Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Asynchronous Distributed Optimization with Minimal Communication

Minyi Zhong and **Christos G. Cassandras**
Division of Systems Engineering
and Center for Information and Systems Engineering
Boston University, Brookline, MA 02446
mzhong@bu.edu, cgc@bu.edu

Abstract— **We consider problems where multiple agents must 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 is asynchronous and can be minimized while still ensuring that the optimization process converges. We propose a 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. 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.**

Keywords: Cooperative Control, Distributed Systems, Distributed Optimization, 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, the nodes must share, at least partially, their state information. However, this may require a large amount of information ow. Moreover, we are interested in systems with wirelessly communicating nodes which are frequently small, inexpensive devices with limited resources. 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. 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 periodically exchange state information which is clearly inefficient and, in fact, unnecessary since often the state of a node may not have changed much or may have only changed in a predictable way. 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 minimize 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?"

This general setting 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],[5],[6],[7],[8]; 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 [9],[10],[11],[12].

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 control mechanism used by the ith node is a state update scheme of the form

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

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*; 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; however, we do not consider such dynamics and 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 much shorter than the time

The authors' work is supported in part by NSF under Grants DMI-0330171 and EFRI-0735974, by AFOSR under grants FA9550-04-1-0133 and FA9550-04-1-0208, and by DOE under grant DE-FG52-06NA27490.

between update events. A key difficulty 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 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 update events, so that the resulting mechanism is fully asynchronous.

We propose a scheme through which a node j maintains an *error function* of its actual state and its estimated state 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 or through unexpected state changes (e.g., if a mobile node encounters an obstacle). 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 [13], but our emphasis is on controlling the asynchronous occurrence of communication events through the threshold-based scheme outlined above in a way that drastically reduces the number of such events while still guaranteeing convergence.

Further, we apply this approach to a coverage control problem we have studied in prior work [8] in which a distributed optimization scheme based on (1) was used. However, it was assumed in [8] that all nodes have perfect state information by synchronizing update events with communication events for all nodes. This imposed significant communication costs. Here, we relax this synchronization requirement and limit communication events to occur according to the policy described above. We demonstrate that convergence to the optimum is attained with only a fraction of the original communication costs.

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 is presented in Section III. In Section IV we show how our approach applies to a coverage control problem for sensor networks and we conclude with Section V.

II. Asynchronous Distributed Optimization Framework

In the framework of N cooperating nodes that 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.

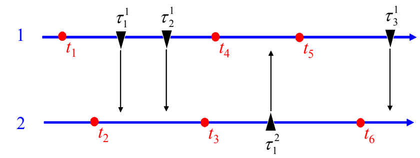

Fig. 1. State update and communication processes for two nodes.

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 periodic or according to some 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 we have $\mathcal{C}^1 = \{1, 4, 5\}$, and $\mathcal{C}^2 = \{2, 3, 6\}$. We assume that $d_i(\mathbf{s}(k)) = 0$ for all $k \notin \mathcal{C}^i$, i.e.,

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

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 $\tau_0^j = 0$. Depending on the network connectivity at that time, it is possible that only a subset of nodes is reached. We assume that at all times the state information broadcast by node j can reach any other node with negligible communication delay either directly or indirectly (in the latter case, through a sequence of transmissions), i.e., we assume that the underlying network is connected. Consider a state update time t_k with $k \in \mathcal{C}^i$. We are interested in the most recent communication event from a node $j \neq i$ and define

$$\tau^j(k) = \max\{\tau_n^j : \tau_n^j < t_k,\ n = 1, 2, \ldots\} \tag{2}$$

as the time of the most recent communication event at node j prior to a state update event at t_k. As an example, in Fig. 1 we see that 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 $\tau^1(6) = \tau^1(3) = \tau_2^1$. Regarding the policy used by node j to determine its communication events, we shall discuss this issue later in this section but 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 set

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

Thus, the state of node j communicated to other nodes at time $\tau^j(k)$ is 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 s_j 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.,

$$s_j^i(k) = x_j(\tau^j(k)) \tag{3}$$

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

$$s_j^i(k) = x_j(\tau^j(k)) + \frac{t_k - \tau^j(k)}{\Delta_j} \cdot \alpha_i \cdot d_j\left(x_j(\tau^j(k))\right) \tag{4}$$

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 $d_j\left(x_j(\tau^j(k))\right)$ 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 state dynamics of j are 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 $[\tau^j(k), t_k]$.

