2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Decentralized Event-Driven Algorithms for Multi-Agent Persistent Monitoring Tasks

Nan Zhou[1], Christos G. Cassandras[1,2], Xi Yu[3], and Sean B. Andersson[1,3]

Abstract— **We address the issue of identifying conditions under which the centralized solution to the optimal multi-agent persistent monitoring problem can be recovered in a decentralized event-driven manner. In this problem, multiple agents interact with a finite number of targets and the objective is to control their movements in order to minimize an uncertainty metric associated with the targets. In one-dimensional settings, it has been shown that the optimal solution can be reduced to a simpler parametric one and that the behavior of agents under optimal control is described by a hybrid system. This hybrid system can be analyzed using Infinitesimal Perturbation Analysis (IPA) to obtain an on-line solution through an event-driven centralized gradient-based algorithm. We show that the IPA gradient can be recovered in a distributed manner based on local information, except for one event requiring communication from a non-neighbor agent. Simulation examples are included to illustrate the effectiveness of this "almost decentralized" algorithm and its fully decentralized counterpart where the aforementioned non-local event is ignored.**

I. INTRODUCTION

Systems consisting of cooperating mobile agents are often used to perform tasks such as coverage [1], surveillance [2], or environmental sampling [3]. A *persistent monitoring* task is one where agents must cooperatively monitor a dynamically changing environment that cannot be fully covered by a stationary team of agents (as in coverage control) [4]. Once the exploration process leads to the discovery of various "points of interest", then these become "targets" which need to be perpetually monitored. Thus, in contrast to sweep coverage and patrolling where *every* point in a mission space is of interest [5], the problem we address here focuses on a *finite number* of data sources or "targets".

In this setting, the agents interact with targets through their sensing capabilities which are normally dependent upon their physical distance from the target. The uncertainty state of a target increases when no agent is visiting it and decreases when it is being monitored by one or more agents (i.e., it is within their sensing range). The objective is to minimize an overall measure of target uncertainty states by controlling the movement of all agents in a cooperative manner. Unlike many other multi-agent systems modeled solely through a network of interconnected agents, here we have two networks, one whose nodes are agents and one whose nodes are targets. Since agents interact with targets, this interaction is modeled by establishing links between nodes belonging to the two different networks. Moreover, since agents are mobile and the overall graph topology is time-varying, the resulting complexity of this class of problems is significant. This has motivated approaches where rather than viewing these as agent-to-target assignment problems [6], [7] (which are computationally intensive), one treats them as trajectory design and optimization problems [4], [8].

[1]Division of Systems Engineering, [2]Department of Electrical and Computer Engineering, [3]Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA. E-mail: {nanzhou,cgc, xyu,sanderss}@bu.edu. * The work of Cassandras and Zhou is supported in part by NSF under grants ECCS-1509084, IIP-1430145, and CNS-1645681, by AFOSR under grant FA9550-15-1-0471, by DOE under grant 46100 and by MathWorks. The work of Andersson and Yu is supported in part by the NSF through grant ECCS-1509084 and CMMI-1562031.

In [9], we studied the persistent monitoring problem in 1D and showed that it can be formulated as an optimal control problem whose solution is parametric. In particular, every optimal agent trajectory is characterized by a finite number of points where the agent switches direction and by a dwell time at each such point. As a result, the behavior of agents under optimal control is described by a hybrid system. This allows us to make use of Infinitesimal Perturbation Analysis (IPA) [10], [11] to determine on-line the gradient of the objective function with respect to these parameters and to obtain a (possibly local) optimal trajectory. Our approach exploits IPA's event-driven nature to render it *scalable* in the number of *events* and not its state space.

The optimal controller developed in [9] is established based on the assumption of a centralized controller which provides information and coordinates all agents. Similar centralized controllers can be found in [3], [4], [12]. Clearly, a centralized controller can be energy-consuming and unreliable in adversarial environments. This motivates us to develop *decentralized* controllers by distributing functionality to agents so that each one acts upon local information or by communicating with only a set of neighbors. Such distributed algorithms have been derived and applied to coverage [1], formation [13], and consensus problems [14] where a static fully connected network is usually assumed. However, decentralization for persistent monitoring problems is particularly challenging due to the time-varying topology of agent network in which interactions between agents and the environment cannot be easily shared through the network.

In this paper, we identify explicit conditions under which the centralized solution to the persistent monitoring problem studied in [9] can be recovered through an "almost decentralized" and entirely event-driven manner. In particular, each agent uses (i) its own local information (to be precisely defined later), (ii) information (in the form of observable events) from its neighbors, and (iii) a single specific event type communicated by a non-neighbor agent. The decentralization result exploits the structure of the IPA gradient associated with each agent which turns out to depend only on a limited number of local events, except for one event requiring communication with a non-neighbor agent.

II. PROBLEM FORMULATION

We begin by reviewing the persistent monitoring model and problem formulation introduced in [9].

Agent dynamics. We consider N agents moving in a 1D mission space $[0, L]$. Each agent can control its speed and direction. The speed input is scaled and bounded in $[-1, 1]$. The position of each agent j is represented as $s_j(t) \in [0, L]$ with the dynamics:

$$\dot{s}_j(t) = u_j(t), \quad |u_j(t)| \leq 1, \quad \forall j = 1, 2 \ldots, N \qquad (1)$$

Agent sensing model. The ability of an agent to sense its environment is modeled by a function $p_j(x, s_j)$ that measures the probability that an event at location x is detected by agent j at $s_j(t)$. We assume that $p_j(x, s_j) = 1$ if $x = s_j$, and that $p_j(x, s_j)$ is monotonically non-increasing in the distance $\|x - s_j\|$, thus capturing the reduced effectiveness of a sensor over its sensing range denoted by r_j. Although our analysis is not affected by the precise sensing model $p_j(x, s_j)$, we will limit ourselves to a linear decay model as follows:

$$p_j(x, s_j) = \max\left\{0, 1 - \frac{\|x - s_j\|}{r_j}\right\} \tag{2}$$

Unlike the sweep coverage problem, here we consider a known finite set of targets located at $x_i \in [0, L]$, $i = 1, \ldots, M$. We set $p_j(x_i, s_j(t)) \equiv p_{ij}(s_j(t))$ for simplicity. The sensing capability of N agents on target i, assuming detection independence, can be captured by the joint detection probability function

$$P_i(\mathbf{s}(t)) = 1 - \prod_{j=1}^{N} (1 - p_{ij}(s_j(t))) \tag{3}$$

where we set $\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^{\mathrm{T}}$.

