_j(r)\right\|\right)^2\right]^{1/2} \leq \sum_{j=1}^{N}\left(K_4\delta_j\left(l_{jk}^*\right)+\alpha\sum_{r=k-D}^{k}\left\|d_j(r)\right\|\right).$$

Making use of the inequalities

$$\sum_{j=1}^{N}\delta_j\left(l_{jk}^*\right) \leq \sqrt{N}\left[\sum_{j=1}^{N}\delta_j\left(l_{jk}^*\right)^2\right]^{1/2}$$

$$\sum_{j=1}^{N}\left\|d_j(r)\right\| \leq \sqrt{N}\left[\sum_{j=1}^{N}\left\|d_j(r)\right\|^2\right]^{1/2} = \sqrt{N}\left\|d(r)\right\|$$

we get

$$\left\|\mathbf{s}(k) - \mathbf{s}^i(k)\right\| \leq \sqrt{N}K_4\left[\sum_{j=1}^{N}\delta_j\left(l_{jk}^*\right)^2\right]^{1/2} + \alpha\sqrt{N}\sum_{r=k-D}^{k}\left\|d(r)\right\|.$$

In view of (8), (9), it follows that for all l_{jk}^* and j

$$\sum_{j=1}^{N}\delta_j\left(l_{jk}^*\right)^2 \leq \sum_{j:l_{jk}^*\in\mathcal{C}^j}K_\delta^2\left\|d_j\left(l_{jk}^*\right)\right\|^2 + \sum_{j:l_{jk}^*\notin\mathcal{C}^j}K_\delta^2\left\|d_j\left(L_{l_{jk}^*,j}\right)\right\|^2$$

where $L_{l,j} = \max\{n : n \in \mathcal{C}^j, n = l-B+1,\ldots,l\}$, i.e., $t_{L_{l,j}}$ is the most recent state update time up to t_l when node j took a non-null update step. The existence of $L_{l,j}$ is guaranteed by Assumption 1. Setting

$$\left\|\tilde{d}(k)\right\| = \left[\sum_{j:l_{jk}^*\in\mathcal{C}^j}\left\|d_j\left(l_{jk}^*\right)\right\|^2 + \sum_{j:l_{jk}^*\notin\mathcal{C}^j}\left\|d_j\left(L_{l_{jk}^*,j}\right)\right\|^2\right]^{1/2} \quad (36)$$

we get

$$\left\|\mathbf{s}(k) - \mathbf{s}^i(k)\right\| \leq \sqrt{N}K_4K_\delta\left\|\tilde{d}(k)\right\| + \alpha\sqrt{N}\sum_{r=k-D}^{k}\left\|d(r)\right\|. \quad (37)$$

Then, combining (37) and (32), the latter becomes

$$H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha\right)\|d(k)\|^2 + \alpha K_1\sqrt{N} \times\left(K_4K_\delta\left\|\tilde{d}(k)\right\| + \alpha\sum_{r=k-D}^{k}\|d(r)\|\right)\sum_{i=1}^{N}\sum_{j=1}^{n_i}|d_{i,j}(k)|.$$

Since $\sum_{i=1}^{N}\sum_{j=1}^{n_i}|d_{i,j}(k)| \leq \sqrt{m}\|d(k)\|$, this gives

$$H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha\right)\|d(k)\|^2 + \alpha K_1 \times\sqrt{Nm}\left(K_4K_\delta\left\|\tilde{d}(k)\right\| + \alpha\sum_{r=k-D}^{k}\|d(r)\|\right)\cdot\|d(k)\|.$$

Using $\|\tilde{d}(k)\|\cdot\|d(k)\| \leq (1/2)(\|\tilde{d}(k)\|^2 + \|d(k)\|^2)$, we further obtain

$$H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) - \alpha\left(\frac{1}{K_3} - \frac{K_1}{2}\alpha\right)\|d(k)\|^2 + \frac{1}{2}\alpha K_1\sqrt{Nm}$$

$$
\begin{aligned}
&\times\left[K_4K_\delta\left(\left\|\tilde{d}(k)\right\|^2+\|d(k)\|^2\right)\right.\\
&\left.\quad+\alpha\sum_{r=k-D}^{k}\left(\|d(r)\|^2+\|d(k)\|^2\right)\right]\\
&\le H(\mathbf{s}(k))-\alpha\left(\frac{1}{K_3}-\frac{K_1}{2}\alpha\right)\|d(k)\|^2\\
&\quad+\frac{1}{2}\alpha K_1\sqrt{Nm}\left(K_4K_\delta+(D+1)\alpha\right)\|d(k)\|^2\\
&\quad+\frac{1}{2}\alpha K_1\sqrt{Nm}\\
&\quad\times\left(K_4K_\delta\left\|\tilde{d}(k)\right\|^2+\alpha\sum_{r=k-D}^{k}\|d(r)\|^2\right).
\end{aligned}
$$

