Collision-Free Trajectory Design for 2D Persistent Monitoring Using Second-Order Agents

Yan-Wu Wang, *Senior Member, IEEE,* Ming-Jie Zhao, Wu Yang, Nan Zhou, *Student Member, IEEE* and Christos G. Cassandras, *Fellow, IEEE*

***Abstract*—This paper considers a two-dimensional persistent monitoring problem by controlling movements of second-order agents to minimize some uncertainty metric associated with targets in a dynamic environment. In contrast to common sensing models depending only on the distance from a target, we introduce an active sensing model which considers the distance between an agent and a target, as well as the agent's velocity. We propose an objective function which can achieve a collision-free agent trajectory by penalizing all possible collisions. Applying structural properties of the optimal control derived from the Hamiltonian analysis, we limit agent trajectories to a simpler parametric form under a family of 2D curves depending on the problem setting, e.g. ellipses and Fourier trajectories. Our collision-free trajectories are optimized through an event-driven Infinitesimal Perturbation Analysis (IPA) and gradient descent method. Although the solution is generally locally optimal, this method is computationally efficient and offers an alternative to other traditional time-driven methods. Finally, simulation examples are provided to demonstrate our proposed results.**

***Index Terms*—persistent monitoring, second-order agents, two-dimensional space, obstacle avoidance, collision-free trajectory.**

I. Introduction

In recent years, attention has been drawn to persistent monitoring because of its wide range of applications such as ocean sampling [1], [2], city surveillance [3], [4], and traffic monitoring [5], [6]. Persistent monitoring often arises in a large dynamically changing environment which requires agents to move to cover the monitoring region. The fundamental problem of persistent monitoring is to design suitable motion strategies for agents to meet the monitoring requirements cooperatively. Up to now, there have been many studies on persistent monitoring in both 1D [7], 2D [8] and 3D [9] spaces. For the 1D problems, [10] recently addresses a 1D persistent monitoring problem which only involves a finite number of targets and designed optimal trajectories for agents. Moreover, [11] solves the 1D persistent monitoring utilizing second-order agents with physical constraints, e.g. bounded acceleration and velocity.

Compared with the 1D problems, the 2D monitoring problems find more practical applications; the control strategies are more complicated; and the optimal solutions are more challenging to obtain. [12] restricts agent trajectories to ellipses and proved that they outperform the linear ones under certain conditions in a 2D mission space. [13] studies monitoring finite stationary targets distributed nearby a given closed trajectory in 2D space. And [14] presents optimal speed controllers along a given closed path to stabilize a changing environment.

Different from 1D and 2D problems, the key to solving 3D monitoring is not to focus on trajectories of agents, but to model it as a scheduling or visiting problem. [15] casts a persistent surveillance problem as a Vehicle Routing Problem with Time Windows, and develops a locally optimal path within a set time horizon, which then is executed repeatedly. [16] designs an optimal scheduling scheme for one robot tasked to monitor several events that are occurring at different locations.

The aforementioned literature deals with the persistent monitoring problem without obstacles in the mission space. However, in many applications, the existence of obstacles restricts the movements of agents, thus exacerbating the complexity of the problem. [11] proposes a collision avoidance algorithm by repeatedly adjusting the designed agent trajectories. And [17] solves collision problems in 2D monitoring through stopping policies on given closed paths. Further, in related research on coverage control [18], there are some results involving avoiding obstacles. Specifically, [19] achieves collision avoidance through a bounded repulsive avoidance control law. [20] keeps robots at a safety distance from obstacles through speed functions. However, such methods of avoiding obstacles are not applicable in persistent monitoring. Thus the design of collision-free trajectories is still an open question.

Motivated by the above discussion, we will first introduce the persistent monitoring problem using second-order agents and then consider all possible collisions (agent to agent, agent to obstacle). Our goal is to i) find a solution to an optimal control problem for second-order agents visiting targets with different weights, while also avoiding all possible types of collisions. Through standard optimal control theoretic techniques [21], we show that agents move with maximal accelerations; ii) translate the optimal control problem into a tractable parametric optimization problem. Inspired by [12], we rep-

This work is supported by the National Natural Science Foundation of China under Grants 61773172 and 51537003, the Natural Science Foundation of Hubei Province of China (2017CFA035), the Fundamental Research Funds for the Central Universities (2018KFYYXJJ119) and the Program for HUST Academic Frontier Youth Team.

Y.-W. Wang, M.-J. Zhao and W. Yang are with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, 430074, China, and also with the National Key Laboratory of Science and Technology on Multi-Spectral Information Processing (Huazhong University of Science and Technology), China (e-mail: wangyw@hust.edu.cn; zhaomj@hust.edu.cn; wuyangac@hust.edu.cn; biwuyang@163.com).

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).

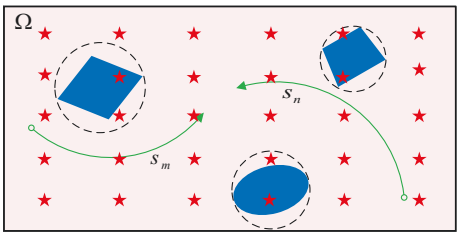


Fig. 1. Persistent monitoring mission space. Blue areas are obstacles. Red stars represent target points. The curves s_m and s_n represent feasible agent trajectories.

resent agent trajectories using various families of parametric curves. iii) determine optimal parametric agent trajectories under the parametric families used. IPA [10] is utilized for the gradient evaluation and the gradients are then used for optimization through gradient descent methods. Moreover, the IPA gradient is updated in an event-driven manner, hence not requiring continuous time-driven updates. Note that our algorithm optimizes the agent trajectories while ensuring agent safety by penalizing all possible collisions.

The main contributions of this paper can be summarized as follows:

- Second-order agents are utilized to execute 2D monitoring task, which brings our treatment of persistent monitoring one step closer to realistic applications. Moreover, bounds with respect to agent accelerations and velocities are both considered.
- Targets with different weight coefficients are considered, which is different from [12] where all targets are homogeneous.
- A new sensing model is constructed, which is quite different from those limited to dependence only on the distance between an agent and a target in [7], [10]–[12]. Our sensing model depends not only on relative distances between agents and targets but also on agents' velocities [22], [23], and finds use in applications such as vision-based monitoring [24], [25].
- Obstacles in the mission space are considered. Thus, we propose a novel objective function which leads to a collision-free solution by penalizing all possible collisions.

The rest of the paper is organized as follows. In Section II, an optimal control formulation for the persistent monitoring problem is proposed. In Section III, some preliminary insights about the solution are given through Hamiltonian analysis. Section IV parameterizes the agent trajectories under a selected family of 2D parametric curves and optimize the parameters using an IPA-based gradient descent algorithm. Section V provides simulations and Section VI concludes the paper.

II. Persistent Monitoring Problem Formulation

Consider a 2D rectangular mission space $\mathbf{\Omega} \equiv [0, L_1] \times [0, L_2] \subset \mathbb{R}^2$ as shown in Fig. 1. Select $\{z_i = [z_i^x, z_i^y] \in \mathbf{\Omega}, i = 1, ..., M\}$ as target points to be monitored. Note that some target points may be located within obstacles. We assume sensors can monitor these points without going into obstacles. This can model, for example, sensors located outside buildings which can detect targets inside the buildings.

A. Agent Dynamics

We consider N cooperating agents assigned to monitor stationary target points over the time horizon $[0, T]$. The dynamics of agent n can be described by

$$\dot{x}_n(t) = f(x_n, U_n) = \begin{bmatrix} \mathbf{0}_{2,2} & \mathbf{I}_2 \\ \mathbf{0}_{2,2} & \mathbf{0}_{2,2} \end{bmatrix} x_n(t) + \begin{bmatrix} \mathbf{0}_{2,2} \\ \mathbf{I}_2 \end{bmatrix} U_n(t) \tag{1}$$

where $n = 1, ..., N$, $\mathbf{0}_{2,2}$ is a 2×2 zero matrix, $\mathbf{I}_2$ is a 2×2 identity matrix and U_n is the control input. The state $x_n(t) = [s_n(t), v_n(t)]^\top$, where $s_n(t) = [s_n^x(t), s_n^y(t)]^\top$ and $v_n(t) = [v_n^x(t), v_n^y(t)]^\top$ represents the position and the velocity of the agent n at time t respectively. In addition, we define the control input $U_n(t) = [u_n(t)\cos\theta_n(t), u_n(t)\sin\theta_n(t)]^\top$, where $u_n(t)$ is the magnitude of the acceleration and $\theta_n(t)$ is the agent's heading that satisfies $0 \le \theta_n(t) < 2\pi$. Without loss of generality, we assume that the velocity and the acceleration of each agent are bounded such that

$$\begin{aligned} 0 \le \|v_n(t)\| \le v_n^{max}, \ n = 1, ..., N \\ 0 \le \|u_n(t)\| \le u_n^{max}, \ n = 1, ..., N \end{aligned} \tag{2}$$

where v_n^{max} is subject not only to the maximum power constraint [11], but also to the performance of the sensor aboard the agent (which will be mentioned in the following).

Remark 1. Note that the control input $u_n(t)$ here is the acceleration of the agent n as opposed to the velocity in [12]. Moreover, as will be seen in the next section, following the centralized determination of optimal trajectories, the monitoring task is executed in an open-loop fashion by each agent.

