

Event-Driven Trajectory Optimization for Data Harvesting in Multiagent Systems

Yasaman Khazaeni , *Member, IEEE*, and Christos G. Cassandras , *Fellow, IEEE*

Abstract—We propose a new event-driven method for online trajectory optimization to solve the data harvesting problem: in a 2-D mission space, N mobile agents are tasked with the collection of data generated at M stationary sources and delivery to a base with the goal of minimizing expected collection and delivery delays. We define a new performance measure that addresses the event excitation problem in event-driven controllers and formulate an optimal control problem. The solution of this problem provides some insight on its structure, but it is computationally intractable, especially in the case where the data generating processes are stochastic. We propose an agent trajectory parameterization in terms of general function families, which can be subsequently optimized online through the use of infinitesimal perturbation analysis. Properties of the solutions are identified, including robustness with respect to the stochastic data generation process and scalability in the size of the event set characterizing the underlying hybrid dynamical system. Explicit results are provided for the case of elliptical and Fourier series trajectories, and comparisons with a state-of-the-art graph-based algorithm are given.

Index Terms—Data harvesting, event-driven optimization, infinitesimal perturbation analysis, multiagent systems, optimal control, trajectory planning.

I. INTRODUCTION

SYSTEMS consisting of cooperating mobile agents have been extensively studied and used in a broad spectrum of applications such as environmental sampling [1], [2], surveillance [3], coverage [4]–[6], persistent monitoring [7], [8], task assignment [9], and data harvesting and information collection [10]–[12].

The *data harvesting* problem in particular (and its variant, the “minimum latency” problem [13]) arises in many settings, where wireless-sensor networks (WSNs) are deployed for purposes of monitoring the environment, road traffic, infrastructure for transportation and for energy distribution, surveillance, and a variety of other specialized purposes [14], [15]. Although many efforts focus on the analysis of the vast amount of data gathered,

Manuscript received January 15, 2017; revised April 9, 2017; accepted May 11, 2017. Date of publication May 24, 2017; date of current version September 17, 2018. This work was supported in part by the National Science Foundation under Grants CNS-1239021, ECCS-1509084, and IIP-1430145, in part by the Air Force Office of Scientific Research under Grant FA9550-12-1-0113, and in part by the Office of Naval Research under Grant N00014-09-1-1051. Recommended by Associate Editor L. Shi. (*Corresponding author: Yasaman Khazaeni.*)

Y. Khazaeni was with the Division of Systems Engineering and the Center for Information and Systems Engineering, Boston University, Boston, MA 02215 USA. She is now with IBM Research, Cambridge, MA 02141 USA (e-mail: yasaman.khazaeni@us.ibm.com).

C. G. Cassandras is with the , Division of Systems Engineering Boston University, Boston, MA 02215 USA (e-mail: cgc@bu.edu).

Digital Object Identifier 10.1109/TCNS.2017.2707920

we must first ensure the existence of robust means to collect all data in a timely fashion when the size of the network and the level of node interference do not allow for a fully connected wireless system. In such cases, sensors can locally gather and buffer data, while mobile elements (e.g., vehicles and aerial drones) retrieve the data from each part of the network. Similarly, mobile elements may themselves be equipped with sensors and visit specific points of interest, called “targets,” where a direct communication path does not exist between them and the central sink or base node where the data must be delivered. In a delay-tolerant system, the control scheme commonly used is to deploy mobile agents referred to as “data mules,” “message ferries,” or simply “ferries” [16]–[19]. The mobile agents visit the data generation nodes and collect data, which are then delivered to the base. Moreover, since agents generally have limited buffer sizes, visits to the base are also needed once a buffer is full; the same is true due to limited energy that requires them to be periodically recharged. In general, the paths followed by the mobile agents need to be optimized (in some sense to be defined) so as to ensure timely delivery of data through sufficiently frequent visits at each data source and the base and within the constraints of a given environment (e.g., an urban setting).

Interestingly, there are analogs to the data harvesting problem outside the sensor network realm. For instance, in disaster planning, evacuation and rescue operations, pickup/delivery and transportation systems, and unmanned aerial vehicle surveillance operations, the general theme involves a network of cooperating mobile agents that need to frequently visit points of interest and transfer data/goods/people to a base. Base visits may also be needed to recharge/renew power supply/fuel. For example, the flying time span of a drone on a single battery charge is limited, so that flight trajectories need to be optimized and returns to base may be scheduled for recharge or loading/unloading. Thus, we may view data harvesting in the broader context of a multiagent system, where mobile agents must cooperatively design trajectories to visit a set of targets and a base so as to optimize one or more performance criteria.

Having its root in WSNs, the data harvesting problem is normally studied on a directed or undirected graph, where minimum length tours or subtours are to be found. This graph topology view of the problem utilizes a multitude of routing and scheduling algorithms developed for WSNs (e.g., [20]–[22] and references therein). One of its main advantages is the ability to accommodate environmental constraints (e.g., obstacles) by properly selecting graph edges; thus, movements inside a building or within a road network are examples where a graph topology

is suitable. On the other hand, these methods also have several drawbacks: they are generally combinatorially complex; they do not account for limitations in motion dynamics, which should not, for instance, allow an agent to form a trajectory consisting of a sequence of straight lines; they become computationally infeasible as online methods in the presence of stochastic effects such as random target rewards or failing agents, since the graph topology has to be re-evaluated as new information becomes available.

As an alternative to the graph-based approach, we view data harvesting as a trajectory optimization problem in a 2-D space, where the mobile agents are freely (or with some specified constraints) moving and “visit” targets whenever they reach their vicinity. For example, each target can be assumed to have a finite range within which a mobile agent can initiate wireless communication with it and exchange data. These trajectories do not necessarily consist of straight lines, that is, edges in an underlying graph topology; therefore, an advantage of this approach is that an agent trajectory can be designed to conform to physical limitations in the agent’s mobility. In addition, trajectories can be adjusted online when target locations are uncertain. Constraining such trajectories to obstacles is also still possible. Such continuous topologies have been used in [11], where the problem is viewed as a polling system with a mobile server visiting data queues at fixed targets and trajectories are designed to stabilize the system, keeping queue contents (modeled as fluid queues) uniformly bounded, and in [23], where parameterized trajectories are optimized to solve a multiagent persistent monitoring problem.

A key benefit of a trajectory optimization view of the data harvesting problem is the ability to parameterize the trajectories with different types of functional representations and then optimize them over the given parameter space. This reduces a dynamic optimization problem into a much simpler parametric optimization one. If the parametric trajectory family is broad enough, we can recover the true optimal trajectories; otherwise, we can approximate them within a desired accuracy. Moreover, adopting a parametric family of trajectories has several additional benefits. First, it allows trajectories to be periodic, often a desirable property in practice. Second, it allows one to restrict solutions to trajectories with desired features that the true optimal cannot have, for example, smoothness properties required for physically realizable agent motion.

In this paper, we cast data harvesting as an optimal control problem. Defining an appropriate optimization criterion is nontrivial in this problem (as we will explain) and introducing appropriate performance metrics is the first contribution of this work. Obtaining optimal agent trajectories ultimately requires the solution of a two-point boundary value problem (TPBVP). Although a complete solution of such a TPBVP is computationally infeasible in general, we identify structural properties of the optimal control policy, which allow us to reduce the agent-target/base interaction process to a hybrid system with a well-defined set of events that cause discrete state transitions. The second contribution is to formulate and solve an optimal parametric agent trajectory problem. In particular, similar to the idea introduced in [23], we represent an agent trajectory in

terms of general function families characterized by a set of parameters that we seek to optimize, given an objective function. We consider elliptical trajectories as well as the much richer set of Fourier series trajectory representations. We then show that we can make use of infinitesimal perturbation analysis (IPA) for hybrid systems [24] to determine gradients of the objective function with respect to these parameters and subsequently obtain (at least locally) optimal trajectories. This approach also allows us to exploit 1) robustness properties of IPA to allow stochastic data generation processes, 2) the *event-driven* nature of the IPA gradient estimation process, which is *scalable* in the event set of the underlying hybrid dynamic system, and 3) the *online* computation, which implies that trajectories adjust as operating conditions change (e.g., new targets); in contrast, the solution of a TPBVP is computationally challenging even for strictly *offline* methods. Finally, we provide comparisons of our approach to algorithms based on a graph topology of the mission space. These comparisons show that while the latter generate a spatial partitioning of the target set among agents, our approach results in a temporal partitioning, which adds robustness with respect to agent failures or other environmental changes. The graph-based approaches are mostly offline and normally assume that the agents would be able to travel straight lines and meet targets in exact locations. On the other hand, the trajectory optimization approach allows us to accommodate limitations in agent mobility and to adjust trajectories online.

In Section II, we formulate the data harvesting problem using a queueing model and present the underlying hybrid system. In Section III, we provide a Hamiltonian analysis leading to a TPBVP. In Section III, we formulate the alternative problem of determining optimal trajectories based on general function representations and provide solutions through a gradient-based algorithm using IPA for two particular function families. Section IV presents numerical results and comparisons with state-of-the-art data harvesting algorithms, and Section VI contains conclusions.

II. PROBLEM FORMULATION

We consider a data harvesting problem, where N mobile agents collect data from M stationary targets in a 2-D mission space S . Each agent may visit one or more of the M targets, collect data from them, and deliver them to a base. It then continues visiting targets, possibly the same as before or new ones, and repeats this process. The objective of the agent team is to minimize data collection and delivery delays over all targets within a fixed time interval T . This minimization problem is formalized in the following.

The data harvesting problem described above can be viewed as a polling system, where mobile agents are serving the targets by collecting data and delivering it to the base. As seen in Fig. 1, there are three sets of queues. The first set includes the data contents $X_i(t) \in \mathbb{R}^+$ at each target $i = 1, \dots, M$, where we use $\sigma_i(t)$ as the instantaneous inflow rate. In general, we treat $\{\sigma_i(t)\}$ as a random process assumed only to be piecewise continuous; we will treat it as a deterministic constant only for the Hamiltonian analysis in the next section. Thus, at time t ,

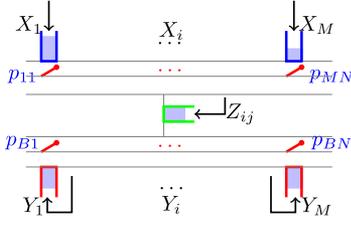


Fig. 1. Data harvesting queueing model for M targets and N agents.

$X_i(t)$ is a random variable resulting from the random process $\{\sigma_i(t)\}$.

The second set of queues consists of data contents $Z_{ij}(t) \in \mathbb{R}^+$ on-board agent j collected from targets $i = 1, \dots, M$. The last set consists of queues $Y_i(t) \in \mathbb{R}^+$ containing data at the base, one queue for each target, delivered by some agent j . Note that $\{Z_{ij}(t)\}$ and $\{Y_i(t)\}$ are also random processes.