Now let us consider what criterion a node i might use in determining its communication events, recalling that our objective is to reduce communication costs. Certainly, if node i's state remains unchanged, there is no reason to communicate it. Moreover, 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. We require that $g(x_i(t), x_i^j(t))$ is continuous and satisfies

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

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

$$\tilde{k}_t^i = \max\left\{n : n \in \mathcal{C}^i,\ t_n \le t\right\}$$

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, $n = 1, 2, \ldots$, i.e., $x_i^j(\tau_n^{ij}) = x_i(\tau_n^{ij})$ assuming negligible communication delay. 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

$$\tau_n^{ij} = \inf\left\{t : g(x_i(t), x_i^j(t)) = \delta_i(\tilde{k}_t^i),\ t > \tau_{n-1}^{ij}\right\} \tag{6}$$

If, on the other hand, all nodes use the exact same estimation method, then we may set $\hat{x}_i(t) = x_i^j(t)$ for all $j \neq i$ and replace τ_n^{ij} in (6) 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 $\delta_i(\tilde{k}_t^i)$. In view of (5)-(6), we have, for all i, j, t,

$$g(x_i(t), x_i^j(t)) \le \delta_i(\tilde{k}_t^i) \tag{7}$$

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 $\left\|d_i\left(\mathbf{s}^i(k)\right)\right\|_2$, the Euclidean norm of the update direction at the kth update event henceforth denoted by $\|\cdot\|$. Thus, let

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

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

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

where $s_j^i(0) = x_j(0)$. Clearly, the computation in (8) requires only local information.

III. Convergence Analysis

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

$$s_i(k+1) = s_i(k) + \alpha d_i\left(\mathbf{s}^i(k)\right) \tag{10}$$

used by nodes $i = 1, \ldots, N$, where $d_i\left(\mathbf{s}^i(k)\right)$ is an update direction which satisfies $d_i\left(\mathbf{s}^i(k)\right) = 0$ for all $k \notin \mathcal{C}^j$. 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 $d_i\left(\mathbf{s}^i(k)\right)$. 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 [13]. The distinctive feature in our analysis is the presence of the controllable state communication process defined by (6), (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 (10) in a way that can ensure the minimum number of communication events while still guaranteeing convergence.

We begin with a number of assumptions, most of which are the ones commonly used in the analysis of distributed asynchronous algorithms [13].

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

This assumption imposes a bound on the state update frequency of every node. 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 nodes 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 $d(k) = [d_1(\mathbf{s}^i(k))', \ldots, d_N(\mathbf{s}^i(k))']'$. 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 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. There exists a positive constants K_4 such that the error function satisfies $\|x_i(t) - x_i^j(t)\| \leq K_4 g(x_i(t), x_i^j(t))$, for all i, j, t.

In the common case where $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|$, this is obviously satisfied with $K_4 = 1$.

Theorem 1: Under Assumptions 1-4 and the state update scheme (10), 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 $\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$.

The proof of Theorem 1 can be found in [14]. We will present some key steps of the proof here. First, we apply the "descent lemma" (Prop. A.32 in [13]) and get $H(\mathbf{s}(k+1)) \leq H(\mathbf{s}(k)) + \alpha \sum_{i=1}^{N} d_i(k)' \nabla_i H(\mathbf{s}(k)) + \frac{K_1}{2}\alpha^2 \|d(k)\|^2$. After using the conditions in the assumptions and policy (8), we obtain the following inequality

$$H(\mathbf{s}(0)) \geq \alpha \left(\frac{1}{K_3} - \frac{K_1}{2}\alpha - \frac{(1+B)}{2} K_1 K_4 K_\delta \sqrt{m} \right) \sum_{r=0}^{k} \|d(r)\|^2 \quad (11)$$

We can always select positive α and K_δ such that $1/K_3 - K_1\alpha/2 - (1+B) K_1 K_4 K_\delta \sqrt{m}/2 > 0$. Consequently, we have:

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

and finally $\lim_{k\to\infty} \|d(k)\| = 0$

Corollary 1. If $0 < \alpha < 2/K_1 K_3$, then

$$K_\delta < \frac{1}{(1+B) K_4 \sqrt{m}} \left(\frac{2}{K_1 K_3} - \alpha \right) \quad (12)$$

