Optimal Persistent Monitoring Using Second-Order Agents With Physical Constraints

Yan-Wu Wang[®], Senior Member, IEEE, Yao-Wen Wei[®], Xiao-Kang Liu[®], Nan Zhou[®], Student Member, IEEE, and Christos G. Cassandras[®], Fellow, IEEE

Abstract—This paper addresses a one-dimensional optimal persistent monitoring problem using second-order agents. The goal is to control the movements of agents to minimize a performance metric associated with the environment (targets) over a finite time horizon. In contrast to earlier results limited to first-order dynamics for agents, we control their accelerations rather than velocities, thus leading to a better approximation of agent behavior in practice and to smoother trajectories. Bounds on both velocities and accelerations are also taken into consideration. Despite these added complications to agent dynamics, we derive a necessary condition for optimality and show that the optimal agent trajectories can be fully characterized by two parameter vectors. A gradient-based algorithm is proposed to optimize these parameters and yield a minimal performance metric. In addition, a collision avoidance algorithm is proposed to solve potential collision and boundary-crossing problems, thus extending the gradient-based algorithm solutions. Finally, simulation examples are included to demonstrate the effectiveness of our results. Index Terms—Optimal control, persistent monitoring,

second-order agent.

I. INTRODUCTION

R ECENT developments in cooperative multiagent systems have enabled applications in which a group of autonomous agents is used to perform tasks collectively in order to optimize a global objective. In particular, persistent monitoring arises in applications, such as city patrolling [1], [2], ecological surveillance [3], [4], traffic monitoring [5], [6], smart-grid security [7],

Manuscript received February 8, 2018; revised July 28, 2018; accepted October 18, 2018. Date of publication November 7, 2018; date of current version July 26, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61773172. Grant 61572210, and Grant 51537003, in part by the Natural Science Foundation of Hubei Province of China under Grant 2017CFA035, and in part by the Academic Frontier Youth Team of HUST. Recommended by Associate Editor D. Hristu-Varsakelis. (Corresponding author: Yan-Wu Wang.)

Y.-W. Wang, Y.-W. Wei, and X.-K. Liu are with the School of Automation and the Key Laboratory of Image Processing and Intelligent Control, Ministry of Education, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: wangyw@hust.edu.cn; weiyw@ hust.edu.cn; xkliu@hust.edu.cn).

N. Zhou is with the Division of Systems Engineering, Boston University, Boston, MA 02446 USA (e-mail: nanzhou@bu.edu).

C. G. Cassandras is with the Division of Systems Engineering and Department of Electrical and Computer Engineering, Boston University, Boston, MA 02446 USA (e-mail: 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.2018.2879946

[8], and ocean sampling [9], [10]. The dynamically changing environments in these applications require the agents to perpetually move in a large mission space. The challenge in this type of problems is to design the agent trajectories under physical motion constraints in order to optimize an overall performance metric.

Relevant studies on the persistent monitoring problem can be categorized into two classes, namely, one with predefined trajectories [11]–[13] and one without predefined trajectories [14]–[18]. For the monitoring problem with predefined trajectories, the main challenge is to design appropriate motion laws for agents to patrol on the given trajectories. Persistent monitoring of a changing environment is addressed in [19], where the object is to control the agents' velocities to prevent the unbounded growth of an accumulation function defined on a finite number of locations. The increase or decrease of the accumulation function depends on whether the location is covered under an agent's footprint. For the monitoring problem without predefined trajectories, the main challenge is to find an optimal target visiting schedule and conditions for agents to switch if the problem is discrete [20], [21] or to search for optimal trajectories if the problem is continuous [18], [22], [23]. The latter paradigm is more flexible without predefined agent trajectories and finds wider applications, such as maneuvering targets [24], [25], detecting random events [20], [26], and monitoring dynamically changing environments [22], [27] or fields with motion constraints [28], [29]. An optimal control framework for persistent monitoring problems is proposed in [23], where an uncertainty metric is minimized subject to first-order agent dynamics. Compared with the accumulation function in [19], the uncertainty metric in [23] is more general because the detection probability of a point may vary depending on the distance between the agent and the point.

The aforementioned literature deals with the monitoring task using first-order agents by controlling their velocities. However, in practice, agents are subject to maximum power constraint that leads to bounds on both accelerations and velocities. In this paper, we consider *second-order* agents with such a power constraint. We control the agent accelerations rather than the velocities leading to a better approximation of agent behavior in practice and to smoother trajectories. In addition, due to the fact that (unlike the setting in [23]) an agent must decelerate before stopping, there is the potential of collisions or of crossing the boundaries of the mission space, a problem that we also consider in this paper. In particular, whereas in [23] it was shown that

0018-9286 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

it is never optimal for agents to have trajectories that cross each other, this is no longer generally true when second-order dynamics are considered. This is a consequence of the implicit cost incurred by agents whenever they have to decelerate, which results in wasting time needed to sense one or more targets.

Based on the above-mentioned discussion, we formulate the persistent monitoring problem as a minimization problem of a performance metric represented as an integral function of average uncertainty over a fixed time horizon. Specifically, this paper uses the accelerations of agents as control inputs and takes into account the physical constraints that bound both the acceleration and velocity. Through a Hamiltonian analysis, we obtain a necessary condition for optimality. The resulting optimal controller contains the following four modes, e.g.,

- 1) maximal acceleration mode;
- 2) maximal velocity mode;
- 3) maximal deceleration mode;
- 4) dwell mode.

Under such an optimal control structure, the agent trajectories can be fully characterized and parameterized by the starting points of each mode and the associated dwell times. The original optimal control problem can then be transformed to a simpler parametric one, and thus, the search for the optimal control is reduced from a functional space to some finite number of parameters along the agent trajectories. Compared to [23], the presence of four modes in the controller, instead of only two, causes nontrivial complications in the online derivation of the gradient information. Finally, a gradient-based algorithm is proposed to minimize the performance metric and to determine the optimal trajectories. In addition, we propose an improved optimization algorithm that prevents collisions or the possibility of an agent crossing the boundaries of the mission space resulted by the agent's inertia in the second-order dynamics.

This paper is organized as follows. Section II formulates the optimal persistent monitoring problem using second-order multiagent systems. Section III-A analyzes the optimal control structure, Section III-B shows how to determine the optimal trajectories through a gradient-based algorithm, and Section III-C proposes an improved optimization algorithm to prevent agent collisions and boundary crossings. Simulation results are presented in Section IV to demonstrate the effectiveness of the proposed algorithm and to show the results of the persistent monitoring task. Section V concludes this paper.

II. PROBLEM FORMULATION

Consider a one-dimensional (1-D) mission space [0, L]. N cooperating agents are assigned to move on the mission space to accomplish a persistent monitoring task over the time horizon [0, T].

In this paper, we control the movement of each agent through its acceleration as opposed to the velocity in the first-order case. The dynamics of agent i are described by

$$\begin{cases} \dot{s}_i(t) = v_i(t), \\ \dot{v}_i(t) = u_i(t), \end{cases} i = 1, 2, \dots, N$$
(1)

where $t \in [0, T]$, $s_i(t) \in [0, L]$ is the agent position, $v_i(t)$ is the velocity, and $u_i(t)$ is the acceleration control input. We assume that the velocity of each agent *i* is bounded by

$$|v_i(t)| \le v_i^{\max}, \ i = 1, 2, \dots, N$$
 (2)

and the acceleration input is bounded by

$$\mathcal{U}: \begin{cases} |u_i(t)| \le C_i^a, \text{ if } u_i(t)v_i(t) \ge 0, \\ |u_i(t)| \le C_i^d, \text{ if } u_i(t)v_i(t) < 0, \end{cases} (3)$$

where v_i^{max} , C_i^a , and C_i^d are the maximal velocity, the maximal acceleration, and the maximal deceleration, respectively. Note that in the deceleration mode, the control direction is opposite to the motion direction (i.e., $u_i(t)v_i(t) < 0$). The agent dynamics under boundary constraints (2) and (3) can be rewritten as

$$\begin{cases} \dot{s}_{i}(t) = v_{i}(t) \\ 0, & \text{if } |v_{i}(t)| = v_{i}^{\max} \\ u_{i}(t) = \begin{cases} 0, & \text{if } |v_{i}(t)| = v_{i}^{\max} \\ 0, & \text{and } v_{i}(t^{+})u_{i}(t^{+}) \ge 0 \\ u_{i}(t), & \text{otherwise} \end{cases}$$
(4)

Considering that sensors have various physical characteristics, we model the sensing capability of the *i*th agent for detecting a target located at $x \in [0, L]$ by a probability function $p(s_i(t), x)$ [30] that

$$p(s_i(t), x) = \begin{cases} 1 - \frac{(x - s_i(t))^2}{r_i^2}, & \text{if } |x - s_i(t)| < r_i \\ 0, & \text{otherwise} \end{cases}$$

where r_i is the effective sensing radius. The probability that target x is sensed by all agents simultaneously can be formulated as

$$P(S(t), x) = 1 - \prod_{i=1}^{N} [1 - p(s_i(t), x)]$$
(5)

where $S(t) = [s_1(t), s_2(t), \dots, s_N(t)]$ is the position vector of all agents.

Referring to previous research, this paper adopts the definition of uncertainty from [19] and [23]. A time-varying function R(x,t) is defined to describe the uncertainty of target x at time t with the following properties. If target x cannot be sensed by any agent, R(x,t) increases with a prespecified rate I(x). Meanwhile, if x is sensed with probability P(S(t), x), then R(x,t)increases with rate I(x) - D(x)P(S(t), x), where D(x) is the maximal monitoring effect on R(x,t). However, if R(x,t) = 0and I(x) - D(x)P(S(t), x) < 0, the uncertainty remains 0. In this way, the dynamics of the uncertainty function are as follows:

$$\dot{R}(x,t) = \begin{cases} 0 & \text{if } I(x) - D(x)P(S(t),x) < 0\\ & \text{and } R(x,t) = 0\\ I(x) - D(x)P(S(t),x) & \text{otherwise.} \end{cases}$$
(6)

As in [23], we note that a simple stability condition for such a system over a mission time T is that the agents have adequate capacity to handle the total uncertainty input, i.e., $\int_0^T \int_0^L I(x,t) dx dt < \int_0^T \int_0^L D(x) P(S(t),x) dx dt$, where we have used I(x,t) instead of I(x) to allow for possible time

dependence in general. The purpose of this persistent monitoring task is to minimize the performance metric that is defined as the average uncertainty of all targets over the time horizon [0, T]. In light of [23], suppose there are M targets (points of interest) in the mission space and x_j is the position of target j, j = 1, 2, ..., M, then the optimal control problem is formulated as

$$\min_{U(t)\in\mathcal{U}} J = \frac{1}{M} \int_0^T \sum_{j=1}^M R(x_j, t) dt$$
(7)

where $U(t) = [u_1(t), u_2(t), \dots, u_N(t)]$ is the acceleration vector of all agents.