Target dynamics. We define uncertainty functions $R_i(t)$ associated with targets $i = 1, \ldots, M$, so that they have the following properties: (i) $R_i(t)$ increases with a prespecified rate A_i if $P_i(\mathbf{s}(t)) = 0$ (as shown in [9], this can be allowed to be a random process $\{A_i(t)\}$), (ii) $R_i(t)$ decreases with a fixed rate B_i if $P_i(\mathbf{s}(t)) = 1$ and (iii) $R_i(t) \geq 0$ for all t. It is then natural to model uncertainty dynamics associated with each target as follows:

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0,\ A_i \leq B_i P_i(\mathbf{s}(t)) \\ A_i - B_i P_i(\mathbf{s}(t)) & \text{otherwise} \end{cases} \tag{4}$$

where we assume that initial conditions $R_i(0)$ for all i are given and that $B_i > A_i > 0$ to ensure a strict decrease in $R_i(t)$ when $P_i(\mathbf{s}(t)) = 1$.

Optimal control problem. Our goal is to control the movement of agents through $u_j(t)$ in (1) so that the cumulative average uncertainty over all targets is minimized over a fixed time horizon T. Setting $\mathbf{u}(t) = [u_1(t), \ldots, u_N(t)]^{\top}$ we aim to solve the following optimal control problem:

$$\mathbf{P1}: \quad \min_{\mathbf{u}(t)} \quad J = \frac{1}{T}\int_0^T \sum_{i=1}^{M} R_i(t)dt \tag{5}$$

subject to the agent dynamics (1) and target uncertainty dynamics (4). Generally, the classical solution of (5) involves solving a Two Point Boundary Value Problem which requires global information of all agents and targets. In this paper, we will limit the information of each agent to itself and its neighbors and study whether this objective function can be optimized in a distributed manner.

Limited information model for decentralization. In our model, an agent is capable of observing information within its sensing range, specifically the state $R_i(t)$ of all targets i such that $p_{ij}(s_j(t)) > 0$. Moreover, agents can communicate with their neighboring agents to acquire information such as agent positions, speeds, and the states of targets which are within their sensing ranges. In contrast to traditional multi-agent systems modeled through a network of agents, in the persistent monitoring setting agents move to interact with targets. Therefore, the network model includes both agents and targets as shown in Fig. 1. We need to revisit the concept of neighborhood, accounting as well for the time-varying network topology.

Definition 1. *The agent neighborhood of agent j is the set* $\mathcal{A}_j(t) = \{k : \|s_k(t) - s_j(t)\| \leq r_c,\ k \neq j,\ k = 1, \ldots, N\}$.

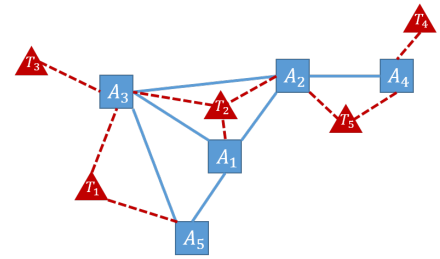

Fig. 1: Agent-target network. Red triangles are targets and blue squares are agents.

This is a conventional definition of neighbors in multi-agent systems, where r_c is a communication range. As an example, in Fig. 1, $\mathcal{A}_1 = \{A_2, A_3, A_5\}$.

Definition 2. *The target neighborhood of agent j is the set* $\mathcal{T}_j(t) = \{i : |x_i - s_j(t)| \leq r_j,\ i = 1, \ldots, M\}$.

This includes all targets which are within agent j's sensing range. In Fig. 1, $\mathcal{T}_3 = \{T_1, T_2, T_3\}$. Assuming the agents are homogeneous with a common sensing range r, we require that $r_c \geq 2r$ in order to establish communication among agents that are sensing the same target.

Definition 3. *The agent neighborhood of target i is the set* $\mathcal{B}_i(t) = \{j : |s_j(t) - x_i| \leq r_j,\ j = 1, \ldots, N\}$.

This set captures all the neighbor agents of target i. In Fig. 1, $\mathcal{B}_2 = \{A_1, A_2, A_3\}$. Using Definition 3, the joint sensing probability in (3) can be rewritten as:

$$P_i(\mathbf{s}(t)) = 1 - \prod_{j \in \mathcal{B}_i(t)} (1 - p_{ij}(s_j(t))) \tag{6}$$

where $\mathcal{B}_i(t) \subseteq \{1, \ldots, N\}$. We further define

$$\mathcal{N}_{ij}(t) = \mathcal{B}_i(t) \setminus \{j\} \tag{7}$$

to indicate the "collaborators" of agent j in sensing target i. Note that $\mathcal{N}_{ij}(t) = \{k : k \in A_j(t) \text{ and } k \in B_i(t)\}$, thus capturing a neighbor of agent j and target i at the same time.

Our limited information model restricts observations of each agent to the agent's sensing range. However, any agent j is allowed to communicate with its neighbors in $\mathcal{A}_j(t)$. Therefore, the local information of an agent is the union of the observations of agent j and the observations of agents $k \in \mathcal{A}_j(t)$. In Section IV-A, we will explicitly define the precise meaning of "information" above to consist of observable events such as "agent stops" or "target state becomes $R_i(t) = 0$". In Section V, we will show how **P1** can be solved by each agent under this limited information model as opposed to the centralized one in [9].