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

Proof: This follows directly from (11) by ensuring that the term $1/K_3 - K_1\alpha/2 - (1+B) K_1 K_4 K_\delta \sqrt{m}/2$ is positive.■

Note that (12) provides an upper bound of K_δ that guarantees convergence and, therefore, the smallest possible number of communication events 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 (10) 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 with $K_4 = 1$. It follows that the choice of α is $\alpha < 2/K_1$ and (12) leads to a K_δ arbitrarily close from below to $(2/K_1 - \alpha)[(1+B)\sqrt{m}]^{-1}$. Observe that this value is inversely proportional to $\sqrt{m} = \sqrt{N}$ if node states are scalar. 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, smaller step size α (slower change of states) allows us to choose a larger K_δ, which means greater tolerance to estimation error.

IV. Asynchronous Distributed Coverage Control

In this section, we apply the proposed asynchronous distributed framework to the coverage control problem encountered when a sensor network is called upon to determine the optimal positions of sensor nodes in order to maximize the probability of detecting events occurring in a given two-dimensional mission space.

A. *Problem Formulation*

Here we follow the coverage control problem formulation in [8] and brie y review it. 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 location is denoted by a $2N$-dimensional vector $\mathbf{s} = (s_1, \ldots, s_N)$.

The probability that sensor node i detects an event occurring at $x \in \Omega$ (assuming a clear line-of-sight between them), 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, 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

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

in which we use the locations of the sensor nodes as decision variables to maximize the probability of event detection in Ω. A *synchronous* gradient-based solution was obtained in [8] in which the next way point on the ith mobile sensor's trajectory is determined through

$$s_i(k+1) = s_i(k) + \alpha \frac{\partial H(\mathbf{s})}{\partial s_i}, \quad k = 0, 1, \ldots \tag{13}$$

The gradient can be evaluated by

$$\frac{\partial H(\mathbf{s})}{\partial s_i} = \int_{\Omega_i} R(x) \prod_{k \in \mathcal{B}_i} [1 - p_k(x, s_k)] \frac{dp_i(x, s_i)}{dd_i(x)} \frac{s_i - x}{d_i(x)} dx \tag{14}$$

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

The state update rule (13) allows a fully distributed implementation based on local information only. 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, (14) shows that node i needs the exact locations of all nodes in $\mathcal{B}_i$ in order to carry out (13) in a state update. As already mentioned, the communication involved in 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 [15] when the mission space contains obstacles.

B. *Asynchronous vs Synchronous Coverage Control*

We present numerical results based on simulations of the coverage control setting that may be found in an interactive Java-based simulation environment (along with instructions) at *http://codescolor.bu.edu*. We compare three versions of the 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).

The coverage control problem considered involves four nodes deployed into a rectangular mission space with uniform event density from the lower left corner. All nodes update their states at approximately the same frequency

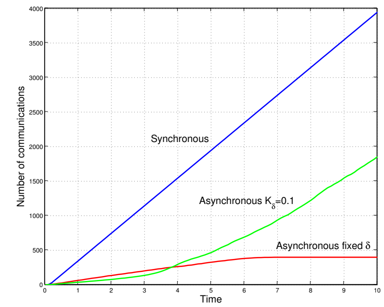

Fig. 2. Communication cost comparison of three distributed optimization algorithms

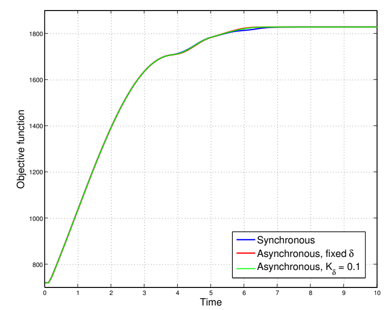

Fig. 3. Objective function trajectories of three distributed optimization algorithms

using the same step size in all three schemes. For the asynchronous versions, (3) is used for state estimation and $g(x_i(t), x_i^j(t)) = \|x_i(t) - x_i^j(t)\|$.

In Figs. 2 and 3, we compare these three algorithms. In Fig. 2, 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 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. Figure 4 shows the node trajectories for these three methods. Methods 1 and 3 converge to the same node configuration which is indicated by the black squares, while method 2 converges to a configuration close to it.