B. Agent Sensing Model

According to [22], [23], the detection probability of a sensor is inversely proportional to the movement velocity. Thus, here we assume that the actual sensing strength of a sensor is determined by the distance between a sensor and a target, as well as its velocity. Specifically, a closer distance and a smaller velocity lead to higher sensing strength. The sensing effectiveness is zero when the distance exceeds a finite sensing range r_n or the velocity of the agent exceeds a certain threshold β_n ($> v_n^{max}$ to avoid meaningless agent movements). Define a function $p_n(z_i)$ which represents the probability that an event at location $z_i = [z_i^x, z_i^y]$ is detected by agent n. In addition to those properties of the sensing model considered in [12], $p_n(z_i)$ is also monotonically non-increasing function in the agent's velocity.

$$p_n(z_i) = \begin{cases} (1 - \frac{D(z_i, s_n)}{r_n})(1 - \frac{\|v_n\|}{\beta_n}), \\ \text{if } D(z_i, s_n) \le r_n \text{ and } \|v_n\| \le \beta_n \\ \\ 0, \text{if } D(z_i, s_n) > r_n \text{ or } \|v_n\| > \beta_n \end{cases} \tag{3}$$

where $D(z_i, s_n) = \|z_i - s_n\|$ is the Euclidean distance between the agent n and the target position z_i, and v_n is the agent velocity (we assume that targets are stationary).

Since there may be multiple agents, the joint probability that an event occurring at z_i is detected, denoted by $P_i(\mathbf{x}(t))$, is given by

$$P_i(\mathbf{x}(t)) = 1 - \prod_{n=1}^{N}[1 - p_n(z_i)] \tag{4}$$

where $\mathbf{x}(t) = [\mathbf{s}(t), \mathbf{v}(t)]^\top$ with $\mathbf{s}(t) = [s_1(t), ..., s_N(t)]$ and $\mathbf{v}(t) = [v_1(t), ..., v_N(t)]$.

C. Target Dynamics

We associate an uncertainty function $R_i(t)$ with every target point z_i, which possesses similar properties to the model in [12]: (*i*) $R_i(t)$ increases with a prespecified rate A_i if $P_i(t) = 0$, i.e. there is no agent detecting target z_i; (*ii*) $R_i(t)$ decreases with a fixed rate B if $P_i(t) = 1$; (*iii*) the decrease of $R_i(t)$ is proportional to the joint probability $P_i(\mathbf{x}(t))$; (*iv*) $R_i(t) \geq 0$ for all t. Thus, the dynamics of $R_i(t)$ are

$$\dot{R}_i(t) = \begin{cases} 0, & \text{if } R_i(t) = 0,\ A_i \leq BP_i(\mathbf{x}(t)) \\ A_i - BP_i(\mathbf{x}(t)), & \text{otherwise} \end{cases} \tag{5}$$

We further assume that initial conditions $R_i(0), i = 1, ..., M$, are given and that $B > A_i > 0$ for stability.

D. Optimal Control Problem

Our goal is to minimize the uncertainty accumulated across all target points. We define $J_1(t)$ to be the weighted sum of target uncertainties:

$$J_1(t) = \sum_{i=1}^{M} \sigma_i R_i(t) \tag{6}$$

The weight coefficients σ_i are set to capture the relative importance of different targets. Note that the problem setting in [12], where $\sigma_i = 1,\ \text{for } i = 1, ..., M$, is a special case of our setting here.

Moreover, in contrast to [12] where each agent is represented as a point mass and collisions among agents are ignored, we will consider the sizes of agents in this work. Note that to avoid collisions in persistent monitoring settings, any two agents cannot share the same location at the same time instant. Considering the size of each agent, for agent n we define a safety radius $\rho_n > 0$, and the corresponding safety disk $Q_n = \{x \in \mathbf{\Omega} \mid \|x - s_n(t)\| \leq \rho_n\}$. We consider that a collision occurs between agents p and q at some location only if $Q_p \cap Q_q \neq \emptyset$. Obviously, to avoid agent collisions, we must ensure that the Euclidean distance $d_{pq}(t) = \|s_p(t) - s_q(t)\| \geq \rho_p + \rho_q$ at all times. To capture the collisions among agents, we define

$$d_{pq}^-(t) = \min(0, d_{pq}(t) - \rho_p - \rho_q) \tag{7}$$

First, $d_{pq}^-(t) \leq 0$ for all $t \in [0, T]$. Second, a collision-free trajectory satisfies $d_{pq}^-(t) = 0$ for all t. Considering all possible collisions among agents, we define the agent collision avoidance component $J_2(t)$ of the objective function as follows:

$$J_2(t) = \sum_{q=p+1}^{N} \sum_{p=1}^{N-1} d_{pq}^-(t) \tag{8}$$

In addition, the presence of obstacles in the mission space (as shown in Fig. 1) usually restricts the feasible trajectories. This brings another level of difficulty to the optimization problem. For simplicity, we ignore shapes of the obstacles by covering them using circumscribed circles (shown as dashed circles in Fig. 1) whose centers are denoted by $\omega_l = [\omega_l^x, \omega_l^y]$ and radii by $r_l, l = 1, ..., L$. Conservatively, we can ensure the safety of agent n by letting the Euclidean distance $d_{ln}(t) = \|\omega_l - s_n(t)\| \geq r_l + \rho_n$. Similar to (7), we define

$$d_{ln}^-(t) = \min(0, d_{ln}(t) - r_l - \rho_n) \tag{9}$$

Note that $d_{ln}^-(t) \leq 0$ for all $t \in [0, T]$. Similar to (7), a collision-free trajectory satisfies $d_{ln}^-(t) = 0$ for all t. Considering all agents and obstacles, we define the obstacle collision avoidance component $J_3(t)$ of the objective function as follows:

$$J_3(t) = \sum_{n=1}^{N} \sum_{l=1}^{L} d_{ln}^-(t) \tag{10}$$

Now we are ready to formulate the optimal persistent monitoring problem to obtain collision-free agent trajectories so that the cumulative uncertainty over all weighted target points $\{z_1, ..., z_M\}$ in the mission space is minimized over a fixed time horizon T. Let $\mathbf{u}(t) = [u_1(t), ..., u_N(t)]^\top$ and $\boldsymbol{\theta}(t) = [\theta_1(t), ..., \theta_N(t)]^\top$. Then the optimal control problem **P1** is formulated as follows:

$$\textbf{P1:}\ \min_{\mathbf{u}(t), \boldsymbol{\theta}(t)} J = \frac{1}{T}\int_0^T \Big(J_1(t) + M_2 J_2(t) + M_3 J_3(t)\Big) dt \tag{11}$$

where M_2 and M_3 are large negative numbers.

Remark 2. The objective function is designed on the basis of imposing collisions penalties [26]. Along a collision-free trajectory both $J_2(t)$ and $J_3(t)$ are zeros. The large negative numbers M_2 and M_3 penalize all forms of collision (agent to agent, agent to obstacle) along the agent trajectory.

III. Optimal Control Solution

In this section, we use optimal control theory [21] to derive necessary conditions for the optimal solutions of **P1**. We define the state vector

$$\begin{aligned} \mathbf{y}(t) = &[s_1^x(t), s_1^y(t), ..., s_N^x(t), s_N^y(t), v_1^x(t), v_1^y(t), ..., \\ &v_N^x(t), v_N^y(t), R_1(t), ..., R_M(t)]^\top \end{aligned} \tag{12}$$

and the associated costate vector

$$\begin{aligned} \boldsymbol{\lambda}(t) = &[\lambda_1^x(t), \lambda_1^y(t), ..., \lambda_N^x(t), \lambda_N^y(t), \mu_1^x(t), \mu_1^y(t), ..., \\ &\mu_N^x(t), \mu_N^y(t), \gamma_1(t), ..., \gamma_M(t)]^\top \end{aligned} \tag{13}$$

In addition, we introduce $\boldsymbol{\eta} = [\eta_1(t), ..., \eta_N(t)]^\top$ to handle the inequality constraint on the state variable $v_n(t)$ such that

$$\begin{cases} \eta_n = 0,\ \text{if } \|v_n(t)\| < v_n^{max} \\ \eta_n > 0,\ \text{if } \|v_n(t)\| = v_n^{max} \end{cases} \tag{14}$$

The Hamiltonian is given by

$$\begin{aligned} H = &\Big(J_1(t) + M_2 J_2(t) + M_3 J_3(t)\Big) + \sum_{i=1}^{M} \gamma_i(t)\dot{R}_i(t) \\ &+ \sum_{n=1}^{N} \lambda_n^x(t) v_n^x(t) + \sum_{n=1}^{N} \lambda_n^y(t) v_n^y(t) \\ &+ \sum_{n=1}^{N} \mu_n^x(t) u_n(t) \cos\theta_n(t) + \sum_{n=1}^{N} \mu_n^y(t) u_n(t) \sin\theta_n(t) \\ &+ \sum_{n=1}^{N} \eta_n(t)(\|v_n(t)\| - v_n^{max}) \end{aligned} \tag{15}$$

and the costate equations can be obtained through $\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{y}}$.