III. From Optimal Control to Parametric Optimization

We begin by defining the state vector $\mathbf{x}(t) = [R_1(t), \ldots R_M(t), s_1(t) \ldots s_N(t)]$ and associated costate vector $\lambda = [\lambda_1(t), \ldots, \lambda_M(t), \lambda_{s_1}(t), \ldots, \lambda_{s_N}(t)]$. Due to the discontinuity in the dynamics of $R_i(t)$ in (4), the optimal state trajectory may contain a boundary arc when $R_i(t) = 0$ for some i; otherwise, the state evolves in an interior arc. Using (1) and (4), the Hamiltonian is

$$H(\mathbf{x}, \lambda, \mathbf{u}) = \sum_{i=1}^{M} R_i(t) + \sum_{i=1}^{M} \lambda_i(t)\dot{R}_i(t) + \sum_{j=1}^{N} \lambda_{s_j}(t)u_j(t) \tag{8}$$

Applying the Pontryagin Minimum Principle to (8) with $\mathbf{u}^\star(t)$, denoting an optimal control, a necessary condition for optimality is

$$u_j^*(t) = \begin{cases} 1 & \text{if } \lambda_{s_j}(t) < 0 \\ -1 & \text{if } \lambda_{s_j}(t) > 0 \end{cases} \tag{9}$$

Note that there exists a possibility that $\lambda_{s_j}(t) = 0$ over some finite singular intervals [15], in which case $u_j^*(t)$ may take values in $\{-1, 0, 1\}$.

Since the optimal control structure is fully characterized by $u_j^*(t) \in \{1, 0, -1\}$, we can parameterize the optimal trajectory so as to determine (i) control switching points in $[0, L]$, where an agent switches its control from ± 1 to ∓ 1 or possibly 0 and (ii) corresponding dwell times so that the cost in (5) is minimized. In other words, the optimal trajectory of each agent j is fully captured by two parameter vectors: switching points $\boldsymbol{\theta}_j = [\theta_{j1}, \theta_{j2} ... \theta_{j\Gamma}]$ and dwell times $\mathbf{w}_j = [w_{j1}, w_{j2} ... w_{j\Gamma'}]$ where Γ and Γ' depend on the given time horizon T. This defines a hybrid system with state dynamics (1), (4). The agent and target dynamics remain unchanged in between events, i.e., the points $\theta_{j1}, \ldots, \theta_{j\Gamma}$ above and instants when $R_i(t)$ switches from > 0 to 0 or vice versa. Therefore, the overall cost function (5) can be parametrically expressed as $J(\boldsymbol{\theta}, \mathbf{w})$ and rewritten as the sum of costs over corresponding inter-event intervals over a given time horizon:

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}, \mathbf{w} \geq \mathbf{0}} \quad J(\boldsymbol{\theta}, \mathbf{w}) = \frac{1}{T} \sum_{k=0}^{K} \int_{\tau_k(\boldsymbol{\theta}, \mathbf{w})}^{\tau_{k+1}(\boldsymbol{\theta}, \mathbf{w})} \sum_{i=1}^{M} R_i(t) dt \tag{10}$$

This allows us to apply IPA to determine a gradient $\nabla J(\boldsymbol{\theta}, \mathbf{w})$ with respect to those parameters of the agent trajectories and apply any standard gradient descent algorithm to obtain an optimal solution.

IV. Infinitesimal Perturbation Analysis

We briefly review the IPA framework for general stochastic hybrid systems as presented in [10]. Let $\{\tau_k(\theta)\}$, $k = 1, \ldots, K$, denote the occurrence times of all events in the state trajectory of a hybrid system with dynamics $\dot{x} = f_k(x, \theta, t)$ over an interval $[\tau_k(\theta), \tau_{k+1}(\theta))$, where $\theta \in \Theta$ is some parameter vector and Θ is a given compact, convex set. Set $\tau_0 = 0$ and $\tau_{K+1} = T$. We use the Jacobian matrix notation: $x'(t) \equiv \frac{\partial x(\theta, t)}{\partial \theta}$ and $\tau_k' \equiv \frac{\partial \tau_k(\theta)}{\partial \theta}$, for all state and event time derivatives. It is shown in [10] that

$$\frac{d}{dt} x'(t) = \frac{\partial f_k(t)}{\partial x} x'(t) + \frac{\partial f_k(t)}{\partial \theta}, \tag{11}$$

for $t \in [\tau_k, \tau_{k+1})$ with boundary condition:

$$x'(\tau_k^+) = x'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)] \tau_k' \tag{12}$$

for $k = 1, ..., K$. In order to complete the evaluation of $x'(\tau_k^+)$ in (12), we need to determine τ_k'. If the event at τ_k is *exogenous* (i.e., independent of θ), $\tau_k' = 0$. However, if the event is *endogenous*, there exists a continuously differentiable function $g_k : \mathbb{R}^n \times \Theta \to \mathbb{R}$ such that $\tau_k = \min\{t > \tau_{k-1} \ : \ g_k(x(\theta, t), \theta) = 0\}$ and

$$\tau_k' = -[\frac{\partial g_k}{\partial x} f_k(\tau_k^-)]^{-1} (\frac{\partial g_k}{\partial \theta} + \frac{\partial g_k}{\partial x} x'(\tau_k^-)) \tag{13}$$

as long as $\frac{\partial g_k}{\partial x} f_k(\tau_k^-) \neq 0$ (details may be found in [10]).

In our setting, following (10) the gradient for each agent j denoted by $\nabla_j J(\boldsymbol{\theta}, \mathbf{w}) = [\frac{\partial J(\boldsymbol{\theta}, \mathbf{w})}{\partial \boldsymbol{\theta}_j}, \frac{\partial J(\boldsymbol{\theta}, \mathbf{w})}{\partial \mathbf{w}_j}]^{\mathrm{T}}$ is

$$\nabla_j J(\boldsymbol{\theta}, \mathbf{w}) = \frac{1}{T} \sum_{k=0}^{K} \sum_{i=1}^{M} \int_{\tau_k(\boldsymbol{\theta}, \mathbf{w})}^{\tau_{k+1}(\boldsymbol{\theta}, \mathbf{w})} \nabla_j R_i(t) dt \tag{14}$$

where $\nabla_j R_i(t) = [\frac{\partial R_i(t)}{\partial \boldsymbol{\theta}_j}, \frac{\partial R_i(t)}{\partial \mathbf{w}_j}]^{\mathrm{T}}$.

We begin by deriving the gradient within any inter-event interval $[\tau_k, \tau_{k+1})$ when the dynamics of both agent j and target i remain unchanged. We proceed with the derivation of $\frac{\partial R_i(t)}{\partial \boldsymbol{\theta}_j}$, since $\frac{\partial R_i(t)}{\partial \mathbf{w}_j}$ can be derived in a similar way.