Remark 1: The performance metric (7) is the average over all target uncertainties instead of the average over the time horizon in [23]. In addition, U(t) in this paper consists of the accelerations of agents, which is more practical than controlling velocities directly and makes it possible to keep the trajectory smooth.

III. MAIN RESULTS

In this section, we focus on solving the optimal control problem (7) by finding the optimal control policy. The main challenge lies in the following four aspects.

- 1) The velocity and acceleration are bounded, thus, the dynamics of agents are hybrid.
- 2) R(x,t) has nonsmooth switching dynamics as seen in (6).
- 3) The performance metric (7) is not a function of U(t) explicitly.
- 4) We must ensure that no agent crosses the mission space boundaries or collides with its neighbors.

The work in this section contains three parts. Section III-A shows the characteristics of the optimal control by using a Hamiltonian analysis. Under such optimal control policies, the agent trajectories can be parameterized by a sequence of locations of control switches and the associated dwell times when an agent switches its control from $\pm C_i^d$ to 0. Thus, the performance metric in (7) can be transformed into a parametric form as we will show later in Section III-B. A gradient-based algorithm is designed to calculate the minimal performance metric and to determine the optimal trajectory of each agent. In Section III-C, a collision avoidance algorithm is proposed to determine a suboptimal trajectory that prevents an agent from crossing the mission space boundaries or from running across its neighbors.

A. Optimal Control Policy

In this section, the optimal policy will be determined. Let the velocity vector be denoted by $V(t) = [v_1(t), v_2(t), \ldots, v_N(t)]$, and the uncertainty vector by $\Re(t) = [R(x_1, t), R(x_2, t), \ldots, R(x_M, t)]$. The performance metric in (7) will be minimized subject to the agent dynamics (1), uncertainty dynamics (6), and control constraints (3). Introduce the associated Lagrange multipliers $\lambda_u(t) = [\lambda_{u_1}(t), \lambda_{u_2}(t), \ldots, \lambda_{u_N}(t)], \lambda_v(t) = [\lambda_{v_1}(t), \lambda_{v_2}(t), \ldots, \lambda_{v_N}(t)]$, and $\lambda_R(t) = [\lambda_{R_1}(t), \lambda_{R_2}(t), \ldots, \lambda_{R_M}(t)]$ with the boundary conditions $\lambda_u(T) = 0, \lambda_v(T) = 0$ and

 $\lambda_R(T) = 0$ separately. The performance metric is also subject to the velocity constraints (2) and we introduce $\mu(t) = [\mu_1(t), \mu_2(t), \dots, \mu_N(t)]$

$$\begin{cases} \mu_i(t) = 0, & \text{if } v_i < v^{\max} \\ \mu_i(t) > 0, & \text{if } v_i = v^{\max}. \end{cases}$$

Then, the minimization problem (7) can be rewritten as

$$\min_{U(t)\in\mathcal{U}} J = \frac{1}{M} \int_0^T \left[\mathbf{1}_M \mathfrak{R}^T(t) + \lambda_v(t) \left(V^T(t) - \dot{S}^T(t) \right) + \lambda_u(t) \left(U^T(t) - \dot{V}^T(t) \right) + \lambda_R(t) \left(\dot{\mathfrak{R}}^T(t) - \dot{\mathfrak{R}}^T(t) \right) + \mu(t) \left(V(t) - V^{\max} \right)^T \right] dt$$
(8)

where $\mathbf{1}_M = [1, 1, \dots, 1]_M$ and $V^{\max} = [v_1^{\max}, v_2^{\max}, \dots, v_N^{\max}]$. The Hamiltonian is defined as

$$H(\mathfrak{R}, S, V, U, \lambda, t) = \mathbf{1}_{M} \mathfrak{R}^{T}(t) + \lambda_{v}(t)V^{T}(t) + \lambda_{u}(t)U^{T}(t) + \lambda_{R}(t)\dot{\mathfrak{R}}^{T}(t) + \mu(t)(V(t) - V^{\max})^{T}.$$
(9)

For simplicity, we write $H \equiv H(\mathfrak{R}, S, V, U, \lambda, t)$ and (8) can be rewritten as

$$\min_{U(t)\in\mathcal{U}} J = \frac{1}{M} \int_0^T \left[H - \lambda_v(t) \dot{S}^T(t) - \lambda_u(t) \dot{V}^T(t) - \lambda_R(t) \dot{\mathcal{R}}^T(t) \right] dt$$
(10)

Furthermore, integrate the last three terms on the right side of (10) by parts, which yields $\int_0^L \lambda_v(t) \dot{S}^T(t) dt = \lambda_v(t) S^T(t) |_0^T - \int_0^L \dot{\lambda}_v(t) S^T(t) dt$ and the same result applies to $\int_0^L \lambda_u(t) \dot{V}^T(t)$ and $\int_0^L \lambda_R(t) \dot{\Re}^T(t)$. Hence, combining with the boundary conditions of Lagrange multipliers, (10) can be rewritten as

$$\min_{U(t)\in\mathcal{U}} J = \frac{1}{M} \int_0^T \left[H + \dot{\lambda}_v(t) S^T(t) + \dot{\lambda}_u(t) V^T(t) + \dot{\lambda}_R(t) \mathfrak{R}^T(t) \right] dt.$$
(11)

Based on the optimal necessary conditions of the Hamiltonian analysis, the costates require to satisfy

$$\dot{\lambda}_R(t) = -\frac{\partial H}{\partial \Re(t)} = -\mathbf{1}_M \tag{12}$$

$$\dot{\lambda}_{v}(t) = -\frac{\partial H}{\partial S(t)} = -\lambda_{R}(t)\frac{\partial \dot{\Re}(t)}{\partial S(t)}$$
(13)

$$\dot{\lambda}_u(t) = -\frac{\partial H}{\partial V(t)} = -\lambda_v(t).$$
 (14)

Based on the above-mentioned analysis, the following proposition is established to reveal the relationships between the optimal control input and the Lagrange multipliers.

Proposition 1: For any given trajectory, the optimal control policy satisfies

1

$$u_i^*(t) \in \left\{ \pm C_i^a, \pm C_i^d, 0 \right\}, \text{ for all } t \in [0, T].$$
 (15)

Proof: Let $\Gamma_0(t) = \{i | \lambda_{u_i}^*(t) = 0, i = 1, 2, ..., N\}$, $\Gamma_1(t) = \{i | \lambda_{u_i}^*(t) > 0, i = 1, 2, ..., N\}$, and $\Gamma_2(t) = \{i | \lambda_{u_i}^*(t) < 0, i = 1, 2, ..., N\}$. The Pontryagin Minimum Principle [31] holds for the optimal control problem (11) and asserts that

$$H(\mathfrak{R},S,V,U^*,\lambda^*,t)=\min_{U(t)}H(\mathfrak{R},S,V,U,\lambda^*,t)$$

where U^* , λ^* denote the vector of optimal controller and Lagrange multiplier, respectively. Clearly, it is necessary for the optimal control of agent *i* to satisfy

$$u_i^*(t) = -\operatorname{sgn}(\lambda_{u_i}^*(t)) \max_{u_i(t) \in \mathcal{U}} |u_i(t)|.$$

For $i \in \Gamma_1(t)$, agent *i* moves toward the negative direction with maximal acceleration or toward the positive direction with maximal deceleration, i.e.,

$$u_i^*(t) = -C_i^{\ a}, \quad \text{if } v_i(t) < 0$$

 $u_i^*(t) = -C_i^{\ d}, \quad \text{if } v_i(t) > 0.$

Similarly, for $i \in \Gamma_2(t)$ the optimal control of agent *i* satisfies

$$u_i^*(t) = C_i^{\ a}, \quad \text{if } v_i(t) > 0$$

 $u_i^*(t) = C_i^{\ d}, \quad \text{if } v_i(t) < 0.$

It is possible (see [32, Ch. 3]) that there are some agents $i \in \Gamma_0(t)$ at some time intervals. For these $i \in \Gamma_0(t)$, $\lambda_{u_i}^*(t)u_i(t) = 0$ that is omitted in the above-mentioned analysis. From (9), $H(\mathfrak{R}, S, V, U, \lambda, t)$ is not an explicit function of t and $H(\mathfrak{R}, S, V, U^*, \lambda^*, t) \equiv C$ is constant [33]. Therefore, $\frac{dH}{dt} = 0$ that gives

$$\frac{dH}{dt} = \mathbf{1}_{M} \dot{\mathfrak{R}}^{T}(t) + \left(\dot{\lambda}_{v}(t)V^{T}(t) + \lambda_{v}(t)\dot{V}^{T}(t)\right) \\
+ \left(\dot{\lambda}_{u}(t)U^{T}(t) + \lambda_{u}(t)\dot{U}^{T}(t)\right) \\
+ \left(\dot{\lambda}_{R}(t)\dot{\mathfrak{R}}^{T}(t) + \lambda_{R}(t)\ddot{\mathfrak{R}}^{T}(t)\right) \\
+ \dot{\mu}(t)\left(V(t) - V^{\max}\right)^{T} + \mu(t)U^{T}(t).$$
(16)

Meanwhile, based on the necessary conditions (12) and (14)

$$\mathbf{1}_{M}\dot{\mathbf{R}}^{T}(t) + \dot{\lambda}_{R}(t)\dot{\mathbf{R}}^{T}(t) = 0$$
$$\lambda_{v}(t)\dot{V}^{T}(t) + \dot{\lambda}_{u}(t)U^{T}(t) = 0$$