For the case $\mu_n^y(t) = 0$, we get

$$\begin{aligned} H = &\Big(J_1(t) + M_2 J_2(t) + M_3 J_3(t)\Big) + \sum_{i=1}^{M} \gamma_i(t)\dot{R}_i(t) \\ &+ \sum_{n=1}^{N} \lambda_n^x(t) v_n^x(t) + \sum_{n=1}^{N} \lambda_n^y(t) v_n^y(t) \\ &+ \sum_{n=1}^{N} \mu_n^x(t) u_n(t) \cos\theta_n(t) + \sum_{n=1}^{N} \eta_n(t)(\|v_n(t)\| - v_n^{max}) \end{aligned} \tag{16}$$

For the case $\mu_n^y(t) \neq 0$, after some trigonometric operations on (15), we get

$$\begin{aligned} H = &\Big(J_1(t) + M_2 J_2(t) + M_3 J_3(t)\Big) + \sum_{i=1}^{M} \gamma_i(t)\dot{R}_i(t) \\ &+ \sum_{n=1}^{N} \lambda_n^x(t) v_n^x(t) + \sum_{n=1}^{N} \lambda_n^y(t) v_n^y(t) \\ &+ \sum_{n=1}^{N} \mathrm{sgn}(\mu_n^y(t)) u_n(t) \sqrt{(\mu_n^x(t))^2 + (\mu_n^y(t))^2} \\ &\times \left[\frac{\mathrm{sgn}(\mu_n^y(t)) \mu_n^x(t) \cos\theta_n(t)}{\sqrt{(\mu_n^x(t))^2 + (\mu_n^y(t))^2}} + \frac{|\mu_n^y(t)| \sin\theta_n(t)}{\sqrt{(\mu_n^x(t))^2 + (\mu_n^y(t))^2}} \right] \\ &+ \sum_{n=1}^{N} \eta_n(t)(\|v_n(t)\| - v_n^{max}) \end{aligned} \tag{17}$$

where sgn$(\cdot)$ represents the sign function. Combining the trigonometric function terms, we obtain

$$\begin{aligned} H = &\Big(J_1(t) + M_2 J_2(t) + M_3 J_3(t)\Big) + \sum_{i=1}^{M} \gamma_i(t)\dot{R}_i(t) \\ &+ \sum_{n=1}^{N} \lambda_n^x(t) v_n^x(t) + \sum_{n=1}^{N} \lambda_n^y(t) v_n^y(t) \\ &+ \sum_{n=1}^{N} \mathrm{sgn}(\mu_n^y(t)) u_n(t) \sqrt{(\mu_n^x(t))^2 + (\mu_n^y(t))^2} \sin\big(\theta_n(t) \\ &+ \Psi_n(t)\big) + \sum_{n=1}^{N} \eta_n(t)\big(\|v_n(t)\| - v_n^{max}\big) \end{aligned} \tag{18}$$

where $\Psi_n(t)$ is defined so that $\tan\Psi_n(t) = \frac{\mu_n^x(t)}{\mu_n^y(t)}$ for $\mu_n^y(t) \neq 0$. In the following discussion, we assume that any "singular interval" where both $\mu_n^x(t) = 0$ and $\mu_n^y(t) = 0$ is excluded.

Note that the initial states of this problem are given but the terminal states are free. Thus, the terminal costate are $\gamma_i(T) = 0, i = 1, ..., M$, $\lambda_n^x(T) = \lambda_n^y(T) = 0$ and $\mu_n^x(T) = \mu_n^y(T) = 0, n = 1, ..., N$. According to [21], the minimal solution can be obtained by solving the Two Point Boundary Value Problem. Applying the Pontryagin Minimum Principle to (16) and (18) with $u_n^\star(t)$, $\theta_n^\star(t)$, $t \in [0, T)$ denoting the optimal controls, we have

$$H(\mathbf{y}^\star, \boldsymbol{\lambda}^\star, \mathbf{u}^\star, \boldsymbol{\theta}^\star) = \min_{\mathbf{u}(t), \boldsymbol{\theta}(t)} H(\mathbf{y}, \boldsymbol{\lambda}, \mathbf{u}, \boldsymbol{\theta}) \tag{19}$$

The necessary conditions for the optimal controls are:

$$\|u_n^\star(t)\| = u_n^{max} \tag{20}$$

and

$$\begin{cases} \sin(\theta_n^\star(t) + \Psi_n(t)) = 1, & \text{if } \mu_n^y(t) < 0 \\ \sin(\theta_n^\star(t) + \Psi_n(t)) = -1, & \text{if } \mu_n^y(t) > 0 \\ \cos\theta_n^\star(t) = 1, & \text{if } \mu_n^y(t) = 0 \text{ and } \mu_n^x(t) < 0 \\ \cos\theta_n^\star(t) = -1, & \text{if } \mu_n^y(t) = 0 \text{ and } \mu_n^x(t) > 0 \end{cases} \tag{21}$$

Remark 3. Unlike the 1D persistent monitoring analysis in [11] where $u_i^*(t)$ depends absolutely on the sign of the costate $\lambda_{u_i}^*(t)$, in 2D $u_n^\star(t) = u_n^{max}$ and the orientation $\theta_n^\star(t)$ depends on the sign of costate variables.

Note that in (2) and (14), if $\|v_n(t)\| = v_n^{max}$, then $\dot{v}_n(t) = 0$. In particular, as long as the agent n reaches its maximal velocity, it will keep the maximal velocity moving.

Revisiting the Hamiltonian in (15) and according to [21], $\theta_n^\star(t)$ can be obtained by solving:

$$\frac{\partial H}{\partial \theta_n} = -\mu_n^x(t) u_n(t) \sin\theta_n(t) + \mu_n^y(t) u_n(t) \cos\theta_n(t) = 0 \tag{22}$$

from which we obtain

$$\tan\theta_n^\star(t) = \frac{\mu_n^y(t)}{\mu_n^x(t)} \tag{23}$$

So far, we know $u_n^\star(t) = u_n^{max}$ and there is only $\theta_n^\star(t)$ left to be evaluated. This can be accomplished by solving a standard Two Point Boundary Value Problem (TPBVP), which involves a forward integration of the states and a backward integration of the costates. However, solving the TPBVP problem is computationally intractable as numbers of agents and targets increase. Furthermore, the switches on the state dynamics and the presence of obstacles exacerbate the computational complexity. Therefore, we will search alternatives in the next section.

IV. Agent Trajectory Parameterization And Optimization

The result of [12] indicates that under some assumptions an elliptical trajectory outperforms a linear one when using the average uncertainty metric as a comparison criterion. In fact, elliptical trajectories degenerate to linear ones when the minor axis of the ellipse becomes zero. Based on the result

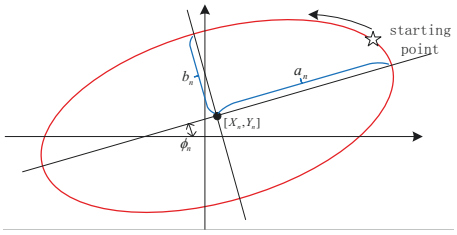


Fig. 2. Illustrations of the parametric form of an elliptical trajectory.

that elliptical trajectories are smooth and periodic, and are more suitable for 2D monitoring [12], we also select them for agents to execute the monitoring task. Under the optimal control derived in Section III, the agent first accelerates along the elliptical trajectory with the maximal acceleration u_n^{max}. Ever since it reaches the maximal velocity, it maintains the maximal velocity along the monitoring task.

A. Elliptical Trajectories

We assign each agent an elliptical trajectory parameterized by center coordinate, major axis, minor axis, and orientation and rewrite **P1** as a parametric optimization problem.

Define an elliptical trajectory such that the agent position $s_n(t) = [s_n^x(t), s_n^y(t)]$ follows the general parametric form of the ellipse:

$$\begin{aligned} s_n^x(t) &= X_n + a_n \cos\varphi_n(t)\cos\phi_n - b_n \sin\varphi_n(t)\sin\phi_n \\ s_n^y(t) &= Y_n + a_n \cos\varphi_n(t)\sin\phi_n + b_n \sin\varphi_n(t)\cos\phi_n \end{aligned} \tag{24}$$

As illustrated in Fig. 2, $[X_n, Y_n]$ is the center of the ellipse, a_n and b_n are the radii of the major and minor axes respectively, $\phi_n \in [0, 2\pi)$ is the orientation, and $\varphi_n(t)$ is the eccentric anomaly. Upon initializing its position at the starting point, the agent always moves along the ellipse.

Taking the derivatives of (24), we obtain the velocity information of the agent as follows,

$$\begin{aligned} \dot{s}_n^x(t) &= -\dot{\varphi}_n(t)\big(a_n \sin\varphi_n(t)\cos\phi_n + b_n \cos\varphi_n(t)\sin\phi_n\big) \\ \dot{s}_n^y(t) &= -\dot{\varphi}_n(t)\big(a_n \sin\varphi_n(t)\sin\phi_n - b_n \cos\varphi_n(t)\cos\phi_n\big) \end{aligned} \tag{25}$$

Again, taking the derivatives of (25), we get the acceleration information of the agent as follows,

$$\begin{aligned} \ddot{s}_n^x(t) =& -a_n \cos\phi_n\big(\ddot{\varphi}_n(t)\sin\varphi_n(t) + \dot{\varphi}_n(t)^2\cos\varphi_n(t)\big) \\ & - b_n \sin\phi_n\big(\ddot{\varphi}_n(t)\cos\varphi_n(t) - \dot{\varphi}_n(t)^2\sin\varphi_n(t)\big) \\ \ddot{s}_n^y(t) =& -a_n \sin\phi_n\big(\ddot{\varphi}_n(t)\sin\varphi_n(t) + \dot{\varphi}_n(t)^2\cos\varphi_n(t)\big) \\ & + b_n \cos\phi_n\big(\ddot{\varphi}_n(t)\cos\varphi_n(t) - \dot{\varphi}_n(t)^2\sin\varphi_n(t)\big) \end{aligned} \tag{26}$$

Since the agent moves with constant acceleration u_n^{max} on the elliptical trajectory (if not at the maximal velocity), we have

$$\ddot{s}_n^x(t)^2 + \ddot{s}_n^y(t)^2 = (u_n^{max})^2 \tag{27}$$

$$\dot{s}_n^x(t)^2 + \dot{s}_n^y(t)^2 = \|v_n(t)\|^2 \tag{28}$$

for some $t \in [0, T]$, where $v_n(t)$ is the current velocity of agent n. Applying (26) to (27) and (25) to (28), we can obtain $\ddot{\varphi}_n(t)$ and $\dot{\varphi}_n(t)$, respectively.