It follows from (11), observing that the first term vanishes since $f_k(t) = \dot{R}_i(t)$ is not an explicit function of $R_i(t)$, that $\frac{d}{dt} \frac{\partial R_i(t)}{\partial \boldsymbol{\theta}_j} = \frac{\partial \dot{R}_i(t)}{\partial \boldsymbol{\theta}_j}$. Then, in view of (4), we have for all $t \in [\tau_k, \tau_{k+1})$:

$$\frac{\partial R_i(t)}{\partial \boldsymbol{\theta}_j} = \begin{cases} \dfrac{\partial R_i(\tau_k^+)}{\partial \boldsymbol{\theta}_j} & \text{if } \dot{R}_i(t) = 0 \\ \dfrac{\partial R_i(\tau_k^+)}{\partial \boldsymbol{\theta}_j} - B_i \displaystyle\int_{\tau_k}^{t} \frac{\partial P_i(\mathbf{s}(\tau))}{\partial \theta_j} d\tau & \\ \qquad \text{if } \dot{R}_i(t) = A_i - B_i P_i(\mathbf{s}(t)) & \end{cases} \tag{15}$$

The integrand in (15) is obtained from (6):

$$\frac{\partial P_i(\mathbf{s}(\tau))}{\partial \theta_j} = \frac{\partial p_{ij}(s_j(\tau))}{\partial s_j} \frac{\partial s_j(\tau)}{\partial \boldsymbol{\theta}_j} \prod_{\substack{g \in \mathcal{B}_i(\tau) \\ g \neq j}} [1 - p_{ig}(s_g(\tau))] \tag{16}$$

Note that $\frac{\partial p_{ij}(s_j(\tau))}{\partial s_j}$ is piece-wise constant and takes values in $\{0, \pm \frac{1}{r_j}\}$ depending on $|s_j(t) - x_i|$ and r_j (see agent sensing mode (2)). We can, therefore, factor the constant $\frac{\partial p_{ij}(s_j(\tau))}{\partial s_j}$ out of the integral in (15). As for the term $\frac{\partial s_j(\tau)}{\partial \boldsymbol{\theta}_j}$, we apply (11) and (1) to obtain $\frac{d}{dt} \frac{\partial s_j(\tau)}{\partial \boldsymbol{\theta}_j} = 0$. Therefore,

$$\frac{\partial s_j(\tau)}{\partial \boldsymbol{\theta}_j} = \frac{\partial s_j(\tau_k^+)}{\partial \boldsymbol{\theta}_j} \tag{17}$$

which is also a constant. The product term in (16) captures the contributions from all agents other than j in monitoring target i. It can be restricted to the set $\mathcal{N}_{ij}(t)$ defined in (7), since for any agent $g \notin \mathcal{N}_{ij}(t)$, $p_{ig}(s_g(t)) = 0$. For notational simplicity, we define the integral of this term over $[\tau_k, t)$, $t < \tau_{k+1}$, as:

$$G_{ij}(t) = \int_{\tau_k}^{t} \prod_{g \in \mathcal{N}_{ij}(\tau)} [1 - p_{ig}(s_g(\tau))] \, d\tau \tag{18}$$

which can be interpreted as a "collaboration factor" involving all agents in $\mathcal{N}_{ij}(\tau)$. This is affected by an agent leaving or joining the neighbor set $\mathcal{N}_{ij}(\tau)$ which motivates defining an event associated with such changes in Section IV-A.

Through the above analysis, the derivative $\frac{\partial R_i(t)}{\partial \boldsymbol{\theta}_j}$, $i = 1, \ldots, M$, over any inter-event interval $[\tau_k, \tau_{k+1})$ becomes:

$$\frac{\partial R_i(t)}{\partial \boldsymbol{\theta}_j} = \frac{\partial R_i(\tau_k^+)}{\partial \boldsymbol{\theta}_j} - \begin{cases} 0 & \text{if } R_i(t) = 0, \ A_i \leq B_i P_i(\mathbf{s}(t)) \\ B_i \frac{\partial p_{ij}(s_j(\tau_k^+))}{\partial s_j} \frac{\partial s_j(\tau_k^+)}{\partial \boldsymbol{\theta}_j} G_{ij}(t) & \text{otherwise} \end{cases} \tag{19}$$

A similar derivation can be applied to $\frac{\partial R_i(t)}{\partial \mathbf{w}_j}$ and gives:

$$\frac{\partial R_i(t)}{\partial \mathbf{w}_j} = \frac{\partial R_i(\tau_k^+)}{\partial \mathbf{w}_j} - \begin{cases} 0 & \text{if } R_i(t) = 0, \ A_i \leq B_i P_i(\mathbf{s}(t)) \\ B_i \frac{\partial p_{ij}(s_j(\tau_k^+))}{\partial s_j} \frac{\partial s_j(\tau_k^+)}{\partial \mathbf{w}_j} G_{ij}(t) & \text{otherwise} \end{cases} \tag{20}$$

TABLE I: Events in agent-target system

Event Name	Description
ρ_i^0	$R_i(t)$ hits 0
ρ_i^+	$R_i(t)$ leaves 0
π_{ij}^0	$p_{ij}(s_j(t))$ hits 0
π_{ij}^+	$p_{ij}(s_j(t))$ leaves 0
$\nu_j^{(1,0)}$	$u_j(t)$ switches from 1 to 0
$\nu_j^{(-1,0)}$	$u_j(t)$ switches from -1 to 0
$\nu_j^{(0,1)}$	$u_j(t)$ switches from 0 to 1
$\nu_j^{(0,-1)}$	$u_j(t)$ switches from 0 to -1
$\nu_j^{(1,-1)}$	$u_j(t)$ switches from 1 to -1
$\nu_j^{(-1,1)}$	$u_j(t)$ switches from -1 to 1
Δ_{ij}^+	$\mathcal{N}_{ij}(\tau^+) = \mathcal{N}_{ij}(\tau^-) \cup \{k\}$, $k \notin \mathcal{N}_{ij}(\tau^-)$
Δ_{ij}^-	$\mathcal{N}_{ij}(\tau^+) = \mathcal{N}_{ij}(\tau^-) \setminus \{k\}$, $k \in \mathcal{N}_{ij}(\tau^-)$

Note: events in the table include all $i = 1, \ldots, M$ and $j = 1, \ldots, N$

A. Events in the hybrid system

We now define as "events" all switches in the hybrid system which can result in changes in the derivatives in (19) and (20) so we can apply (12) to determine the initial conditions $\frac{\partial R_i(\tau_k^+)}{\partial \boldsymbol{\theta}_j}$ and $\frac{\partial R_i(\tau_k^+)}{\partial \mathbf{w}_j}$ at $t = \tau_k$, as well as the terms $\frac{\partial s_j(\tau_k^+)}{\partial \boldsymbol{\theta}_j}$ and $\frac{\partial s_j(\tau_k^+)}{\partial \mathbf{w}_j}$.