In Fig. 1, collection and delivery switches are shown by p_{ij} and p_{Bj} (formally defined in the following). These switches are ‘‘ON’’ when agent j is connected to target i or the base respectively. All queues are modeled as flow systems whose dynamics are given next (however, as we will see, the agent trajectory optimization is driven by events observed in the underlying system where queues contain discrete data packets so that this modeling device has minimal effect on our analysis).

Let $s_j(t) = [s_j^x(t), s_j^y(t)] \in S$ be the position of agent j at time t . Then, the state of the system can be defined as

$$\begin{aligned} \mathbf{X}(t) = & [X_1(t), \dots, X_M(t), Y_1(t), \dots, Y_M(t), \\ & Z_{11}(t), \dots, Z_{MN}(t), s_1^x(t), s_1^y(t), \dots, s_N^x(t), s_N^y(t)]. \end{aligned} \quad (1)$$

The position of the agent follows single-integrator dynamics at all times:

$$\begin{aligned} \dot{s}_j^x(t) &= u_j(t) \cos \theta_j(t), & \dot{s}_j^y(t) &= u_j(t) \sin \theta_j(t) \\ s_j^x(0) &= X_B & s_j^y(0) &= Y_B, \quad \forall j \end{aligned} \quad (2)$$

where $u_j(t)$ is the scalar speed of the agent (normalized so that $0 \leq u_j(t) \leq 1$), $0 \leq \theta_j(t) < 2\pi$ is the angle relative to the positive direction and $[X_B, Y_B]$ is the location of the base. Thus, we assume that the agent controls its orientation and speed. The agent states $\{s_j(t)\}$, $j = 1, \dots, N$, are also random processes since the controls are generally dependent on the random queue states. Thus, we ensure that all random processes are defined on a common probability space.

An agent is represented as a particle, so that we will omit the need for any collision avoidance control. The agent dynamics above could be more complicated without affecting the essence of our analysis, but we will limit ourselves here to (2).

We consider a set of data sources as points $w_i \in S$, $i = 1, \dots, M$, with associated ranges r_{ij} , so that agent j can collect data from w_i only if the Euclidean distance $d_{ij}(t) = \|w_i - s_j(t)\|$ satisfies $d_{ij}(t) \leq r_{ij}$. Similarly, the base is at $w_B = [X_B, Y_B] \in S$, which receives all data collected by the agents and an agent can only deliver data to the base if the Euclidean distance $d_{Bj}(t) = \|w_B - s_j(t)\|$ satisfies $d_{Bj}(t) \leq r_{Bj}$. Using $p : S \times S \rightarrow [0, 1]$, we define a function $p_{ij}(t)$ represent-

ing the collection switches in Fig. 1 as

$$p_{ij}(t) = p(w_i, s_j(t)). \quad (3)$$

$p_{ij}(t)$ is viewed as the normalized data collection rate from target i when the agent is at $s_j(t)$ and we assume that: **(A1)** it is monotonically nonincreasing in the value of $d_{ij}(t) = \|w_i - s_j(t)\|$, and **(A2)** it satisfies $p_{ij}(t) = 0$ if $d_{ij}(t) > r_{ij}$. Thus, $p_{ij}(t)$ can model communication power constraints, which depend on the distance between a data source and an agent equipped with a receiver (similar to the model used in [11]) or sensing range constraints if an agent collects data using on-board sensors. For simplicity, we will also assume that **(A3)** $p_{ij}(t)$ is continuous in $d_{ij}(t)$. Similarly, we define

$$p_{Bj}(t) = p(w_B, s_j(t)) \quad (4)$$

The maximum rate of data collection from target i by agent j is μ_{ij} , so that the instantaneous rate is $\mu_{ij}p_{ij}(t)$ if j is connected to i . We will assume that **(A4)** only one agent at a time is connected to a target i even if there are other agents l with $p_{il}(t) > 0$; this is not the only possible model, but we adopt it based on the premise that simultaneous downloading of packets from a common source creates problems of proper data reconstruction at the base.

We can now define the dynamics of the queue-related components of the state vector in (1). The dynamics of $X_i(t)$, assuming that agent j is connected to it, are

$$\dot{X}_i(t) = \begin{cases} 0, & \text{if } X_i(t) = 0 \text{ and } \sigma_i(t) \leq \mu_{ij}p_{ij}(t) \\ \sigma_i(t) - \mu_{ij}p_{ij}(t), & \text{otherwise.} \end{cases} \quad (5)$$

Obviously, $\dot{X}_i(t) = \sigma_i(t)$ if $p_{ij}(t) = 0$, $j = 1, \dots, N$.

In order to express the dynamics of $Z_{ij}(t)$, let

$$\tilde{\mu}_{ij}(t) = \begin{cases} \min\left(\frac{\sigma_i(t)}{p_{ij}(t)}, \mu_{ij}\right), & \text{if } X_i(t) = 0 \text{ and } p_{ij}(t) > 0 \\ \mu_{ij}, & \text{otherwise.} \end{cases} \quad (6)$$

This gives us the dynamics

$$\begin{aligned} \dot{Z}_{ij}(t) &= \\ &= \begin{cases} 0, & \text{if } Z_{ij}(t) = 0 \text{ and } \tilde{\mu}_{ij}(t)p_{ij}(t) - \beta_{ij}p_{Bj}(t) \leq 0 \\ \tilde{\mu}_{ij}(t)p_{ij}(t) - \beta_{ij}p_{Bj}(t), & \text{otherwise} \end{cases} \end{aligned} \quad (7)$$

where β_{ij} is the maximum rate of data from target i delivered to B by agent j . For simplicity, we assume that **(A5)** $\|w_i - w_B\| > r_{ij} + r_{Bj}$ for all $i = 1, \dots, M$ and $j = 1, \dots, N$, that is, the agent cannot collect and deliver data at the same time. Therefore, in (7), it is always the case that for all i and j , $p_{ij}(t)p_{Bj}(t) = 0$. Finally, the dynamics of $Y_i(t)$ depend on $Z_{ij}(t)$, the content of the on-board queue of each agent j from target i as long as $p_{Bj}(t) > 0$. We define $\beta_i(t) = \sum_{j=1}^N \beta_{ij}p_{Bj}(t)\mathbf{1}[Z_{ij}(t) > 0]$, where $\mathbf{1}[\cdot]$ is the indicator function, as the total instantaneous delivery rate for target i data, so that the dynamics of $Y_i(t)$ are

$$\dot{Y}_i(t) = \beta_i(t). \quad (8)$$

Hybrid system model: Taking into account the state vector in (1) and the dynamics in (2), (5), (7), and (8), the data harvesting

TABLE I
HYBRID SYSTEM EVENTS

Event Name	Description
1. ξ_i^0	$X_i(t)$ hits 0, for $i = 1, \dots, M$
2. ξ_i^+	$X_i(t)$ leaves 0, for $i = 1, \dots, M$.
3. ζ_{ij}^0	$Z_{ij}(t)$ hits 0, for $i = 1, \dots, M, j = 1, \dots, N$
4. δ_{ij}^+	$d_{ij}^+(t)$ leaves 0, for $i = 1, \dots, M, j = 1, \dots, N$
5. δ_{ij}^0	$d_{ij}^+(t)$ hits 0, for $i = 1, \dots, M, j = 1, \dots, N$
6. Δ_j^+	$d_{Bj}^+(t)$ leaves 0, for $j = 1, \dots, N$
7. Δ_j^0	$d_{Bj}^+(t)$ hits 0, for $j = 1, \dots, N$

process is a stochastic hybrid system. Discrete modes of the system are defined by intervals, over which we have the following.

- 1) An agent is visiting a target and the target queue is not empty, here named state C_1 .
- 2) An agent is visiting a target and the target queue is empty, here named state C_2 .
- 3) An agent is visiting the base, here named state D .
- 4) An agent is moving when not connected to any target or base, here named state F .

The events that trigger mode transitions are defined in Table I (the superscript 0 denotes events causing a variable to reach a value of zero from above and the superscript + denotes events causing a variable to become strictly positive from a zero value). We also use the following definitions:

$$d_{ij}^+(t) = \max(0, d_{ij}(t) - r_{ij}), \quad d_{Bj}^+(t) = \max(0, d_{Bj}(t) - r_{Bj}). \quad (9)$$

The variables above are zero if agent j is within range of target i or the base, respectively.

Observe that each of the events in Table I causes a change in at least one of the state variables in (5), (7), and (8). For example, ξ_i^0 (i.e., the queue at target i is emptied) causes a switch in (5) from $\dot{X}_i(t) = \sigma_i(t) - \mu_{ij}p_{ij}(t)$ to $\dot{X}_i(t) = 0$. Also note that we have omitted an event ζ_{ij}^+ for $Z_{ij}(t)$ becoming strictly positive since this event is immediately induced by δ_{ij}^+ when agent j comes within range of target i and starts collecting data causing $Z_{ij}(t) > 0$ if $Z_{ij}(t) = 0$ and $X_i(t) > 0$. Finally, note that all events are directly observable during the execution of any agent trajectory, and they do not depend on our flow-based queueing model. For example, if $X_i(t)$ becomes zero, this defines event ξ_i^0 regardless of whether the corresponding queue is based on a flow or on discrete data packets; this observation is very useful in the sequel. A high-level hybrid automaton is presented in Fig. 2 for a single target i and one agent j system. This automaton becomes much more complicated once more targets and agents are included.

A. Performance Measures

Our objective is to minimize the data content of all target queues while maximizing the utilization of our agent resources. Because the optimization is over a finite time interval, a terminal cost also needs to be incorporated. Moreover, as mentioned in the introduction of this paper, our goal is to develop an event-driven approach, which will necessitate the inclusion of a metric

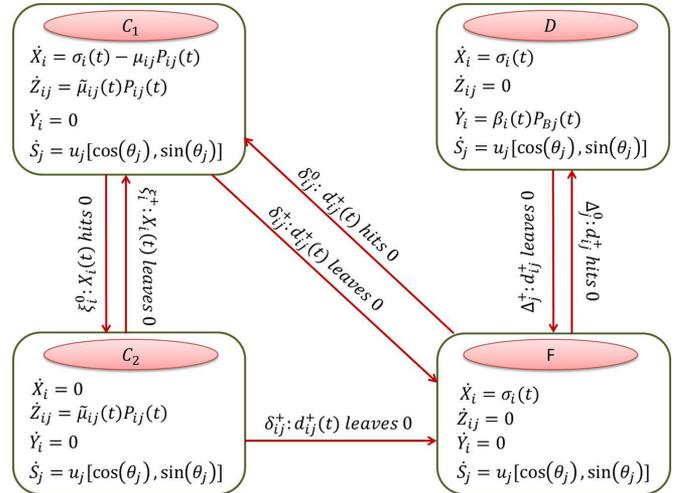


Fig. 2. One target i and one agent j hybrid automaton.

to address the event excitation issue further elaborated in this subsection. In order to make the formulation clear, we proceed in several steps, with every step discussing each component of the optimization problem.