We now make use of Assumption 1 to bound $\|\tilde{d}(k)\|^2$ in (36). Based on this assumption, for every $l\ge 0$, from time t_{l-B+1} to t_l each node performs at least one non-null state update step, i.e., $l-B+1\le L_{l,j}\le l-1$ for all j such that $l\notin\mathcal{C}^j$. From the definition of l_{jk}^* (35), we also know that $k-D-1\le l_{jk}^*\le k$. Thus, we have $k-D-B\le L_{l_{jk}^*,j}\le k-1$ for all j such that $l\notin\mathcal{C}^j$. Thus, the two sums in (36) can be written as

$$
\sum_{j:l_{jk}^*\in\mathcal{C}^j}\left\|d_j\left(l_{jk}^*\right)\right\|^2=\sum_{u=k-D-1}^{k}\sum_{\substack{j:l_{jk}^*\in\mathcal{C}^j,\\ l_{jk}^*=u}}\|d_j(u)\|^2
$$

$$
\sum_{j:l_{jk}^*\notin\mathcal{C}^j}\left\|d_j\left(L_{l_{jk}^*,j}\right)\right\|^2=\sum_{u=k-D-B}^{k-1}\sum_{\substack{j:l_{jk}^*\notin\mathcal{C}^j,\\ L_{l_{jk}^*,j}=u}}\|d_j(u)\|^2.
$$

It follows that:

$$
\begin{aligned}
\left\|\tilde{d}(k)\right\|^2&=\sum_{j:l_{jk}^*\in\mathcal{C}^j}\left\|d_j\left(l_{jk}^*\right)\right\|^2+\sum_{j:l_{jk}^*\notin\mathcal{C}^j}\left\|d_j\left(L_{l_{jk}^*,j}\right)\right\|^2\\
&\le\sum_{u=k-D-B}^{k}\|d(u)\|^2.
\end{aligned}
$$

Since it is possible that $k-D-B<0$, let us for convenience set $\|d(u)\|=0$, for all $u<0$. Then

$$
\begin{aligned}
H(\mathbf{s}(k+1))\le H(\mathbf{s}(k))&\\
-\alpha\Bigg(\frac{1}{K_3}-\frac{K_1}{2}\alpha-\frac{1}{2}K_1\sqrt{Nm}&\\
\times\left(K_4K_\delta+(D+1)\alpha\right)\Bigg)\|d(k)\|^2&\\
+\frac{\alpha}{2}K_1\sqrt{Nm}&\\
\times\Bigg(K_4K_\delta\sum_{u=k-D-B}^{k}\|d(u)\|^2&\\
+\sum_{r=k-D}^{k}\alpha\|d(r)\|^2\Bigg).& \qquad (38)
\end{aligned}
$$

Adding all inequalities of the form (38) for $k=0,1,\ldots$, we can see that all terms $H(\mathbf{s}(k))$ cancel each other except for $H(\mathbf{s}(k+1))$ and $H(\mathbf{s}(0))$. Moreover, all sums $\sum_{u=k-D-B}^{k}\|d(u)\|^2$ added together result in $\sum_{r=0}^{k}(D+B+1)\|d(r)\|^2$ (recalling that $\|d(r)\|=0$, for all $r<0$). Therefore, we finally get

$$
\begin{aligned}
&H(\mathbf{s}(k+1))\\
&\quad\le H(\mathbf{s}(0))\\
&\qquad-\alpha\left[\frac{1}{K_3}-\frac{K_1}{2}\alpha-\frac{1}{2}K_1\sqrt{Nm}\left(K_4K_\delta+(D+1)\alpha\right)\right]\\
&\qquad\times\sum_{r=0}^{k}\|d(r)\|^2+\sum_{r=0}^{k}\frac{\alpha}{2}K_1\sqrt{Nm}\\
&\qquad\times\left[K_4K_\delta(D+B+1)\|d(r)\|^2+\alpha(D+1)\|d(r)\|^2\right]\\
&\quad=H(\mathbf{s}(0))\\
&\qquad-\alpha\left[\frac{1}{K_3}-\frac{K_1}{2}\alpha-\frac{1}{2}K_1\sqrt{Nm}\left(K_4K_\delta+(D+1)\alpha\right)\right]\\
&\qquad\times\sum_{r=0}^{k}\|d(r)\|^2+\frac{\alpha}{2}K_1\sqrt{Nm}\\
&\qquad\times\left[K_4K_\delta(D+B+1)+(D+1)\alpha\right]\sum_{r=0}^{k}\|d(r)\|^2.
\end{aligned}
$$

By Assumption 2(a), we have $H(\mathbf{s}(k+1))\ge 0$, so that the inequality above implies:

$$
\begin{aligned}
0\le\ &H(\mathbf{s}(0))\\
&-\alpha\Bigg(\frac{1}{K_3}-\frac{K_1}{2}\alpha-\frac{K_1\sqrt{Nm}}{2}\left(K_4K_\delta+(D+1)\alpha\right)\\
&\qquad-\frac{K_1\sqrt{Nm}}{2}\left(K_4K_\delta(D+B+1)+(D+1)\alpha\right)\Bigg)\\
&\times\sum_{r=0}^{k}\|d(r)\|^2\\
=\ &H(\mathbf{s}(0))-\alpha\Bigg[\frac{1}{K_3}-K_1\left(\frac{1}{2}+\sqrt{Nm}(D+1)\right)\alpha\\
&\qquad-\frac{K_1K_4\sqrt{Nm}(D+B+2)}{2}K_\delta\Bigg]\\
&\times\sum_{r=0}^{k}\|d(r)\|^2.
\end{aligned}
$$