Once the agent reaches the maximal velocity, it will maintain this maximal velocity thereafter and we have

$$\dot{s}_n^x(t)^2 + \dot{s}_n^y(t)^2 = (v_n^{max})^2 \tag{29}$$

for some $t \in [0, T]$. Applying (25) to (29), we obtain $\dot{\varphi}_n(t)$.

Based on these processes, $\varphi_n(t), t \in [0, T]$, can be obtained through iterations, thus the complete trajectory of the agent on the ellipse can be obtained.

B. Optimal Trajectory Design

The trajectories of all agents can be parameterized by $\Theta = [\Theta_1, ..., \Theta_N]^\top$ with $\Theta_n = [X_n, Y_n, a_n, b_n, \phi_n], n = 1, ..., N$. Therefore, the objective function (11) can be rewritten as $J(\Theta)$. We seek to obtain $\Theta^\star = [\Theta_1^\star, ..., \Theta_N^\star]^\top$ by minimizing $J(\Theta)$. We use the standard gradient descent method [27] as follows,

$$\Theta^h = \Theta^{h-1} - \alpha^h \nabla J(\Theta^{h-1}) \tag{30}$$

where α^h is a suitable step size and $\nabla J(\Theta^{h-1})$ is the gradient of J with respect to $\Theta = [\Theta_1, ..., \Theta_N]^\top$. The trajectory parameters can be optimized iteratively through (30) and the terminal condition is given by

$$|J(\Theta^h) - J(\Theta^{h-1})| < \varepsilon \tag{31}$$

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

C. Gradient Calculation

***1)* Infinitesimal Perturbation Analysis**. Now we briefly review Infinitesimal Perturbation Analysis (IPA) as applied in [10]. IPA calculates the gradient of the objective function for a hybrid dynamic system which contains events leading to possible discontinuities on the gradient. Denote the continuous states of the hybrid system by $\boldsymbol{\chi}(t)$, the k-th event time by ρ_k and the controllable parameter by $\boldsymbol{\Theta}$. For each $[\rho_k, \rho_{k+1})$, the state dynamics is continuous and can be written as $\dot{\boldsymbol{\chi}} = f_k(\boldsymbol{\chi}, \boldsymbol{\Theta}, t)$. The events at ρ_k can be classified into three categories:

- Exogenous events. These events will cause a discrete state transition and there exists $\frac{d\rho_k}{d\boldsymbol{\Theta}} = 0$.
- Endogenous events. These events occur when a continuous differentiable guarding function $g_k(\Theta, \boldsymbol{\chi}) = 0$.
- Induced events. These events are triggered by another event occurring at an earlier time.

Let $\boldsymbol{\chi}'(t) = \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{\Theta}}$ and $\rho_k' = \frac{\partial \rho_k}{\partial \boldsymbol{\Theta}}$. The IPA shows that $\boldsymbol{\chi}'(t)$ satisfies:

$$\frac{d}{dt}\boldsymbol{\chi}'(t) = \frac{\partial f_k}{\partial \boldsymbol{\chi}}\boldsymbol{\chi}'(t) + \frac{\partial f_k}{\partial \boldsymbol{\Theta}}, \quad t \in [\rho_k, \rho_{k+1}) \tag{32}$$

with boundary condition:

$$\boldsymbol{\chi}'(\rho_k^+) = \boldsymbol{\chi}'(\rho_k^-) + [f_{k-1}(\rho_k^-) - f_k(\rho_k^+)]\rho_k' \tag{33}$$

If the event at ρ_k is exogenous, then $\rho_k' = 0$. Otherwise, if the event at ρ_k is endogenous, then ρ_k' satisfies

$$\rho_k' = -\Big[\frac{\partial g_k}{\partial \boldsymbol{\chi}} f_k(\rho_k^-)\Big]^{-1}\Big[\frac{\partial g_k}{\partial \boldsymbol{\Theta}} + \frac{\partial g_k}{\partial \boldsymbol{\chi}}\boldsymbol{\chi}'(\rho_k^-)\Big] \tag{34}$$

TABLE I
OPTIMIZATION PROBLEM EVENT SET

Event Set	Illustration
1. ξ_i^0	$R_i(t)$ hits 0, for $i = 1, ..., M$
2. ξ_i^+	$R_i(t)$ leaves 0, for $i = 1, ..., M$
3. u_n^0	$u_n(t)$ switches from u_n^{max} to 0, for $n = 1, ..., N$
4. ζ_{pq}^0	$d_{pq}^-(t)$ hits 0, for $p = 1, ..., N-1, q = p+1, ..., N$
5. ζ_{pq}^-	$d_{pq}^-(t)$ leaves 0, for $p = 1, ..., N-1, q = p+1, ..., N$
6. δ_{ln}^0	$d_{ln}^-(t)$ hits 0, for $l = 1, ..., L$, $n = 1, ..., N$
7. δ_{ln}^-	$d_{ln}^-(t)$ leaves 0, for $l = 1, ..., L$, $n = 1, ..., N$

as long as $\frac{\partial g_k}{\partial \boldsymbol{\chi}} f_k(\rho_k^-) \neq 0$. Then for $t \in [\rho_k, \rho_{k+1})$, $\boldsymbol{\chi}'(t)$ can be determined by

$$\boldsymbol{\chi}'(t) = \boldsymbol{\chi}'(\rho_k^+) + \int_{\rho_k}^{t} \frac{d}{dt}\boldsymbol{\chi}'(t)dt \tag{35}$$

Recalling Section II, we summarize all events which may trigger the state transitions in our optimization problem in TABLE I. The superscript 0 indicates an event causing a variable to change from a non-zero value to zero; the superscript $+$ indicates an event causing a variable to change from zero to a positive value; and the superscript $-$ indicates an event causing a variable to change from zero to a negative value). Define E as a set of "event types" that can be associated with either an agent or a target. We use $e(\rho_k) \in E$ to represent the event happened at time ρ_k.

2) **Gradient computation using IPA**. We write the parametric form of the objective function in (11) as

$$J(\Theta) = \frac{1}{T}\int_0^T \Big(J_1(\Theta) + M_2 J_2(\Theta) + M_3 J_3(\Theta)\Big)dt \tag{36}$$

Define a time sequence $\{\rho_k(\Theta), k = 1, ..., K\}$ (with$\rho_0 = 0$ and $\rho_{K+1} = T$) to describe all switching instants. Then $\nabla J(\Theta)$ can be written as

$$\begin{aligned}\nabla J(\Theta) = \frac{1}{T}\sum_{k=0}^{K}\Big[\int_{\rho_k(\Theta)}^{\rho_{k+1}(\Theta)} \Big(&\nabla J_1(\Theta) \\ &+ M_2\nabla J_2(\Theta) + M_3\nabla J_3(\Theta)\Big)dt\Big]\end{aligned} \tag{37}$$

The evaluation of $\nabla J(\Theta)$ therefore depends entirely on $\nabla J_1(\Theta)$, $\nabla J_2(\Theta)$ and $\nabla J_3(\Theta)$. Note that there may be discontinuities in these derivatives and the effects of such discontinuities can be captured by IPA. Next we will focus on seeking $\nabla J_1(\Theta_n)$, $\nabla J_2(\Theta_n)$ and $\nabla J_3(\Theta_n)$ for every agent n. Referring to (6), (8) and (10), we can get

$$\nabla J_1(\Theta_n) = \sum_{i=1}^{M}\sigma_i \nabla R_i(t) \tag{38}$$

$$\nabla J_2(\Theta_n) = \sum_{p\neq n, p\in\{1,...,N\}} \nabla d_{pn}^-(t) \tag{39}$$

$$\nabla J_3(\Theta_n) = \sum_{l=1}^{L} \nabla d_{ln}^-(t) \tag{40}$$

We need to calculate $\nabla R_i(t) = [\frac{\partial R_i(t)}{\partial X_n}, \frac{\partial R_i(t)}{\partial Y_n}, \frac{\partial R_i(t)}{\partial a_n}, \frac{\partial R_i(t)}{\partial b_n}, \frac{\partial R_i(t)}{\partial \phi_n}]^\top$, $\nabla d_{pn}^-(t) = [\frac{\partial d_{pn}^-(t)}{\partial X_n}, \frac{\partial d_{pn}^-(t)}{\partial Y_n}, \frac{\partial d_{pn}^-(t)}{\partial a_n}, \frac{\partial d_{pn}^-(t)}{\partial b_n}, \frac{\partial d_{pn}^-(t)}{\partial \phi_n}]^\top$ and $\nabla d_{ln}^-(t) = [\frac{\partial d_{ln}^-(t)}{\partial X_n}, \frac{\partial d_{ln}^-(t)}{\partial Y_n}, \frac{\partial d_{ln}^-(t)}{\partial a_n}, \frac{\partial d_{ln}^-(t)}{\partial b_n}, \frac{\partial d_{ln}^-(t)}{\partial \phi_n}]^\top$.