1) *Queue Contents*: We define $J_1(t)$ to be the weighted sum of expected target queue content (recalling that $\{\sigma_i(t)\}$ are random processes):

$$J_1(t) = E \left[\sum_{i=1}^M \alpha_i X_i(t) \right] \quad (10)$$

where the weight α_i represents the relative importance factor of target i . Similarly, we define a weighted sum of expected base queue content:

$$J_2(t) = E \left[\sum_{i=1}^M \alpha_i Y_i(t) \right]. \quad (11)$$

Therefore, the first performance metric of interest is the convex combination of $J_1(t)$ and $-J_2(t)$ leading to the minimization problem:

$$\min_{\mathbf{u}(t), \boldsymbol{\theta}(t)} J(T) = \frac{1}{T} \int_0^T (qJ_1(t) - (1-q)J_2(t)) dt \quad (12)$$

where \mathbf{u} and $\boldsymbol{\theta}$ are the vectors formed by the agent speed and headings, and $q \in [0, 1]$ is a weight capturing the relative importance of collected data as opposed to delivered data.

This performance measure captures the collection and delivery of data, which are processes taking place while an agent is connected to any of the targets or the base. However, it lacks any information regarding the interaction of an agent with the environment when this agent is not connected to any target or base and is due to the fact that the environment has only a finite number of points of interest (targets). This motivates two new performance measures we introduce next.

2) *Agent Utilization*: In accessing the targets, we must ensure that the agents maximize their utilization, that is, the fraction of time spent performing a useful task by being within

range of a target or the base. Equivalently, we aim to minimize the nonproductive idling time of each agent, during which it is not visiting any target or the base. Using (9), agent j is idling when $d_{i_j}^+(t) > 0$ for all i and $d_{B_j}^+(t) > 0$. We define the idling function $I_j(t)$ as follows:

$$I_j(t) = \log \left(1 + d_{B_j}^+(t) \prod_{i=1}^M d_{i_j}^+(t) \right). \quad (13)$$

This function has the following properties. First, $I_j(t) = 0$ if and only if the product term inside the bracket is zero, that is, agent j is visiting a target or the base; otherwise, $I_j(t) > 0$. Second, $I_j(t)$ is monotonically nondecreasing in the number of targets M . The logarithmic function is selected to prevent the value of $I_j(t)$ from dominating those of $J_1(\cdot)$ and $J_2(\cdot)$ when included in a single objective function. Thus, we define

$$J_3(t) = E \left[\sum_{j=1}^N I_j(t) \right]. \quad (14)$$

Note that $I_j(t)$ is also a random variable since it is a function of the agent states $s_j(t)$, $j = 1, \dots, N$.

3) *Terminal Cost*: Since we address the data harvesting problem over a finite interval T , we define a terminal cost at T capturing the expected value of the amount of data left on board the agents:

$$J_f(T) = \frac{1}{T} E \left[\sum_{i=1}^M \sum_{j=1}^N \alpha_i Z_{ij}(T) \right]. \quad (15)$$

Clearly, the effect of this term vanishes as $T \rightarrow \infty$ as long as all $E[Z_{ij}(T)]$ remains bounded. Moreover, if we constrain trajectories to be periodic, this terminal cost may be omitted. Finally, for simplicity, we will assume that $\alpha_i = 1$ for all i .

4) *Event Excitation*: As already mentioned, our goal is to develop an event-driven approach for online trajectory optimization. In other words, we seek a controller whose actions are based on events observed during the operation of the hybrid system described earlier. Clearly, the premise of this approach is that the events involved are observable so as to “excite” the underlying event-driven controller. However, it is not always obvious that these events actually take place under every feasible control, in which case the controller may be useless. This is illustrated in Fig. 3, where two different trajectories are shown for the agent. The blue and red trajectories pass through none of the targets. Consequently, there is an infinite number of trajectories, for which the value of the objective function in (12) is given by

$$J(T) = \frac{q}{T} \int_0^T t \sigma_i(t) dt \quad (16)$$

which is simply the total amount of data generated at all targets through (5). This cannot be affected by any event-driven control action, since none of the events in Table I is excited. Clearly, the same is true for $J_3(t)$ in (14).

To address this issue, our goal is to “spread” each target cost (accumulated data) $X_i(t)$ over all $w \in S$. This will create a

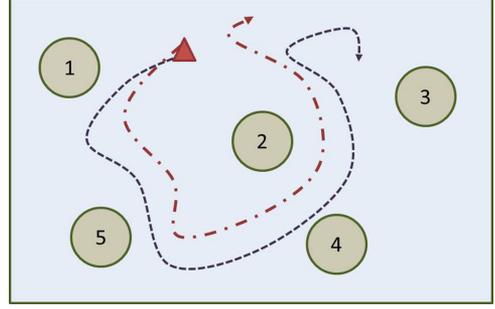


Fig. 3. Two trajectories with the same objective function value.

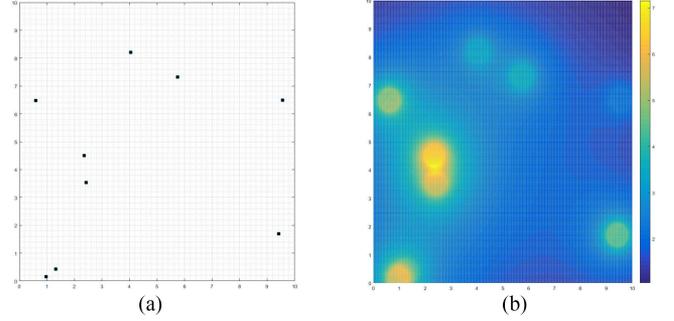


Fig. 4. R function illustration. (a) Mission space with dots as target locations. (b) R function at a sample time t .

potential field throughout the mission space. Following [25], we begin by determining the convex hull produced by the targets, since the trajectories need not go outside this polygon. Let $\mathcal{T} = \{w_1, w_2, \dots, w_M\}$ be the set of all target points. Then, their convex hull is

$$\mathcal{C} = \left\{ \sum_{i=1}^M \beta_i w_i \mid \sum_i \beta_i = 1, \forall i, \beta_i \geq 0 \right\}. \quad (17)$$

Given that $\mathcal{C} \subset S$, we seek a function $R(w, t)$ that satisfies the following property for some constants $c_i > 0$:

$$\int_{\mathcal{C}} R(w, t) dw = \sum_{i=1}^M c_i X_i(t). \quad (18)$$

Thus, $R(w, t)$ can be viewed as a time-varying density function defined for all points $w \in \mathcal{C}$, which generates a total cost equivalent to a weighted sum of the target data $X_i(t)$, $i = 1, \dots, M$. Letting $d_i^+(w) = \max(\|w - w_i\|, r_i)$, where $r_i = \min_j r_{ij}$, we then define

$$R(w, t) = \sum_{i=1}^M \frac{\alpha_i X_i(t)}{d_i^+(w)}. \quad (19)$$

Intuitively, a target’s cost (numerator above) is spread over all $w \in S$ so as to obtain the “total weighted cost density” at w . Note that $d_i^+(w)$ is defined to ensure that the target cost remains positive and fixed for all points $w \in \mathcal{C}(w_i)$. In order to illustrate this construction, Fig. 4(a) shows a sample mission space with nine target locations and Fig. 4(b) shows the value of $R(w, t)$ at a specific time t .

Proposition 1: There exist $c_i > 0$, $i = 1, \dots, M$, such that

$$\int_C R(w, t) dw = \sum_{i=1}^M c_i X_i(t). \quad (20)$$

Proof: See the appendix in [26]. ■

Using the same idea for the base, we define

$$R_{B_j}(w, t) = \frac{\sum_{i=1}^M \alpha_i Z_{ij}}{d_B^+(w)} \quad (21)$$

where $d_B^+(w) = \max(\|w_B - w\|, r_B)$ is a constant and $r_B = \min_j r_{B_j}$.

Proposition 1 asserts that the total cost due to data accumulated at targets may indeed be spread over all points in the mission space allowing an agent to “interact” with these points through the resulting potential field. In order to capture this interaction (see also [25]), we define the travel cost for an agent j to reach point w as the quadratic of the distance between them $\|s_j(t) - w\|^2$ and the total travel cost as

$$P(w, \mathbf{s}(t)) = \sum_{j=1}^N \|s_j(t) - w\|^2. \quad (22)$$

Using these definitions, we can now introduce a new performance metric:

$$J_4(t) = E \left[\sum_{j=1}^N \int_S \left(R(w, t) + R_{B_j}(w, t) \right) P_j(w, t) dw \right]. \quad (23)$$

B. Optimization Problem

We can now formulate a stochastic optimization problem **P1**, where the control variables are the agent speeds and headings denoted by the vectors $\mathbf{u}(t) = [u_1(t), \dots, u_N(t)]$ and $\boldsymbol{\theta}(t) = [\theta_1(t), \dots, \theta_N(t)]$, respectively (omitting their dependence on the full system state at t). Combining the components in (14), (15), and (23), we obtain

$$\begin{aligned} \min_{\mathbf{u}(t), \boldsymbol{\theta}(t)} J(T) &= \frac{1}{T} \int_0^T \left(\frac{q}{M_X} J_1(t) - \frac{(1-q)}{M_Y} J_2(t) \right. \\ &\quad \left. + \frac{1}{M_I} J_3(t) + \frac{1}{M_R} J_4(t) \right) dt + \frac{1}{M_Z} J_f(T) \end{aligned} \quad (24)$$

where we introduce the normalizing factors M_X , M_Y , M_I , M_R , and M_Z . This normalization ensures that all different components of $J(T)$ are in the same range so that none of them may dominate any other. We use an upper bound for the value of each component as follows, where we assume that $\sigma_i(0) > 0$ w.p. 1:

$$M_X = M_Y = M_Z = T \sum_i \sigma_i(0) \quad (25)$$

$$M_I = \log \left(1 + \sqrt{L_1^2 + L_2^2} \right)^{M+1} \quad (26)$$

$$M_R = \frac{TL_1L_2(L_1^2 + L_2^2)}{r} \sum_i \sigma_i(0), r = \frac{\sum_i r_i}{M} \quad (27)$$

where L_1 and L_2 define the size of the rectangular mission space (the normalization factors can be adapted to different mission space shapes). Observe that an unattainable lower bound of the total objective function is $-(1-q)$, which occurs if $J_1 = J_3 = J_4 = 0$ and J_2 is at its maximum of M_Y . If $q = 0$, then the lower bound is at its minimum of -1 .