We can always select positive α and K_δ such that

$$
0<\frac{1}{K_3}-K_1\left(\frac{1}{2}+\sqrt{Nm}(D+1)\right)\alpha-\frac{K_1K_4\sqrt{Nm}(D+B+2)}{2}K_\delta
$$

(the argument is similar to the proof of Theorem 1, but the values of α and K_δ in this case are generally smaller). Therefore, we can write

$$
\sum_{r=0}^{k}\|d(r)\|^2\le\frac{H(\mathbf{s}(0))}{\alpha}\left[\frac{1}{K_3}-K_1\left(\frac{1}{2}+\sqrt{Nm}(D+1)\right)\alpha-\frac{K_1K_4\sqrt{Nm}(D+B+2)}{2}K_\delta\right]^{-1}.
$$

Observe that the right-hand side is a finite positive constant independent of k and the inequality holds for all k, therefore

$$\sum_{r=0}^{\infty}\|d(r)\|^2 \leq \frac{H(\mathbf{s}(0))}{\alpha}\left[\frac{1}{K_3}-K_1\left(\frac{1}{2}+\sqrt{Nm}(D+1)\right)\alpha - \frac{K_1K_4\sqrt{Nm}(D+B+2)}{2}K_\delta\right]^{-1} < \infty. \quad (39)$$

The remainder of the proof is based on the same arguments as in the proof of Theorem 1. Specifically, since $\|d(r)\|^2 > 0$, the tail of the sequence must vanish, i.e.

$$\lim_{k\to\infty}\|d(k)\| = 0. \quad (40)$$

It follows from (11) that:

$$\lim_{k\to\infty}\|\mathbf{s}(k+1)-\mathbf{s}(k)\| = 0. \quad (41)$$

Since $l_{jk}^* \to \infty$ and $L_{l_{jk}^*,j} \to \infty$ when $k\to\infty$ for all j, we have $\lim_{k\to\infty}\|\tilde{d}(k)\| = 0$. Due to (37), we have for all j

$$\lim_{k\to\infty}\left\|\mathbf{s}^j(k)-\mathbf{s}(k)\right\| = 0. \quad (42)$$

Assumption 3(b) and (40) imply that for all j, $\lim_{k\to\infty,k\in\mathcal{C}^j}\|\nabla_j H(\mathbf{s}^j(k))\| = 0$. Recalling the definition of $L_{k,j}$, we have $\lim_{k\to\infty}\|\nabla_j H(\mathbf{s}^j(L_{k,j}))\| = 0$ and due to (42) we conclude that $\lim_{k\to\infty}\|\nabla_j H(\mathbf{s}(L_{k,j}))\| = 0$. Finally, by (41) we obtain $\lim_{k\to\infty}\|\mathbf{s}(L_{k,j})-\mathbf{s}(k)\| = 0$ and Assumption 2(b) implies that $\lim_{k\to\infty}\|\nabla_j H(\mathbf{s}(k))\| = 0$ for all j, which completes the proof. ■

Comparing the bounds in (19) and (39) for the case of negligible and non-negligible communication delays respectively, note that by setting $D=0$ we can reduce the latter to the former except for the presence of $\sqrt{Nm}$ rather than $\sqrt{m}$ in the multiplicative factor for K_δ. This was necessitated by the looser bound used for $\|\mathbf{s}(k)-\mathbf{s}^i(k)\|$ in the proof of Theorem 3. In addition, if both the state update noise introduced in Section III-A and non-negligible communication delays are present in the optimization process, as long as Assumptions 5 and 6 are still satisfied, a convergence result (omitted here) similar to Theorem 2 can also be derived.

V. Asynchronous Distributed Coverage Control

In this section, we apply the proposed asynchronous distributed optimization framework to the class of coverage control problems mentioned in Section I. Specifically, we consider the problem of deploying a sensor network so that sensor nodes are located in a way that maximizes the probability of detecting events occurring in a given two-dimensional mission space. Our goal is to show how an asynchronous algorithm may significantly reduce the energy expended on communication among nodes with no adverse effect on performance. We will also illustrate the convergence of the objective function without and with communication delays and examine the effect of K_δ on the convergence speed and the communication cost.

A. Coverage Control Problem Formulation

We begin by briefly reviewing the coverage control problem following the formulation in [9]. We define an event density function $R(x)$ over the *mission space* $\Omega \subset \mathbb{R}^2$, which captures the frequency of random event occurrences at some point $x\in\Omega$. $R(x)$ satisfies $R(x)\geq 0$ for all $x\in\Omega$ and $\int_\Omega R(x)dx < \infty$. We assume that when an event takes place, it will emit some signal which may be observed by some sensor nodes. The cooperative system consists of N mobile sensor nodes deployed into Ω to detect the random events. Their positions are captured by a $2N$-dimensional vector $\mathbf{s}=(s_1,\ldots,s_N)$.