Therefore, we have

$$\frac{dH}{dt} = \dot{\lambda}_v(t)V^T(t) + \lambda_R(t)\ddot{\mathbf{R}}^T(t) + \lambda_u(t)\dot{U}^T(t)
+ \dot{\mu}(t)(V(t) - V^{\max})^T + \mu(t)U^T(t).$$
(17)

For those agents $i \in \Gamma_0(t)$, we have $\lambda_{u_i}(t) = 0$, so that $\sum_{i \in \Gamma_0(t)} \lambda_{u_i}(t) \dot{u}_i(t) = 0$, and for those agents $i \in \Gamma_1(t) \cup \Gamma_2(t)$, we have $u_i(t) = \pm C_i^a$ or $\pm C_i^d$, $\dot{u}_i(t) = 0$ and $\sum_{i \in \Gamma_1(t) \cup \Gamma_2(t)} \lambda_{u_i}(t) \dot{u}_i(t) = 0$. The above-mentioned analysis suggests that $\lambda_u(t) \dot{U}^T(t) = 0$. According to (14), if $i \in \Gamma_0(t)$, $\lambda_{u_i}(t) = 0$, $\dot{\lambda}_{u_i}(t) = 0 = -\lambda_{v_i}(t)$. Thus, for any time instant, (17) is reduced to

$$\frac{dH}{dt} = \sum_{i \in \Gamma_1(t) \cup \Gamma_2(t)} \dot{\lambda}_{v_i}(t) v_i(t) + \lambda_R(t) \ddot{\mathfrak{R}}^T(t)
+ \dot{\mu}(t) (V(t) - V^{\max})^T + \mu(t) U^T(t).$$
(18)

From (6), since $\dot{R}(s(t), x)$ is not an explicit function of t, we have

$$\ddot{\mathfrak{R}}(t) = \frac{d(\mathfrak{R}(t))}{d(S(t))} \dot{S}^T(t).$$

From (13), it follows that for $i \in \Gamma_1(t) \cup \Gamma_2(t)$

$$\dot{\lambda}_{v_i}(t)v_i(t) + \lambda_{R_i}(t)rac{d(\dot{\mathfrak{R}}(t))}{d(s_i(t))}\dot{s}_i(t) = 0.$$

which provides the fact that

$$\frac{dH}{dt} = \sum_{i \in \Gamma_0(t)} \lambda_{R_i}(t) \frac{d(\hat{\mathfrak{R}}(t))}{d(s_i(t))} \dot{s}_i(t)
+ \dot{\mu}(t) (V(t) - V^{\max})^T + \mu(t) U^T(t). \quad (19)$$

If the state evolves in an interior arc of velocity constraint (2), i.e., $v_i(t) < v_i^{\max}$ and $\mu(t) = 0$, then $\dot{\mu}(t) = 0$. Otherwise, the state evolves in the boundary arc of (2), i.e., $|v_i(t)| \equiv v_i^{\max}$, then $u_i(t) = 0$. Then, it can be obtained that $\dot{\mu}(t)(V(t) - V^{\max})^T + \mu(t)U^T(t) = 0$. From (19)

$$\frac{dH}{dt} = \sum_{i \in \Gamma_0(t)} \lambda_{R_i}(t) \frac{d(\mathfrak{R}(t))}{d(s_i(t))} \dot{s}_i(t).$$
(20)

Thus, to ensure (20) holds at all $t \in [0, T]$, $\dot{s}_i(t) = 0$ for those $i \in \Gamma_0(t)$ that means $v_i(t) = 0, u_i(t) = 0$.

Also note that from (4), $u_i(t) = 0$ if $|v_i(t)| = v_i^{\max}$ for a finite time period. Based on the above-mentioned analysis, (15) holds, which completes the proof.

From Proposition 1, $u_i^*(t) \in \{\pm C_i^a, \pm C_i^d, 0\}$, the optimal control requires the agents to move with maximal acceleration or fixed maximal velocity or maximal deceleration or remain at rest. To be more specific, the motion of an agent, moving from one point to another, may include the following four modes:

- 1) *Maximal acceleration mode:* the agent moves with maximal acceleration from one point to another, where the control direction is agree with the motion direction.
- Maximal velocity mode: the agent moves at a fixed maximal velocity for a period of time larger than zero.
- Maximal deceleration mode: the agent moves with maximal deceleration, where the control direction is opposite to the motion direction.
- Dwell mode: the agent dwells at some points for some time (possibly zero).

The resulting optimal agent trajectories can be fully characterized by the starting points and ending points of each mode and the dwell times associated with each dwell mode.

Remark 2: Proposition 1 reveals the agent's optimal acceleration for any given trajectory. From the optimal analysis of Proposition 1, when the maximal velocity constraint is not active, the associated Lagrange multiplier $\lambda_{u_i}(t) > 0, < 0, = 0$,

and the agent's optimal acceleration should be the maximal acceleration or deceleration, or zero acceleration (with zero velocity), which leads to the maximal acceleration or deceleration mode, or dwell mode, respectively. During the period that the maximal velocity constraint is active, i.e., $v_i^*(t) = v_i^{\max}$, we have $u_i^*(t) = 0$ for a given time interval, and this leads to the maximal velocity mode.

For agent *i*, define a sequence of dwelling points $\theta_i = [\theta_{i1}, \theta_{i2}, \ldots, \theta_{iK_i}]$ and the associated dwelling times $\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{iK_i}]$. Within each two successive dwelling points, we further define two switching points of the agent dynamics, i.e., the switching points $\theta_{ik_i}^a$ (ending points of maximal acceleration mode), and $\theta_{ik_i}^d$ (starting points of maximal deceleration mode), $\theta_{ik_i} < \theta_{ik_i}^a \leq \theta_{ik_i}^d < \theta_{i(k_i+1)}$, $k_i = 1, 2, \ldots, K_{i-1} - 1$. For $\theta_{ik_i}^a = \theta_{ik_i}^d$ being a same point, at this point $u_i(t)$ changes from $u_i(t) = \pm C_i^a$ to $u_i(t) = \mp C_i^d$ directly. For $\theta_{ik_i}^a \neq \theta_{ik_i}^d$, agent *i* moves at fixed maximal velocity between $\theta_{ik_i}^a$ and $\theta_{ik_i}^d$, $u_i(t)$ changes from $u_i(t) = \pm C_i^a$ to $u_i(t) = \pm C_i^a$ to $u_i(t) = 0$ at $\theta_{ik_i}^a$ and from $u_i(t) = 0$ to $u_i(t) \mp C_i^d$ at $\theta_{ik_i}^d$.

Remark 3: Note that once the dwelling points θ_i are determined, the acceleration, deceleration, and the possibly maximum velocity modes between these points can be subsequently calculated. Therefore, if the dwelling points and the dwelling times are determined, the trajectory of agent *i* will be fully determined and represented as $[\theta_i, \omega_i]$, and the trajectories of all agents can be represented as $[\theta, \omega]$ where $\theta = [\theta_1, \theta_2, \dots, \theta_N], \omega = [\omega_1, \omega_2, \dots, \omega_N].$

To proceed, the existence of the maximum velocity modes can be determined using the following proposition.

Proposition 2: The maximum velocity mode exists between θ_{ih} and $\theta_{i(h+1)}$ when agent *i* moves from θ_{ih} to $\theta_{i(h+1)}$ if and only if

$$|\theta_{i(h+1)} - \theta_{ih}| > \frac{v_i^{\max^2}}{2C_i^a} + \frac{v_i^{\max^2}}{2C_i^d}.$$
 (21)

Proof: Note that θ_{ih} and $\theta_{i(h+1)}$ are both dwelling points at which $v_i(t) = 0$. The time it takes for the velocity of agent *i* to increase from 0 to v_i^{\max} with the maximal acceleration C_i^a is $\frac{v_i^{\max}}{C_i^a}$ and the time duration it takes for the velocity of agent *i* to decrease from v_i^{\max} to 0 with the maximal deceleration C_i^d is $\frac{v_i^{\max}}{C_i^d}$. The average velocity of both the uniform acceleration and deceleration process is $\frac{v_i^{\max}}{2}$. Therefore, the shortest distance between θ_{ih} and $\theta_{i(h+1)}$ for agent *i* to reach the maximal velocity is

$$\frac{v_i^{\max}}{C_i^a}\frac{v_i^{\max}}{2} + \frac{v_i^{\max}}{C_i^d}\frac{v_i^{\max}}{2}$$

Hence, the maximal velocity mode exists between θ_{ih} and $\theta_{i(h+1)}$ if and only if (21) holds.

Section III-B is devoted to determining the optimal dwelling points and the corresponding optimal dwelling times, so as to determine the optimal trajectory.

B. Optimal Trajectory

Since the agent trajectories are represented as $[\theta, \omega]$, combining with Proposition 1, the controller U(t) is also determined by $[\theta, \omega]$. In other words, the problem is simplified as a parametric minimization problem of finding the optimal dwelling points and the optimal dwelling times. Thus, the objective is to determine $[\theta, \omega]$ satisfying

$$\min_{U(t)\in\mathcal{U}}J=\min_{[\theta,\omega]}J(\theta,\omega).$$

A gradient-based iterative algorithm is designed as follows:

$$[\theta(m), \omega(m)] = [\theta(m-1), \omega(m-1)] + [\tilde{\theta}, \tilde{\omega}] \nabla J(\theta(m-1), \omega(m-1))$$
(22)

where m = 1, 2, ... is the index of iterations, $[\bar{\theta}, \tilde{\omega}]$ is the step-size of the iterative algorithm, and $\nabla J(\theta(m), \omega(m)) = [\frac{\partial J(\theta(m), \omega(m))}{\partial \theta(m)}, \frac{\partial J(\theta(m), \omega(m))}{\partial \omega(m)}]^T$ is the gradient of J with respect to $\theta(m)$ and $\omega(m)$. The trajectory parameters are optimized through (22) iteratively with the terminal condition

$$|J(\theta(m+1), \omega(m+1)) - J(\theta(m), \omega(m))| < \varepsilon$$
(23)

where $\varepsilon > 0$ is a predetermined constant.