III. OPTIMIZATION METHODOLOGY

In this section, we consider problem **P1** in a setting, where all data arrival processes are deterministic, so that all expectations in (10)–(15) degenerate to their arguments. We will see in Section III-A that the gradient evaluation process (in particular, Corollary 5) leading to the agent trajectory generation is independent of all random data arrival processes $\sigma_i(t)$, $i = 1, \dots, M$. This will enable us to return to the stochastic problem formulation at that point. We proceed with a standard Hamiltonian analysis leading to a TPBVP [27], where the states and costates are known at $t = 0$ and $t = T$, respectively. We define the costate vector associated with (1):

$$\begin{aligned} \boldsymbol{\lambda}(t) &= [\lambda_1(t), \dots, \lambda_M(t), \gamma_1(t), \dots, \gamma_M(t), \\ &\quad \phi_{11}(t), \dots, \phi_{MN}(t), \eta_1^x(t), \eta_1^y(t), \dots, \eta_N^x(t), \eta_N^y(t)]. \end{aligned} \quad (28)$$

The Hamiltonian is

$$\begin{aligned} H(\mathbf{X}, \boldsymbol{\lambda}, \mathbf{u}, \boldsymbol{\theta}) &= \frac{1}{T} [qJ_1(t) - (1-q)J_2(t) + J_3(t) + J_4(t)] \\ &\quad + \sum_i \lambda_i(t) \dot{X}_i(t) + \sum_i \gamma_i(t) \dot{Y}_i(t) + \sum_i \sum_j \phi_{ij}(t) \dot{Z}_{ij}(t) \\ &\quad + \sum_j (\eta_j^x(t) u_j(t) \cos \theta_j(t) + \eta_j^y(t) u_j(t) \sin \theta_j(t)) \end{aligned} \quad (29)$$

where the costate equations are

$$\begin{aligned} \dot{\lambda}_i(t) &= - \frac{\partial H}{\partial X_i} \\ &= - \frac{1}{T} \left[\frac{q}{M_X} + \frac{1}{M_R} \sum_j \int_S \frac{\alpha_i P_j(w, t)}{d_i^+(w)} dw \right] \lambda_i(T) = 0 \\ \dot{\gamma}_i(t) &= - \frac{\partial H}{\partial Y_i} = \frac{1-q}{TM_Y} \quad \gamma_i(T) = 0 \\ \dot{\phi}_{ij}(t) &= - \frac{\partial H}{\partial Z_{ij}} = - \frac{1}{M_R} \int_S \frac{\alpha_i P_j(w, t)}{d_B^+(w)} dw \\ \phi_{ij}(T) &= \frac{\partial J_f}{\partial Z_{ij}} \Big|_T \\ \dot{\eta}_j^x(t) &= - \frac{\partial H}{\partial s_j^x} \\ &= - \left[\frac{1}{TM_I} \frac{\partial I_j(t)}{\partial s_j^x} + \frac{1}{TM_R} \sum_j \int_S (R(w, t) \right. \\ &\quad \left. + R_{B_j}(w, t)) \frac{\partial P_j(w, t)}{\partial s_j^x} dw \right. \\ &\quad \left. + \sum_i \frac{\partial}{\partial s_j^x} \lambda_i(t) \dot{X}_i(t) + \sum_i \frac{\partial}{\partial s_j^x} \gamma_i(t) \dot{Y}_i(t) \right] \end{aligned}$$

$$\begin{aligned}
 \dot{\eta}_j^y(t) &= - \left[\frac{\partial H}{\partial s_j^y} \right. \\
 &= - \left[\frac{1}{TM_I} \frac{\partial I_j(t)}{\partial s_j^y} + \frac{1}{TM_R} \sum_j \int_S (R(w, t) \right. \\
 &\quad \left. + R_{B_j}(w, t)) \frac{\partial P_j(w, t)}{\partial s_j^y} dw \right. \\
 &\quad \left. + \sum_i \frac{\partial}{\partial s_j^y} \lambda_i(t) \dot{X}_i(t) + \sum_i \frac{\partial}{\partial s_j^y} \gamma_i(t) \dot{Y}_i(t) \right. \\
 &\quad \left. + \sum_i \frac{\partial}{\partial s_j^y} \phi_{ij}(t) \dot{Z}_{ij}(t) \right] \\
 \eta_j^x(T) &= \eta_j^y(T) = 0.
 \end{aligned}$$

From (29), after some trigonometric manipulations, we get

$$\begin{aligned}
 H(\mathbf{X}, \boldsymbol{\lambda}, \mathbf{u}, \boldsymbol{\theta}) &= \frac{1}{T} [qJ_1(t) - (1-q)J_2(t) + J_3(t) + J_4(t)] \\
 &+ \sum_i \lambda_i(t) \dot{X}_i(t) + \sum_i \gamma_i(t) \dot{Y}_i(t) + \sum_i \sum_j \phi_{ij}(t) \dot{Z}_{ij}(t) \\
 &+ \sum_j u_j(t) \operatorname{sgn} \eta_j^y(t) \sqrt{\eta_j^x(t)^2 + \eta_j^y(t)^2} \sin(\theta_j(t) + \psi_j(t))
 \end{aligned} \quad (30)$$

where $\tan \psi_j(t) = \frac{\eta_j^x(t)}{\eta_j^y(t)}$ for $\eta_j^y(t) \neq 0$ and $\psi_j(t) = \operatorname{sgn} \eta_j^x(t) \frac{\pi}{2}$ if $\eta_j^y(t) = 0$. Applying the Pontryagin principle to (29) with $(\mathbf{u}^*, \boldsymbol{\theta}^*)$ being the optimal control, we have

$$H(\mathbf{X}^*, \boldsymbol{\lambda}^*, \mathbf{u}^*, \boldsymbol{\theta}^*) = \min_{\mathbf{u}(t), \boldsymbol{\theta}(t)} H(\mathbf{X}, \boldsymbol{\lambda}, \mathbf{u}, \boldsymbol{\theta}). \quad (31)$$

From (30), we see that we can always set the control $\theta_j(t)$ to ensure that $\operatorname{sgn} \eta_j^y(t) \sin(\theta_j(t) + \psi_j(t)) < 0$. Hence, recalling that $0 \leq u_j(t) \leq 1$,

$$u_j^*(t) = 1 \quad (32)$$

and

$$\begin{aligned}
 \sin(\theta_j^*(t) + \psi_j(t)) &= 1, \text{ if } \mu_j^y(t) < 0 \\
 \sin(\theta_j^*(t) + \psi_j(t)) &= -1, \text{ if } \mu_j^y(t) > 0.
 \end{aligned} \quad (33)$$

Following the Hamiltonian definition in (29), we have

$$\frac{\partial H}{\partial \theta_j} = -\eta_j^x(t) u_j(t) \sin \theta_j(t) + \eta_j^y(t) u_j(t) \cos \theta_j(t) \quad (34)$$

and setting $\frac{\partial H}{\partial \theta_j} = 0$, the optimal heading $\theta_j^*(t)$ should satisfy

$$\tan \theta_j^*(t) = \frac{\eta_j^y(t)}{\eta_j^x(t)}. \quad (35)$$

Since $u_j^*(t) = 1$, we only need to evaluate $\theta_j^*(t)$ for all $t \in [0, T]$. This is accomplished by discretizing the problem in time and numerically solving a TPBVP with a forward integration

of the state and a backward integration of the costate. Solving this problem becomes intractable as the number of agents and targets grows. The fact that we are dealing with a hybrid dynamic system further complicates the solution of a TPBVP. On the other hand, it enables us to make use of IPA [24] to carry out the parametric trajectory optimization process discussed in the next section. In particular, we propose a parameterization of agent trajectories allowing us to utilize IPA to obtain an unbiased estimate for the objective function gradient with respect to the trajectory parameters.

A. Agent Trajectory Parameterization and Optimization

The key idea is to represent each agent's trajectory through general parametric equations

$$s_j^x(t) = f(\Theta_j, \rho_j(t)), \quad s_j^y(t) = g(\Theta_j, \rho_j(t)) \quad (36)$$

where the function $\rho_j(t)$ controls the position of the agent on its trajectory at time t and Θ_j is a vector of parameters controlling the shape and location of the agent j trajectory. Let $\Theta = [\Theta_1, \dots, \Theta_N]$. We now replace problem **P1** in (25) by problem **P2**:

$$\begin{aligned}
 \min_{\Theta \in F_\Theta} \frac{1}{T} \int_0^T [qJ_1(\Theta, t) - (1-q)J_2(\Theta, t) + J_3(\Theta, t) \\
 + J_4(\Theta, t)] dt + J_f(\Theta, T)
 \end{aligned} \quad (37)$$

where we return to allowing arbitrary stochastic data arrival processes $\{\sigma_i(t)\}$ so that **P2** is a parametric stochastic optimization problem with the feasible parameter set F_Θ appropriately defined depending on (36). The cost function in (37) is written as

$$J(\Theta, T; \mathbf{X}(\Theta, 0)) = E[\mathcal{L}(\Theta, T; \mathbf{X}(\Theta, 0))]$$

where $\mathcal{L}(\Theta, T; \mathbf{X}(\Theta, 0))$ is a sample function defined over $[0, T]$ and $\mathbf{X}(\Theta, 0)$ is the initial value of the state vector. For convenience, in the following, we will use \mathcal{L}_i , $i = 1, \dots, 4$, and \mathcal{L}_f to denote sample functions of J_i , $i = 1, \dots, 4$, and J_f , respectively. Note that, in (37), we suppress the dependence of the four objective function components on the controls $\mathbf{u}(t)$ and $\boldsymbol{\theta}(t)$ and stress instead their dependence on the parameter vector Θ .

In the rest of this paper, we will consider two families of trajectories motivated by a similar approach used in the multiagent persistent monitoring problem in [23]: *elliptical* trajectories and a more general *Fourier series* trajectory representation better suited for nonuniform target topologies. The hybrid dynamics of the data harvesting system allow us to apply the theory of IPA [24] to obtain online the gradient of the sample function $\mathcal{L}(\Theta, T; \mathbf{X}(\Theta, 0))$ with respect to Θ . The value of the IPA approach is twofold: 1) The sample gradient $\nabla \mathcal{L}(\Theta, T)$ can be obtained online based on observable sample path data *only*; and 2) $\nabla \mathcal{L}(\Theta, T)$ is an unbiased estimate of $\nabla J(\Theta, T)$ under mild technical conditions, as shown in [24]. Therefore, we can use $\nabla \mathcal{L}(\Theta, T)$ in a standard gradient-based stochastic optimization algorithm

$$\Theta^{l+1} = \Theta^l - \nu_l \nabla \mathcal{L}(\Theta^l, T), \quad l = 0, 1, \dots \quad (38)$$

to converge (at least locally) to an optimal parameter vector Θ^* with a proper selection of a step-size sequence $\{\nu_l\}$ [28]. We emphasize that this process is carried out *online*, that is, the gradient is evaluated by observing a trajectory with given Θ over $[0, T]$ and is iteratively adjusted until convergence is attained.

1) *IPA Calculus Review and Implementation:* Based on the events defined earlier, we will specify event time derivative and state derivative dynamics for each mode of the hybrid system. In this process, we will use the IPA notation from [24] so that τ_k is the k th event time in an observed sample path of the hybrid system and $\tau'_k = \frac{d\tau_k}{d\Theta}$, $\mathcal{X}'(t) = \frac{d\mathcal{X}}{d\Theta}$ are the Jacobian matrices of partial derivatives with respect to all components of the controllable parameter vector Θ . Throughout the analysis, we will be using $(\cdot)'$ to show such derivatives. We will also use $f_k(t) = \frac{d\mathcal{X}}{dt}$ to denote the state dynamics in effect over an interevent time interval $[\tau_k, \tau_{k+1})$. We review next the three fundamental IPA equations from [24], based on which we will proceed.