2.1) **Calculate** $\nabla R_i(t)$. From (5) and (35), for $t \in [\rho_k, \rho_{k+1})$, we can obtain

$$\nabla R_i(t) = \nabla R_i(\rho_k^+) + \begin{cases} 0, \text{ if } R_i(t) = 0, A_i < BP_i(\mathbf{x}(t)) \\ \int_{\rho_k}^{t} \frac{d}{dt}\nabla R_i(t)dt, \qquad \text{otherwise} \end{cases} \tag{41}$$

where the integral term calculated by (32) gives

$$\frac{d}{dt}\nabla R_i(t) = -B\nabla P_i(\mathbf{x}(t)) \tag{42}$$

From (4), we can get

$$\begin{aligned}\nabla P_i(\mathbf{x}(t)) = &\frac{\partial P_i(\mathbf{x}(t))}{\partial s_n^x(t)}\nabla s_n^x(t) + \frac{\partial P_i(\mathbf{x}(t))}{\partial s_n^y(t)}\nabla s_n^y(t) \\ &+ \frac{\partial P_i(\mathbf{x}(t))}{\partial v_n^x(t)}\nabla v_n^x(t) + \frac{\partial P_i(\mathbf{x}(t))}{\partial v_n^y(t)}\nabla v_n^y(t)\end{aligned} \tag{43}$$

where the derivative terms in (43) can be easily obtained from (4). Then, $\nabla s_n^x(t)$, $\nabla s_n^y(t)$, $\nabla v_n^x(t)$ and $\nabla v_n^y(t)$ can be obtained from (24) and (25) as follows,

$$\begin{aligned}&\frac{\partial s_n^x(t)}{\partial X_n} = 1, \quad \frac{\partial s_n^x(t)}{\partial a_n} = \cos\varphi_n(t)\cos\phi_n \\ &\frac{\partial s_n^x(t)}{\partial Y_n} = 0, \quad \frac{\partial s_n^x(t)}{\partial b_n} = -\sin\varphi_n(t)\sin\phi_n \\ &\frac{\partial s_n^x(t)}{\partial \phi_n} = -a_n\cos\varphi_n(t)\sin\phi_n - b_n\sin\varphi_n(t)\cos\phi_n\end{aligned} \tag{44}$$

$$\begin{aligned}&\frac{\partial s_n^y(t)}{\partial X_n} = 0, \quad \frac{\partial s_n^y(t)}{\partial a_n} = \cos\varphi_n(t)\sin\phi_n \\ &\frac{\partial s_n^y(t)}{\partial Y_n} = 1, \quad \frac{\partial s_n^y(t)}{\partial b_n} = \sin\varphi_n(t)\cos\phi_n \\ &\frac{\partial s_n^y(t)}{\partial \phi_n} = a_n\cos\varphi_n(t)\cos\phi_n - b_n\sin\varphi_n(t)\sin\phi_n\end{aligned} \tag{45}$$

$$\begin{aligned}&\frac{\partial v_n^x(t)}{\partial X_n} = 0, \quad \frac{\partial v_n^x(t)}{\partial a_n} = -\dot{\varphi}_n(t)\sin\varphi_n(t)\cos\phi_n \\ &\frac{\partial v_n^x(t)}{\partial Y_n} = 0, \quad \frac{\partial v_n^x(t)}{\partial b_n} = -\dot{\varphi}_n(t)\cos\varphi_n(t)\sin\phi_n \\ &\frac{\partial v_n^x(t)}{\partial \phi_n} = \dot{\varphi}_n(t)\big(a_n\sin\varphi_n(t)\sin\phi_n - b_n\cos\varphi_n(t)\cos\phi_n\big)\end{aligned} \tag{46}$$

$$\begin{aligned}&\frac{\partial v_n^y(t)}{\partial X_n} = 0, \quad \frac{\partial v_n^y(t)}{\partial a_n} = -\dot{\varphi}_n(t)\sin\varphi_n(t)\sin\phi_n \\ &\frac{\partial v_n^y(t)}{\partial Y_n} = 0, \quad \frac{\partial v_n^y(t)}{\partial b_n} = \dot{\varphi}_n(t)\cos\varphi_n(t)\cos\phi_n \\ &\frac{\partial v_n^y(t)}{\partial \phi_n} = -\dot{\varphi}_n(t)\big(a_n\sin\varphi_n(t)\cos\phi_n + b_n\cos\varphi_n(t)\sin\phi_n\big)\end{aligned} \tag{47}$$

Remark 4. Note that the calculations of (44)-(47) depend on $\varphi_n(t)$ and $\dot{\varphi}_n(t)$. When the agent n is not at the maximal velocity, $\varphi_n(t)$ and $\dot{\varphi}_n(t)$ can be obtained by solving (27) and (28). When the agent n is at the maximal velocity, i.e. the event $\{u_n^0\}$ in TABLE I occurs, they can be obtained by solving (29).

The following proposition derives the value of $\nabla R_i(\rho_k^+)$ in (41) after the event time $t = \rho_k$. Note that the calculation of $\nabla R_i(\rho_k^+)$ only involves the first two events in TABLE I.

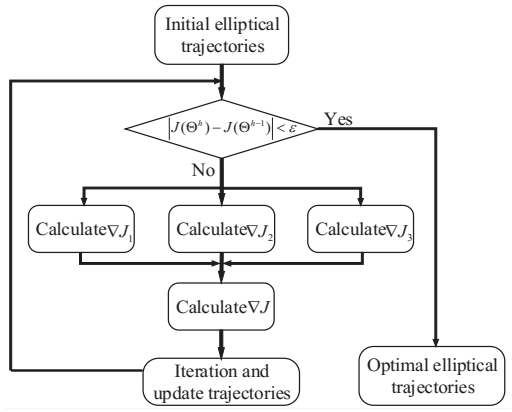


Fig. 3. Flow chart of Algorithm 1.

Proposition 1. If an event occurs at $t = \rho_k$, the state derivative $\nabla R_i(\rho_k^+)$ satisfies:

$$\nabla R_i(\rho_k^+) = \begin{cases} 0, & \text{if } e(\rho_k) = \xi_i^0 \\ \nabla R_i(\rho_k^-), & \text{if } e(\rho_k) = \xi_i^+ \end{cases} \tag{48}$$

Proof. Due to the space limitation, please see [28] for the detailed proof. ■

2.2) **Calculate** $\nabla d_{pn}^-(t)$. This involves the derivative of the min function in (7). For a given pair of agents p and n, following the definition of $d_{pn}^-(t)$ in (7),

$$\nabla d_{pn}^-(t) = \begin{cases} 0, & \text{if } t \in [\zeta_{pn}^0, \zeta_{pn}^-] \\ \nabla \|s_p(t) - s_n(t)\|, & \text{if } t \in (\zeta_{pn}^-, \zeta_{pn}^0) \end{cases} \tag{49}$$

2.3) **Calculate** $\nabla d_{ln}^-(t)$. This involves the derivative of the min function (9). For a given pair (l, n) defined by obstacle l and agent n, $\nabla d_{ln}^-(t)$ can be calculated as follows:

$$\nabla d_{ln}^-(t) = \begin{cases} 0, & \text{if } t \in [\delta_{ln}^0, \delta_{ln}^-] \\ \nabla \|\omega_l(t) - s_n(t)\|, & \text{if } t \in (\delta_{ln}^-, \delta_{ln}^0) \end{cases} \tag{50}$$

Based on the above analysis, the procedure for obtaining optimal elliptical trajectories is summarized in Algorithm 1, which is an off-line centralized algorithm. From Algorithm 1, we can get optimal parameters for every agent. Then each agent will execute the monitoring task in an open-loop manner, which has been mentioned in **Remark 1**. In addition, the flow chart of the algorithm is presented in Fig. 3.

Algorithm 1: IPA-based iteration algorithm

Input: The initial parametric trajectory $\Theta_n^1 = [X_n^1, Y_n^1, a_n^1, b_n^1, \phi_n^1]$, maximal acceleration u_n^{max}, maximal velocity v_n^{max}, sensing range r_n, velocity threshold β_n for effective monitoring, safety radius ρ_n, initial values of uncertainty for all targets $R_i(0)$, positions of obstacles, terminal condition ε, iterative index $h = 1$, and $J(\Theta^0) = 0$.

Output: Optimal parameters $[X_n^\star, Y_n^\star, a_n^\star, b_n^\star, \phi_n^\star]$.

while $|J(\Theta^h) - J(\Theta^{h-1})| > \varepsilon$ **do**
- Compute agent trajectory $s_n(t)$, $t \in [0, T]$ defined by $[X_n^h, Y_n^h, a_n^h, b_n^h, \phi_n^h]$ through (24)-(29);
- **for** $i = 1 : M$ **do**
  - Calculate $\dot{R}_i(t)$ according to (5);
  - **if** $\dot{R}_i(t)$ *switches* **then**
    - Compute $\nabla R_i(t)$ according to (48);
  - **else**
    - Compute $\nabla R_i(t)$ according to (41);
- Compute ∇J_1 according to (38);
- **for** *agent* $p \neq n$ **do**
  - Calculate $d_{pn}^-(t)$ according to (7);
  - **if** $d_{pn}^-(t) = d_{pn}(t) - \rho_p - \rho_n$ **then**
    - Calculate $\nabla d_{pn}^-(t)$ using $\nabla s_n(t)$ through (44) and (45);
  - **else**
    - $\nabla d_{pn}^-(t) = 0$;
- Compute ∇J_2 according to (39);
- **for** $l = 1 : L$ **do**
  - Calculate $d_{ln}^-(t)$ according to (9);
  - **if** $d_{ln}^-(t) = d_{ln}(t) - r_l - \rho_n$ **then**
    - Calculate $\nabla d_{ln}^-(t)$ using $\nabla s_n(t)$ through (44) and (45);
  - **else**
    - $\nabla d_{ln}^-(t) = 0$;
- Compute ∇J_3 according to (40);
- Compute the overall gradient ∇J using (37);
- Update $[X_n^h, Y_n^h, a_n^h, b_n^h, \phi_n^h]$ through (30);
- Set $h = h + 1$;

return $[X_n^\star, Y_n^\star, a_n^\star, b_n^\star, \phi_n^\star] = [X_n^h, Y_n^h, a_n^h, b_n^h, \phi_n^h]$;

V. Simulation Examples

In this section, simulations are provided to illustrate the results obtained by Algorithm 1.