Therefore, we are ready to calculate $\nabla J(\theta, \omega)$. As seen in the following, the four modes derived from Proposition 1 result in nontrivial complications relative to [22] in evaluating this gradient information. By (6), the uncertainty dynamics have the switching properties. Define a time sequence to describe the switching instants of $\dot{\Re}(t)$, denoted as $\tau(\theta, \omega) = \{\tau_l(\theta, \omega)\}, l = 0, 1, \dots, \mathcal{L} - 1$, with boundary conditions $\tau_0(\theta, \omega) = 0, \tau_{\mathcal{L}}(\theta, \omega) = T$. Thus, the performance metric (7) can be represented as

$$J(\theta,\omega) = \frac{1}{M} \sum_{l=0}^{\mathcal{L}-1} \int_{\tau_l(\theta,\omega)}^{\tau_{l+1}(\theta,\omega)} \sum_{j=1}^M R(x_j,t) dt.$$

Then, $\nabla J(\theta, \omega)$ can be rewritten as

$$\nabla J(\theta,\omega) = \frac{1}{M} \sum_{l=0}^{\mathcal{L}-1} \sum_{j=1}^{M} \int_{\tau_l(\theta,\omega)}^{\tau_{l+1}(\theta,\omega)} \nabla R(x_j,t) dt \qquad (24)$$

where $\nabla R(x_j, t) = \left[\frac{\partial R(x_j, t)}{\partial \theta}, \frac{\partial R(x_j, t)}{\partial \omega}\right]$. Therefore, in order to compute $\nabla J(\theta, \omega)$, we need to first compute $\nabla R(x_j, t)$.

It is obvious that $R(x_j, t)$ is not an explicit function of (θ, ω) . Therefore, transforming the performance metric in (7) to a function of (θ, ω) is necessary in the following analysis. Note that from the motion law in Proposition 1, the optimal trajectories are described by $[\theta, \omega]$, therefore, the positions of agents are also represented by $[\theta, \omega]$, i.e., the position vector is $S(t) \equiv S(t, (\theta, \omega))$. The following discussion is carried out by three steps, in which a numerical computation method is designed to calculate $\nabla R(x_j, t)$. Let $t_k = k\delta, k = 1, 2...$ be the computation time sequence in [0, T], where the computation step δ is sufficiently small and $t_0 = 0$.

Step 1) For two adjacent instants t_k and t_{k-1} , there are two cases to be discussed.

Case 1.1: No switches between t_k and t_{k-1} . In this case, $t_{k-1}, t_k \in (\tau_l(\theta, \omega), \tau_{l+1}(\theta, \omega))$ and the dynamics of target uncertainties do not switch. According to (6), utilizing the Euler

method
$$\nabla R(x_j, t_k) = \nabla R(x_j, t_{k-1}) + \nabla R(x_j, t_{k-1})\delta$$

$$\nabla R(x_j, t_k) = \nabla R(x_j, t_{k-1})$$

$$-\begin{cases} 0, & \text{if } \dot{R}(x_j, t_{k-1}) = 0\\ D(x_j) \frac{\partial P(S(t_{k-1}), x_j)}{\partial S(t_{k-1})} \nabla S(t_{k-1}) \delta, \text{ otherwise} \end{cases}$$
(25)

where $\nabla S(t_{k-1}) = (\frac{\partial S(t_{k-1})}{\partial \theta}, \frac{\partial S(t_{k-1})}{\partial \omega})$ needs further to be calculated in Steps 2) and 3).

Case 1.2: There exists a switch between t_k and t_{k-1} . In this case, it has $t_{k-1} \leq \tau_l(\theta, \omega) \leq t_k$. Without loss of generality, we suppose j^* switches its dynamics: $\dot{R}(x_{j^*}, t)$ switches from 0 to $I(x_{j^*}) - D(x_{j^*})P(S(t), x_{j^*})$, or from $I(x_{j^*}) - D(x_{j^*})P(S(t), x_{j^*})$ to 0 at $\tau_l(\theta, \omega)$. Therefore, there are two subcases to be discussed.

Before the analysis, we introduce the Infinitesimal Perturbation Analysis method [23], [34] to specify how the arguments θ influences the system state $s(\theta, t)$, ultimately, how they influence the performance metric that can be expressed in terms of such arguments. Let $\{\tau_l(\theta)\}, l = 0, 1, \ldots, \mathcal{L} - 1$, be the occurrence time of all events in the state trajectory. For convenience, we set $\tau_0 = 0$ and $\tau_{\mathcal{L}} = T$. For $t \in [\tau_{l-1}, \tau_l)$, based on the Jacobian matrix notation, we define $s'(t) \equiv \frac{\partial s(\theta, t)}{\partial \theta}$ and the state dynamics $\dot{s}(t) = f_l(s, \theta, t)$. In the following, we use $f_l(t)$ for simplicity. Since t is independent on θ

$$\frac{d}{dt}s'(t) = \frac{\partial f_l(t)}{\partial s}s'(t) + \frac{\partial f_l(t)}{\partial \theta}$$
(26)

with boundary condition

$$s'(\tau_l^+) = s'(\tau_l^-) + [f_{l-1}(\tau_l^-) - f_l(\tau_l^+)] \frac{\partial \tau_l(\theta)}{\partial \theta}.$$
 (27)

Then, let us focus on the gradient $\frac{\partial \tau_l(\theta)}{\partial \theta}$. If there exists a continuously differentiable function $g_l(s(\theta, t), \theta)$, such that $g_l(s(\theta, \tau_l), \theta) = 0$ for any τ_l (in the following, we use g_l for simplicity) holds, then lwe have

$$\begin{aligned} \frac{d}{d\theta}g_l &= \frac{\partial g_l}{\partial s} \left[\frac{\partial s}{\partial \theta} + \frac{\partial s}{\partial \tau_l} \frac{\partial \tau_l}{\partial \theta} \right] + \frac{\partial g_l}{\partial \theta} \\ &= \frac{\partial g_l}{\partial s} \left[s'(\tau_l) + f_l(\tau_l) \frac{\partial \tau_l}{\partial \theta} \right] + \frac{\partial g_l}{\partial \theta} \\ &= 0. \end{aligned}$$

Thus, if $\frac{\partial g_l}{\partial s} f_l(\tau_l^-) \neq 0$, then

$$\frac{\partial \tau_l(\theta)}{\partial \theta} = -\left[\frac{\partial g_l}{\partial s} f_l(\tau_l^-)\right]^{-1} \left(\frac{\partial g_l}{\partial \theta} + \frac{\partial g_l}{\partial s} s'(\tau_l^-)\right).$$
(28)

We are now ready to discuss the two subcases.

Case 1.2.1: $R(x_{j^*}, t_{k-1}) < 0$ switches to $R(x_{j^*}, t_k) = 0$. In this case, $R(x_{j^*}, t)$ satisfies the *endogenous* condition in [23]. From (28) with $g_l = R(x_{j^*}, \tau_l) = 0$, we get

$$\nabla t_k = -\frac{\nabla R(x_{j^*}, t_{k-1})}{I(x_{j^*}) - D(x_{j^*})P(S(t_{k-1}), x_{j^*})}$$

and from (27)

$$\nabla R(x_{j^*}, t_k) = 0. \tag{29}$$



Fig. 1. Schematic for the motion of agent *i*.

Case 1.2.2: $\dot{R}(x_{j^*}, t_{k-1}) = 0$ switches to $\dot{R}(x_{j^*}, t_k) > 0$. In this case, $\dot{R}(x_{j^*}, t)$ is continuous, so that $f_l(\tau_l^+) = f_l(\tau_l^-)$ in (27). Along with the definition of s'(t), we have

$$\nabla R(x_{j^*}, t_k) = \nabla R(x_{j^*}, t_{k-1}).$$
 (30)

Note that it is impossible for the uncertainty dynamics to switch from $\dot{R}(x_{j^*}, t_{k-1}) > 0$ to $\dot{R}(x_{j^*}, t_k) = 0$; this is because if $\dot{R}(x_{j^*}, t_{k-1}) > 0$, $\dot{R}(x_{j^*}, t_k) > \dot{R}(x_{j^*}, t_{k-1}) > 0$, the uncertainty dynamics remain $\dot{R}(x_{j^*}, t_k) > 0$ and switching does not take place. Also, it is impossible for the dynamics to switch from $\dot{R}(x_{j^*}, t_{k-1}) = 0$ to $\dot{R}(x_{j^*}, t_k) < 0$.

Step 2) In this step, case 1.1 will be further discussed.

Based on (25), the remaining work is to calculate $\nabla S(t_k)$. From the agent dynamics (4), S(t) is a continuously differentiable function. According to the optimal control structure in Proposition 1, the dynamics of S(t) fall into four cases. Between each adjacent dwelling points $[\theta_{ih}, \theta_{ih+1}]$, we define t_{ih} as the time instant when agent *i* leaves from θ_{ih} and $t_{ih}^a, t_{ih}^u, t_{ih}^d$ are the time intervals that agent *i* spends in the *h*th acceleration mode, maximal velocity mode, and deceleration mode, respectively. Then, we have

$$t_{ih} = \sum_{q=1}^{h-1} \left[\omega_{iq} + t^a_{iq} + t^u_{iq} + t^d_{iq} \right] + \omega_{ih}.$$
 (31)

To proceed, the mode of agent *i* will first be discussed from the *h*th dwelling point θ_{ih} to the (h + 1)th dwelling point $\theta_{i(h+1)}$, which is illustrated in Fig. 1 (in the figure, the case of $\theta_{ih} < \theta_{i(h+1)}$ is illustrated; the opposite case is similar).