First, events may be classified as exogenous or endogenous. An event is exogenous if its occurrence time is independent of the parameter Θ ; hence, $\tau'_k = 0$. Otherwise, an endogenous event takes place when a condition $g_k(\Theta, \mathcal{X}) = 0$ is satisfied, that is, the state $\mathcal{X}(t)$ reaches a switching surface described by $g_k(\Theta, \mathcal{X})$. In this case, it is shown in [24] that

$$\tau'_k = - \left(\frac{dg_k}{d\mathcal{X}} f_k(\tau_k^-) \right)^{-1} \left(g'_k + \frac{dg_k}{d\Theta} \mathcal{X}'(\tau_k^-) \right) \quad (39)$$

as long as $\frac{\partial g_k}{\partial \mathcal{X}} f_k(\tau_k^-) \neq 0$. It is also shown in [24] that the state derivative $\mathcal{X}'(t)$ satisfies

$$\frac{d}{dt} \mathcal{X}'(t) = \frac{df_k}{d\mathcal{X}} \mathcal{X}'(t) + f'_k(t), \quad t \in [\tau_k, \tau_{k+1}) \quad (40)$$

$$\mathcal{X}'(\tau_k^+) = \mathcal{X}'(\tau_k^-) + [f_{k-1}(\tau_k^-) - f_k(\tau_k^+)] \tau'_k \quad (41)$$

Then, $\mathcal{X}'(t)$ for $t \in [\tau_k, \tau_{k+1})$ is calculated through

$$\mathcal{X}'(t) = \mathcal{X}'(\tau_k^+) + \int_{\tau_k}^t \frac{d}{dt} \mathcal{X}'(t) dt. \quad (42)$$

Table I contains all possible *endogenous* event types for our hybrid system. To these, we add *exogenous* events $\kappa_i, i = 1, \dots, M$, to allow for possible discontinuities (jumps) in the random processes $\{\sigma_i(t)\}$ which affect the sign of $\sigma_i(t) - \mu_{ij} p_{ij}(t)$ in (5). We will use the notation $e(\tau_k)$ to denote the event type occurring at $t = \tau_k$ with $e(\tau_k) \in E$, the event set consisting of all endogenous and exogenous events. Finally, we make the following assumption, which is needed in guaranteeing the unbiasedness of the IPA gradient estimates: **(A6)** Two events occur at the same time w.p. 0 unless one is directly caused by the other.

2) *Objective Function Gradient:* The sample function gradient $\nabla \mathcal{L}(\Theta, T)$ needed in (38) is obtained from (37) assuming a

total number of K events over $[0, T]$ with $\tau_{K+1} = T$ and $\tau_0 = 0$:

$$\begin{aligned} \nabla \mathcal{L}(\Theta, T; \mathbf{X}(\Theta; 0)) &= \\ &= \frac{1}{T} \nabla \left[\int_0^T (q\mathcal{L}_1(\Theta, t) - (1-q)\mathcal{L}_2(\Theta, t) + \mathcal{L}_3(\Theta, t) \right. \\ &\quad \left. + \mathcal{L}_4(\Theta, t)) dt \right] + \nabla \mathcal{L}_f(\Theta, T) \\ &= \frac{1}{T} \nabla \left[\sum_{k=0}^K \int_{\tau_k}^{\tau_{k+1}} (q\mathcal{L}_1(\Theta, t) - (1-q)\mathcal{L}_2(\Theta, t) + \mathcal{L}_3(\Theta, t) \right. \\ &\quad \left. + \mathcal{L}_4(\Theta, t)) dt \right] + \nabla \mathcal{L}_f(\Theta, T) \\ &= \frac{1}{T} \left[\sum_{k=0}^K q \left(\int_{\tau_k}^{\tau_{k+1}} \nabla \mathcal{L}_1(\Theta, t) dt + \mathcal{L}_1(\Theta, \tau_{k+1}) \tau'_{k+1} \right. \right. \\ &\quad \left. \left. - \mathcal{L}_1(\Theta, \tau_k) \tau'_k \right) \right. \\ &\quad \left. - (1-q) \left(\int_{\tau_k}^{\tau_{k+1}} \nabla \mathcal{L}_2(\Theta, t) dt + \mathcal{L}_2(\Theta, \tau_{k+1}) \tau'_{k+1} \right. \right. \\ &\quad \left. \left. - \mathcal{L}_2(\Theta, \tau_k) \tau'_k \right) \right. \\ &\quad \left. + \left(\int_{\tau_k}^{\tau_{k+1}} \nabla \mathcal{L}_3(\Theta, t) dt + \mathcal{L}_3(\Theta, \tau_{k+1}) \tau'_{k+1} \right. \right. \\ &\quad \left. \left. - \mathcal{L}_3(\Theta, \tau_k) \tau'_k \right) \right. \\ &\quad \left. + \left(\int_{\tau_k}^{\tau_{k+1}} \nabla \mathcal{L}_4(\Theta, t) dt + \mathcal{L}_4(\Theta, \tau_{k+1}) \tau'_{k+1} \right. \right. \\ &\quad \left. \left. - \mathcal{L}_4(\Theta, \tau_k) \tau'_k \right) \right] + \nabla \mathcal{L}_f(\Theta, T) \\ &= \frac{1}{T} \left[\sum_{k=0}^K \int_{\tau_k}^{\tau_{k+1}} (q\nabla \mathcal{L}_1(\Theta, t) - (1-q)\nabla \mathcal{L}_2(\Theta, t) \right. \\ &\quad \left. + \nabla \mathcal{L}_3(\Theta, t) + \nabla \mathcal{L}_4(\Theta, t)) dt \right] + \nabla \mathcal{L}_f(\Theta, T). \quad (43) \end{aligned}$$

The last step follows from the continuity of the state variables, which causes adjacent limit terms in the sum to cancel out. Therefore, $\nabla \mathcal{L}(\Theta, T)$ does not have any direct dependence on any τ'_k ; this dependence is indirect through the state derivatives involved in the four individual gradient terms.

Referring to (10), the first term in (43) involves $\nabla \mathcal{L}_1(\Theta, t)$, which is as a sum of $X'_i(t)$ derivatives. Similarly, $\nabla \mathcal{L}_2(\Theta, t)$ is a sum of $Y'_i(t)$ derivatives and $\nabla \mathcal{L}_f(\Theta, T)$ requires only $Z'_{ij}(T)$. The third term, $\nabla \mathcal{L}_3(\Theta, t)$, requires derivatives of $I_j(t)$ in (13), which depend on the derivatives of the max function in (9) and the agent state derivatives $s'_j(t)$ with respect to Θ . The term $\nabla \mathcal{L}_4(\Theta, t)$ needs the values of $X'_i(t)$ and $Z'_{ij}(t)$. The gradients of the last two terms are derived in the appendix in [26]. Possible discontinuities in these derivatives occur when any of the last four events in Table I takes place.

In summary, the evaluation of (43) requires the state derivatives $X'_i(t)$, $Z'_{ij}(t)$, $Y'_i(t)$, and $s'_j(t)$. The latter are easily obtained for any specific choice of f and g in (36) and are shown in [26, Appendix II]. The former require a rather laborious use of (39)–(41), which, however, reduces to a simple set of state derivative dynamics as shown next.

Proposition 2: After an event occurrence at $t = \tau_k$, the state derivatives $X'_i(\tau_k^+)$, $Y'_i(\tau_k^+)$, $Z'_{ij}(\tau_k^+)$, with respect to the controllable parameter Θ satisfy the following:

$$X'_i(\tau_k^+) = \begin{cases} 0, & \text{if } e(\tau_k) = \xi_i^0 \\ X'_i(\tau_k^-) - \mu_{il} p_{il}(\tau_k) \tau'_k, & \text{if } e(\tau_k) = \delta_{ij}^+ \\ X'_i(\tau_k^-), & \text{otherwise} \end{cases}$$

where $l \neq j$ with $p_{il}(\tau_k) > 0$ if such l exists and $\tau'_k = \frac{\partial d_{ij}(s_j)}{\partial s_j} s'_j \left(\frac{\partial d_{ij}(s_j)}{\partial s_j} \dot{s}_j(\tau_k) \right)^{-1}$.

$$Y'_i(\tau_k^+) = \begin{cases} Y'_i(\tau_k^-) + Z'_{ij}(\tau_k^-), & \text{if } e(\tau_k) = \zeta_{ij}^0 \\ Y'_i(\tau_k^-), & \text{otherwise} \end{cases}$$

$$Z'_{ij}(\tau_k^+) = \begin{cases} 0, & \text{if } e(\tau_k) = \zeta_{ij}^0 \\ Z'_{ij}(\tau_k^-) + X'_i(\tau_k^-), & \text{if } e(\tau_k) = \xi_i^0 \\ Z'_{ij}(\tau_k^-), & \text{otherwise} \end{cases}$$

where $e(\tau_k) = \xi_i^0$ occurs when j is connected to target i .

Proof: See [26, eqs. (77), (80), (83), (88), (89), (91), (94), (96) in Appendix IV]. ■

This result shows that only three of the events in E can actually cause discontinuous changes to the state derivatives. Furthermore, note that $X'_i(t)$ is reset to zero after a ξ_i^0 event. Moreover, when such an event occurs, note that $Z'_{ij}(t)$ is coupled to $X'_i(t)$, similarly for $Z'_{ij}(t)$ and $Y'_i(t)$ when event ζ_{ij}^0 occurs, showing that perturbations in Θ can only propagate to an adjacent queue when that queue is emptied.

Proposition 3: The state derivatives $X'_i(\tau_{k+1}^-)$ and $Y'_i(\tau_{k+1}^-)$ with respect to the controllable parameter Θ satisfy the following after an event occurrence at $t = \tau_k$:

$$X'_i(\tau_{k+1}^-) = \begin{cases} 0, & \text{if } e(\tau_k) = \xi_i^0 \\ X'_i(\tau_k^+) - \int_{\tau_k}^{\tau_{k+1}} \mu_{ij} p'_{ij}(u) du, & \text{otherwise} \end{cases}$$

$$Y'_i(\tau_{k+1}^-) = Y'_i(\tau_k^+) + \int_{\tau_k}^{\tau_{k+1}} \beta'_i(u) du$$

where j is such that $p_{ij}(t) > 0$, $t \in [\tau_k, \tau_{k+1}]$.

Proof: See [26, eqs. (76), (79), and (81) in Appendix IV]. ■

Proposition 4: The state derivatives $Z'_{ij}(\tau_{k+1}^+)$ with respect to the controllable parameter Θ satisfy the following after an event occurrence at $t = \tau_k$:

1) If j is connected to target i ,

$$\begin{aligned} Z'_{ij}(\tau_{k+1}^+) &= \begin{cases} Z'_{ij}(\tau_k^+), & \text{if } e(\tau_k) = \xi_i^0, \zeta_{ij}^0 \text{ or } \delta_{ij}^+ \\ Z'_{ij}(\tau_k^+) + \int_{\tau_k}^{\tau_{k+1}} \mu_{ij} p'_{ij}(u) du, & \text{otherwise.} \end{cases} \end{aligned}$$

2) If j is connected to B with $Z_{ij}(\tau_k) > 0$,

$$Z'_{ij}(\tau_{k+1}^+) = Z'_{ij}(\tau_k^+) - \int_{\tau_k}^{\tau_{k+1}} \beta_{ij} p'_{Bj}(u) du.$$

3) Otherwise, $Z'_{ij}(\tau_{k+1}^+) = Z'_{ij}(\tau_k^+)$.