We classify events into four categories. In what follows, we define all events types and their corresponding effects on (19) and (20) and summarize them in Table I.

Event type I: switches in target dynamics $\dot{R}_i(t)$. Referring to (4), when $R_i(t)$ either reaches zero or leaves zero, the IPA derivative switches between the two branches in (15). We denote the former event as ρ_i^0 and the latter as ρ_i^+ for all $i = 1, \ldots, M$ (see Table I). When such events occur, the dynamics of $s_j(t)$ in (1) remain unchanged, so it follows from (12) that $\nabla_j s_j(\tau_k^-) = \nabla_j s_j(\tau_k^+)$. However, the target dynamics switch between $\dot{R}_i = A_i - B_i P_i(\mathbf{s}(t))$ and $\dot{R}_i = 0$ and cause discontinuities in $\nabla_j R_i(t)$ as follows.

Event ρ_i^0: This event causes a transition from $\dot{R}_i(t) = A_i - B_i P_i(\mathbf{s}(t))$, $t < \tau_k$ to $\dot{R}_i(t) = 0$, $t \geq \tau_k$. It is an endogenous event because its occurrence depends on the parameters $\boldsymbol{\theta}, \mathbf{w}$ which dictate switches in $\mathbf{s}(t)$. We first evaluate τ_k' from (13) with $g_k(R_i(t), t) = R_i(t) = 0$ to get $\tau_k' = -\frac{\nabla_j R_i(\tau_k^-)}{A_i - B_i P_i(\mathbf{s}(\tau_k^-))}$ and then apply (12) to obtain $\nabla_j R_i(\tau_k^+) = \nabla_j R_i(\tau_k^-) + \left[A_i - B_i P_i(\mathbf{s}(\tau_k^-)) - 0\right] \tau_k'$. Replacing τ_k', we get

$$\nabla_j R_i(\tau_k^+) = 0 \quad \text{if event } \rho_i^0 \text{ occurs at } \tau_k \tag{21}$$

Event ρ_i^+: This event causes a transition from $\dot{R}_i(t) = 0$, $t < \tau_k$ to $\dot{R}_i(t) = A_i - B_i P_i(\mathbf{s}(t))$, $t \geq \tau_k$. It is easy to see that the dynamics in both (1) and (4) are continuous when this happens and since $A_i - B_i P_i(\mathbf{s}(\tau_k)) = 0$ we have $\dot{R}_i(\tau_k^-) = \dot{R}_i(\tau_k^+) = 0$. It follows from (12) that $\nabla_j R_i(\tau_k^+) = \nabla_j R_i(\tau_k^-)$. Moreover, since $R_i(t) = 0$, $\dot{R}_i(t) = 0$, $t < \tau_k$, we have $\nabla_j R_i(\tau_k^-) = 0$ and we get

$$\nabla_j R_i(\tau_k^+) = 0 \quad \text{if event } \rho_i^+ \text{ happens at } \tau_k \tag{22}$$

Remark 1: Combining (21) and (22) with (19) and (20), we conclude that a ρ_i^0 event occurring at $t = \tau_k$ resets the value of $\nabla_j R_i(t)$ to $\nabla_j R_i(t) = 0$ for all $j = 1, \ldots, N$ regardless of the value $\nabla_j R_i(\tau_k^-)$ and the state of the agents. Moreover, $R_i(t) = 0$ and $\nabla_j R_i(t) = 0$ for $t > \tau_k$ until the next ρ_i^+ event occurs.

Event type II: switches in agent sensing $p_{ij}(s_j(t))$. These events trigger a switch in $\frac{\partial p_{ij}(s_j(t))}{\partial s_j}$ from $\pm\frac{1}{r_j}$ to 0 or vice versa in (19) and (20). We denote the former event as π_{ij}^0 and the latter as π_{ij}^+. The dynamics in both (1) and (4) remain unchanged when this happens (due to the continuity of the sensing function $p_{ij}(s_j(t))$) and it follows from (12) that $\nabla_j R_i(\tau_k^+) = \nabla_j R_i(\tau_k^-)$ and $\nabla_j s_j(\tau_k^+) = \nabla_j s_j(\tau_k^-)$.

Event type III: switches in agent dynamics $\dot{s}_j(t)$. Referring to (1), these are events that cause a switch in the optimal control values $u_j^*(\tau_k)$: (i) $\pm 1 \to 0$, (ii) $0 \to \pm 1$, and (iii) $\pm 1 \to \mp 1$. We denote these events as $\nu_j^{(1,0)}, \nu_j^{(-1,0)}, \nu_j^{(0,1)}, \nu_j^{(0,-1)}, \nu_j^{(-1,1)}, \nu_j^{(1,-1)}$ using the general notation $\nu_j^{(*,*)}$ with the superscript corresponding to the six total possible control switches. The effect of these events in (19) and (20) is through possible discontinuities in the terms $\frac{\partial s_j(t)}{\partial \boldsymbol{\theta}_j}$ and $\frac{\partial s_j(t)}{\partial \mathbf{w}_j}$ at $t = \tau_k$. Clearly, the gradient cannot be affected by future events, so we consider all prior and current control switches indexed by $l = 1, 2..., \xi$ where ξ is the current control switch and θ_{jl}, w_{jl} are the l-th switching point and dwelling time respectively. These agent control switches are endogenous events with switching functions $g_k(s_j(t), t) = s_j - \theta_{jl} = 0$. We can now apply (12) and (13) to (1), similar to the derivation for type I events. We omit the details and present the final results.