Next, in Figs. 5 and 6 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 the clear trend that a larger K_δ leads to fewer communication events. But we also notice in Fig. 6 that when $K_\delta = 5$, the objective function curve exhibits considerable oscillations before it converges. This suggests that K_δ is set too high and some "necessary" communication events between nodes have been omitted. In other words, when K_δ is increased, although convergence to a local optimum may

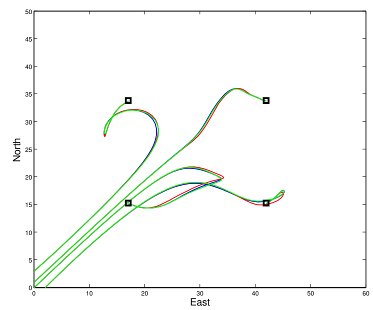

Fig. 4. Node trajectory comparison of three distributed optimization algorithms in a coverage control mission. See Fig. 3 for the legend.

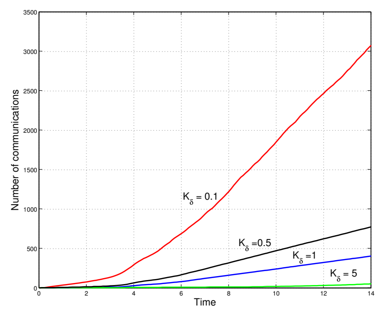

Fig. 5. Communication cost comparison of asynchronous distributed algorithm with diffent K_δ

still be guaranteed, so is the risk of slower convergence.

V. Conclusions and Future Work

The main result of this paper is showing that synchronized or asynchronous frequent communication among cooperating agents (nodes in a network) seeking to optimize a common objective is not necessary in order to guarantee convergence. Specifically, we have proposed a scheme that limits communication to instants when some state estimation error function at a node exceeds a threshold and proved that, under certain conditions, the convergence of a gradient-based fully asynchronous distributed algorithm is still guaranteed.

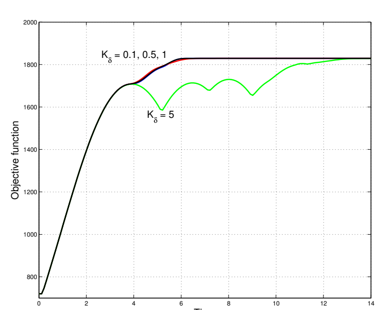

Fig. 6. Objective function trajectories of asynchronous distributed algorithm with diffent K_δ

In addition, we have quantified the range of the two crucial parameters on which such convergence depends. We have applied this approach to a coverage control problem common in the deployment of wireless sensor networks and confirmed through numerical examples that the effect of asynchronous limited communication is minimal on the optimization objective while resulting in very substantial energy savings which can prolong the life of such a network.

Our ongoing work is aimed at incorporating communication delays into our proposed framework, which are not expected to affect the basic convergence result. Moreover, we are considering enhancement of the estimation error function that guides communication events so as to use second derivative information (see also [16]) which may be critical when controlling the state of a node is sensitive to the state of other nodes.

References

[1] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. of the 2nd international conf. on Embedded networked sensor systems*, 2004.

[2] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative Control of Mobile Sensor Networks: Adaptive Gradient Climbing in a Distributed Environment," *IEEE Trans. Automatic Control*, vol. 49, no. 8, 2004.

[3] J. Lawton, B. Young, and R. Beard, "A decentralized approach to elementary formation maneuvers," in *Proc. of the 2000 IEEE International Conf. on Robotics and Automation*, 2000.

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

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 2, 2004.

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

[7] L. Mihaylova, T. Lefebvre, H. Bruyninckx, and K. Gadeyne, "Active sensing for robotics-a survey," in *Proc. of the 5th Intl. Conf. on Numerical Methods and Applications*, Borovets, Bulgaria, 2002.

[8] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4-5, pp. 436–463, 2005.

[9] M. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[10] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel Type of Phase Transition in a System of Self-Driven Particles," *Physical Review Letters*, vol. 75, no. 6, 1995.

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

[12] A. Nedic and A. Ozdaglar, "Distributed Asynchronous Subgradient Methods for Multi-agent Optimization," *IEEE Trans. Automatic Control*, 2007, to appear.

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

[14] M. Zhong and C. G. Cassandras, "Asynchronous Distributed Optimization," Boston University, Tech. Rep., 2008, http://codescolor.bu.edu/docs.html.

[15] ——, "Distributed coverage control in sensor networks environments with polygonal obstacles," in *Proc. of 17th IFAC World Congress*, 2008.

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