Proof: See [26, eqs. (84), (85), (92), (99) in Appendix IV]. ■

Corollary 5: The state derivatives $X'_i(t)$, $Z'_{ij}(t)$, and $Y'_i(t)$ with respect to the controllable parameter Θ are independent of the random data arrival processes $\{\sigma_i(t)\}$, $i = 1, \dots, M$.

Proof: The result for any $t = \tau_k$ follows directly from the three propositions. For all other $t \in (\tau_k, \tau_{k+1}]$, the result follows from [26, eqs. (78), (80), and (83) in Appendix IV]. ■

There are a few important consequences of these results. First, as the corollary asserts, one can apply IPA regardless of the characteristics of the random processes $\{\sigma_i(t)\}$. The corollary ensures that the independence applies not only at event times τ_k , but all $t \in (\tau_k, \tau_{k+1}]$, as can be seen in [26, Appendix IV]. This robustness property does not mean that these processes do not affect the values of the $X'_i(t)$, $Z'_{ij}(t)$, and $Y'_i(t)$; this effect is captured through the observable event times τ_k , $k = 1, 2, \dots$, which are an integral part of the gradient evaluation.

Second, the IPA estimation process is event-driven: $X'_i(\tau_k^+)$, $Y'_i(\tau_k^+)$, and $Z'_{ij}(\tau_k^+)$ are evaluated at event times and then used as initial conditions for the evaluations of $X'_i(\tau_{k+1}^-)$, $Y'_i(\tau_{k+1}^-)$, and $Z'_{ij}(\tau_{k+1}^-)$ along with the integrals appearing in Propositions 3 and 4, which can also be evaluated at $t = \tau_{k+1}$. Consequently, this approach is scalable in the number of events in the system as the number of agents and targets increases.

Third, despite the elaborate derivations in the appendix in [26], the actual implementation reflected by the three propositions is simple. Finally, returning to (43), note that the integrals involving $\nabla \mathcal{L}_1(\Theta, t)$ and $\nabla \mathcal{L}_2(\Theta, t)$ are directly obtained from $X'_i(t)$ and $Y'_i(t)$, the integral involving $\nabla \mathcal{L}_3(\Theta, t)$ is obtained from straightforward differentiation of (13), and the final term is obtained from $Z'_{ij}(T)$.

3) *Objective Function Optimization:* This is carried out using (38) with an appropriate diminishing step size sequence.

a) *Elliptical trajectories:* Elliptical trajectories are described by their center coordinates, minor and major axes, and orientation. Agent j 's position $s_j(t) = [s_j^x(t), s_j^y(t)]$ follows the general parametric equation of the ellipse:

$$\begin{aligned} s_j^x(t) &= A_j + a_j \cos \rho_j(t) \cos \phi_j - b_j \sin \rho_j(t) \sin \phi_j \\ s_j^y(t) &= B_j + a_j \cos \rho_j(t) \sin \phi_j + b_j \sin \rho_j(t) \cos \phi_j. \end{aligned} \quad (44)$$

Here, $\Theta_j = [A_j, B_j, a_j, b_j, \phi_j]$, where A_j and B_j are the coordinates of the center, a_j and b_j are the major and minor axis, respectively, while $\phi_j \in [0, \pi)$ is the ellipse orientation, which is defined as the angle between the x -axis and the major axis of the ellipse. The time-dependent parameter $\rho_j(t)$ is the eccentric anomaly of the ellipse. Since the agent is moving with constant speed of 1 on this trajectory from (32), we have $\dot{s}_j^x(t)^2 + \dot{s}_j^y(t)^2 = 1$, which gives

$$\dot{\rho}_j(t) = \left[\frac{(a \sin \rho_j(t) \cos \phi_j + b_j \cos \rho_j(t) \sin \phi_j)^2}{+(a \sin \rho_j(t) \sin \phi_j - b_j \cos \rho_j(t) \cos \phi_j)^2} \right]^{-\frac{1}{2}}.$$

In the data harvesting problem, trajectories that do not pass through the base are inadmissible since there is no delivery of data. Therefore, we add a constraint to force the ellipse to pass

through $w_B = [w_B^x, w_B^y]$, where

$$\begin{aligned} w_B^x &= A_j + a_j \cos \rho_j(t) \cos \phi_j - b_j \sin \rho_j(t) \sin \phi_j \\ w_B^y &= B_j + a_j \cos \rho_j(t) \sin \phi_j + b_j \sin \rho_j(t) \cos \phi_j. \end{aligned} \quad (45)$$

Using the fact that $\sin^2 \rho(t) + \cos^2 \rho(t) = 1$, we define a quadratic constraint term added to $J(\Theta, T; \mathbf{X}(\Theta, 0))$ with a sufficiently large multiplier. This can ensure the optimal path passes through the base location w_B . We define $\mathcal{C}_j(\Theta_j)$:

$$\mathcal{C}_j(\Theta_j) = (1 - f_j^1 \cos^2 \phi_j - f_j^2 \sin^2 \phi_j - f_j^3 \sin 2\phi_j)^2 \quad (46)$$

where

$$\begin{aligned} f_j^1 &= \left(\frac{w_B^x - A_j}{a_j} \right)^2 + \left(\frac{w_B^y - B_j}{b_j} \right)^2, f_j^2 = \left(\frac{w_B^x - A_j}{b_j} \right)^2 \\ &+ \left(\frac{w_B^y - B_j}{a_j} \right)^2, f_j^3 = \frac{(b_j^2 - a_j^2)(w_B^x - A_j)(w_B^y - B_j)}{a_j^2 b_j^2}. \end{aligned}$$

Multiple visits to the base may be needed during the mission time $[0, T]$. We can capture this by allowing an agent trajectory to consist of a sequence of admissible ellipses. For each agent, we define \mathcal{E}_j as the number of ellipses in its trajectory. The parameter vector Θ_j^κ with $\kappa = 1, \dots, \mathcal{E}_j$ defines the κ th ellipse in agent j 's trajectory and \mathcal{T}_j^κ is the time that agent j completes ellipse κ . Therefore, the location of each agent is described through κ during $[\mathcal{T}_j^{\kappa-1}, \mathcal{T}_j^\kappa]$, where $\mathcal{T}_j^0 = 0$. Since we cannot optimize over all possible \mathcal{E}_j for all agents, an iterative process needs to be performed in order to find the optimal number of segments in each agent's trajectory. At each step, we fix \mathcal{E}_j and find the optimal trajectory with that many segments. The process is stopped once the optimal trajectory with \mathcal{E}_j segments is no better than the optimal one with $\mathcal{E}_j - 1$ segments (obviously, this is generally not a globally optimal solution). We can now formulate the parametric optimization problem $\mathbf{P2}_e$, where $\Theta_j = [\Theta_j^1, \dots, \Theta_j^{\mathcal{E}_j}]$ and $\Theta = [\Theta_1, \dots, \Theta_N]$:

$$\begin{aligned} \min_{\Theta \in F_\Theta} J_e &= \frac{1}{T} \int_0^T [qJ_1(\Theta, t) - (1-q)J_2(\Theta, t) + J_3(\Theta, t) \\ &+ J_4(\Theta, t)] dt + M_C \sum_{j=1}^N \mathcal{C}_j(\Theta_j) + J_f(\Theta, T) \end{aligned} \quad (47)$$

where M_C is a large multiplier. The evaluation of $\nabla \mathcal{C}_j$ is straightforward and does not depend on any event (details are shown in [26, Appendix II]).

b) Fourier series trajectories: The elliptical trajectories are limited in shape and may not be able to cover many targets in a mission space. Thus, we next parameterize the trajectories using a Fourier series representation of closed curves [29]. Using a Fourier series function for f and g in (36), agent j 's trajectory

can be described as follows with base frequencies f_j^x and f_j^y :

$$\begin{aligned} s_j^x(t) &= a_{0,j} + \sum_{n=1}^{\Gamma_j^x} a_{n,j} \sin(2\pi n f_j^x \rho_j(t) + \phi_{n,j}^x) \\ s_j^y(t) &= b_{0,j} + \sum_{n=1}^{\Gamma_j^y} b_{n,j} \sin(2\pi n f_j^y \rho_j(t) + \phi_{n,j}^y). \end{aligned} \quad (48)$$

The parameter $\rho(t) \in [0, 2\pi]$, similar to elliptical trajectories, represents the position of the agent along the trajectory. In this case, forcing a Fourier series curve to pass through the base is easier. For simplicity, we assume a trajectory to start at the base and set $s_j^x(0) = w_B^x$, $s_j^y(0) = w_B^y$. Assuming $\rho(0) = 0$, with no loss of generality, we can calculate the zero frequency terms by means of the remaining parameters:

$$\begin{aligned} a_{0,j} &= w_B^x - \sum_{n=1}^{\Gamma_j^x} a_{n,j} \sin(\phi_{n,j}^x), b_{0,j} \\ &= w_B^y - \sum_{n=1}^{\Gamma_j^y} b_{n,j} \sin(\phi_{n,j}^y). \end{aligned}$$

The parameter vector for agent j is $\Theta_j = [f_j^x, a_{0,j}, \dots, a_{\Gamma_j^x}, b_{0,j}, \dots, b_{\Gamma_j^y}, \phi_{1,j}, \dots, \phi_{\Gamma_j^x}, \xi_{1,j}, \dots, \xi_{\Gamma_j^y}]$ and $\Theta = [\Theta_1, \dots, \Theta_N]$. Note that the shape of the curve is fully captured by the ratio f_j^x / f_j^y , so that one of these two parameters can be kept constant. For the Fourier trajectories, the fact that $\mathbf{u}_j^* = 1$ allows us to calculate $\dot{\rho}_j(t)$ as follows:

$$\begin{aligned} \dot{\rho}_j(t) &= \frac{1}{2\pi} \\ &\times \left[\left(f_j^x \sum_{n=1}^{\Gamma_j^x} a_{n,j} n \cos(2\pi f_j^x \rho_j(t) + \phi_{n,j}^x) \right)^2 \right. \\ &\left. + \left(f_j^y \sum_{n=1}^{\Gamma_j^y} b_{n,j} n \cos(2\pi f_j^y \rho_j(t) + \phi_{n,j}^y) \right)^2 \right]^{-1/2}. \end{aligned}$$

Problem $\mathbf{P2}_f$ is the same as $\mathbf{P2}$, but there are no additional constraints in this case:

$$\begin{aligned} \min_{\Theta \in F_\Theta} J_f &= \frac{1}{T} \int_0^T [qJ_1(\Theta, t) - (1-q)J_2(\Theta, t) + J_3(\Theta, t) \\ &+ J_4(\Theta, t)] dt + J_f(T). \end{aligned} \quad (49)$$

IV. NUMERICAL RESULTS

In this section, numerical results are presented to illustrate our approach. The mission space S is considered to be $[0, 10] \times [0, 10]$ in all cases. The first case we consider is a small mission to obtain the TPBVP results and confirm the fact that it is not scalable to bigger problems.