Events $\nu_j^{(1,0)}, \nu_j^{(-1,0)}$: These are switches such that $u_j(\tau_k^-) = \pm 1$, $u_j(\tau_k^+) = 0$ and we get

$$\frac{\partial s_j}{\partial \theta_{jl}}(\tau_k^+) = \begin{cases} 1 & \text{if } l = \xi \\ 0 & \text{if } l < \xi \end{cases} \tag{23}$$

$$\frac{\partial s_j}{\partial \omega_{jl}}(\tau_k^+) = 0 \quad \text{for all } l \leq \xi \tag{24}$$

Events $\nu_j^{(0,1)}$, $\nu_j^{(0,-1)}$: These are switches such that $u_j(\tau_k^-) = 0$, $u_j(\tau_k^+) = \pm 1$ and we get

$$\frac{\partial s_j}{\partial \theta_{jl}}(\tau_k^+) = \begin{cases} \frac{\partial s_j}{\partial \theta_{jl}}(\tau_k^-) - u_j(\tau_k^+) sgn\big(\theta_{j\xi} - \theta_{j(\xi-1)}\big) & \text{if } l = \xi \\ \frac{\partial s_j}{\partial \theta_{jl}}(\tau_k^-) - u_j(\tau_k^+)\Big[sgn(\theta_{jl} - \theta_{j(l-1)}) \\ \qquad - sgn(\theta_{j(l+1)} - \theta_{jl})\Big] & \text{if } l < \xi \end{cases} \tag{25}$$

$$\frac{\partial s_j}{\partial w_{jl}}(\tau_k^+) = -u_j(\tau_k^+) \quad \text{for all } l \leq \xi \tag{26}$$

Events $\nu_j^{(-1,1)}, \nu_j^{(1,-1)}$: These are switches such that $u_j(\tau_k^-) = \pm 1$, $u_j(\tau_k^+) = \mp 1$ so that a dwell time is not involved and we get

$$\frac{\partial s_j}{\partial \theta_{jl}}(\tau_k^+) = \begin{cases} 2 & \text{if } l = \xi \\ -\frac{\partial s_j}{\partial \theta_{jl}}(\tau_k^-) & \text{if } l < \xi \end{cases} \tag{27}$$

Remark 2: Observe that $\nabla_j s_j(t)$ is independent of the states of other agents $k \neq j$.

Event type IV: changes in neighbor sets $\mathcal{N}_{ij}(t)$. These events change the topology of the agent-target network by altering the neighbors of agent j, hence affecting the value of $G_{ij}(t)$ in (18) which in turn affects (19) and (20). We denote by Δ_{ij}^+ the event causing the addition of an agent to the neighbor set $\mathcal{N}_{ij}(t)$ and by Δ_{ij}^- the event causing the removal of an agent from the neighbor set $\mathcal{N}_{ij}(t)$. However, the dynamics of both $R_i(t)$ and $s_j(t)$ remain unchanged when these events occur. Due to the continuity of the sensing function $p_{ig}(s_g(\tau))$ in (18), the addition/removal of an agent g to/from the set $\mathcal{N}_{ij}(\tau)$ does not affect the continuity of $G_{ij}(t)$, which implies $\nabla_j R_i(\tau_k^+) = \nabla_j R_i(\tau_k^-)$ as well as $\nabla_j s_j(\tau_k^+) = \nabla_j s_j(\tau_k^-)$.

The set of all events defined above and summarized in Table I is denoted by $\mathcal{E}$. Furthermore, we define the set of all type III events of the form $\nu_j^{(*,*)}$ as the *agent event set* $\mathcal{E}^A$ and the set of all other events (type I, III, and IV) as the *target event set* $\mathcal{E}^T$. The subset of $\mathcal{E}^A$ that contains only events related to agent j is denoted by $\mathcal{E}_j^A$. Similarly, the subset of $\mathcal{E}^T$ that contains only events related to target i is denoted by $\mathcal{E}_i^T$. We then have:

Definition 4. *The local event set of any agent j is the union of agent events $\mathcal{E}_j^A$ and target events $\mathcal{E}_i^T$ for all $i \in \mathcal{T}_j(t)$:*

$$\mathcal{E}_j(t) = \mathcal{E}_j^A \bigcup_{i\in\mathcal{T}_j(t)} \mathcal{E}_i^T \tag{28}$$

In contrast, the global event set for agent j includes all non-neighboring target events in $\mathcal{E}_i^T$ for all $i \notin \mathcal{T}_j$ and non-neighboring agent events $\mathcal{E}_k^A$, for all $k \notin \mathcal{A}_j$.

Definition 5. *The local information set of any agent j is the union of its local event set and those of its neighbors in $\mathcal{N}_{ij}(t)$ for all $i \in \mathcal{T}_j(t)$:*

$$I_j(t) = \mathcal{E}_j(t) \bigcup_{k\in\mathcal{N}_{ij}(t), i\in\mathcal{T}_j(t)} \mathcal{E}_k(t). \tag{29}$$

This includes all local information necessary for agent j to evaluate the IPA gradient $\nabla_j R_i(t)$ for $i \in \mathcal{T}_j(t)$. Observe that agent j does not need to communicate with all its neighbors in $\mathcal{A}_j(t)$, but only a subset which includes those neighbors sharing the same target(s) as j since $\bigcup_{i\in\mathcal{T}_j(t)} \mathcal{N}_{ij}(t) \subseteq \mathcal{A}_j(t)$.

Remark 3: It is clear from the analysis thus far, that IPA is entirely *event-driven*, since all gradient updates happen exclusively at events occurring at times $\tau_k(\boldsymbol{\theta}, \mathbf{w})$, $k = 1, 2, \ldots$. Thus, this approach scales with the number of events characterizing the hybrid system, and not its (generally much larger) state space.

V. Decentralized Gradient Evaluation and Optimization

We begin with the following lemma which asserts that the gradient $\nabla_j R_i(t)$ takes a very simple form as long as $i \notin \mathcal{T}_j(t)$, i.e., while target i cannot be sensed by agent j.

Lemma 1. *Let $t \in [t_1, t_2]$ such that $i \notin \mathcal{T}_j(t)$. Then,*

1) *If $R_i(t) > 0$ for all $t \in [t_1, t_2]$, then*

$$\nabla_j R_i(t) = \nabla_j R_i(t_1^+) \tag{30}$$

2) *If there exists an event ρ_i^0 at $\tau \in (t_1, t_2)$, then*

$$\nabla_j R_i(t) = \begin{cases} \nabla_j R_i(t_1^+) & t \in [t_1, \tau) \\ 0 & t \in [\tau, t_2] \end{cases} \tag{31}$$

Due to space limitations all proofs (some of which have technical details) are omitted and can be found in [16].