In the following examples, the mission space is $\mathbf{\Omega} \equiv [0, 10] \times [0, 5] \subset \mathbb{R}^2$. Note that in our work, the event excitation problem discussed in [10] may be caused by two situations including initial agent trajectories not visiting any targets and targets with large weights being not monitored. In either case, the performance metric gradients may not be excited, resulting in zero values that prevent the iterative algorithm (30) from converging to an optimum. This problem can be overcome by introducing a "potential field" as detailed in earlier work [10]. Here, to avoid the occurrence of event excitation problem, select $\{z_i = [z_i^x, z_i^y], z_i^x = 0, 1, .., 10, z_i^y = 0, 1, .., 5\}$ as the target points, which are uniformly distributed in the mission space. The initial values of uncertainty for these points are $R_i(0) = 0, i = 1, ..., 66$. In addition, the effective sensing range for all agents is $r_n = 2$, the velocity threshold for effective monitoring is $\beta_n = 5$, the maximal acceleration is $u_n^{max} = 1$ and the maximal velocity is $v_n^{max} = 1.5$. Moreover, the safety radius ρ_n is 0.2 for any agent n. Further, for avoiding slight collisions, i.e. $M_2 J_2(t)$ and $M_3 J_3(t)$ ending up as small numbers, we set an extra safety distance as 0.02 in the simulation. M_2 and M_3 are selected as -30000.

Case A. **One agent case.** In this one agent case, we use both the elliptical trajectory and the Fourier trajectory

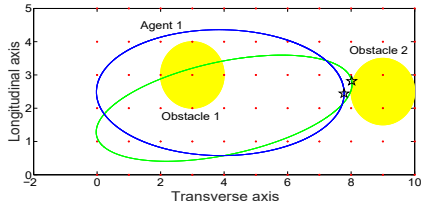
(Detailed derivation of the results can be found in [28]) to illustrate the effectiveness of Algorithm 1 respectively. Since the objective function J is non-convex, there may be many local optima depending on initial trajectories. Therefore, we apply multiple initial parameters and select the best. Then on the basis of elliptical trajectory, we further give some examples of comparison with two cases, i.e. monitoring with no obstacles and monitoring with the existing sensing model in [12], to illustrate the effectiveness of our collision avoidance method and of the new sensing model. The time horizon is set as $40s$. Let all target points have equal importance such that $\sigma_i = 1, i = 1, ..., 66$. The increasing rate is $A_i = 1, i = 1, ..., 66$ and the decreasing rate is $B = 15$.

From (24), the positions of agent n come down to the calculation of $\varphi_n(t)$. Further, since the ellipse is a curve, we take the discrete way to calculate the positions. Denote Δt as a small time interval. When the agent n is not at the maximal velocity, $\varphi_n(t)$ can be determined by repeatedly calculating $\varphi_n(m+1) = \varphi_n(m)+\dot{\varphi}_n(m)\Delta t+\frac{1}{2}\ddot{\varphi}_n(m)\Delta t^2$, where $\dot{\varphi}_n(m)$ can be obtained by solving $\dot{s}_n^x(t)^2+\dot{s}_n^y(t)^2 = v_n(t)^2$ ($v_n(t)$ is the current velocity) and $\ddot{\varphi}_n(m)$ can be obtained by solving (27). When the agent n is at the maximal velocity, i.e. the event $\{u_n^0\}$ in TABLE I occurs, $\varphi_n(m+1) = \varphi_n(m) + \dot{\varphi}_n(m)\Delta t$, where $\dot{\varphi}_n(m)$ can be obtained by solving (29). Through all these processes, we can obtain $\varphi_n(t)$, $\dot{\varphi}_n(t)$ and $s_n(t)$ for all t. Therefore, the derivatives from (44) to (47) can be calculated.

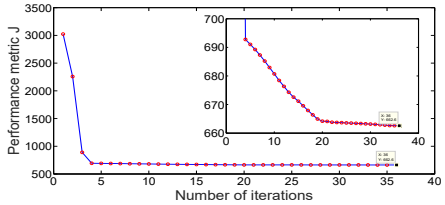
Example 1. **Illustration example of Algorithm 1.**

In Fig. 4, there is a persistent monitoring task with obstacles executed by one agent moving on an elliptical trajectory and we show the best from multiple elliptical results. In Fig. 4(a), the mission space is given where there exist obstacles covered by yellow circular areas whose centers are $[3, 3]$, $[9, 2.5]$, respectively and radii are both 1. Agent 1 moves counterclockwise on the trajectory with the pentagram as the starting point (subsequent results are the same). Fig. 4(b) indicates that the performance metric decreases as the number of iterations increases and ultimately converges, which verifies the effectiveness of our IPA-based iteration algorithm. As we can see, there is a very abrupt drop in the first several iterations and after it the performance metric decreases with a relatively slow rate, which indicates that the initial trajectory with obstacle collisions has a very large cost and Agent 1 will continue to search for a better monitoring trajectory after avoiding obstacles. The final performance metric $J(\Theta^{36}) = 662.6$ and $|J(\Theta^{36}) - J(\Theta^{35})| < \varepsilon = 0.01$. According to Fig. 4(c), we can intuitively observe that distances between Agent 1 and obstacles are greater than 1.2 all the time, which indicates that there are no collisions in the monitoring task. Actually, all simulaion results ultimately have no collisions, which shows the correctness of our objective function setting. Furthermore, detailed velocity components of this work and of [12] are shown in Fig. 4(d) and Fig. 4(e), respectively. Compared with Fig. 4(e) where the agent starts with a certain velocity, the result in Fig. 4(d) can better mimic agent behavior in practice.

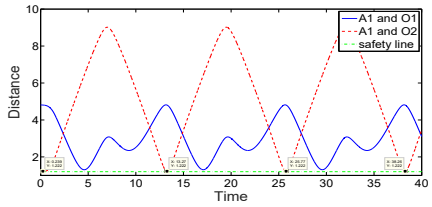
With the same setting as the elliptical example, simulation results of the Fourier trajectory example with one agent can be seen in Appendix A. Furthermore, the final numerical results of the two trajectories are presented in TABLE II, and the final costs of collisions (agent to agent, agent to obstacle), i.e. $J_2^\star$ and $J_3^\star$, are zero, which indicates that we achieve collision-free monitoring.



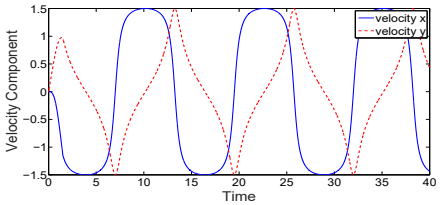
(a) Green ellipse: initial trajectory. Blue ellipse: final trajectory. Yellow circular areas: areas covering obstacles. Red $*$: target points. Black pentagrams: starting points.



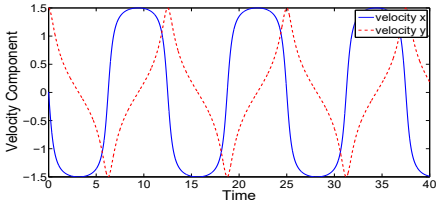
(b) The evolution of performance metric J.



(c) Distances between Agent 1 and Obstacles.



(d) Velocity component of this work.



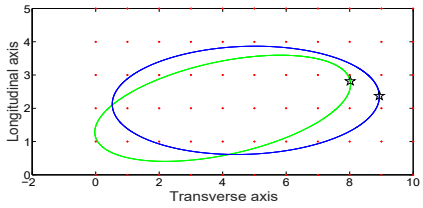
(e) Velocity component of [12].

Fig. 4. Elliptical trajectory: persistent monitoring task with obstacles using the IPA-based iteration algorithm for one agent.

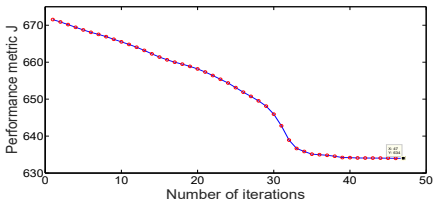
TABLE II
RESULT COMPARISON FOR CASE A

$Method$	$J^\star$	$J_2^\star$	$J_3^\star$
$Elliptical$	662.6	0	0
$Fourier$	654	0	0

Example 2. **Comparison with the case without obstacles.**



(a) Green ellipse: initial trajectory. Blue ellipse: final trajectory. Red $*$: target points. Black pentagrams: starting points.



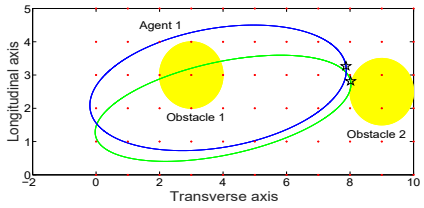
(b) The evolution of performance metric J.