The probability that sensor node i detects an event occurring at $x\in\Omega$, denoted by $p_i(x,s_i)$, is a monotonically decreasing differentiable function of $\|x-s_i\|$, the Euclidean distance between the event and the sensor. Since multiple sensor nodes are deployed to cover the mission space, the joint probability that an event occurring at x is detected (assuming independent sensor detection processes), denoted by $P(x,\mathbf{s})$, is given by

$$P(x,\mathbf{s}) = 1-\prod_{i=1}^{N}[1-p_i(x,s_i)]$$

and the optimization problem of interest is

$$\max_{\mathbf{s}} H(\mathbf{s}) = \max_{\mathbf{s}} \int_\Omega R(x)P(x,\mathbf{s})dx$$

in which we use the locations of the sensor nodes as decision variables to maximize the probability of event detection in Ω. As discussed in [21], this objective function aims at maximizing the joint event detection probability without considering the issue of maintaining a balance between a region within Ω which may be extremely well covered and others which may not. Thus, a more general form of the objective function is

$$H(\mathbf{s}) = \int_\Omega R(x)M(P(x,\mathbf{s}))\,dx \quad (43)$$

where $M(P):[0,1]\to\mathbb{R}$ is a (possibly piecewise) differentiable concave non-decreasing function of the joint detection probability P. Clearly, $M(P)$ may be selected so that the same amount of marginal gain in $P(x,\mathbf{s})$ is assigned a higher reward at a lower value of P. A special case is $M(P)=P$ where coverage balance is not taken into account.

A *synchronous* gradient-based solution was obtained in [9] in which the next way point on the ith mobile sensor's trajectory is determined through

$$s_i(k+1) = s_i(k)+\alpha\frac{\partial H(\mathbf{s})}{\partial s_i}, \quad k=0,1,\ldots \quad (44)$$

The gradient is given by

$$\frac{\partial H(\mathbf{s})}{\partial s_i} = \int_{\Omega_i} R(x)\prod_{k\in\mathcal{B}_i}[1-p_k(x,s_k)]\frac{dp_i(x,s_i)}{dd_i(x)}\frac{s_i-x}{d_i(x)}dx \quad (45)$$

where $d_i(x)\equiv\|x-s_i\|$ and $\Omega_i=\{x : d_i(x)\leq D\}$ is the node i's region of coverage where D denotes the sensing radius of node i (i.e., $p_i(x,s_i)=0$ for $d_i(x)>D$). In addition,

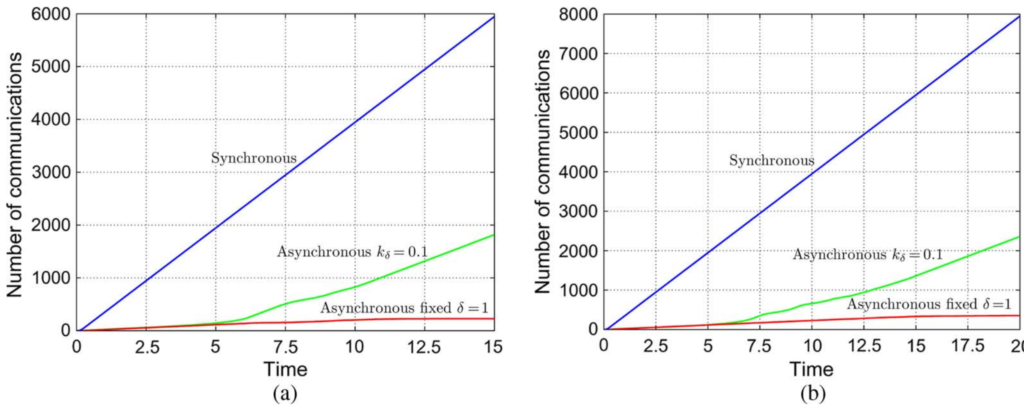


Fig. 5. Communication cost comparison of three distributed optimization algorithms under different communication delay assumptions. (a) Negligible communication delay; (b) non-negligible communication delay

$\mathcal{B}_i = \{k : \|s_i - s_k\| < 2D, k = 1, \ldots, N, k \neq i\}$ is the set of neighbor nodes of i.

The state update rule (44) allows a fully distributed implementation based on node i's local information and the state of its neighbors (instead of the state of all nodes). This eliminates the communication burden of transferring information to and from a central controller and the vulnerability of the whole system which would be entirely dependent on this controller. However, (45) shows that node i needs the exact locations of all nodes in $\mathcal{B}_i$ in order to carry out (44) in a state update. As already mentioned, the communication involved to ensure such state synchronization has a high energy cost which is often unnecessary because the locations of neighboring nodes may be accurately estimated due to minor changes in their locations or update directions (see (3) and (4)).