Corollary 1. *$\nabla_j R_i(t)$ is independent of events ρ_i^+ for $i \notin \mathcal{T}_j(t)$.*

Lemma 1 and its Corollary imply that agent j does not need any knowledge of non-neighboring target events except for ρ_i^0 with $i \notin \mathcal{T}_j(t)$ in order to evaluate its gradient. We can further establish that the gradient $\nabla_j J(\boldsymbol{\theta}, \mathbf{w})$ along the agent trajectory is affected by only local events in $I_j(t)$, as defined in (29), and a small subset of global events.

Lemma 2. *A sufficient event set to evaluate $\nabla_j J(\boldsymbol{\theta}, \mathbf{w})$ is $I_j(t) \cup \{\rho_i^0 : i \notin \mathcal{T}_j(t)\}$.*

Remark 4: Although an event ρ_i^0 for $i \notin \mathcal{T}_j(t)$ is non-local to agent j, it must be observed by at least one agent $k \neq j$ such that $i \in \mathcal{T}_k(t)$. This is because ρ_i^0 at some time τ_k can only take place if one or more agents in its neighborhood cause a transition from $R_i(\tau_k^-) > 0$ to $R_i(\tau_k) = 0$ in (4). Therefore, such events can be communicated to agent j through the agent network, possibly with some delays. The implication of Lemma 2 is an "almost decentralized" algorithm in which each agent optimizes its trajectory through the gradient $\nabla_j J(\boldsymbol{\theta}, \mathbf{w})$ using only agent local information; the only exception is occasional target uncertainty depletion events transmitted to it from other agents.

Returning to the parametric optimization problem (10), a centralized solution was obtained in [9] using the IPA gradients in (19) and (20) and a standard gradient descent scheme to optimize the parameter vector $[\boldsymbol{\theta}, \mathbf{w}]^{\mathrm{T}}$ as follows:

$$\left[\boldsymbol{\theta}^{l+1}, \mathbf{w}^{l+1}\right]^{\mathrm{T}} = \left[\boldsymbol{\theta}^{l}, \mathbf{w}^{l}\right]^{\mathrm{T}} - [\alpha_\theta^l, \alpha_w^l]\nabla J(\boldsymbol{\theta}, \mathbf{w}) \tag{32}$$

where $l = 0, 1, \ldots$ is the iteration index and α_θ^l and α_w^l are diminishing step-size sequences satisfying $\sum_{l=0}^{\infty} \alpha_\theta^l = \infty, \lim_{l\to\infty} \alpha_\theta^l = 0$ and $\sum_{l=0}^{\infty} \alpha_w^l = \infty, \lim_{l\to\infty} \alpha_w^l = 0$. A decentralized version of (32) is

$$\left[\boldsymbol{\theta}_j^{l+1}, \mathbf{w}_j^{l+1}\right]^{\mathrm{T}} = \left[\boldsymbol{\theta}_j^{l}, \mathbf{w}_j^{l}\right]^{\mathrm{T}} - [\alpha_\theta^l, \alpha_w^l]\nabla_j J(\bar{\boldsymbol{\theta}}, \bar{\mathbf{w}}) \tag{33}$$

where $\bar{\boldsymbol{\theta}}$ and $\bar{\mathbf{w}}$ are agent j's estimates based on the limited information provided in Lemma 2.

Theorem 1. *Any centralized solution of (10) through (32) can be recovered by (33) in which each agent j optimizes its trajectory given the following conditions:*

1) *Initial parameters $[\boldsymbol{\theta}_j^0, \mathbf{w}_j^0]$;*
2) *The local information set $I_j(t)$;*
3) *The subset of the global information set $\{\rho_i^0, i \notin \mathcal{T}_j(t)\}$.*

It is important to point out that Theorem 1 relies on the gradient $\nabla_j R_i(t)$ for $i \notin \mathcal{T}_j(t)$ and not on $R_i(t)$. In fact, there is no attempt by agent j to reconstruct or estimate the states of targets $i \notin \mathcal{T}_j(t)$; the only information from such targets is provided through the occasional ρ_i^0 events.

While the event-driven nature of IPA has several computational advantages (see **Remark 3**), the optimization process depends on these events being observed so as to "excite" algorithms such as (32) and (33). To resolve this *event excitation* issue, potential field methods were proposed in [17] and [9].

VI. Simulation Examples

We present two simulation examples to demonstrate the performance of the proposed decentralized scheme.

In the first example, three homogeneous agents are allocated to monitor seven targets in the 1D mission space for $T = 300$ seconds. The targets are located at $x_i = 5i$ for $i = 1, \ldots, 7$. The uncertainty dynamics in (4) are defined by the parameters $A_i = 1$, $B_i = 5$, with initial values $R_i(0) = 1$ for all i. Each agent has a sensing range of $r = 3$ and is initialized with $s_j(0) = 0.5(j-1)$, $u_j(0) = 1$, $\boldsymbol{\theta}_1^0 = [5, 10, 15, 10, 5, \ldots]$, $\boldsymbol{\theta}_2^0 = [15, 20, 25, 20, 15, \ldots]$, $\boldsymbol{\theta}_3^0 = [25, 30, 35, 30, 25, \ldots]$, and $\mathbf{w}_j^0 = [0.5, 0.5, 0.5, \ldots]$ for $j = 1, \ldots, 3$. Results of the method in Theorem 1 are shown in Fig. 2. The top plot depicts the optimal trajectories of each agent determined after 200 iterations of (33), while the bottom plot shows the overall cost $J(\boldsymbol{\theta}, \mathbf{w})$ as a function of iteration number. The exact same results (not shown here) as in Fig. 2 were also obtained through the centralized scheme (32) where all information is available to every agent.

As pointed out earlier, the method of Theorem 1 does not involve any knowledge by agent j of the states of targets $i \notin \mathcal{T}_j(t)$. This is illustrated in Fig. 3 which shows (in blue) the fraction of time that agent 1 has any information on the state of target 3 because it happens that $3 \in \mathcal{T}_1(t)$. The rest of the time (shown in red) agent 1 is unable to accurately estimate the state of this target, but such information is unnecessary. The agent only needs a small subset of is non-local information, as illustrated by the green dots in Fig. 3.