According to the kinematic laws of uniformly variable motion and uniform motion, the motion of agent *i* between θ_{ih} and $\theta_{i(h+1)}$ is obtained as follows. *Case 2.1:* For the *h*th maximal acceleration mode, $s_i(t) \in [\theta_{ih}, \theta_{ih}^a), t \in [t_{ih}, t_{ih} + t_{ih}^a)$, then

$$s_i(t) = \theta_{ih} + \text{sgn}(\theta_{i(h+1)} - \theta_{ih}) \frac{1}{2} C_i^a (t - t_{ih})^2 \qquad (32)$$

$$\nabla s_i(t) = \nabla \theta_{ih} - \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih}) v_i(t) \nabla t_{ih}.$$
 (33)

Case 2.2: For the *h*th maximal velocity mode, the existence depends on the length of $|\theta_{ih+1} - \theta_{ih}|$ as presented in Proposition 2. If the *h*th maximal velocity mode exists, the two points θ_{ih}^a and θ_{ih}^d are not the same and $t_{ih}^u > 0$. Thus, for $s_i(t) \in [\theta_{ih}^a, \theta_{ih}^d), t \in [t_{ih} + t_{ih}^a, t_{ih} + t_{ih}^a + t_{ih}^u)$

$$s_i(t) = \theta_{ih}^a + \text{sgn}(\theta_{i(h+1)} - \theta_{ih})v_i^{\max}(t - t_{ih} - t_{ih}^a)$$
(34)

$$\nabla s_i(t) = \nabla \theta_{ih}^a - \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih}) v_i^{\max} \nabla (t_{ih} + t_{ih}^a).$$
(35)

Case 2.3: For the *h*th maximal deceleration mode, $s_i(t) \in [\theta_{ih}^d, \theta_{i(h+1)}), t \in [t_{ih} + t_{ih}^a + t_{ih}^u, t_{i(h+1)} - \omega_{i(h+1)})$, then

$$s_{i}(t) = \theta_{ih}^{d} + \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih})[v_{i}^{\max}(t - t_{ih} - t_{ih}^{a} - t_{ih}^{u}) - \frac{1}{2}C_{i}^{d}(t - t_{ih} - t_{ih}^{a} - t_{ih}^{u})^{2}]$$
(36)

$$\nabla s_i(t) = \nabla \theta_{ih}^d - \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih})v_i(t)$$
$$\times \nabla (t_{ih} + t_{ih}^a + t_{ih}^u).$$
(37)

Case 2.4: For the *h*th dwell mode, $s_i(t) = \theta_{ih}, t \in [t_{ih} - \omega_{ih}, t_{ih})$, then

$$\nabla s_i(t) = \nabla \theta_{ih}. \tag{38}$$

Step 3) In (33), (35), (37), and (38), ∇t_{ih}^a , ∇t_{ih}^u , $\nabla \theta_{ih}$, $\nabla \theta_{ih}^a$, $\nabla \theta_{ih}^d$, and ∇t_{ih} remain to be calculated. From (31)

$$\nabla t_{ih} = \sum_{q=1}^{h-1} \nabla \left[\omega_{iq} + t^a_{iq} + t^u_{iq} + t^d_{iq} \right] + \nabla \omega_{ih} \qquad (39)$$

and obviously

$$\frac{\partial \omega_h}{\partial \theta_{iq}} = 0 \tag{40}$$

and

$$\begin{cases} \frac{\partial \omega_{ih}}{\partial \omega_{ih}} = 1\\ \frac{\partial \omega_{ih}}{\partial \omega_{q}} = 0 \text{ for } q = 1, \dots, h - 1, h + 1, \dots, K_i. \end{cases}$$
(41)

From Proposition 2, the existence of the maximum velocity modes is determined by the length of $|\theta_{i(h+1)} - \theta_{ih}|$. Thus, the motion between θ_{ih} and $\theta_{i(h+1)}$ can be divided into two categories according to Proposition 2.

Case 3.1: If $t_{ih}^u > 0$, the maximal velocity mode exists at a time interval $[t_{ih} + t_{ih}^a, t_{ih} + t_{ih}^a + t_{ih}^u)$ (refer to Fig. 1) when agent *i* moves from θ_{ih} to $\theta_{i(h+1)}$.

In this case, agent *i* first moves from θ_{ih} to θ_{ih}^a with fixed acceleration C_i^a , then moves to θ_{ih}^d with fixed velocity v_i^{max} , and finally moves to $\theta_{i(h+1)}$ with fixed deceleration C_i^d . According

to the kinematics law, the following conditions hold:

$$\begin{cases} t_{ih}^{a} = \frac{v_{ia}^{\max}}{C_{i}^{a}} \\ t_{ih}^{d} = \frac{v_{ia}^{\max}}{C_{i}^{d}} \\ t_{ih}^{u} = \frac{|\theta_{i(h+1)} - \theta_{ih}| - \left(t_{ih}^{a} \frac{v_{ia}^{\max}}{2} + t_{ih}^{d} \frac{v_{ia}^{\max}}{2}\right)}{v_{ia}^{\max}}. \end{cases}$$
(42)

Therefore, taking the derivative with respect to the parameters, we obtain:

$$\begin{aligned} \nabla t_{ih}^a &= 0 \\ \nabla t_{ih}^d &= 0 \end{aligned} \tag{43}$$

$$\frac{\partial t_{ih}^{u}}{\partial \theta_{i(h+1)}} = \frac{\operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih})}{v_{i}^{\max}}$$

$$\frac{\partial t_{ih}^{u}}{\partial \theta_{ih}} = -\frac{\operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih})}{v_{i}^{\max}}$$

$$\frac{\partial t_{ih}^{u}}{\partial \theta_{iq}} = 0, q = 1, \dots, h - 1, h + 2, K_{i}$$
(44)

and

$$\frac{\partial t_{ih}^u}{\partial \omega_{iq}} = 0, q = 1, \dots, K_i.$$
(45)

In addition, the associated control switching point from maximal acceleration mode to maximal velocity mode is θ_{ih}^a and from maximal velocity mode to maximal deceleration mode is θ_{ih}^d that can be calculated as follows. The switching points between θ_{ih} and $\theta_{i(h+1)}$ are θ_{ih}^a , θ_{ih}^d , where

$$\theta_{ih}^{a} = \theta_{ih} + \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih})t_{ih}^{a}\frac{v_{i}^{\max}}{2}$$
$$\theta_{ih}^{d} = \theta_{i(h+1)} - \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih})t_{ih}^{d}\frac{v_{i}^{\max}}{2}.$$
 (46)

Subsequently

$$\begin{cases} \frac{\partial \theta_{ih}^{a}}{\partial \theta_{iq}} = 1\\ \frac{\partial \theta_{ih}^{a}}{\partial \theta_{iq}} = 0, q = 1, \dots, h - 1, h + 1, \dots, K_{i} \end{cases}$$
(47)

$$\frac{\partial \theta_{ih}^a}{\partial \omega_{iq}} = 0, q = 1, \dots, K_i \tag{48}$$

$$\begin{cases} \frac{\partial \theta_{ih}^d}{\partial \theta_{i(h+1)}} = 1\\ \frac{\partial \theta_{ih}^d}{\partial \theta_{iq}} = 0, q = 1, \dots, h, h+2, \dots, K_i \end{cases}$$
(49)

and

$$\frac{\partial \theta_{ih}^d}{\partial \omega_{iq}} = 0, q = 1, \dots, K_i.$$
(50)

Case 3.2: If $t_{ih}^u = 0$, the maximal velocity mode does not exist when agent *i* moves from θ_{ih} to $\theta_{i(h+1)}$. The velocity of agent *i* does not have enough time to accelerate to v_i^{max} or it can increase to v_i^{max} but then decrease immediately.

In this case, agent *i* leaves from θ_{ih} with acceleration C_i^a until θ_{ih}^a ($\theta_{ih}^a = \theta_{ih}^d$) and then moves with deceleration C_i^d until $\theta_{i(h+1)}$. According to the kinematics law of uniformly variable motion, the following conditions hold:

$$\begin{cases} C_i^a t_{ih}^a = C_i^d t_{ih}^d \\ \frac{1}{2} C_i^a (t_{ih}^a)^2 + \frac{1}{2} C_i^d (t_{ih}^d)^2 = |\theta_{i(h+1)} - \theta_{ih}|. \end{cases}$$
(51)

Solving (51) in terms of control switching time t_{ih}^a and t_{ih}^d , we have

$$\begin{cases} t_{ih}^{a} = \sqrt{\frac{2C_{i}^{d}}{C_{i}^{a}(C_{i}^{a}+C_{i}^{d})}} |\theta_{i(h+1)} - \theta_{ih}| \\ t_{ih}^{d} = \sqrt{\frac{2C_{i}^{a}}{C_{i}^{d}(C_{i}^{d}+C_{i}^{a})}} |\theta_{i(h+1)} - \theta_{ih}| \\ t_{ih}^{u} = 0. \end{cases}$$
(52)

Taking the derivative we obtain

$$\nabla t^u_{ih} = 0 \tag{53}$$

$$\begin{cases} \frac{\partial t_{ih}^{a}}{\partial \theta_{i(h+1)}} = \sqrt{\frac{2C_{i}^{d}}{C_{i}^{a}(C_{i}^{a}+C_{i}^{d})}} \frac{\operatorname{sgn}(\theta_{i(h+1)}-\theta_{ih})}{2\sqrt{|\theta_{i(h+1)}-\theta_{ih}|}} \\ \frac{\partial t_{ih}^{a}}{\partial \theta_{ih}} = -\sqrt{\frac{2C_{i}^{d}}{C_{i}^{a}(C_{i}^{a}+C_{i}^{d})}} \frac{\operatorname{sgn}(\theta_{i(h+1)}-\theta_{ih})}{2\sqrt{|\theta_{i(h+1)}-\theta_{ih}|}} \\ \frac{\partial t_{ih}^{a}}{\partial \theta_{ig}} = 0, q = 1, \dots, h-1, h+2, \dots, K_{i} \end{cases}$$
(54)

$$\frac{\partial t_{ih}^a}{\partial \omega_{ig}} = 0, q = 1, \dots, K_i$$
(55)

$$\begin{cases} \frac{\partial t_{ih}^{d}}{\partial \theta_{i(h+1)}} = \sqrt{\frac{2C_{i}^{d}}{C_{i}^{a}(C_{i}^{a}+C_{i}^{d})}} \frac{\operatorname{sgn}(\theta_{i(h+1)}-\theta_{ih})}{2\sqrt{|\theta_{i(h+1)}-\theta_{ih}|}} \\ \frac{\partial t_{ih}^{d}}{\partial \theta_{ih}} = -\sqrt{\frac{2C_{i}^{d}}{C_{i}^{a}(C_{i}^{a}+C_{i}^{d})}} \frac{\operatorname{sgn}(\theta_{i(h+1)}-\theta_{ih})}{2\sqrt{|\theta_{i(h+1)}-\theta_{ih}|}} \\ \frac{\partial t_{ih}^{d}}{\partial \theta_{iq}} = 0, q = 1, \dots, h-1, h+2, \dots, K_{i} \end{cases}$$
(56)

and

$$\frac{\partial t_{ih}^d}{\partial \omega_{iq}} = 0, q = 1, \dots, K_i.$$
(57)