Next, we will apply the asynchronous method developed above to this problem and compare the results with the synchronous approach in terms of communication cost, performance, and convergence behavior. The asynchronous method is also applied to a nonsmooth version of the coverage control problem recently developed in [21] when the mission space contains obstacles.

B. Asynchronous vs Synchronous Optimal Coverage Control

We present numerical results based on simulations of the coverage control setting using an interactive Java-based simulation environment.[1] We compare three versions of the optimal coverage control solution:

1) Synchronous iterations where all nodes perform state updates using state information from all other nodes.
2) Asynchronous iterations performed by node i with fixed error threshold $\delta_i(k)$, i.e., $\delta_i(k) = \delta_i$ for all k, where δ_i is a positive constant.
3) Asynchronous iterations performed by node i using (8), (9).

The coverage control problem considered here involves four nodes deployed into a rectangular mission space (50 units by 60 units) from the lower left corner. All nodes update their states at approximately the same frequency (one update in every 10 simulation time steps augmented by some uniformly distributed noise) using the same step size in all three schemes. All nodes have identical sensing capability and $p_i(x, s_i) = e^{-0.08\|x - s_i\|}$ for all i, which means that, if deployed individually, a node can only effectively cover a relatively small portion of the mission space ($p_i(x, s_i) < 0.50$ when $\|x - s_i\| > 8.67$). For the asynchronous versions, (3) is used as a simple static state estimation scheme and the error function is $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|$.

In Figs. 5 and 6, we compare these three algorithms under both the negligible and non-negligible communication delay assumptions. When we compare communication costs in Fig. 5, every time a node broadcasts its state information, the total number of communications is increased by one. By looking at these figures, it is clear that the asynchronous method can substantially reduce the communication cost (hence, the energy consumption at all nodes) while performance convergence is virtually indistinguishable from that of the synchronous method. The asynchronous algorithm with fixed $\delta_i(k)$ has the added advantage that it usually stops incurring communication cost earlier than the other two methods. However, it does not guarantee convergence to stationary points. In Figs. 5(b) and 6(b), we provide results for a case with a fairly large communication delay (50 simulation time steps for every transmitted message). Comparing Fig. 6(b) to Fig. 6(a), it is clear that introducing delays does not affect convergence but it does slow it down; in Fig. 6(b), the optimal value of the objective function is attained approximately 5 time units later than the time seen in Fig. 6(a).

Fig. 7 shows the node trajectories for these three methods under the negligible communication delay assumption. Methods 1 and 3 converge to the same node configuration which is indicated by the squares, while method 2 converges to a configuration close to it (we cannot mark the points that configuration method 2 converges to due to the limited resolution of the figure).

In Figs. 8 and 9, under the negligible communication delay assumption, we compare the performance of the asynchronous method with different values of the constant K_δ in (8) under the same coverage control setting as before. We can see that a larger K_δ leads to fewer communication events. But we also notice in Fig. 9 that when $K_\delta = 5$, the objective function curve exhibits

[1]The interactive simulation environment used in all results shown in this section (along with instructions) is located at http://codescolor.bu.edu/simulators.html.

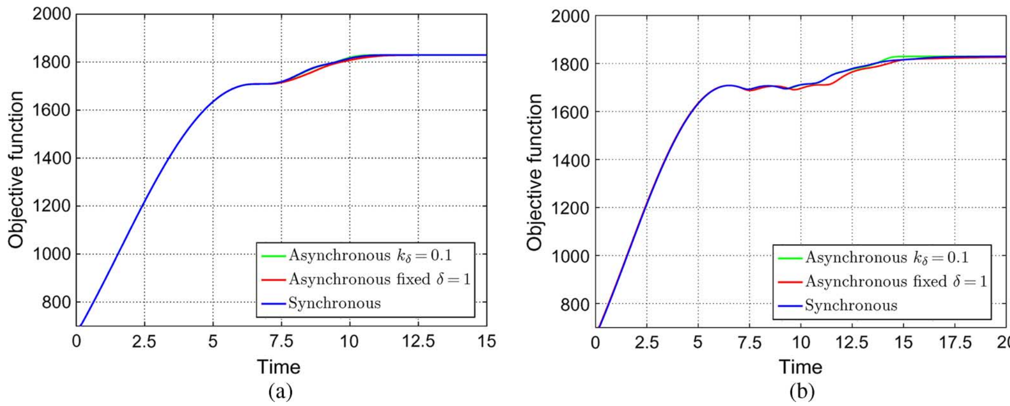


Fig. 6. Objective function trajectories of three distributed optimization algorithms under different communication delay assumptions. (a) Negligible communication delay; (b) Non-negligible communication delay.