In Case I, we consider a two-target and two-agent setting. We assume deterministic arrival processes with $\sigma_i = 0.5$ for all i . For (3) and (4), we have used $p(w, v) = \max(0, 1 - \frac{d(w, v)}{r})$, where r is the corresponding value of $r_{i,j}$ or $r_{B,j}$. We set $\mu_{ij} = 100$ and $\beta_{ij} = 500$ for all i and j . Other parameters used

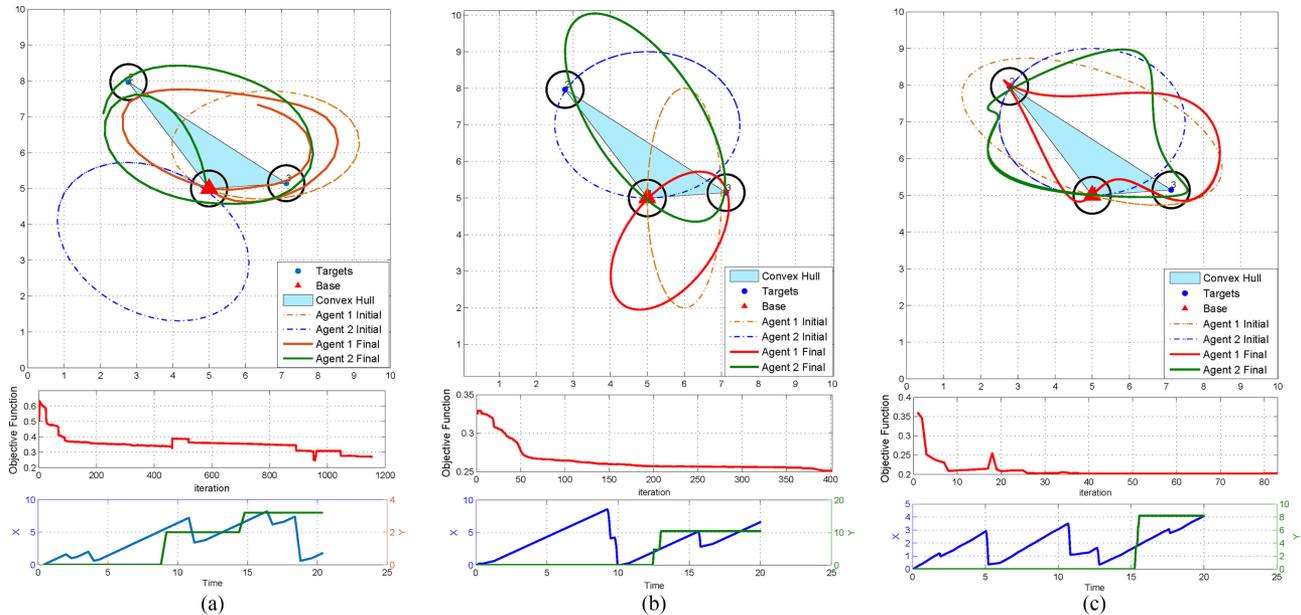


Fig. 5. Simulation results for the two-target and two-agent case. (a) TPBVP Trajectories for case I. (b) Elliptical Trajectories for case I. (c) Fourier Trajectories for case I.

TABLE II
RESULT COMPARISON FOR CASE I

Method	J^*	J_1^*	$-J_2^*$
TPBVP	0.272	0.098	-0.038
Elliptical	0.255	0.092	-0.095
Fourier	0.202	0.089	-0.095

are $q = 0.5$, $r_{ij} = r_{Bj} = 0.5$, and $T = 20$. The trajectory comparison from TPBVP, elliptical, and Fourier parametric solutions is shown in Fig. 5(a)–(c). In each figure, the trajectories are shown in the top part, while the actual objective function convergence behavior is shown in the middle graph. The lower graph shows the total amount of data at targets at any time (in blue) and the total amount of data at the base (in green).

In the TPBVP results, the main limitation is the number of the time steps in the discretization of the interval $[0, T]$, since the number of control values grows with it. To bring this into perspective, for this sample problem with $T = 20$, we considered 300 time steps, that is, 300 values for the heading of each agent need to be calculated, which brings the total number of controls to 600. In contrast, for the same problem, the total number of controls for the elliptical trajectories are ten parameters, and for the Fourier trajectories, it is 28. This explains why the TPBVP cannot be a viable solution for larger values of T . Note that, in this scenario, the total time is only 20 time steps so we can obtain a TPBVP solution. This, however, causes a poor representation for the parameterized trajectories, which are spending significant time outside the convex hull since they have not converged after only 20 time steps.

In Table II, the actual values for J^* , J_1^* , and J_2^* are shown for the three different trajectories of Fig. 5(a)–(c). It should be

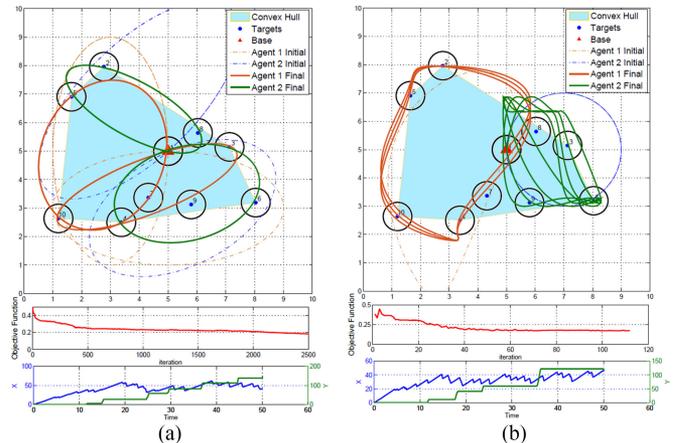


Fig. 6. Simulation results for the nine-target and two-agent case. (a) Elliptical trajectories for case II. (b) Fourier trajectories for case II.

emphasized that in all numerical results that follow, we show the normalized values of the objective function. Note that the objective is to minimize J by minimizing $J_1 - J_2$.

Next, in Case II, we consider nine targets and two agents. The base is located at the center of the mission space. We have $\sigma_i(t) = 0.5$, $\mu_{ij} = 50$, and $\beta_{ij} = 500$ for all i and j . Other parameters used are $q = 0.5$, $r_{ij} = 0.55$, $r_{Bj} = 0.65$, and $T = 50$. In Fig. 6(a), the solution with two ellipses in each agent's trajectory is shown. As can be seen, the trajectory correctly finds all the target locations and empties the target queues periodically. Fig. 6(b) shows the Fourier trajectories. The two graphs on the bottom show the objective function value and the instantaneous total content at targets and base. The results for Case II are summarized in Table III.

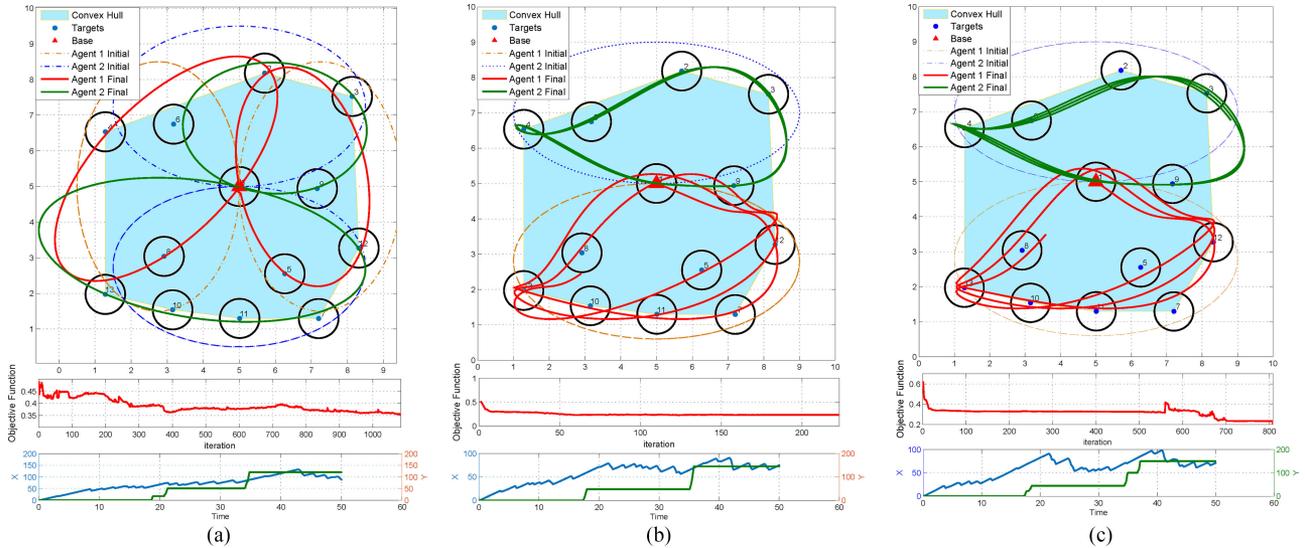


Fig. 7. Simulation results for the 12-target and two-agent case. (a) Elliptical trajectories for case III. (b) Fourier trajectories for case III. (c) Fourier trajectories for case III with stochastic arrival.

TABLE III
RESULTS COMPARISON FOR CASE II

Method	J^*	J_1^*	$-J_2^*$
<i>Elliptical</i>	0.19	0.090	-0.124
<i>Fourier</i>	0.18	0.069	-0.117

TABLE IV
RESULTS COMPARISON FOR CASE III

Method	J^*	J_1^*	$-J_2^*$
<i>Elliptical</i>	0.35	0.12	-0.09
<i>Fourier</i>	0.23	0.09	-0.1
<i>Fourier (Stochastic Arrival)</i>	0.23	0.13	-0.13

TABLE V
RESULT COMPARISON WITH PSH FOR CASE II

Method	J_1^*	$-J_2^*$
<i>PSH Sequence</i>	0.023	-0.22
<i>Elliptical Sequence</i>	0.027	-0.21
<i>Fourier Sequence</i>	0.024	-0.21

TABLE VI
RESULT COMPARISON WITH PSH FOR CASE III

Method	J_1^*	$-J_2^*$
<i>PSH Sequence</i>	0.0257	-0.21
<i>Elliptical Sequence</i>	0.0304	-0.199
<i>Fourier Sequence</i>	0.0212	-0.21

Case III has 12 targets that are uniformly distributed in the mission space. Here, we try to examine the robustness of our approach with respect to the arrival rate process at targets. We use the same parameters as in case II and solve the problem for deterministic $\sigma_i(t) = 0.5$ using the elliptical trajectories and Fourier trajectories. The same mission is simulated assuming that $\sigma_i(t)$ is a stochastic process with the piecewise linear arrival rate. The average arrival rate is kept at 0.5. The results in Fig. 7(a)–(c) and Table IV show that the Fourier parametric trajectories achieve almost the same performance by the optimization algorithm in the stochastic setting.