Moreover, in this case, $\theta_{ih}^a = \theta_{ih}^d$

$$\theta_{ih}^{a} = \theta_{ih} + \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih}) \frac{1}{2} C_{i}^{a} t_{ih}^{a}^{2}$$
$$\nabla \theta_{ih}^{a} = \nabla \theta_{ih} + \operatorname{sgn}(\theta_{i(h+1)} - \theta_{ih}) C_{i}^{a} t_{ih}^{a} \nabla t_{ih}^{a}.$$
(58)

Replacing ∇t^a_{ih} with (54) and (55)

$$\begin{cases} \frac{\partial \theta_{ih}^{a}}{\partial \theta_{ih}} = 1 - \frac{C_{i}^{d}}{(C_{i}^{a} + C_{i}^{d})} \\ \frac{\partial \theta_{ih}^{a}}{\partial \theta_{i(h+1)}} = \frac{C_{i}^{d}}{(C_{i}^{a} + C_{i}^{d})} \\ \frac{\partial \theta_{ih}^{a}}{\partial \theta} = 0, q = 1, \dots, h - 1, h + 1, \dots, K_{i} \end{cases}$$
(59)

$$\frac{\partial \theta_{ih}^{i}}{\partial \theta_{iq}} = 0, q = 1, \dots, h - 1, h + 1, \dots, K_i$$

$$\frac{\partial \theta_{ih}^a}{\partial \omega_{iq}} = 0, q = 1, \dots, K_i.$$
(60)

This ends the analysis of Step 3) in which $\nabla t_{ih}^a, \nabla t_{ih}^u, \nabla \theta_{ih}, \nabla \theta_{ih}^a, \nabla \theta_{ih}^d$, and ∇t_{ih}^a are obtained. In addition, for $k \neq i$ and $\frac{\partial s_i(t)}{\partial \theta_k} = 0$, it has $\frac{\partial s_i(t)}{\partial \omega_k} = 0$.

In summary, Steps 2) and 3) show the calculation of $\nabla S(t)$.

Based on the above-mentioned analysis, $\nabla J(\theta, \omega)$ is obtained and the gradient-based iterative algorithm is designed as illustrated in Algorithm 1.

Remark 4: Note that our gradient-based solution depends on the initial trajectories of the agents and the time horizon. Moreover, for a certain trajectory of agent i, the total number of dwelling points K_i is related to the time horizon T. Based on the analysis of Step 3), the dwelling points and corresponding dwelling time will be updated constantly until the optimal

Algorithm 1: Gradient-Based Iteration Algorithm.

Initialization: The maximal velocities, accelerations and decelerations of agents, v_i^{max} , C_i^a , C_i^d . The initial trajectories of all agents $[\theta(0), \omega(0)]$, a given terminal condition ε , a sufficient small computation step δ , and the index of iterations m = 1.

1: repeat

- 2: Calculate S(t) using (4)(15) and $[\theta(m), \omega(m)]$;
- 3: for j = 1 : M do
- Calculate $R(x_j, t)$ according to (6); 4:
- 5: if $\dot{R}(x_i, t)$ switches then
- 6: Calculate $\nabla R(x_i, t)$ according to (29) and (30); 7: else
- 8: if $R(x_i, t)$ does not switch then
- 9: Calculate $\nabla R(x_i, t)$ according to (25) by activating Algorithm 2 to calculate $\nabla S(t)$; end if
- 10: 11: end if
- 12: end for
- 13: Calculate ∇J according to (24);
- 14: Update $[\theta(m), \omega(m)]$ according to (22) and set m = m + 1;

15: **until**
$$|J(\theta(m+1), \omega(m+1)) - J(\theta(m), \omega(m))| < \varepsilon;$$

16: The optimal trajectory: $|\theta, \check{\omega}| = |\theta(m), \omega(m)|$;

Algorithm 2: Calculation of $\nabla S(t)$.	
Input: Time t, trajectories $[\theta(m), \omega(m)]$ and $S(t)$.	
1:	for $i = 1 : N$ do
2:	Get the length of vector $\theta_i(m)$, denoted by H ;
3:	for $h = 1 : H$ do
4:	if $ heta_{i(h+1)} - heta_{ih} > rac{v_i^{\max^2}}{2C_i^a} + rac{v_i^{\max^2}}{2C_i^d}$ then
5:	Calculate $\nabla t_{ih}^a, \nabla t_{ih}^u, \nabla t_{ih}^d, \nabla \theta_{ih}^a, \nabla \theta_{ih}^a, \nabla \theta_{ih}^d$
	according to (43), (44), (45), (47), (48), (49), (50);
6:	else
7:	$\mathbf{if} \left \theta_{i(h+1)} - \theta_{ih} \right \leq \frac{v_i^{\max^2}}{2C_i^a} + t_{ih}^d \frac{v_i^{\max^2}}{2C_i^d} \mathbf{then}$
8:	Calculate $\nabla t_{ih}^a, \nabla t_{ih}^u, \nabla t_{ih}^d, \nabla \theta_{ih}, \nabla \theta_{ih}^a, \nabla \theta_{ih}^d$
	according to (53), (54), (55), (56),(57), (59), (60)
9:	end if
10:	end if
11:	Calculate $\nabla \omega_{ih}$ according to (40),(41);
12:	Calculate ∇t_{ih} according to (39);
13:	end for
14:	end for
15:	Calculate $\nabla S(t)$ according to (33), (35), (37), (38).

trajectory is obtained. Therefore, the total number of dwelling points K_i will change during the iterative process of specifying the optimal trajectory.

C. Collision Avoidance

In [23], since the agent can change its velocity instantly, the optimal trajectory will never visit the boundary points and avoid collisions as shown in [23, Propositions III.1 and III.4]. However, in this paper, the agent will endure a deceleration process before it can fully stop and change its moving direction, which will affect the optimal trajectory. Therefore, theoretically it may be a better strategy for the agent to cross the boundary points or its neighbor rather than stopping before crossing.

Note that in some 1-D application settings, collisions are inherently avoided, e.g., in a transportation network where vehicles move in parallel lanes or in a waterway where multiple agents can move along the same 1-D mission space. However, in some cases the physical system does not allow an agent to exceed the boundary points 0 and L of the mission space or to cross its neighbor by running in different lanes. Therefore, in this section, we will present a modified algorithm to avoid boundary crossing and collisions.

Introduce two virtual agents with indexes 0 and N+1where $s_0(t) = 0$ and $s_{N+1}(t) = L, \forall t \in [0, T]$. Let $\{tc_k\}, k =$ $1, 2, \ldots$, denote the crossing time instance of all agents in the resulting trajectory of Algorithm 1, which satisfies

$$\prod_{i=0}^{N} \left[s_i(tc_k) - s_{i+1}(tc_k) \right] = 0$$
(61)

and let ζ be the safe distance between two agents. In (61), the case of border exceeding corresponds to i = 0 and i = N, and the case of trajectory collision between agents corresponds to $i = 1, 2, \ldots, N - 1.$

We propose Algorithm 3 that checks and finds the earliest crossing time tc_1 of the trajectory if there is any, and adjusts the trajectory by adding a safety distance to the corresponding dwelling points that cause the collision. By repeating the check and update procedure, eventually we will obtain a suboptimal but collision-free trajectory. Note that Algorithm 3 is an ad hoc mechanism for avoiding collisions and boundary crossings; a more rigorous treatment of this issue remains the subject of continuing research.

IV. ILLUSTRATIVE EXAMPLES

In this section, three simulation examples are presented for persistent monitoring in a 1-D space, to show the normal case, the boundary crossing case and agent collision case. These examples aim at, respectively,

- 1) illustrating the optimal trajectory under second-order dynamics with physical constraints;
- 2) verifying the result of Proposition 1;
- 3) demonstrating the effectiveness of the gradient-based it-

eration algorithms and the collision avoidance algorithm. In the simulation, there are three aspects that should be paid attention to the following.

- 1) The overflow problem. Since it is impossible to know the number of optimal dwelling points in advance, the dimensions of both $\theta_i(m)$ and $\omega_i(m)$ are unknown; it is, therefore, necessary to choose the dimensions large enough and use 0 entries to fill in the vectors as needed.
- 2) The step-sizes $[\theta, \tilde{\omega}]$. Diminishing step-size should be applied as the performance metric is approaching to the optimal value.
- 3) The calculation of derivative. In the numerical simulations, in order to calculate $R(s(t), x_i)$ in Algorithm

Algorithm 3: Collision Avoidance Algorithm.

Input: The parametric trajectory $[\dot{\theta}, \dot{\omega}]$ obtained by Algorithm 1 and a safety distance ζ .

- 1: repeat
- 2: Calculate the agent physical trajectory S(t) using (4), (15) and $[\theta, \check{\omega}]$;
- According to (61), calculate the first collision time 3: tc_1 and find the corresponding colliding agents i and i + 1.
- if $tc_1 \in (t_{\tilde{i}h_{\tilde{z}}}, t_{\tilde{i}(h_{\tilde{z}}+1)}]$ and $tc_1 \in$ 4: $(t_{(\tilde{i}+1)h_{\tilde{i}+1}}, t_{(\tilde{i}+1)(h_{\tilde{i}+1}+1)}]$ then

5: Set
$$\check{\theta}_{\tilde{i}(h_{\tau}+1)} = s_{\tilde{i}}(tc_1) - \zeta/2$$
 for $\tilde{i} \neq 0$.