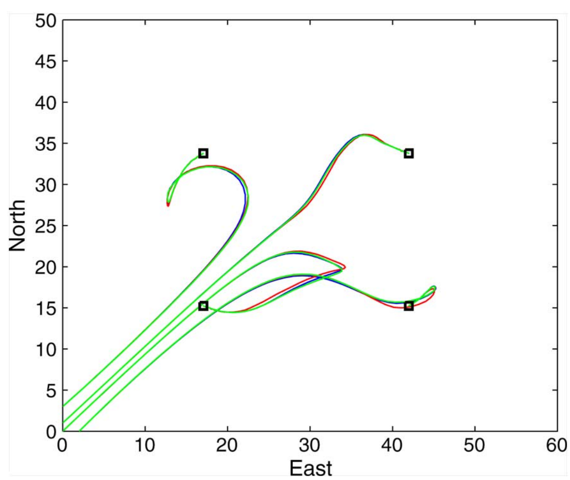


Fig. 7. Node trajectory comparison of three distributed optimization algorithms in a coverage control mission. See Fig. 6 for legend.

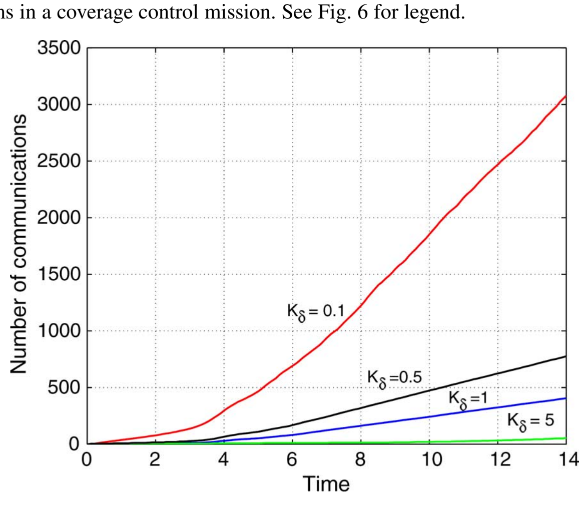


Fig. 8. Communication cost comparison of asynchronous distributed algorithm with different K_δ.

considerable oscillations before it converges. This suggests that K_δ is set too high and some "necessary" communications between nodes have been omitted at the expense of effective cooperation. In other words, when K_δ is increased, although convergence to a local optimum may still be guaranteed if (23) is satisfied, so is the risk of slower convergence.

In Fig. 10, we show results from a larger coverage control problem with 100 nodes with no obstacle or communication

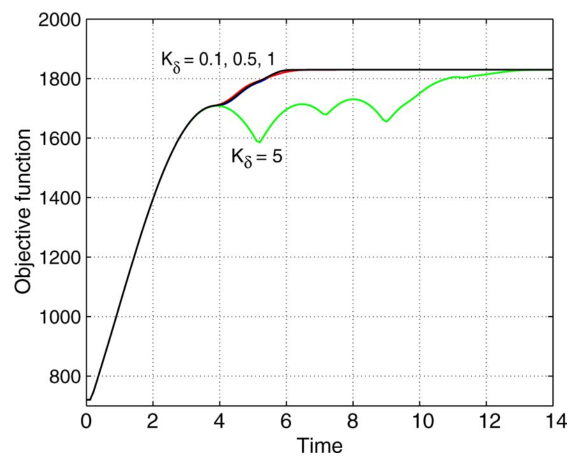


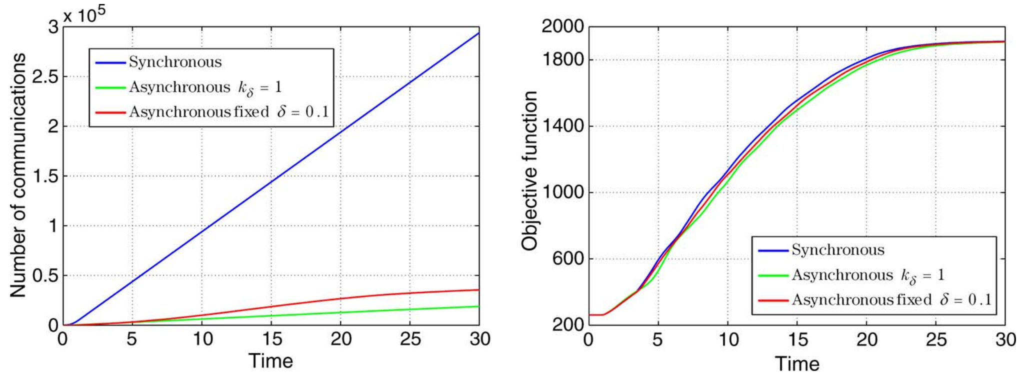
Fig. 9. Objective function trajectories of asynchronous distributed algorithm with different K_δ.

delay and reduced sensing range. We observe similar communication savings as in the smaller problem of Fig. 5 with only 4 nodes.

Next, we apply the asynchronous algorithm to a problem with polygonal obstacles in the region to be covered and employ (4) as the state estimation method. Five nodes are deployed into a mission space cluttered with obstacles as shown in Fig. 11. In this case, we have a nonsmooth optimization problem with gradients that are more complicated than (45) in order to account for discontinuities in the sensor detection probability functions $p_i(x, s_i)$; details on their derivation are provided in [21]. Convergence under (44) cannot always be guaranteed and there are multiple local optima which nodes can oscillate around. In order to achieve a more balanced coverage, we use the objective function in (43). In Fig. 12, we can once again see the cost-cutting benefits of using asynchronous methods. Fig. 13 shows that all three methods eventually achieve the same approximate level of coverage.