One point worth mentioning regarding the comparison between elliptical and Fourier series trajectories is that the elliptical ones are more convenient for environments with a more structured pattern in target placement, whereas the Fourier trajectories are better suited in less structured environments where targets are randomly placed.

V. COMPARISON WITH A GRAPH-BASED ALGORITHM

We begin with the observation that the final parametric trajectories provide a sequence of targets visits, similar to the functionality of a tour selection algorithm that uses the underlying graph topology of the mission space to determine such sequences.

We have compared the results of our approach with a graph topology algorithm called path splitter heuristic (PSH) developed in [12]. This algorithm starts with the best Hamiltonian sequence and then uses a heuristic method to divide the Hamiltonian tour into several subtours that go through a few targets and then return to the base. The algorithm then provides a sequence of these subtours for each agent. We compare the sequences from Case I and Case II in both elliptical and Fourier trajectories, and results are shown in Tables V and VI. For a fair comparison, we adopt each sequence and apply it with the system dynamics in our model, that is, an agent can collect the data once within range of a target and the data collection does not

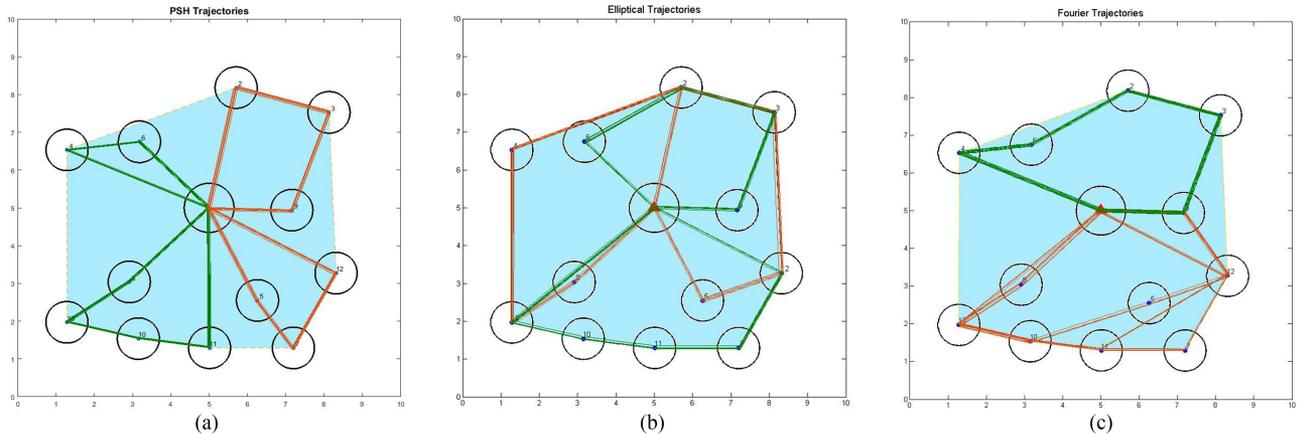


Fig. 8. Sequence comparison. (a) PSH sequences for case III. (b) Elliptical sequences for case III. (c) Fourier sequences for case III.

happen instantaneously. This, however, is not the basic modeling assumption used in PSH, where agents pick up all the data at the target instantaneously once at the target location. We compare the sum total of data at targets and the base for $T = 200$. A larger value of T is used for this comparison in order to approximate infinite time results. These sequences are shown in Fig. 8(a)–(c), where each color represents one agent trajectory. We can see from these comparisons that in the graph-based approach, targets are completely divided between agents. This generates a spatial partitioning, giving each agent full responsibility for a set of targets. However, in the trajectory planning results, in most of the cases, we see a temporal partitioning where agents can visit the same targets but at different times of the mission. This clearly allows for more robustness with respect to potential agent failures or changes in an agent’s operation. Moreover, even though the computational complexity of the PSH algorithm and the parametric trajectory optimization approach are comparable, methods such as PSH need to resolve the complete problem each time a new target may appear in the mission space. In contrast, the online event-driven parametric optimization process is a methodology designed to adapt to targets, which may randomly appear in the mission space. These results are not necessarily aiming to prove performance enhancement but to put the two approaches into contrast in terms of suitability for online and offline applications. Also, PSH agent trajectories consist of straight line segments. These are not physically realizable, given limitations on the motion of agents, which must smoothly turn direction from one target to the next, whereas the parametrical trajectories are easier to realize by most of the agents given these motion limitations.

VI. CONCLUSION

We have demonstrated a new event-driven methodology for online trajectory optimization with application in the data harvesting problem. We proposed a new performance measure that addresses the event excitation problem in event-driven controllers. The proposed optimal control problem is then formulated as a parametric trajectory optimization utilizing general function families, which can be subsequently optimized online

through the use of IPA. Several numerical results are provided for the case of elliptical and Fourier series trajectories and some properties of the solution are identified, including robustness with respect to the stochastic data generation processes and scalability in the size of the event set characterizing the underlying hybrid dynamic system.

Although the proposed methodology is focused on applying the event-driven optimization approach to the data harvesting problem, it should be noted that the new metric that was introduced in [25] to ensure event excitation allows us to generalize the methodology to other applications as well. The new metric introduces a potential field or density map over the entire mission space. This can be viewed as a probability distribution of potential targets in problems, where the exact locations of targets are unknown. Used as a prior distribution, it can be improved while the agents move within the mission space and gather more information. In addition, this density can be dynamically changing if the targets are moving and their location changes with time assuming some prior information about the target paths. This provides a tool to apply agent trajectory optimization in a much broader range of problems tracking moving points of interest.

REFERENCES

- [1] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore, “Environmental wireless sensor networks,” *Proc. IEEE*, vol. 98, no. 11, pp. 1903–1917, Nov. 2010.
- [2] R. N. Smith, M. Schwager, S. L. Smith, D. Rus, and G. S. Sukhatme, “Persistent ocean monitoring with underwater gliders: Towards accurate reconstruction of dynamic ocean processes,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1517–1524.
- [3] Z. Tang and U. Özgüner, “Motion planning for multitarget surveillance with mobile sensor agents,” *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 898–908, Oct. 2005.
- [4] M. Zhong and C. G. Cassandras, “Distributed coverage control and data collection with mobile sensor networks,” *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2445–2455, Oct. 2011.
- [5] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.
- [6] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” in *Proc. 24th Ann. INFOCOM*, 2005, pp. 1976–1984.

- [7] 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, pp. 138–154, 2014.
- [8] 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.
- [9] D. Panagou, M. Turpin, and V. Kumar, "Decentralized goal assignment and trajectory generation in multi-robot networks: A multiple Lyapunov functions approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 6757–6762.
- [10] A. T. Klesh, P. T. Kabamba, and A. R. Girard, "Path planning for cooperative time-optimal information collection," in *Proc. Amer. Control Conf.*, 2008, pp. 1991–1996.
- [11] J. L. Ny, M. A. Dahleh, E. Feron, and E. Frazzoli, "Continuous path planning for a data harvesting mobile server," in *Proc. IEEE Conf. Decision Control*, 2008, pp. 1489–1494.
- [12] R. Moazzez-Estanjini and I. C. Paschalidis, "On delay-minimized data harvesting with mobile elements in wireless sensor networks," *Ad Hoc Netw.*, vol. 10, pp. 1191–1203, 2012.
- [13] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan, "The minimum latency problem," in *Proc. 26th Annu. ACM Symp. Theory Comput.*, 1994, pp. 163–171.
- [14] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *Eur. J. Control*, vol. 11, no. 4–5, pp. 436–463, 2005.
- [15] J. K. Hart and K. Martinez, "Environmental sensor networks: A revolution in the earth system science?" *Earth-Sci. Rev.*, vol. 78, no. 34, pp. 177–191, 2006.
- [16] O. Tekdas, V. Isler, J. H. Lim, and A. Terzis, "Using mobile robots to harvest data from sensor fields," *IEEE Wireless Commun.*, vol. 16, no. 1, pp. 22–28, Feb. 2009.
- [17] W. Wei, V. Srinivasan, and C. Kee-Chaing, "Extending the lifetime of wireless sensor networks through mobile relays," *IEEE/ACM Trans. Netw.*, vol. 16, no. 5, pp. 1108–1120, Oct. 2008.
- [18] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Mar. 2005, vol. 2, pp. 1407–1418.
- [19] A. Pandya, A. Kansal, and G. Pottie, "Goodput and delay in networks with controlled mobility," in *Proc. IEEE Aerosp. Conf.*, 2008, pp. 1–8.
- [20] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Netw.*, vol. 3, pp. 325–349, 2005.
- [21] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," *Lecture Notes Comput. Sci.*, vol. 6843, pp. 243–253, 2011.
- [22] C. Chang, G. Yu, T. Wang, and C. Lin, "Path construction and visit scheduling for targets using data mules," *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 44, no. 10, pp. 1289–1300, Oct. 2014.
- [23] X. Lin and C. G. Cassandras, "An optimal control approach to the multi-agent persistent monitoring problem in two-dimensional spaces," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1659–1664, Jun. 2015.
- [24] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao, "Perturbation analysis and optimization of stochastic hybrid systems," *Eur. J. Control*, vol. 16, no. 6, pp. 642–661, 2010.
- [25] Y. Khazaeni and C. G. Cassandras, "Event excitation for event-driven control and optimization of multi-agent systems," in *Proc. 13th Int. Workshop Discrete Event Syst.*, 2016, pp. 197–202.
- [26] Y. Khazaeni and C. G. Cassandras, "Event-driven trajectory optimization for data harvesting in multi-agent systems," *arXiv:1608.06336*, 2016.
- [27] A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*. Boca Raton, FL, USA: CRC, 1975.
- [28] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York, NY, USA: Springer, 2003.
- [29] C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Comput.*, vol. C-21, no. 3, pp. 269–281, Mar. 1972.



Yasaman Khazaeni (M'13) received two B.S. degrees in electrical engineering and petroleum engineering from Sharif University of Technology, Tehran, Iran, in 2005, the M.S. degree in petroleum engineering from West Virginia University, Morgantown, WV, USA, in 2010, and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2016.

From 2010 to 2011, she was a Research Associate at West Virginia University, where she was involved in managing projects on the artificial intelligence application in reservoir engineering. She is currently a Research Staff Member with the Cognitive User Experience Laboratory, IBM Research, Cambridge, MA, USA. Her research experiences are in the area of control theory and mathematical modeling, event-driven control, stochastic systems, and artificial intelligence.

Dr. Khazaeni is a member of Phi Kappa Phi and has received several awards including the 2008 and 2009 West Virginia University distinguished graduate student, a second prize in Nico van Wingen SPE annual Scholarship, a 2011 Boston University Dean's Fellowship, and a 2014 NSF Travel Award.



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.

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, Boston, 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 more than 380 refereed papers and five 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. He has received 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.