Fig. 5. Elliptical trajectory: persistent monitoring task without obstacles using the IPA-based iteration algorithm for one agent.

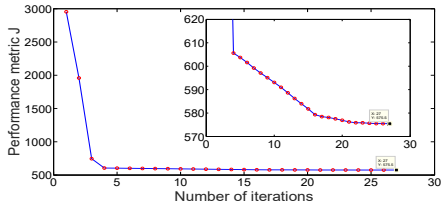
In this example, we set an idealized mission space like [12] for agents, namely, without obstacles. With the same initial conditions as in Fig. 4 of ***Example 1*** except that there are no obstacles, the results are shown in Fig. 5. Fig. 5(a) presents the optimal elliptical trajectory obtained by Algorithm 1. Due to the potential slight collisions, it is unreasonable to use the final trajectory in Fig. 5(a) to execute the persistent monitoring task with obstacles. From Fig. 5(b), the performance metric converges to $J(\Theta^{47}) = 634$ and $|J(\Theta^{47}) - J(\Theta^{46})| < \varepsilon = 0.01$. Compared with the performance metric in Fig. 4(b), our method can avoid collisions at the expense of some monitoring, which is quite practical in monitoring tasks where collisions with obstacles are strictly prohibited.

Example 3. **Comparison with the existing sensing model.**

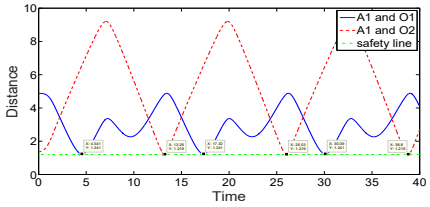
In this example, we use the existing sensing model in [12], which depends only on the relative distance between an agent and a target. Intuitively, we expect a better result than ***Example 1***, owing to the negative effect of moving agents in order to monitor targets. With the same initial conditions as in Fig. 4 of ***Example 1***, the results using the sensing model in [12] are shown in Fig. 6. Note that the final ellipse $[4.7291, 2.2391, 4.2087, 1.6230, 0.0321]$ in Fig. 6(a) is different from the final ellipse $[3.8791, 2.4675, 3.8994, 1.8926, -0.0066]$ in Fig. 4(a), which is due to the influence of the new sensing model on the final optimal trajectory. In addition, the value of the performance metric in Fig. 6(b) is less than that in Fig. 4(b). In Fig. 4(b), the new sensing model is used which includes the effect of velocity on the quality of sensing. In Fig. 6(b) the original sensing model from [12] is used. This ignores agent movements, therefore, it gives a more "optimistic" result in the sense that the average uncertainty metric is lower than it ought to be with velocity taken into account. Actually, since the influence of the agent's velocity on sensing strength cannot be ignored, such result may be not achievable in practice. In addition, Fig. 6(c) can still verify that collisions



(a) Green ellipse: initial trajectory. Blue ellipse: final trajectory. Yellow circular areas: areas covering obstacles. Red $*$: target points. Black pentagrams: starting points.



(b) The evolution of performance metric J.



(c) Distances between Agent 1 and Obstacles.

Fig. 6. Elliptical trajectory: persistent monitoring task with obstacles using the IPA-based iteration algorithm for one agent (using the the sensing model in [12]).

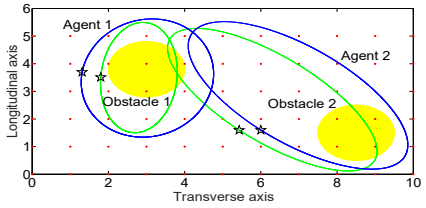
are avoided through our method. Furthermore, we use the final ellipse in Fig. 6(a) to execute the monitoring task using the sensing model in this work, and the performance metric is $J = 662.6$, which is only slightly different from the result in Fig. 4(b).

Case B. **Two agents case.**

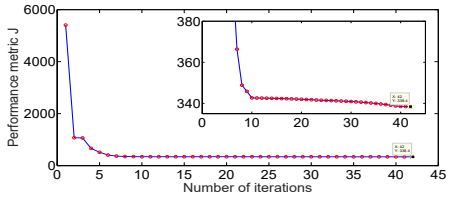
In this case, we carry out multiple simulation experiments and show the best results of two agents using the elliptical trajectory and the Fourier series trajectory to execute the monitoring task. The time horizon is set as $30s$. Suppose there are four targets $[5, 1], [5, 2], [5, 3], [5, 4]$ with larger weights $\sigma_{32} = \sigma_{33} = \sigma_{34} = \sigma_{35} = 2$ such that they require more attention than others. Moreover, obstacles in the mission space are covered by yellow circular areas whose centers are $[3, 3.8]$, $[8.5, 1.5]$, respectively and radii are both 1.

Fig. 7(a) shows the optimal elliptical trajectories obtained by Algorithm 1. In Fig. 7(b), the performance metric decreases as the number of iterations with the final value $J(\Theta^{42}) = 338.4$ and $|J(\Theta^{42}) - J(\Theta^{41})| < \varepsilon = 0.01$. Moreover, we show distances between agents and obstacles in Fig. 7(c). It is obvious that distances between agents and obstacles are always greater than 1.2. Fig. 7(d) shows the distance between Agent 1 and Agent 2, which is always greater than 0.4. In other words, there are no potential collisions (agent to agent, agent to obstacle) in the final optimal trajectories.

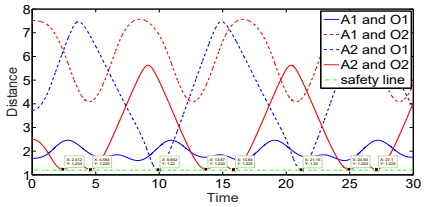
Moreover, with the same setting as the elliptical example, simulation results of the Fourier trajectory example with two



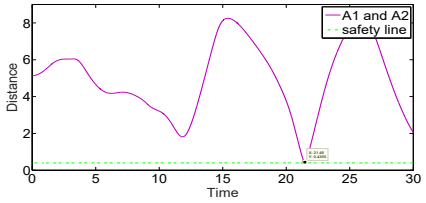
(a) Green ellipses: initial trajectories. Blue ellipses: final trajectories. Yellow circular areas: areas covering obstacles. Red $*$: target points. Black pentagrams: starting points.



(b) The evolution of performance metric J.



(c) Distances between Agents and Obstacles.



(d) Distance between Agent 1 and Agent 2.

Fig. 7. Elliptical trajectory: persistent monitoring task with obstacles using the IPA-based iteration algorithm for two agents.

agents can be seen in Appendix A. And the final numerical results of the two trajectories are presented in TABLE III, and the final costs of collisions (agent to agent, agent to obstacle), i.e. $J_2^\star$ and $J_3^\star$, are zero, which indicates that we achieve collision-free monitoring.

TABLE III
RESULT COMPARISON FOR CASE B

Method	$J^\star$	$J_2^\star$	$J_3^\star$
Elliptical	338.4	0	0
Fourier	305.9	0	0

VI. CONCLUSION

We have studied a 2D persistent monitoring problem using second-order agents. We have proposed a new and more practical sensing model which takes the effects of both distances between agents and targets and agents' velocities into account. In particular, we have considered no-collision constraints (both agent to agent and agent to obstacle). Through parameterizing agent trajectories and utilizing IPA-based gradient algorithm, our method provides collision-free optimal (locally) trajectories.

In addition, decentralizing the controller is particularly challenging in persistent monitoring. We have been working on this problem and report results in [7]. And we show that it is possible to achieve an "almost decentralized" solution based exclusively on using local information. We also quantify the "price of decentralization" in the sense of losing some performance. Thus, imposing a decentralized solution to the persistent monitoring problem is of course possible but not without some (quantifiable) performance degradation. Future research will focus on escaping the local optimal solution, developing decentralized solutions, developing nonconservative ways of dealing with obstacles and exploiting on-line strategies of dealing with unexpected events for the 2D persistent monitoring problems with potential collisions.

APPENDIX A
SIMULATIONS OF FOURIER TRAJECTORIES

With the same setting as the elliptical trajectory, the simulation results of the Fourier trajectory are shown in Fig. 8 and Fig. 9, respectively.

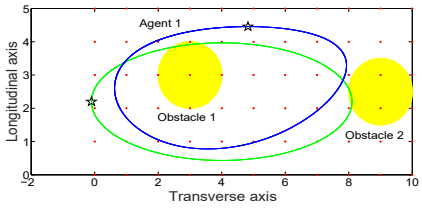
In Fig. 8, there is a persistent monitoring task with obstacles executed by one agent moving on a Fourier series trajectory and we select the best from multiple Fourier results. Because of the complexity and variability of this trajectory, we used $\Gamma_1^x = \Gamma_1^y = 2$ for simplicity. From Fig. 8(a), there is an interesting difference between the initial and final trajectory, which reflects the unpredictability of this trajectory. In Fig. 8(b), the performance metric decreases as the iteration progresses and ultimately converges, thus the effectiveness of Algorithm 1 applied to a Fourier series trajectory is verified. The final performance metric $J(\Theta^{31}) = 654$ and $|J(\Theta^{31}) - J(\Theta^{30})| < \varepsilon = 0.01$. From the distances between Agent 1 and obstacles in Fig. 8(c), we can observe that Agent 1 safely completed the monitoring task.