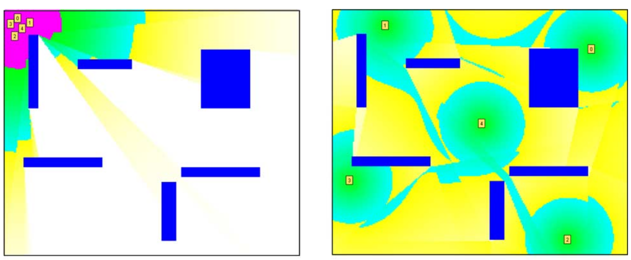
VI. Conclusion

The issue we have addressed in this paper is the extent to which communication among agents could be reduced in a cooperative optimization setting. This is crucial when the cooperating nodes have limited energy resources, as in the case of small inexpensive devices wirelessly networked. In addition,



(a) (b)

Fig. 10. Coverage control simulation result of a large network with 100 nodes. (a) Communication cost comparison; (b) obective function trajectories.



(a) (b)

Fig. 11. Initial and final deployment of an optimal coverage mission in an environment with polygonal obstacles. The blue rectangles represent obstacles. Areas covered with cyan color have joint detection probability $P(x, \mathbf{s}) \geq 0.97$. Green areas have $0.50 \leq P(x, \mathbf{s}) < 0.97$. Yellow and white areas have $0 \leq P(x, \mathbf{s}) < 0.50$. If the image is viewed in black and white, a lighter color indicates lower event detection probability in that area. (a) Initial deployment; (b) final deployment.

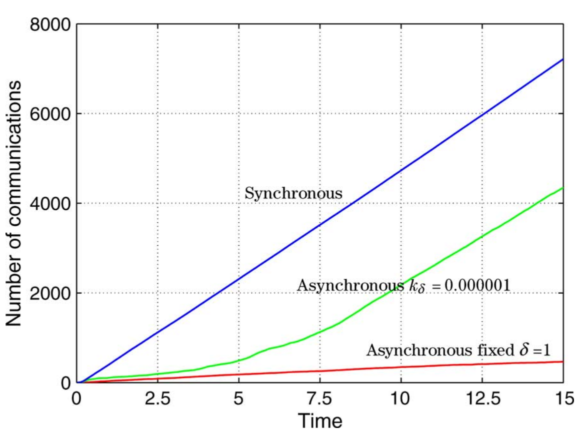


Fig. 12. Communication cost comparison of three distributed optimization algorithms in a nonsmooth problem.

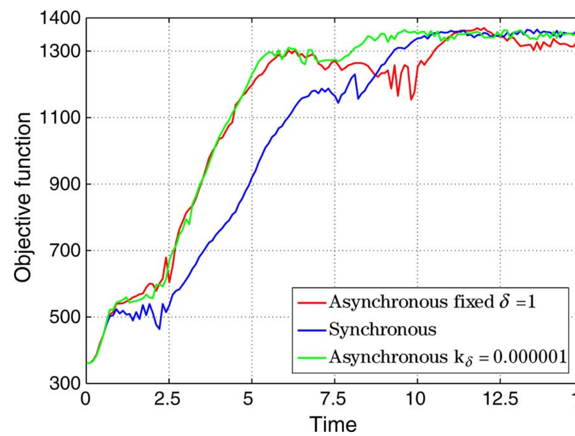


Fig. 13. Objective function trajectories of three distributed optimization algorithms in a nonsmooth problem.

limiting communication reduces the possibility of jamming or other sources of information corruption that can damage the cooperative process. The main result is showing that frequent communication among cooperating agents (nodes in a network) is not necessary in order to guarantee convergence of a process seeking to optimize a common objective. Specifically, we have proposed a scheme that limits communication events to "last resort only" when some state estimation error function at a node exceeds a threshold. We have proved that, under certain conditions, the convergence of a gradient-based asynchronous distributed algorithm employing this communication policy is still guaranteed. In addition, we have quantified the range of the two crucial parameters on which such convergence depends. Both cases where communication delays are negligible and where they are not have been analyzed. In the latter case, as expected, simulation results show convergence to be generally slower. We have also shown that our convergence analysis still applies when noise is explicitly included in the node state update process, provided that some additional conditions hold which appropriately bound this noise term.

We have applied this approach to a coverage control problem common in the deployment of wireless sensor networks and confirmed through numerical examples that limited asynchronous (event-driven) communication results in substantial energy savings which can prolong the life of such a network with no adverse effect on the optimization objective. Thus, from a practical standpoint, the proposed method can significantly benefit wireless networks set up to enable missions of multiple cooperating agents by reducing their use of energy with no compromise in performance. However, there is no guarantee that an event-driven communication approach always provides energy savings relative to time-driven mechanisms. What this approach offers is the flexibility to exploit intervals over which there is no need for a node to unnecessarily communicate minimal errors in its state trajectory.