6: Set
$$\check{\theta}_{(\tilde{i}+1)(h_{\tilde{i}+1}+1)} = s_{\tilde{i}+1}(tc_1) + \zeta/2$$
 for $\tilde{i} \neq N$

- Update the trajectory $[\check{\theta}, \check{\omega}]$ based on the above 7: adjustment.
- 8: end if
- 9: Execute Algorithm 1 to optimize the updated trajectory $[\dot{\theta}, \check{\omega}]$ but replace step 13 in Algorithm 1 with following steps from 10 to 21.
- 10: for $i = 1: N \operatorname{do}$

11: Calculate
$$h_i$$
 satisfying $tc_1 \in (t_{ih_i}, t_{i(h_i+1)})$
12: **for** $h = 1 : h_i + 1$ **do**

- for $h = 1 : h_i + 1$ do $\frac{\partial J}{\partial(\theta_{ih}, \omega_{ih})} = 0.$ end for 13:
- 14:
- 15: if $i = \tilde{i}$ or $i = \tilde{i} + 1$ then

Calculate
$$\frac{\partial J}{\partial \omega_{i(h_i+1)}}$$
 according to (24)

17: end if

16:

18: **for**
$$h = (h_i + 2) : K_i$$
 de

- **r** $h = (h_i + 2) : K_i$ **do** Calculate $\frac{\partial J}{\partial(\theta_{ih}, \omega_{ih})}$ according to (24). 19:
- end for 20:
- 21: end for
- 22: **until** (61) does not hold for any $t \in [0, T]$ along the agent trajectories.
- 23: The sub-optimal collision-free trajectory: $[\theta^*, \omega^*] =$ $[\theta, \check{\omega}].$

1, (6) is modified as follows. If $R(x_i, t_k) = 0$ and $I(x_j) - D(x_j)P(x_j, t_k) < 0, \ \dot{R}(x_j, t_{k+1}) = 0.$ Otherwise, $\dot{R}(x_i, t_{k+1}) = I(x_i) - D(x_i)P(x_i, t_{k+1}).$

In the following examples, the uncertainty dynamics of sampling points remain the same, and the initial value of its uncertainty is $R(x_i, 0) = 0, j = 1, 2..., M$. For the agents used in the examples, the effective sensing ranges are $r_i = 1$, the maximum acceleration and deceleration are $C_i^a = 0.5, C_i^d = 0.5,$ and the maximum velocity $v_i^{\text{max}} = 1.5, i = 1, 2, \dots, N$. The error tolerance $\varepsilon = 1.0 \times 10^{-4}$ in the termination condition (23). The persistent monitoring time horizon is 55 s.

Example 1. The normal case: In the normal case, the interested targets are evenly distributed. The persistent monitoring task is executed by one or two agents, respectively. The simulation results of the two persistent monitoring examples are shown in Figs. 2 and 3, respectively.

First, in Fig. 2, the persistent monitoring task is executed by single agent. The mission space is [0,10] and the



Fig. 2. Persistent monitoring task executed by single agent. (a) Initial trajectory (green dashed) and optimal trajectory (blue solid) obtained by Algorithm 1. (b) Optimal velocity (blue solid) and acceleration (red dash-dotted) of agent 1. (c) Performance metric J_1 (decreases as the number of iterations increases).



Fig. 3. Persistent monitoring task executed by two agents. (a) Initial trajectory (green dashed) and optimal trajectory (blue solid) obtained by Algorithm 1. (b) Optimal velocity (blue solid) and acceleration (red dash-dotted) of agent 1. (c) Performance metric J_2 (decreases as the number of iterations increases).

set of target points of interest is $X = \{x_j\}, x_j = 0 + 0.5j, j = 1, 2, ..., M, M = 20$ with the increasing and decreasing rates $I(x_j) = 0.1$ and $D(x_j) = 0.5$, respectively. The step-sizes $[\tilde{\theta}, \tilde{\omega}] = [0.02, 0.01]$ for the 1st seven iterations, $[\tilde{\theta}, \tilde{\omega}] = [0.01, 0.005]$ between the 8th and the 17th iterations and $[\tilde{\theta}, \tilde{\omega}] = [0.002, 0.001]$ after the 17th iteration. The initial

trajectory is $\theta(0) = [0, 10, 0, 10, ...], \quad \omega(0) = [0.2, 0.2, ...]$ shown by the green dashed line in Fig. 2(a).

As we can see, the optimal trajectory is found and shown by the blue solid line in Fig. 2(a). Comparing with the simulation results in [23], the obtained optimal trajectory is smooth in this paper, and the velocity of agent is continuous [see the veloc-



Fig. 4. Persistent monitoring task executed by single agent. (a) Initial trajectory (green dashed) and optimal trajectory (blue solid) obtained by Algorithm 1. (b) Performance metric J_1 (decreases as the number of iterations increases).

ity profile: blue solid line in Fig. 2(b)]. This demonstrates that our second-order model better approximates the agent behavior in practice. The optimal velocity and acceleration are shown in Fig. 2(b). $u(t) = \pm 1$ in the acceleration/deceleration modes and u(t) = 0 in the maximal velocity and dwell modes that is consistent with Proposition 1. Moreover, there exists the maximal velocity mode in Fig. 2(b) since $|\theta_{1(h+1)} - \theta_{1h}| > \frac{v_1^{\max^2}}{2C_1^a} + \frac{v_1^{\max^2}}{2C_1^d}$, which verifies the conclusion of Proposition 2. According to Algorithm 1, the performance metric decreases as the number of iterations increases in Fig. 2(c), which demonstrates the effectiveness of our gradient-based algorithm. At the 28th iteration, the performance metric $J_1(\theta(28), \omega(28)) = 63.72$ satisfies the terminal condition $|J_1(\theta(28), \omega(28)) - J_1(\theta(27), \omega(27))| < \varepsilon$.

In Fig. 3, the persistent monitoring task executed by two agents. The mission space is [0,5] and the set of



Fig. 5. Persistent monitoring task executed by single agent. (a) Initial trajectory (green dashed) and optimal trajectory (blue solid) obtained by Algorithm 1. (b) Evolution of the Performance metric J.

target points of interest is $X = \{x_j\}, x_j = 0 + 0.25j, j = 1, 2, \ldots, M, M = 20$ with the increasing and decreasing rates $I(x_j) = 0.1$ and $D(x_j) = 0.5$, respectively. The step-sizes are $[\tilde{\theta}, \tilde{\omega}] = [0.06, 0.03]$ for the 1st three iterations, $[\tilde{\theta}, \tilde{\omega}] = [0.008, 0.004]$ between the 4th and the 10th iterations and $[\tilde{\theta}, \tilde{\omega}] = [0.002, 0.001]$ after the 10th iteration. The initial trajectories are $\theta_1(0) = [0, 2.4, 0, \ldots], \omega_1(0) = [0.2, 0.2, \ldots]$ and $\theta_2(0) = [5, 2.6, 5, \ldots], \omega_2(0) = [0.2, 0.2, \ldots]$ shown by the green dashed line in Fig. 3(a).

The optimal trajectories are shown by the blue solid line in Fig. 3(a) and the optimal velocity (blue solid) and acceleration (red dash-dotted) of agent 1 are shown in Fig. 3(b). Please note that the distances between the switching points do not satisfy condition in Proposition 2, then the maximal velocity mode does not exist and the velocity of agent 1 cannot increase to v_1^{max} . The performance metric decreases as the number of iterations increases in Fig. 3(c) with the final cost $J_2(\theta(13)\omega(13)) = 4.709$ and terminal condition $|J_2(\theta(13), \omega(13)) - J_2(\theta(12), \omega(12))| < \varepsilon$.





Fig. 6. Persistent monitoring task executed by two agents. (a) Initial trajectory (green dashed) and optimal trajectory (blue solid) obtained by Algorithm 1. (b) Performance metric J_2 (decreases as the number of iterations increases).

Example 2. The border exceeding case: In this simulation, the persistent monitoring task is executed by one single agent. An extreme case is considered in this task. The mission space is [0, 6], the set of target points of interest is X = [0, 0.1, 5.9, 6] with the increasing and decreasing rates I(X) = [0.5, 0.1, 0.1, 0.5] and D(X) = [1, 0.5, 0.5, 1], respectively. The step-sizes $[\tilde{\theta}, \tilde{\omega}] = [0.006, 0.002]$ for the first 7 iterations, $[\tilde{\theta}, \tilde{\omega}] = [0.003, 0.001]$ between the 8th and the 20th iterations and $[\tilde{\theta}, \tilde{\omega}] = [0.001, 0.0005]$ after the 20th iteration. The initial trajectory is $\theta_1(0) = [1, 5, 1, \ldots]$ and $\omega_1(0) = [0.3, 0.3, \ldots]$ shown by the green dashed line in Fig. 4 (a).

The optimal trajectory obtained by Algorithm 1 is shown in Fig. 4(a) and the performance metric is shown in Fig. 4(b), the final cost $J_1(\theta(50), \omega(50)) = 205.9$, and the terminal condition $|J_1(\theta(50), \omega(50)) - J_1(\theta(49), \omega(49))| < \varepsilon$.

From Fig. 4(a), we can see that the agent will exceed the border of mission space (shown by red dash-dotted line). However, for cases where border crossing is not allowed, the strategy described in Algorithm 3 is applied. The corresponding simulation result is shown in Fig. 5(a), in which the agent never exceeds

Fig. 7. Persistent monitoring task executed by two agents. (a) Initial trajectory (green dashed) and optimal trajectory (blue solid) obtained by Algorithm 1. (b) Evolution of Performance metric J.

the border of the mission space. In order to show the evolution of the performance metric, let round i be the result of each round when an border exceeding is found and Algorithm 1 is called to obtain a trajectory for this round in Algorithm 3. From Fig. 5(b), it is obvious that the value of J increases as more collision incidents are detected and eliminated (note that a total of seven such incidents were detected as shown in the figure). This is quite reasonable since we modify the optimal trajectory by keeping a safety distance to avoid exceeding border constraints.

Example 3. The agent collision case: In this simulation, the persistent monitoring task is executed by two agents. The mission space is [-1, 14] and the set of interested targets is X = [0, 0.2, 0.4, 0.6, 6.2, 12, 12.2, 12.4, 12.6] with the increasing and decreasing rates I(X) = [0.1, 0.1, 0.1, 0.1, 0.5, 0.1, 0.1, 0.1, 0.1] and D(X) = [0.5, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 0.5], respectively. The step-sizes $[\tilde{\theta}, \tilde{\omega}] = [0.008, 0.004]$ for the 1st two iterations, $[\tilde{\theta}, \tilde{\omega}] = [0.002, 0.001]$ for the 3rd iteration and $[\tilde{\theta}, \tilde{\omega}] = [0.0006, 0.0003]$ after the 3rd iteration. The initial trajectories are $\theta_1(0) = [1, 7, 1, \ldots], \omega_1(0) = [0.1, 0.1, \ldots]$ and $\theta_2(0) = [13, 6, 13, \ldots], \omega_2(0) = [0.1, 0.1, \ldots]$, which are

shown by the green dashed lines in Fig. 6(a). The optimal trajectories obtained by Algorithm 1 are shown by the blue solid lines in Fig. 6(a) and the cost as a function of iteration is shown in Fig. 6(b). The final cost $J_2(\theta(15), \omega(15)) = 56.95$, and the terminal condition $|J_2(\theta(15), \omega(15)) - J_2(\theta(14), \omega(14))| < \varepsilon$.