In Fig. 9, there is a persistent monitoring task with obstacles executed by two agents moving on Fourier series trajectories. We set $\Gamma_1^x = \Gamma_1^y = \Gamma_2^x = \Gamma_2^y = 2$ for simplicity. Fig. 9(a) shows the interesting changes between the initial and final trajectories, which is due to agents tending to avoid obstacles. In Fig. 9(b), the performance metric decreases as the iteration progresses with $J(\Theta^{21}) = 305.9$ and $|J(\Theta^{21}) - J(\Theta^{20})| < \varepsilon = 0.01$. From Fig. 9(c), we can observe that the two agents both can safely execute the monitoring task. And in Fig. 9(d), the distance between Agent 1 and Agent 2 is always greater than 0.4, i.e. this is a safe monitoring task.

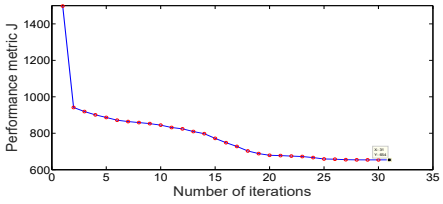
REFERENCES

[1] D. A. Paley, F. Zhang, and N. E. Leonard, "Cooperative control for ocean sampling: The glider coordinated control system," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 735–744, 2008.

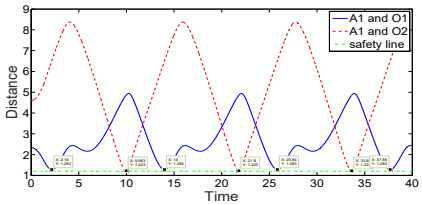
[2] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.



(a) Green Fourier trajectory: initial trajectory. Blue Fourier trajectory: final trajectory. Yellow circular areas: areas covering obstacles. Red $*$: target points. Black pentagrams: starting points.

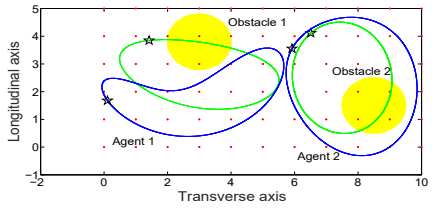


(b) The evolution of performance metric J.

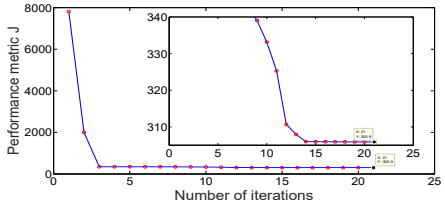


(c) Distances between Agent 1 and Obstacles.

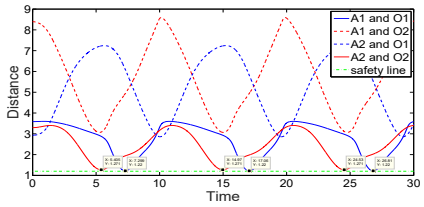
Fig. 8. Fourier series trajectory: persistent monitoring task with obstacles using the IPA-based iteration algorithm for one agent.



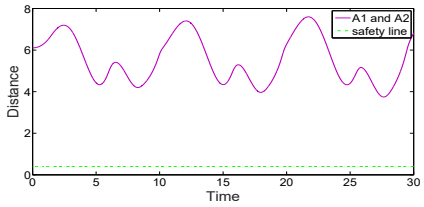
(a) Green Fourier trajectories: initial trajectories. Blue Fourier trajectories: final trajectories. Yellow circular areas: areas covering obstacles. Red $*$: target points. Black pentagrams: starting points.



(b) The evolution of performance metric J.



(c) Distances between Agents and Obstacles.



(d) Distance between Agent 1 and Agent 2.

Fig. 9. Fourier series trajectory: persistent monitoring task with obstacles using the IPA-based iteration algorithm for two agents.

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

[4] L. C. B. da Silva, R. M. Bernardo, H. A. de Oliveira, and P. F. F. Rosa, "Unmanned aircraft system coordination for persistent surveillance with different priorities," in *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on*, pp. 1153–1158, IEEE, 2017.

[5] A. Puri, "A survey of unmanned aerial vehicles (uav) for traffic surveillance," *Department of computer science and engineering, University of South Florida*, pp. 1–29, 2005.

[6] J. L. Vian, A. R. Mansouri, and E. W. Saad, "Traffic and security monitoring system and method," Feb. 4 2014. US Patent 8,643,719.

[7] N. Zhou, C. G. Cassandras, X. Yu, and S. B. Andersson, "Decentralized event-driven algorithms for multi-agent persistent monitoring tasks," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 4064–4069, IEEE, 2017.

[8] P. Tokekar and V. Kumar, "Visibility-based persistent monitoring with robot teams," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3387–3394, IEEE, 2015.

[9] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106–1118, 2016.

[10] N. Zhou, X. Yu, S. B. Andersson, and C. G. Cassandras, "Optimal event-driven multi-agent persistent monitoring of a finite set of data sources," *IEEE Transactions on Automatic Control*, 2018.

[11] Y.-W. Wang, Y.-W. Wei, X.-K. Liu, N. Zhou, and C. G. Cassandras, "Optimal persistent monitoring using second-order agents with physical constraints," *IEEE Transactions on Automatic Control*, 2018.

[12] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1659–1664, 2015.

[13] C. Song, L. Liu, G. Feng, and S. Xu, "Optimal control for multi-agent persistent monitoring," *Automatica*, vol. 50, no. 6, pp. 1663–1668, 2014.

[14] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.

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

[16] J. Yu, S. Karaman, and D. Rus, "Persistent monitoring of events with stochastic arrivals at multiple stations," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 521–535, 2015.

[17] D. E. Soltero, S. L. Smith, and D. Rus, "Collision avoidance for persistent monitoring in multi-robot systems with intersecting trajectories," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3645–3652, IEEE, 2011.

[18] C. Song, L. Liu, G. Feng, Y. Wang, and Q. Gao, "Persistent awareness coverage control for mobile sensor networks," *Automatica*, vol. 49, no. 6, pp. 1867–1873, 2013.

[19] C. Franco, D. M. Stipanović, G. López-Nicolás, C. Sagüés, and S. Llorente, "Persistent coverage control for a team of agents with collision avoidance," *European Journal of Control*, vol. 22, pp. 30–45, 2015.

[20] J. M. Palacios-Gasós, Z. Talebpour, E. Montijano, C. Sagüés, and A. Martinoli, "Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1321–1327, IEEE, 2017.

[21] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[22] S. Adlakha and M. Srivastava, "Critical density thresholds for coverage in wireless sensor networks," in *Wireless communications and network-*

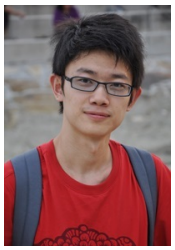
ing, 2003. WCNC 2003. 2003 IEEE, vol. 3, pp. 1615–1620, IEEE, 2003.
[23] C. Zang, W. Liang, and H. Yu, "The critical speeds and radii for coverage in sensor networks," in *Intelligent Control and Automation*, pp. 106–117, Springer, 2006.
[24] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato, "Distributed perimeter patrolling and tracking for camera networks," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 2093–2098, IEEE, 2010.
[25] R. Carli, A. Cenedese, and L. Schenato, "Distributed partitioning strategies for perimeter patrolling," in *American Control Conference (ACC), 2011*, pp. 4026–4031, IEEE, 2011.
[26] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*, vol. 1. Springer Science & Business Media, 2006.
[27] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[28] Y.-W. Wang, M.-J. Zhao, W. Yang, N. Zhou, and C. G. Cassandras, "Collision-free trajectory design for 2d persistent monitoring using second-order agents," *arXiv: 1911.04297*, 2019.

Yan-Wu Wang (M10–SM13)received the B.S. degree in automatic control, the M.S. degree and the Ph.D. degree in control theory and control engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1997, 2000, and 2003, respectively. Currently, she is a Professor with the School of Artificial Intelligence and Automation, HUST. Her research interests include hybrid systems, cooperative control, and multi-agent systems, with their 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 PhD 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.

Ming-Jie Zhao received the B.S. degree in 2017 from the College of Electrical and Information Engineering, Hunan University (HNU), Changsha, China. Currently, he is pursuing the M.S. degree from the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology (HUST), Wuhan, China. His research interests include optimization and persistent monitoring.

Wu Yang received the B.S. degree in Information and Computing Science and the M.S. degree in Applied Mathematics from Guangxi University, Nanning, China, in 2011 and 2014, respectively, and the Ph.D. degree in Control Science and Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2017, where he is currently post-doctoral fellow at School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His current research interests include singularly perturbed systems, hybrid systems, complex systems and networks, and so on.



Nan Zhou received the B.S. degree in control engineering from Zhejiang University, Hangzhou, China, in 2011, the M.S. degree in control engineering from University of Chinese Academy of Sciences, Beijing, China, in 2014, and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2019. His research interests include control and optimization of multi-agent systems with applications to robotics and smart transportation. He is currently working at MathWorks Inc. and is a student member of the IEEE Control Systems Society.

Christos G. Cassandras (F96) received the B.S. degree from Yale University, New Haven, CT, USA, in 1977, the M.S.E.E. degree from Stanford University, Stanford, CA, USA, in 1978, and the M.S. and Ph.D. degrees from Harvard University, Cambridge, MA, USA, in 1979 and 1982, respectively. He was with ITP Boston, Inc., Cambridge, from 1982 to 1984, where he was involved 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 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 over 400 refereed papers in these areas, and six books.

Dr. Cassandras is 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.