There are several interesting questions we are currently addressing to improve upon the basic approach we have proposed.

First, we have assumed that communication among all nodes is always possible. To maintain this property when nodes are mobile, they need to be endowed with the capability to remain within a given feasible communication range and, if such loss does occur due to uncontrollable random events, it is necessary that nodes have the ability to re-establish a link. Regarding the estimation error function used to trigger the asynchronous communication events essential in this approach, we are currently studying the use of second derivative information (see also [22]) which may be critical when controlling the state of a node when it is sensitive to the state of other nodes.

References

[1] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *In Sensys*. New York: ACM Press, 2004, pp. 188–200.

[2] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1292–1302, Aug. 2004.

[3] J. Lawton, B. Young, and R. Beard, "A decentralized approach to elementary formation maneuvers," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 2728–2733.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. IEEE INFOCOM*, 2001, pp. 1380–1387.

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[6] A. Ganguli, J. Cortés, and F. Bullo, "Distributed deployment of asynchronous guards in art galleries," in *Proc. Amer. Control Conf.*, Minneapolis, MN, 2006, pp. 1416–1421.

[7] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proc. IEEE INFOCOM*, 2003, pp. 1293–1303.

[8] L. Mihaylova, T. Lefebvre, H. Bruyninckx, and K. Gadeyne, "Active sensing for robotics-a survey," in *Proc. 5th Intl. Conf. Numer. Methods Appl.*, Borovets, Bulgaria, 2002, pp. 316–324.

[9] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *Eur. J. Control*, vol. 11, no. 4–5, pp. 436–463, 2005.

[10] I. Hussein and D. Stipanovic, "Effective coverage control using dynamic sensor networks with flocking and guaranteed collision avoidance," in *Proc. Amer. Control Conf.*, 2007, pp. 3420–3425.

[11] P. Hokayem, D. Stipanovic, and M. Spong, "Dynamic coverage control with limited communication," in *Proc. Amer. Control Conf.*, 2007, pp. 4878–4883.

[12] M. DeGroot, "Reaching a consensus," *J. Amer. Stat. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[13] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, 1995.

[14] A. Jadbabaie, J. Lin, and S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[15] L. Moreau, "Stability of multi-agent systems with time-dependent communication links," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005.

[16] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[17] D. P. Bertsekas and J. N. Tsitsiklis*, Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[18] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with minimal communication," in *Proc. 47th IEEE Conf. Decision Control*, 2008, pp. 363–368.

[19] T. Shima, S. Rasmussen, and P. Chandler, "UAV team decision and control using efficient collaborative estimation," *ASME J. Dyn. Syst., Meas., Control*, vol. 129, no. 5, pp. 609–619, 2007.

[20] P. Wan and M. D. Lemmon, "Event triggered distributed optimization in sensor networks," in *Proc. 8th ACM/IEEE Int. Conf. Inform. Processing Sensor Networks*, San Francisco, CA, Apr. 2009, pp. 49–60.

[21] M. Zhong and C. G. Cassandras, "Distributed coverage control in sensor networks environments with polygonal obstacles," in *Proc. 17th IFAC World Congress*, 2008, pp. 4162–4167.

[22] G. Mathews, H. Durrant-Whyte, and M. Prokopenko, "Asynchronous gradient-based optimisation for team decision making," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 3145–3150.

Minyi Zhong (S'08) received the B.E. degree in electrical and information engineering from Shanghai Jiaotong University, Shanghai, China in 2005 and the M.S. degree from Boston University, Boston, MA , in 2009 where he is currently pursuing the Ph.D. in the Division of Systems Engineering and the Center for Information and Systems Engineering (CISE).

His research interests include distributed optimization and cooperative control, with applications to communication networks, sensor networks, manufacturing systems, intelligent vehicle systems and robotics.

Christos G. Cassandras (S'82–M'82–SM'91–F'96) received the B.S. degree from Yale University, New Haven, CT, in 1977, the M.S.E.E degree from Stanford University, Stanford, CA, in 1978, and the S.M. and Ph.D. degrees from Harvard University, Cambridge, MA, in 1979 and 1982, respectively.

From 1982 to 1984, he was with ITP Boston, Inc. where he worked on the design of automated manufacturing systems. From 1984 to 1996, he was a Faculty Member at the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. Currently, he is Head of the Division of Systems Engineering and Professor of Electrical and Computer Engineering, Boston University, Boston, MA and a founding member of the Center for Information and Systems Engineering (CISE). He specializes in the areas of discrete event and hybrid systems, stochastic optimization, and computer simulation, with applications to computer networks, sensor networks, manufacturing systems, transportation systems, and command/control systems. He has published over 250 papers in these areas, and five books.

Dr. Cassandras was Editor-in-Chief of the IEEE Transactions on Automatic Control from 1998 through 2009 and has served on several editorial boards and as Guest Editor for various journals. He also serves on the IEEE Control Systems Society Board of Governors. He received several awards, including the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize, and a 1991 Lilly Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi, and a Fellow of the IFAC.