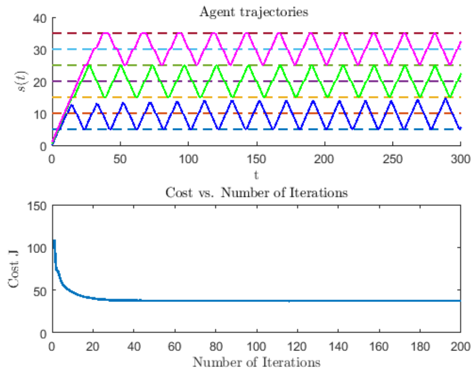

Fig. 2: "Almost decentralized" optimization using Theorem 1. Top plot: optimal agent trajectories. Bottom plot: cost as a function of number of iterations with final cost $J^\star = 37.38$.

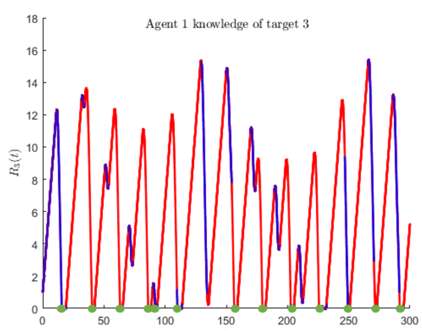

Fig. 3: Red curve: $R_3(t)$, the state of target 3. Blue segments: $R_3(t)$ known to agent 1 when target 3 in its neighborhood. Green dots: instants when agent 1 receives non-local events ρ_3^0.

The second example uses the same environment as the first one and agents start with the same initial trajectories. However, we eliminate the non-local information (condition 3 in Theorem 1) and each agent calculates its own IPA-based gradient using only local information in set $I_j(t)$. Figure 4 shows the results after 200 of iterations of (33). Even though the gradient estimate for agent j is no longer accurate without the ρ_i^0 event information when $i \notin \mathcal{T}_j(t)$, the cost still decreases and converges as shown in Fig. 4, illustrating the robustness of the IPA-based gradient descent method.

VII. Conclusions and Future Work

We have shown that in 1D an optimal centralized solution can be recovered by an event-driven "almost decentralized" algorithm which significantly reduces communication costs while yielding the same performance. In particular, each agent uses only local information except for one event

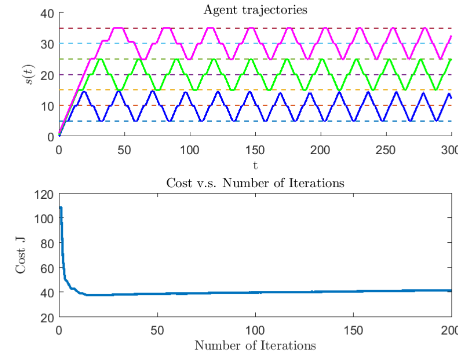

Fig. 4: Fully decentralized optimization without any non-local information. Top plot: optimal agent trajectories. Bottom plot: cost as a function of number of iterations with $J^\star = 41.66$.

requiring communication with a non-neighbor agent when it occurs. The extension of this approach to the 2D case is the subject of ongoing research. The derivation in this paper holds if agents move in straight-lines under graph-limited mobility constraints as shown in [18].

References

[1] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Trans. on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[2] N. Michael, E. Stump, and K. Mohta, "Persistent surveillance with a team of mavs," in *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots Systems*, 2011, pp. 2708–2714.

[3] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang, "Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay," *Journal of Field Robotics*, vol. 27, no. 6, pp. 718–740, 2010.

[4] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Trans. on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2013.

[5] S. L. Smith, M. Schwager, and D. Rus, "Persistent Robotic Tasks: Monitoring and Sweeping in Changing Environments," *IEEE Trans. on Robotics*, vol. 28, no. 2, pp. 410–426, Apr. 2012.

[6] E. Stump and N. Michael, "Multi-robot persistent surveillance planning as a vehicle routing problem," in *Proc. IEEE Conf. on Automation Science and Engineering*, 2011, pp. 569–575.

[7] X. Yu, S. B. Andersson, N. Zhou, and C. G. Cassandras, "Optimal dwell times for persistent monitoring of a finite set of targets," in *Proc. American Control Conference (ACC)*, 2017, pp. 5544–5549.

[8] B. Horling and V. Lesser, "A survey of multi-agent organizational paradigms," *The Knowledge Engineering Review*, vol. 19, no. 04, pp. 281–316, 2004.

[9] N. Zhou, X. Yu, S. B. Andersson, and C. G. Cassandras, "Optimal event-driven multi-agent persistent monitoring of a finite set of targets," in *Proc. IEEE Conf. on Decision and Control*, 2016, pp. 1814–1819.

[10] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao, "Perturbation analysis and optimization of stochastic hybrid systems," *European Journal of Control*, vol. 16, no. 6, pp. 642–661, 2010.

[11] Y. Wardi, R. Adams, and B. Melamed, "A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case," *IEEE Trans. on Autom. Contr.*, vol. 55, no. 1, pp. 89–103, 2010.

[12] K. Leahy, D. Zhou, C.-I. Vasile, K. Oikonomopoulos, M. Schwager, and C. Belta, "Provably correct persistent surveillance for unmanned aerial vehicles subject to charging constraints," in *Experimental Robotics*. Springer, 2016, pp. 605–619.

[13] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 324–333, 2008.

[14] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[15] A. E. Bryson and Y.-C. Ho, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[16] N. Zhou, C. G. Cassandras, X. Yu, and S. B. Andersson, "Decentralized event-driven algorithms for multi-agent persistent monitoring," *arXiv preprint arXiv:1708.06432*, 2017.

[17] Y. Khazaeni and C. G. Cassandras, "Event excitation for event-driven control and optimization of multi-agent systems," in *Proc. IEEE Intl. Workshop on Discrete Event Systems*, 2016, pp. 197–202.

[18] N. Zhou, C. G. Cassandras, X. Yu, and S. B. Andersson, "Optimal event-driven multi-agent persistent monitoring with graph-limited mobility," in *Proc. of 20th IFAC World Congress*, 2017, pp. 2217–2222.