From Fig. 6(a), we can see that the trajectories of the two agents cross each other. However, if this is not allowed, the collision avoidance algorithm (see Algorithm 3) is applied to solve this trajectory collision problem. The corresponding simulation result is shown in Fig. 7(a), in which the agents never collide with each other. From Fig. 7(b), the value of J increases as more collision incidents are detected and eliminated (note that a total of four such incidents were detected in this example), a behavior similar to that observed in Example 2.

V. CONCLUSION

In this paper, optimal persistent monitoring tasks are performed via second-order multiple agents. The results of this paper bring persistent monitoring one step closer to realistic applications in the sense that the existing results are improved in the following three aspects:

- 1) the physical constraints on both the velocity and the acceleration are taken into consideration;
- the control is on the acceleration leading to smooth agent trajectories;
- 3) a collision avoidance algorithm is proposed to solve potential collision and boundary-crossing problems.

Our future work is to extend this model to 2-D spaces [35] considering the presence of obstacles and the possibility of controlling agents in a distributed manner.

REFERENCES

- S. Alamdari, E. Fata, and S. L. Smith, "Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 139–155, 2012.
- [2] W. Meng, Z. He, R. Teo, and R. Su, "Integrated multi-agent system framework: decentralised search, tasking and tracking," *IET Control Theory Appl.*, vol. 9, no. 4, pp. 493–502, 2014.
- [3] D. Kingston, R. W. Beard, and R. S. Holt, "Decentralized perimeter surveillance using a team of UAVs," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1394–1404, Dec. 2008.
- [4] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks," in *Proc. 26th IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 342–349.
- [5] V. Srivastava and F. Bullo, "Stochastic surveillance strategies for spatial quickest detection," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1438–1458, 2013.
- [6] R. Reshma, T. Ramesh, and P. Sathishkumar, "Security situational aware intelligent road traffic monitoring using UAVs," in *Proc. 2nd IEEE Int. Conf. VLSI Syst., Arch., Technol. Appl.*, 2016, pp. 1–6.
- [7] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Secur. Privacy*, vol. 8, no. 1, pp. 81–85, Jan./Feb. 2010.
- [8] V. C. Gungor *et al.*, "A survey on smart grid potential applications and communication requirements," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 28–42, Feb. 2013.
- [9] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *J. Field Robot.*, vol. 28, no. 5, pp. 714–741, 2011.
- [10] 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," *J. Field Robot.*, vol. 27, no. 6, pp. 718–740, 2010.

- [11] D. E. Soltero, S. L. Smith, and D. Rus, "Collision avoidance for persistent monitoring in multi-robot systems with intersecting trajectories," in *Proc. 25th IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, vol. 32, no. 14, pp. 3645–3652.
- [12] C. Song, L. Liu, G. Feng, and S. Xu, "Optimal control for multiagent persistent monitoring," *Automatica*, vol. 50, no. 6, pp. 1663–1668, 2014.
- [13] S. L. Smith, M. Schwager, and D. Rus, "Persistent monitoring of changing environments using a robot with limited range sensing," in *Proc. 28th IEEE Int. Conf. Robot. Automat.*, 2011, pp. 5448–5455.
- [14] X. Lan and M. Schwager, "A variational approach to trajectory planning for persistent monitoring of spatiotemporal fields," in *Proc. Amer. Control Conf.*, 2014, pp. 5627–5632.
- [15] D. Portugal and R. P. Rocha, "Multi-robot patrolling algorithms: Examining performance and scalability," *Adv. Robot. J.*, vol. 27, no. 5, pp. 325– 336, 2013.
- [16] H. L. Choi and J. P. How, "Continuous trajectory planning of mobile sensors for informative forecasting," *Automatica*, vol. 46, no. 8, pp. 1266– 1275, 2010.
- [17] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and it application to persistent monitoring tasks," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1106–1118, Oct. 2016.
- [18] N. Zhou, X. Yu, S. B. Andersson, and C. G. Cassandras, "Optimal eventdriven multi-agent persistent monitoring of a finite set of targets," in *Proc.* 55th IEEE Conf. Decis. Control, 2016, pp. 1814–1819.
- [19] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 410–426, Apr. 2012.
- [20] J. Yu, S. Karaman, and D. Rus, "Persistent monitoring of events with stochastic arrivals at multiple stations," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 521–535, Jun. 2015.
- [21] 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. 35th Amer. Control Conf.*, 2017, pp. 5544–5549.
- [22] X. Lin and C. G. Cassandras, "An optimal control approach to the multiagent persistent monitoring problem in two-dimensional spaces," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1659–1664, Jun. 2015.
- [23] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 947–961, Apr. 2013.
- [24] H. Wei and S. Ferrari, "A geometric transversals approach to sensor motion planning for tracking maneuvering targets," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2773–2778, Oct. 2015.
- [25] G. Foderaro, P. Zhu, H. Wei, T. A. Wettergren, and S. Ferrari, "Distributed optimal control of sensor networks for dynamic target tracking," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 142–153, Mar. 2016.
- [26] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 592–606, Jun. 2012.
- [27] M. Jadaliha and J. Choi, "Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 899–905, May 2013.
- [28] A. Jones, M. Schwager, and C. Belta, "Information-guided persistent monitoring under temporal logic constraints," in *Proc. 33rd Amer. Control Conf.*, 2015, pp. 1911–1916.
- [29] G. Foderaro, S. Ferrari, and T. A. Wettergren, "Distributed optimal control for multi-agent trajectory optimization," *Automatica*, vol. 50, no. 1, pp. 149–154, 2014.
- [30] S. Adlakha and M. Srivastava, "Critical density thresholds for coverage in wireless sensor networks," in *Proc. 3rd IEEE Wireless Commun. Netw. Conf.*, 2003, vol. 3, pp. 1615–1620.
- [31] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Rev.*, vol. 37, no. 2, pp. 181–218, 2008.
- [32] A. E. Bryson and Y. C. Ho, Applied Optimal Control: Optimization, Estimation, and Control. New York, NY, USA: Wiley, 1975.
- [33] M. Athans and M. Canon, "On the fuel-optimal singular control of nonlinear second-order systems," *IEEE Trans. Autom. Control*, vol. AC-9, no. 4, pp. 360–370, Oct. 1964.
- [34] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and Y. Chen, "Perturbation analysis and optimization of stochastic hybrid systems," *Eur. J. Control*, vol. 16, no. 6, pp. 642–664, 2010.
- [35] N. Zhou, C. G. Cassandras, X. Yu, and S. B. Andersson, "Optimal eventdriven multi-agent persistent monitoring with graph-limited mobility," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2181–2186, 2017.



Yan-Wu Wang (M'10–SM'13) received the B.S. degree in automatic control and the M.S. and Ph.D. degrees in control theory and control engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1997, 2000, and 2003, respectively.

She is currently a Professor with the School of Automation, HUST, and with the Key Laboratory of Image Processing and Intelligent Control, Ministry of Education, Wuhan, China. Her research interests include hybrid systems, copulligent automa with applications in appendix

operative control, and multiagent systems, with applications in smart grid.

Dr. Wang was a recipient of several awards, including the first prize of Hubei Province Natural Science in 2014, the first prize of the Ministry of Education of China in 2005, and the Excellent Ph.D. Dissertation of Hubei Province in 2004, China. In 2008, she was awarded the title of "New Century Excellent Talents" by the Chinese Ministry of Education.



Nan Zhou (S'13) received the B.S. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2011 and the M.S. degree in control engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014. He is currently working toward the Ph.D. degree in systems engineering at Boston University, Boston, MA, USA.

His research interests include optimal control of hybrid systems, cooperative control, and multiagent systems, with applications in robotics, mobile sensor networks, and smart cities.

Mr. Zhou is a student member of the IEEE Control Systems Society and Robotics Automation Society.



Yao-Wen Wei received the B.S. degree in 2015 from the School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan, China, where he is currently working toward the M.S. degree.

His research interests include optimization and persistent monitoring.

timization volved in the o to 1996, he wa Computer Eng MA, USA. He i Boston Univers



Xiao-Kang Liu received the B.S. degree in 2014 from the School of Automation, Huazhong University of Science and Technology, Wuhan, China, where he is currently working toward the Ph.D. degree.

His research interests include output regulation, hybrid system, and smart grid.



Christos G. Cassandras (F'96) received the B.S. degree in engineering and applied science from Yale University, New Haven, CT, USA, in 1977, the M.S.E.E. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1978, and the M.S. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, MA, USA, in 1979 and 1982, respectively.

From 1982 to 1984, he was with ITP Boston, Inc., Cambridge, MA, USA, where he was in-

volved in the design of automated manufacturing systems. From 1984 to 1996, he was a faculty member with the Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA, USA. He is currently a Distinguished Professor of engineering with Boston University, Brookline, MA, USA, the Head of the Division of Systems Engineering, and a Professor with the Department of Electrical and Computer Engineering. He specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has authored more than 400 refereed papers in these areas, and six books.

Dr. Cassandras is currently a member of Phi Beta Kappa and Tau Beta Pi. He is also a Fellow of the International Federation of Automatic Control (IFAC). He was a recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the 2006 Distinguished Member Award of the IEEE Control Systems Society, the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook), a 2011 prize and a 2014 prize for the IBM/IEEE Smarter Planet Challenge competition, the 2014 Engineering Distinguished Scholar Award at Boston University, several honorary professorships, a 1991 Lilly Fellowship, and a 2012 Kern Fellowship. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, from 1998 to 2009. He serves on several editorial boards and has been a Guest Editor for various journals. He was the President of the IEEE Control Systems Society in 2012.